

M E M O Nr. 115

**Unterstützung von verteilten
Softwareentwicklungsprozessen durch integrierte
Planungs-, Workflow- und
Groupware-Ansätze**



Universität Dortmund, Fachbereich Informatik
Lehrstuhl Softwaretechnologie

Volker Gruhn (Ed.)



Informatik Centrum Dortmund e.V.
Softwaretechnik

Lothar Schöpe (Ed.)

März 2001

in Zusammenarbeit mit DaimlerChrysler AG, Research and Technology, Ulm

DAIMLERCHRYSLER
Research and Technology Center

Internes Memorandum des
Lehrstuhls Software-Technologie
Prof. Dr. Ernst-Erich Doberkat
Fachbereich Informatik
Universität Dortmund

Baroper Str. 301

D- 44227 Dortmund

ISSN 0933-7725

Abstrakt

Zentraler Gegenstand dieser Ausarbeitung sind verteilte Softwareentwicklungsprozesse. Da die Softwareentwicklung in der Regel eine komplexe Aktivität ist, benötigt man Softwaresysteme, die bei der Softwareentwicklung unterstützt werden können. Der Oberbegriff dieser Softwaresysteme ist CSCW (*Computer Supported Cooperative Work*). Workflowmanagementsysteme oder Groupwaresysteme, als Vertreter der CSCW-Systeme, können zur Unterstützung der Softwareentwicklung eingesetzt werden, wobei diese einen strukturierten Entwicklungsprozess voraussetzen. Dies bedingt, dass der Entwicklungsprozess beschrieben oder modelliert werden kann. Dies kann natürlich sprachlich, semiformal oder formal mit oder ohne Verwendung einer graphischen Notation erfolgen. Der Entwicklungsprozess für Softwaresysteme lässt sich als schwach strukturierter Prozess auffassen. Ist dieser Entwicklungsprozess darüber hinaus verteilt, d.h. innerhalb der Softwareentwicklung finden einzelne Aktivitäten an verschiedenen räumlich getrennten Organisationseinheiten statt, ergeben sich daraus besondere Anforderungen an die Modellierung von Softwareentwicklungsprozessen. Es muss möglich sein, Aspekte, die aus einer Verteilung resultieren, zu beschreiben oder modellieren zu können. Aus diesen erweiterten Entwicklungsprozessen ergeben sich u.a. dann weitere Anforderungen an CSCW-Systeme.

Nach einer Einleitung, in der die wichtigen Begriffe vorgestellt und erläutert werden, erfolgt im Kapitel 2 ein Überblick über das Gebiet des CSCW. An Softwareentwicklungsprozessen nehmen Personen oder Personengruppen teil, die innerhalb der Softwareentwicklung eine oder mehrere Rollen einnehmen. Eine Rolle beschreibt die Qualifikation, über die Personen verfügen müssen und die Aufgaben und Ergebnisse, die bei Wahrnehmung dieser Rolle durch eine Person bewirkt bzw. erstellt werden müssen. Diese Rollen werden im Kapitel 3 beschrieben. Ein Anwendungsgebiet für Softwareentwicklungsprozesse wird im Kapitel 4 exemplarisch vorgestellt. Eine Vorstellung der Besonderheiten von verteilten Systemen, wobei auf die Besonderheiten des zentralen und dezentralen Managements von Prozessen unterschiedlich eingegangen wird, erfolgt im Kapitel 5. Die Vorstellung von unterschiedlichen graphischen Beschreibungssprachen für Softwareentwicklungsprozesse unter besonderer Berücksichtigung der Modellierung von verteilter Kommunikation erfolgt im Kapitel 6. Im Kapitel 7 wird ein Prozess vorgestellt, wie ein Softwaresystem im Anwendungsgebiet Electronic Commerce entwickelt werden kann. Die Beschreibung erfolgt mit einer graphischen Prozessbeschreibungssprache. Im Kapitel 8 werden die Besonderheiten, die für die Unterstützung von verteilten Softwareentwicklungsprozessen eine wesentliche Rolle spielen beschrieben. Diese Rahmenbedingungen müssen bei der Entwicklung von CSCW-Systemen zur Unterstützung von verteilten Softwareentwicklungsprozessen als Anforderungen berücksichtigt werden. Ein wesentliches Merkmal der verteilten Softwareentwicklung sind die verschiedenen Arten der zeitlichen und räumlichen Kooperation, Kommunikation und Koordination und den daraus resultierenden Anforderungen an die Modellierung von Softwareentwicklungsprozessen durch geeignete Prozessbeschreibungssprachen. Ein weiteres wesentliches Merkmal der Softwareentwicklung ist ihre i.d.R. lange Projektlaufzeit. Aus dieser langen Projektlaufzeit erwachsen Anforderungen an das Management der Arbeitsorganisation, denn externe und interne Projektteilnehmer scheiden aus oder kommen hinzu. Dies hat Auswirkungen auf die Geschäftsprozesse im allgemeinen und die Softwareentwicklungsprozesse im Besonderen; sie müssen dynamisch angepasst und geändert werden. Dies betrifft nicht nur das zentrale Prozessmodell, sondern bei verteilten Prozessen, die lokalen Prozessmodelle -, sondern auch die Prozesse, die sich schon in der Ausführung befinden und durch ein CSCW-System unterstützt werden. Im Kapitel 9 werden die Ergebnisse zusammengefasst und bewertet, sowie klassifiziert, ob und welche der vorher genannten Merkmale bei der Konzeption von CSCW-Systemen berücksichtigt werden müssen.

Inhaltverzeichnis

1	Einleitung.....	4
2	Computer Supported Cooperative Work (CSCW)	5
2.1	Computergestützte Gruppenarbeit.....	6
2.1.1	Klassifikation von CSCW-Systemen.....	8
2.1.2	Groupwaresysteme und Workflowmanagementsysteme	14
2.1.3	Informations- und Kommunikationstechniken	18
3	Arten von Softwareentwicklungsprozessen.....	19
3.1	Vorgehensmodelle	19
3.1.1	Das klassische Wasserfallmodell.....	19
3.1.2	Das iterierte Wasserfallmodell	20
3.1.3	Schnelle Prototypenerstellung.....	20
3.1.4	Evolutionäre Softwareentwicklung	21
3.1.5	Das Spiralmodell.....	21
3.1.6	Das V-Modell.....	22
3.2	Rollen.....	23
3.2.1	Projektmanager	23
3.2.2	Risikomanager	24
3.2.3	Anforderungsanalytiker.....	25
3.2.4	Spezifizierer	26
3.2.5	Designer	26
3.2.6	Systemtechniker	27
3.2.7	Entwickler / Programmierer	28
3.2.8	Tester / Qualitätssicherer.....	28
3.2.9	Qualitätsmanager.....	29
3.2.10	Wartungsexperte.....	30
3.2.11	Konfigurationsmanager	30
3.2.12	Risiko-Management-Board	31
3.2.13	Projekt-Management-Board	31
3.2.14	Change-Management-Board.....	32
4	Anwendungsgebiete für Softwareentwicklungsprozesse.....	32
5	Verteilte Systeme und Verteiltes Prozessmanagement	36
5.1	Zentrales Prozessmanagement.....	37
5.2	Dezentrales Prozessmanagement.....	38
6	Beschreibungssprachen für Prozesse.....	41
6.1	Ereignisgesteuerte Prozessketten (EPK).....	42
6.2	Unified Modeling Language (UML)	42
6.3	Petrinetze und Petrinetze	43
6.3.1	B/E-Netze.....	44
6.3.2	S/T-Netze	44
6.3.3	Pr/T-Netze.....	44
6.3.4	Funsoft-Netze.....	45
7	Szenarien für konkrete Entwicklungsprozesse	45
7.1.1	Entwicklungsprozess.....	47
7.1.2	Besonderheiten des Entwicklungsprozesses	53
8	Notwendige Erweiterungen zur Unterstützung	54
8.1	Räumliche und zeitliche Bedingungen	55
8.2	Grundzüge von Sprache und Konversation	56
8.3	Erweiterung von Rollen	57
8.4	Modellierung von verteilter Kommunikation	57
8.5	Modellierung von Zeiten und Kosten.....	59
8.6	Dynamische Veränderung von Geschäftsprozessen	59
8.6.1	Klassifikation von dynamischen Modellierungen.....	60
8.6.2	Klassifikation von dynamischen Veränderungsoperationen	61
8.7	Organisationsplanung und -struktur.....	62
8.7.1	Lean-Management.....	63
8.7.2	Unterstützung der LM-Konzepte durch CSCW-Systeme	65
9	Zusammenfassung und Bewertung	66
10	Literaturverzeichnis	70
11	Abbildungsverzeichnis.....	73

1 Einleitung

Die Softwaretechnik ist eine Forschungsdisziplin (in der Industrie und in den Hochschulen), die sich mit dem Einsatz, aber auch der Entwicklung von Techniken (Methoden, Verfahren/Prozesse, Prinzipien und Hilfsmitteln) für die Softwareentwicklung befasst. Die Techniken werden bei der Entwicklung von Softwaresystemen in einem Softwareentwicklungsprojekt eingesetzt. Ein Softwareprojekt umfasst den organisatorischen und zeitlichen Rahmen (d.h. das Zusammenwirken von Personal, Werkzeugen, Maschinen, etc.), welche der Entwicklung eines Softwaresystems auf der Basis eines Softwareentwicklungsprozess zugrunde liegt. *„[Ein Softwareprojekt ist ein] Vorhaben, das im wesentlichen durch Einmaligkeit der Bedingungen in ihrer Gesamtheit gekennzeichnet ist, wie z.B. Zielvorgabe; zeitliche, finanzielle, personelle und andere Bedingungen, projektspezifische Organisation ...“* /Sc91/ Der Softwareentwicklungsprozess legt die Art und die Reihenfolge der durchzuführenden Aktivitäten sowie die Hilfsmittel für die Durchführung der Aktivitäten fest. Hilfsmittel bei der Durchführung der Aktivitäten können z.B. Softwarewerkzeuge, aber auch Formulare, Interviews, Moderationstechniken, etc. sein. Softwarewerkzeuge sind Softwaresysteme, die zur Unterstützung der Entwicklung von anderen Softwaresystemen in Softwareprojekte eingesetzt werden. *„Eine Prozessarchitektur legt den allgemeinen Rahmen für die Spezifikation von Softwareentwicklungsprozessen fest. Sie besteht aus einer Standardmenge von fundamentalen Prozessschritten. Regeln bestimmen, wie Prozesse beschrieben und in Beziehung gesetzt werden. Ein Prozess beschreibt Aktivitäten, Methoden und Verfahren, die zur Entwicklung und \langle berprüfung von Software benötigt werden.“* /Ba98/. Das Ergebnis eines Softwareentwicklungsprozesses ist ein Softwaresystem, welches eine Softwaredokumentation (jede Art der Beschreibung, die im Laufe des Softwareentwicklungsprozesses für ein Softwaresystem erstellt wird) und das eigentliche Softwareprogramm (der Objektcode als ausführbare Bitfolge) umfasst.

Durch den Einsatz von Methoden, *„Methoden sind planmäßig, begründete Vorgehensweisen zur Durchführung von Tätigkeiten mit Hilfe von Beschreibungsmitteln zur Erreichung von [einem festgelegtem Ziel]“* /Ba82/) Verfahren/Prozessen, Prinzipien *„Prinzipien sind Grundsätze, die man seinem Handeln zugrunde legt“* /Ba82/) und Hilfsmitteln soll die Komplexität bei der Entwicklung von Softwaresystemen reduziert und der Entwicklungsprozess effizienter und produktiver gestaltet werden. Hierdurch soll die Entwicklungsdauer reduziert und die Entwicklungskosten gesenkt werden, bei gleichzeitiger Verbesserung und Steigerung der Qualität.

Die Softwareentwicklungsprozesse haben innerhalb der Softwareentwicklung eine große Bedeutung, denn durch sie können die drei Faktoren Zeit, Kosten, Qualität erheblich beeinflusst werden. Voraussetzung für eine positive Beeinflussung dieser drei Faktoren bei der Softwareentwicklung durch Softwareentwicklungsprozesse ist jedoch einerseits eine Beschreibung des Softwareentwicklungsprozesses und andererseits eine automatische Unterstützung des Softwareentwicklungsprozesses. Die Beschreibung von Softwareentwicklungsprozessen wird als Prozessmodellierung bezeichnet, während die automatische Unterstützung von Prozessen allgemein (hierzu zählen auch die Softwareentwicklungsprozesse) durch Softwaresysteme erfolgt. Zu diesen Softwaresystemen zählen CSCW-Systeme (*Computer Supported Cooperative Work*), zu denen auch Workflowmanagementsysteme und Groupwaresysteme zählen, aber auch alle anderen einfachen und komplexen Softwaresysteme, die dieses Ziel verfolgen. Neben der Erhaltung der Qualität des zu entwickelnden Softwaresystems in kürzerer Zeit und bei geringeren Kosten soll durch die Softwareprozessmodellierung folgende Ziele erreicht werden /CuKe92/, /RoVe95/:

- Kommunikations- und Diskussionsgrundlage ,ber Softwareentwicklungsprozesse,
- Wiederverwendbarkeit von Softwareprozessmodellen,
- Weiterentwicklung, Verbesserung und Anpassung von Softwareprozessmodellen,
- Unterst, tzung des Prozessmanagements,
- Unterst, tzung des Projektmanagements, Qualitätsmanagements, etc.

Eine formale Beschreibung eines Softwareentwicklungsprozess erleichtert zwar eine automatische Unterst, tzung, ist aber nicht zwangsl, ufig erforderlich, um einen positiven Effekt f, r die Softwareentwicklung zu erzielen. Ein gemeinsames Verst, andnis ,ber den Softwareentwicklungsprozess l, oest sich bei allen beteiligten Personen schon durch eine strukturierte und weitestgehend umfassende Beschreibung (Modellierung) erreichen. Das Wissen ,ber bew, orte Vorgehensweisen wurde und wird in oftmals schon in unternehmensinternen Dokumenten und Entwicklungsrichtlinien beschrieben. So schreibt z.B. die ISO 9000 (Teil 1-3) nur die Inhalt der Beschreibung von Vorgehensweisen und Entwicklungsrichtlinien vor, jedoch nicht die Notation /Th97/,/Th98/. Allerdings birgt eine Beschreibung in einer nat, rlichen Sprache die Gefahr in sich, dass es zu Fehl- und Missinterpretationen kommen kann. Dies liegt in der Tatsache begr, ndet, dass diese Art der Beschreibungen in der Regel einen enormen Umfang haben und das durch die Verwendung der nat, rlichen Sprache bestimmte Sachverhalte, Abh, angigkeiten und Voraussetzungen nicht immer pr, zise ausgedr, ckt werden oder werden k, nnen.

Dar, ber hinaus existiert kein Standardentwicklungsprozess in der Softwareentwicklung, denn die Rahmenbedingungen f, r jedes Softwareprojekt und f, r jedes Softwaresystem sind immer individuell /Fl94/,/Th97/. Dennoch wird versucht mindestens einen Rahmen f, r Beschreibung und Verbesserung von Softwareentwicklungsprozesse vorzugeben, wie z.B. ISO9000 /Th97/, V-Modell /Mu99/, /St99/, *Capability Maturity Model* (CMM) /Hu89/, *Spice* /Th98/,/EmDr98/, *Bootstrap*.

2 Computer Supported Cooperative Work (CSCW)

Bestand fr, her die Aufgabe einer Unternehmensorganisation in der m, glichst detaillierten und eindeutigen Definition der Kompetenzen und Verantwortlichkeiten der einzelnen Mitarbeiter, so spielte eine gesamtheitliche Betrachtung und Optimierung der Gesch, ftsprozesse im Rahmen der Organisationsgestaltung nur eine untergeordnete Rolle. Dieses Verst, andnis von der Aufgabe einer Unternehmensorganisation hat sich ge, andert, die Dominanz der Aufbau-, ber die Ablauforganisation wird zunehmend in Frage gestellt /De96/, /HaCh93/. In den Mittelpunkt der Betrachtung r, cken mehr und mehr die Gesch, ftsprozesse, in denen die Leistungen des Unternehmens erbracht und die Ertr, ge erwirtschaftet werden. Die Wertsch, pfung eines Unternehmens soll in ganzheitlicher Form als Menge von Gesch, ftsprozessen betrachtet werden. *Ein Gesch, ftsprozess (Business Process) ist eine logisch zusammengeh, rende Menge von Aktivit, ten, die einem bestimmten Gesch, ftszweck dienen* /Gr96/. In einem Telekommunikationsunternehmen ist die Bearbeitung eines Antrags f, r einen Handyvertrag ebenso ein Gesch, ftsprozess, wie in einem Softwareunternehmen die Erstellung eines einfachen oder komplexen Softwaresystems. Die Komplexit, eines Gesch, ftsprozesses wird durch die Menge der Aktivit, ten bestimmt. Je gr, fler die Menge der Aktivit, ten ist, desto mehr Personen sind auch an der Durchf, hrung der Aktivit, ten beteiligt. Dar, ber hinaus wird die Komplexit, eines Gesch, ftsprozesses auch durch die Art der Aktivit, ten bestimmt, einfache Aktivit, ten, wie z.B. die Erfassung von Daten eines Antragsstellers oder komplexe Aktivit, ten, wie z.B. Durchf, hrung eines Integrationstest bei einer Softwareentwicklung. Sind in der Menge der Aktivit, ten eines Gesch, ftsprozesses viele komplexe Aktivit, ten vorhanden, dann steigt auch die Anzahl der an der Durchf, hrung der Aktivit, ten involvierten Personen, was wiederum zus, zlich noch zur Komplexit, des Gesch, ftsprozesses beitr, gt. Ein Gesch, ftsprozess umfasst eine Menge von einfachen oder komplexen Aktivit, ten, die parallel, sequentiell oder alternativ von einer Person oder mehreren Personen durchgef, hrt werden, um dem Gesch, ftszweck gerecht zu werden, sei es eine Antragbearbeitung oder eine Softwareentwicklung. Der Gesch, ftszweck kann komplex sein, so dass mehrere Gesch, ftsprozesse ben, tigt werden, um dem Gesch, ftszweck gerecht zu werden. So umfasst z.B. die Softwareentwicklung die Gesch, ftsprozesse Projektmanagement (PM), Qualitätsmanagement (QM), Konfigurationsmanagement (KM), f, nderungsmanagement (CM), Systementwicklung (SE).

Die Bearbeitung eines Antrags für einen Handyvertrag kann z.B. die Geschäftsprozesse Schufa-Auskunft, Versand, und Rechnungserstellung umfassen.

Das bedeutet wiederum, dass einzelne Geschäftsprozesse einfach und komplex sein können und miteinander interagieren müssen, um dem Geschäftszweck gerecht zu werden. Ebenso können Geschäftsprozesse zeitlich und räumlich verteilt sein, z.B. der Versand eines Handys erfolgt an einem anderen Standort innerhalb eines Unternehmens als die Rechnungsstellung für einen Kunden.

Es werden daher drei Klassen von Geschäftsprozessen unterschieden. Der Routineprozess ist durch ein hohes Volumen gekennzeichnet und wird zur Bewältigung von Routineaufgaben verwendet, da er stark strukturiert ist. Durch die große Anzahl der involvierten Personen ist dieser Routineprozess hinreichend komplex. Es existieren keine Abweichungen von diesem Routineprozess. Abweichungen vom Routineprozess, die durch individuelle Besonderheiten gegeben sind, führen dann zu einem Regelprozess. Der Regelprozess beinhaltet zwar eine Menge von Routineaufgaben, beinhaltet aber im Gegensatz zum Routineprozess auch einfache und komplexe Aufgaben, die aufgrund von individuellen Besonderheiten immer wieder unterschiedlich parallel oder alternativ zueinander erledigt werden müssen. Diese individuellen Besonderheiten können auch dazu führen, dass Aufgaben nicht oder nur teilweise erledigt werden müssen. Der Regelprozess wird als schwach strukturiert bezeichnet. Als Ad-Hoc-Geschäftsprozesse werden einmalige Prozesse bezeichnet, die speziell für eine einzelne Aufgabe durchgeführt werden. Diese einmaligen Prozesse sind unstrukturiert.

Teilaufgaben im Prozess Prozessart	Einzelfallaufgabe	Sachbezogene Aufgabe	Routineaufgabe
Einmaliger Prozess			
Regelprozess			
Routineprozess			

Groupware

Workflow-Management

Abbildung 1: Klassifikation von Geschäftsprozessen /PiRo95/

2.1 Computergestützte Gruppenarbeit

Ob Ziele in Organisationen erreicht werden, hängt in vielen Fällen von kooperativer Aufgabenerfüllung ab. Die computergestützte Gruppenarbeit unterstützt diese Koordination durch Überbrückung räumlicher und zeitlicher Distanzen. CSCW ist zudem ein Hilfsmittel für das Management arbeitsteiliger verteilter Geschäftsprozesse. Das primäre Ziel ist dabei immer die Steigerung der Effektivität und Effizienz betrieblicher Abläufe und damit die Leistungssteigerung der gesamten Organisation. CSCW gewinnt in zunehmendem Maße an Bedeutung, da in vielen Fällen die Hardwareinfrastruktur für die Unterstützung der Gruppenarbeit bereits vorhanden ist. Wesentliches Ziel ist die Unterstützung der Arbeit einer Gruppe, die räumlich und/oder zeitlich verteilt ist. Es können grundsätzlich Gruppen, Arbeitsgruppen und Teams unterschieden werden.

Eine Gruppe ist daher ein soziales System. Sie umfasst eine Menge von Personen, zwischen denen eine relativ dauerhafte Struktur von Beziehungen besteht. Von einer Gruppe spricht man, wenn zwei oder mehrere Personen interagieren und dabei eine gegenseitige Beeinflussung stattfindet. Dies gilt auch für Arbeitsgruppen. Eine Arbeitsgruppe ist eine Gruppe mit einer gemeinsamen Aufgabe. Geht man noch etwas genauer auf den Begriff der Arbeitsgruppe ein, so kann man diese genauer dadurch bestimmen, dass sie aus einer Gruppe von Personen besteht, die in einem bestimmten funktional zusammenhängenden Abschnitt in all den Fragen, die nicht überge-

ordneten Interessen im Wege stehen, die anfallenden Entscheidungen selbst treffen kann. Hierzu zählen unter anderem die Einrichtung der Arbeitsplätze, die Reihenfolge der Bearbeitung, das Arbeitstempo, die Arbeitsverteilung und Leistungsbeurteilung, die Qualitätskontrolle, die Einarbeitung neuer Mitglieder, usw. Bei solchen Gruppen wird die hierarchische Struktur auf den unteren Ebenen relativiert und die unmittelbaren Vorgesetzten arbeiten primär als Berater die außerdem für die Fortbildung innerhalb der Gruppe und den Informationsfluss zuständig sind. Es kommt auf den unteren Ebenen zu einer teilweisen Reintegration von planender und ausführender Arbeit, während die mittleren und zielrelevanten Entscheidungen nach wie vor auf den oberen Hierarchieebenen verbleiben. Solche Arbeitsgruppen werden als ein Team bezeichnet, wobei deren Mitglieder den Willen haben, ein gemeinsames Ziel zu erreichen

Innerhalb von Teams findet Gruppenarbeit statt. Sie dient dem Zweck der Zielerreichung. Dazu müssen zielbezogene Aufgaben erfüllt werden. Dies geschieht, indem aufgabenbezogene Aktivitäten ausgeführt werden. Die Elemente der Gruppenarbeit sind also Gruppenziele, Gruppenaufgabe und Gruppenfähigkeit. Gruppenarbeit kann demnach wie folgt definiert werden: Gruppenarbeit ist die Summe aller aufgabenbezogenen Aktivitäten, die von Gruppenmitgliedern ausgeführt werden, um zielbezogene Aufgaben zu erfüllen und somit Gruppenziele zu erreichen.

Um einzelne Gruppenfähigkeiten aufgabenbezogen und damit zielorientiert ausführen zu können, sind bestimmte Gruppenprozesse erforderlich. Gruppenprozesse können in drei Klassen unterschieden werden:

- Kommunikation ist die Verständigung mehrerer Personen untereinander.
- Koordination bezeichnet jene Kommunikation, die zur Abstimmung aufgabenbezogener Aktivitäten, die im Rahmen der Gruppenarbeit ausgeführt werden, erforderlich ist.
- Kooperation bezeichnet die Kommunikation, die zur Koordination und Vereinbarung gemeinsamer Ziele erforderlich ist.

Diese oben gemachten Definitionen sind ein abstraktes Hilfsmittel, sie umfassen jedoch nicht den komplexen Schwierigkeitsgrad der Gruppenarbeit im vollständigen Detail.

Beim CSCW steht im Mittelpunkt der Betrachtung, wie die Arbeit in einem Unternehmen verrichtet wird. Dabei wird die Arbeit als etwas angesehen, das durch die verschiedenen Arbeitsplätze und Instanzen einer Unternehmensorganisation fließt, wobei diese jeweils einen vorher klar definierten Teil der Gesamtarbeit verrichten. Ziel aller Bemühungen ist es, unter Verwendung aller zur Verfügung stehenden Mittel der Informations- und Kommunikationstechniken, Gruppenprozesse zu unterstützen und dabei die Effektivität und Effizienz der Gruppenarbeit zu erhöhen. CSCW sollen insgesamt dafür sorgen, dass

- die Mitarbeiter von administrativen Routineaufgaben entlastet werden: Nicht wertschöpfende Arbeiten, die zu vergrößerten Transport- und Reisezeiten führen (Dateien kopieren, Dokumente suchen, usw.), können weitgehend automatisiert werden und so die Ausführung insbesondere routinemäßiger Aktivitäten (auch über Anwendungsgrenzen, Funktionsbereiche und organisatorische Einheiten hinweg) beschleunigen.
- eine mangelhafte Transparenz der Geschäftsprozesse und der Verantwortlichkeiten beseitigt wird: Zuvor steckte das Wissen über den Ablauf der Geschäftsprozesse nur unstrukturiert und uneinheitlich in den Köpfen der Mitarbeiter und war in der Regel nicht dokumentiert. Ein Prozessmodell beschreibt einen Geschäftsprozess in nachvollziehbarer Weise und macht ihn unter organisatorischen Gesichtspunkten auswertbar. Das Tätigkeitsprofil für jeden Mitarbeiter ergibt sich aus seiner Einordnung in die Aufbauorganisation des Unternehmens.
- der tatsächliche Ablauf der Geschäftsprozesse protokolliert, archiviert und überwacht wird: Der zuvor oft herrschende Mangel an gesicherten Informationen über den tatsächlichen Prozessablauf (Häufigkeit, Durchlaufzeit, Beteiligte, Rückfragen, usw.) wird beseitigt und führt daher zu einer Verkürzung der Liegezeit von Dokumenten.

- eine effiziente Kommunikation zwischen den am Geschäftsprozess involvierten Personen möglich wird: Ein integrierter Eingangskorb wird zur wichtigsten Schnittstelle eines Mitarbeiters zur betriebswirtschaftlichen Anwendungsfunktionalität. Alle zur Bearbeitung anstehenden Aktivitäten können übersichtlich dargestellt und z.B. per Mausklick ausgeführt werden.
- durch eine Integration aller eingesetzten Anwendungen keine Daten mehrfach erfasst werden müssen: Das Prozessmodell beschreibt den aktiven Kontroll- und Datenfluss zwischen den Bearbeitungsschritten eines Prozesses. Jeder Schritt wird automatisch mit den benötigten Daten und Dokumenten versorgt und erzeugt seinerseits Ergebnisse, welche die Auswahl der nachfolgenden Schritte steuern und inhaltlich in späteren Schritten zur Verfügung stehen. Dies kann zur einer Verkürzung der Bearbeitungszeit des Teilschrittes führen.

Neben der automatischen Unterstützung von Geschäftsprozessen ist ein weiteres Ziel von CSCW die ständige Verbesserung der unterstützten Geschäftsprozesse. Die Betrachtung und Optimierung dieser Geschäftsprozesse, um in kürzerer Zeit und bei geringeren Kosten den Geschäftszweck zu erreichen, ist das Ziel von Reorganisationsprojekten, die mit dem Begriff *Business Process Reengineering (BPR)* bezeichnet werden können. *"BPR steht für eine grundlegende und kompromisslose Erneuerung der Unternehmensstrukturen mit dem Ziel, kundenorientierte Geschäftsprozesse zu gestalten. Dabei findet eine Fokussierung auf wesentliche Geschäftsprozesse ohne vordefinierte organisatorische Grenzen statt"* /De96/. Eine Optimierung von Geschäftsprozessen kommt in der Regel nur bei Geschäftsprozessen in Betracht, die ein hohes Potential aufweisen. Hierzu muss in dem Reorganisationsprojekt untersucht werden, welche Geschäftsprozesse häufig durchgeführt werden, d.h. ein hohes Volumen haben. Geschäftsprozesse, die zur Erreichung eines Geschäftszwecks nur eine untergeordnete Rolle spielen, sind nicht Gegenstand einer Optimierungsbetrachtung. Dies können Geschäftsprozesse sein, die nur in Ausnahmesituationen durchgeführt werden. Ein hohes Potential ist neben dem Volumen ebenfalls durch die Komplexität des Geschäftsprozesses gegeben.

Dadurch wird der Zusammenhang zwischen *Business Process Reengineering* und CSCW deutlich. Beide erfordern, dass im Vorfeld ein vollständiges Modell der Geschäftsprozesse erstellt wird. Die in einem Reorganisationsprojekt überarbeiteten und optimierten Geschäftsprozesse können in ein CSCW-System abgebildet und anschließend rechnergestützt durchgeführt werden.

2.1.1 Klassifikation von CSCW-Systemen

Die automatische Unterstützung von Geschäftsprozessen im allgemeinen ist das Ziel von CSCW-Systemen. Durch den Einsatz von Softwaresystemen sollen die involvierten Personen eines Geschäftsprozesses bei ihrer Arbeit unterstützt werden. Die Rationalisierungspotentiale liegen dann insbesondere in der Verkürzung von Zeitaufwänden (Transportzeit, Liegezeit, Rüstzeit, Bearbeitungszeit).

CSCW-Systeme sind für den unternehmensweiten Einsatz vorgesehen und müssen geeignete Mechanismen zur Übertragung von Informationen zwischen den einzelnen Benutzern bereitstellen (Kommunikation), den Benutzern einen gemeinsamen Zugriff auf einen Pool von Informationen ermöglichen und die einzelnen Beteiligten über erledigte bzw. noch zu erledigende Aufgaben aktiv benachrichtigen (Kooperation). Aber erst eine Abstimmung der von den beteiligten Personen auszuführenden Aktivitäten (Koordination) schafft die Voraussetzung für eine effiziente Kommunikation und effektive Kooperation zwischen den beteiligten Personen. In diesem Sinne ist ein CSCW-System dafür verantwortlich, dass eine mehrfache oder gar eine sich widersprechende Ausführung von Aktivitäten vermieden wird.

Durch den Einsatz von CSCW-Systeme hat ein Rollenwechsel bezüglich der Verwendung eines Rechnersystems stattgefunden: vom passiven Problemlöser zum aktiven Medium, das die Interaktion zwischen den menschlichen Benutzern unterstützen und auch kontrollieren soll. Die drei zentralen Aspekte *Kommunikation*, *Kooperation* und *Koordination* können unter diesem Oberbegriff Interaktion zusammengefasst werden /M, Ko97/, Ja95/.

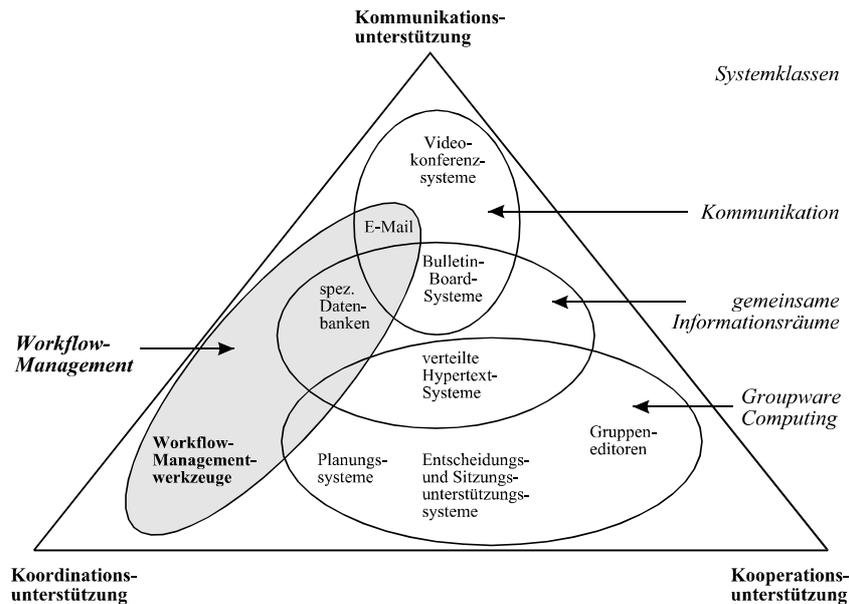


Abbildung 2: Klassifikationsschema für CSCW-Systeme /M, Ko97/

Neben der Klassifikation nach dem Grad der Interaktion können CSCW-Systeme nach den folgenden Kriterien unterschieden werden,

- bzgl. der verwendeten Medientypen: Medientypen sind Seh- und Hör-sinn ansprechende Typen wie Text, Bild und Audio- bzw. Videosequenzen.
- bzgl. der räumlichen Verteilung: Bei der räumlichen Verteilung unterscheidet man in räumlich benachbarte (lokale) und räumlich entfernte (verteilte) Verteilung.
- bzgl. der zeitlichen Kommunikation: Bei der zeitlichen Kommunikation unterscheidet man zwischen zeitgleicher (synchroner) und zeitverschiedener (asynchroner) Kommunikation.
- bzgl. der Anzahl der beteiligten Personen: Hier werden Kommunikationssituationen mit einem, zwei oder mehreren Teilnehmern unterschieden, die unterschiedlich unterstützt werden.
- bzgl. der Art der Kommunikation: Hier wird zwischen expliziter und impliziter (direkt oder indirekter) Kommunikation unterschieden. Eine explizite Kommunikation liegt dann vor, wenn die Teilnehmer aktiv Informationen austauschen, z.B. in einem persönlichen Gespräch Angesicht zu Angesicht oder über elektronische Medien (E-Mail, Talk, Chat). Dagegen ist eine Kommunikation indirekt, wenn sie über das Medium gemeinsamer Informationsobjekte stattfindet, z.B. über Dokumente, die mit Hilfe einer gemeinsam genutzten Datenbank verwaltet werden.

Durch CSCW-Systeme werden meist unterschiedliche Kriterien erfüllt, daher ist es schwierig CSCW-Systeme immer einer Kategorie zuzuordnen.

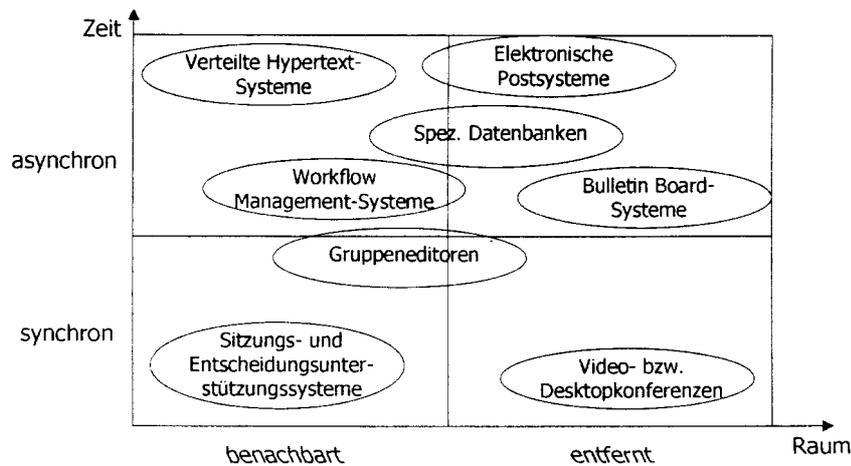


Abbildung 3: Raum-Zeit-Matrix

Auf der Basis der Unterstützungsfunktionen lassen sich CSCW-Systeme entsprechend klassifizieren:

- **Bulletin Board Systeme**
Bulletin Board Systeme sind spezielle Datenbanken, bei denen Informationen, Nachrichten und Hinweise verschiedener Autoren nach Themenschwerpunkten gespeichert und einer Menge von Teilnehmern zur Verfügung gestellt werden. Diese Themenschwerpunkte werden auch als Newsgroups bezeichnet. Die Beiträge zu diesen Themenschwerpunkten besitzen ein semi-strukturiertes Format. Bulletin Board Systeme ermöglichen eine schnelle Verfügbarkeit von Informationen für räumlich und zeitliche verteilte Gruppen und werden auch für asynchrone verteilte Diskussionen verwendet. In Bulletin Board Systemen findet eine indirekte Kommunikation statt, d.h. der Verfasser einer Information stellt diese einer grundsätzlich anonymen Menge anderer Teilnehmer zur Verfügung.
- **Nachrichtensysteme**
Elektronische Nachrichtensysteme unterstützen den asynchronen Austausch von Informationen. Dies hat gegenüber der synchronen Kommunikation den Vorteil, dass keine gleichzeitige Anwesenheit der Kommunikationspartner erforderlich ist. Das Nachrichtensystem besteht mindestens aus einer elektronischen Post (E-Mail). Weiterhin können elektronische Pinbretter installiert sein oder sogar ein Filtersystem, welches Nachrichten nach frei wählbaren Kriterien entweder weiterverteilt, filtert oder speichert. Ein wichtiger Vorteil von elektronischen Postsystemen ist die einfache Handhabung von Verteilerlisten, die es erlauben, eine Nachricht sehr einfach an eine größere Menge von Empfängern zu adressieren. Dieses Konzept wird unter anderem für elektronische Konferenzsysteme genutzt, bei denen es für die Teilnehmer darum geht, Beiträge zu bestimmten Themen sowohl zu verfassen, als auch von anderen Teilnehmern verfasste Beiträge zu lesen (und eventuell zu kommentieren). Hierbei ist es im Gegensatz zu Videokonferenzen nicht notwendig, dass alle Teilnehmer der Konferenz gleichzeitig anwesend sind, da die neu verfassten Beiträge mittels elektronischer Post (über Verteilerlisten) an die anderen Teilnehmer der Gruppe entweder direkt oder nach Begutachtung (und eventuell der Bearbeitung) durch den sogenannten Konferenzmoderator weitergeleitet werden. Jeder Teilnehmer erhält dann bei jeder Sitzung, eine Übersicht über alle neu eingegangenen Beiträge und kann sich diese dann ansehen. Ziel von elektronischen Postsystemen ist es, die Kommunikation bzw. die Zusammenarbeit effektiver zu gestalten, als dies bislang der Fall war. Auch Systeme, die direkte, zeitgleiche Kommunikation unterstützen, haben sich dieses Ziel gesetzt. Hierbei wird allerdings zusätzlich ein anderer Aspekt in die Betrachtung miteinbezogen, nämlich die der informellen Konversation (siehe Kap. 8.2)

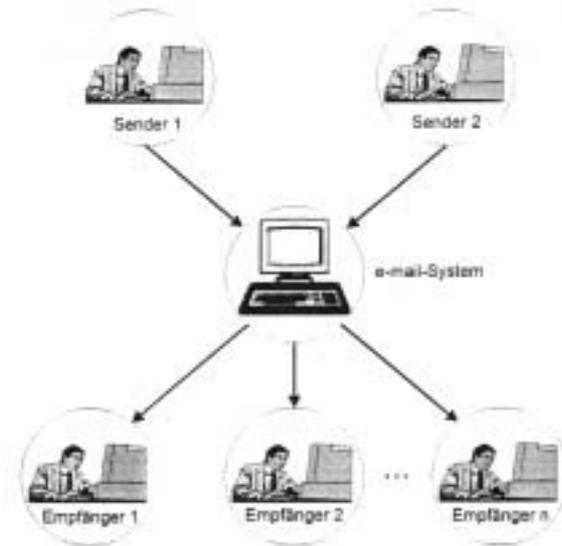


Abbildung 4: Nachrichtensysteme

- Gruppeditoren

Mit Gruppeditoren ist es möglich, dass eine Menge von Personen gleichzeitig ein Dokument bearbeiten. Es lassen sich hierbei verschiedene Arten unterscheiden. Annotationssysteme erlauben es, dass Teilnehmer Kommentare und Korrekturen zu einem Dokument hinzufügen. Entweder wird das Originaldokument direkt verändert oder nur Änderungen als Kommentare vermerkt. Kautorensysteme unterstützen die Erstellung eines Dokuments durch mehrere Autoren. Dabei können mehrere Autoren gleichzeitig an verschiedenen Stellen Veränderungen vornehmen. Dabei kann entweder das Dokument segmentiert werden, so dass jede Person nur einen Teil des Dokuments bearbeitet und verändern kann, oder es findet keine Segmentierung statt. Je nach dem welches Konzept verfolgt wird (optimistisch oder pessimistisch), werden Änderungen an Textteilen allen anderen Personen sofort mitgeteilt oder erst bei der Speicherung des Dokuments. Eine Besonderheit hierbei ist, dass ein Textteil auch dann nicht für die anderen Teilnehmer gesperrt werden muss, wenn er sich gerade in Bearbeitung durch eine Person befindet. Eine Bearbeitung eines Textteils wird nur durch optische Anzeige den anderen Beteiligten gegenüber deutlich gemacht¹. Gemeinsam verwendbare Zeichen- und Malwerkzeuge erlauben das zeitgleiche Erstellen von Bildern und Zeichnungen.

- Konferenzsysteme

Konferenzsysteme unterstützen die synchrone Gruppenarbeit von räumlich verteilten Teilnehmern. Sie unterstützen eine bestimmte und bekannte Menge von Teilnehmern, die alle zur gleichen Zeit miteinander kommunizieren wollen. Konferenzsysteme können mit verschiedenen Medien realisiert werden, Es gibt rein textuelle, audiosequenzbasierte oder videosequenzbasierte Systeme bzw. hybride Formen.

- Textbasierte Konferenzsysteme

Die textbasierten Konferenzsysteme verfügen nicht über eine graphische Benutzeroberfläche. Bei diesen Systemen wird der Bildschirm horizontal in mehrere Bereiche eingeteilt, die von verschiedenen Teilnehmern zur Eingabe von Textmitteilungen und bei anderen Teilnehmern zur Darstellung von Textmitteilungen verwendet werden. Diese Systeme gehören zum Stan-

¹ Die Koordination muss dann in mündlicher Absprache erfolgen. In Erweiterung kann auch (im Gegensatz zu normalen sitzungsunterstützten Systemen) das räumlich verteilte Zusammenarbeiten an einem Text unterstützt werden. Hierzu ist dann aber zusätzlich wenigstens eine Audio-Verbindung der Teilnehmer notwendig, um die erforderlichen Absprachen bezüglich der zeitgleichen Änderung abzusprechen.

Umfang von verschiedenen Betriebssystemen, aber auch zum Umfang von PC-basierten Remotesteuerungssystemen.

- Videokonferenzsysteme

Um Videokonferenzen durchzuführen, werden permanente Audio- und Videoverbindungen zur Verfügung gestellt, die eine direkte Kommunikation von räumlich getrennten Personen ermöglichen. Gerade die informelle Kommunikation vor und nach den Sitzungen oder auf dem Flur (*„Coffeepot conversation“*) kann wesentlich zur erfolgreichen Kooperation beitragen. In Form einer Labor-Konferenz werden Videokonferenzen durchgeführt. Zwischen zwei räumlich getrennten Laboren wird eine permanente Audio- und Videoverbindung eingerichtet. Jedes Teil-Labor ist mit einer Video Wall ausgestattet, auf die ein Bild des jeweils anderen Laborteils projiziert wurde. Auf diese Weise existierte virtuell nur ein einziges Labor (bzw. nur ein einziger Raum), in dem sich die Leute wie in einem einzigen physikalischen Raum treffen und miteinander reden konnten. Durch dieses System war auch der Unterstützung der informellen Kommunikation genüge getan, da man in diesem virtuellen Labor auch quasi zufällig andere Leute treffen konnte.

- Desktopkonferenzsysteme

Desktopkonferenzsysteme sind eine Weiterentwicklung der Videokonferenzsysteme. Die Kommunikationspartner müssen nicht mehr in einem speziellen Videokonferenzraum wechseln, sondern jeder Teilnehmer verfügt auf an seinem Arbeitsplatz über die Möglichkeit Video- und Audiodaten zu erhalten und zu vermitteln. Es können neben mehreren Konferenzverbindungen gleichzeitig andere Applikationen benutzt werden. Eine Besonderheit bei den Desktopkonferenzen sind virtuelle Flure. Hier werden virtuelle Flure zusammengestellt, so dass beispielsweise ein Blick in ein anderes Büro entworfen werden kann das sich in Wirklichkeit in einiger räumlicher Distanz befindet. Jeder Teilnehmer kann seine virtuelle Bürotür halb oder ganz geöffnet oder auch geschlossen halten, um zu signalisieren, ob er derzeit ansprechbar ist, oder nicht. Ein Vorbeischauen in einem anderen Büro wird hierbei durch einfaches Anklicken auf dem Bildschirm realisiert. Signalisiert der gewünschte Teilnehmer seine Gesprächsbereitschaft, so wird automatisch eine Video-Verbindung zwischen den beiden Büros hergestellt.

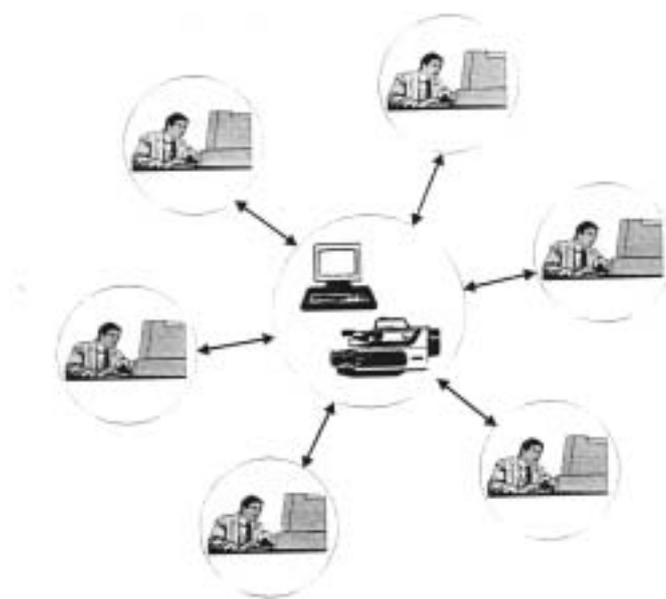


Abbildung 5: Konferenzsysteme

- Sitzungsunterstützungssysteme

Sitzungsunterstützende Systeme, werden oftmals auch als entscheidungsunterstützende Systeme bezeichnet. Diese Systeme unterstützen per definitionem das zeitgleiche Zusammenarbeiten mehrerer Personen im selben Raum. Wird eine solche Sitzung durch ein elektronisches System unterstützt, so steht hierbei die Produktion eines gemeinsamen Ergebnisses im Vordergrund, wobei es sich hierbei sowohl um den Entwurf eines veröffentlichten Artikels oder eines zu implementierenden Systems handeln könnte. Die Produktion eines solchen Ergebnisses setzt in jedem Fall Kommunikation, Verhandlungen, Beurteilungen, Abstimmungen und Entscheidungen der anwesenden Personen voraus. Um dieses entsprechend unterstützen zu können, werden vom System alle eingegebenen Beiträge aufgenommen (und anonym gespeichert) und stehen dann allen Teilnehmern zur Verfügung (gleiche Sicht auf das Material). Durch verschiedene Arbeitsphasen, die vom System vorgegeben werden, kann der Arbeitsprozess strukturiert werden. Meist sind Phasen vorgesehen, die zunächst die Sammlung von Ideen, dann das Ordnen dieser Ideen und schließlich die Bewertung mit angeschlossener Entscheidung beinhalten. Die Sitzungsunterstützung unterteilt sich in drei Phasen: die Sammelphase (*brainstorming*), die Ordnungsphase (*ordering*) und die Evaluationsphase (*evaluating*). Innerhalb der Sammelphase werden die Ideen der Teilnehmer zunächst durch Eintippen an den Terminals gesammelt und unstrukturiert auf den Bildschirm gebracht. Zu jedem Eintrag, der auf dem Bildschirm angezeigt wird (Stichwort) kann eine Erläuterung mit angegeben werden. Die allgemein anerkannten Regeln einer Brainstorming-Phase gelten selbstverständlich auch für ein computergestütztes Brainstorming, so dass die Einträge in dieser Phase von anderen Teilnehmern weder in Frage gestellt noch gelächelt werden dürfen. In der Ordnungsphase werden inhaltlich zusammengehörige Einträge durch Striche miteinander verbunden oder zusammengefasst und hinter einem neuen (Meta-)Eintrag einsortiert. Die Evaluationsphase schließlich dient dazu fehlende Punkte zu ergänzen, Unwichtiges auszusortieren und eine Gliederung des Vortrags zu erstellen. Alle Einträge und Angaben erfolgen in der Regel anonym, also ohne dass die anderen Sitzungsteilnehmer erfahren, von wem ein spezieller Eintrag vorgenommen wurde, es sei denn, die entsprechende Person kommentiert seinen Eintrag im Gespräch.



Abbildung 6: Sitzungsunterstützungssysteme

- **Planungssysteme**

Die Koordination von Ressourcen und Kapazitäten ist neben der Unterstützung der eigentlichen Gruppenarbeit das Ziel von Planungssystemen. Planungssysteme haben nicht den Schwerpunkt auf Unterstützung der Kooperation, Kommunikation oder Koordination der Zusammenarbeit von Gruppenmitgliedern, sondern dienen der Ermittlung von Daten und Informationen, um eine bessere Unterstützung zu erreichen und im Laufe der Arbeit kontinuierlich zu verbessern.

2.1.2 Groupwaresysteme und Workflowmanagementsysteme

Groupwaresysteme und Workflowmanagementsysteme dienen der Optimierung von Geschäftsprozessen durch Erhöhung der Produktivität und Flexibilität von Abläufen. Groupwaresysteme unterscheiden sich von Workflowmanagementsystemen vor allem dadurch, dass es keine häufig vorkommenden Standardprozesse gibt, die Organisation nicht streng hierarchisch, sondern eher teamorientiert aufgebaut ist, und es somit bewusst keine statische Verteilung der Aufgaben und der Verantwortlichkeiten gibt, der Benutzer die Initiative ergreift und das Groupwaresystem als Kommunikationsmedium zur rechnerunterstützten Zusammenarbeit mit anderen Mitarbeitern benutzt. Workflowmanagementsysteme sind vor allem für solche Anwendungsgebiete geeignet, in denen

- standardisierte und strukturierte Geschäftsprozesse in großer Zahl ablaufen,
- die Organisationsstruktur statisch und oft auch hierarchisch aufgebaut ist,
- es eine klare Kompetenz- und Aufgabenverteilung gibt,
- der Ablauf der Prozesse aktiv gesteuert und überwacht werden soll, die Initiative also auch vom System ausgehen kann.

Bei Workflowmanagementsystemen und Groupwaresystemen ist die Unterstützung der Koordination der Aufgaben unter den einzelnen am Geschäftsprozess beteiligten Personen die Hauptzielsetzung. Während dazu auch ein gewisses Maß an Kommunikationsunterstützung erforderlich ist (z.B. für die Benachrichtigung über zu erledigende Aufgaben), spielt die Kooperation keine wesentliche Rolle; denn durch das Prozessmodell ist die Art und Weise der Kooperation schon klar vorgegeben und somit definiert, wer welche Aufgaben zu erledigen hat. Eine weitergehende Kommunikationsunterstützung der Benutzer, beispielsweise durch Video- oder Audioverbindungen, wird in der Regel vom Workflowmanagementsystem nicht angeboten. Der Unterschied zwischen beiden Systemen liegt bei dem Strukturierungsgrad der Aufgabenstellung. Groupwaresysteme werden bei sehr unstrukturierten Aufgabenstellungen benutzt, deren Wiederholungsfrequenz außerdem noch sehr gering ist. Praktische Beispiele sind Gruppenterminkalender oder E-Mail-Systeme. Workflowmanagementsysteme unterstützen dagegen gut modellierbare Prozesse, die zudem häufig ablaufen sollen. Workflowmanagementsysteme und Groupwaresysteme unterscheiden sich auch anhand der Arbeitsverteilung. *Push-Systeme* übernehmen demnach eine aktive Rolle bei der Aufgabebearbeitung, da sie den beteiligten Personen die Aufgabenstellungen inklusiv aller dafür benötigten Unterlagen zur Verfügung stellen. Dies ist bei Workflowmanagementsystemen der Fall. *Pull-Systeme* beschränken sich dagegen auf die Verwaltung der aufgabenbezogenen Informationen und nehmen eine passive Stellung bezüglich der Aufgabenverteilung ein. Die Steuerung des Aufgabenflusses obliegt den Benutzern. Dies trifft für Groupwaresysteme zu.

Zwischen Groupwaresystemen und Workflowmanagementsystemen gibt es Gemeinsamkeiten und somit insbesondere die Notwendigkeit für Workflowmanagementsysteme Schnittstellen zu Groupwaresystemen zu unterstützen und anzubieten. Es folgt daraus, dass für die unternehmensweite Kommunikation sowohl Groupwaresysteme (Kooperationsunterstützung) als auch Workflowmanagementsysteme (Koordinationsunterstützung) benötigt werden. Somit wird auch deutlich, wie wichtig die Integrationsfähigkeit von Workflowmanagementsystemen und Groupwaresystemen ist, damit in Unternehmen beide Systeme zu einer benutzerfreundlichen Gesamtlösung kombiniert werden können. Die Funktionalität eines Workflowmanagementsystems sollte in die Benutzeroberfläche eines Groupwaresystems integrierbar sein, damit z.B. der Benutzer nicht mit mehreren Posteingangskästen konfrontiert wird, sondern beide Systeme für ihn transparent zum Einsatz kommen.

Damit ein Groupwaresystem und/oder ein Workflowmanagementsystem in einem Unternehmen eingesetzt werden kann, muss die Umgebung modelliert werden. In einem Unternehmen sind Akteure, Organisationseinheiten, Gruppen, Stellen und Rollen integrale Bestandteile der Umgebung, in der das System eingesetzt werden soll.

Ein Akteur ist eine Person, die an einem oder mehreren Geschäftsprozessen seines Unternehmens beteiligt ist. Er erledigt Aufgaben, die ihm von dem Workflowmanagementsystem zugeteilt werden. Im weiteren Sinne können

auch Computerprogramme als Akteure bezeichnet werden, wenn sie selbständig Aufgaben innerhalb eines Geschäftsprozesses erledigen. Um die κ bersichtlichkeit eines Unternehmens zu erh \hat{u} hen, wird dieses in Einheiten unterteilt. Die Einteilung wird dabei oft durch funktionale Zust α ndigkeiten bestimmt. Organisationseinheiten, auch als Abteilungen bezeichnet, gruppieren Stellen und k \hat{u} nnen hierarchisch geschachtelt sein. Eine Gruppe umfasst im Sinne der o.g. Definition Personen in zumeist zeitlich begrenzten Arbeitsgemeinschaften zusammen. Dies erfolgt z.B. im Rahmen von Projekten, die eine flexible Zusammenstellung von Arbeitskr α ften erfordern. Stellen beschreiben Arbeitspl α tze innerhalb einer Organisation und besitzen bestimmte Kompetenzen und Zust α ndigkeiten. Sie k \hat{u} nnen in Einheiten oder Gruppen zusammengefasst werden. Eine Rolle beschreibt das dynamische Verhalten von Akteuren. Diese k \hat{u} nnen eine Funktion innerhalb der Organisation (z.B. eine Stelle) nur dann ausf \ddot{u} hren, wenn sie die entsprechende Rolle besitzen. An eine Rolle sind Kompetenzen gebunden.

Neben der Modellierung der statischen Bestandteile der Umgebung ist eine notwendige Voraussetzung die Modellierung der Geschäftsprozesse. Dieses Modell ist notwendige Voraussetzung f \ddot{u} r die automatische Unterst \ddot{u} tzung von Geschäftsprozesse durch Softwaresysteme. Ein Geschäftsprozess wird in einem Prozessmodell beschrieben, das mit rechnergest \ddot{u} tzten Modellierungswerkzeugen bearbeitet und in der Laufzeitumgebung eines Groupwaresystems oder Workflowmanagementsystems ausgef \ddot{u} hrt werden kann. Das Prozessmodell besteht aus einem Netzwerk von Aktivit α ten und Beziehungen zwischen den einzelnen Aktivit α ten, sowie Personen (Akteuren), Rollen und Organisationseinheiten, die diese Aktivit α ten durchf \ddot{u} hren sollen.

2.1.2.1 Architektur eines Workflowmanagementsystems

Ein Workflowmanagementsystem setzt sich aus verschiedenen funktionalen Komponenten zusammen. Bei der Modellierung wird ein bestehender Geschäftsprozess in das System abgebildet. Neben den eigentlichen Aktivit α ten geh \hat{e} ren dazu auch essentielle Dinge wie der Organisationsstruktur, den Personalressourcen und der Rollenvergabe. Die Modellierungskomponente ist meist ein separates Programm, welches die Abbildung des Geschäftsprozesses entweder mit Hilfe eines Graphenmodells oder anhand von textuellen Beschreibungen erlaubt. An Workflowmanagementsysteme besteht auch die Forderung, bestehende Applikationen in die Vorg α nge einbinden zu k \hat{u} nnen. Dabei m \ddot{u} ssen vom System die notwendigen Daten an die Applikation \ddot{u} bergeben werden und nach Abschluss des Arbeitsschrittes wieder in die Datenbasis \ddot{u} bernommen werden.

Die Steuerung bildet den eigentlich funktionalen Kern eines jeden Workflowmanagementsystems. Sie ist f \ddot{u} r die Verteilung der Aufgaben an die vorgesehen Benutzer zust α ndig. Weiterhin muss sie anhand des Prozessmodells entscheiden, in welcher Reihenfolge Aktivit α ten gestartet werden sollen. Daf \ddot{u} r ben \hat{e} tigt sie Zugriff auf die Datenbasis, da Verzweigungen im Arbeitsfluss h α ufig von bestimmten Informationen (Steuerungsdaten) abh α ngig sind.

Mit der Bearbeitungskomponente wird es den Benutzern erm \hat{u} glicht, die vom Workflowmanagementsystem zugeteilten Aufgaben zu erledigen. Genau wie die Modellierungskomponente ist sie oft ein separates Programm, welches in seiner einfachsten Ausf \ddot{u} hrung neben der Arbeitsbeschreibung die zur Bearbeitung notwendigen Daten bereitstellt. Anspruchsvollere Workflowmanagementsysteme starten automatische externe Applikationen, wenn dies erforderlich ist. Ein grof \ddot{u} rer Vorteil des Einsatzes eines Workflowmanagementsystems ist die Tatsache, dass alle f \ddot{u} r den Geschäftsprozess ben \hat{e} tigten elektronischen Daten und Dokumente zentral von diesem verwaltet werden. Daf \ddot{u} r wird in der Regel eine Datenbank verwendet, welche die komfortable und effiziente Speicherung und Bereitstellung der verschiedenen Informationen erm \hat{u} glicht. Neben den vom Workflowmanagementsystem ben \hat{e} tigten Arbeits- und Steuerungsdaten wird auch das eigentliche Prozessmodell in der Datenbank abgelegt.

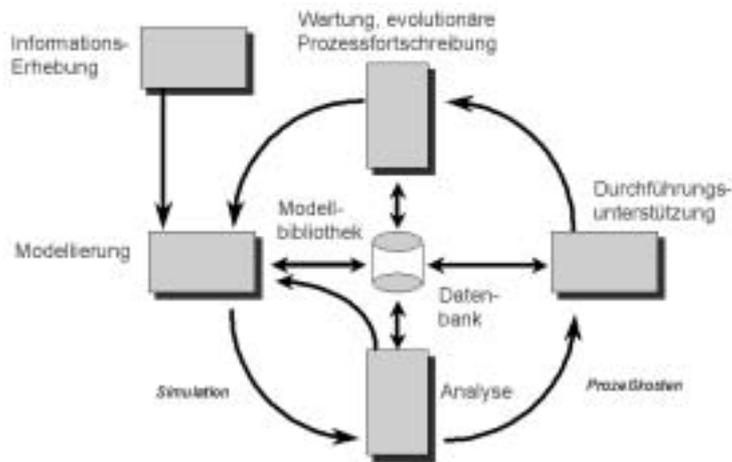


Abbildung 7: Architektur eines Workflowmanagementsystems

Während die bisher vorgestellten Komponenten primär die aktuellen Arbeitsflüsse wichtig sind, dient das Überwachungsmodul für allgemeine Rechercheaufgaben. Zum Beispiel werden Analysen über Liegezeiten und Verzögerungen möglich, die der späteren Optimierung des Vorgangs dienen. Ebenso können Arbeitsnachweise erbracht oder der Zustand eines gerade laufenden Geschäftsprozesses inspiziert werden.

2.1.2.2 Workflow-Management-Coalition (WfMC)

Die Workflow Management Coalition /WfMC96/, hat versucht, die relevanten Begriffe zu definieren und in einen Zusammenhang zu setzen, definiert ein Workflowmanagementsystem wie folgt: *“A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.”* /WfMC96/

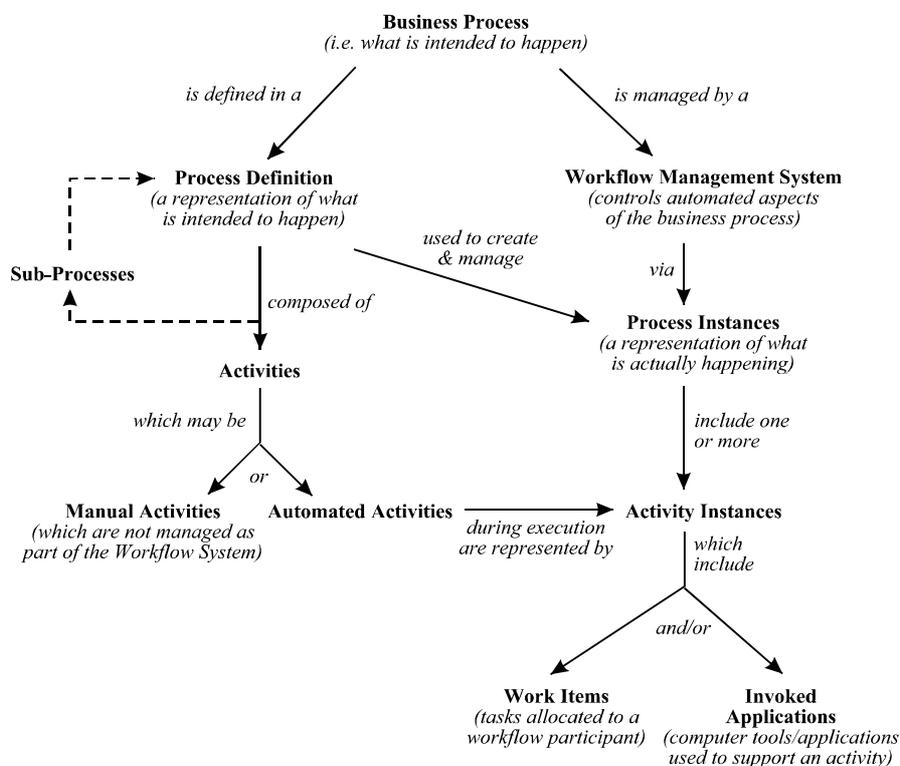


Abbildung 8: Workflow-Terminologie /WfMC96/

Ein Prozessschritt, auch Aktivität (Activity) genannt, besteht aus logisch zusammengehörenden Aufgaben, die von einer Person ausgeführt werden. Für jede Aktivität wird festgelegt, welche Dokumente oder Informationen

und welche menschlichen, technischen und organisatorischen Ressourcen zur Ausführung benötigt werden. Zur Bearbeitung der Aufgaben in einer Aktivität können interaktive Anwendungsprogramme verwendet werden. Alternativ sind auch manuelle Aktivitäten ohne Computerunterstützung möglich (*Manual Activities*) oder automatische Aktivitäten (*Automated Activities*), die ohne menschliche Interaktion auskommen. Eine Aktivität stellt die kleinste Einheit in einem Prozessmodell dar. Die Basis für eine flexible Ausführung von Aktivitäten ist, im Rahmen der Modellierung der Aufbauorganisation, die Realisierung eines Berechtigungssystems unter Verwendung eines Rollenkonzepts. Im Prozessmodell werden dann den unterschiedlichen Aktivitäten keine konkreten Personen, sondern ausschließlich Rollen zugeordnet. Die einzelnen Personen können dann einer oder mehreren Rollen innerhalb einer Unternehmensorganisation zugeordnet sein. Das Prozessmodell eines Workflowmanagementsystems dient zur Ausführung und Steuerung von Prozessen, indem es eine neue Prozessinstanz (*Process Instance*) erzeugt, die dann den aktuellen Zustand des laufenden Prozesses repräsentiert. Diese Prozessinstanz beinhaltet eine oder mehrere Instanzen der Aktivitäten (*Activity Instances*) die zum aktuellen Zeitpunkt bearbeitet und durchgeführt werden.

Durch die WfMC wurde ebenfalls ein Referenzmodell für ein Workflowmanagementsystem beschrieben. Durch dieses Modell werden u.a. verschiedene Schnittstellen spezifiziert, die für eine Erweiterung bzw. Integration von Workflowmanagementsystemen benutzt werden können. Zum Beispiel werden die Möglichkeiten der Interoperabilität von Workflowmanagementsystemen spezifiziert. Für die Integration eines Workflowmanagementsystems in externe Standardanwendungen und für die Übermittlung von Ereignissen und Nachrichten (*Notifications*) nach Außen ist spezifiziert worden. Da ein Benutzer in der betrieblichen Praxis häufig nicht nur mit einer Anwendung (*Workflow-Client*) zur Darstellung von Ereignissen und Nachrichten in nur einem Posteingangskorb konfrontiert wird, sondern auch E-Mail- oder Groupwaresysteme, wie z.B. MS Outlook oder Lotus Notes verwendet, kann es zu der misslichen Situation kommen, dass der Benutzer in mehreren unterschiedlichen Posteingangskörben prüfen muss, ob Ereignisse und Nachrichten eingetroffen sind, welche die Bearbeitung einer Aktivität initiieren. Durch eine Standardisierung der Ereignisse und Nachrichten kann erreicht werden, dass eine Anwendung sowohl als Client für das Workflowmanagementsystem als auch als Client für ein E-Mail-System oder als Client für ein Groupwaresystem verwendet wird.

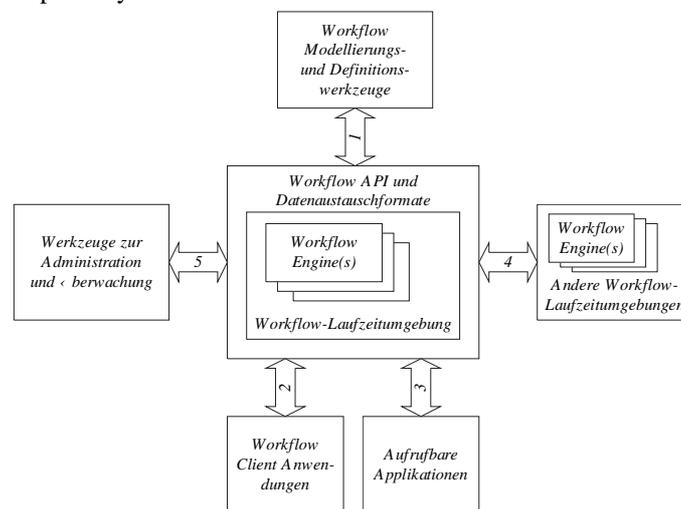


Abbildung 9: Workflowreferenzmodell /WfMC96/

- **Modellierung**
Die Modellierung geschieht über die Schnittstelle 1 mit den Modellierungswerkzeugen. Diese können Teil des Workflowmanagementsystems sein oder eigenständige Programme, die über die Schnittstelle 1 die Modellierungsdaten an das Workflowmanagementsystem weitergeben.
- **Steuerung**

Die Steuerung der Geschäftsprozesse übernimmt nach dem Referenzmodell die Workflow Laufzeitumgebung, bzw. die darin enthaltene *Workflow-Engine*. Die *Engine* ist der Kern des Modells, welcher die über Schnittstelle 1 gelieferten Daten verarbeitet.

- **Bearbeitung**
Die Bearbeitung der vom System zugeteilten Aufgaben geschieht über die Client-Anwendung, angebunden über Schnittstelle 2. Wie bei der Modellierung können hier integrierte oder externe Programme zum Einsatz kommen.
- **Datenbank**
Eine Datenbank ist nicht explizit aufgeführt, gehört aber teilweise zur Workflow Laufzeitumgebung, welche für seine Aufgaben Steuerungsdaten speichern muss.
- **Überwachung, Protokollierung**
Die hierfür notwendigen Werkzeuge und Applikationen werden über die Schnittstelle 5 angebunden.

2.1.3 Informations- und Kommunikationstechniken

CSCW-Systeme stellen teilweise spezielle Ansprüche an Informations- und Kommunikationstechniken. Aus der Sicht der technischen Infrastruktur müssen geeignete Netzwerke vorhanden sein und entsprechende Standards etabliert werden, die den effizienten Informationsaustausch zwischen Personen ermöglichen, die räumlich oder zeitlich getrennt sind. Sicherheitsüberlegungen spielen im Bereich der Kommunikationstechnik ebenfalls eine wichtige Rolle. Bei der Verbindung von räumlich getrennten Computern, die zum Zwecke des Datenaustausches miteinander verbunden sind, werden zwei Arten von Netzwerken unterschieden. Lokale Netze (*Local Area Network LAN*) sind prinzipiell Netze, die auf ein Privatgelände beschränkt sind. Durch sie werden bis zu mehreren hundert Computer miteinander verbunden. Durch Weitverkehrsnetze (*Wide Area Network WAN*) werden Tausende Computer miteinander verbunden. Sie sind in der Regel nicht regional begrenzt. Das derzeit größte Weitverkehrsnetz ist das Internet. Diese Arten von Netzwerken haben unmittelbaren Einfluss auf die CSCW-Systeme, die sich dieser Netzwerke bedienen. Dies kann entweder die Möglichkeiten zur Kommunikationsunterstützung, Koordinationsunterstützung, Kooperationsunterstützung oder die Sicherheit der Kommunikation betreffen.

Ein Ausgangspunkt für die Sicherheitsüberlegungen bei und während einer Kommunikation in Netzen ist die Art und Weise der Bedrohung. Anschließend daran werden die Methoden entsprechend der Bedrohung angewandt. Der Verlust der Vertraulichkeit bedeutet, dass die Daten nicht nur den berechtigten Kommunikationspartner übermittelt werden, sondern auch unbefugten Personen, die nicht über die entsprechende Berechtigung verfügen. Ein Verlust der Integrität von Daten bedeutet, dass die Daten aufgrund von Fehleingaben, fehlerhaften Programmen oder auch Verfälschung durch dritte Personen nicht mehr der Realität entsprechen. Ein Verlust der Verfügbarkeit liegt vor, wenn Daten nicht mehr erfasst und verarbeitet werden können oder wenn die Funktionalität eingeschränkt wird. Ein Verlust der Verbindlichkeit von Daten kann beispielsweise auftreten, wenn eine Teilnahme an einer Kommunikation durch einen Teilnehmer nachträglich bestritten wird, so dass der Empfang oder das Absenden von Daten nicht nachgewiesen werden kann.

Durch verschiedene Verfahren können diese Sicherheitsbedrohungen abgewiesen werden. Die Verschlüsselung dient dem Schutz der Vertraulichkeit und der Verbindlichkeit. Sie kann auf zwei Arten angewandt werden. Zu einen für die Verschlüsselung von Daten an einen bestimmten Empfänger und zum anderen für die Erzeugung von elektronischen Unterschriften. Zugriffskontrollverfahren dienen dazu, nur autorisierten Teilnehmern Nutzungsrecht zu gewähren, z.B. Lese- und Schreibrechte auf Daten oder Ausführungsrechte für Applikationen. Die Integritätssicherung muss bei Konferenzsystemen gewährleisten, dass Audio- und Videoübertragungen wirklich zusammengehören und nicht manipuliert sind. Eine derartige Integritätsverletzung kommt in der Regel nur bei Standbildern zusammen mit einer Audiosequenzübertragung vor. Bei der Verkehrserzeugung handelt es sich die Erzeugung von nutzlosen Fülldaten mit dem Ziel entweder die Kommunikation zu stören oder zu verhindern oder beteiligten Systeme durch Überlast zu sabotieren.

3 Arten von Softwareentwicklungsprozessen

In der Realität liegt häufig der Fall vor, dass mehrere Organisationseinheiten zusammenarbeiten müssen oder wollen, um gemeinsam an einer Softwareentwicklung zu arbeiten. Dies liegt in der Regel in der Tatsache begründet, dass Softwaresysteme immer häufiger komplex sind und die Entwicklung (ausgehend von einem Prozessmodell \bar{n} von der Anforderungsanalyse bis zum Systemtest -) nicht mehr von einer Person bzw. kleinen Personengruppe alleine bewerkstelligt werden kann. Die Zusammenarbeit (Kooperation, Kommunikation, Koordination) zwischen diesen Personen oder Personengruppen muss festgelegt werden. Dieser Festlegung kommt eine besondere Bedeutung zu, wenn diese Personen oder Personengruppe räumlich verteilt und verschiedenen Organisationseinheiten zugeordnet sind. Durch einfache und komplexe Vorgehensmodelle werden die Phasen und Tätigkeiten, die durch Personengruppen (Rollen) während der Softwareentwicklung durchgeführt werden müssen beschrieben.

3.1 Vorgehensmodelle

Durch Vorgehensmodelle (oder auch Phasenmodelle) werden auf abstrakter Ebene die Art und die Reihenfolge von durchzuführenden Aktivitäten beschrieben [PaSi94], [Ba96]. Darüber hinaus werden die Dokumente, die das Ergebnis von Aktivitäten sein sollen, definiert. Die einzelnen Vorgehensmodelle unterscheiden sich oft hinsichtlich der Anordnung der Aktivitäten. Den einzelnen Aktivitäten werden spezifische Rollen zugeordnet, die für diese Aktivitäten verantwortlich sind.

3.1.1 Das klassische Wasserfallmodell

Das Wasserfallmodell entstand durch die Einteilung des Entwicklungsprozesses in die in sich abgeschlossenen Phasen Analyse, Entwurf, Implementation und Test. Charakteristisch für das Wasserfallmodell ist der zeitlich sequentielle Ablauf der Phasen, wobei in jeder Phase nur Aktivitäten eines bestimmten Typs ausgeführt werden. Aufgrund des sequentiellen Ablaufs ist es beim Wasserfallmodell auch nicht möglich in frühere Phasen zurückzuspringen.

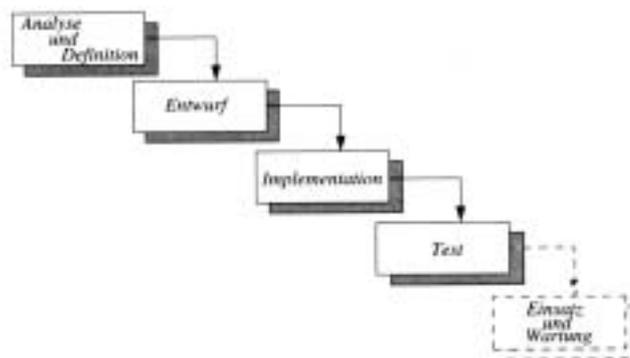


Abbildung 10: Das klassische Phasen- oder Wasserfallmodell

In der Analysephase werden die spezifischen Produktanforderungen ermittelt und daraus ein Pflichtenheft erstellt, welches die Vertragsbasis zwischen Auftraggeber und Auftragnehmer bildet. Beim Entwurf wird mit Hilfe des Pflichtenheftes die innere Struktur der Software festgelegt. Hierzu findet eine Zerlegung in Komponenten statt. Als Ergebnis dieser Phase wird die Softwarespezifikation erstellt, welche die einzelnen Komponenten und deren Zusammenwirken festlegt. In der Implementationsphase werden die Komponenten in einer konkreten Programmiersprache implementiert und einzeln auf ihre Funktionalität getestet. In der Testphase findet die sukzessive Integration der einzelnen Komponenten sowie ein abschließender Systemtest statt. Anschließend wird es auf nicht-operative Anforderungen getestet (z.B. Effizienz) und beim Anwender installiert. Die Phase Einsatz und Wartung gehört nicht zum eigentlichen Entwicklungsprozess, sondern erstreckt sich über die Zeit von der Installation der Software bis zum Einsatzende. Die hier anfallenden Aufgaben umfassen unter anderem die Korrektur von Fehlern, die Anpassung an andere Systemumgebungen und die Erweiterung der Funktionalität.

/PaSi94/. Das Wasserfallmodell bietet gegenüber der *Code and fix* Methode erhebliche Vorteile. Durch die qualitative und quantitative Abgeschlossenheit der einzelnen Phasen scheinen hier die Techniken zur Planung und Überwachung, wie z.B. Meilensteine, wieder zu greifen. Dies ist vor allem aus der Sicht der Projektverfolgung und Vertragsgestaltung nützlich. In der Praxis zeigen sich aber bald gravierende Mängel dieses Modells. Die Annahme, die einzelnen Phase seien vollständig abgeschlossen, entspricht nicht der Entwicklungspraxis. Beispielsweise entstehen während des Entwicklungsprozesses neue Nutzeranforderungen bzw. anfangs unscharfe Nutzeranforderungen werden erst im Verlauf des Projektes präzisiert. Das Wasserfallmodell berücksichtigt diese Tatsachen aber nicht, da das Wasserfallmodell eine Beteiligung des Auftraggebers, über die erste Phase hinaus nicht vorsieht. Weiterhin liegt beim Wasserfallmodell erst in der letzten Phase eine lauffähige Version vor. So werden Fehler und Missverständnisse aus früheren Phasen erst sehr spät sichtbar und deren Behebung aufwendig und teuer. Aufgrund dieser Kritik wurde das Phasenmodell auf verschiedene Weise modifiziert und es wurden neue Vorgehensmodelle entwickelt.

3.1.2 Das iterierte Wasserfallmodell

Beim iterierten Wasserfallmodell werden zwischen den einzelnen Phasen Rückkopplungspfeile eingefügt. So wird eine Rückkehr in frühere Phasen ermöglicht, um dort Fehler zu beseitigen, die erst in späteren Phasen aufgedeckt werden.

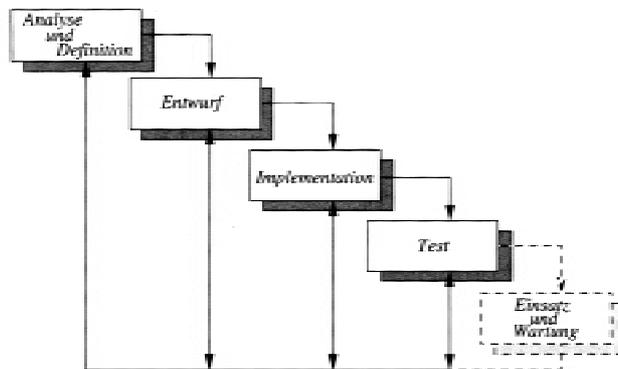


Abbildung 11: Das iterierte Phasenmodell

3.1.3 Schnelle Prototypenerstellung

Bei diesem Modell wird das iterierte Wasserfallmodell um eine Phase erweitert. Diese Phase dient der Verifizierung der Produktdefinition, um von vornherein Missverständnissen zwischen Auftraggeber und Auftragnehmer entgegenzuwirken. Diese Phase ergänzt die Analyse-Phase und wird iterativ durchlaufen, bis die Anforderungen an das Produkt feststehen. Dieses Verfahren eignet sich besonders für die Erstellung von Prototypen für die Benutzungsoberfläche.

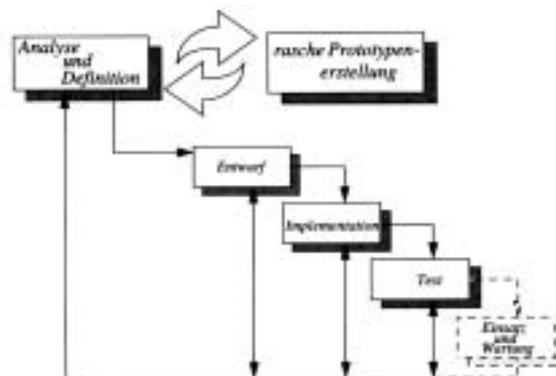


Abbildung 12: Das iterierte Prototypingphasenmodell

3.1.4 Evolutionäre Softwareentwicklung

Betrachtet man bei der schnellen Prototypenerstellung die Prototypen nicht als Wegwerfprodukte, sondern als frühe Versionen des Programms, gelangt man zu der Methode der evolutionären Softwareentwicklung. Dieses Vorgehensmodell sieht die Softwareentwicklung nicht mehr als linearen Prozess, sondern als eine Folge von Entwicklungszyklen. Am Ende eines jeden Zyklus steht eine verbesserte Version des Produkts. Als Folge existiert auch keine Wartungsphase mehr, da diese einfach durch weitere Entwicklungszyklen abgedeckt wird.

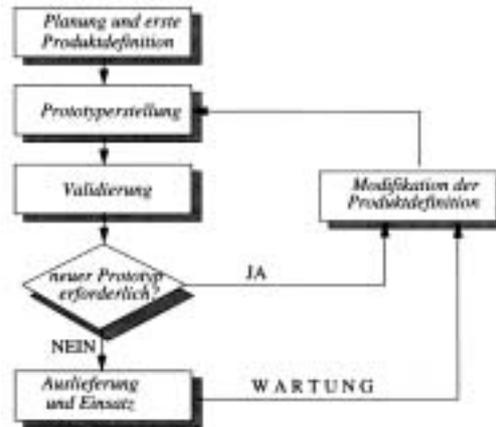


Abbildung 13: Modell der evolutionären Softwareentwicklung

Grundsätzlich wird zwischen zwei Wachstumsarten der evolutionären Prototypen unterschieden: inkrementell und evolvierend. Beim inkrementellen Wachstum wird die unvollständige Anfangslösung schrittweise zu einer vollständigen Lösung erweitert. Im Gegensatz dazu steht beim evolvierenden Wachstum nicht die Vollständigkeit im Mittelpunkt, sondern die Konsistenz und Angemessenheit. Der große Vorteil dieses Vorgehensmodells ist die ständige Interaktion mit dem Auftraggeber. Durch diesen Kontakt werden frühzeitig Missverständnisse entdeckt und neue (oder geänderte) Anforderungen können durchgehend berücksichtigt werden. Als weiterer Vorteil bleibt die gesteigerte Motivation zu nennen, da sich durch die laufenden Prototypen Erfolgserlebnisse einstellen. Andererseits treten bei dieser Art der Softwareentwicklung auch neue Probleme auf. Zum einen bereitet die Dokumentation große Schwierigkeiten, da auch diese evolutionär erweitert werden muss. Zum anderen muss die Software extrem übersichtlich und leicht modifizierbar gestaltet werden, da an ihr fortlaufend Änderungen vorgenommen werden. Auch ist das Management bei diesem Vorgehensmodell vor größere Probleme gestellt, da nur schwer abzuschätzen ist, wie schnell sich die Prototypen weiterentwickeln werden. Folglich ist eine Planung der Ressourcen kaum möglich.

3.1.5 Das Spiralmodell

Das Spiralmodell ist ein risiko-basiertes Rahmenwerk, in das beliebige Prozessmodelle eingefügt werden können. Es stellt somit ein generisches Prozessmodell dar, das an das jeweilige Projekt angepasst werden muss. Der Entwicklungsprozess wird als iterativer Prozess verstanden, bei dem jede Windung der Spirale einer Iteration entspricht. Eine Iteration besteht aus den vier Stufen

- Festlegung von Zielen, Alternativen und Rahmenbedingungen
- Evaluierung der Alternativen, Erkennen und Reduzieren von Risiken
- Realisierung und Überprüfung der Zwischenergebnisse
- Planung der Projektfortsetzung.

Die folgende Grafik veranschaulicht das Spiralmodell:

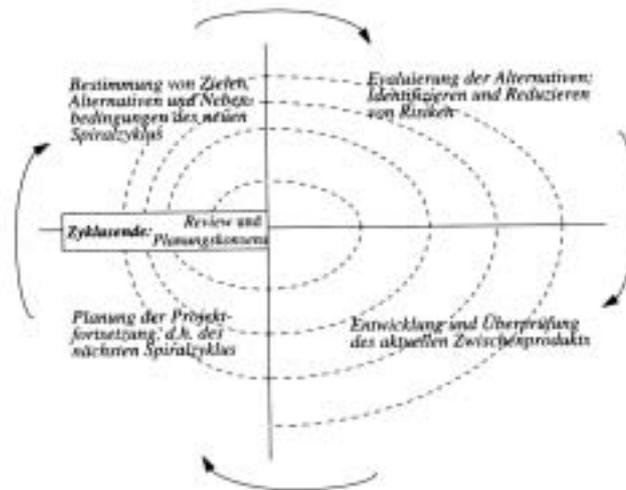


Abbildung 14: Das Spiralmodell des Softwareentwicklungsprozesses

In der ersten Phase einer Iteration werden zunächst die Anforderungen an das zu erstellende Teilprodukt und mögliche alternative Vorgehensweisen erarbeitet. Hier steht beispielsweise zur Diskussion, ob das Teilprodukt selber entwickelt oder hinzu gekauft wird bzw. welche Entwicklungsmethoden- und Werkzeuge hierfür eingesetzt werden. Aufbauend auf diesen Ergebnissen werden in der zweiten Stufe die Alternativen auf Risiken untersucht und bewertet. Hierzu können die unterschiedlichsten Techniken wie *Prototyping*, *Benchmark* - Tests oder Marktforschungen eingesetzt werden. So bildet die Entscheidung für eine der Alternativen den Abschluss der zweiten Stufe. Im dritten Abschnitt der aktuellen Iteration findet die Realisierung und der Test der gewählten Alternative statt. Hier können beliebige Methoden des Software - Engineering, wie z.B. evolutionäre oder transformationelle Entwicklung, eingesetzt werden, wobei der gewählte Ansatz nur für die aktuelle Iteration bindend ist. Abschließend findet nach jedem Iterationsschritt ein *Review* statt. Hier werden die Projektfortschritte des letzten Zyklus bewertet und die wirtschaftlichen Risiken einer Projektfortsetzung ermittelt. Sind diese Risiken zu hoch, endet die Spirale nach dieser Windung. Andernfalls werden zu diesem Zeitpunkt die nächsten Windungen inhaltlich und organisatorisch geplant.

3.1.6 Das V-Modell

Das V-Modell versteht den Entwicklungsprozess als eine Folge von Tätigkeiten und deren Ergebnissen. Zu jeder Tätigkeit, im V-Modell Aktivität genannt, existiert eine explizite Arbeitsanleitung. Diese gibt an, welche Produkte als Eingangsprodukte vorliegen müssen, wo diese zuletzt modifiziert wurden, welche Produkte in dieser Aktivität erzeugt bzw. modifiziert werden und in welcher Aktivität diese weiter behandelt werden. In der Praxis findet die Systementwicklung in Ausbaustufen statt, d.h. das zu entwickelnde System wird als Ganzes geplant, aber in Teilen erstellt. Bei einer benutzerorientierten Entwicklung sollte möglichst früh eine erste Version an den Auftraggeber ausgeliefert werden. So kann dieser die Basisfunktionalität prüfen und gegebenenfalls Missverständnisse klären. Eine andere Vorgehensweise orientiert sich zunächst an risikoreichen Teilen, um das Projektrisiko frühzeitig zu verringern. Der Entwicklungsprozess ist im V-Modell aufgeteilt in vier Submodelle, die über einen Austausch von Produkten eng miteinander zusammenarbeiten.

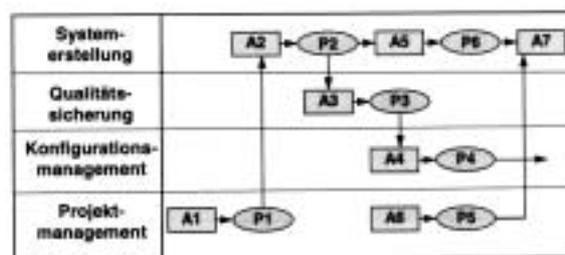


Abbildung 15: Beispiel für den Aktivitäts- und Produktfluss über die vier Submodelle

Jedes der Submodelle besteht aus Hauptaktivitäten, die sich wiederum in Teilaktivitäten aufteilen lassen. Das wichtigste Submodell stellt die Systemerstellung (SE) dar, in dem die Softwareeinheiten erstellt werden. Das Projektmanagement (PM) dient der Planung, Initiierung und Kontrolle des Entwicklungsprozesses. In der Qualitätssicherung (QS) werden die Qualitätsanforderungen für die zu entwickelnde Software sowie wichtige Prüfverfahren vorgegeben. Das Submodell Konfigurationsmanagement (KM) beschäftigt sich mit der Verwaltung aller erstellten Produkte und der Kontrolle der Änderungen. Die folgende Abbildung zeigt die Hauptaktivitäten des Submodells Systemerstellung in ihrer typischerweise V-förmigen Anordnung.



Abbildung 16: Submodell Systemerstellung

3.2 Rollen

Die Definition der Rollen und die Zuordnung von Personen zu einzelnen Rollen erfolgt vor Beginn eines Softwareprojektes. Unter der Definition einer Rolle versteht man die Festlegung der Aufgaben mit den dafür notwendigen Erfahrungen, Kenntnissen und Fähigkeiten, Kompetenzen und Verantwortungen und somit den notwendigen Qualifikationen an das Personal. Eine Rolle wird durch Aktivitäten und seine Verantwortlichkeit beschrieben, gemäß den Projekt- bzw. Unternehmensgegebenheiten, den Aufgaben und Prioritäten. Eine Rolle kann verantwortlich sein für mehrere Aktivitäten oder umgekehrt können auch mehrere Rollen nur einer Aktivität zugeordnet sein. Die Person, die eine Rolle einnimmt wird als Rolleninhaber bezeichnet. Ein Rolleninhaber kann mehrere Rollen einnehmen. Eine Rolle kann aber auch von mehreren Rolleninhabern eingenommen werden. Jede Person hat eine sogenannte Primärrolle und eine (oder mehrere) Sekundärrolle(n), wobei das Hauptaugenmerk auf der Primärrolle liegt. Maßgeblich für den Erfolg einer Softwareentwicklung ist die Zusammenarbeit und Kommunikation der beteiligten Personen eines Projektes. Es gibt auch Zielkonflikte zwischen den an Aktivitäten beteiligten Personen, z.B. zwischen Vertriebsmitarbeiter und Projektleiter oder zwischen Budget-Controller und Entwickler. Diese Zielkonflikte ergeben sich aus den Aufgaben und den Verantwortlichkeiten der einzelnen Rollen. Zielkonflikte werden u.a. auch in sogenannten *Management-Boards* diskutiert und gelöst. Die Beziehungen zwischen den beteiligten Personen und die Wechselwirkungen zwischen den Erledigungen der Aufgaben sind vielfältig und tragen maßgeblich zum Erfolg oder Misserfolg einer Softwareentwicklung bei. Projekte funktionieren, wenn die richtigen Leute unter Verwendung der geeigneten Methoden, Sprachen und Werkzeuge vernünftig miteinander arbeiten. Das Ziel der rollenbasierte Softwareentwicklung ist der Interessensausgleich zwischen den beteiligten Rollen sowie die Kooperation zwischen dem beteiligten Personen durch Verständnis für die rolleninduzierten Aufgaben, Auffassungen und Probleme

3.2.1 Projektmanager

Allgemein kann gesagt werden, dass ein Projektmanager die Aktivitäten der Softwareentwicklung überblicken und den Interessensausgleich zwischen den beteiligten Personen herbeiführen muss. Hierzu benötigt der Projektmanager folgende Qualifikation:

- fachliche Qualifikation (betriebswirtschaftliche Kenntnisse, fachliche Kenntnisse, etc.)
- Projekt-Management-Qualifikation (Planung, Schätzung, Planverfolgung, Zeitmanagement, Verhandlungsgeschick)

- Führungseigenschaften (Kontaktstärke, Kommunikationsfähigkeit, Entscheidungsfähigkeit, Bereitschaft zur Delegation, Fähigkeit zur Motivation usw.)
- Persönliche Eigenschaften (Verantwortungswille, Belastbarkeit, Beharrlichkeit, Teamgeist, Geduld, Konfliktfähigkeit, Fähigkeit zum Umgang mit unterschiedlichen Typen von Menschen)
- Strategisches Denken
- Kreativität und innovatives Verständnis zur Einführung neuer Verfahren und Methoden
- Problem- und Konfliktmanagement
- Fachübergreifendes Denken
- Erfahrung in der Projektabwicklung

Der Projektmanager erstellt während der Bewältigung seiner Aufgaben die Dokumente Projektplan, Projektfortschrittsberichte, Individuelle Zielvereinbarung (quantitative Ziele für alle Teammitglieder) Projektziele.

Projektplanung	Bestandteile: Kosten- und Aufwandsschätzung, Termin- und Meilensteinplanung
Projektverfolgung	Bei Bedarf Anpassungen vornehmen, Überwachung der Produktivität
Zusammenstellung des Projektteams	Personalkosten und die Qualifikation der Mitarbeiter beachten
Mitarbeiterführung	Mitarbeiter fördern, fördern und motivieren
Projektorganisation	Teamorganisation, Infrastruktur (z.B. Projektstatussitungen und deren Tagesordnungen festlegen)
Festlegung einer Projektkultur	Etablierung von Werten wie Termineinhaltung, Qualität, Kundenbewusstsein oder Kundenorientierung
Abnahme von Ergebnissen	In Form von schriftlichen Dokumenten
Einführung eines Berichtswesens	d.h. wesentliche Abweichungen explizit und unmissverständlich nennen, formale Berichtswege beachten und auf passenden Wegen miteinander kommunizieren
Berichten an Vorgesetzte	z.B. an die Geschäftsführung

Tabelle 1: Aufgaben des Projektmanagers

Der Projektmanager kommuniziert mit allen Mitgliedern des Projektteams. Außerdem berichtet er an seine Vorgesetzten (z.B. Geschäftsführung) und tritt meist auch in Kontakt mit dem Kunden. Er nimmt an Sitzungen des Projekt-Management-Boards, des Risiko-Management-Boards und des Change-Management-Boards teil.

3.2.2 Risikomanager

Ziel des Software-Risikomanagements ist es, die Wechselbeziehungen zwischen Risiken und Erfolg zu formalisieren und in anwendbare Prinzipien und Praktiken umzusetzen. Ein Risiko beschreibt die Möglichkeit, dass eine Aktivität einen körperlichen oder materiellen Verlust oder Schaden zur Folge hat. Der Risikomanager benötigt folgende die Qualifikation:

- Vorsicht und Skepsis
- Fähigkeit zur konstruktiven Kritik
- fachliches Know-How (Beherrschung der Methoden der Risikoanalyse und der Risikovermeidung/-verringerung)
- Erfahrung,

um die Dokumente Risikoanalyse (Einschätzung, Vorhersage etc.), Risikoberichte (Regelmäßige Erstellung, Weiterleitung an Projektmanager, Kundenbetreuer, Marketiers, Vertriebsleute und Geschäftsführung) und Protokolle der Sitzungen des Risiko-Management-Boards erstellen zu können.

Regelmäßige Abschätzungen der aktuell größten Risiken	z.B. technische Risiken (Erfüllung von Anforderungen), technologische Risiken (Softwaretechnologie), Kostenrisiken (Einhaltung des Budgets), Zeitplanrisiken (Erfüllung des Zeitplans), Marktrisiken oder Finanzierungsrisiken
Risikovorhersage	Eintrittswahrscheinlichkeit
Risikoeinschätzung	Konsequenzen / Schadenspotential
Erstellung der Risikostrategie	what if? , Entscheidungsalternativen
Identifikation der dringenden Maßnahmen zur Vermeidung / Verringerung von identifizierten Risiken	Personalakquise, Lieferantenmanagement, Performanzuntersuchungen uvm.

Tabelle 2: Aufgaben des Risikomanagers

Der Risikomanager nimmt, wenn auch unregelmäßig, an Projektsitzungen teil und diskutiert Planabweichungen mit dem Projektmanager und dem Budget-Controller. Er veranstaltet und leitet außerdem regelmäßige Risiko-Management-Boards. Zur Einschätzung von Risiken muss der Risikomanager mit dem Kundenbetreuer und externen Lieferanten kommunizieren.

3.2.3 Anforderungsanalytiker

Das Ziel der Anforderungsanalytiker ist es, die Grundlage für das Bauen des richtigen Systems zu erstellen. Dazu führt er eine Anforderungsanalyse aus, die aus der Ermittlung von Anforderungen an ein Softwaresystem besteht. Die Qualifikation, über die der Anforderungsanalytiker verfügt, umfasst

- Geduld
- Sorgfalt
- Lernbereitschaft
- Kenntnisse im Anwendungsbereich
- Hohe Kommunikationsbereitschaft, da der Kontakt mit dem Kunden einen hohen Stellenwert hat
- Abstraktes Denken

Initiales Verständnis der wesentl. Anforderungen	so schnell wie möglich
Gruppierung der Anforderungen	zu sinnvollen Teilmengen (im Sinne einer Release-Planung und einer Priorisierung)
Erstellung und Verabschiedung des Anforderungsdokumentes	enthält die Anforderungsbeschreibung
Förderung der Auseinandersetzung mit dem Anforderungsdokument	Entwickler, Kunden, spätere Anwender, Kundenbetreuer
Validation des Anforderungsdokumentes	Gültigkeit der Anforderungen, Konsistenz, Vollständigkeit, Realisierbarkeit, Sinnhaftigkeit der Prioritäten

Tabelle 3: Aufgaben des Anforderungsanalytikers

Der Anforderungsanalytiker verfasst folgende Dokumente:

- Anforderungsdokument: enthält alle Anforderungen an das zu entwickelnde Softwaresystem
- Vorläufige Einschätzung der Machbarkeit: ist das Projekt überhaupt realisierbar?
- Änderungshistorie der Anforderungsdokumentes: wer hat wann was geändert?

Der Anforderungsanalytiker fördert die Auseinandersetzung mit dem Anforderungsdokument mit dem Entwickler, Kunden und Kundenbetreuer. Der Anforderungsanalytiker nimmt an den Vertragsverhandlungen mit dem Kunden teil; er ist in der Release-Planung, diskutiert den Projektplan mit dem Projektmanager und besucht das Risiko-Management-Board.

3.2.4 Spezifizierer

Der Spezifizierer ist derjenige, der \bar{n} aufbauend auf dem Anforderungsdokument \bar{n} das Verhalten des zu entwickelnden Software-Systems aus Anwendersicht beschreibt. Die Spezifikation enth \bar{a} keine Informationen dar \bar{u} ber, wie dieses Verhalten realisiert werden soll. Das Ergebnis der Spezifikation ist das Spezifikationsdokument.

Erstellung des Spezifikationsdokumentes	enth \bar{a} die vollst \bar{a} ndige Spezifikation
Vorstellung des Spezifikationsdokumentes	gegen \bar{u} ber Kunden, Entwicklungsteam
Abgleich der Anforderungen mit der Spezifikation	sind alle Anforderungen ber \bar{u} cksichtigt?
Identifikation von Testf \bar{a} llen	z.B.: Er \bar{t} nt nach Abnehmen des H \bar{r} sters sp \bar{a} testens nach drei Sekunden ein Ton?

Tabelle 4: Aufgaben des Spezifizierers

Der Spezifizierer wirkt bei der Verabschiedung von Testszenarien, die der Tester dann benutzt, um das fertige Softwaresystem zu testen, und bei der Verabschiedung des Entwurfsdokumentes mit. Vor der Auslieferung muss der Spezifizierer dann noch die Testprotokolle \bar{u} berpr \bar{u} fen.

Der Spezifizierer verfasst folgende Dokumente:

- Spezifikationsdokument: enth \bar{a} die Spezifikation
- Protokoll \bar{u} ber die Abstimmung des Spezifikationsdokumentes mit dem Auftraggeber
- Protokoll der Vorstellung des Dokumentes f \bar{u} r den Entwerfer / Entwickler / Prototyprealisierer
- Testf \bar{a} lle aus Spezifikation
- Dokument zur Er \bar{t} rtung des Zusammenhangs zwischen Anforderungsdokument und Spezifikationsdokument

Der Spezifizierer sollte folgende Qualifikationen besitzen:

- Kommunikationsf \bar{a} higkeit
- Abstraktionsf \bar{a} higkeit
- Kenntnisse, verschiedener Spezifikationssprachen und ihrer Anwendungsgebiete

Der Spezifizierer arbeitet mit dem Tester zusammen (Testf \bar{a} lle, Testszenarien). Er stellt dem Designer, dem Entwickler und dem Kunden das Spezifikationsdokument vor.

3.2.5 Designer

Der Designer geht von der Anforderungsanalyse und dem Spezifikationsdokument aus und erstellt eine Strukturierung des zu erstellenden Softwaresystems. Als Ergebnis steht der Softwareentwurf. Der Entwurf umfasst die Dokumente

- Entwurfsdokument: Komponentenstruktur, Modulstruktur, Beschreibung der einzelnen Module, Zusammenfassung zu Prozessen, Beschreibung der Kommunikationsprotokolle zwischen Prozessen, Aufrufstrukturen, Nennung der wiederverwendeten Bestandteile, Beschreibung der neuen, wiederverwendbaren Komponenten
- Entwurfshistorie: inklusive der nicht gew \bar{a} hlten Entwurfsalternativen

Identifikation der Komponenten	Grobentwurf
Zerlegung der Komponenten in Module	Feinentwurf
Aufteilung der statischen Modulstruktur in Betriebssystemprozesse	Es k \bar{a} nnen mehrere gleichartige Prozesse parallel existieren. Verschiedenartige Prozesse k \bar{a} nnen miteinander kommunizieren
Protokollierung der Entwurfsentscheidungen	um den Entwurf und sp \bar{a} tere \bar{a} nderungen

dungen und der Entscheidungsfindung	nachvollziehbar zu machen
Organisation von Reviews des Entwurfsdokumentes	Entwurfsdokument diskutieren

Tabelle 5: Aufgaben des Designers

Der Designer benötigt folgende Qualifikation:

- Kenntnisse des Anwendungsbereichs
- Software-Technologie-Kenntnisse
- Erfahrung im Software-Entwurf
- Kommunikationsfähigkeit mit dem technischen Personal des Anwenders bzw. Kunden
- Abstraktes Denken (vom fachlich Abstrakten zum software-technisch Konkreten)
- Konzeptionelles Denken (nicht im Detail versinken)
- Kreativität

Der Designer nimmt auch an der Entwicklungsplanung und an Reviews des Entwicklungsprozesses teil. Der Spezifizierer bespricht mit ihm die Spezifikation. Außerdem besucht er das Change-Management-Board.

3.2.6 Systemtechniker

Neue Softwaresysteme müssen immer öfter in existierende Software- und Hardware-Infrastrukturen integriert werden. Die Menge der Software- und Hardware-Systeme, mit der ein Softwaresystem integriert werden soll, wird als Zielplattform bezeichnet. Eine Hauptaufgabe des Systemtechnikers ist die Definition der Systemarchitektur und die Erstellung der Integrationsdokumentation, wobei die neu zu entwickelnde Software als eine Komponente aufgefasst wird. Ebenso sollte er an den Reviews der Systemarchitektur und des Projektplans teilnehmen.

Prüfung der Integrierbarkeit der einzelnen Bestandteile der Zielplattform	DBMS, UIMS, Betriebssysteme, Hardware, Telekommunikationsinfrastruktur, Kommunikationsprotokolle
Vergleich mit alternativen Zielplattformen	Klappt es damit besser?
Vergleich mit Systemen, die auf der gleichen Zielplattform laufen	Welche Probleme gibt es bei denen?
Beschaffung und Installation der benötigten Hard- und Software	Neues Betriebssystem, neue Rechner, etc.
Szenarioartiges Ausprobieren und Dokumentation der Prüfergebnisse	Passen die zu integrierenden Elemente zusammen?
Berücksichtigung von Verteilungsszenarien und Prognose des Performanzverhaltens	Stichwort: thin und fat clients

Tabelle 6: Aufgaben des Systemtechnikers

Der Systemtechniker sollte folgende Qualifikation haben:

- Überwiegend technisches Interesse
- Kommunikationsfähigkeit mit Zulieferern
- Sorgfalt in der Installation und Betrieb von Fremdsystemen (Hardware und Software)

Wegen der immer häufiger auftretenden Integration neuer Softwaresysteme in existierende Software- und Hardware-Infrastrukturen muss der Systemtechniker auch mit dem Kunden kommunizieren. Außerdem muss er mit Zulieferern in Kontakt stehen. Da er auch an Reviews des Projektplanes teilnimmt, kommuniziert der Systemtechniker auch mit dem Projektmanager.

3.2.7 Entwickler / Programmierer

Ziel der Programmierung ist die Umsetzung des Entwurfs in ausführbare Programme, die den Anforderungen, die in der Anforderungsanalyse ermittelt wurden, entsprechen. Ausführbare Programme umfassen Programmtext, Testfallspezifikation, Inline-Dokumentation (Kommentare im Programm selbst) und Dokumentation.

Codierung des Entwurfs	Umsetzung in Programmcode
Wiederverwendung	bereits existierende Komponenten wiederverwenden bzw. selbst wiederverwendbare Komponenten bereitstellen
programmtechnische Anbindung externer Systeme	Verknüpfung mit externer Software
Testen von Prozeduren und Modulen	kleine Programmteile selbst testen, Entlastung des Testers
Messen der eigenen Produktivität	Qualität, benötigte Zeit, Menge
Ermittlung von Aufwands- und Fortschrittskennzahlen	Wie weit ist das Projekt fortgeschritten? Wie viel Aufwand war dazu nötig?

Tabelle 7: Aufgaben des Entwicklers

Der Entwickler nimmt an der Projektplanung und an den Projektbesprechungen teil, er setzt die Qualitätsmanagement-Richtlinien in Kraft und unterstützt die projektexterne Qualitätssicherung. Der Entwickler sollte folgende Qualifikation besitzen:

- Programmierungskennnisse
- grundlegende Kenntnisse des Anwendungsbereiches
- Kenntnisse der verwendeten Entwurfsmethode
- Skepsis gegenüber den eigenen Arbeitsergebnissen
- Kritikfähigkeit
- mathematisches Denkvermögen
- Abstraktionsvermögen
- Kreativität
- Präzision

Der Entwickler nimmt an der Projektplanung und an den Projektbesprechungen teil, kommuniziert somit auch mit dem Projektmanager. Er unterstützt auch die Qualitätsmanagement-Richtlinien. Der Entwickler bespricht mit dem Tester die Testergebnisse und mit dem Designer bei Bedarf den Entwurf. Der Spezifizierer bespricht mit ihm die Spezifikation. Außerdem setzt sich der Entwickler mit dem Anforderungsdokument auseinander.

3.2.8 Tester / Qualitätssicherer

In der Praxis sind Tests die am weitesten verbreitete Qualitätssicherungsmaßnahme, deshalb auch die ausgezeichnete Rolle des Testers. Testen umfasst nicht die Fehlerkorrektur.

Testplanung	für alle Tests außer Modul- und Klassentest
Testdurchführung	für alle Tests außer Modul- und Klassentest
Testprotokollierung	für alle Tests außer Modul- und Klassentest
Testdokumentation	alle Tests sowie deren Ergebnisse auführen
Ermittlung typischer Fehlerursachen	als Hilfe für den Entwickler
Kommunikation typischer Fehlerursachen	mit dem Entwickler

Bewertung des Testergebnisses	fehlerfrei, fehlerbehaftet
-------------------------------	----------------------------

Tabelle 8: Aufgaben des Testers

Der Tester nimmt auch an der Projektplanung teil, bei Freigabebesprechungen. Wie der Entwickler setzt er die Qualitätsmanagement-Richtlinien in Kraft und unterstützt die projektexterne Qualitätssicherung.

Der Tester verfasst folgende Dokumente:

- Explizite Definition der Testziele und Testergebnisse
- Testplan: Testplanidentifikation, Verantwortlichkeiten und Zeitplan, Testvoraussetzung, Testfälle, Testnachweise

Der Tester besitzt folgende Fähigkeiten:

- Kenntnis verschiedener Testverfahren
- Erfahrungen in der Entwicklung
- Sorgfalt
- Sachlichkeit, Fingerspitzengefühl

Der Tester kommuniziert bei der Projektplanung und den Freigabebesprechungen mit dem Projektmanager und unterstützt die Qualitätsmanagement-Richtlinien. Die Testergebnisse bespricht er mit dem Entwickler. Der Spezifizierer unterstützt ihn mit der Entwicklung von Testfällen und Testszenarien.

3.2.9 Qualitätsmanager

Allgemein kann gesagt werden, dass Qualität die Übereinstimmung mit den Anforderungen ist. Um dies sicherzustellen verfasst der Qualitätsmanager im Rahmen seiner Aufgaben die folgenden Dokumente:

- Qualitätssicherungsplan: Ergebnis der Qualitätsplanung
- Organisation und Berichte der Qualitätssicherung: Messdaten, Tests oder Inspektionen, Reviews, Audits
- Verfahrensanweisungen
- Ergebnisse der Post-Mortem-Analysen.

Der Qualitätsmanager benötigt folgende Qualifikation:

- Sorgfalt
- Diplomatie (Ausgleich zwischen den Interessen der einzelnen Rollen)
- Fähigkeit, die Kommunikation in Abhängigkeit vom Adressaten zu gestalten
- Geduld
- Hartnäckigkeit
- Mathematisches Denkvermögen

Qualitätsplanung	Festlegung der Qualitätsanforderungen an den Prozess und das Produkt in verbindlicher Form
Qualitätslenkung und -sicherung	setzt die Qualitätsplanung um, steuert, überwacht und korrigiert den Entwicklungsprozess mit dem Ziel, die vorgegebenen Anforderungen zu erfüllen
Qualitätsprüfung	führt die im Rahmen der Qualitätsplanung festgelegten Maßnahmen durch und überwacht, ob die konstruktiven Maßnahmen umgesetzt wurden
Koordination der projekt-internen und projektexternen QS-Maßnahmen	verschiedene Standards koordinieren
Erstellen von Verfahrensanweisungen	z.B. Software-Entwurf: wie muss ein SE aussehen? Welche Beschreibungssprache? Etc.
In Kraft setzen von	Werden die Maßnahmen durchgeführt? Feedback?

Verfahrensweisungen	
Organisation von Post-Mortem-Analysen	Was hat gut geklappt und was weniger gut?

Tabelle 9: Aufgaben des Qualitätsmanagers

Der Qualitätsmanager kommuniziert mit dem gesamten Projektteam, da im Normalfall die Qualitätsmaßnahmen alle Mitglieder des Teams betreffen. Er nimmt am Projekt-Management-Board und am Change-Management-Board teil. Er kommuniziert auch mit externen Qualitätssicherern.

3.2.10 Wartungsexperte

Mit der Einführung einer Software in den produktiven Betrieb beginnt die Wartung und Pflege dieser Software. Unter Wartung versteht man das Beseitigen von Fehlern, unter Pflege die Anpassung von Software an neue Rahmenbedingungen.

Stabilisierung und Korrektur	Fehlerbeseitigung
Optimierung	Ergonomie, Performanz, andere nicht-funktionale Anforderungen
Anpassung	an neue System-Software, Vorgaben bezüglich Benutzeroberflächen, Gesetzesänderungen, betriebl. Organisationsvorgaben
Erweiterung	funktionale Erweiterungen
Budgetplanung Wartung	Wie viel kostet die Wartung? Wartungsverträge?
Änderungsmanagement	Erfassung von Fehlern / Problemen, Änderungsursachen und -probleme, Entscheidung über die Durchführung von Änderungen, Release-Planung, Release-Freigabe

Tabelle 10: Aufgaben des Wartungsexperten

Der Wartungsexperte nimmt außerdem an der Release-Planung, Projektplanung für die Weiterentwicklung und Technologieplanung teil. Der Wartungsexperte erstellt folgende Dokumente:

- Wartungsplan
- Änderungsprotokolle

Der Wartungsexperte benötigt folgende Qualifikationen:

- Kenntnis der Verfahren des Änderungsmanagements
- Kenntnis des Anwendungsbereichs (so dass die Konsequenzen von Fehlern eingeschätzt werden können)
- Kenntnis der Technologie
- Kenntnis der Software-Prozessmodelle (die firmenweit zum Einsatz kommen)
- Kenntnis von Verfahren der Aufwandsabschätzung
- Kenntnis der von Änderungen potentiell betroffenen Rollen und ihrer Interessen

Der Wartungsexperte steht zum einen mit dem Kunden in Verbindung, zum anderen nimmt er auch am Change-Management-Board teil.

3.2.11 Konfigurationsmanager

Die Ziele des Software-Konfigurationsmanagements sind die Sicherstellung der Sichtbarkeit, Verfolgbarkeit und Kontrollierbarkeit eines Produkts und seiner Teile im Lebenszyklus, außerdem die Überwachung der Konfigurationen während des Lebenszyklus sowie die Sicherstellung, dass jederzeit auf vorangegangene Versionen zurückgegriffen werden kann, damit Änderungen nachvollziehbar und überprüfbar sind.

Konfigurationsplanung	Erstellung eines KM-Plans; Sicherstellung, dass die Releases entsprechend den Bedürfnissen von Anwendern und Kunden erstellt werden; Abstimmung der Release-Planes mit sonstigen Projekterfordernissen
Koordination der Erstellung von Release Notes	Release Notes sind für den Kunden / Anwender eine wichtige Orientierungshilfe
Konfigurationsidentifikation	Identifikation geeigneter Kes
Konfigurationskontrolle	
Überprüfung der Kes	
Statusverfolgung / KE- und Versionsverwaltung	Bereitstellung aller Kes und Versionen gemäß der erlaubten Zugriffe, Mechanismen zur Erzeugung neuer Kes und Versionen gemäß KE-Plan
Auswahl geeigneter Werkzeug- / Prozessunterstützung	Konformität zur Organisation und Kompatibilität zum Software-Prozess
System Build	Erstellen einer auslieferbaren Konfiguration
Auslieferung, Überprüfung	Test der auszuliefernden Konfiguration im Hinblick auf Installierbarkeit, Aufrufbarkeit, Start wesentlicher Funktionalitäten

Tabelle 11: Aufgaben der Konfigurationsmanagers

Außerdem leitet der Konfigurationsmanager das *Change-Management-Board* und nimmt an der Projektplanung teil. Er ermittelt anzukündigende Releasetermine. Der Konfigurationsmanager verfasst folgende Dokumente:

- Konfigurationsmanagement-Plan
- Release Notes
- Patch-Notes

Der Konfigurationsmanager benötigt folgende Qualifikationen:

- Kenntnis der Prozesse und der unterstützenden Werkzeuge für das Konfigurationsmanagement
- Kenntnis des Softwareprozesses
- Sorgfalt

3.2.12 Risiko-Management-Board

Das *Risk-Management-Board* hat den Zweck die Kommunikation der Risiken einzelner Projekte aufrecht zu halten, auf typische Risiken hinzuweisen und bei Bedarf entsprechende Gegenmaßnahmen zu identifizieren und einzuleiten. Es besteht aus den folgenden Teilnehmern:

- Projektmanager
- Risiko-Manager (verantwortlich)
- Anforderungsanalytiker
- Vorsitzender der Geschäftsführung (teilweise)
- Kundenbetreuer (auf Einladung)

3.2.13 Projekt-Management-Board

Am *Project-Management-Board* nehmen der Projektmanager (verantwortlich), der Qualitätsmanager, der Konfigurationsmanager, der Risikomanager (teilweise), der Vorsitzender der Geschäftsführung (teilweise und nur bei großen Projekten) teil. Es dient der gemeinschaftliche Feststellung des Projektfortschritts, der Analyse der aktuellen Risiken und der Analyse potentieller Konflikte zwischen Rollen und Identifikation von Maßnahmen zu ihrer Überwindung.

3.2.14 Change-Management-Board

Am *Change-Management-Board*, welches das Ziel hat die *f*nderungsanforderungen zu priorisieren, nehmen der verantwortliche Projektmanager, der Kundenbetreuer, der Designer, der Konfigurationsmanager, der Wartungsexperte, der Produktmanager (bei Bedarf) und der Qualitätsmanager teil. Durch den Konfigurationsmanager werden die *f*nderungsanforderungen zu Update-Einheiten (*Release, Patches*) zusammengefasst.

4 Anwendungsgebiete f, r Softwareentwicklungsprozesse

Unter der Annahme, dass sich Softwareentwicklungsprozesse, als auch deren automatische Unterst, tzung sich bei sehr unterschiedlichen Anwendungsgebieten auch sehr unterscheiden, wird als exemplarisches Anwendungsgebiete das nachfolgende Szenario *Electronic Commerce Systeme* ausgew, hlt. Die Besonderheiten dieses Anwendungsgebiets wird zun, chst genauer beschrieben.

Der konventionelle Gesch, ftsverkehr ñ der nicht elektronisch unterst, tzte Gesch, ftsverkehr ñ erfolgt heutzutage mittels verschiedener Medien wie Brief, Telefon und Fax in konventioneller Art und Weise. Der elektronische Gesch, ftsverkehr nutzt andere Medien wie z.B. Electronic Mail, EDI, WWW und Internet /Zw96/,/II98/. Die Partner beim Gesch, ftsverkehr ñ sei es elektronisch oder konventionell ñ sind abstrakt betrachtet zun, chst einmal Anbieter und Kunde, jedoch k, nnen je nach Art des Gesch, ftsverkehrs die Partner auch im speziellen als Anbieter und Nachfrager, Empf, nger und Lieferant oder Hersteller und Zulieferer, aber auch als Management und Mitarbeiter bezeichnet werden. Diese Rollen, Anbieter und Kunde, k, nnen von Unternehmen oder Endverbrauchern eingenommen werden. Wenn sowohl die Rolle des Anbieters als auch die Rolle des Kunden von einem Unternehmen eingenommen wird, wird der Gesch, ftsverkehr als *Business-to-Business* (B2B) bezeichnet, wird die Rolle des Kunden dagegen von einem Endverbraucher eingenommen, so wird der Gesch, ftsverkehr als *Business-to-Consumer* (B2C) bezeichnet. Anstelle eines Unternehmens als Anbieter oder als Kunde kann auch eine Beh, rde treten. Dann wird der Gesch, ftsverkehr auch als *Business-to-Administration* (B2A), *Administration-to-Administration* (A2A) oder *Administration-to-Consumer* (A2C) bezeichnet /MeTu99/. Der Gesch, ftsverkehr, der nur intern innerhalb eines Unternehmens zwischen den Mitarbeitern (Gesch, ftsf, hrung, Management, etc.) ñ d.h. ohne externen Partner ñ abl, uft, wird als *Business-to-Employee* (B2E) bezeichnet /LiZi99/. Diese Bezeichnungen des Gesch, ftsverkehrs sind zwar erst mit den Begriffen Electronic Commerce (EC) und Electronic Business (EB) aufgekommen, k, nnen zun, chst jedoch wertfrei auch f, r den konventionellen Gesch, ftsverkehr verwendet werden. Aus der Sicht eines Anbieters von Produkten bzw. Dienstleistungen umfasst Electronic Commerce die Verkaufsf, rderung, Verkaufsdurchf, hrung, Distribution und Verkaufsnachbereitung. Electronic Commerce dient prim, rder Unterst, tzung des *Business-to-Consumer* (B2C) oder *Administration-to-Consumer* (A2C) Gesch, ftsverkehrs. Electronic Business dagegen orientiert sich nicht am Endkunden, sondern dient der Unterst, tzung des Gesch, ftsverkehrs zwischen Unternehmen, Beh, rden oder auch zwischen Unternehmen und Mitarbeitern /Ge99/.



Abbildung 17: Arten des elektronischen Gesch, ftsverkehrs

Der Handel mit wirtschaftlichen G, tern findet auf M, rkten statt und der Gesch, ftsverkehr zwischen den einzelnen Partnern dient dazu, diesen Handel zu f, rdern, zu vereinbaren, abzuwickeln und nachzubereiten. Ein Markt besteht aus allen potentiellen Anbietern und Kunden mit einem bestimmten Bed, rfnis oder Wunsch, die willens

und fr̄ig sind, durch einen Tausch das Bed, rfnis oder den Wunsch nach wirtschaftlichen G, tern zu befriedigen /KoBl95/. Durch das Zusammenspiel von Angebot und Nachfrage kommt es auf dem Markt zur Preisbildung. *Der Marktpreis ist dann das Ergebnis des Zusammentreffens von Angebot und Nachfrage* /ChSc77/. War fr̄her der Markt ein realer Ort des Handels an dem Anbieter und Kunden zusammenkamen, um ihre G, ter zu tauschen, - *Jedes Zusammentreffen von Angebot und Nachfrage nennt man Markt, gleichg, ltig, an welchem Ort, zu welcher Zeit und unter welchen Umst%nden das geschieht* /ChSc77/ - so ist heute der Markt ein virtueller Gesamtmarkt, auf dem alle wirtschaftlichen G, ter gehandelt werden. Dieser Gesamtmarkt teilt sich in verschiedene Partialm%kte auf. Partialm%kte unterscheiden sich in Produktm%kte, in regionale, r%umliche und geographische M%kte, in zeitpunktbezogene M%kte und demographische M%kte auf denen unterschiedliche wirtschaftliche G, ter gehandelt werden. Der elektronische Gesch%ftsverkehr begleitet den Handel auf elektronischen M%kten, die sich in der heutigen Zeit nicht ñ oder nur unwesentlich ñ von dem virtuellen Gesamtmarkt unterscheidet bzw. von den einzelnen Partialm%kten unterscheiden.

Der Gesch%ftsverkehr begleitet den Handelsprozess auf M%kten mit wirtschaftlichen G, tern. Der Handelsprozess ñ auch als Verkaufsprozess oder Markttransaktion bezeichnet ñ umfasst die Phasen F̄rderung, Vereinbarung, Abwicklung und Nachbereitung. /Al98/, /Kr98/, /LiHa97/, /St97/. *Eine Markttransaktion entspricht einer Menge von Interaktionsprozessen zwischen Marktteilnehmern in unterschiedlichen Rollen mit dem Ziel, eine vertragliche Vereinbarung des Austausches von G, tern anzubahnen, zu vereinbaren und abzuwickeln* /ScLi98/. Die Interaktionsprozesse lassen sich in Klassen zusammenfassen und bilden dann die Phasen einer Markttransaktion. Findet ein Handelsprozess ausschließlich mit der Zielsetzung des Verkaufs von wirtschaftlichen G, tern statt, so werden die Phasen als Verkaufsf̄rderung, Verkaufsvereinbarung, Verkaufsabwicklung und Verkaufsnachbereitung bezeichnet. Die T%tigkeiten w%hrend der F̄rderung dienen dazu, das Interesse eines Kunden an Waren oder Dienstleistungen zu wecken. W%hrend der Phase Vereinbarung werden T%tigkeiten zur Angebotserstellung oder Ausschreibungsvorbereitung zur Bestellung durchgef, hrt. Die Phase Abwicklung umfasst die T%tigkeiten zur Bereithaltung, Bereitstellung und Auslieferung von Waren und Dienstleistungen und zur Rechnungsstellung und Zahlung. In der Phase Nachbereitung werden T%tigkeiten zur Kontaktpflege, Kundendienst und Reklamationsbearbeitung durchgef, hrt.



Abbildung 18: Marktphasen /St98/



Abbildung 19: Markttransaktionen /St98/

Das Ziel der Verkaufsf̄rderung seitens des Anbieters ist es, sich am Markt mit seinen Waren und Dienstleistungen zu positionieren und zu etablieren und seitens des Kunden, sich einen Markt, ̄berblick zu verschaffen. Dieser Markt, ̄berblick geht , ̄ber die eigentlichen Waren und Dienstleistungen hinaus. Es ist auch ein Ziel, sich Informationen , ̄ber den Partner zu verschaffen (z.B. Solvenz, Lieferf%higkeit, Renommee, Marktakzeptanz). Die Tiefe und Breite dieser Informationsgewinnung h%ngt auch maßgeblich von der Art des Kunden ab. Ein Endverbraucher als Kunde ben̄tigt andere Informationen , ̄ber den Anbieter als eine Beh̄rde oder Unternehmen als Kunde. So kann es z.B. f, r eine Beh̄rde von Bedeutung sein, ob der Anbieter selbst oder seine Waren und Dienstleistungen zertifiziert sind, f, r den Endverbraucher ist die Solvenz des Anbieters, solange es sich nicht um

hochwertige Waren und Dienstleistungen handelt, nicht von entscheidendem Interesse. Der *Pre-Sales-Support* spielt besonders bei Waren und Dienstleistungen hoher Komplexität eine wichtige Rolle. Zum Verkauf von Werkzeugmaschinen reichen einfache Produktbeschreibungen nicht aus. Ausführliche Beratung durch Fachleute über Erweiterbarkeit, Konfigurierbarkeit des Produkts zusammen mit einer Vorführung des Produkts ist sinnvoll. Auch hier unterscheidet sich der Endverbraucher als Kunde von einem Unternehmen als Kunde. Das Offerieren von alternativen gleichwertigen Waren oder Dienstleistungen durch den Anbieter kann beim Kunden trotzdem zu einer Kaufentscheidung führen; z.B. der Endverbraucher kann durch den Anbieter bei der Auswahl von aktiven und passiven Halbleiterelementen (Transistoren, Thyristoren, Kondensatoren, Widerstände, Dioden) mit gleicher technischer Signatur aktiv beraten werden.

In der Verkaufsvereinbarung werden durch den Anbieter Angebote für Waren oder Dienstleistungen abgegeben und der Kunde erteilt eine Bestellung an den Anbieter, nachdem er die Angebote eingeholt und verglichen hat. Es werden die Konditionen zwischen Anbieter und Kunde ausgehandelt, bevor ein Kaufvertrag geschlossen wird. Hier spielen mögliche Rabattsätze, Steuern, Zahlungsverfahren und -fristen eine Rolle. Tritt eine Behörde als Kunde in Erscheinung, so erfolgt in der Regel die Angebotseinholung durch eine Ausschreibung die, je nach Art des Produkts oder Dienstleistung, bestimmten formalen rechtlichen Kriterien genügen muss (z.B. mindestens drei unterschiedliche Anbieter aus verschiedenen Regionen oder nur zertifizierte Anbieter). Während bei einer Behörde als Kunde die Auswahl eines Anbieters auf der Basis des günstigsten Angebots erfolgt, können bei einem Unternehmen als Kunde auch noch andere Merkmale eine Entscheidung beeinflussen (Art und Dauer der Kundenbeziehung). Im Falle von hochwertigen Waren oder Dienstleistungen erfolgt durch den Anbieter, wenn ein Endverbraucher als Kunde auftritt, die Sicherstellung der Solvenz durch eine Anfrage an die Schufa.

Die Durchführung des Kaufvertrags findet in der Verkaufsabwicklung statt. Neben der Methode der Bezahlung (Nachnahme, Kreditkarte, Bargeld, etc) müssen sich die Partner für noch über das Versandverfahren einigen.

Um eine weitere Kundenbindung zu einem Unternehmen zu erreichen ist hierbei die Rolle des Kunden (Endverbraucher, Unternehmen, ...) unerheblich und wird in der Verkaufsnachbereitung der Kunde weiterbetreut. Dies erfolgt entweder durch Möglichkeit den Status von Lieferungen und Rechnungen zu erfahren oder Verbindungen mit einem Kundenberater aufzunehmen. Darüber hinaus können unter Verwendung des Wissens über die bisher bestellten Waren oder Dienstleistungen ergänzende Waren oder Dienstleistungen angeboten werden. Aber auch die kompetente Bearbeitung von Reklamationen und Retouren ist Bestandteil der Verkaufsnachbereitung.

Innerhalb der Marktphasen findet zwischen den beteiligten Partnern einer Markttransaktion ein Geschäftsverkehr statt. Dieser Geschäftsverkehr lässt sich unter der Betrachtung der beteiligten Rollen klassifizieren. Die Rollen Anbieter und Kunde/Nachfrager einer Leistung können von Unternehmen, Endverbrauchern, Behörden oder von Mitarbeitern in einem Unternehmen eingenommen werden. Die verschiedenen Beziehungen zwischen diesen Rollen werden in der folgenden Tabelle (in Anlehnung an /HeSa99a/) dargestellt.

		Nachfrager der Leistung		
		Consumer	Business	Administration
Anbieter der Leistung	Consumer	Consumer-to-Consumer z.B. Interner Kleinanzeigenmarkt	Consumer-to-Business z.B. Jobbörsen mit Anzeigen von Arbeitssuchenden	Consumer-to-Administration z.B. Steuerabwicklung von Privatpersonen
	Business	Business-to-Consumer z.B. Bestellung eines Kunden in einer Shopping Mall	Business-to-Business z.B. Bestellung eines Unternehmens bei einem Zulieferer per EDI	Business-to-Administration z.B. Steuerabwicklung von Unternehmen
	Administration	Administration-to-Consumer z.B. Abwicklung von Unterstützungsleistungen (SocialNet)	Administration-to-Business z.B. Rechtschutzmaßnahmen öffentlicher Institutionen im Internet	Administration-to-Administration z.B. Transaktionen zwischen öffentlichen Institutionen im In- und Ausland

Abbildung 20: Klassifikation von Markttransaktionen /HeSa99/

Bei Electronic Commerce und Electronic Business verkehren Anbieter und Kunden auf elektronischem Weg über ein Datenkommunikationsnetz miteinander. Das Internet stellt mit seinen Diensten (TCP/IP, FTP, NNTP, SMTP, http, etc.) ein solches Datenkommunikationsnetz dar. Das Internet ist eine Sammlung von individuellen Netzwerken, welche alle einzeln von verschiedenen Organisationen betrieben und bezahlt werden. Zur Weiterleitung von Informationen arbeiten diese Netzwerke zusammen. Die Menge aller Netzwerke bilden gemeinsam das Internet. Jedes Netzwerk verfügt über einen ausgezeichneten Rechner, der die Schnittstelle zu den anderen Netzwerken darstellt, d.h. jedes Netzwerk stellt einen Knoten in dem Internetnetzwerk dar. Hierdurch kann man das Internet auch als Menge von untereinander vernetzten Knotenrechnern bezeichnen. Dieses Datenkommunikationsnetz hat eine dezentrale Struktur und ist weltumspannend. Aufgrund der unterschiedlichen Kommunikationsinfrastruktur der einzelnen Netzwerke ist jedoch nicht immer gewährleistet, dass das Internet immer universal verfügbar ist. Dies liegt in der Tatsache begründet, dass regionale Netzwerke von verschiedenen Unternehmen, Universitäten und Forschungseinrichtungen oder großen und kleinen Internet-Service-Providern (ISP) betrieben werden, die dabei unterschiedliche Medien verwenden, wie z.B. analoge und digitale Telefonleitungen (ISDN), Kupferleitungen oder Glasfaserleitungen (Ethernet, Fast Ethernet, ATM) oder Mobilfunkverbindungen (DECT, GSM, UMTS). Die Basis für die Weiterleitung von Informationen zwischen den Netzwerken ist von einem Absender bis zum Empfänger liefern standardisierte Übertragungsprotokolle (TCP, IP, UDP). Die Entstehung des Internets beruht auf einer Entwicklung des amerikanischen Verteidigungsministeriums (*Department of Defense, DoD*) in den 60er Jahren. Durch das zunächst militärisch genutzte ARPANet (*Advanced Research Projects Agency*) wurde ein ausfallsicheres Netzwerk für den militärischen Bereich aufgebaut, durch das die Rechnerressourcen einiger Großforschungsanlagen besser ausgenutzt werden sollten. Nachdem die Beschränkung auf die ausschließlich militärische Nutzung Anfang der 70er Jahre aufgehoben wurde, entwickelte sich das ARPANet zum ersten herstellerunabhängigen Netzwerk mit verschiedenen Knotenrechnern. Parallel zum ARPANet entwickelten sich andere kleinere und größere Netzwerke, so dass jetzt Verbindungen zwischen den verschiedenen Netzwerken zunächst inneramerikanisch dann auch weltweit aufgebaut wurden. In den 80er Jahren wurde das so entstandene Internet hauptsächlich als Forschungsnetz durch Universitäten und Forschungseinrichtungen verwendet. Die kommerzielle und private Nutzung des Internet begann Anfang bis Mitte der 90er Jahre. Während das Internet zunächst nur die Dienste

- Elektronische Post (e-Mail),
- Dateitransfer,
- Nachrichten und Diskussionsforen (News),
- direkte Kommunikation (Talk, Chat),
- Zugriffe auf Softwaresysteme anderer Rechner (*Telnet, Remote-Login/-Exec*)

für die Informationsübertragung und -gewinnung angeboten, kam später der Dienst

- World Wide Web (WWW)

dazu. Jeder Dienst wird durch ein Protokoll unterstützt, das die Kommunikation zwischen verteilten Rechnern ermöglicht. Während in der Anfangszeit des Internet überwiegend Nachrichten in Form von einfachen Texten ausgetauscht wurden, die entweder direkt als e-Mails oder indirekt als News verschickt wurden, werden jetzt über das World Wide Web Nachrichten mit komplexen Inhalten ausgetauscht. Da diese Inhalte sowohl formatierte/attributionierte Texte, aber auch Grafiken, Fotos, Videos oder Sprache/Musik sein können, bezeichnet man diese Nachrichten auch als Dokumente oder Multimediadokumente, wenn sie Kombinationen davon als Inhalte enthalten. Der Austausch von Nachrichten mit einfachen Texten per e-Mail wird durch das SMTP-Protokoll (*Simple Mail Transfer Protocol*) oder für News durch das NNTP-Protokoll (*Network News Transfer Protocol*) definiert, während u.a. das HTTP-Protokoll (*Hypertext Transfer Protocol*) den Austausch von Nachrichten mit komplexen Inhalten definiert. Durch FTP (*File Transfer Protocol*) wird der Dateitransfer zwischen verschiedenen heterogenen Rechnern ermöglicht.

Es werden die unterschiedlichsten Softwaresysteme eingesetzt, um den Geschäftsverkehr bei Electronic Commerce und Electronic Business zu unterstützen. Diese Softwaresysteme können individuell, einfach oder komplex sein und den Geschäftsverkehr vollständig oder teilweise unterstützen. So werden z.B. Shopsysteme mit oder ohne Integration eines Warenwirtschaftssystems unter Verwendung einer Web-Client/Server-Technik eingesetzt. Zur Unterstützung des Bestellwesens und der Lagerhaltung innerhalb des Electronic Business werden Systeme zum Austausch von Informationen mittels der Protokolle XML oder EDIFACT verwendet. Es interagieren zur Unterstützung des Geschäftsverkehrs immer mindestens zwei Systeme miteinander, die nicht notwendigerweise gleich sein müssen, z.B. ein Warenwirtschaftssystem mit einem Shopsystem und einem Webserver-/Clientensystem oder eine SAP R3-Procurementkomponente mit einem Warenwirtschaftssystem.

5 Verteilte Systeme und Verteiltes Prozessmanagement

Die Forschung im Bereich der verteilten Systeme befasst sich mit den Konzepten, Vorteilen und Problemen, die die Interaktionen und Abhängigkeiten von Prozessen an verschiedenen Orten mit sich bringen. Als Beispiel sind hier die Kommunikation von Rechnern (Dienste, Protokolle, Sicherheit, Kapazitäten) aber auch die Kommunikation von abhängigen Systemen (Client/Server-Architekturen, verteilte Datenbanken, verteilte Algorithmen, gemeinsame Aktionen verteilter Prozesse) zu nennen. Auf der Hardwareebene werden verteilte Systeme z.B. als *„... eine Sammlung voneinander unabhängiger Computer, die dem Benutzer des Systems den Eindruck vermitteln, es handle sich um einen einzigen Computer“* (Ta95) beschrieben. Auf der Softwareebene wird von verteilter Anwendung gesprochen, die in (NiCh97) als *„a system of several independent software components cooperating towards a common purpose, where the components are not all colocated at a single site“* definiert wird. Die Untersuchungen bzw. Forschungen im Bereich der verteilten Systeme sind durch zwei grundlegend verschiedene Ansatzpunkte motiviert. Der erste Ansatzpunkt ist das Erstellen eines verteilten Systems um z.B. Kosten zu reduzieren, die Verfügbarkeit des Systems zu erhöhen oder die Leistung zu steigern. Der zweite Ansatzpunkt geht von einem gegebenen verteilten System aus und versucht dieses zu analysieren und zu verbessern. Diese Verbesserung kann unterschiedliche Aspekte betreffen (Kr94):

- sichere und effiziente Kommunikation (Nachrichtenprozesse),
- gute Kooperation der einzelnen Systeme bzw. der verteilten Prozesse,
- gute Kooperation zwischen geteilten Objekten und verteilten Prozessen,
- gute Kooperation zwischen Datenbank und verteilten Prozessen,
- gute Kooperation zwischen Clientprozessen und Serverprozessen.

Hierzu kommt noch der Aspekt der Verteilung für kooperierende Geschäftsprozesse (GrGr96). Dazu werden Geschäftsprozesse von einzelnen Unternehmungen oder Organisationseinheiten (die mit einander kooperieren)

als Teilprozesse eines ,bergreifenden Gesch%ftsprozesses angesehen. Es wird hier ein zentrales Prozessmanagement angenommen. Diese Teilprozesse liegen durch die Zuordnung von Attributen an verschiedenen Orten und kommunizieren miteinander. Hierbei sind Aspekte von verteilten Systemen (z.B. Effizienz und Sicherheit der Kommunikation, Effizienz und Flexibilität der Verteilung, Ort und Art der Datenspeicherung) auf die lokale Verteilung von Teilprozessen anwendbar.

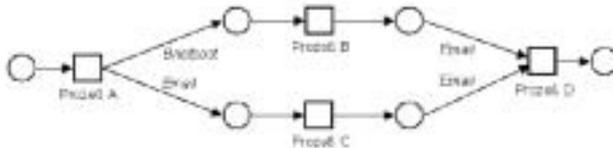


Abbildung 21: Kommunikation in einem verteilten System

Die Abbildung zeigt eine Kommunikation zwischen verteilten Teilprozessen. Der Teilprozess A gibt seinen Daten an Teilprozess B und C weiter. Dabei werden die Daten zu Teilprozess B auf einen Datenträger gespeichert und per Briefpost verschickt. Zu Teilprozess C werden die Daten per Email ,bertragen. Die beiden Prozesse B und C benötigen die gleiche Zeit zur Weiterverarbeitung der Daten und senden diese danach sofort per Email an Teilprozess D weiter, der dann diese Daten weiterverarbeitet. Mit Hilfe einer Analyse der Kommunikation würde die Kommunikation zwischen Teilprozess A und Teilprozess B als ineffizient identifiziert werden. Die Umstellung der Kommunikation zwischen Teilprozess A und Teilprozess B auf Email würde den gesamten Ablauf des Beispiel beschleunigen, so dass Teilprozess B und Teilprozess C dann wirklich parallel, d.h. zur gleichen Zeit an verschiedenen Orten, ausgeführt werden können. Eine solche Analyse kann mit Hilfe einer Simulation durchgeführt werden.

Gegenstand der Betrachtung im Kontext der verteilten Systeme sind nicht die technischen verteilten Systeme bzw. die technischen Prozesse, sondern die Verteilung von Geschäftsprozessen insbesondere die Softwareentwicklungsprozesse, die eine Klasse von Geschäftsprozessen bilden. Das Management von verteilte Softwareentwicklungsprozessen kann entweder zentral oder dezentral erfolgen. In beiden Fällen kommunizieren Teilprozesse ,ber Kommunikationskanäle miteinander, die an autonomen Organisationseinheiten, durchgeführt werden. Bei einer verteilten dezentralen Softwareentwicklung ist die Art und Weise der Kommunikation der Teilprozesse miteinander, um Daten, Informationen auszutauschen, ein Faktor der zum Erfolg eines Softwareprojekts beitragen kann.⁷

5.1 Zentrales Prozessmanagement

Beim zentralen Prozessmanagement werden die Prozesse, die in den beteiligten Organisationseinheiten verteilt ablaufen, durch ein ,r alle Kooperationspartner ,ltiges Prozessmodell beschrieben. Es existiert eine zentrale Einrichtung, die die Modellierung und die Ausführung der Prozesse gem%ß dem ,bergeordneten Prozessmodell kontrolliert und steuert. Dieses zentrale Management bedingt folgende Aspekte /Li99/:

- Sichtbarkeit der Prozesse
Sind an einem Prozess mehrere Organisationseinheiten beteiligt, so muss sichergestellt werden, dass jeder Kooperationspartner seine Dienste, Dokumente und Informationen anbieten kann, ohne dass sein Wissen auf die anderen Prozessbeteiligten ,bergeht, die dann wiederum sich dieses Wissen zu eigen machen und ,r sich nutzen können, d.h. lokale Teilprozesse sollen jedoch nicht zwangsl%ufig offengelegt werden müssen /De97/.
- Geographische Verteilung der Kooperationspartner
Es muss sichergestellt werden, dass bei dem Einsatz eines Softwaresystems zur Unterst,tzung des zentralen Prozessmanagements dessen Verf,gbarkeit immer gew%hrleistet ist, denn sonst ist der gesamte Prozess behindert.

- Komplexität von Prozessen

Die Anzahl und die Komplexität der zu verwaltenden Prozesse bei allen Kooperationspartnern kann groß werden, so dass ein Softwaresystem die Möglichkeit einer Skalierung auf verschiedenen technischen Ebenen bieten muss.

5.2 Dezentrales Prozessmanagement

Der Begriff der Autonomie spielt beim dezentralen Prozessmanagement eine wichtige Rolle. Eine Organisation ist autonom, wenn sie selbstständig und unabhängig die Kontrolle über ihre Teilprozesse ausüben kann. Es lassen folgende Bereiche der Autonomie unterscheiden (ShLa90):

- **Strukturautonomie**
Eine Organisationseinheit kann innerhalb ihrer Organisation, ihre Struktur, d.h. ihre Stellen, die Zuordnung von Rollen und Personen zu Stellen und die Beziehungen zwischen Stellen frei wählen
- **Modellierungsautonomie**
Die Modellierung von Teilprozessen erfolgt unter Berücksichtigung der Belange und Erfordernissen der lokalen Organisationseinheit.
- **Ausführungsautonomie**
Die Reihenfolge der Ausführung von Aktivitäten innerhalb von Teilprozessen wird durch die Organisationseinheit festgelegt.
- **Kommunikationsautonomie**
Die Art und Weise der Aufnahme einer Kommunikationsverbindung zu anderen Organisationseinheiten, über die Informationen zur Verfügung gestellt oder ausgetauscht werden, wird durch die Organisationseinheit festgelegt.
- **Partizipationsautonomie**
Die Bereitstellung von Diensten und deren Ausführung von und zu anderen Organisationseinheiten über eine Kommunikationsverbindung wird von der lokalen Organisationseinheit entschieden.

Die Organisationseinheiten sind im Vergleich zu anderen Organisationseinheiten autonom. Bei jeder Organisationseinheit werden jedoch oft mehrere Teilprozesse durchgeführt. In diesem Fall muss der Aspekt auch auf die Teilprozesse und deren Schnittstellen erweitert werden. Es kann nicht nur die einzelne Organisationseinheit betrachtet werden. Durch eine Kommunikationsverbindung werden Teilprozesse verschiedener Organisationseinheiten unterschiedlich miteinander gekoppelt (Böde96). Hierbei wird die einfache und die mehrfache Kopplung unterschieden. Durch eine einfache Kopplung wird genau ein Teilprozess einer Organisationseinheit mit genau einem anderen Teilprozess einer anderen Organisationseinheit verbunden (one-to-one), während bei der einer mehrfachen Kopplung ein Teilprozess mit mehreren anderen Teilprozessen verbunden wird. Hierbei wird unterschieden, ob ein Teilprozess Daten für mehrere andere Teilprozesse zur Verfügung stellt (one-to-many) oder mehrere Teilprozesse Daten für genau einen anderen Teilprozess zur Verfügung stellt (many-to-one) (GrGr96).

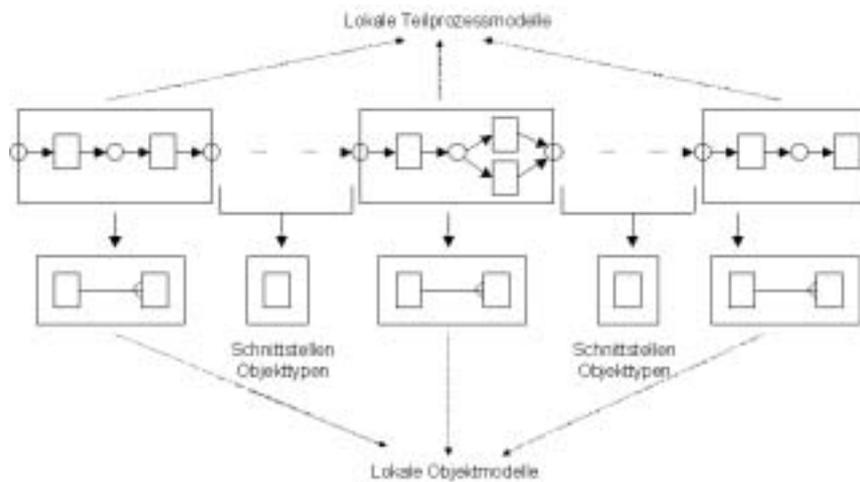


Abbildung 22: Schema verteilter Teilprozesse

Auch für diese Teilprozesse und dessen Schnittstellen gelten Besonderheiten bzgl. ihrer Autonomie. Es können drei Autonomieebenen und fünf verschiedene Autonomiegrade unterschieden werden (GrWe00).

- Eine Aktivität an der Schnittstelle ist datenautonom, wenn die Daten anderen Teilprozessen nur in Kopie zur Verfügung gestellt wird. Mit verschiedenen Autonomiegraden kann festgelegt werden, ob der initierende Teilprozess die Datenhoheit hat und wie die Daten von den empfangenden Teilprozessen bearbeitet werden können.
- Eine Aktivität an der Schnittstelle ist operationsautonom, wenn sie sich wie eine „Black-Box“ verhält und keine Internen über die Art und Weise der erzeugten Daten den anderen Teilprozessen zur Verfügung stellt.
- Eine Aktivität an der Schnittstelle ist kommunikationsautonom, wenn die Entscheidung wie eine Kommunikation zu anderen Teilprozessen aufgebaut wird (verschlüsselt oder nicht, synchron oder asynchron, etc.) ohne Berücksichtigung anderer Teilprozesse erfolgt.

Diese Autonomiearten können ebenso wie Kommunikationsmodi nur in sinnvoller Kombination stattfinden. Durch Autonomiegrade wird die Qualität der Autonomie einer Aktivität eines Teilprozesses oder mehrerer Teilprozesse beschrieben. Die Autonomiegrade werden in fünf verschiedene Klassen unterteilt (GrWe00).

Grad der Autonomie	Aktivitäten desselben Grades sind
0	nicht autonom
1	schwach autonom
2	halb autonom
3	stark autonom
4	vollständig autonom

Tabelle 12: Autonomiegrade

Für jedes Softwareprojekt werden den einzelnen Autonomiegraden individuell Prozentzahlen zugewiesen, die kennzeichnen wie autonom die Teilprozesse der einzelnen Organisationseinheiten sind. So bedeutet z.B. nicht autonom, dass weniger als 10% aller Aktivitäten, die die Schnittstellen von Teilprozessen darstellen, autonom sind, schwach autonom, wenn weniger als 45% aller Aktivitäten das Kriterium erfüllen, halb autonom, wenn weniger als 55%, stark autonom, wenn weniger als 90% dieses Kriterium erfüllen. Wenn mehr als 90% aller Aktivitäten, die die Schnittstellen von Teilprozessen bilden, autonom sind, dann spricht man vollständig autonom. Der Grad der Autonomie ist abhängig von dem Verhältnis von nicht autonomen zu vollständig autonomen Aktivitäten einer Autonomieebene.

Art der Autonomie	Daten-autonom	Operations-autonom	Kommunikationsautonom
Granularität der Autonomie			
Aktivität	{0, 1, ..., 4}	{0, 1, ..., 4}	{0, 1, ..., 4}
Aktivität desselben Teilprozesses	{0, 1, ..., 4}	{0, 1, ..., 4}	{0, 1, ..., 4}, ..., {0, 1, ..., 4}
Aktivität auf der selben Abstraktionsebene	{0, 1, ..., 4}	{0, 1, ..., 4}	{0, 1, ..., 4}
Aktivität zwischen zwei Teilprozessen	{0, 1, ..., 4}	{0, 1, ..., 4}	{0, 1, ..., 4}

Abbildung 23: Klassifikation von Autonomierarten, -granularität und -ebenen

In Erweiterung der Datenautonomie kann unterschieden werden, ob Teilprozesse Daten entweder zur Verarbeitung zur Verfügung stellen oder für eine Verarbeitung benötigen (Push/Pull-Kommunikation). In Abhängigkeit dieser Kommunikation kann unterschieden werden, wo, d.h. bei welchem Teilprozess die Daten persistent oder transient gehalten werden. Die Daten, die übertragen werden, werden durch die Schnittstellen der Teilprozesse zwischengespeichert oder auch nicht. Es werden in diesem Zusammenhang sechs Kommunikationsmodi unterschieden (GrWe99). Diese Kommunikationsmodi unterscheiden sich hinsichtlich des Persistenzverhalten von Daten bei den einzelnen miteinander kommunizierenden Teilprozessen, wobei Konsistenzbedingungen beachtet und eingehalten werden müssen, z.B. wenn ein Teilprozess eine Push-Verbindung aufbauen will, dann darf der andere Teilprozess nur eine Pull-Verbindung zulassen und nicht ebenfalls eine Push-Verbindung.

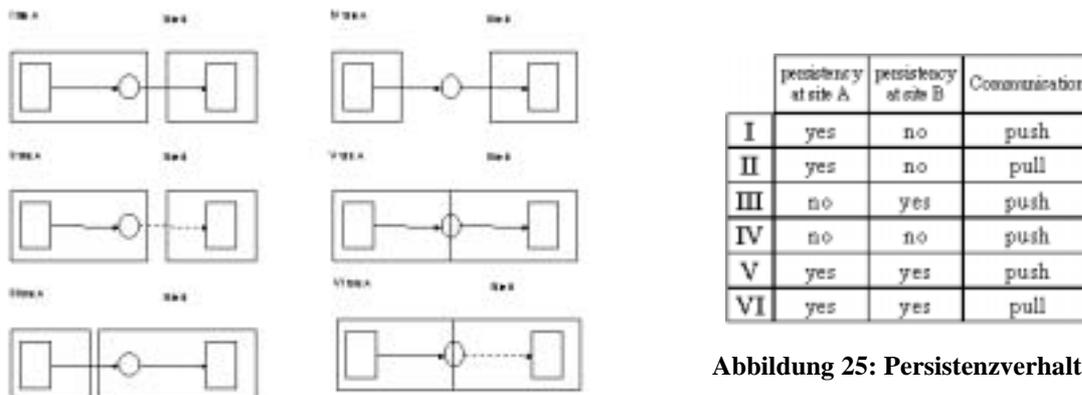


Abbildung 25: Persistenzverhalten

Abbildung 24: Push/Pull-Kommunikation

Beim dezentralen Prozessmanagement werden die Prozesse der Kooperationspartner so aufgeteilt, dass genau eine Organisationseinheit für die Ausführung von Teilprozessen zuständig und verantwortlich ist. Diese Teilprozesse kommunizieren mit anderen Teilprozessen anderer Organisationseinheiten, indem sie Nachrichten, Dokumente und Dienste austauschen. Dies bedingt, dass die verschiedenen Organisationseinheiten über eigene Softwaresysteme zur Unterstützung von Teilprozessen verfügen können. Die Kopplung von Teilprozessen bietet den Vorteil, dass ein Gesamtprozess durch eine Menge von schon existierenden Teilprozessen gebildet werden kann und das eine Kenntnis der internen Struktur der Teilprozesse nicht erforderlich ist. Beim dezentralen Prozessmanagement existiert ein Netz von Softwaresystemen, die miteinander verknüpft sind. Der Gesamtprozess wird durch die Kommunikation der Teilprozesse bestimmt und gesteuert. Da eine Homogenität der Softwaresysteme nicht immer sichergestellt werden kann, liegt hier ein wesentlicher Schwerpunkt auf der Art und Weise der Kommunikation der Systeme untereinander. Wie Workflowmanagementsysteme miteinander Informationen und Daten austauschen können, wird durch die Schnittstellendefinitionen der WfMC beschrieben (WfMC96, WfMC98b). Für alle anderen Softwaresysteme ergibt sich die Notwendigkeit individuelle Kommunikationsprotokolle zu definieren oder auf bestehende Techniken (z.B. EDI, XML, ECML, ICE) zurückzugreifen.

Darüber hinaus kann die Rolle eines Teilprozesses in der Kopplung bestimmt werden. Bei einer vertikalen Kopplung zwischen Teilprozessen übernimmt ein Teilprozess die Rolle eines Initiators (Superprozess) während ein anderer Teilprozess die Rolle eines Agiators (Subprozess) übernimmt. Dabei erzeugt der Subprozess ein Ergebnis für den Superprozess. Da dieses Ergebnis nur in einem bestimmten Kontext erforderlich ist, wird das

Ergebnis des Subprozesses bei der abnormalen Beendigung des Superprozesses nicht mehr benötigt und daher verworfen resp. die Ausführung des Subprozesses angehalten und beendet. Bei einer horizontalen Kopplung werden unabhängige Teilprozesse verbunden, die sich nicht gegenseitig bedingen. Der Abbruch eines Teilprozesses wirkt sich nicht auf die anderen Teilprozesse aus. Bei einer starken Kopplung von Teilprozessen werden, neben den eigentlichen Daten auch Informationen über Ereignisse, die während der Ausführung von Teilprozessen stattfinden ausgetauscht (Start, Ende und Unterbrechung einer Ausführung, Grund einer Unterbrechung der Ausführung, ...). Der Teilprozess, der diese Informationen erhält, entscheidet dann über die weitere Prozessausführung. Bei schwach gekoppelten Teilprozessen findet dieser Informationsaustausch nicht statt. Die Weiterbearbeitung von Daten kann von der Organisationseinheit, die die Daten abschickt hat, nicht mehr kontrolliert werden. Sie erhält in der Regel auch keine Rückmeldung über den Status der Bearbeitung. Zusätzlich gibt es die Möglichkeit Teilprozesse isoliert oder nicht isoliert zu koppeln. Werden Teilprozesse isoliert gekoppelt, so erfolgt ein Austausch von Daten vor dem Start und nach Beendigung der Ausführung eines Teilprozesses. Bei einer nicht isolierten Kopplung kann der Austausch von Daten auch während der Ausführung eines Teilprozesses erfolgen. Alle Kopplungsarten können miteinander kombiniert werden, wobei jedoch nicht jede Kombination sinnvoll ist.

6 Beschreibungssprachen für Prozesse

Da die Entwicklung von komplexen Softwaresystemen oft von großen Teams durchgeführt werden müssen, bieten Softwareprozessmodellierungssprachen ebenfalls die Möglichkeiten, beteiligte Organisationseinheiten, die räumlich verteilt sind, mit modellieren zu können. Darüber hinaus spielt bei einer verteilten Entwicklung die Art und Weise der Kommunikation zwischen den Organisationseinheiten eine Rolle. Bei einer Analyse des Softwareentwicklungsprozesses sollen Schwachstellen, wie Verzerrungen durch Kommunikationswege selbst und Synchronisation von Ergebnissen, Überlast von Kommunikationsstrecken, Deadlocks durch Kommunikationsabhängigkeiten, erkannt werden können. Hierzu müssen die Modellierungssprachen ebenfalls die Möglichkeiten bieten, die Aspekte mit modellieren zu können.

Es gilt nun festzustellen, ob die gängigen graphischen Modellierungssprachen die Möglichkeiten bieten, diese Aspekte bei der Beschreibung und Modellierung von Softwareentwicklungsprozessen mit angeben zu können. Nur dann ist eine anschließende automatische Analyse durch ein unterstützendes Softwaresystem möglich.

Zur Beschreibung von Softwareentwicklungsprozessen können verschiedene Softwareprozessmodellierungssprachen verwendet werden. Diese Sprachen haben entweder eine formale oder informale bzw. semiformale Basis und verwenden eine graphische oder textuelle Notation. Regelbasierte Sprachen (z.B. Marvin, Merlin) haben eine formale Basis, verwenden jedoch keine graphische Notation. Datenflussdiagramme verwenden eine graphische Notation, verfügen jedoch nicht über eine formale Basis. Die ereignisgesteuerten Prozessketten (EPK) und deren Erweiterungen (eEPK) [ScN, 97] haben eine semiformale Basis und eine graphische Notation, während die Aktivitätsdiagramme aus UML (Unified Modeling Language) [BoJa99], bedingt durch die Tatsache, dass generell UML keine formale Basis hat, zwar eine graphische Notation, aber nur eine informale Basis haben. Eine formale Basis und eine graphische Notation haben Petrinetze und hier Petrinetze [Re82, Gr91].

Es ist aber nicht zwangsläufig so, dass zur Beschreibung von Softwareentwicklungsprozessen unbedingt eine Beschreibungssprache benötigt wird, die sowohl über eine formale Basis und eine graphische Notation verfügt. Hier müssen die Anforderungen abgewogen werden. Zur automatisch unterstützten Analyse eines Softwareentwicklungsprozesses ist eine formale Basis unbedingt erforderlich. Hier wäre dann eine graphische Notation wünschenswert, aber nicht unbedingt erforderlich. Zur reinen Dokumentation des Softwareentwicklungsprozesses ist eine graphische Notation mit semiformalen oder informalen Notationen ausreichend. Hierzu kommt noch der Umstand, dass Softwareentwicklungsprozesse nicht zu den stark strukturierten Prozessen gezählt werden, sondern eher den schwach strukturierten Prozessen zugerechnet werden kann, denn nicht während der Entwicklung eines Softwaresystems gibt es in Teilbereichen auch unstrukturierte Vorgehensweisen. Das bedeutet, dass die Soft-

wareprozessmodellierungssprache ebenfalls die Möglichkeit bieten muss, bestimmte Vorgehensweisen unstrukturiert beschreiben zu können, ohne ihr eigenes Paradigma zu verletzen.

6.1 Ereignisgesteuerte Prozessketten (EPK)

Eine ereignisgesteuerte Prozesskette (EPK) ist ein gerichteter Graph. EPKs werden als Variante eines Bedingungs-/Ereignisnetzes (B/E-Netz) verstanden. Die Knoten des gerichteten Graphs können unterschiedliche Semantik haben: Sie haben entweder den Typ Ereignis, Funktion oder binärer logischer Konnektor (or, and, xor). Die Kanten verbinden immer zwei Knoten unterschiedlichen Typs miteinander. Durch die logischen Konnektoren kann eine Kante verzweigen, wobei entweder immer Ereignisse oder immer Funktionen mit dem Konnektor verbunden werden resp. umgekehrt. Die Eingänge eines Konnektors sowie die Ausgänge eines Konnektors sind immer Knoten des gleichen Typs. Eine ereignisgesteuerte Prozesskette beginnt immer mit mindestens einem Start- und einem Zielereignis.

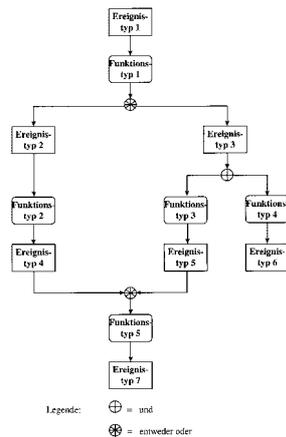


Abbildung 26: ereignisgesteuerte Prozessketten (EPK)

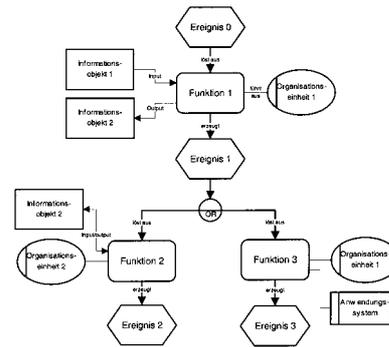


Abbildung 27: erweiterte EPK

Durch die zusätzliche Benennung von Kanten und die Einführung von Organisationseinheiten und Informationsobjekten werden die EPKs semantisch angereichert. Diese EPKs werden als erweiterte ereignisgesteuerte Prozessketten (eEPK) bezeichnet.

6.2 Unified Modeling Language (UML)

Die Unified Modeling Language dient zur Spezifikation von Softwaresystemen entsprechend eines Vorgehens-/Prozessmodells. Beginnend mit der Anforderungsanalyse bis zum System- und Integrationstest. Für jede Phase der Softwareentwicklung wird durch die UML eine Möglichkeit zur Spezifikation geboten. *The Unified Modeling Language (UML) provides system architects working on object analysis and design with one consistent language for specifying, visualizing, constructing, and documenting the artefacts of software systems, as well as for business modelling.* (OMG99). Zur Modellierung von Geschäftsprozessen werden in UML die Aktivitätsdiagramme angeboten. Ein Geschäftsprozess wird hier als eine Reihe von Aktivitäten betrachtet. Der Ablauf der Aktivitäten wird durch Entscheidungen und Synchronisation gesteuert. Ein Aktivitätsdiagramm ist ein gerichteter Graph. Die Knoten in diesem Graph können Aktivitäten, Entscheidungen oder Synchronisationen sein. An einem Synchronisationsknoten kann von einer Aktivität zu mehreren anderen Aktivitäten verzweigt werden, die dann zeitgleich ausgeführt werden oder jeweils den Beginn von mehreren zeitgleichen Teilgeschäftsprozessen sind. Diese Teilgeschäftsprozesse können wiederum wieder durch einen Synchronisationsknoten zusammengeführt werden. Ein bedingter Übergang zu Aktivitäten oder Entscheidungen kann durch Entscheidungsknoten ausgedrückt werden. Durch die vertikale Separierung des Aktivitätsdiagramms kann die Zuständigkeit von Organisationseinheiten modelliert werden.

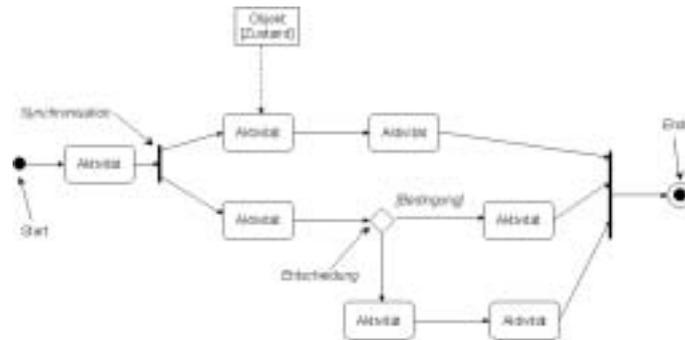


Abbildung 28: UML Aktivitätsdiagramm

6.3 Petrinetze und hier Petrinetze

Ein Petrinetz basiert auf einem gerichteten, bipartiten Graphen. In dem Graphen existieren zwei Typen von Knoten: Die eine Sorte Knoten wird Stellen (passive Komponenten) genannt und bildet einen Zwischenspeicher für Informationen (Marken). Die andere Sorte Knoten wird als Transition (aktive Komponenten) bezeichnet und verarbeitet die Informationen. Die Knoten (Stellen, Transitionen) sind durch gerichtete Kanten verbunden. Eine Kante verläuft immer von einer Sorte der Knoten zu einem Knoten der anderen Sorte. Die Verarbeitung erfolgt durch das Lesen und Entfernen der Informationen aus der Stelle vor einer Transition und der anschließenden Ablage in die Stelle nach einer Transition. Dabei kann eine Transition auch mehrere Ein- bzw. Ausgabestellen besitzen. Eine Transition schaltet, wenn alle Eingabestellen mit Informationen belegt sind. Sie gibt nach der Verarbeitung die Informationen an alle Ausgabestellen weiter. Ein wesentliches Merkmal von Petrinetzen ist ihre Lokalitätseigenschaft. Dies bedeutet, dass die Auswirkungen einer aktiven Komponente auf ihre direkte Umgebung beschränkt sind.

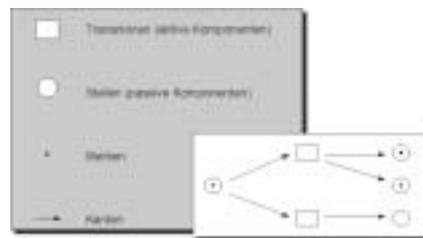


Abbildung 29: Bestandteile von Petrinetzen

Petrinetze und hier Petrinetze (z.B. Bedingungs/Ereignis-Netze, Stellen/Transitions-Netze, Prädikat/Transitions-Netze, FUNSOFT-Netze) gewinnen im Bereich der Prozessmodellierung immer mehr an Bedeutung. Petrinetze sind ein graphischer Formalismus zur Beschreibung und Analyse von Abläufen. Mit Petrinetzen können sequentielle, sich gegenseitig ausschließende, nebenläufige und nichtdeterministische Abläufe beschrieben werden

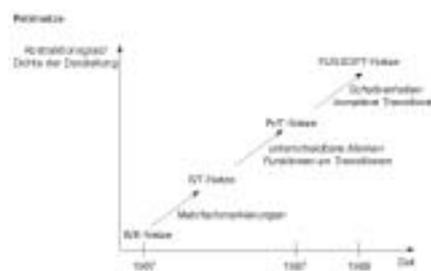


Abbildung 30: Entwicklungshistorie von hier Petrinetzen

6.3.1 B/E-Netze

Netze aus Bedingungen und Ereignissen (B/E-Netze) gehören in die Klasse der einfachen Petrinetze. Bedingungen werden durch passive Komponenten und Ereignisse durch aktive Komponenten dargestellt. Ist eine Bedingung erfüllt, so wird die passive Komponente mit einer sog. Markierung versehen, anderenfalls bleibt die passive Komponente leer. Ein Ereignis kann immer dann eintreten, wenn alle passiven Komponenten vor einer aktiven Komponente eine Markierung tragen, jedoch alle passiven Komponenten nach einer aktiven Komponente leer sind.

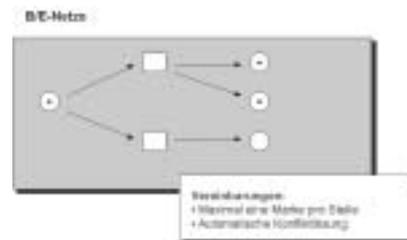


Abbildung 31: Merkmale von B/E-Netzen

6.3.2 S/T-Netze

Der Unterschied bei Stellen/Transitionsnetzen (S/T-Netze) besteht darin, dass aus Bedingungen Stellen gemacht werden, die mehr als eine Markierung haben können. Stellen können dann mit einer Kapazität versehen werden. Ferner können die Kanten mit einer natürlichen Zahl beschriftet werden. Die Zahl gibt an, wie viele Markierungen in Pfeilrichtung fließen, entweder von den Stellen vor einer Transition oder zu den Stellen nach einer Transition.

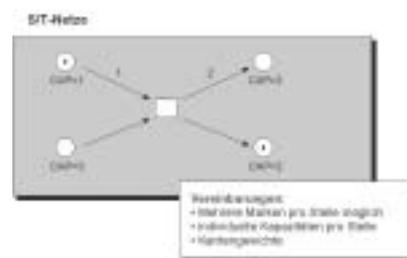


Abbildung 32: Merkmale von S/T-Netzen

6.3.3 Pr/T-Netze

Die Prädikat/Transitionsnetze sind eine Weiterentwicklung der S/T-Netze. Für die unterscheidbaren Marken lassen sich Bezeichner verwenden, die bestimmte Systemeinheiten spezifizieren. Es können mehrere Bezeichner in einem Prädikat vorkommen. Kanten lassen sich mit Variablen beschriften, die angeben in welcher Form die Bezeichner aus den Prädikaten durch die Ereignisse fließen. Ferner lassen sich Gruppierungen von Variablen vornehmen und darüber hinaus lassen sich Funktionsterme auf Variablen anwenden.

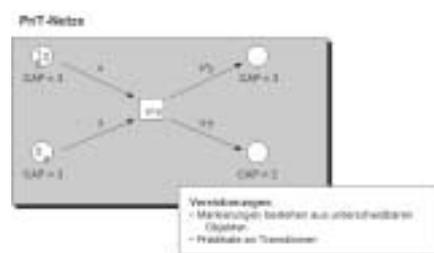


Abbildung 33: Merkmale von Pr/T-Netzen

6.3.4 Funsoft-Netze

Funsoft-Netze sind eine Variante der Petrinetze, in der die Semantik erweitert wurde. Sie besitzen eine eindeutige Semantik, die auf der Basis von Prädikat/Transitions-Netzen (Pr/T-Netze) formal definiert ist. Durch diese Sprache werden spezielle Bedürfnisse der Prozessmodellierung berücksichtigt. Die wichtigsten Erweiterungen bzw. Spezialisierungen gegenüber den Petrinetzen sind:

- die Erweiterung des Schaltverhaltens der aktiven Komponenten (Transitionen),
- die Erweiterung der Attribute der Transitionen,
- die Typisierung der Objekte,
- die Bindung der passiven Komponenten (Stellen) an bestimmte Objekttypen.

Darüber hinaus können Funsoft-Netze hierarchisch strukturiert werden. So können komplexe Aktivitäten durch eine Transition repräsentiert werden. Dieser Repräsentant stellt wiederum ein Funsoft-Netz dar, das aus einer Menge von Stellen und Transitionen bestehen kann.

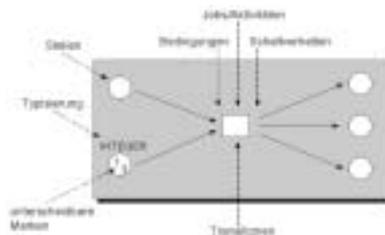


Abbildung 34: Merkmale von FUNSOFT-Netzen

7 Szenarien für konkrete Entwicklungsprozesse

Für das vorher beschriebene Anwendungsgebiet Electronic Commerce wird beispielhaft ein Szenario ausgewählt. Durch dieses Szenario ist die Entwicklung eines *Electronic Commerce Portals* beschrieben, wie ein typisches Softwaresystem aus diesem Anwendungsgebiet entwickelt wurde bzw. entwickelt werden kann.

Ein *Electronic-Commerce-Portal* dient, zumal wenn es für eine spezielle Benutzergruppe in einem Intranet eingesetzt wird, der Unterstützung des Geschäftsverkehrs im Sinne des *Business-to-Employee* (B2E). Hierbei wird der Geschäftsverkehr zwischen dem Management und den verschiedenen Mitarbeitern, aber auch zwischen den Mitarbeitern untereinander unterstützt. Diese Unterstützung des Geschäftsverkehrs erfordert den Einsatz von verschiedenen Softwaresystemen (Legacy-System, Shopsystem, Web-System, Internet-System, Office-System).

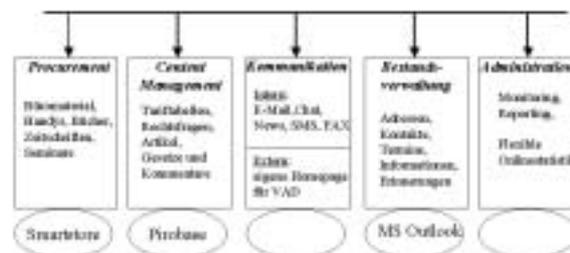


Abbildung 35: Subsysteme des IPSI-Portals

In diesem Sinne dient ein *Electronic-Commerce-Portal* als Integrationsplattform für die verschiedenen Softwaresysteme. Ziel ist es, den Geschäftsverkehr im Sinne des B2E optimal zu unterstützen. Durch dieses *Electronic-Commerce-Portal* (IPSI) /BoGr00a/, /BoGr00b/ sollen insbesondere die Außendienstmitarbeiter (VAD) von Versicherungsunternehmen bei ihrer täglichen Arbeit unterstützt werden, wobei der Einsatz des *Electronic-Commerce-Portals* in einem Intranet erfolgt. Die Versicherungsaussendienstmitarbeiter benötigen Zugriff auf

Informationen über die privaten oder gewerblichen Kunden, die sie zu betreuen haben. Sie müssen aber auch bei der Organisation ihrer Arbeit durch Terminplaner mit Erinnerungsfunktion, Adressverwaltung etc. unterstützt werden. Das Versicherungsunternehmen hat das Bestreben, seine Mitarbeiter mit den aktuellsten Informationen über das Produktportfolio, Tarifen sowie Gesetzestexten und -kommentaren zu versorgen. Ein *Electronic-Commerce-Portal* muss daher eine Vielzahl von Funktionen bereitstellen. Der Vorteil eines Portals liegt in der Integration von verschiedenen Softwaresystemen, die diese Funktionen bereitstellen, so dass der Nutzer eines Portals zielgerichtet an die gewünschten Informationen herangeführt wird. Er muss einerseits nicht umständlich suchen und wird andererseits nicht mit sinnlosen Informationen versorgt. Die objektorientierte Konzeption mittels UML /BoJa99/, die Realisierung von Adaptern zur Integration von Softwaresystemen in der Programmiersprache Java /GoJo96/ und die Verwendung einer *Middleware* (CORBA/RMI) zur Kommunikation innerhalb des Portals waren Gegenstand dieser Entwicklung.

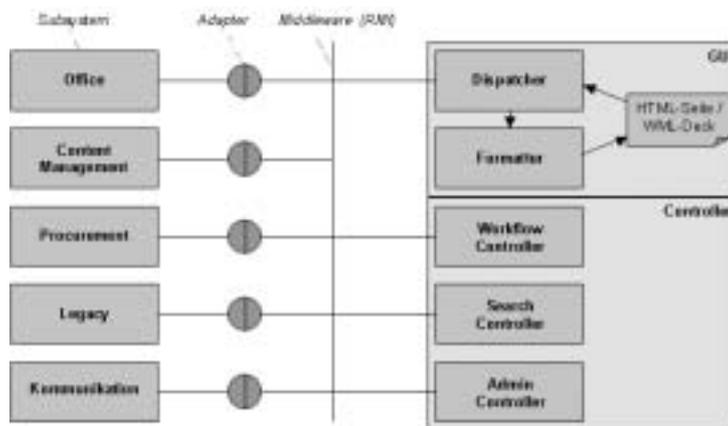


Abbildung 36: Architektur des IPSI-Portals

Die Funktionalität der Subsysteme *Office*, *Content Management*, *Procurement* und *Communication* wird durch bereits existierende, externe Systeme bereitgestellt, die über sogenannte Adapter in das *Electronic-Commerce-Portal* integriert werden. Durch die Kombination der Funktionalität verschiedener Subsysteme entstehen neue Funktionen des Gesamtsystems, die sich an den Geschäftsprozessen der Benutzer orientieren: Um z.B. einen wichtigen Vertragsstichtag aus dem Legacy-System (Partnerdatenbank) zu lesen, einen entsprechenden Eintrag in den Terminkalender des Office-Systems zu schreiben und das rechtzeitige Versenden einer Erinnerung über das Kommunikationssystem zu veranlassen, wäre bei getrennten Subsystemen eine Reihe von unterschiedlichen Tätigkeiten erforderlich. Im *Electronic-Commerce-Portal* kann der VAD diese Aufgabe hingegen in wenigen Schritten lösen.

Die dazu notwendige Verknüpfung von Tätigkeiten und Konsolidierung von Daten wird durch Controller geleistet, die als Gehirn des *Electronic-Commerce-Portals* betrachtet werden können: Sie erhalten vom Benutzer Aufträge zur Ausführung komplexer Geschäftsprozesse, steuern daraufhin die Subsysteme entsprechend an und liefern die konsolidierten Daten an den Anwender zurück.

Controller und Subsysteme kommunizieren durch den Austausch von Business-Objekten, d.h. Einheiten, die zentrale Bedeutung für den Workflow im *Electronic-Commerce-Portal* besitzen. Die folgenden Business-Objekte sind darum allen Controllern und Subsystemen bekannt:

- Benutzer
- Kontakt
- Termin
- Aufgabe
- Nachricht
- Shop-Artikel
- Bestellung
- Bestellhistorie
- Suchanfrage
- Suchergebnis

Um zum Beispiel einen Termin einzutragen, erzeugt der Workflow-Controller ein Termin-Objekt aus den vom Dispatcher empfangenen Daten und gibt es an eine Methode des Office-Subsystems weiter, die den Termin in den Kalender des Benutzers einträgt. Wenn der Benutzer angibt, dass er rechtzeitig per e-Mail an den Termin erinnert werden möchte, erzeugt der Workflow-Controller zusätzlich ein Nachricht-Objekt, verbindet es mit einer Kopie des Termin-Objekts und gibt es an das Kommunikationssystem weiter, wo es in die Warteschlange für e-Mail-Nachrichten eingereiht und zum gewählten Zeitpunkt versandt wird.

Die Subsysteme werden nicht direkt vom Benutzer angesprochen und treten nicht selbst mit ihm in Kontakt. Aus diesem Grund wurden Such- und Administrationssystem nicht als externe Systeme, sondern als Controller klassifiziert, da sie aktiv Informationen von den anderen Systemen anfordern oder ihnen übergeben. Alle weiteren Geschäftsprozesse werden vom Workflow-Controller realisiert und gesteuert. Die externen Systeme verfügen über ein oder mehrere Schnittstellen (APIs), durch welche die jeweilige Funktionalität genutzt werden kann. Jedes externe System wird daher über genau einen Adapter mit der zentralen *Middleware* verbunden. Jeder Adapter stellt der *Middleware* eine Menge von Methoden zur Verfügung, welche die ursprüngliche Schnittstelle des externen Systems kapseln. Der Benutzer interagiert hauptsächlich über einen Web-Browser mit dem *Electronic-Commerce-Portal*.

7.1.1 Entwicklungsprozess

Der gesamte Entwicklungsprozess umfasst verschiedene einfache und komplexe Aktivitäten.

Die Tätigkeit Anforderungsanalyse wird initiiert durch eine initiale Systemarchitektur und einer initialen Anforderungsliste (White Paper). Das Ziel der Anforderungsanalyse ist die Konkretisierung der initialen Anforderungen. Das Ergebnis der Anforderungsanalyse sind konsolidierte Anforderungen für das Portal (es wird beschrieben, was entwickelt werden soll). Aus den Anforderungen an das Portal werden Anwendungsfälle abgeleitet. Durch Anwendungsfälle wird das Verhalten des Portals beschrieben. Die Architektur des Portals wird durch die initiale Systemarchitektur grob vorgegeben. Bestandteile dieser Architektur sind Subsysteme. Die in der Anforderungsanalyse konsolidierten Anforderungen werden entsprechend den Subsystemen strukturiert.

Die Entwicklung eines Benutzungsoberflächenprototyps dient einerseits zu Marketingzwecken und andererseits zur Verifikation der Anforderungen.

Auf der Basis der Anforderungen für Subsysteme muss entschieden werden, ob existierende Softwaresysteme den Anforderungen entsprechen oder ob eine Eigenentwicklung erforderlich ist. Nach der Entscheidung existierende Softwaresysteme für Subsysteme einzusetzen wurden nach einer Marktanalyse, bei der nichtfunktionale Kriterien, wie z.B. Preis, Verfügbarkeit, Support oder Plattform, eine Rolle spielten, existierende Softwaresysteme, z.B. MS Outlook 2000, Smartstore und Pirobase ausgewählt. Für diese Softwaresysteme musste nun eruiert werden, ob diese Systeme über eine Programmierschnittstelle (API) verfügen oder ob eine Schnittstelle entwickelt werden kann. Dies erfolgt durch die Entwicklung von Prototypen auf der Basis der Anwendungsfälle und der ermittelten Anforderungen.

Die initiale Systemarchitektur wird im Grobentwurf derart konkretisiert, in dem festgelegt wird, welche Subsysteme durch eine Eigenentwicklung realisiert werden und für welche Subsysteme externe Softwaresysteme eingesetzt werden. Eine Entscheidungshilfe hierfür liefert die Prototypentwicklung.

Parallel zum Grobentwurf erfolgt der Entwurf der graphischen Benutzungsoberfläche (GUI) für das Portal. Wiederum werden die Anwendungsfälle verwendet. Dieser GUI-Entwurf wird beim Feinentwurf und der Realisierung verwendet.

Der Grobentwurf wird durch einen Feinentwurf konkretisiert, um eine Basis f_r die Realisierung zu liefern. Sowohl der GUI-Entwurf, aber auch die Anwendungsfälle und die grobe Systemarchitektur (unterteilt nach eigenentwickelte Subsysteme und existierenden Softwaresystemen) sind die Voraussetzung f_r den Entwurf einer detaillierten Systemarchitektur.

In der Realisierung wird der Entwurf, genauer die konkrete Systemarchitektur mittels einer Programmiersprache umgesetzt. Bei dieser Umsetzung werden elementare Bestandteile der Systemarchitektur (Controller, Adaptor und Formatter) inkrementell umgesetzt. Diese werden dann als Komponenten bezeichnet und einem Test unterzogen. Die getesteten Komponenten werden nach der Realisierung in der Integration zum Portal integriert und einem Systemtest unterzogen, bevor das fertige Softwaresystem ausgeliefert wird.

Bei der Auslieferung wird das erstellte Softwaresystem mit dem Benutzertutorial ausgeliefert.

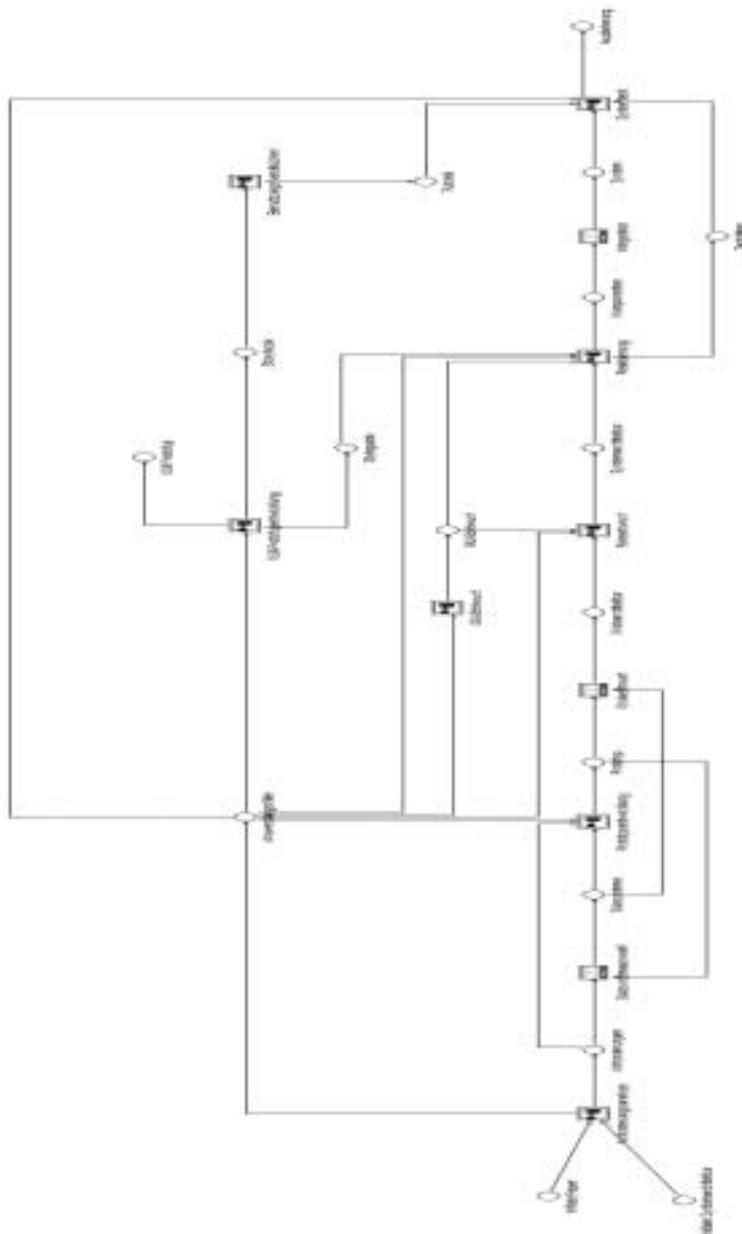


Abbildung 37: Gesamter Entwicklungsprozess

7.1.1.1 Anforderungsanalyse

Die initiale Systemarchitektur und die initiale Anforderungsliste (White Paper) sind der Ausgangspunkt zur Erarbeitung eines Anforderungskatalogs für das gesamte zu entwickelnde Softwaresystem. Dieser Anforderungskatalog wird auf verschiedene Art und Weise auf Widerspruchsfreiheit, Vollständigkeit, Redundanz geprüft. Hierzu werden in Form von Interviews Anwender und Betreiber befragt. Anwender, sind die Personen und Personengruppen, die das Portal letztendlich benutzen sollen, während Betreiber, die Personen und Personengruppen sind, die das Portal den Anwendern zur Verfügung stellt. Im Kontext der Projektgruppe sind hiermit Versicherungsaufendienstmitarbeiter (VAD) und Versicherungsunternehmen gemeint. Sowohl Anwender als auch Betreiber haben unterschiedliche, unter Umständen auch konkurrierende Anforderungen. Weitere Anforderungen werden durch die Betrachtung von existierenden Softwaresystemen ermittelt. Einige Versicherungsunternehmen setzen schon Unterstützungssysteme für den Versicherungsaufendienst ein. Diese Systeme wurden genauer betrachtet, um weitere Anforderungen zu ermitteln. Nach der Konsolidierung der einzeln ermittelten Anforderungen wird bei Bedarf der Anforderungskatalog korrigiert und evtl. erweitert und eine Überprüfung der Anforderungen erneut vorgenommen. Die Architektur des Portals wird durch die initiale Systemarchitektur grob vorgegeben. Für diese einzelnen Subsysteme werden die Anforderungen bestehend aus funktionalen und nicht-funktionalen Anforderungen beschrieben. Zusätzlich zu diesen beiden Arten von Anforderungen werden die Anforderungen an die Schnittstellen beschrieben, d.h. welche Informationen werden zwischen den Subsystemen voraussichtlich ausgetauscht. Die letztendlich abschließend konsolidierten Anforderungen werden dann zur weiteren Verwendung zur Verfügung gestellt.

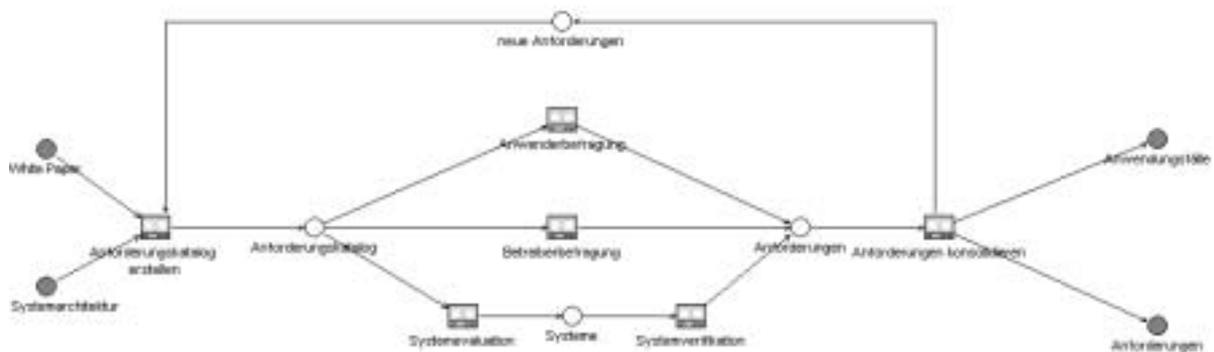


Abbildung 38: Anforderungsanalyse

7.1.1.2 Prototypentwicklung

Das Ziel der Prototypentwicklung ist die Ermittlung der Möglichkeiten zur Integration der Softwaresysteme untereinander. Diese Ermittlung erfolgt durch die Realisierung von Prototypen für jedes ausgewählte Softwaresystem. Für jedes Softwaresystem resp. jeden Prototyp werden Anforderungen, sog. *Key-Features*, festgelegt, die durch den Prototyp realisiert werden sollen. Diese Anforderungen an die Prototypen werden aus den Anwendungsfällen und den Anforderungen aus der Anforderungsanalyse erstellt. Auf der Basis der erarbeiteten Subsysteme werden zunächst existierende Softwaresysteme evaluiert, die eingesetzt werden können. Die Prototyprealisierung erfolgt durch Realisierung von Funktionen für die Softwaresysteme, welche die aufgestellten Anforderungen (*Key-Features*) erfüllen. Durch die Prototypen soll nachgewiesen werden, ob über eine Programmierschnittstelle auf Funktionen des zugrundeliegenden Softwaresystems zugegriffen werden kann.

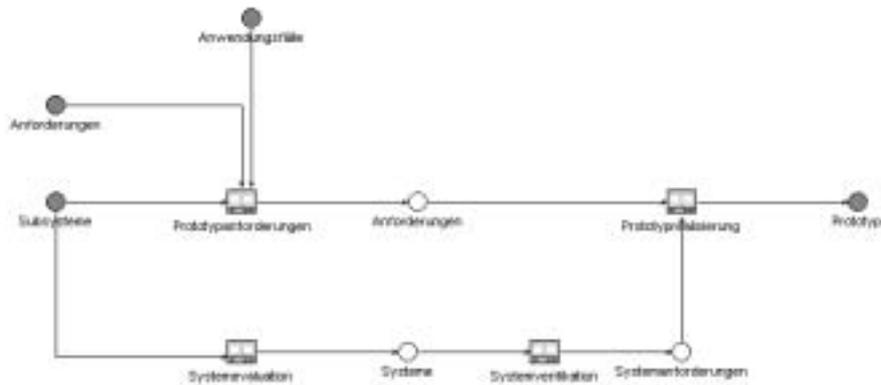


Abbildung 39: Prototypentwicklung

7.1.1.3 GUI-Prototypentwicklung

Ausgehend von den Anwendungsrufen wird in der GUI-Prototypentwicklung zunächst ein Storybook erstellt. Dieses Storybook wird in der GUI-Prototypentwicklung zur Erstellung eines Styleguides und dann zusammen mit dem Styleguide zur Realisierung des Prototyps verwendet.

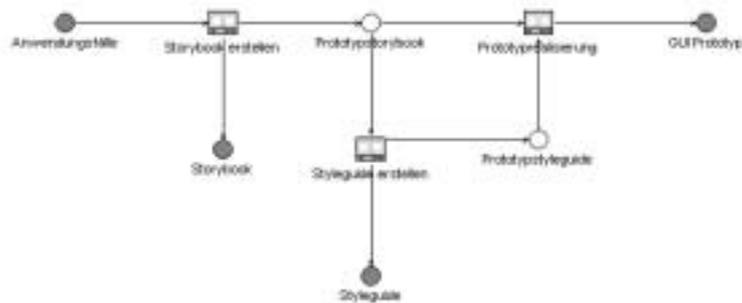


Abbildung 40: GUI-Prototypentwicklung

7.1.1.4 GUI-Entwurf

Ziel des GUI-Entwurfs ist die Beschreibung der Veränderungen der Benutzungsoberfläche nach dem Eintreten eines Ereignisses. Je nach Art des Ereignisses bzw. der Interaktion, das von einem Benutzer oder vom Portal selbst, ausgelöst werden kann, findet an der Benutzungsoberfläche eine Zustandsänderung statt. Sowohl die Interaktionen als auch die Zustandsänderungen werden beschrieben und ergeben den GUI-Entwurf.



Abbildung 41: GUI-Entwurf

7.1.1.5 Feinentwurf

Durch die grobe Systemarchitektur wird beschrieben, für welche Subsysteme existierende Software verwendet werden und welche Subsysteme durch eine Eigenentwicklung realisiert werden. Der Feinentwurf umfasst daher den Entwurf von Adaptern und Controllern. Adapter sind Schnittstellen für existierende Softwaresysteme. Sie ermöglichen den Zugriff auf Funktionen dieser Systeme. Daneben wird durch Controller der Workflow, resultierend aus den Anwendungsfällen, beschrieben. Für jeden Controller werden ein oder mehrere Formater benannt, welche die Eingabe und die Darstellung von Informationen an der Benutzeroberfläche ermöglichen. Der Entwurf erfolgt mittels UML durch die Beschreibung von Klassen. Durch die Identifizierung der Fassadenklassen und die Komposition von Klassen zu Paketen wird die letztendliche Systemarchitektur entworfen.

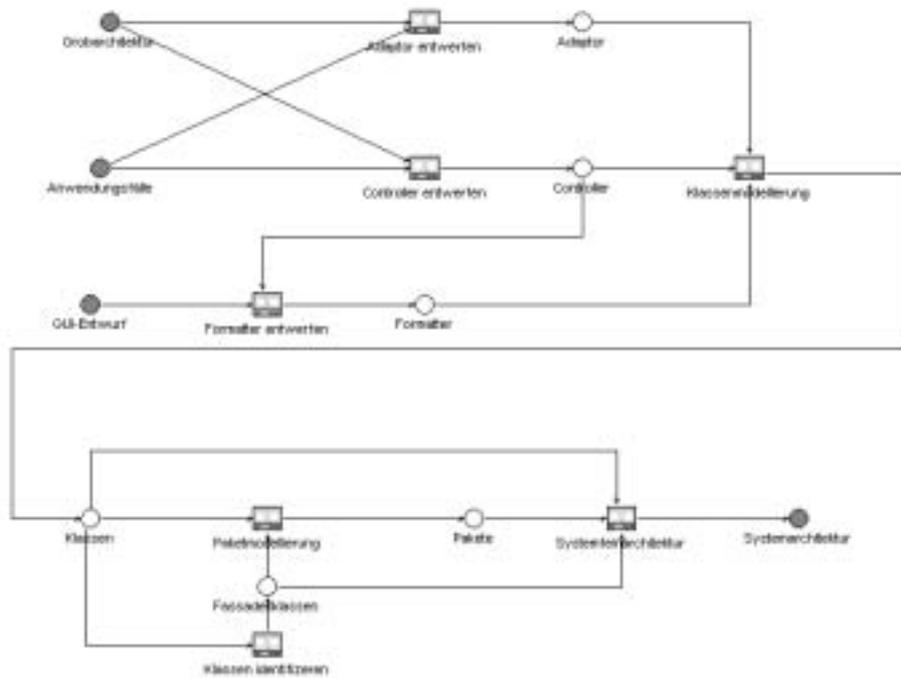


Abbildung 42: Feinentwurf

7.1.1.6 Realisierung

Die Implementierung der entworfenen Systemarchitektur, i.d.S. der Controller, Adaptern und Formater erfolgt in der Realisierung. Bei der Realisierung der Formater muss der vorgegebene Styleguide beachtet werden. Eine weitere notwendige Voraussetzung für die Realisierung der Formater ist die Realisierung einer GUI-Library. Durch die GUI-Library werden die erforderlichen elementaren Funktionen für die Ein- und Ausgabe von Informationen bereitgestellt, die bei der Realisierung der Formater verwendet werden können. Durch einen Middlewareprototyp wird vor Beginn der Realisierung der Controller und Adaptern eruiert, welche Middleware-technik (CORBA, RMI) zur Kommunikation eingesetzt werden kann. Alle realisierten Komponenten durchlaufen abschließend einen Komponententest. Auf der Grundlage der Anwendungsfälle werden Testdaten erstellt, die zum Testen der Komponenten dienen.

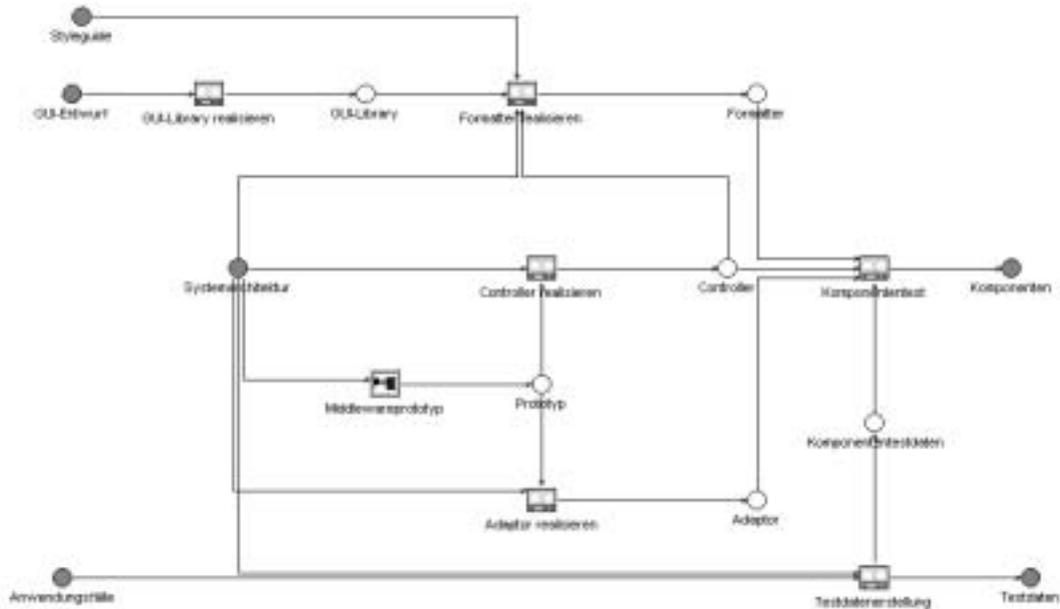


Abbildung 43: Realisierung

7.1.1.6.1 Middlewareprototyp

Um entscheiden zu können, welche Middleware-Technik eingesetzt werden kann, werden zunächst die Anforderungen an die Middleware-Technik aufgestellt. Hierbei werden nur Anforderungen erfasst, die für den speziellen Anwendungskontext auch relevant sind. Auf dieser Basis werden Systeme ausgewählt und verifiziert. Für diese Systeme werden Prototypen realisiert zum Nachweis der Umsetzbarkeit der aufgestellten Anforderungen. Ein realisierter Prototyp wird zur weiteren Verwendung und Verbesserung zur Verfügung gestellt.

7.1.1.7 Benutzungshandb,cher

Bei der Erstellung von Benutzungshandb,chern wird zwischen Benutzertutorials und Referenzhandb,chern unterschieden. Zur Gestaltung von Benutzungshandb,chern wird ein Styleguide Benutzerdokumentation verwendet. Hierin wird beschrieben, wie das Erscheinungsbild der fertigen Benutzerdokumentation aussehen soll. Für die Erstellung eines Tutorials wird das Storybook verwendet.

7.1.1.8 Systemtest

Das vollständig integrierte System durchläuft einen System- und Integrationstest. Hierzu werden Testdaten aufgebaut resp. die schon für den Komponententest verwendeten Testdaten ergänzt. Mit diesen Testdaten wird der Systemtest vollzogen. Nach erfolgreichem Test wird in der Systemauslieferung das System zusammen mit einem Benutzungstutorial an den Kunden ausgeliefert.

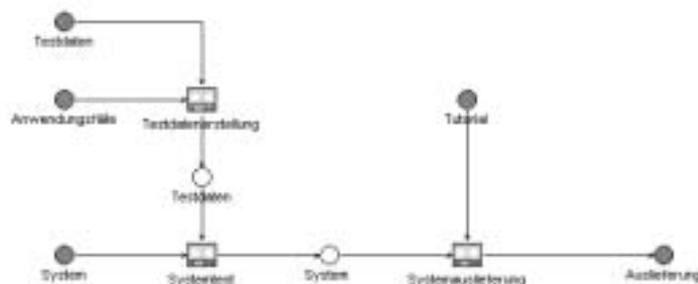


Abbildung 44: Systemtest

7.1.2 Besonderheiten des Entwicklungsprozesses

Dieser Entwicklungsprozess (in dem oben beschrieben Szenario) weist einige Besonderheiten auf. Electronic Commerce Systeme sind neben ihrer spezifischen Funktionalität, r ihre Anwendungsdomäne (B2B, B2C, ...) auch gekennzeichnet durch ihren Inhalt. Der Inhalt (auch als Content bezeichnet) umfasst neben den Informationen, die ein Electronic Commerce System zur Bearbeitung zur Verfügung stellt, auch deren Darstellung an der Benutzungsoberfläche. Dieser Inhalt ist oft multimedial, d.h. er umfasst eine Menge von textuellen Informationen, graphischen Darstellungen, Beweg- und Standbildern und Audioinformationen. Demzufolge kommt einem Contentmanager, der sich um diese multimedialen Informationen während des Entwicklungsprozesses kümmert, eine besondere Bedeutung zu.

7.1.2.1 Contentmanager / Ergonomieberater

Der Contentmanager ist eine zusätzliche Rolle während dieses Entwicklungsprozesses. Der Contentmanager kann wiederum andere Rolle beinhalten. Hierzu zählen der

- Medienautor
Diese Rolle übernimmt die Aufgaben der Beschaffung von Informationen und deren inhaltlicher Aufbereitung.
- Mediendidaktiker
Diese Rolle ist vor allem in Multimediasystemen mit lernspezifischen Elementen wichtig.
- Mediendesigner
Diese Rolle übernimmt Aufgaben zur audio-visuellen Gestaltung einer Anwendung.
- Medienproduzenten
Diese Rolle übernimmt die Erstellung von Bildern, Grafiken, Animationen, Audio- und Videosequenzen, ebenso wie die Recherche nach Medien und die Klärung von Urheberrechten.
- Medienredakteure
Sie übernehmen die Aufgaben der Qualitätssicherung für den multimedialen Teil einer Anwendung.

Zusätzlich zu dem Contentmanager mit seinen vielfältigen Aufgaben und Zuständigkeiten ist die Rolle des Ergonomieberaters zu besetzen. Die Aufgabe des Ergonomieberaters ist sicherzustellen, dass die entwickelte Software ergonomisch ist, d.h.

- Die Software muss an die auszuführende Arbeit angepasst sein.
- Die Systeme müssen den Benutzern Angaben über die jeweiligen Dialogabläufe unmittelbar oder auf Verlangen machen.
- Die Systeme müssen den Benutzern die Beeinflussung der jeweiligen Dialogabläufe ermöglichen sowie eventuelle Fehler bei der Handhabung beschreiben und deren Beseitigung mit begrenztem Arbeitsaufwand erlauben.
- Die Software muss entsprechend den Kenntnissen und Erfahrungen der Benutzer im Hinblick auf die auszuführende Aufgabe angepasst werden können.

Der Ergonomieberater fasst den Ergonomiebericht, in dem er darstellt, welche Richtlinien zur Zeit gelten und welche Rahmenbedingungen eingehalten werden müssen. Der Ergonomieberater sollte sich mit den arbeitsplatzschutzrechtlichen Richtlinien, besonders mit der Bildschirmarbeitsverordnung, auskennen. Außerdem sollte er wissen, was der spätere Anwender für Wünsche z.B. an die Benutzeroberfläche hat. Der Ergonomieberater steht im engen Kontakt mit dem Anforderungsanalytiker, dem Spezifizierer, dem Entwickler und dem Designer, so dass er überprüfen kann, ob die Software auch ergonomisch wird.

Das Ergonomie-Board an dem neben dem Contentmanager und dem Ergonomieberater, auch der Anforderungsanalytiker, der Spezifizierer, der Designer und der Entwickler teilnehmen, hat die Aufgabe die Einhaltung der Vorgaben sicherzustellen und Konflikte zu lösen.

8 Notwendige Erweiterungen zur Unterstützung

Durch CSCW-Systeme soll die Kooperation, Kommunikation und Koordination von mehreren bis vielen Personen, die sich gemeinsam als Gruppe oder Arbeitsgruppe verstehen, unterstützt werden. Die Gruppe hat ein gemeinsames Ziel, z.B. die Entwicklung eines Softwareprodukts in einem Softwareprojekt. Aufgrund der Komplexität der Zusammenarbeit innerhalb einer Gruppe unterstützen CSCW-Systeme in der Regel immer nur einen Aspekt besonders gut. So unterstützen Workflowmanagementsysteme die Koordination sehr stark, während Groupwaresysteme die Kooperation gut unterstützen und Konferenzsysteme ihren Schwerpunkt auf der Unterstützung der Kommunikation haben. Innerhalb ihres Schwerpunkts können CSCW-Systeme hinsichtlich ihres Aktivitätsgrades unterschieden werden: Aktive Systeme (Push-Systeme) steuern die Kooperation oder Kommunikation oder Koordination der Mitglieder einer Gruppe untereinander, indem sie kontextabhängig die erforderlichen Informationen und Dokumente zur Bearbeitung anbieten oder Aktionen ausführen. Workflowmanagementsysteme zählen im allgemeinen zu diesen aktiven Systemen. Aber auch intelligente Konferenzsysteme, die eine Interpretation der Arbeitsorganisationsstruktur vornehmen, können zielgerichtet Kommunikationsverbindungen aufbauen bzw. anbieten. Passive Systeme (Pull-Systeme) verwalten die aufgabenbezogenen Informationen und steuern nicht aktiv die Aufgabenverteilung. Zu diesen Systemen zählen die Groupwaresysteme.

Um jedoch eine komplette Unterstützung aller Aspekte (Kommunikation, Koordination, Kooperation) der Zusammenarbeit innerhalb einer Gruppe zu erreichen, müssen da nicht ein umfassendes CSCW-System existiert nicht CSCW-Systeme miteinander integriert werden. Nur dadurch kann eine umfassende Abdeckung der Funktionen erreicht werden. Ein Ansatz andere Systeme mit Workflowmanagementsystemen nicht aber von Workflowmanagementsystemen untereinander nicht zu integrieren, wird durch die Spezifikation der Schnittstellen von Workflowmanagementsystemen durch die *Workflow Management Coalition* erreicht. Durch die genormten und definierten Schnittstellen können m.E. auch Konferenzsysteme oder Planungssysteme mit Workflow- und Groupwaresystemen integriert werden.

Das bedeutet, wenn man als Basissystem für das computerunterstützte Arbeiten, ein Workflowmanagementsystem einsetzt, ist man m.E. in der Lage durch Integration von anderen CSCW-Systemen unter Einhaltung der Schnittstellendefinition, ein umfassendes CSCW-System zu konstruieren und zu entwickeln, welche alle Aspekte hinreichend unterstützen.

Hierbei hat es keine wesentliche Bedeutung, welche Rollen die Mitglieder einer Gruppe innehaben. Ob ein Gruppenmitglied als Projektleiter oder ob eine Untergruppe als Qualitätsmanagementteam agiert, führt zu keinen konkurrierenden oder ausschließenden Anforderungen an ein CSCW-System. Alle Rollen haben im wesentlichen die gleichen grundlegenden Anforderungen nicht bzgl. der Aspekte Kommunikation, Koordination und Kooperation nicht an ein CSCW-System. Lediglich durch die Vergabe von Berechtigungen wird der Zugriff auf Daten und Informationen, aber die Art und Weise der Kommunikation und der Kooperation gesteuert. So hat z.B. der Gesamtprojektleiter, der mittels eines Konferenzsystems mit seinen Teilprojektleitern kommunizieren will die gleichen Anforderungen an das CSCW-System, wie die Menge aller Entwickler, die über ein neues Release eines Softwareprodukts diskutieren möchten. In beiden Fällen muss durch ein Berechtigungs- und Sicherheitskonzept sichergestellt werden, dass nur identifizierte und autorisierte Personen an der Konferenz teilnehmen können.

Die Softwareentwicklung als eine Ausprägung der Gruppenarbeit kann demnach ebenfalls durch CSCW-Systeme unterstützt werden, wobei hier durch die CSCW-Systeme die Besonderheiten der Softwareentwicklung berücksichtigt werden müssen. Zwar gehört die Softwareentwicklung zu den schwach strukturierten Geschäftsprozessen, dennoch resultiert aus der Notwendigkeit, die Hauptziele Reduzierung der Entwicklungszeit, Reduzierung der Entwicklungskosten und Erhöhung der Qualität des Softwareprodukts und der daraus



Abbildung 45: Zeit-Kosten-Qualitäts-Matrix

resultierenden Unterziele erreichen zu müssen, der Wunsch die Softwareentwicklung durch einen durch einen geregelten Geschäftsprozess, mit den notwendigen individuellen Freiheiten, zu beschreiben. Die Softwareentwicklung als solche ist jedoch in der Regel hinreichend komplex. So werden umfasst z.B. die Softwareentwicklung als gesamter Geschäftsprozesse die Teilprozesse Projektmanagement (PM), Qualitätsmanagement (QM), Konfigurationsmanagement (KM), Änderungsmanagement (CM), Systementwicklung (SE).

Damit durch ein CSCW-System eine möglichst umfassende Unterstützung des Softwareentwicklungsprozesses einschließlich aller Teilprozesse erfolgen kann, wobei jedoch der Schwerpunkt nicht ausschließlich auf nur einem Aspekt einschließlich Kooperation, Kommunikation, Koordination einschließlich liegen sollte, müssen bestimmte Anforderungen an die Ausführungsunterstützung, oder die Modellierung etc. eingehalten werden.

8.1 Räumliche und zeitliche Bedingungen

Die Anforderungen an ein CSCW-System auf der Grundlage von zeitlichen und räumlichen Aspekten lassen sich klassifizieren. Die verschiedenen Rollen innerhalb einer Gruppe haben unter Berücksichtigung dieser zeitlichen und räumlichen Aspekte unterschiedliche Anforderungen an das CSCW-System.

Gleicher Ort einschließlich gleiche Zeit

Alle Gruppenmitglieder unabhängig von ihrer Rolle befinden sich zur gleichen Zeit in einem Raum und halten eine Sitzung ab. Folgende Anforderungen resultieren aus diesem Szenario:

1. Eine Gruppensitzung soll umfassend durch ein Softwaresystem unterstützt werden. Dies kann z.B. ein Sitzungssystem sein, alle Personen verfügen über einen Arbeitsplatz auf dem alle Kommentare aller Gruppenmitglieder sortiert ausgegeben werden. Das System übernimmt die Darstellung und Steuerung der Benutzeroberfläche für jeden einzelnen Arbeitsplatz individuell. Es werden automatisch Fragen und Antworten verwaltet und sortiert.
2. Der Projektleiter hat Möglichkeiten des individuellen Eingriffs. Er wird bei seiner Moderation unterstützt. So verfügt er z.B. über einen Terminkalender oder ein Werkzeug zur Modellierung der Aktivitäten aller Teilnehmer und aller nach zu erledigen Teilaufgaben.
3. Die Gruppe kann Hilfe durch ein Entscheidungsunterstützungssystem erhalten. Dieses System soll unstrukturierte Probleme aufzeigen und unter anderem eine Möglichkeit zur Durchführung und Kontrolle von Abstimmungen beinhalten.
4. Jedes Gruppenmitglied kann aus einer Präsentationssoftware heraus, Ergebnisse oder Daten anderen Personen mitteilen und vermitteln.

Gleiche Zeit einschließlich unterschiedlicher Ort

Alle Gruppenmitglieder befinden sich zur gleichen Zeit an verschiedenen Orten. Die Art und Weise der Verteilung von Personen an unterschiedliche Orten kann u.a. durch ihre verschiedenen Rollen begründet sein. So muss z.B. das QS-Team nicht unbedingt am gleichen Ort wie das Entwicklungsteam sein oder der Projektleiter ist zur

Zeit am Firmensitz, um der Geschäftsleitung zu berichten. Verschiedene CSCW-Systeme erleichtern mittels elektronischer Kommunikationswege das Zusammentreffen dieser verschiedenen Gruppenmitglieder durch folgende Hilfsmittel:

1. Audio- und Videokonferenzsysteme sorgen für die Visualisierung des anderen Gesprächspartners und den Austausch von mündlichen Informationen.
2. Screen-Sharing-Tools stellen den Bildschirminhalt nach dem WYSIWYG-Prinzip bei den anderen Gesprächspartnern dar. Je nach Implementierung hat jeder Partner Zugriff auf einzelne Objekte und kann diese auch verändern. (z.B. Grafiken)
3. Häufig ist es auch erforderlich, Dateien und andere Informationen durch E-Mail zu den Partnern zu versenden (z.B. Source-Code)

unterschiedliche Zeit ñ gleicher Ort

Alle Gruppenmitglieder arbeiten an einem Ort, um das Projektziel zu erreichen. Jedoch ist immer unbedingt eine direkte synchrone Kommunikation erforderlich. CSCW-Systeme in dem Bereich werden zur Unterstützung der administrativen Abläufe benötigt. Folgende Anforderungen sollten diese Systeme erfüllen:

1. Mithilfe einer Terminkalenderverwaltung für Gruppen kann eine rechtzeitige Abwicklung von Teilaufgaben realisiert werden
2. Projektmanagementsysteme zeigen den Status aller Aktivitäten an und simuliert unter Umständen die Zusammenarbeit der Gruppe, so dass erkannt werden kann, welche Aufgaben zuerst bearbeitet werden müssen. Weiterhin werden zu bearbeitende Aufgaben aufgeführt, damit keine Wartezeiten einzelner Personen entstehen.
3. Im Laufe der Teamarbeit entstehen viele Dokumente, die nicht für alle Teilnehmer von Interesse sind. Aus diesem Grunde existieren Textfilterungsprogramme, die eine Sortierung der Dokumente zulassen. In einigen Implementierungsformen wird jedem Benutzer eine Rolle zugeordnet. Durch diese Vereinbarung ist es möglich Daten zu filtern.
4. Elektronische Pinbretter enthalten Informationen zu allgemeinen Konventionen oder Terminen der Gruppe.

Unterschiedliche Zeit ñ unterschiedlicher Ort

Da sich die Gruppenmitglieder an verschiedenen Orten zu unterschiedlichen Zeiten befinden muss ein CSCW-System auf die Verbesserung der laufenden Koordination abzielen. So stellen sich folgende Anforderungen:

1. Ein ausgiebiges Konferenzsystem sorgt für die Kommunikation mehrerer Gruppenmitglieder. Dazu zählen Audio-, Video und Desktopverbindungen
2. Mithilfe von Werkzeugen zur Konversationsstrukturierung wird eine zielgerichtete und aufgabenbezogene Kommunikation gesteuert.

8.2 Grundzüge von Sprache und Konversation

Die Sprache ist nicht nur die formale Schablone für menschliche Äußerungen, sondern Teil einer Tätigkeit oder einer Lebensform (Watzlawick). Die Sprache ñ als gerufenes Wort ñ und die damit verwobene Tätigkeit ñ die Handlung nach den Worten ñ werden wissenschaftlich als Sprachspiel bezeichnet. Als Ausgangspunkt für die Untersuchung von Sprache und Konversation werden die einzelnen Aspekte der Linguistiktheorie ñ Syntax, Semantik, Pragmatik ñ betrachtet.

Syntax ist die sichtbare oder hörbare Form der Sprache. Die Grundelemente der Sprache (Buchstaben und Wörter) und ihre Kombination sind durch die Grammatik der Sprache bestimmt. Die Analyse der Syntaxebene ist frei von Interpretationen. Semantik stellt die Verbindung zwischen Strukturen in einer Sprache und den möglichen Bedeutungen her. Dies umfasst nicht nur einzelne Elemente (z.B. Worte) sondern auch Kombinationen daraus. Die Bedeutung von Worten und Sätzen ergibt sich aus der Art ihrer Verwendung (Gebrauch der Sprache). Prag-

matik behandelt die Konsequenzen von Sprachbenutzung. Die Pragmatik stellt den zentralen Aspekt der Betrachtung sprachlicher Konversation dar, nämlich das Hervorrufen und Interpretieren von Handlungen.

Unter dem Aspekt sprachliche Konversation als Grundlage für CSCW zu betrachten, bietet es sich an, diese drei Unterteilungen der Linguistik ebenfalls zu betrachten. Die Konsequenzen von Bedeutungen hängen von \langle berlegungen \rangle sprachliche Handlungen und Zusammenhänge ab und die Syntax ist in erster Linie wegen ihrer Flexibilität interessant, bedeutsame Unterschiede in der Konversation herauszustellen.

8.3 Erweiterung von Rollen

Die Grundlage für jegliche automatische Unterstützung eines Geschäftsprozesses durch CSCW-Systeme ist hierzu zählen auch Softwareentwicklungsprozesse ist ein genaues präzises gemeinsames Verständnis, über den zu unterstützenden Prozess erforderlich. Um dieses Verständnis zu erreichen, muss der Prozess beschrieben werden. Die Beschreibung des Prozesses unter Verwendung verschiedener Notationen (natürlichsprachlich, semi-formal, formal) erfolgen. Um diese Beschreibung erstellen zu können benötigt man Fachwissen. In Erweiterung der bisherigen Rollen bei der Softwareentwicklung werden neue Rollen benötigt. Die Softwareentwicklungsprozesse werden durch Prozessingenieure modelliert. Diese Prozessingenieure sind Experten in dem jeweiligen Anwendungskontext. Sie befragen Prozessabwickler, die Aktivitäten ausführen und evtl. auch die Prozessbesitzer, die für die Durchführung der Prozesse verantwortlich sind. Der Prozessingenieur erstellt ein Gesamtmodell des Prozesses.

8.4 Modellierung von verteilter Kommunikation

Ein Aspekt des CSCW ist die Unterstützung der Kommunikation. Der Aspekt der Kooperation wird durch die Modellierung der Geschäftsprozesse berücksichtigt, d.h. durch einen beschriebenen und modellierten Softwareentwicklungsprozess wird festgelegt, welche Aktivitäten in welchen Reihenfolgen \tilde{n} sei es sequentiell, parallel, oder alternativ zueinander \tilde{n} von welchen Personen oder Rollen ausgeführt werden. In den meisten Fällen besteht die Modellierung der Kommunikation zwischen Personen, die in diesen Entwicklungsprozess involviert sind, aus der Modellierung der Verteilung von Informationen und Dokumenten. Bei einer Softwareentwicklung, die bei großen Softwareprojekten zeitlich und räumlich verteilt durch größere Gruppen erfolgt, reicht dieser Grad der Modellierung von Kommunikation nicht aus. Wenn die Softwareentwicklung verteilt erfolgt, d.h. verschiedene räumlich verteilte Organisationseinheiten nehmen an der Softwareentwicklung teil, dann müssen auch die Kommunikationsstrukturen und -wege zwischen den Organisationseinheiten bzw. zwischen den Teilprozessen, die in den verschiedenen Organisationseinheiten ausgeführt werden, beschrieben werden.

Diese Kommunikationsstrukturen und -wege werden als Kommunikationskanäle bezeichnet [St98]. Beschreibungssprachen für Entwicklungsprozesse müssen die Möglichkeit bieten, Kommunikationskanäle mit zusätzlichen Attributen zu beschreiben. Diese Attribute umfassen: Verbindungsart, Kardinalität der Verbindung, Kapazität der Verbindung, Authentisierungsart der Verbindung, Datenvolumen einer Verbindung und Kostenmodell einer Verbindung. Das Attribut Verbindungsart kann zwei Ausprägungen annehmen: WAN-Verbindung oder Standverbindung. Unter einer WAN-Verbindung sind Telefonverbindungen, über analoge oder digitale terrestrische oder nicht terrestrische Netze zu verstehen, während unter einer Standverbindung eine permanente Kommunikationsleitung verstanden wird (LAN, WAN, Standleitung). Die Stelligkeit einer Verbindung legt fest, ob mehrere Teilprozesse einer Organisationseinheit gleichzeitig, über eine Verbindung Daten austauschen können (Multiplex-Verfahren). Hier werden in Analogie zur Entity-Relationship-Modellierung (ERM) die Kardinalitäten $\tilde{N}:1$, $\tilde{N}:\tilde{n}$ und $\tilde{N}:\tilde{m}$ angenommen. Mit dem Attribut Kapazität wird die maximale Datenmenge pro definierter Zeiteinheit angegeben, mit der ein Kommunikationskanal belastet werden kann (z.B. 64 Kbit/s bei ISDN, 10/100Mbit bei Ethernet). Die Wahl der Höhe der Kapazität richtet sich nach dem zu erwartenden Datenvolumen auf dem Kommunikationskanal. Das Datenvolumen ist die Summe aller Daten aller Teilprozesse einer Organisationseinheit mit der ein Kommunikationskanal maximal belastet werden kann. Mit dem Attribut Authentifizierung wird ein sicherer Kommunikationskanal gefordert. Unter Berücksichtigung entsprechender Schutzziele kann dies

entweder durch eine Call-Back-Verbindung (d.h. durch Identifikation der Organisationseinheiten) oder Verschlüsselung der Daten erfolgen. Die Definition der Schutzziele hängt im wesentlichen von der Art der Daten, die zwischen den Organisationseinheiten übertragen werden sollen, ab. Durch das Attribut Kostenmodell soll die Kosten für einen Kommunikationskanal angegeben werden.

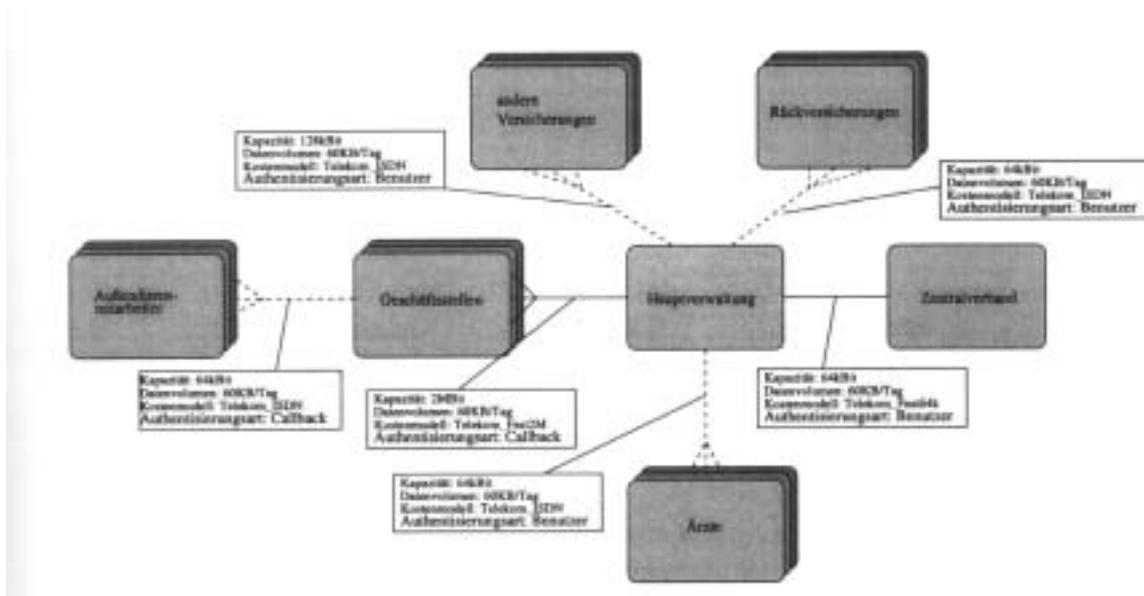


Abbildung 46: Kommunikationsattribute /St98/

Neben den Kommunikationskanälen spielen auch die Schnittstellen der einzelnen Teilprozesse zu den Kommunikationskanälen eine wichtige Rolle. Die Schnittstellen sind Aktivitäten von Teilprozessen, die entweder Daten einer Organisationseinheit zur Verfügung stellen oder Daten von einer anderen Organisationseinheit zur Weiterbearbeitung erhalten. Das besondere an diesen Schnittstellen ist, dass sie in einigen Fällen für alle Teilprozesse, die über einen Kommunikationskanal verbunden sind, gelten müssen. So muss z.B. das Attribut Datenformat einer Schnittstelle, durch das das Datenformat, mit dem die Daten über einen Kommunikationskanal übertragen werden, definiert wird, sowohl beim sendenden Teilprozess als auch beim empfangenden Teilprozess gleich sein (z.B. EDIFACT oder XML). Bei dem Attribut Übertragungsart jedoch, welches definiert, ob und welcher Teilprozess Initiator einer Kommunikationsverbindung ist und welcher Teilprozess nur reagiert, dürfen sich die Attribute nicht widersprechen.

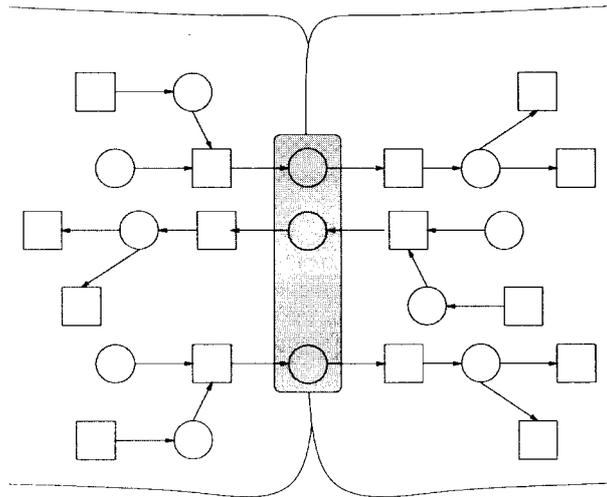


Abbildung 47: Kommunikationskanäle als Schnittstellen /St98/

CSCW-Systeme müssen bei der automatischen Unterstützung von Softwareentwicklungsprozessen die Besonderheiten der Modellierung von Prozessen berücksichtigen und syntaktische, semantische und pragmatische Funktionen vornehmen und die Ergebnisse der Funktionen bei der automatischen Unterstützung berücksichtigen. Dies bedeutet nicht nur, dass bei Bedarf die Weiterleitung von Dokumenten bei Auswahl oder Kapazitätsbelastung einer Kommunikationsverbindung sinnvoll verzögert wird, sondern auch, dass Informationen und Dokumente automatisch mit weiteren zusätzlichen Informationen

sog. Kontrollinformationen versehen werden. Diese Informationen dienen dem CSCW-System bei einem zentralen Ansatz oder den verschiedenen CSCW-Systemen bei einem dezentralen Ansatz zur Sicherstellung einer wieder aufsetzbaren Kommunikationsverbindung.

8.5 Modellierung von Zeiten und Kosten

Die Unterstützung von verteilten Softwareentwicklungsprozessen durch CSCW-Systeme umfasst nicht nur die Unterstützung der Gruppenarbeit, um eine Reduzierung der Entwicklungszeit, Reduzierung der Entwicklungskosten und Erhöhung der Qualität des Softwareprodukts zu erreichen. Es müssen diese Ziele auch berücksichtigt werden können. Aus diesem Grund ist es erforderlich, dass bei der Modellierung, der für ein CSCW-System zugrunde liegenden Geschäftsprozesse, die Attribute Liege-, Transport- und Reisezeit sowie Liege-, Transport- und Reisekosten mit berücksichtigt werden.

8.6 Dynamische Veränderung von Geschäftsprozessen

Verteilte Softwareentwicklungsprozesse zählen zwar zu den schwach strukturierten Geschäftsprozessen, trotzdem liegt den CSCW-Systemen eine Geschäftsprozessmodellierung zugrunde. Ein Merkmal dieser Regelprozesse ist, dass sie zwar eine Menge von Routineaufgaben beinhalten, aber im Gegensatz zum Routineprozess auch einfache und komplexe Aufgaben, die aufgrund von individuellen Besonderheiten immer wieder unterschiedlich parallel oder alternativ zueinander erledigt werden müssen. Diese individuellen Besonderheiten können auch dazu führen, dass Aufgaben nicht oder nur teilweise erledigt werden.

Dies hat zur Folge, dass sich die Modellierung eines Geschäftsprozesses während der Ausführung durch ein CSCW-System dynamisch ändern kann oder auch muss. Ein starres Prozessmodell, das nicht dynamisch angepasst werden kann, behindert die Mitglieder in ihrer eigenverantwortlichen Gruppenarbeit. Ihnen fehlen Eingriffsmöglichkeiten in das CSCW-System, um eigenverantwortlich und auch fallbezogen Veränderungen an dem

Prozessmodell vor oder während der Ausführung eines Geschäftsprozesses durchzuführen. So entstehen schnell Frustrationen, falls unvorhergesehene Ausnahmen auftreten, und die Gruppenmitglieder diese auf manuellem Wege behandeln müssen, nur weil das CSCW-System nicht dynamisch angepasst werden kann. Ein CSCW-System darf sich also nicht auf starre Prozessmodelle festlegen, sondern sollte den Gruppenmitgliedern möglichst viel Spielraum für Ausnahmeregelungen lassen. Hierdurch wird die Flexibilität als Unterziel der Kundenorientierung verbessert, um auf kurzfristige Änderungen von Kundenwünschen reagieren zu können und um dem Kunden jederzeit auskunftsfähig zu sein.

8.6.1 Klassifikation von dynamischen Modellierungen

Viele Geschäftsprozesse im Unternehmen sind von ihrem Aufbau und ihrer Abfolge zur Modellierungszeit nicht definiert genug. Deswegen können diese Prozesse nur zu Teilen in das System abgebildet werden, es ergeben sich unvollständige Prozessmodelle, die zur Laufzeit eines Geschäftsprozesses vervollständigt werden müssen. Denkbar sind auch Geschäftsprozesse, die sich von Fall zu Fall unterscheiden und so von den Mitarbeitern jeweils angepasst werden müssen (Goju97/).

- **Vervollständigung:**
Sind Geschäftsprozesse nur unvollständig im System abgebildet, dann müssen die entsprechenden Prozessdefinitionen zur Laufzeit des Geschäftsprozesses vervollständigt werden. Denkbar ist eine explizite Vorgehensweise durch einzelne Gruppenmitglieder oder ein implizites Ableiten des Arbeitsflusses durch das CSCW-System anhand der vorhandenen Daten.
- **Abweichung:**
Auch wenn ein Geschäftsprozess vollständig durch eine Prozessdefinition im CSCW-System abgebildet ist, können Änderungen notwendig werden. Beispiele hierfür sind spontane Kundenwünsche oder andere fallbezogene Besonderheiten. Hierbei muss das einzelne Gruppenmitglied (oder mehrere Gruppenmitglieder) vom vorhandenen Modell abweichen können.
- **Late-Modelling:**
Bei dieser Technik kann entweder ein Gruppenmitglied zur Laufzeit die fehlenden Informationen im Prozessmodell ergänzen, oder das CSCW-System selbst kann aus festgelegten Regeln und bisher vorhandenen Daten den weiteren Verlauf des Geschäftsprozesses festlegen. Late-Modelling setzt also voraus, dass fehlende Teile in dem Prozessmodell spätestens zur Laufzeit ergänzt werden und sich so ein vollständiges Modell des Geschäftsprozesses ergibt.
- **Post-Modelling:**
Können fehlende Teile in dem Prozessmodell auch nach dem Start des Geschäftsprozesses nicht ergänzt werden, da die hierfür notwendigen Informationen von keinem Gruppenmitglied verfügbar gemacht werden können, dann ist zur keiner Zeit ein klarer Geschäftsprozess zu erkennen. Häufig tritt dieser Fall bei kooperativer Arbeit auf, bei denen nicht ein schwach strukturierter Prozess, sondern die Zusammenarbeit einer Gruppe im Vordergrund steht. Bietet ein CSCW-System auch die Möglichkeit dieser Ad-hoc-Geschäftsprozesse an, dann muss für die spätere Auswertung aus dem spontanen Aktionen implizit ein vollständiges Prozessmodell erstellt werden können. Dafür muss das CSCW-System die Aktionen des Benutzers mitprotokollieren und anschließend modellieren oder einen Vorschlag für ein Modell anbieten.

Geschäftsprozesse kann man anhand ihrer Komplexität und ihres Strukturierungsgrads unterscheiden. Routineprozesse sind hoch strukturiert, Regelprozesse sind teilstrukturiert und Ad-hoc-Prozesse sind gänzlich unstrukturiert. Die Änderungsarten und ihre Maßnahmen lassen sich gut auf diese Prozesstypen abbilden. Routineprozesse erfordern häufig Abweichung vom normalen Arbeitsfluss als Reaktion auf unvorhergesehene Ereignisse. Während das Post-Modelling für Ad-hoc-Geschäftsprozesse ein adäquates Mittel ist, um die spontanen Aktionen in ein Prozessmodell abbilden zu können, werden die bei den Regelprozessen häufig auftretenden Modellierungslücken durch Late-Modelling zur Laufzeit beseitigt.

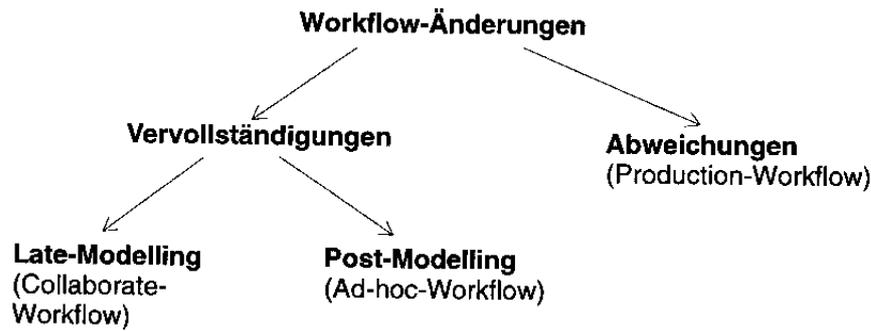


Abbildung 48: Klassifikation von Geschäftsprozessänderungen /K, 00/

8.6.2 Klassifikation von dynamischen Änderungsoperationen

Um Anpassungen und Modifikationen eines Geschäftsprozesses vornehmen zu können, muss ein CSCW-System entsprechende Funktionen bereitstellen. In der folgende Tabelle werden die erforderlichen Funktionen mit ihrem Zweck beschrieben.

Zielobjekt	Funktion	Zweck
Struktur des Workflows	Hinzufügen einer Aktivität	Ein neuer Arbeitsschritt ist erforderlich.
	Löschen einer Aktivität	Ein vorhandener Arbeitsschritt ist nicht mehr erforderlich.
	Überspringen einer Aktivität	Ein vorhandener Arbeitsschritt ist für den aktuellen Workflow oder die aktuelle Interaktion nicht mehr erforderlich.
	Vorziehen einer Aktivität	Ein erst später vorgesehener Arbeitsschritt soll schon zum aktuellen Zeitpunkt durchgeführt werden.
	Wiederholen einer Aktivität	Ein abgeschlossener Arbeitsschritt wird erneut durchgeführt, ohne die alten Aktionen explizit rückgängig zu machen.
	Zurücknehmen einer abgeschlossenen Aktivität	Ein abgeschlossener Arbeitsschritt wird erneut durchgeführt, wobei die alten Aktionen explizit rückgängig gemacht werden.
	Serialisieren zweier paralleler Aktivitäten	Zwei vorher parallele Arbeitsschritte nacheinander zur Durchführung bringen.
	Parallelisieren zweier serieller Aktivitäten	Zwei vorher serielle Arbeitsschritte gleichzeitig zur Durchführung bringen.
	Zusammenfassung von Geschäftsprozessmodellen	Zwei Teilgeschäftsprozessmodelle werden durch die Zusammenfassung ihres letzten bzw. ersten Arbeitsschrittes zu einem Geschäftsprozessmodell zusammengefasst.
	Aufteilung von Geschäftsprozessmodellen	Ein Geschäftsprozessmodell wird in zwei Teilgeschäftsprozessmodelle aufgeteilt.
	Vergrößerung einer Aktivität	Eine Menge von Arbeitsschritten des Teilgeschäftsprozesses werden zu einem Arbeitsschritt abstrahiert.
	Verfeinerung einer Aktivität	Ein Arbeitsschritt, die als Abstraktion für einen Teilgeschäftsprozess dient, wird durch die Arbeitsschritte des Teilgeschäftsprozesses konkretisiert.
Steuerung des Arbeitsflusses	Verzweigungsregel ändern	ändert die Entscheidungsregeln für eine Menge möglicher nachfolgend zu startender Arbeitsschritte
	Zusammenführungsregel ändern	ändert die Regel nachdem Arbeitsschritte ihren gemeinsamen Nachfolger starten.

Aktivitäten	Hinzufügen von Dokumenten	Bindung eines weiteren Dokuments an einen Arbeitsschritt
	Löschen von Dokumenten	Lösung eines Dokuments von einem Arbeitsschritt
	Hinzufügen eines Akteurs	Eine weitere Person ist für die Durchführung eines Arbeitsschrittes zuständig.
	Löschen eines Akteurs	Entbindet eine Person von der Durchführung eines Arbeitsschrittes
	Explizite Rollenaufweisung	Auswahl eines bestimmten Akteurs aus einer Menge von möglichen Akteuren
	Ändern von Zeitvorgaben	Verlegt den frühestmöglichen Zeitpunkt zur Durchführung eines Arbeitsschrittes auf einen anderen Termin.

Tabelle 13: Arten von dynamischen Änderungsoperationen /K, 00/

8.7 Organisationsplanung und -struktur

Neben den technischen Voraussetzungen, die für den Einsatz eines CSCW-Systems erforderlich sein müssen, gibt es eine weitere Kategorie von Vorbedingungen, die ein Unternehmen bei der Einführung eines solchen neuen Systems berücksichtigen muss; die zu erfüllenden organisatorischen Voraussetzungen. CSCW-Systeme sollen das Arbeiten in Gruppen in irgendeiner Weise unterstützen. Dies bedeutet offensichtlich, dass im Unternehmen zunächst einmal Gruppen vorhanden sein müssen, deren Zusammenarbeit dann durch ein entsprechendes System unterstützt werden kann. Eine Gruppe ist ein soziales System. Sie umfasst eine Menge von Menschen, zwischen denen eine relativ dauerhafte Struktur von Beziehungen besteht. Dies gilt auch für Arbeitsgruppen. Geht man noch etwas genauer auf den Begriff der Arbeitsgruppe ein, so kann man diese genauer dadurch bestimmen, dass sie aus einer Gruppe von Beschäftigten besteht, die in einem bestimmten funktional zusammenhängenden Abschnitt in all den Fragen, die nicht übergeordneten Interessen im Wege stehen, die anfallenden Entscheidungen selbst treffen kann. Hierzu zählen unter anderem die Einrichtung der Arbeitsplätze, die Reihenfolge der Bearbeitung, das Arbeitstempo, die Arbeitsverteilung und Leistungsüberwachung, die Qualitätskontrolle, die Einarbeitung neuer Mitglieder, usw. Bei solchen Gruppen wird die hierarchische Struktur auf den unteren Ebenen relativiert und die unmittelbaren Vorgesetzten arbeiten primär als Berater die außerdem für die Fortbildung innerhalb der Gruppe und den Informationsfluss zuständig sind. Es kommt auf den unteren Ebenen zu einer teilweisen Reintegration von planender und ausführender Arbeit, während die mittleren und zielrelevanten Entscheidungen nach wie vor auf den oberen Hierarchieebenen verbleiben.

Eine wichtige Voraussetzung für den Einsatz eines CSCW-Systems, ist die Aufweichung einer streng hierarchisch organisierten Unternehmung zumindest auf den unteren Hierarchieebenen vorangehen muss. Das bedeutet selbstverständlich nicht, dass ein bislang hierarchisch aufgebautes Unternehmen seine hierarchisierte Aufbaustruktur komplett neu organisieren muss. Es ist lediglich gefordert, dass Gruppen auf einer Hierarchieebene geschaffen werden, die mit dem CSCW-System entsprechend arbeiten können. Es ist hierbei durchaus denkbar, mehrere Gruppen zu schaffen, die jeweils für sich gesehen den Anforderungen an eine Arbeitsgruppe entsprechen. Ist dies der Fall, so kommt zusätzlich zur Kommunikation zwischen den Mitgliedern einer Gruppe der Aspekt der Kontaktaufnahme und -erhaltung zu anderen Arbeitsgruppen hinzu, der ebenfalls vom einzusetzenden CSCW-System unterstützt werden sollte. Neben den bislang genannten organisatorischen Voraussetzungen, die eventuell ein Überdenken der unternehmenseigenen Aufbaustruktur (zumindest auf tiefer liegenden Hierarchieebenen) zur Folge haben können, gibt es noch weitere Vorbedingungen, die ebenfalls wichtig sein können, um den reibungslosen Einsatz des neuen CSCW-Systems zu gewährleisten.

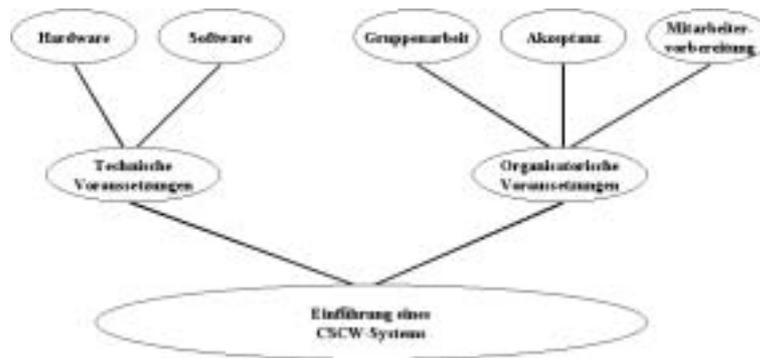


Abbildung 49: Vorbedingungen für ein CSCW-System

Zum einen spielt die Akzeptanz, die neuen Techniken entgegengebracht wird, eine entscheidende Rolle, wenn es darum geht, die Einführung eines CSCW-Systems in einem Unternehmen zu einem Erfolg zu machen. Diese Akzeptanz muss von allen Betroffenen erwartet werden können. In höheren Hierarchieebenen (insbesondere im mittleren und höheren Management) treten im allgemeinen bei der Einführung neuer Systeme keine Akzeptanzprobleme auf, was darauf zurückzuführen ist, dass diese Stellen in den meisten Fällen in den Entscheidungsfindungsprozess mit einbezogen sind und so bereits von vornherein die Vorteile der neu einzusetzenden Technologie erkennen können. Gerade bei der Einführung von CSCW-Systemen, mit denen ja auch oder vor allem Beschäftigte der unteren Hierarchieebenen konfrontiert werden (Sachbearbeiter, Schreibkräfte, etc.), ist es jedoch wichtig, dass auch diese Betroffenen dem neuen System ausgeschlossen gegenüberstehen und die Notwendigkeit erkennen. Hier muss also vor allem Aufklärungsarbeit geleistet werden, um möglichst allen Betroffenen das Gefühl zu geben, am Entscheidungsprozess beteiligt gewesen zu sein, oder doch zumindest um die oftmals erkennbare Skepsis bei Beschäftigten unterer Hierarchiestufen möglichst gering zu halten bzw. gänzlich zu verhindern. Eine letzte Voraussetzung, die hier aufgeführt werden soll, ist die Vorbereitung der Mitarbeiter auf die Einführung des neuen Systems. Hier geht es weniger um den Abbau der Schwellenangst, als vielmehr um die systematische Hinführung auf die tägliche Arbeit mit dem CSCW-System. Besonders wichtig ist hier zum einen die theoretische Einführung des divergenten Gruppenarbeitskonzepts im Hinblick auf die Unterstützung dieses Konzepts durch das einzuführende System und zum anderen die Durchführung von Schulungen für alle von der Einführung betroffenen Mitarbeiter, um ein sicheres Umgehen mit dem neuen System bereits von vornherein gewährleisten zu können. All diese Voraussetzungen organisatorischer Art tragen zur gelungenen Einführung eines CSCW-Systems bei und sind entscheidende Faktoren auch für den langfristigen Erfolg solcher Systeme. Denn nur durch eine systematische Vorbereitung aller Mitarbeiter kann verhindert werden, dass die Nutzung eines CSCW-Systems nach einer anfänglichen Neugierde-Phase zu einer Farce wird, weil die vom System zur Verfügung gestellten Funktionen entweder nur zwangsweise oder gar nicht mehr genutzt werden.

8.7.1 Lean-Management

Aufgrund der gegenseitigen Abhängigkeit von Ablauf- und Aufbauorganisation hat ein Geschäftsprozessmanagement Einfluss auf die gesamte Unternehmensorganisation (bzw. unter Einbeziehung der spezifischen Unternehmensziele auf das gesamte Managementsystem). Hierbei versteht man unter Managementsystem eine Methode mit der die Ablauforganisation, Aufbauorganisation und Ziele in Firmenpolitik in eines Unternehmens oder einer Organisationseinheit innerhalb eines Unternehmens festgelegt werden. Zwei unterschiedliche Managementsysteme, das Managementsystem nach Taylor und das Lean Management werden hier unterschieden.

Das tayloristische Management ist sowohl bei Produktionsbetrieben im herstellenden Gewerbe als auch bei Behörden und Verwaltungen aber auch bei großen Dienstleistungsunternehmen weitverbreitet. Allgemein findet man bei diesen Unternehmen stark strukturierte Arbeitsabläufe (Ablauforganisationen) und große hierarchische Aufbauorganisationen mit dem Ziel standardisierter Produkte zu erstellen. Bei den Arbeitsabläufen wird die Komplexität des Erstellungsprozesses reduziert, indem er in kleinste Einheiten zerlegt wird (als extremes Beispiel sei hier die Fließbandarbeit genannt). Dies erweist sich vorteilhaft zur Herstellung von Massenprodukten

gleichbleibender Qualität, die auch von gering qualifizierten Arbeitskräften erstellt werden können. Das Ziel ist die Erstellung konformer Produkte, mit möglichst hoher Stückzahl, ohne individuellen Charakter bzw. die Durchführung eines einheitlichen Verwaltungsaktes ohne die Schaffung von Ausnahme- und/oder Spezialfällen. Der Mitarbeiter ist ein Spezialist für eine kleine Einheit innerhalb des Arbeitsablaufs, kennt aber vielfach weder andere Einheiten und schon gar nicht den gesamten Arbeitsablauf. Er arbeitet funktionsorientiert, d.h. die schon erfolgte, als auch die anschließende Bearbeitung des Produkts ist nicht Teil seiner stark reglementierten Arbeit, sondern nur die mit seinem Arbeitsplatz verbundene Funktionalität, die von ihm permanent und gleichförmig durchgeführt wird. Die Aufbauorganisation wird perfektioniert und dadurch starr und geprägt von klar abgegrenzten Zuständigkeiten und festdefinierten Arbeitsplätzen. Die Entscheidungsspielräume der Mitarbeiter sind proportional zu ihrer Stellung in der Hierarchie und lassen keinen Raum für spontane und kreatives Handeln.

Demgegenüber steht das Lean Management. Ausgehend von der Idee des Lean Production, was erstmalig in der Automobilbranche zum Einsatz kam, kam man zu dem Ergebnis, dass die Produktivität weitaus weniger von dem Grad der Automatisierung eines Unternehmens abhängt, als von der Art und Weise, wie die Arbeit organisiert ist. Dieser Aspekt kann auch allgemein auf andere Unternehmensbereiche und andere Branchen übertragen werden, so dass man heute nicht mehr nur von Lean Production, sondern allgemein von Lean Management spricht. Um den Einsatz dieses Managementsystems auf unterschiedlichen Branchen und die dadurch bedingten Besonderheiten zu verdeutlichen, findet man vielfach auch die Begriffe Lean Banking, Lean Assurance, Lean Selling, Lean Education, usw. Die verschiedenen Prinzipien des Lean Managements verdeutlichen die Besonderheiten dieses Managementsystems, wobei unter Prinzipien verdichtete Handlungsanweisungen zur Gestaltung von Entscheidungsprozessen zu verstehen sind. Jede Aktivität in einem Unternehmen hat ungeachtet auf welche Ebenen der Unternehmensorganisation (Stelle, Abteilung, Werk, Konzern) und in welchen Funktionsbereichen (Entwicklung, Forschung, Marketing) sie stattfindet eine funktionale (Input/Output), eine strukturelle (Betriebsmittel, Personal) und eine prozessuale (Einordnung in die Ablauforganisation) Dimension.

Jede noch so kleine Änderung der Aktivität hat organisatorische Konsequenzen, die geprüft werden müssen. Wird z.B. einer der oben genannten Faktoren (z.B. der Output) müssen die anderen Faktoren darauf abgestimmt werden, um die Aktivität (aus als Subsystem bezeichnet) wieder optimal auszustatieren. Einige wesentliche Prinzipien des Lean Managements, die als Anforderungen für CSCW-Systeme zu übertragen sind, werden im folgenden beschrieben:

- **Vermeidung von Verschwendung**
Unter der Vermeidung von Verschwendung zählt im wesentlichen die in diesem Zusammenhang häufig genannte Beschränkung auf Kernaktivitäten. Ziel ist es, Aktivitäten ohne Wertzuwachs, d.h. solche Aktivitäten für die der Kunde nicht bereit ist zu zahlen, grundsätzlich zu vermeiden bzw. zu minimieren. Zu diesen Aktivitäten zählen z.B. Mehrfacherfassung von Daten, unnötige Kopien und Ablagen, das Warten auf Material oder das Führen unnötiger Lagerbestände.
- **Kundenorientierung**
Es wird davon ausgegangen, dass ein Individualisierungstrend bei der Nachfrage existiert und daher sollte das Konzept bei der Organisation individual- bzw. kundenauftragsorientiert sein. Grundvoraussetzung hierfür sind zur Verfügung stehenden Informationen über die Kundenumwelt bzw. Mechanismen zu deren Beschaffung und Aktualisierung. Eine logische Weiterentwicklung der Kundenorientierung ist der Versuch die Nachfrage beim Kunden gezielt zu beeinflussen und als Trendsetter zu operieren. Als Kunde versteht das Lean Management jedoch nicht nur den sog. externen Kunden, sondern jede Person der jeweils nächsten Aktivität im Unternehmen, welche die eigene Arbeit als Input verwerten muss. Output bzw. Input-Parameter können Material und/oder Informationen sein.
- **Prozessoptimierung**
Die Optimierung der Durchlaufzeit eines Prozesses geht vor der Optimierung der einzelnen Aktivitäten innerhalb eines Prozesses. Grund dafür ist die Erkenntnis, dass durch starke Segmentierung sich Liege-, Transport-, und Wartezeiten überproportionale gegenüber den Bearbeitungszeiten bei der Durchlaufzeit bemerk-

bar machen. Es ist ein Umkehrtrend von der Arbeitsteilung hin zur Erhöhung des Aufgabenumfangs des einzelnen Mitarbeiters zu beobachten, der allerdings erst möglich wurde durch den gestiegenen Bildungsstand der Mitarbeiter in den letzten Jahrzehnten und den ihn unterstützenden Informationstechniken.

Diese allgemeinen Prinzipien des Lean Managements, bei denen der Mitarbeiter in einer Gruppe im Vordergrund steht können durch verschiedene Arbeitsorganisationskonzepte umgesetzt und konkretisiert werden. Sind in einem Unternehmen bereits Arbeitsgruppen für den Einsatz eines CSCW-Systems geschaffen worden, so gehen hiermit einige Arbeitsorganisationskonzepte einher, deren Umsetzung CSCW-Systeme unterstützen sollten und die somit als allgemeine Ziele von CSCW-Systemen klassifiziert werden können. Im folgenden sollen diese Arbeitsorganisationskonzepte erläutert sowie ihr Bezug zu Groupwaresystemen herausgestellt werden (KrNa86/:

- **Job Rotation**
Job Rotation beinhaltet das Konzept des Arbeitsplatzwechsels, bei dem in einer bestimmten Zeit- und Reihenfolge Arbeitsplätze und Aufgaben gewechselt werden. Es ist also so, dass ein Beschäftigter mehrere zergliederte Arbeitsaufgaben beherrscht, ohne hierfür höher qualifiziert sein zu müssen, da alle Aufgaben im selben Tätigkeitsbereich liegen.
- **Job Enlargement**
Das Konzept des Job Enlargements (Arbeitserweiterung) sieht eine Anfügung zusätzlicher, gleich schwieriger Aufgabenelemente zur eigentlich ausgeübten Tätigkeit vor. Es ist jedoch in der Regel so, dass die zusammengefassten Aufgabenelemente auf der operativ-ausführenden Ebene angesiedelt sind, also keine planenden, steuernden oder kontrollierenden Tätigkeiten beinhalten. Diese Aufgaben verbleiben weiter in der Hand hierarchisch höher gelegener Stellen.
- **Job Enrichment**
Job Enrichment bezeichnet die Integration strukturell verschiedenartiger, auf unterschiedlichen Anforderungsniveaus angesiedelter Tätigkeiten zu umfassenden Arbeitsaufgaben. Es unterscheidet sich also im Gegensatz zu Job Enlargement nicht nur der Umfang, sondern auch die Art des Arbeitsinhalts. Beim Job Enrichment ist auch die Integration von sogenannten dispositiven Aufgaben, wie zum Beispiel Planungs-, Steuerungs- oder Koordinationsfunktionen möglich.
- **Fix-Vario-System**
Beim sogenannten Fix-Vario-System haben die Beschäftigten die Möglichkeit zu entscheiden, ob sie gleichbleibende Aufgaben an einem bestimmten Arbeitsplatz oder wechselnde Tätigkeiten an unterschiedlichen Arbeitsplätzen ausüben wollen. Man könnte dieses System also als Wahlmöglichkeit der Gruppenmitglieder zwischen Job- und Job Enrichment beschreiben. Es wird hierbei deutlich, dass die vorgestellten Konzepte lediglich Grundkonzepte darstellen, die im praktischen Einsatz beliebig miteinander vermischt werden können.

8.7.2 Unterstützung der LM-Konzepte durch CSCW-Systeme

Die arbeitsorganisatorischen Konzepte des Lean-Managements lassen sich implizit durch den Einsatz eines CSCW-Systems unterstützen. Im Grunde handelt es sich jedoch um Arbeitskonzepte die ohne jegliche CSCW-Systemunterstützung verwirklicht werden können bzw. müssen. Es ist lediglich so, dass CSCW-Systeme solche bereits umgesetzten Konzepte funktional unterstützen sollten, also lediglich als implizite Zielsetzungen anzusehen sind, deren praktische Umsetzung nicht durch das System selbst gewährleistet werden kann. Wohl aber sollten CSCW-Systeme die Möglichkeiten bieten, die oben erläuterten Arbeitskonzepte durch entsprechende Konfiguration des Systems zu unterstützen.

Wird ein CSCW-System in einem Unternehmen eingeführt, so können bei der Konfiguration Rechte und Berechtigungen vergeben werden, mit deren Hilfe zum einen festgelegt wird, wer überhaupt mit dem System arbeiten darf und zum anderen, welche Tätigkeiten ein Benutzer im System ausführen kann, bzw. welche Dokumente von ihm in welcher Weise bearbeitet werden dürfen. Insbesondere Job Enlargement und Job Enrichment können durch geeignete Konfiguration des CSCW-Systems unterstützt werden: Denkbar wäre hier zum Beispiel eine Vergabe von Berechtigungen in der Form, dass ein Sachbearbeiter, der normalerweise mit der Tätigkeit einer Auftragskorrektur befasst ist, durch eine Änderung im Berechtigungssystem nun auch in die Lage versetzt wird, den Auftrag vorher eigenständig zu prüfen. Was vorher immer durch einen entsprechenden anderen Bearbeiter erledigt werden musste. Im gleichen Zug könnte der vorher nur prüfende Bearbeiter auch die Berechtigung für die sofortige Korrektur des Auftrags erhalten. So dass beide Bearbeiter nun die gleichen Rechte bzw. Berechtigungen bezüglich der in der Datenbank vorliegenden Dokumente (in diesem Fall Aufträge) erhalten hätten. Je nach Einstufung der Tätigkeiten Auftragsprüfung und Auftragskorrektur kann hier von einer Unterstützung von Job Enlargement bzw. Job Enrichment durch das System geredet werden. Job Rotation als Arbeitskonzept von einem System unterstützen zu lassen dürfte hingegen schwerer fallen wobei es auch hier durchaus möglich wäre. Das System entsprechend zu konfigurieren, so dass das hinter Job Rotation stehende Arbeitskonzept unterstützt. Auch hier kommt es im Grunde nur auf die geeignete Vergabe von Rechten bzw. Berechtigungen an. Es wäre beispielsweise denkbar, dass Schreibkräfte von Zeit zu Zeit auch die am Empfang eines Unternehmens anfallenden Tätigkeiten übernehmen könnten. Beim Wechsel des Arbeitsplatzes wäre es dann allerdings notwendig, das System in gewissem Sinne umzukonfigurieren, da für eine Tätigkeit im Sekretariatsbereich (Schreibkraft) andere Berechtigungen auf die Datenbank vorhanden sein müssen (zum Beispiel auf Kundenadressen oder Formatvorlagen für Standardbriefe, etc.), während bei der Erledigung der Empfangstätigkeiten diese Rechte nicht gewahrt werden müssen. Hier wäre es zum Beispiel notwendig, Zugriff auf die internen Telefonnummern der Mitarbeiter zu haben, um eingehende Telefongespräche entsprechend weitervermitteln zu können. Man sieht hier deutlich, dass die Unterstützung von Job Rotation ebenfalls möglich ist, jedoch im Normalfall mit mehr Aufwand verbunden sein dürfte, wenn jedes Mal die entsprechenden Rechte bzw. Berechtigungen für die den Arbeitsplatz wechselnden Personen neu vergeben werden müssen. Es wäre natürlich auch denkbar, den entsprechenden Personen von vornherein alle Rechte für die Arbeitstätigkeiten in den verschiedenen Bereichen auf einmal zu geben. Jedoch widerspricht dies im allgemeinen der in Unternehmen üblichen Arbeitsplatz- und Aufgabentrennung [LS1].

9 Zusammenfassung und Bewertung

Die Betrachtungen zur Unterstützung von Geschäftsprozessen im allgemeinen und von Softwareentwicklungsprozessen im besonderen in den vorherigen Kapiteln haben gezeigt, dass keine der diskutierten Techniken alleine ausreicht, um Softwareentwicklungsprozesse geeignet und umfassend zu unterstützen.

Softwareentwicklungsprozesse zeichnen sich dadurch aus, dass sie einerseits komplex sind und andererseits wiederum individuell sind. Das bedeutet, es sind keine Routineprozesse, sondern immer auf den Anwendungskontext zugeschnitten sind, d.h. auf die Art der Anwendung, die es zu entwickeln gilt. Darüber hinaus sind sie bestimmt durch unvorhersehbare Umstände, die während der Softwareentwicklung auftreten können und durch eine hohe Ausnahmefallbehandlung. Trotzdem sind sie nicht vollständig individuell und kreativ, denn einige Bestandteile der Softwareentwicklung können detailliert beschrieben werden und einige Aktivitäten sind elementare Bestandteile einer Softwareentwicklung und damit vorgegeben. Hierdurch bedingt decken Softwareentwicklungsprozesse ein breites Spektrum ab. Einige Aktivitäten innerhalb der Softwareentwicklung erfordern strikte Koordination (Konfigurationsmanagement, Releasemanagement), andere wiederum weniger strikte Koordination aber mehr Kooperation (Programmierung, Entwurf) und wieder andere lassen sich nur durch Kommunikation unterstützen (Anforderungsanalyse). Nicht immer können jedoch komplexe Aktivitäten innerhalb eines Softwareentwicklungsprozesses strikt voneinander getrennt werden. Durch die Tatsache begründet, dass Softwareentwicklungsprozesse immer auf den Anwendungskontext zugeschnitten sind, können auch Budgets, Zeitpläne und Qualitätsrichtlinien nicht immer in einer zuverlässigen Weise geplant werden. Der wesentliche Faktor jedoch, der Softwareentwicklungsprozesse schwer planbar macht, sind die i.d.R. hohe Anzahl von Personen von

denen der Erfolg eines Softwareprojekts abhängt. Dieser Erfolg hängt maßgeblich von der Kommunikation, Koordination und Kooperation dieser Personen untereinander ab, wobei die Arbeitsorganisationsstruktur und die sozialen Faktoren eine wichtige, zum Teil nicht planbare, Rolle spielen (vgl. Kapitel 8.7 *Organisationsplanung und -struktur*). Die in Kapitel 2.1.1 *Klassifikation von CSCW-Systemen* vorgestellten Klassen von CSCW-Systemen unterstützen immer einen Aspekt der Gruppenarbeit (Koordination, Kommunikation und Kooperation) funktional ausreichend, während die anderen Aspekte nicht oder nur schwach unterstützt werden.

Darüber hinaus unterliegen Softwareentwicklungsprozesse bedingt durch ihre Natur einer Evolution. Diese Evolution umfasst (wie in Kapitel 8.6 *Dynamische Veränderung von Geschäftsprozessen* beschrieben) neben der Prozessmodellevolution auch die Prozessevolution während der Ausführung. Unterstützungssysteme (insbesondere aus dem akademischen Umfeld) bieten zwar einige Basisoperationen zur Modell- und zur Prozessänderung an, doch erfolgt keine vollständige automatische Unterstützung durch diese Systeme. Dies hat zur Folge, dass Änderungen an Prozessmodellen oder auch an bereits gestarteten Prozessen zwangsläufig manuell erfolgen. Hier müssen CSCW-Systeme funktionell erweitert werden.

Erschwerend kommt hierzu, dass Softwareentwicklungsprozesse, besonders bei großen Softwareprojekten, implizit verteilte Prozesse sind. Verteilte Entwicklungsgruppen, ein verteiltes Konfigurationsmanagement und verteilte Datenbanken bedingen andere Anforderungen an die Koordination, Kommunikation und Kooperation. Dies erfordert auch andere Prozessmodellierungssprachen oder zu mindestens Erweiterungen von existierenden Sprachen und auch Unterstützungssysteme mit mehr Funktionalität. Wenn Softwareprozesse oder bzw. Teilprozesse darüber hinaus nicht immer stationär bleiben, sondern mobil sind, muss das CSCW-System nicht nur die Kommunikation, Kooperation und Koordination der Arbeit der Gruppenmitglieder leisten, sondern auch die Kommunikation, Kooperation und Koordination der verteilten und mobilen Teilprozesse. Sicher ist nur, dass die Unterstützung von föderativen Softwareentwicklungsprozessen aufgrund ihrer Autonomiegegebenheiten, wie in Kapitel 5 (*Verteilte Systeme und Verteiltes Prozessmanagement*) beschrieben, neue oder erweiterte Softwareprozessbeschreibungssprachen erfordern. Die bisherigen Sprachen sind, bis auf wenige akademische Ansätze, nicht in der Lage, diese Sachverhalte adäquat ausdrücken zu können und weisen daher hier Defizite auf.

Des Weiteren stellt sich die Frage, ob die Art der Gruppenarbeit oder resp. welcher Geschäftsprozess oder welche Geschäftsprozesse die Gruppenarbeit kennzeichnen, die durch ein CSCW-System unterstützt werden soll, einen Einfluss auf die Anforderungen an ein CSCW-System hat. Die automatische Unterstützung von Gruppenarbeit durch CSCW-Systeme impliziert die automatische Unterstützung von Prozessen. Um jedoch Prozesse unterstützen zu können, benötigt man ein Prozessmodell; denn erst durch ein Prozessmodell wird definiert, wie sich konkrete Geschäftsprozesse während der Ausführung verhalten (vgl. Kapitel 6 *Beschreibungssprachen für Prozesse*). Dies bedeutet, dass durch Prozessmodelle die Anordnung von Aktivitäten, die Art der Ausführung von Aktivitäten durch Methoden, Funktionen und Werkzeuge, die Art und Typen von Dokumenten und Objekten, die erzeugt und bearbeitet werden sowie die Organisationsstrukturen, die für die konkrete Ausführung zuständig sind modelliert werden. Beim generellen Prozessmanagement wird das Softwareprozessmanagement und das Geschäftsprozessmanagement als jeweils eine Ausprägung unterschieden. Diese beiden Ausprägungen haben m.E. unterschiedliche Auswirkungen auf die automatische Unterstützung durch CSCW-Systeme. Der Schwerpunkt beim Softwareprozessmanagement liegt auf dem Management von kreativen und schwer veränderlichen Prozessen. Für diese Prozesse soll in diesem Kontext durch Verwendung einer geeigneten Modellierungssprache die Möglichkeit geschaffen werden, eine formale Analyse durchführen zu können, um eine Prozessoptimierung zu erreichen. Dies führt auch dazu, dass Unterstützungssysteme für diese Art von Prozessen ihren Schwerpunkt nicht auf der Ausführungsunterstützung des Geschäftsprozesses, sondern auf der Analyse von Geschäftsprozessen haben. Diese Systeme sind oft noch einem akademischen prototypischen Zustand und nicht für den industriellen Einsatz geeignet. Im Gegensatz dazu liegt der Schwerpunkt beim Geschäftsprozessmanagement auf der Unterstützung von starren Routinegeschäftsprozessen. Hier stehen zusätzlich Organisationsstrukturen im Vordergrund, so dass die Kooperation zwischen mehreren größeren und kleinen Organisationseinheiten gesteuert

werden kann. Da eine Analyse eines Geschäftsprozesses mit der Zielsetzung Prozessverbesserung zu erreichen nicht im Vordergrund steht, sind die Modellierungssprachen, die bei diesen Systemen verwendet werden, in den meisten Fällen nicht formal.

Die planbaren Bestandteile einer Softwareentwicklung sind dann auch in Form eines Prozessmodells modellierbar. Ein Softwareentwicklungsprozessmodell besteht aus einem Ablaufmodell, einem Datenmodell und einem Organisationsmodell. Durch das Ablaufmodell werden die durchzuführenden Aktivitäten, deren Reihenfolge und die Art und Weise der Durchführung der Aktivitäten spezifiziert. Durch das Datenmodell werden die Typen von Informationen und Objekten, die durch Aktivitäten erzeugt werden oder von Aktivitäten zur Ausführung benötigt werden spezifiziert. Durch das Organisationsmodell werden die Rollen spezifiziert, die bei der Ausführung von Aktivitäten involviert sind. Alle drei einzelnen Modelle ergeben ein vollständiges Softwareentwicklungsprozessmodell. Gerade das Organisationsmodell ist, weil Organisationsstrukturen und die damit verbundenen Personen und Rollen (vgl. Kapitel 3.2 *Rollen*) einen erheblichen Einfluss auf den Erfolg eines Softwareprojekts und damit auf den Softwareentwicklungsprozess haben (vgl. Kapitel 8.7 *Organisationsplanung und -struktur*) ein entscheidender Faktor. CSCW-Systeme müssen die den Softwareentwicklungsprozessen zugrundeliegenden Organisationsstrukturen dynamisch verwalten können. Dies ist wiederum auch nur bei einer Klasse von CSCW-Systemen der Fall.

Zur Beschreibung von Softwareentwicklungsprozessmodellen können verschiedene Sprachen eingesetzt werden. Diese Sprachen basieren entweder auf einem regelbasierten, einem netzbasierten oder einem natursprachlichen Ansatz. Jeder dieser Ansätze hat verschiedene Vor- und Nachteile, je nach dem, welcher Zweck bei der Modellierung verfolgt wird, ob zur Dokumentation des Prozesses, Analyse des Prozesses, Verbesserung des Prozesses oder zur Ausführungsunterstützung des Prozesses (vgl. Kapitel 6 *Beschreibungssprachen für Prozesse*).

Systeme zur Unterstützung der Ausführung eines Softwareentwicklungsprozesses neigen dazu eher als anwendungsspezifische Middleware eingesetzt zu werden. Dies bedeutet, dass weniger die Koordination im Vordergrund steht, sondern mehr die Kooperation, in dem Zugriff auf verteilte Informationen oder verteilte Kommunikation unterstützt wird. So unterscheidet sich die Unterstützung zur Ausführung eines Softwareprozessmodells vom Workflowmanagement als ein Vertreter der Kooperationsunterstützung in dem Detaillierungsgrad der Unterstützung. Zusätzlich zur Kooperationsunterstützung können Schwerpunkte auf der Anleitung und Führung oder dem Monitoring von Anforderungen liegen.

Ein Prozessmodell ist eine absolut notwendige Voraussetzung für eine sinnvolle automatische Unterstützung durch ein CSCW-System. Dies ist kein Widerspruch zu der Aussage, dass Softwareentwicklungsprozesse, dazu zählen besonders auch die verteilten Softwareentwicklungsprozesse, schwach strukturierte Prozesse sind. CSCW-Systeme müssen in der Lage sein, nicht nur die Aspekte der Verteilung bei der Bildung eines Prozessmodells zu berücksichtigen, sondern auch dynamische Prozessänderungen, wie in Kapitel 8.6 *Dynamische Änderung von Geschäftsprozessen* beschrieben sind sowohl an dem Prozessmodell, als auch an der Instanz eines Prozessmodells während der Ausführung zu zulassen.

Diese Fähigkeit eines CSCW-Systems ist umso höher zu bewerten, da gerade die in Kapitel 8.7 *Organisationsplanung und -struktur* beschriebenen Arbeitskonzepte des Lean-Managements bei der Softwareentwicklung, die eben keine stark strukturierte Entwicklung ist, immer häufiger zum Tragen kommen. Es werden gerade bei verteilter Softwareentwicklung und bei großen Softwareprojekten immer öfter neue Teams gebildet, da wenn die Projekte lange laufen, Gruppenmitglieder ausscheiden oder hinzukommen.

Ein Softwareentwicklungsprozess kann nur durch ein CSCW-System ausreichend unterstützt werden, wenn der Softwareentwicklungsprozess modelliert wurde. Dies gilt im verstärkten Maße für verteilte Softwareentwick-

lungsprozesse. Welche Art eines CSCW-Systems eingesetzt wird, ist bei der Unterstützung von einfacher Kommunikation, Koordination und Kooperation (vgl. Kapitel 2.1.1 *Klassifikation von CSCW-Systemen*) nicht ausschlaggebend. Jedes CSCW-System legt durch seine Funktionalität den Schwerpunkt auf einen evtl. zwei Aspekte. Dies führt zwangsläufig zu der Forderung, dass verschiedene CSCW-Systeme miteinander integriert werden müssen, um alle Aspekte bei einer verteilten Softwareentwicklung umfassend zu unterstützen.

Vielmehr ist es so, dass einige grundlegende CSCW-Funktionalitäten benötigt werden, welche die Kooperation, d.h. das gemeinsame Bearbeiten von Dokumenten ermöglichen und dass einige dedizierte Softwareentwicklungsprozessunterstützungsfunktionalitäten erforderlich sind.

Allgemein kann diese Argumentation belegt werden, wenn man zunächst von der Zielsetzung ausgeht, dass die Unterstützung von Gruppenarbeit (und als solche sind die Aktivitäten in der Softwareentwicklung in aller Regel aufzufassen) die Automatisierung von einfachen und weniger kreativen Aktivitäten nach sich zieht. Eine solche Teilautomatisierung von Prozessen bzw. Teilprozessen wiederum erfordert die Definition der Prozesse.

Die Unterstützung von einfacher Kommunikation, Koordination und Kommunikation stellt für CSCW-Systeme im allgemeinen keine Herausforderung dar. Zur Unterstützung von flexiblen verteilbaren Softwareentwicklungsprozessen sind deshalb neue Techniken erforderlich. Dies sollte ebenso eine strikte Unterstützung der Kooperation (evtl. durch Workflowfunktionalität) umfassen, genauso wie für die nicht planbaren Aktivitäten eines Softwareentwicklungsprozesse eine lockere Unterstützung zulassen. Inwieweit mobile Teilprozesse einen Einfluss auf die Entwicklungsprozessmodelle und Prozessbeschreibungssprachen haben, ist noch nicht abzusehen.

Es bleibt letztendlich eine softwaretechnische Herausforderung, einerseits die notwendigen Erweiterungen an den Modellierungssprachen für Softwareentwicklungsprozesse konzeptionell – d.h. syntaktisch und semantisch – zu spezifizieren, als auch durch sinnvolle Integration von einzelnen CSCW-Systemen ein CSCW-System zu entwickeln und zu realisieren, welches einen hohen Grad der Anforderungen erfüllt, wobei eine erweiterte Modellierungssprache wesentliche Grundlage des CSCW-Systems sein muss.

10 Literaturverzeichnis

- /AI98/ P. Alpar *Kommerzielle Nutzung des Internet* Springer Verlag, München, 1998
- /AlFr99/ K. Alfert, A. Fronk, et. al. *Abschlussbericht der Projektgruppe HEU* Memo Nr. 103, Universität Dortmund, FB 4, LS 10, 1999
- /Ba82/ H. Balzert *Die Entwicklung von Softwaresystemen. Prinzipien, Methoden, Sprachen und Werkzeuge* Reihe Informatik, Band 34, BI-Verlag, 1982
- /Ba94/ R. Baumann *Datenverarbeitung unter Zeitbedingungen* in: IS, April 1984
- /Ba96/ H. Balzert *Lehrbuch der Software-Technik: Software-Entwicklung* Spektrum Akademischer Verlag, 1996
- /Ba98/ H. Balzert *Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung* Spektrum Akademischer Verlag, 1998
- /BeLe95/ J. BergÉ, O. Levia, J. Rouliard *High-Level System Modeling* Kluwer Academic Publishers, 1995
- /B^De96/ M. B^hm, W. Deiters, et.al. *Workflow Management as Teleservice* in: Proc. 7th Joint European Networking Conference, Terena Verlag, Amsterdam, 1996
- /BoGr00a/ M. Book, V. Gruhn, L. Sch^pe *Realizing An Integrated Electronic Commerce Portal System* in: Proc. AMCIS 2000, Aug. 2000
- /BoGr00b/ M. Book, V. Gruhn, L. Sch^pe *Ein EC-Portal als Integrationsplattform f, r Electronic Commerce Systeme* in: Proc. EMISA 2000, Nov. 2000
- /BoGr01/ M. Book, V. Gruhn, L. Sch^pe *A Specific Software Development Process for an Electronic Commerce Portal* In. Proc. APAQS 2001, Dez. 2001
- /BoJa99/ G. Booch, I. Jacobson, J. Rumbaugh *The Uniifed Modeling Language User Guide* Addison Wesley, Reading, MA, USA, 1999
- /ChSc77/
- /Cr82/ D.H. Crocker *RFC 822: Standard for the Format of ARPA Internet Text Messages* <ftp://ftp.isi.edu/in-notes/rfc822.txt> (zitiert: Mai 2000)
- /CuKe92/ B. Curtis, M. Kellner, J. Over *Process Modeling* in: CACM, Vol. 35, No. 9, ACM Press, 1992
- /De96/ G. Derszteler *Workflow Management Cycle ñ Ein Ansatz zur Integration von Modellierung, Ausführung und Bewertung workflowgest, tzier Gesch^asprozesse* In: WI, Band 38, Heft 6, Springer Verlag, 1996
- /De97/ W. Deiters *Prozessmodelle als Grundlage f, r ein systematisches Management von Gesch^asprozessen* in: IFE, Band 12, Heft 2, Springer Verlag, 1997
- /PiRo95/ A. Picot, P. Rohrbach *Organisatorische Einsatzm^glichkeiten von Workflow-Management-Systemen* Deutsches Institut f, r Normung, DIN-Mitteilungen, Band 74, Nr. 4, Beuth Verlag, 1995
- /Fl94/ C. Floyd *Software Engineering ñ und dann?* In: IS, Band 17, Heft 1, Springer Verlag, 1994
- /EmDr98/ K. El Emam, J.-N. Drouin, W. Melo (eds.) *SPICE* IEEE Computer Society, 1998
- /Ge99/ A. Gehmeyr *Electronic Commerce - ein < berblick* in: Objektspektrum, Vol, No.2 1999
- /GoJo96/ J. Gosling, B. Joy, G. Steele *The Java Language Specification* Addison Wesley Longman Inc. Reading, MA, USA, 1996
- /GoJu97/ T. Goesmann, K. Just-Hahn, et.al *Flexibilit^als Ziel beim Einsatz von Workflow-Management-Systemen* in: Proc. EMISA Fachgruppentreffen, 1997
- /Gr91/ V. Gruhn *Validation and Verification of Software Process Models* Dissertation, Universität Dortmund Fachbereich Informatik, 1991
- /Gr96/ V. Gruhn *Gesch^asprozess-Management als Grundlage der Software-Entwicklung* In: IFE, Band, x, Heft 11, Springer Verlag, 1996
- /GrGr95/ G. Graw, V. Gruhn *Distributed Modelling and Distributed Enaction of Business Processes* in: Proc. 5th European Software Engineering Conference, Springer Verlag (LNCS 989), 1995

/GrGr96/	G. Graw, V. Gruhn, H. Krumm	<i>Support of Cooperating and Distributed Business Processes</i>	in: Proc. Int. Conf. On Parallel and Distributed Systems, IEEE Computer Society Press, June 1996
/GrKa96/	V. Gruhn, M. Kampmann	<i>Modellierung unternehmens, bergreifender Gesch%asprozesse mit FUNSOFT-Netzen</i>	in: Wirtschaftsinformatik, Band 38, Heft 4, Vieweg Verlag, 1996
/GrWe00/	V. Gruhn, U. Wellen	<i>Autonomies in a Software Process Landscape</i>	submitted to TOSEM, 2000
/GrWe99/	V. Gruhn, U. Wellen	<i>Software Support for Distributed Business Processes</i>	in: Proc. Asia-Pacific Software Engineering Conference, IEEE Computer Society Press, Dec. 1999
/HaCh93/	M. Hammer, J. Champy	<i>Reengineering the Corporation</i>	Harper Collins Publishers, 1993
/HaMa99/	C. Hastedt-Marckwardt	<i>Workflow-Management-Systeme: Grundlagen, Standards und Trends</i>	in: IS, Vol. 22, Heft xx, 1999
/He98/	D. Henning	<i>Eine Systemarchitektur f, r dezentrales Prozessmanagement in autonomen Organisationseinheiten</i>	Diplomarbeit, Universit%oDortmund, FB 4, LS 10, 1998
/HeSa99/	A. Hermanns, M. Sauter	<i>Management-Handbuch Electronic Commerce</i>	Vahlen Verlag, M, nchen, 1999
/HeSa99a/	A. Hermanns, M. Sauter	<i>Electronic Commerce ñ Grundlagen, Potentiale, Marktteilnehmer und Transaktionen</i>	in: /HeSa99/
/Hu89/	W. Humphrey	<i>Managing the Software Process</i>	Addison-Wesley, SEI-Series, USA, 1989
/Il99/	J.A. Illik	<i>Electronic Commerce</i>	Oldenbourg Verlag, 1999
/Ja97/	S. Jablonski	<i>Architektur von Workflow-Management-Systemen</i>	in: IFE, Band 12, Heft xx, 1997
/KeN, 92/	G. Keller, M. N, ttgens, A.-W. Scheer	<i>Semantische Prozessmodellierung auf der Grundlage Nereignisgesteuerter Prozessketten (EPK)i</i>	Ver^ fentlichungen des Instituts f, r Wirtschaftsinformatik (Iwi), Heft 89, 1992
/Ko00/	C. Kopka	<i>Ein Vorgehensmodell f, r die Entwicklung multi-medialer Lernsysteme</i>	Memo Nr. 104, Universit%oDortmund, FB 4, LS 10, 1999
/KoB195/	P. Kotler, F. Bliemel	<i>Marketing-Management</i>	Sch%er Poeschel Verlag, Stuttgart, 1995
/Kr94/	H. Krumm	<i>Rechnernetze und verteilte Systeme</i>	Vorlesung, Universit%oDortmund, FB 4, LS 12, 1994
/KrNa86/	D. Kr, ger, A. Nagel	<i>Mischarbeit im B, ro- und Verwaltungsbereich beim Einsatz neuer Technologien</i>	Bundesanstalt f, r Arbeitsschutz, Dortmund, 1986
/K, 00/	R. K, hl	<i>Ein internetbasiertes objektorientiertes Workflow-Management-System mit laufzeitdynamischen Verhalten</i>	Diplomarbeit, Universit%oDortmund, FB 4, LS 10, 2000
/LaSc97/	P. Langner, C. Schneider, J. Wehler	<i>Prozessmodellierung mit EPKis und Petrinetzen</i>	in: Wirtschaftsinformatik, Band 39, Heft 5, Vieweg Verlag, 1997
/Le99/	C. Leifkes	<i>Entwicklungsprozesse f, r die Softwareentwicklung</i>	Seminararbeit, Universit%oDortmund, FB 4, LS 10, 1999
/Li99/	F. Lindert	<i>Fraktales Prozessmanagement</i>	Dissertation, TU Berlin, FB 13, 1999
/LiHa97/	D. Linke, P. Haertsch, C. Hoffmann, A. Lindemann	<i>Integrierte Electronic Commerce Systeme: Auswahlkriterien und Evaluation aktueller Produktangebote</i>	Universit%oSt. Gallen, Institut f, r Wirtschaftsinformatik, Technischer Bericht BusinessMedia/58/WZMO, 1997
/LiZi99/	D. Lincke, H. Zimmermann	<i>Integrierte Standardanwendungen f, r Electronic Commerce ñ Anforderungen und Evaluationskriterien</i>	in: /HeSa99/
/Ma99/	P. Marwedel	<i>Prozessrechner-technik (Eingebettete Realzeit-Systeme)</i>	Vorlesung, Universit%oDortmund, FB 4, LS 12, 1999
/MeTu99/	M. Merz, T. Tu, W. Lamersdorf	<i>Electronic Commerce ñ Technologische und organisatorische Grundlagen</i>	in: IS, Vol 22, No. , 1999
/Mu99/	G. M, ller-Ettrich	<i>Objektorientierte Prozessmodelle</i>	Addison-Wesley, Bonn, 1999
/M, Ko97/	G. M, ller, U. Kohl, D. Schoder	<i>Unternehmenskommunikation: Telematiksysteme f, r vernetzte Unternehmen</i>	Addison-Wesley, Bonn, 1997
/NiCh97/	P. Nixon, V. Cahill, F. Rabhi	<i>Building Parallel and Distributed Systems</i>	in: Computer Journal, Vol. 30, No. 8, 1997
/OMG99/	Object Management Group	<i>OMG site for CORBA and UML</i>	http://www.omg.org (zitiert: Juni 2000)

/PaSi94/	H.J.Pagel, W. Six	<i>Software Engineering</i>	Addison Wesley, 1991
/PfWe94/	W. Pfeiffer, E. Weiss	<i>Lean Management</i>	Erich Schmidt Verlag, Berlin, 1994
/Po00/	A. Pohlmann	<i>Visualisierung und Simulation von Prozesslandschaften - Die lokale Sichtweise</i>	Diplomarbeit, Universität Dortmund, FB 4, LS 10, 2000
/Re82/	W. Reisig	<i>Petrinetze</i>	Springer Verlag, 1982
/RoVe95/	H. Rombach, M. Verlage	<i>Directions in software process research</i>	in: M. Zelkowitz (eds.) <i>Advances in Computers</i> Vol. 41, Academic Press, 1995
/RTCA92/	Requirements, Technology and Concepts for Aviation	<i>Software Considerations in Airbone Systems and Equipment Certification RTCA/DO-178B</i>	RTCA Inc. Washington, USA, 1992
/Sc91/	H.-J. Schneider (Hrsg.)	<i>Lexikon der Informatik und Datenverarbeitung</i>	Oldenbourg Verlag, 1991
/Sc97/	H. Schuster	<i>Architektur verteilter Workflowmanagementsysteme</i>	Dissertation, Universität Erlangen-Nürnberg, Institut für math. Maschinen und Datenverarbeitung (Informatik), 1997
/ScLi98/	B. Schmid, M. Lindemann	<i>Elements of a Reference Model for Electronic Markets</i>	in: HICSS, 1998
/ScN, 97/	A.-W. Scheer, M. Nüttgens, V. Zimmermann	<i>Objektorientierte Ereignisgesteuerte Prozesskette (oEPK) - Methode und Anwendung</i>	Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Heft 141, 1997
/ShLa90/	A.P. Sheth, J.A. Larson	<i>Federated Database Systems for Managing Distributed, Heterogenous, and Autonomous Databases</i>	in: ACM Computing Surveys, Vol. 22, Nr. 3, 1990
/St98/	C. Stoppe	<i>Service Engineering - Eine Methode zur Definition, Visualisierung und Simulation von EC-Anwendungen</i>	Diplomarbeit, Universität Dortmund, FB 4, LS 10, 1998
/St98/	J.W. Steuck	<i>Geschäftserfolg im Internet</i>	Cornelson Verlag, 1998
/St99/	C. Steinweg	<i>Projektkompass Softwareentwicklung</i>	Vieweg Verlag, Wiesbaden, 1999
/Ta96/	A.S. Tanenbaum	<i>Computer Networks</i>	Prentice Hall, New York, USA, 1996
/Th97/	G. Thaller	<i>Der Individuelle Software-Prozess</i>	BHV Verlag, 1997
/Th98/	G. Thaller	<i>SPICE - ISO 9001 und Software in der Zukunft</i>	BHV Verlag, 1998
/Ve97/	M. Verlage	<i>Ein Ansatz zur Modellierung grosser Software-Entwicklungsprozesse durch Integration unabhängig erfasster rollenspezifischer Sichten</i>	Dissertation, Universität Kaiserslautern Fachbereich Informatik, 1997
/WfMC96/	Workflow Management Coalition	<i>The Workflows Management Coalition Specification: Terminologie & Glossary</i>	Document Number WfMC-TC-1011, (Vers. 2.0), 1996
/WfMC98a/	Workflow Management Coalition	<i>The Workflows Management Coalition Specification: Workflow Management Application & Programming Interface</i>	Document Number WfMC-TC-1009, (Vers. 2.0), 1998
/WfMC98b/	Workflow Management Coalition	<i>The Workflows Management Coalition Specification: Process Definition Interchange</i>	Document Number WfMC-TC-1016-P, (Vers. 7.05beta), 1998
/Wi95/	L. Wittgenstein	<i>Tractatus logico-philosophicus Logisch-philosophische Abhandlung</i>	Edition Suhrkamp Nr.12, 1995
/Zw96/	V. Zwass	<i>Electronic Commerce: Structures and Issues</i>	in: IJEC, Vol. 1, No. 1, 1996
/Zw99/	V. Zwass	<i>Structure and Macro-Level Impacts of Electronic Commerce: From Technological Infrastructure to Electronic Marketplaces</i>	in: K.E. Kendall "Emerging Information Technology" Sage Publications 1999

11 Abbildungsverzeichnis

Abbildung 1: Klassifikation von Geschäftsprozessen /PiRo95/	6
Abbildung 2: Klassifikationsschema für CSCW-Systeme /M ₁ Ko97/	9
Abbildung 3: Raum-Zeit-Matrix	10
Abbildung 4: Nachrichtensysteme	11
Abbildung 5: Konferenzsysteme	12
Abbildung 6: Sitzungsunterstützungssysteme	13
Abbildung 7: Architektur eines Workflowmanagementsystems	16
Abbildung 8: Workflow-Terminologie /WfMC96/	16
Abbildung 9: Workflowreferenzmodell /WfMC96/	17
Abbildung 10: Das klassische Phasen- oder Wasserfallmodell	19
Abbildung 11: Das iterierte Phasenmodell	20
Abbildung 12: Das iterierte Prototypingphasenmodell	20
Abbildung 13: Modell der evolutionären Softwareentwicklung	21
Abbildung 14: Das Spiralmodell des Softwareentwicklungsprozesses	22
Abbildung 15: Beispiel für den Aktivitäts- und Produktfluss über die vier Submodelle	22
Abbildung 16: Submodell Systemerstellung	23
Abbildung 17: Arten des elektronischen Geschäftsverkehrs	32
Abbildung 18: Marktphasen /St98/	33
Abbildung 19: Markttransaktionen /St98/	33
Abbildung 20: Klassifikation von Markttransaktionen /HeSa99/	35
Abbildung 21: Kommunikation in einem verteilten System	37
Abbildung 22: Schema verteilter Teilprozesse	39
Abbildung 23: Klassifikation von Autonomierarten, -granularität und -ebenen	40
Abbildung 24: Push/Pull-Kommunikation	40
Abbildung 25: Persistenzverhalten	40
Abbildung 26: ereignisgesteuerte Prozessketten (EPK)	42
Abbildung 27: erweiterte EPK	42
Abbildung 28: UML Aktivitätsdiagramm	43
Abbildung 29: Bestandteile von Petrinetzen	43
Abbildung 30: Entwicklungshistorie von höheren Petrinetzen	43
Abbildung 31: Merkmale von B/E-Netzen	44
Abbildung 32: Merkmale von S/T-Netzen	44
Abbildung 33: Merkmale von Pr/T-Netzen	44
Abbildung 34: Merkmale von FUNSOFT-Netzen	45
Abbildung 35: Subsysteme des IPSI-Portals	45
Abbildung 36: Architektur des IPSI-Portals	46
Abbildung 37: Gesamter Entwicklungsprozess	48
Abbildung 38: Anforderungsanalyse	49
Abbildung 39: Prototypentwicklung	50
Abbildung 40: GUI-Prototypentwicklung	50
Abbildung 41: GUI-Entwurf	50
Abbildung 42: Feinentwurf	51
Abbildung 43: Realisierung	52
Abbildung 44: Systemtest	52
Abbildung 45: Zeit-Kosten-Qualitäts-Matrix	55
Abbildung 46: Kommunikationsattribute /St98/	58
Abbildung 47: Kommunikationskanäle als Schnittstellen /St98/	59
Abbildung 48: Klassifikation von Geschäftsprozessänderungen /K ₁ 00/	61
Abbildung 49: Vorbedingungen für ein CSCW-System	63

Um den Rahmen der Ausarbeitung nicht zu sprengen, werden im folgenden Konfigurationsbeispiele für Systeme zur asynchronen Bearbeitung gemeinsamen Materials, jeweils bezogen auf die vorgestellten Arbeitskonzepte vorgestellt und diskutiert. Die Beschränkung auf Systeme zur asynchronen Bearbeitung gemeinsamen Materials geschieht an dieser Stelle aus dem Grund, weil die meisten der heutzutage verbreiteten und eingesetzten Groupwaresysteme dieses Konzept verfolgen, während die anderen Groupware-Konzepte entweder nicht dem intuitiven Verständnis von Groupware entsprechen (asynchroner Nachrichtenaustausch, Email), nicht sehr verbreitet sind (synchroner Nachrichtenaustausch bzw. Videokonferenzen) oder schlicht und einfach bislang kaum eingesetzt werden und eher zu Forschungszwecken dienen (synchrone Bearbeitung gemeinsamen Materials). Geht man also davon aus, dass ein Groupwaresystem zur asynchronen Bearbeitung gemeinsamen Materials eingesetzt werden soll (im weiteren wird unter dem Begriff Groupwaresystem ein solches System verstanden), so ergeben sich bei der Einführung bzw. Konfiguration dieses Systems gewisse Möglichkeiten, durch deren Realisierung sich bestimmte Konsequenzen für den Personenkreis ergeben, der im Endeffekt mit dem System arbeiten soll.

Als Beispiel für eine solche Konfiguration soll hier das Groupwaresystem Lotus-Notes dienen. Bevor ein konkretes System als Beispiel herangezogen werden kann, muss zunächst natürlich geklärt werden, ob es den Anforderungen und Kriterien überhaupt genügt und somit als Groupwaresystem im oben definierten Sinne gelten kann.

Was ist also Lotus Notes?

Lotus Notes ist ein auf einer Client-Server-Architektur basierendes Informationsverwaltungssystem, das Benutzern ermöglicht, gemeinsame Informationen mittels Netzwerken und Workflows zu sammeln, zu strukturieren und untereinander auszutauschen.

Diese Definition des Systems, die einer Broschüre der Lotus Corporation entnommen ist, lässt erkennen, dass es sich bei Notes offensichtlich (zumindest laut Herstellerangaben) um ein Groupwaresystem in unserem Sinne handelt, da gemeinsame Informationen (gemeinsames Material) von Benutzern gesammelt, strukturiert und ausgetauscht werden können (zeitversetzt, also asynchron bearbeitet werden können).

Eine Lotus Notes-Anwendung umfasst hierbei eine oder mehrere (Notes-) Datenbanken, mit denen verschiedene, dem System bekannte Benutzer arbeiten können.

Die Rechte und Berechtigungen werden hierbei über eine sogenannte Zugriffskontrollliste (auch Access Control List bzw. ACL genannt) festgelegt. Diese ACL besitzt im wesentlichen drei wichtige Funktionen:

- sie bestimmt, ob ein Benutzer auf eine bestimmte Datenbank zugreifen darf,
- sie legt genau fest welche Aktionen ein Benutzer innerhalb der Datenbank ausführen darf,
- sie wird zur Definition von Gruppen eingesetzt, deren Zugriffsrechte auf (Bildschirm-) Masken und Ansichten auf die Datenbank weiter eingeschränkt werden können.

Für jeden Benutzer lassen sich in dieser ACL die Berechtigungen auf Datenbanken über sogenannte Zugriffsebenen festlegen. Lotus Notes bietet hierzu verschiedene solcher Zugriffsebenen an. Dazu zählen zum Beispiel der Entwickler, der Leser oder auch die Zugriffsebene kein Zugriff. Die einzelnen Ebenen unterscheiden sich

jeweils durch unterschiedliche Berechtigungen auf bestimmte Datenbanken (beispielsweise darf ein Benutzer der Zugriffsebene Leser nicht schreibend auf eine Datenbank zugreifen, während die Ebene kein Zugriff den ihr zugewiesenen Benutzern jegliche Zugriffsversuche (schreibender oder lesender Art) von vornherein untersagt). Wie kann nun ein solches Berechtigungssystem dazu beitragen, die vorgestellten Arbeitskonzepte zu unterstützen?

Durch die Verteilung von Rechten und Berechtigungen auf einzelne Datenbanken in beliebig feiner Granularität lassen sich die Arbeitsaufgaben einzelner Benutzer relativ einfach abbilden. Es lässt sich (in Lotus Notes) also mittels der ACL und Zugriffsebenen genau definieren, wer welche Aufgaben innerhalb des Einsatzgebiets von Notes erledigen kann bzw. darf. Hierdurch bietet sich nun auf relativ einfache Art und Weise die Möglichkeit, die verschiedenen Arbeitskonzepte zu unterstützen.