

Unterstützung organisatorischen Lernens in Softwareunternehmen durch Projektdatenbanken

**Michael Bächle
Bernd Jahnke**

Herausgeber:

**Professor Dr. Bernd Jahnke, Universität Tübingen
Abteilung für Betriebswirtschaftslehre, insb. Wirtschaftsinformatik,
Melanchthonstraße 30, 72074 Tübingen,
Telefon: 07071/29-75422, Telefax: 07071/21229
E-Mail jahnke@uni-tuebingen.de
WWW <http://www.uni-tuebingen.de/uni/w04/wi/leitung.html>**

Unterstützung organisatorischen Lernens in Softwareunternehmen durch Projektdatenbanken

von

Dr. Michael Bächle
Prof. Dr. Bernd Jahnke

Abteilung für Betriebswirtschaftslehre,
insb. Wirtschaftsinformatik,
Universität Tübingen

{michael.baechle, jahnke}@uni-tuebingen.de

Zusammenfassung

Der Beitrag stellt die Ergebnisse eines Forschungsprojekts zur Erfassung und Speicherung organisatorischen Produktivitätswissens von Softwareunternehmen vor. Die zugrundeliegende Arbeitsthese geht davon aus, daß das Management den eigentlichen Engpaß für den Erfolg von Softwareunternehmen darstellt. Zur Verbesserung organisatorischer Lernprozesse durch das Management wird der betriebswirtschaftliche Zusammenhang zwischen Produktivität und Kosten im Prozeß der Softwareentwicklung dargestellt und anhand einer Fallstudie ein empirischer Lösungsansatz zum Aufbau einer Projektdatenbank als organisatorisches Gedächtnis vorgestellt, die im Rahmen des Forschungsprojekts in Kooperation mit einem Softwareunternehmen erstellt wurde.

Inhaltsverzeichnis

Abbildungsverzeichnis	V
1 Organisatorisches Lernen: Problemfaktor Management	1
2 Organisatorisches Wissen und organisatorisches Gedächtnis	2
3 Prozeßproduktivität und -kosten	3
4 Wissensartefakte: Softwaremaße	5
5 Organisatorisches Gedächtnis: Projektdatenbank	8
5.1 Vorüberlegungen	8
5.2 Modellierung der Wissensartefakte: Datenerfassungsstandard	8
5.3 Modellierung des organisatorischen Gedächtnisses: Konzeptionelles Datenmodell der Projektdatenbank	10
5.3.1 Projekt als Basis des ERM	10
5.3.2 Modellierung von Projektkunden	11
5.3.3 Modellierung von Mitarbeitern	11
5.3.4 Modellierung von Projektphasen	12
5.3.5 Modellierung des Projektumfelds	12
5.3.6 Modellierung der Projektkennzahlen	12
5.3.7 Modellierung von Phasenkenzahlen	13
5.4 Architektur des organisatorischen Gedächtnisses: Relationenmodell der Projektdatenbank	13
5.5 Implementierung des organisatorischen Gedächtnisses: Prototypische Realisierung der Projektdatenbank	16
6 Resümee	18
Literatur	20
Bisher erschienene Arbeitsberichte	21

Abbildungsverzeichnis

Abb. 1:	Teilprozesse der Softwareentwicklung	3
Abb. 2:	Merkmals- und Skalenarten	5
Abb. 3:	Überblick des vorhandenen organisatorischen Wissens (Ergebnis des Brainstormings)	9
Abb. 4:	Beispiel des Datenerfassungsstandards (Teil I)	9
Abb. 5:	Beispiel des Datenerfassungsstandards (Teil II)	10
Abb. 6:	ERM der Projektdatenbank	11
Abb. 7:	Projektsicht der Projektdatenbank	16
Abb. 8:	Filter für eine Suchanfrage in der Projektdatenbank	17

1 Organisatorisches Lernen: Problemfaktor Management

Seit der wegweisenden NATO-Tagung „Working Conference on Software Engineering“ 1968 wird von der Notwendigkeit zur Lösung der Softwarekrise gesprochen. Hierunter wird die Problematik der Softwareunternehmen subsumiert, daß die von ihnen entwickelten Softwareprodukte häufig zu teuer, zu spät und mit nicht immer befriedigender Qualität ausgeliefert werden. Lange Zeit wurde die mangelnde Beherrschung der technischen Probleme im Prozeß der Softwareentwicklung als hierfür ursächlich betrachtet und eine Vielzahl von Lösungsvorschlägen erarbeitet, die in Methoden, Verfahren, Techniken und Werkzeugen einmündeten. Mittlerweile wird jedoch zunehmend die fast ausschließliche Fokussierung auf technische Aspekte bei der Überwindung der Softwarekrise als nicht ausreichend kritisiert und die eigentliche Ursache in anderen Faktoren vermutet: „Die Softwarekrise stellt sich dar als vor allem eine Krise des Projektmanagements, und zwar des Managements im doppelten Sinne: der Prozesse der Steuerung, der Prozesse der Entscheidungs- und Konsensfindung und der Akteure, d.h. der Entscheidungsträger und Promotoren in den Projekten, und insbesondere in deren Umfeld. Notwendig sind Verfahren, die eine Dynamisierung und Synchronisierung von Entwicklungsprozessen und Projektsteuerung leisten.“¹

Nicht allein die Technik, sondern das Management muß untersucht werden, sollen Produktivität und Qualität eines Softwareunternehmens wettbewerbsfähig bleiben. Wettbewerbsfähigkeit ist nicht zuletzt das Ergebnis der Fähigkeit, einmal gelerntes Wissen über den Softwareentwicklungsprozeß und seine Rahmenbedingungen zu konservieren und zu aktivieren. Dieses organisatorische (oder auch: kontextuelle) Wissen dient der Selbststeuerung des komplexen Prozesses und ist Ausdruck der Lernfähigkeit im Sinne lernender Organisationseinheiten. Hier weisen viele Softwareunternehmen eine gefährliche Unfähigkeit zur Nutzung ihres kontextuellen Wissens auf. Weltz/Ortmann stellen dazu in einer empirischen Untersuchung fest: „Folge des fehlenden Transfers des kontextuellen Wissens war die erstaunlich geringe institutionelle Lernfähigkeit, der wir in zahlreichen Unternehmen begegneten und die dafür sorgte, daß mit bemerkenswerter Beharrlichkeit Fehler wiederholt wurden bzw. Fehlentwicklungen unkorrigiert blieben.“² Ein Beispiel hierfür ist der „mythische Personenmonat“, also die häufig getroffene Management(Fehl-)entscheidung, daß terminliche Probleme durch eine rein numerische Erhöhung der Personalkapazität zu lösen seien.³

Festzuhalten ist damit, daß viele Managementprobleme in Softwareunternehmen durch die mangelnde Umsetzung der Idee des organisatorischen Lernens verursacht werden. Als organisatorisches Lernen wird dabei der „Prozeß der Erhöhung und Veränderung der organisationalen Wert- und Wissensbasis, die Verbesserung der Problemlösungs- und Handlungskompetenz sowie die Veränderung des gemeinsamen Bezugsrahmens von und für Mitglieder innerhalb der Organisation“⁴ verstanden. Organisatorisches Lernen setzt die Speicherung organisatorischen Wissens in Form eines organisatorischen Gedächtnisses voraus.

2 Organisatorisches Wissen und organisatorisches Gedächtnis

Organisatorisches Wissen umfaßt „jegliche Kenntnisse, die der Organisation momentan zur Verfügung stehen.“⁵ Es kann durch ein organisatorisches Gedächtnis konserviert und für eine spätere Nutzung im Unternehmen bereitgestellt werden.

Das organisatorische Gedächtnis bezeichnet die Fähigkeit, Inhalte im Unternehmen über einen beliebigen Zeitraum hinweg zu speichern sowie gezielt zu nutzen. Es bezieht sich u.a. auf die Frage nach den geeigneten Speichermedien für die Archivierung des organisatorischen Wissens. Als solche kommen verschiedene Arten von Medien in Betracht, wie Handbücher, Pläne, Standards und Normen, aber auch informationstechnikbasierte Informationssysteme. „Es liegt auf den ersten Blick nahe, Informationssysteme als ideale Träger des organisatorischen Gedächtnisses zu betrachten, weil sie bezüglich ihrer Speicherkapazität und Verarbeitungsgeschwindigkeit die anderen Speichermedien, teilweise auch die menschliche Informationsverarbeitungsfähigkeit, bei weitem übertreffen.“⁶ Dabei sind aktive und passive Informationssysteme unterscheidbar: Aktiv bedeutet, daß hinterlegte Prozeduren aufgrund eintretender Ereignisse (z.B. Umweltveränderungen) selbständig Schlußfolgerungen ziehen und dem Anwender zur Verfügung stellen. Passive Informationssysteme hingegen können das in ihnen gespeicherte organisatorische Wissen nicht aktiv aufgrund kontextueller Reize verarbeiten. Sie stehen dem Manager jedoch als komfortables Speicher- und Abfragemedium zur Verfügung.

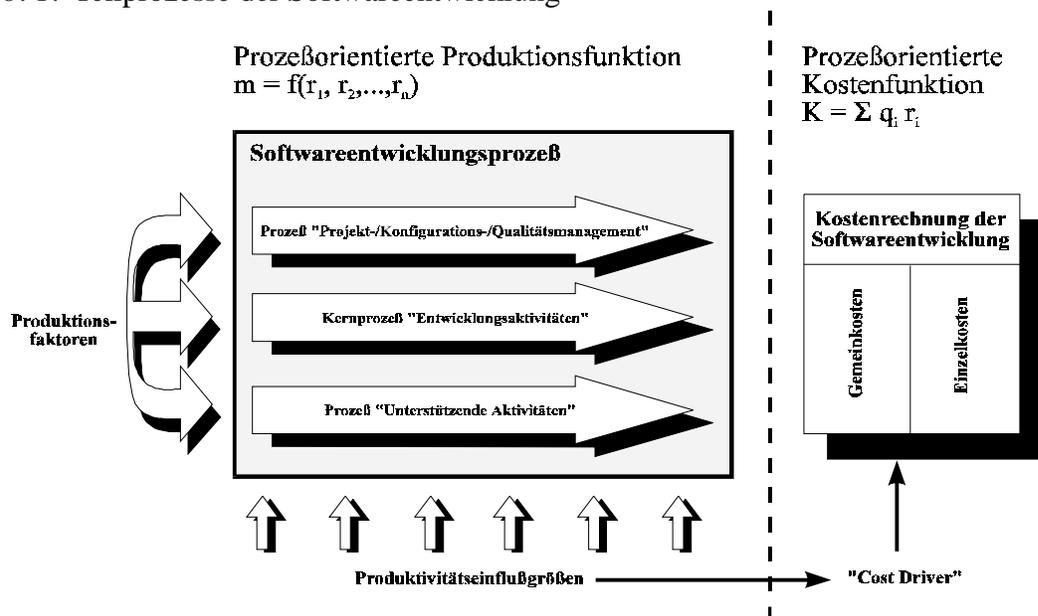
In diesem Beitrag geht es vorrangig um die Fragestellung, wie die Produktivität der Softwareunternehmen durch Konservierung und Aktivierung des organisatorischen Wissens über Erfolgsfaktoren von Softwareentwicklungsprozessen durch ein passives Informationssystem verbessert werden kann. Ausgangspunkt zur Lösung ist die Beschreibung des Entwicklungsprozesses und der auf ihn einwirkenden Produktivitätseinflußgrößen aus betriebswirtschaftlicher Sicht. Dem schließt sich der Vorschlag eines Lösungskonzepts an, wie er in Zusammenarbeit mit einem Softwareunternehmen (ca. 800 Mitarbeiter) als organisatorisches Gedächtnis entwickelt wurde.

3 Prozeßproduktivität und -kosten

Die Analyse der Produktivitätseinflußgrößen setzt bei der Beschreibung der Zusammenhänge im Prozeß der Softwareentwicklung auf. Wie Abb. 1. zeigt, sind prinzipiell drei Teilprozesse unterscheidbar: (1) Der Teilprozeß der Managementtätigkeiten beschreibt die Querschnittsaufgaben des Projekt-, Qualitäts- und Konfigurationsmanagements; (2) Ebenfalls eine Querschnittsaufgabe bilden die unterstützenden Tätigkeiten, wie z.B. Sekretariat, Beschaffung; (3) Bestandteile des zentralen Teilprozesses „Entwicklungsaktivitäten“ sind alle (technisch ausgeprägten) Tätigkeiten der Erstellung des Softwareprodukts, wie Entwurf, Programmierung und Test.

Analog zu industriellen Fertigungsprozessen fließen in den Entwicklungsprozeß verschiedene Produktionsfaktoren als Einsatzgüter ein. Charakteristisch ist dabei für Softwareunternehmen die Dominanz der menschlichen Arbeitskraft, die ca. 80% der Herstellkosten ausmacht. Die Wirtschaftlichkeit des Einsatzes der Produktionsfaktoren wird durch die Produktivitätseinflußgrößen determiniert. Auf der Produktionsseite liegt eine rein mengenmäßige Betrachtung vor, die analytisch in Form von Produktionsfunktionen beschrieben werden kann. Welche Produktionsfunktionen im Softwareentwicklungsprozeß relevant sind, ist bislang allerdings unbekannt, da die Zusammenhänge zwischen Produktionsfaktoren und Einflußgrößen auf der Produktionsseite noch weitestgehend unerforscht sind.⁷ Die wertmäßige Betrachtung dieser Zusammenhänge führt auf der Kostenseite zu analogen Kostenfunktionen. Wichtige Produktivitätseinflußgrößen können - insoweit eine Prozeßkostenrechnung zur Aufschlüsselung der Gemeinkosten gewünscht und sinnvoll ist - als „cost driver“ interpretiert werden.

Abb. 1: Teilprozesse der Softwareentwicklung



Legende: m - Produktmenge
 r_i - Einsatzmenge des Produktionsfaktors i
 K - Kosten des Produkts
 q_i - Preis des Produktionsfaktors i

Die Untersuchung der geschilderten Zusammenhänge, insbesondere die Ableitung geeigneter Produktionsfunktionen, setzt die Identifikation und Quantifizierung der relevanten Produktivitätseinflußgrößen voraus. Hierzu sind geeignete Kennzahlen zu entwickeln, die in Softwareunternehmen terminologisch zumeist als Softwaremaße bzw. -metriken bezeichnet werden. Im Kontext organisatorischen Lernens stellen sie Wissensartefakte dar. Ihrer Identifikation und Quantifizierung muß jedoch die begriffliche Strukturierung des interessierenden Realitätsausschnitts vorausgehen.

4 Wissensartefakte: Softwaremaße

Die Begriffsbildung ist ein Mittel, die Diskurswelt, also den zu untersuchenden Realitätsausschnitt, begrifflich zu strukturieren. Sie dient demnach der Ordnung der zunächst zusammenhangslos erscheinenden Eindrücke und fördert damit das organisatorische Lernen, indem sie Wissensartefakte schafft. „Am Anfang einer Wissenschaft oder der Erkundung eines einzelnen neuen Bereichs einer Wissenschaft steht daher der Versuch, diesen Ausschnitt der Realität...in Elemente zu zerlegen, die Objekte und Merkmale zu bezeichnen.“⁸ Dies trifft auch auf eine erste Annäherung an den Aufbau eines organisatorischen Gedächtnisses für die Wissensartefakte der Produktivität von Softwareunternehmen zu.

Sind die Probleme der Begriffsbildung gelöst, müssen Überlegungen über die Abbildbarkeit der begrifflich strukturierten Wissensobjekte der Diskurswelt angestellt werden. Im Falle der Produktivitätsanalyse sind hierbei v.a. die interessierenden Merkmale von Produkt und Prozeß auf geeigneten Skalen abzubilden. Dabei geht es nicht ausschließlich um die quantitativ-objektive Messung von Softwaremaßen. Gemäß Abb. 2 gibt es verschiedene Möglichkeiten der Messung. „Messen wird als Zuordnung von Zahlen zu Merkmalen (Eigenschaften und Relationen) nach Regeln definiert, den Meßobjekten werden Zahlen zugeordnet.“⁹

Abb. 2: Merkmals- und Skalenarten

Skalenart		Diversitätsskalen					
		Nominalskalen		Präferenzskalen			
				Ordinalskalen		Metrische Skalen	
		ohne	mit	ohne	mit	Intervallskalen	Verhältnisskalen
Merkmalsart		Indifferenz		Indifferenz			
		1	2	3	4	5	6
Klassifikatorisch	1	11	12				
Abstufbar	Komparativ	2	21	22	23	24	
	Quantitativ	3	31	32	33	34	35 36

Für die Messung von Meßobjekten wurden verschiedentlich Anforderungskataloge definiert. Dabei sind drei Anforderungen im Sinne von Gütekriterien von essentieller Bedeutung:

1. *Reliabilität* einer Messung ist dann gegeben, wenn die resultierenden Meßwerte bei unveränderten Rahmenbedingungen reproduzierbar sind.

2. *Validität* ist das Ausmaß, mit dem ein Meßwert diejenigen Eigenschaften, die er zu messen beansprucht, tatsächlich mißt.
3. *Starke Meßskala* bedeutet, daß gemäß Abb. 2 eine normale Zuordnung von Merkmalsarten und Skalen bei der Messung vorliegt (dick umrandete Diagonale der Abbildung).

Wie Chmielewicz zeigt, ist die Einhaltung der essentiellen Anforderungen nicht immer vollständig möglich, da hierbei ein sog. Dreiecksproblem besteht: Jeweils zwei Anforderungen lassen sich problemlos erreichen, allerdings nur zu Lasten der dritten Anforderung. Insbesondere bei einer möglichen metrischen Skalierung führt die Forderung nach einer starken Skala u.U. schnell zur Reduktion der Validität. Umgekehrt bedeutet der Wunsch nach einer reliablen und validen Messung oftmals den Verzicht auf eine metrische Messung. In solchen Fällen wird auf zusätzliche Informationen verzichtet, um die Gültigkeit und Zuverlässigkeit der Messung nicht zu gefährden. Ein Beispiel stellen die in der Softwaremetrie vielfach aufgestellten Komplexitätsmaße dar. Hier wird oftmals eine metrische Messung vorgenommen, um die Anforderung nach einer starken Meßskala zu erfüllen. Wie Zuse jedoch zeigt, kann in der Regel von einer Verhältnisskala als starker Meßskala nicht ausgegangen werden. Die meisten Maße lassen bestenfalls eine Messung und Interpretation im Sinne einer Intervallskalierung zu.¹⁰

Ziel muß es sein, Maße zu entwickeln, die den genannten essentiellen Anforderungen genügen: „A goal of software measurement is to define measures that are on the most sophisticated scale possible, given the constraints of the real world. However, we still have only crude empirical relations - and hence crude measurement scales - for attributes like software quality and productivity.”¹¹

Ein erster Schritt zur Lösung muß das Dreiecksproblem der Softwaremetrie beachten:

1. Die Maße müssen valide und reliabel sein, da sie die Basis von Managemententscheidungen darstellen;
2. Daraus folgt ferner: Eine starke Meßskala darf nicht zugunsten der beiden anderen Anforderungen gewählt werden, solange ihre Anwendbarkeit nicht empirisch nachgewiesen ist.

Es muß deshalb auf mögliche Informationen bei der Erfassung der Meßwerte zugunsten der Validität und Reliabilität verzichtet werden. Dies bedeutet konkret, daß insbesondere bei quantitativen Merkmalen auf eine Verhältnisskalierung in der Praxis dann zu verzichten ist, wenn der Nachweis ihrer Verwendbarkeit noch nicht erbracht wurde. In diesem Fall ist auf Maße mit nominaler, ordinaler oder Intervallskalierung auszuweichen.

Die Problematik der mangelnden Meßbarkeit und Quantifizierung von Softwareentwicklungsprozessen und -produkten bedingt die Notwendigkeit zum Aufbau organisatorischen Wissens zur Speicherung der vorhandenen Wissensartefakte, um einmal aufgetretene Probleme, gefundene Lösungen und gemachte Erfahrungen dem organisatorischen Gedächtnis systematisch zuführen zu können. Eine Projektdatenbank als passives Informations-

system stellt einen ersten sinnvollen Ansatz zur Förderung der notwendigen organisatorischen Lernprozesse dar.

5 Organisatorisches Gedächtnis: Projektdatenbank

5.1 Vorüberlegungen

Der praktische Aufbau einer Projektdatenbank als organisatorisches Gedächtnis zur Unterstützung organisatorischer Lernprozesse in Softwareunternehmen gestaltet sich äußerst schwierig. Dies liegt zum einen am noch recht geringen Alter der Softwareindustrie sowie am in der Vergangenheit mangelnden Kostenbewußtsein der Softwareunternehmen. Zum anderen spielt aber auch der immanente Dienstleistungscharakter eine Rolle, der eine systematische Speicherung von Wissensartefakten erschwert. Im zu beschreibenden Forschungsprojekt wurde daher ein dreistufiges Vorgehen gewählt:

1. Identifikation der potentiellen Produktivitätseinflußgrößen (PEG) im kooperierenden Softwareunternehmen und Modellierung der Wissensartefakte in einem Datenerfassungsstandard als Vorbereitung zum Aufbau des organisatorischen Gedächtnisses;
2. Modellierung des konzeptionellen Datenmodells der Wissensartefakte und
3. Architektur sowie Implementierung der Projektdatenbank als organisatorisches Gedächtnis.¹²

5.2 Modellierung der Wissensartefakte: Datenerfassungsstandard

Ausgangspunkt der Entwicklung einer Projektdatenbank war die inhaltliche Identifikation der relevanten Einflußgrößen auf die Produktivität des Entwicklungsprozesses beim kooperierenden Softwareunternehmen. Als Kreativitätstechnik zur Identifikation der potentiellen Produktivitätseinflußgrößen wurde eine Brainstormingsitzung gewählt. Diese Technik hat den Vorteil, daß strukturiert und schnell ein erster Überblick über das im Unternehmen vorhandene organisatorische Wissen gewonnen werden kann. An der Brainstormingsitzung nahmen, neben dem Entwicklungsteam des Lehrstuhls, die Fachpromotoren mit ihrem organisatorischen Wissen auf Unternehmensseite teil.

Als Ergebnis dieser Sitzung entstand ein zunächst noch recht unstrukturierter Überblick des vorhandenen organisatorischen Wissens über Produktivitätseinflußgrößen (vgl. Abb. 3).

Abb. 3: Überblick des vorhandenen organisatorischen Wissens (Ergebnis des Brainstormings)

Einflußgrößen auf die Produktivität							
Mitarbeiter-bezogene Daten	Organisations-bezogene Daten	Umfeld-bezogene Daten	Qualitäts-bezogene Daten	Projekt-steuerungs-bezogene Daten	Projekt-planungs-bezogene Daten	Umfangs-bezogene Daten	Kunden-bezogene Daten
Qualifikation Motivation Partizipation Erfahrung Führungsstil Konfliktregelung Kommunikation	Berichtswesen Projektcontrolling Phasenmodell Qualitätssicherung	Hilfsmittelverfügbarkeit Technische Umgebung Arbeitsumfeld	Terminreue Reviews Methoden Gewährleistung	Dämpfung DCR Change Management Problem Management Kostenkontrolle Restzeitsteuerung Pufferbildung	Projektplanung Qualitätsplanung Aufwandschätzung Ressourcenverfügbarkeit Nutzung von Erfahrungsdaten	Rüstzeiten Anzahl Mitarbeiter Komplexität Projektrisiko	Technische Umgebung Anwenderbeteiligung Kommunikation mit Kunde Erfahrung des Kunden

Dieser erste Überblick wurde nach mehreren Iterationen der Überarbeitung und Rückkopplung mit den Fachpromotoren in einem Datenerfassungsstandard, kurz: DES, institutionalisiert, um das so gewonnene organisatorische Gedächtnis allgemein zugänglich zu machen. Der DES präzisiert inhaltlich die als relevant angesehenen Einflußgrößen der Produktivität anhand eines standardisierten Schemas für die zu erfassenden Datenarten (im Sinne von Wissensartefakten).

Abb. 4: Beispiel des Datenerfassungsstandards (Teil I)

Daten-Nr.	Bezeichnung der Datenart
I.1.1	Projektkennung
Zu erfassende Daten	
<ul style="list-style-type: none"> • Kunde • Projektname • Versions-Nr. • Angebots-Nr./Projekt-Nr. (interne Nr.) • Projektart 	
Anmerkungen	
<ul style="list-style-type: none"> • Inhalt der Projektart: Projektgröße (klein/mittel/groß), Preisbindung, Gewährleistung • Darstellung als Tabelle 	
Literaturquellen	
vgl. Projekthandbuch	

Zur besseren Anwendbarkeit wurde der DES in zwei Teile untergliedert. Teil I (vgl. Beispiel der Abb. 4) enthält ausschließlich qualitative Projektübersichtsdaten in Form verbaler Beschreibungen. Ziel ist eine einheitliche Projektcharakterisierung, um schnell und ohne großen Aufwand Aufgaben und Ziele eines Projekts ermitteln zu können.

nen. Teil II (vgl. Beispiel der Abb. 5) enthält die detaillierten Projektdaten, welche zur Produktivitätsauswertung benötigt werden. Die Ermittlung dieser Daten erfolgt anhand von vorab im DES definierten Softwaremaßen. In der Praxis lassen sich leider zur Zeit nur sehr wenige quantitative Daten objektiv erfassen. Der DES schlägt in solchen Fällen zumeist eine geeignete Ratingskala vor, um zumindest das subjektiv vorhandene organisatorische Wissen aus den erfaßten Projekten speichern zu können.

Abb. 5: Beispiel des Datenerfassungsstandards (Teil II)

PEG-Nr.	PEG-Name	Bezug
II.2.2	Projektcontrolling	Organisation
Maß-Bezeichnung	Controllingaufwand	
Maß-Nr.	II.2.2.1	
Klassifikation	Produkt/Prozeß/Ressourcee	
Erfassung	subjektiv/objektiv	
	qualitativ/quantitativ	
Definition	Zeitaufwand (Arbeitsstunden) für Tätigkeiten des Projektcontrollings	
Anmerkungen	keine	
Literaturquellen	keine	

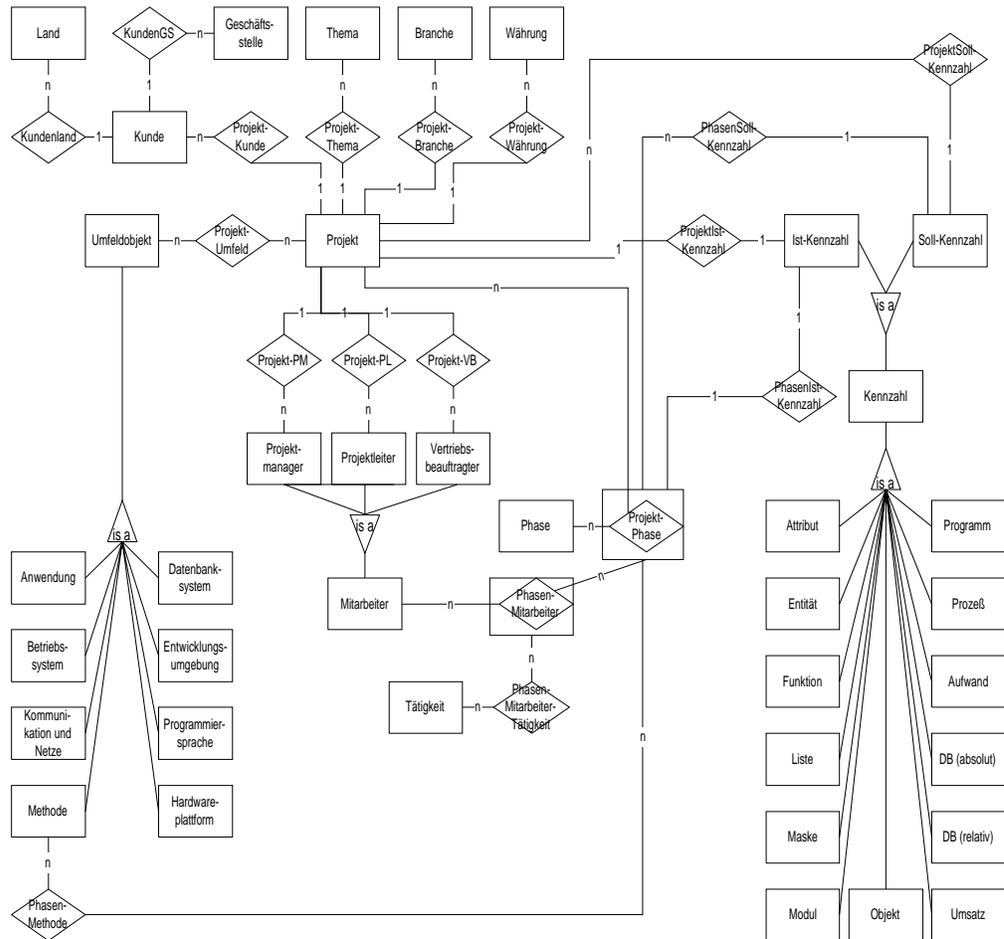
5.3 Modellierung des organisatorischen Gedächtnisses: Konzeptionelles Datenmodell der Projektdatenbank

Im folgenden wird die Entwicklung des konzeptionellen Datenmodells der Projektdatenbank als Entity Relationship-Modell (ERM) dargestellt. Dabei soll gezeigt werden, welche Überlegungen bei der Bildung einzelner Datenkonstruktionen für die Wissensartefakte eine Rolle spielen, die letztlich zum Gesamtmodell des organisatorischen Gedächtnisses führen. Die Entwicklung stützt sich auf den Datenerfassungsstandard und die zusätzlich definierten Anforderungen des Managements beim Kooperationspartner.¹³ Auf die Darstellung einzelner Attribute der Wissensartefakte wird aus Platzgründen verzichtet.

5.3.1 Projekt als Basis des ERM

Ausgangspunkt ist das aus betriebswirtschaftlicher Sicht im Vordergrund stehende Softwareprojekt, womit sich ein erster Entitytyp PROJEKT ergibt. Jedes Projekt hat einen thematischen (wie Benutzerservice, Client-Server-Projekt etc.) und einen branchenspezifischen Bezug (wie Handel, Banken etc.). Da Teile des betrachteten Unternehmens ihren Sitz im Ausland haben, müssen für die Modellierung von Zahlungsströmen der Projektleistungen die Währungsbezeichnungen berücksichtigt werden. Dies führt zur Einführung der Entitytypen Thema, Branche und Währung, die über die Beziehungstypen PROJEKT-THEMA, PROJEKTBRANCHE und PROJEKTWÄHRUNG mit PROJEKT verknüpft werden. Abb. 6 stellt die Beziehungen zwischen diesen Entitytypen dar.

Abb. 6: ERM der Projektdatenbank



5.3.2 Modellierung von Projektkunden

Ein Projekt hat einen Auftraggeber, den Kunden. Jeder Kunde wird bei einer Geschäftsstelle geführt und ist einem bestimmten Land zuordenbar. Der Entitytyp KUNDE steht dann mit den Entitytypen PROJEKT, GESCHÄFTSSTELLE und LAND über die Beziehungstypen PROJEKTKUNDE, KUNDENGs und KUNDENLAND in Beziehung (vgl. Abb. 6).

5.3.3 Modellierung von Mitarbeitern

In jedem Projekt arbeiten Mitarbeiter, die projektübergreifende Positionen einnehmen: Projektmanager, Projektleiter und Vertriebsbeauftragter. Daher wird der Entitytyp MITARBEITER eingeführt, in diese Positionen spezialisiert und die Teilklassen PROJEKTMANAGER, PROJEKTLLEITER und VERTRIEBSBEAUFTRAGTER mit PROJEKT in Beziehung gesetzt (vgl. Abb. 6).

5.3.4 Modellierung von Projektphasen

Jedes Projekt läßt sich ablauforganisatorisch in Phasen untergliedern. Daher werden die Entitytypen PROJEKT und PHASE zum Beziehungstypen PROJEKTPHASE aggregiert. Innerhalb des Phasenmodells müssen Rücksprünge zu einzelnen Phasen möglich sein, so daß eine Phase u.U. mehrfach zyklisch durchlaufen wird. Aus diesem Grund wird der Beziehungstyp PROJEKTPHASE in einen Entitytypen uminterpretiert.

In jeder Phase werden Mitarbeiter eingesetzt, die spezielle Tätigkeiten ausführen. Es würde sich daher anbieten, die Entitytypen PROJEKTPHASE, MITARBEITER und TÄTIGKEIT in einem Beziehungstypen zu aggregieren. Dies würde aber bedeuten, daß ein Mitarbeiter nur in Verbindung mit einer Tätigkeit einer Phase zuordenbar wäre. Da laut Anforderung auch unvollständige Projekte in die Projektdatenbank aufgenommen werden sollen, ist es jedoch denkbar, daß einzelnen Mitarbeitern noch gar keine Tätigkeit zugeordnet werden konnte. Deshalb wird MITARBEITER zu PROJEKTPHASE in Beziehung gesetzt, der Beziehungstyp PHASENMITARBEITER in einen Entitytypen uminterpretiert und dieser wiederum zu TÄTIGKEIT in Beziehung gesetzt.

Die jeweiligen Verrichtungen in den Phasen stützen sich auf entsprechende Methoden. Deshalb wird der Entitytyp METHODE eingeführt und über den Beziehungstypen PHASENMETHODE zu PROJEKTPHASE in Beziehung gesetzt. Damit wird ein vorhandener Zusammenhang zwischen den eingesetzten Methoden, den mit diesen verrichteten Tätigkeiten und den sie einsetzenden Mitarbeitern zwar nicht im Modell explizit erfaßt; dieses wurde aber auch nicht gefordert, da über die Tätigkeiten eines Mitarbeiters auf die eingesetzten Methoden implizit geschlossen werden kann.

5.3.5 Modellierung des Projektumfelds

Das Projektumfeld beschreibt die projektspezifischen technischen Gegebenheiten des Auftraggebers. Im DES wurden etwa das Betriebs- oder Datenbanksystem bzw. die verwendeten Methoden als Umfeldobjekttypen benannt. Im Rahmen der Modellierung des ERM wurde zudem festgestellt, daß die in den Phasen eingesetzten Methoden identisch mit denjenigen aus dem Projektumfeld sind. Zur Modellierung dieses Sachverhalts wird der Entitytyp UMFELDOBJEKT eingeführt. Dieser wird über den Beziehungstypen PROJEKTUMFELD mit PROJEKT verbunden und zum Entitytypen METHODE spezialisiert. Daneben sind noch weitere dieser Spezialisierung von UMFELDOBJEKT möglich, wie Abb. 6 zeigt.

5.3.6 Modellierung der Projektkennzahlen

Für jedes Projekt werden Kennzahlen erfaßt, die sich nach Klassen (z.B. Attribute, Funktionen, etc.) unterscheiden lassen. Die einzelnen Kennzahlen werden einem Projekt zugeordnet. Deshalb wird der Entitytyp KENNZAHL eingeführt, in die möglichen Teilklassen spezialisiert (vgl. Abb. 6) und mit PROJEKT zum Beziehungstypen PROJEKTKENNZAHLN aggregiert. Die einzelnen Kennzahlen lassen sich in Ist- und Sollwerte untertei-

len. Aus diesem Grund erfolgt die Spezialisierung von KENNZAHL in die beiden Teilklassen ISTKENNZAHL und SOLLKENNZAHL. Da sich immer nur ein Istwert je Kennzahl und abgeschlossenem Projekt ergibt, aber beliebig viele Sollwerte bestimmbar sind, werden die Beziehungstypen PROJEKTISTKENNZAHL und PROJEKTSOLLKENNZAHL eingeführt, die PROJEKT mit den korrespondierenden Teilklassen von KENNZAHL verbinden.

5.3.7 Modellierung von Phasenkenzzahlen

Wie auf der Projektebene so sind auch auf Phasenebene verschiedene Kennzahlen zu erfassen, die sich unter den gleichen Bedingungen wie auf der Projektebene in Ist- und Sollkennzahlen unterscheiden lassen. Deshalb werden die Entitytypen ISTKENNZAHL und SOLLKENNZAHL mit PROJEKTPHASE über die Beziehungstypen PHASENISTKENNZAHL und PHASENSOLLKENNZAHL verbunden.

Nach diesem Schritt ergibt sich das in Abb. 6 dargestellte vollständige konzeptionelle Datenmodell der Projektdatenbank, das in einem weiteren Schritt in ein Relationenmodell überführt wird.

5.4 Architektur des organisatorischen Gedächtnisses: Relationenmodell der Projektdatenbank

Im Relationenmodell erfolgt die Darstellung von realitäts- und vorstellungsorientierten Sachverhalten durch zweidimensionale Tabellen. Diese werden als Relationen bezeichnet. Die Spalten jeder Relation repräsentieren die Attribute des zu erfassenden Sachverhalts. Die Zeilen einer Relation stellen einzelne Ausprägungen des Sachverhalts dar und werden als Tupel bezeichnet. Damit besteht ein Tupel aus einer Reihe von Eigenschaftswerten. Die Reihenfolge der Attribute spielt ebensowenig eine Rolle wie die der Tupel, da der Zugriff auf die einzelnen Ausprägungen symbolisch über die Werte der Attribute erfolgt.

Unter Berücksichtigung der allgemeinen Regeln zur Umsetzung eines ERM in ein Relationenmodell werden nun die Entity- und Beziehungstypen des vorgestellten Modells der Projektdatenbank auf die erforderlichen Relationen bzw. Tabellen hin analysiert. Als Ausgangspunkt dient auch hier wiederum das Projekt.

Zwischen den Entitytypen PROJEKT und THEMA besteht eine Beziehung des Grades 1:n. Da die Teilnahme der Ausprägungen der 1-Seite obligatorisch ist, sind lediglich zwei Relationen erforderlich. Der Entityschlüssel von THEMA wird als Attribut in die Relation von PROJEKT aufgenommen (siehe unten). THEMA steht zu keinem anderen Entitytypen in Beziehung, so daß sich als erste Relation ergibt:

THEMA(ThemaID, ThemaName).

Analog erfolgt die Behandlung von BRANCHE und WÄHRUNG:

BRANCHE(BrancheID, BrancheName),

WÄHRUNG(WährungID, WährungsName).

Bei den Entitytypen KUNDE, LAND und GESCHÄFTSSTELLE stellt KUNDE die obligatorische 1-Seite und die beiden anderen Typen die nicht obligatorische n-Seite dar. Somit erfolgt auch hier die Umsetzung der einzelnen Beziehungen in nur zwei Relationen, wobei die Entitätschlüssel von LAND und GESCHÄFTSSTELLE als Attribute in die Relation von KUNDE eingehen:

LAND(LandID, LandKürzel, LandName),

GESCHÄFTSSTELLE(GeschäftsstelleID, GeschäftsstelleName),

KUNDE(KundeID, KundeKürzel, KundeFirma, KundeName, KundePLZ, KundeOrt, LandID, GeschäftsstelleID, Bemerkungen).

Zwischen UMFELDOBJEKT und PROJEKT besteht eine Beziehung des Grades n:m, die grundsätzlich in drei Relationen aufgelöst wird. Jeder beteiligte Entitytyp wird in eine Relation umgesetzt. Die dritte Relation nimmt die Beziehung zwischen den beiden Entitytypen auf und enthält unter anderem die Entitätschlüssel beider Entitytypen. Zu UMFELDOBJEKT existieren jedoch verschiedene Entitytypen als Spezialisierungen. Diese Teilklassen sind zueinander disjunkt, so daß sie problemlos in eigene Relationen abgebildet werden können. Dabei werden die Attribute der Oberklasse jeweils zu den Attributen der Teilklassen hinzugefügt. Damit die Beziehung zwischen PROJEKT und UMFELDOBJEKT nicht verlorenght, wird zudem der Beziehungstyp in so viele Relationen umgesetzt wie Teilklassen vorhanden sind. Dies wird beispielhaft an den sich ergebenden Relationen für den Entitytyp METHODE dargestellt:

METHODE(MethodeID, MethodeName),

PROJEKTMETHODE(ProjektID, MethodeID).

Die Beziehung des Grades n:m zwischen PROJEKT und PHASE wird in drei Relationen aufgelöst. Der Beziehungstyp zwischen den beiden Typen wurde im ERM zusätzlich noch in einen Entitytypen uminterpretiert, der unter anderem mit METHODE in Verbindung steht. Damit sind die folgenden Relationen festgelegt:

PHASE(PhaseID, PhaseName),

PHASENMETHODE(ProjektPhaseID, MethodeID).

Bei der Umsetzung von MITARBEITER auf Projektebene werden dessen Teilklassen analog zum Vorgehen beim Projektumfeld jeweils in eigene Relationen umgesetzt. Aufgrund der obligatorischen 1:n-Beziehungen zwischen PROJEKT und jeder Teilklassen von

MITARBEITER sollen die Entityschlüssel der Teilklassen als Attribute in die Relation von PROJEKT eingehen. Da diese aber alle gleich sind, werden in PROJEKT Synonyme verwendet, die sich aus der Beziehung der eingenommenen Position und dem für die Schlüssel gewählten Kürzel ID zusammensetzen (z.B. ProjektleiterID). Die Relation MITARBEITER ihrerseits sieht wie folgt aus:

MITARBEITER(MitarbeiterID, MitarbeiterKürzel, MitarbeiterName).

MITARBEITER ist an einer Beziehung des Grades n:m mit PROJEKTPHASE beteiligt. Dieser Beziehungstyp war in einen Entitytypen uminterpretiert worden und wird demzufolge jetzt in eine eigene Relation umgesetzt:

PHASENMITARBEITER(ProjektPhaseMitarbeiterID, ProjektPhaseID, MitarbeiterID).

Zwischen PHASENMITARBEITER und TÄTIGKEIT besteht eine Beziehung des Grades n:m. Daraus resultiert die Notwendigkeit von zwei weiteren Relationen:

TÄTIGKEIT(TätigkeitID, TätigkeitName),

PHASENMITARBEITERTÄTIGKEIT(ProjektPhaseMitarbeiterID, TätigkeitID).

Es bleibt noch die Umsetzung von KENNZAHLEN und ihren Teilklassen. Da sich zu jedem Projekt oder einem Teil davon immer nur eine Istkennzahl einer bestimmten Ausprägung ergeben kann, liegt damit für jede dieser speziellen Istkennzahlen eine obligatorische Beziehung des Grades 1:1 mit PROJEKT und PROJEKTPHASE vor. Aus diesem Grund werden sämtliche Ausprägungen von ISTKENNZAHLEN in Form von Attributen in die Relationen von PROJEKT und PROJEKTPHASE aufgenommen.

Für SOLLKENNZAHLEN ergibt sich für jede Ausprägung eine auf der 1-Seite obligatorische Beziehung des Grades 1:n zwischen PROJEKT und PROJEKTPHASE. Da jede Teilklassse von SOLLKENNZAHLEN ausschließlich zu PROJEKT oder PROJEKTPHASE in Beziehung steht, wird jede dieser Teilklassen in eine eigene Relation überführt, in die der Entityschlüssel der beteiligten Entitytypen aufgenommen wird. Aus den sich ergebenden Relationen wird beispielhaft eine dargestellt:

SOLLUMSATZ(UmsatzID, ProjektPhaseID, SollUmsatz, Datum).

Die im Rahmen der besprochenen Umsetzung entstandenen Relationen für PROJEKT und PROJEKTPHASE haben damit folgendes endgültiges Aussehen:

PROJEKT(ProjektID, ProjektKennung, ProjektBezeichnung, ProjektBeginn, ProjektEnde, ProjektKlassifikation, Auftragsart, Sonstiges, BrancheID, ThemaID, WährungID, KundeID, ProjektmanagerID, ProjektleiterID, VertriebsbeauftragterID, IstProjektAufwand, IstDeckungsbeitrag(absolut), IstDeckungsbeitrag(relativ),

IstProjektUmsatz, IstAttributanzahl, IstEntitätenanzahl, IstFunktionenanzahl,
IstListenanzahl, IstMaskenanzahl, IstModulanzahl, IstObjektanzahl,
IstProgrammanzahl, IstProzeßanzahl),

PROJEKTPHASE(ProjektPhaseID, ProjektID, PhaseID, PhaseStart, PhaseEnde,
IstPhaseAufwand, IstPhaseDB(absolut), IstPhaseDB(relativ), IstPhaseUmsatz).

5.5 Implementierung des organisatorischen Gedächtnisses: Prototypische Realisierung der Projektdatenbank

Das entwickelte Relationenmodell wurde abschließend mit Hilfe des relationalen Datenbankmanagementsystems Microsoft Access und der Programmiersprache Visual Basic for Applications auf einen PC überführt. Abb. 7 gibt einen Eindruck des entstandenen Informationssystems und zeigt die Projektsicht mit Eckdaten für ein selektiertes Projekt, basierend auf dem Datenerfassungsstandard.

Abb. 7: Projektsicht der Projektdatenbank

Selektiertes Projekt	
Kennung:	Projekt 01
Branche:	Banken
Bezeichnung:	Bezeichnung zu Projekt 01
Thema:	Benutzerservice
Klassifikation:	Klassifikation zu Projekt 01
Auftragsart:	Auftragsart 1
Kunde:	K1 Firma 01
Aufwand:	3.000,00 Stunden
Beginn:	15.06.1989
Vertriebsbeauftragter:	HUP
Umsatz:	3.000.000,00 DM
Ende:	19.11.1989
Projektmanager:	BGM
DB (absolut):	150.000,00 DM
DB (relativ):	5,00 %
Projektleiter:	KA

Auswählbare Projekte	
Projekt 01	15.06.1989: Bezeichnung zu Projekt 01
Projekt 02	02.05.1990: Bezeichnung zu Projekt 02
Projekt 03	03.03.1990: Bezeichnung zu Projekt 03
Projekt 04	03.03.1991: Bezeichnung zu Projekt 04
Projekt 05	07.05.1991: Bezeichnung zu Projekt 05
Projekt 06	12.02.1992: Bezeichnung zu Projekt 06
Projekt 07	05.09.1992: Bezeichnung zu Projekt 07
Projekt 08	22.04.1993: Bezeichnung zu Projekt 08
Projekt 09	05.01.1994: Bezeichnung zu Projekt 09
Projekt 10	15.11.1994: Bezeichnung zu Projekt 10

Neben den Eckdaten können noch spezifische Informationen über Kennzahlen, Daten des Kunden, der Projektphasen sowie des Projektumfelds gespeichert und abgerufen werden. Darüber hinaus ist auch eine Selektion ähnlicher Projekte anhand der im Relationenmodell definierten Attribute der einzelnen Relationen möglich. Diese Attribute sind als Suchanfrage (Filter) frei kombinierbar. Die Suchanfrage kann zudem gespeichert und wiederholt aufgerufen werden, so daß der Projektmanager die Möglichkeit hat, spezifizierte Suchanfragen nach ihm interessierenden Fällen zu speichern. Dies gestattet eine einfache individuelle Auswertung des gespeicherten organisatorischen Wissens (vgl. Abb. 8).

Abb. 8: Filter für eine Suchanfrage in der Projektdatenbank



6 Resümee

Die Entwicklung der Projektdatenbank als organisatorisches Gedächtnis in Softwareunternehmen wurde im beschriebenen Forschungsprojekt prototypisch demonstriert. Sie ist damit jedoch keineswegs abgeschlossen:

- So finden sich z.Z. im Datenmodell lediglich interessierende Entitytypen für produktivitätsbeeinflussende Merkmale des technischen Entwicklungsprozesses. Das Datenmodell bildet damit sehr gut die produktionstheoretische Sicht für Produktionsfaktoren und Produktivitätseinflußgrößen gemäß Abb. 1 ab. Für eine kostentheoretische Analyse sind jedoch im Datenmodell noch keine Entitytypen vorgesehen. Die für das Projektcontrolling und die Unternehmensleitung wichtigen kostenrechnerischen Informationen können im entwickelten Prototypen nicht gespeichert bzw. aktiv genutzt werden.
- Die Funktionalität des Prototypen ist bislang lediglich ansatzweise in einem Funktionenmodell modelliert und in der Projektdatenbank realisiert. Sie beschränkt sich z.Z. auf einige wenige graphische Darstellungen interessierender Softwaremaße. Eine über die geschilderte Filterfunktion (vgl. Abb. 8) hinausgehende flexible Informationsaufbereitung ist nicht implementiert. So sind keine Aggregation, keine Verknüpfung, kein Data Mining, etc. möglich. Überlegenswert ist auch der Einsatz deduktiver Datenbankmanagementsysteme und Neuronaler Netze für die Weiterentwicklung zu einem aktiven Informationssystem.
- Die Weiterentwicklung sollte zudem multimediale Potentiale berücksichtigen, deren Nutzung sich unter den gegebenen Rahmenbedingungen anbietet. So wäre eine Kommunikation mittels Notebook und Mobiltelefon für viele Projektmanager eine sinnvolle Möglichkeit der Speicherung und Aktivierung organisatorischen Wissens, da Softwareprojekte oftmals beim Kunden vor Ort abgewickelt werden.

Im kooperierenden Softwareunternehmen konnte anhand des Prototypen eine Sensibilisierung für die Notwendigkeit des Aufbaus eines organisatorischen Gedächtnisses erreicht werden. Die Notwendigkeit zur Schaffung einer organisatorischen Wissensbasis war bis dahin nicht sehr ausgeprägt und stieß auf Widerstand. Dies lag zum einen im persönlichen Aufwand für die systematische Erfassung von Daten begründet. Zum anderen fühlen sich aber gerade im Bereich der Softwareentwicklung viele Manager bzw. Mitarbeiter durch den Zwang zur Erfassung und Speicherung ihres Wissens eingeengt. Dieser Widerstand konnte im Verlauf der Entwicklung des Prototypen spürbar vermindert werden, so daß nunmehr dazu übergegangen wurde, in ausgewählten Pilotprojekten mit der systematischen Datenerfassung zur Unterstützung des organisatorischen Lernprozesses auf Basis des vorgestellten Forschungsprojekts zu beginnen.

Anmerkungen

Am beschriebenen Forschungsprojekt waren außer den Autoren noch Dipl.-Kfm. Erik Redl, Dipl.-Kfm. Jörg Heitzenröther und Dipl.-Kfm. Peter Hunkirchen beteiligt. Wir danken ihnen sehr herzlich für ihre Mithilfe.

1. Weltz/Ortmann (1992), S. 162.
2. Weltz/Ortmann (1992), S. 122.
3. Vgl. die Monographie von Brooks (1975).
4. Probst/Büchel (1994), S. 17.
5. Oberschulte (1994), S. 62.
6. Oberschulte (1994), S. 213.
7. Zur prozeßorientierten Produktionstheorie im Kontext organisatorischen Lernens siehe Bea (1995), S. 35ff.
8. Friedrichs (1990), S. 73.
9. Chmielewicz (1994), S. 67.
10. Vgl. die Monographie von Zuse (1991).
11. Fenton/Pfleeger/Glass (1994), S. 88.
12. Zu Projektdatenmodellen siehe auch Kindler (1995), S. 157ff.
13. Dies kann beispielsweise dazu führen, daß Größen, die sich zwar auch auf Phasenebene identifizieren lassen (etwas bestimmte Kennzahlen), ausschließlich auf Projektebene abgebildet werden, weil nur die Erfassung auf dieser Ebene aus betriebswirtschaftlich-fachlicher Sicht vom Management gewünscht wurde.

Literatur

- Bächle, M. (1996): Qualitätsmanagement der Softwareentwicklung. Das QEG-Verfahren als Instrument des Total Quality Managements, Wiesbaden.
- Bea, F.X. (1995): Prozeßorientierte Produktionstheorie und Lernen, in: ZfB-Ergänzungsheft 3/95, S. 35-46.
- Brodbeck, F.C.; Frese, M. (Hrsg., 1994): Produktivität und Qualität in Software-Projekten. Psychologische Analyse und Optimierung von Arbeitsprozessen in der Software-Entwicklung, München, Wien.
- Brooks, F.P. (1975): The Mythical Man-Month, Reading (Mass.) et al.
- Friedrichs, J. (1990): Methoden der empirischen Sozialforschung, 14. Aufl., Opladen.
- Chmielewicz, K. (1994): Forschungskonzeptionen der Wirtschaftswissenschaft, 3., unveränd. Aufl., Stuttgart.
- Fenton, N.; Pfleeger, L.; Glass, R. (1994): Science and Substance. A Challenge to Software Engineers, in: IEEE Software (1994), H. July, S. 85-95.
- Kindler, A. (1995): Wirtschaftlichkeit von Software-Entwicklungsprojekten. Ansätze zur Verbesserung der Aufwandschätzung, Wiesbaden.
- Oberschulte, H. (1994): Organisatorische Intelligenz. Ein integrativer Ansatz des organisatorischen Lernens, München, Mering.
- Pasch, J. (1994): Software-Entwicklung im Team. Mehr Qualität durch das dialogische Prinzip bei der Projektarbeit. Berlin et al.
- Probst, G.J.; Büchel, B.S. (1994): Organisationales Lernen. Wettbewerbsvorteil der Zukunft, Wiesbaden.
- Schweiggert, F. (Hrsg., 1992): Wirtschaftlichkeit von Software-Entwicklung und -Einsatz. Investitionssicherung, Produktivität, Qualität, Stuttgart.
- Weltz, F.; Ortman, R.G. (1992): Das Softwareprojekt. Projektmanagement in der Praxis, Frankfurt.
- Zuse, K. (1991): Software Complexity. Measures and Methods, Berlin, New York.

Bisher erschienene Arbeitsberichte

1990

Band 1 *Jahnke, Bernd*: Konzeption und Entwicklung eines Führungsinformationssystems.

(erschienen in: *Bartmann, D.* (Hrsg.): Lösungsansätze der Wirtschaftsinformatik im Lichte der praktischen Bewährung, Berlin/Heidelberg/New York 1991, S. 39-65)

Band 2 *Wallau, Siegfried*: Akzeptanz betrieblicher Informationssysteme - eine empirische Untersuchung.

1991

Band 3 *Jahnke, Bernd*: Informationsverarbeitungs-Controlling, Konzepte - Inhalte - Methoden.

(erschienen in: *Huch, Burkhard/Behme, Wolfgang/Schimmelpfeng, Katja* (Hrsg.): EDV-gestützte Controlling-Praxis: Anwendungen in der Wirtschaft, Frankfurt 1992, S. 119-143, Vorabveröffentlichung in der FAZ - Blick durch die Wirtschaft, 3.3.92, S. 7)

Band 4 *Fehling, Georg/Groffmann, Hans-Dieter/Jahnke, Bernd*: Entwicklung der Benutzerschnittstelle eines computergestützten Informationssystems im Rahmen des SAA-CUA Konzepts - Dargestellt am Beispiel eines Führungsinformationssystems für die Württembergische Gebäudebrandversicherung.

1992

Band 5 *Groffmann, Hans-Dieter*: Kennzahlenmodell (KDM) als Grundlage aktiver Führungsinformationssysteme.

(erschienen in: *Rau, Karl-Heinz/Stickel, Eberhard* (Hrsg.): Daten- und Funktionsmodellierung. Erfahrungen - Konzepte - Perspektiven, Wiesbaden 1992, S. 1-29)

Band 6 *Jahnke, Bernd*: Einsatzkriterien, kritische Erfolgsfaktoren und Einführungsstrategien für Führungsinformationssysteme.

(erschienen in: *Behme, Wolfgang/Schimmelpfeng, Katja* (Hrsg.): Führungsinformationssysteme. Neue Entwicklungstendenzen im EDV-gestützten Berichtswesen, Wiesbaden 1993, S. 29-43)

Band 7 *Jahnke, Bernd/Bächle, Michael*: Produktivität im Softwareentwicklungsprozeß, Problematik und Einflußgrößen.

1993

Band 8 *Jahnke, Bernd*: Entscheidungsunterstützung der oberen Führungsebene durch Führungsinformationssysteme.

(erschienen in: *Preßmar, D.B.* (Hrsg.): Informationsmanagement, Band 49 der Schriften zur Unternehmensführung, Wiesbaden 1993, S. 123-147)

Band 9 *Jahnke, Bernd/Groffmann, Hans-Dieter: Führungsinformationssysteme zwischen Anspruch und Realisierbarkeit.*

1994

Band 10 *Jahnke, Bernd/Bächle, Michael/Simoneit, Monika: Methodische Analyse von Vertriebsprozessen zur Zertifizierungsvorbereitung nach ISO 9004.*

(In leicht gekürzter Form erschienen in: Heilmann, Heidi et al. (Hrsg.): Handbuch der modernen Datenverarbeitung, Heft 175, Januar 1994, S. 50-60.

Eine englische Fassung des Arbeitsberichts ist erschienen in: International Journal of Quality and Reliability Management, Quality improvements in manufacturing and service industries: recent trends and perspectives, Vol. 12, No. 9 (1995), pp. 76-99)

Band 11 *Jahnke, Bernd/Tjiok, Clifford: Business Process Reengineering and Software Systems Strategy.*

1995

Band 12 *Bächle, Michael/Jahnke, Bernd/Kindler, Achim: Aufwandschätzung und Produktivität in der Softwareentwicklung. Probleme und Problemlösungsansätze.*

Band 13 *Groffmann, Hans-Dieter/Jahnke, Bernd/Kruppa, Stephan: Information Broker: Kooperative Führungsinformationssysteme in der Finanzwirtschaft.*

1996

Band 14 *Bächle, Michael: Anforderungen an das Qualitätsmanagement der Softwareentwicklung. Produkt- und Prozeßnormen.*

Band 15 *Bächle, Michael/Jahnke, Bernd: Unterstützung organisatorischen Lernens in Softwareunternehmen durch Projektdatenbanken.*