



Thema:

**Datenintegration in der Qualitätssicherung eines Industriebetriebes durch
Erweiterung eines ERP-Systems**

Studienarbeit

Arbeitsgruppe Wirtschaftsinformatik

Themensteller: Prof. Dr. Hans-Knud Arndt

Betreuer: Dipl. -Wirtsch. -Inf. Stefan Breitenfeld

Vorgelegt von: Lars Dankworth

Abgabetermin: 19.08.05

Inhaltsverzeichnis

Inhaltsverzeichnis	II
Verzeichnis der Abkürzungen und Akronyme	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
1 Motivation für die Datenintegration in der Qualitätssicherung	1
2 Qualitätssicherung in einem Industriebetrieb	2
2.1 Der Qualitätsbegriff	2
2.2 Qualitätssicherung als Teil des Qualitätsmanagements	3
2.3 Qualität als Wettbewerbsfaktor	4
3 Datenintegration	7
3.1 Daten, Information und Wissen	7
3.2 Bedeutung von Information und Wissen im Wettbewerb	8
3.3 Integration	8
3.4 Enterprise Resource Planning Systeme als Basis für die Datenintegration	11
4 Entwicklung einer Software zur teilintegrierten Erweiterung des ERP-Systems	14
4.1 Vorbetrachtung	14
4.2 Problembeschreibung	17
4.3 Anforderungsanalyse	18
4.4 Spezifikation und Entwurf	19
4.5 Implementierung und Test	20
4.6 Erprobung	21
4.7 Einführung und Nutzung	22
5 Ergebnisse und Auswertung	23
5.1 Ergebnisbetrachtung	23
5.2 Kritik und Ausblick	24
A Prozessbeschreibung der Bearbeitung eines Non Conformance Reports	26
Literaturverzeichnis	27

Verzeichnis der Abkürzungen und Akronyme

eEPK	Erweiterte ereignisgesteuerte Prozesskette
ERP	Enterprise Resource Planning
ISO	International Standardization Organization
JIT	Just In Time
MRB	Material Review Board
NCR	Non Conformance Report
PIMS	Profit Impact of Market Strategies
ROI	Return On Investment
SGE	Strategische Geschäftseinheit
UML	Unified Modeling Language

Abbildungsverzeichnis

Abb. 2.1: Zusammenhang zwischen ROI und relativer Qualität	5
Abb. 3.1: Begriffsabgrenzung Zeichen – Daten – Information – Wissen	7
Abb. 3.2: Funktionsintegration	10
Abb. 3.3: Architekturschema der R/3-Anwendungen.....	12
Abb. 4.1: Datenbank-Beziehung ERP-System – Framework.....	14
Abb. A.1: Vereinfachte Prozessbeschreibung ‚Non Conformance Report‘	26

Tabellenverzeichnis

Tab. 3.1: Aspekte der Integration	9
--	---

1 Motivation für die Datenintegration in der Qualitätssicherung

Dem Qualitätsmanagement¹ mit dem Teilbereich Qualitätssicherung kommt als Querschnittsfunktion im Unternehmen im Zuge zunehmenden Wettbewerbsdrucks eine immer größere Bedeutung zu. Umso wichtiger ist es daher, die Gewinnung, Verarbeitung, Verwaltung und Speicherung anfallender Daten wie Reklamationen, Messreihen, organisatorische Daten etc. in einer Form vorzunehmen, die es erlaubt, diese effizient und effektiv zu handhaben und als Grundlage für die Umwandlung in Informationen und Wissen bereitzustellen.

In diesem Unternehmen erfolgte die Erfassung und Verarbeitung relevanter Daten in der Qualitätssicherung im Teilbereich Materialprüfung bisher mit einer Vielzahl von Hilfsmitteln wie Anwendungen für Tabellenkalkulation und Textverarbeitung, Papierformularen und mündlichen Abreden. Die Vielzahl der heterogenen Medien zur Datenverarbeitung zeigt, dass der Bedarf nach einer Lösung bestand, um die Datenverarbeitung mit informationstechnologischen Mitteln zu unterstützen. Des Weiteren sollte eine enge Verzahnung mit dem bestehenden Enterprise Resource Planning System (ERP-System) erreicht werden, um im Sinne einer integrierten Lösung u. a. die Anzahl von Medienbrüchen zu verringern und die Datenintegration zu erhöhen.

Ziel dieser Arbeit ist es, die theoretischen Grundlagen der Erweiterung eines ERP-Systems zu erarbeiten und die konkrete Realisierung in einem Unternehmen darzustellen. Dazu werden zunächst die theoretischen Grundlagen für eine Datenintegration durch Erweiterung eines bestehenden ERP-Systems erläutert, sowie die allgemeine Qualitätssicherung beschrieben. Dem schließt sich die Entwicklungsbeschreibung der Software zur Erweiterung des ERP-Systems an. Die Arbeit schließt mit einer kritischen Auswertung der erhaltenen Ergebnisse und einem Ausblick ab.

¹ Definition und Erläuterung aller im Motivationsteil benutzter Fachtermini erfolgt im Hauptteil der Arbeit.

2 Qualitätssicherung in einem Industriebetrieb

Der Begriff Qualität ist im täglichen Sprachgebrauch mit vielen Bedeutungen belegt. Die Spanne reicht dabei von Verständnissen, die aus der Wortbedeutung des lateinischen Begriffes „qualitas“ (Eigenschaft) abgeleitet sind, bis hin zum Begriff des Qualitätserzeugnisses, das marktübliche Anforderungen erfüllt bzw. übererfüllt (vgl. Masing (1999a), S. 3). Um für die vorliegende Arbeit ein einheitliches Verständnis vom Begriff der Qualität zu schaffen, erfolgt zunächst eine Definition. Daran schließt sich eine Betrachtung der wichtigsten Aspekte des Qualitätsbegriffes an.

2.1 Der Qualitätsbegriff

In der Diskussion um den Qualitätsbegriff lassen sich im Allgemeinen zwei Parteien unterscheiden, welche die Qualität eines Produktes² bewerten: Lieferant und Kunde. Diese können sowohl extern als Verkäufer und Käufer, als auch intern im Sinne eines verrechnungsorientiert organisierten Unternehmens auftreten. Sie haben z. T. gegensätzliche Interessen bzgl. des Produktes. „Der Kunde hat Wünsche [...]. Er artikuliert sie dem Hersteller gegenüber als Qualitätsforderungen oder setzt sie in Form von Erwartungen voraus. [...] Der Lieferant (Hersteller/Anbieter) hat ein Produkt mit einer bestimmten Beschaffenheit. Sie ist die wertfreie (!) Gesamtheit aller Merkmale und Eigenschaften des Produkts.“ (Masing (1999a), S. 3 f., Hervorhebung im Original). Ein analoges Verständnis ist in der ISO 9000:2000 zu finden, die Qualität definiert als „Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt.“ (ISO 9000:2000, S. 18). Merkmale werden als „kennzeichnende Eigenschaft“ (ISO 9000:2000, S. 25) und Anforderungen als „Erfordernis oder Erwartung, das oder die festgelegt, üblicherweise vorausgesetzt oder verpflichtend ist.“ (ISO 9000:2000, S. 19) definiert. Diese Definition geht über die von JURAN eingebrachte „Gebrauchstüchtigkeit“ („fitness for use“) hinaus, indem es die spezifischen Anforderungen berücksichtigt, die ein Kunde hat (vgl. Juran (1993), S. 20 ff.). Ein Produkt kann objektiv ‚gebrauchstüchtig‘ sein, vom Kunden dennoch als minderwertig betrachtet werden, weil es subjektive Erwartungen nicht erfüllt. Vgl. hierzu auch Abschnitt 2.3.

Somit wird für diese Arbeit definiert, dass *Qualität* eine vom Kunden wahrgenommene Beschaffenheit bzw. den subjektiven Nutzwert eines Produktes darstellt (vgl. Masing (1999a), S. 6 f.).

² Aus Gründen der Einfachheit werden im gesamten Verlauf dieser Arbeit unter dem Begriff ‚Produkt‘ materielle und auch immaterielle Güter subsumiert.

2.2 Qualitätssicherung als Teil des Qualitätsmanagements

Qualitätssicherung stand lange Zeit als Oberbegriff für das gesamte Qualitätswesen eines Unternehmens. Seit der Einführung der internationalen Normenreihe ISO 9000 wird stattdessen der Begriff Qualitätsmanagement verwendet. Die Qualitätssicherung ist nunmehr definiert als „Teil des Qualitätsmanagements, der auf das Erzeugen von Vertrauen darauf gerichtet ist, dass Qualitätsanforderungen erfüllt werden.“ (ISO 9000:2000, S. 21).

Die vorliegende Arbeit betrachtet den Teilbereich Materialprüfung in der Abteilung Qualitätssicherung. Aus historischen Gründen wird die organisatorische Einheit, die für das Qualitätswesen im hier betrachteten Unternehmen verantwortlich ist, als ‚Quality Assurance‘ bezeichnet. Obwohl diese Funktion nun als Bestandteil eines umfassenden Qualitätsmanagements verstanden wird, soll dieser Term für die betreffende organisatorische Einheit hier beibehalten werden.

Erste Systeme des Qualitätswesens bestanden aus einer Qualitätskontrolle, die lediglich ein Prüfen und Aussortieren fehlerhafter Teile nach der Produktion zur Aufgabe hatte (vgl. Timischl (1995), S. 2 ff; Masing (1999a), S. 5 ff.). Dies hatte den Nachteil, dass durch Kontrollen fertiger Produkte keine aktive Fehlervermeidung möglich war. Daher setzte sich eine mit statistischen Methoden unterstützte Qualitätssicherung durch. Sie befasste sich u. a. mit der Vorbeugung und Verhütung von Fehlern und dem Ausweiten des Qualitätswesens auf andere Unternehmensbereiche (vgl. Seghezzi (2003), S. 9 f., Masing (1999a), S. 5 ff.). Mit dem Aufkommen betriebswirtschaftlicher Konzepte wie dem Just In Time (JIT), der Internationalisierung des Handels und der zunehmenden Konkurrenz aus Asien stieg die Bedeutung von ganzheitlichen Konzepten im Qualitätswesen. Die Beherrschung der gesamten Prozesskette vom Lieferanten bis zum Kunden steht dabei im Mittelpunkt (vgl. Seghezzi (2003), S. 9 f). Aus der Qualitätssicherung entstand das *Qualitätsmanagement*. Es ist definiert als das Leitungshandeln in Bezug auf Qualität (vgl. ISO 9000:2000, S. 21). Nach ISO 9000:2000 gehören dazu „das Festlegen der Qualitätspolitik und der Qualitätsziele, die Qualitätsplanung, die Qualitätslenkung, die Qualitätssicherung und die Qualitätsverbesserung“ (ISO 9000:2000, S. 21). Es ist somit die umfassende Verankerung des Qualitätsgedankens auf allen aufbau- und ablauforganisatorischen Ebenen. Für Details zu den verbleibenden Aspekten des Qualitätsmanagements wird auf die entsprechende Literatur verwiesen, z. B. SEGHEZZI (2003), MASING (1999b).

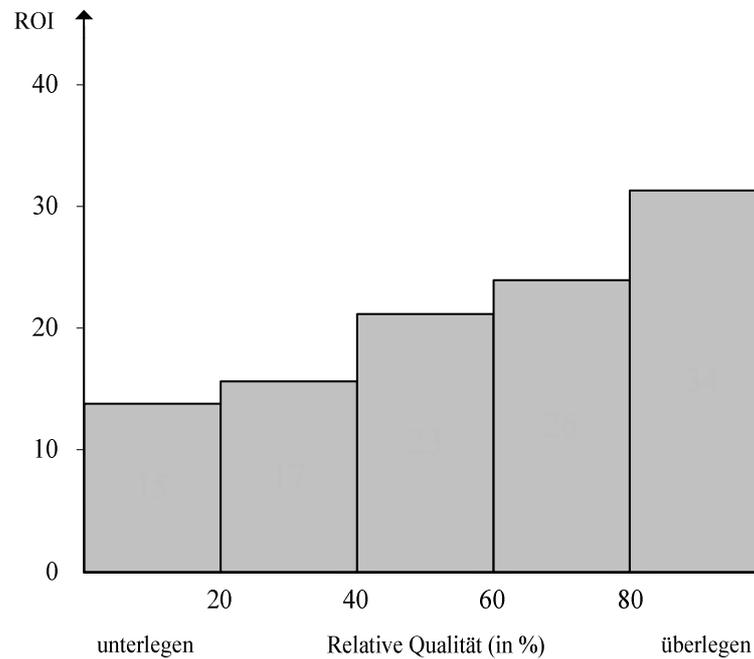
2.3 Qualität als Wettbewerbsfaktor

Unternehmen des produzierenden Gewerbes und der Dienstleistungsbranche sind seit Jahren einem immer stärker werdenden Wettbewerb um die Qualität ausgesetzt (vgl. Pfeifer (1996), S. 2 ff.). Diese Entwicklung begann mit der Expansion der asiatischen Konsumgüterindustrie, v. a. der japanischen, auf europäische und nordamerikanische Märkte. Ihr gelang es, beginnend in den 1960er und 1970er Jahren, mit qualitativ hochwertigen Produkten den etablierten Industrieunternehmen Marktanteile abzunehmen (vgl. Pfeifer (1996), S. 2 ff.; Juran (1993), S. 12 f., Seghezzi (2003), S. 17). Qualitativ hochwertig heißt in diesem Zusammenhang nicht, dass die Produkte der europäischen und nordamerikanischen Hersteller unzuverlässig oder gebrauchsuntauglich waren. Vielmehr war der Nutzwert der asiatischen Produkte auf Grund der technischen Eigenschaften und Ausstattungen höher (vgl. Pfeifer (1996), S. 3 ff.). Hier lag der Vorteil der asiatischen Hersteller, die sich konsequent am Kunden und seinen Wünschen und Erwartungen orientierten, anstatt lediglich fehlerfreie und gebrauchstaugliche Produkte zu produzieren. Dies korrespondiert mit dem in Abschnitt 2.1 hergeleiteten Qualitätsbegriff.

Dass Qualität einen hohen Einfluss auf die Marktchancen eines Produktes hat und somit ein wichtiger Wettbewerbsfaktor ist, wurde durch eine in den 1970er Jahren begonnene Studie empirisch bestätigt. In der so genannten PIMS-Studie (Profit Impact of Market Strategies) wurden strategisch relevante Merkmale von ca. 450 Unternehmen mit nahezu 3000 strategischen Geschäftseinheiten (SGE) erfasst (vgl. Buzzel/Gale (1989), S. 3). Ziel war es, den Einfluss einzelner Bestimmungsfaktoren wie zum Beispiel Marktanteil³, relative Produktqualität und Investmentintensität auf den Erfolg der jeweiligen SGE zu untersuchen. Einer der ermittelten Zusammenhänge ist der, dass die Qualität der Produkte und Dienstleistungen „auf lange Sicht [...] der wichtigste Einzelfaktor [ist], der den Erfolg einer Geschäftseinheit bestimmt“ (Buzzel/Gale (1989), S. 7 f., S. 91 ff.). Abb. 2.1 verdeutlicht nochmals den Zusammenhang von Produktqualität und Unternehmenserfolg.

An der PIMS-Studie wurde mehrfach Kritik geäußert. Zum Beispiel wurde bemängelt, dass der Erfolg der SGE hauptsächlich am Return On Investment (ROI) gemessen wird oder dass einige Variablen wie die Produktqualität aus subjektiven und zum Teil schwer quantifizierbaren Größen bestehen. Trotz dieser Kritik soll die PIMS-Studie in dieser Arbeit als hinreichender Beleg für den Einfluss der Qualität auf den Unternehmenserfolg genügen (vgl. Meffert (1994), S. 66 ff.).

³ Zur Definition zentraler Begriffe und Variablen vgl. Strategic Planning Institute (1979); Strategic Planning Institute (1984); Meffert (1994), S. 75



Quelle: nach Buzzel/Gale (1989), S. 93

Abb. 2.1: Zusammenhang zwischen ROI und relativer Qualität

Das hier betrachtete Unternehmen befindet sich in einem Marktumfeld, in dem ein starker Preisdruck auf die Produzenten wirkt. Verhandlungen und Angebote werden zum großen Teil auf Basis des Preises geführt und entschieden. [REDACTED]

[REDACTED]

Diese Maßnahme reicht jedoch nicht aus, um im Wettbewerb der von den Kunden bevorzugte Produzent zu sein, da es durch die Vorarbeiten in Deutschland und die damit verbundenen, vergleichsweise hohen Personalkosten nicht gelingt, der Preisführer am Markt zu sein. Aus diesem Grund hat man sich gegen eine rein preis- und für eine qualitätsorientierte Wettbewerbsstrategie entschieden, weil eine überlegene Qualität kurz- als auch langfristig zu höheren Gewinnen führt und den Unternehmensbestand sichern kann (vgl. Buzzel/Gale (1989), S. 7). Ein erster Schritt dorthin ist die Zertifizierung nach der Norm ISO 9001:2000. Der Qualitätsgedanke wird aktiv gelebt, indem z. B. Selbstverpflichtungen über die Forderungen der ISO-Norm hinaus auferlegt, die Mitarbeiter ständig in Qualitätsbelangen geschult werden und interne Qualitätsaudits stattfinden. Außerdem

werden den Kunden zusätzliche Dienste angeboten, welche die gesamte, vom Kunden wahrgenommene Produktqualität steigern sollen. Dazu zählen z. B. Lagerhäuser in Kundennähe, um einfacher auf deren Anforderungen, z. B. JIT-Lieferungen, reagieren zu können und Web-Portale, über welche die Kunden die gewünschten Waren konfigurieren und bestellen können.

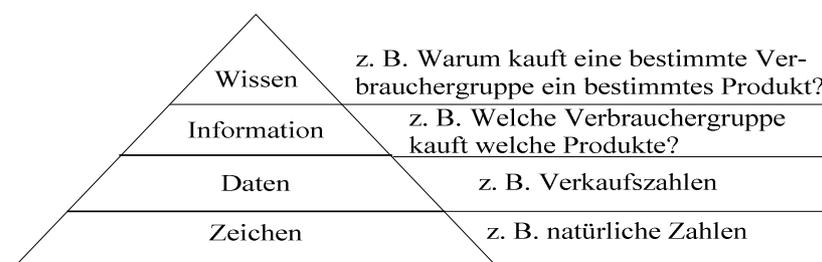
3 Datenintegration

Die Begriffe Daten, Information und Wissen werden häufig synonym gebraucht. Dabei ist eine Abgrenzung notwendig für das weitere Verständnis. Aus diesem Grund erfolgt in diesem Abschnitt zunächst eine Begriffsklärung, der eine Abhandlung über die Aspekte der Integration und der Bedeutung von Information und Wissen im wirtschaftlichen Wettbewerb folgt.

3.1 Daten, Information und Wissen

Daten werden hier als „maschinell verarbeitbare Zeichen mit Bezug zu Objekten“ verstanden (Mertens et al. (2001), S. 58). Ein korrespondierendes Verständnis dieses Begriffes ist die Auffassung von LUFT (1994), der Daten als elementare Mitteilungen versteht, die Bezüge zu einem Wissensbestand haben (vgl. Luft (1994), S. 72 ff.). Hervorzuheben ist bei beiden Aussagen, der Bezug zu Objekten. Ohne diesen Bezug wären es keine Daten, sondern lediglich *Zeichen*. Dieser Bezug führt direkt zum übergeordneten Begriff der *Information*. Diese wird häufig als Untermenge des Wissens, als handlungsrelevantes Wissen angesehen (vgl. z. B. Rautenstrauch/Schulze (2003), S. 7 f.; Luft (1997), S. 195 f.). Durch diese Definition von Information ergibt sich, dass sie im Gegensatz zum Wissen keine permanente Entität ist. Sie entsteht vielmehr punktuell, ist empfängerspezifisch und basiert auf einem Bestand an Erkenntnissen. Informationen sind demnach potenziell im Wissen enthalten (vgl. Rautenstrauch/Schulze (2003), S. 7). *Wissen* ist die Gesamtheit aller Erkenntnisse, die bei einem Menschen, einer Organisation etc. als kognitive Struktur vorhanden ist (vgl. Rautenstrauch/Schulze (2003), S. 7 f.).

Ungeachtet der laufenden Diskussion um die definitorische Abgrenzung der o. a. Begriffe z. B. in der Semiotik und nicht zuletzt (Wirtschafts-) Informatik sollen die angegebenen Definitionen als Basis für die weitere Betrachtung dienen. Abb. 3.1 soll die soeben gegebene Abgrenzung nochmals verdeutlichen.



Quelle: nach Forst (1999)

Abb. 3.1: Begriffsabgrenzung Zeichen – Daten – Information – Wissen

3.2 Bedeutung von Information und Wissen im Wettbewerb

Unternehmen werden gegründet und existieren, um menschliche Bedürfnisse zu befriedigen. Dies ist der Grundgedanke, des wirtschaftlichen Strebens (vgl. Picot (2003), S. 23). Erreicht wird es durch die Erzeugung von Produkten, d. h. durch die Anwendung und Kombination von Produktionsfaktoren. Diese bestehen aus Elementarfaktoren, also Arbeitskraft, Betriebsmitteln, Werkstoffen, sowie dem dispositiven Faktor, der Betriebs- und Geschäftsleitung (vgl. Gutenberg (1983), S. 2 ff.; Hadelar (2000), S. 900, S. 2496). Wirtschaftlicher Erfolg ist nur dann möglich, wenn durch die Kombination der Produktionsfaktoren ein Mehrwert für den Kunden geschaffen wird. Dies kann auf zwei grundsätzliche Weisen geschehen. Zum einen können andersartige Produkte durch neuartige Kombination der Produktionsfaktoren entstehen, zum anderen können bestehende Produkte kostengünstiger durch eine bessere Anwendung der vorhandenen Produktionsfaktoren erstellt werden. Diese zwei Varianten der Faktorkombination entsprechen der Auffassung von PORTER (1995), wonach es für Unternehmen zwei⁴ reine Wettbewerbsstrategien gibt, die Differenzierung und die Kostenführerschaft (vgl. Porter (1995), S. 62 ff.).

„Das Entstehen einer unternehmerischen Idee im Sinne eines Entdeckens neuer Möglichkeiten zur Befriedigung von Kundenproblemen oder effizienterer Formen der Ressourcennutzung [...] sind in höchstem Maß informationsbezogene Tätigkeiten.“ (Picot (1988), S. 544). Information und auch das daraus abgeleitete Wissen spielen demnach bei der Kombination von Produktionsfaktoren eine Hauptrolle. Daraus folgt, dass eine effektive Datenverarbeitung eine Voraussetzung für die Gewinnung von Informationen und Wissen und damit wirtschaftlichen Erfolg darstellt. Vgl. hierzu Abschnitt 3.1 und 3.4.

3.3 Integration

Ausgehend von der Wortbedeutung kann *Integration* als Wiederherstellung eines Ganzen verstanden werden. Die Wirtschaftsinformatik versteht darunter die Verknüpfung von Aufgaben, Technik und Menschen, um den durch starke Aufgabenteilung und Spezialisierung künstlich erzeugten Funktions-, Prozess- und Abteilungsgrenzen entgegenzuwirken (vgl. Mertens (2004), S. 1 ff.).

⁴ In Porter (1995) werden drei Strategie-Typen diskutiert. Die dritte, hier nicht erwähnte, ist jedoch eine Mischform („Konzentration auf Schwerpunkte“), weswegen an dieser Stelle von lediglich zwei *reinen* Strategien gesprochen wird. Vgl. hierzu Porter (1995), S. 62 ff.

Integration kann unter verschiedenen Gesichtspunkten betrachtet werden. Dieser werden, ohne Anspruch auf Vollständigkeit, in Tab. 3.1 gezeigt. Im Folgenden sollen die Daten- sowie die Funktionsintegration näher erläutert werden, da sie für die folgenden Betrachtungen eine besondere Bedeutung haben. Für Details zu den verbleibenden Aspekten der Integration vgl. z. B. MERTENS (2004), RAUTENSTRAUCH/SCHULZE (2003), SCHWARZE (1997).

Tab. 3.1: Aspekte der Integration

Merkmal	Ausprägung				
	Integrationsgegenstand	Datenintegration	Funktionsintegration	Prozess-/Vorgangsintegration	Methodenintegration
Integrationsrichtung	Horizontale Integration			Vertikale Integration (Auf- und Abwärtsintegration)	
Integrationsreichweite	Bereichsintegration	Funktionsbereich- und prozessübergreifende Integration		Innerbetriebliche Integration	Zwischenbetriebliche Integration
Automatisierungsgrad	Vollautomation			Teilautomation	

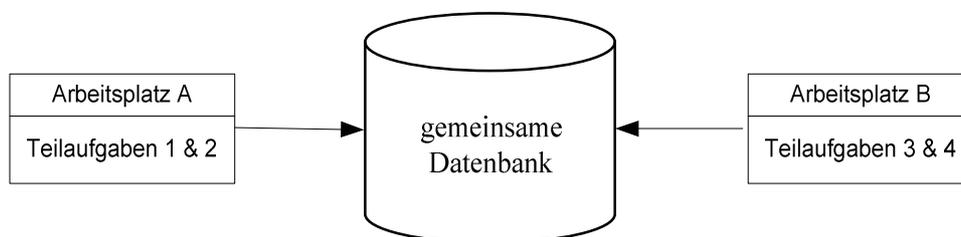
Quelle: nach Mertens (2004), S. 2; Rautenstrauch/Schulze (2003), S. 220

Bezieht man den Integrationsgedanken auf den Bereich der Daten in einem Unternehmen, spricht man von *Datenintegration*. Somit kann darunter die Zusammenfassung ehemals redundant gehaltener Daten verstanden werden (vgl. Rautenstrauch/Schulze (2003), S. 221). Ihr kommt als einer grundlegenden Integrationsform besondere Bedeutung zu, da es im Unternehmen keine Bereiche gibt, die nicht auf Daten anderer Bereiche zugreifen (vgl. Scheer (1990), S. 519 ff.; Rautenstrauch/Schulze (2003), S. 221). Datenintegration ist wichtig, weil bei Insellösungen unweigerlich Redundanzen entstehen (vgl. Rautenstrauch/Schulze (2003), S. 221). Diese erfordern zusätzlichen Aufwand, da jedes dieser redundanten Daten erfasst und gepflegt werden muss und nicht sicher ist, welches das jeweils aktuellste Datum ist. Realisiert werden kann eine Datenintegration prinzipiell durch zwei Vorgehensweisen. Teilsysteme können Daten automatisch an andere weitergeben oder die Daten werden in gemeinsamen Datenbanken gehalten (vgl. Mertens (2004), S. 2). Dadurch lassen sich folgende Vorteile erzielen (vgl. Mertens et al. (2001), S. 57):

- Verbesserte Informationsversorgung der Unternehmensleitung durch erhöhte Datenqualität
- Reduktion des Datenerfassungsaufwandes
- Verminderung der Gefahr von Erfassungsfehlern durch nur einmal erforderliche Eingabe der Daten
- Sinkender Speicher- und Dokumentationsaufwand
- Schnellere Entdeckung von Falscheingaben

Eine weitere Bedeutung kommt der Datenintegration als Voraussetzung für die Funktionsintegration zu. Erst durch eine gemeinsame Datenbasis ist eine Funktionsintegration möglich (Rautenstrauch/Schulze (2003), S. 221).

Funktionen sind Aufgaben, die computergestützt durchgeführt werden. (vgl. Rautenstrauch/Schulze (2003), S. 221). Somit ist *Funktionsintegration* die Zusammenfassung mehrerer „Arbeitsschritte an einem Arbeitsplatz, [...] die in konventionellen betrieblichen Datenverarbeitungssystemen zur Ausnutzung von Spezialisierungsvorteilen getrennt waren.“ (Schwarze (1997), S. 152). Funktionsintegration ist in dieser Ausprägung die technische Umsetzung der Aufgabenintegration. Diese versteht sich als teilweise Umkehrung des tayloristischen Denkmodells, wonach eine möglichst hohe Aufgabenteilung die Produktivität steigert. Das ist nur bis zu einem gewissen Grad der Fall, da bei zu starker Teilung der Koordinationsaufwand zwischen den Arbeitsplätzen höher ist als der eingesparte Verrichtungsaufwand (vgl. Rautenstrauch/Schulze (2003), S. 221). Da die auszuführenden Arbeitsschritte von Daten abhängen bzw. auf diesen operieren, ist eine integrierte Datenhaltung Voraussetzung für eine Funktionsintegration. Abb. 3.2 soll diesen Zusammenhang verdeutlichen.



Quelle: Schwarze (1997), S. 152

Abb. 3.2: Funktionsintegration

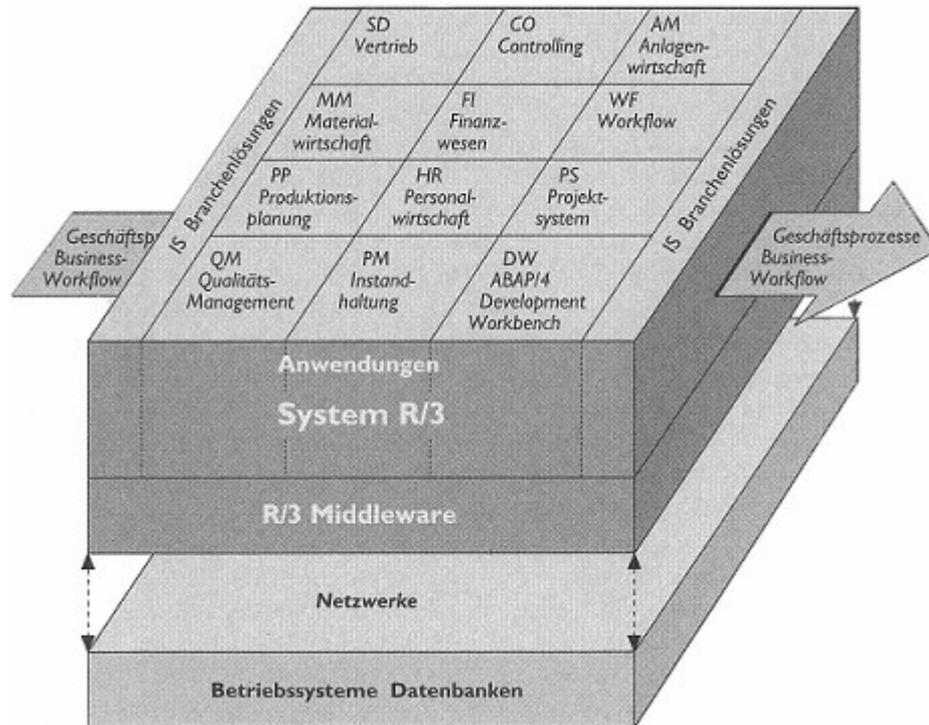
Durch Funktionsintegration sind folgende Vorteile zu erwarten (vgl. Mertens (2004), S. 9 ff.; Rautenstrauch/Schulze (2003), S. 221):

- Zurückdrängen künstlicher Abteilungsgrenzen
- Verringerung der Häufigkeit von Eingabefehlern durch Reduzierung der Anzahl der Mitarbeiter pro Vorgang
- Wegfall von Übergangs- und Einarbeitungszeiten

3.4 Enterprise Resource Planning Systeme als Basis für die Datenintegration

Beginnend in den 1960er Jahren gab es zwei Bereiche, in denen Daten verarbeitende Systeme in Unternehmen Einzug hielten. Dies war zum einen das Rechnungswesen, da man hier die größten Automatisierungspotenziale auf Grund der standardisierten und gesetzlich vorgeschriebenen Verfahren sah. Zum anderen wurden große Rationalisierungspotenziale auch in der betrieblichen Produktion gesehen (vgl. Rautenstrauch/Schulze (2003), S. 313 f.). So entstanden zunächst getrennte Lösungen für einzelne Bereiche. Da dies im Gegensatz zum im Abschnitt 3.3 erläuterten Integrationsgedanken stand, traten an Stelle bereichsspezifischer Lösungen komplexe, integrierte Systeme, die *ERP*-Systeme. Sie sind definiert als integrierte Standardsoftwarepakete zur Abbildung betriebswirtschaftlicher Prozesse in Organisationen (vgl. Haderler (2000), S. 959). Im Weiteren folgt eine allgemeine Erläuterung des Aufbaus und der Funktionsweise von ERP-Systemen. Dabei wird das System ‚R/3‘ der SAP AG als Referenz benutzt, da die SAP AG mit einem geschätzten weltweiten Marktanteil von 56%, im Jahre 2004, als der führende Anbieter für ERP-Systeme gelten kann (vgl. SAP AG (2005), S. 59).

Das ERP-System ‚SAP R/3‘, im Folgenden ‚R/3-System‘ genannt, ist durch einen modularen Aufbau in horizontaler, sowie vertikaler Hinsicht gekennzeichnet (vgl. Rautenstrauch/Schulze (2003), S. 313 ff.; Abb. 3.3). In vertikaler Hinsicht weist das R/3-System einen Schichtenaufbau auf, eine sog. Three-Tier-Architektur. Auf oberster Ebene besteht sie aus einer Präsentationsschicht, der Benutzerschnittstelle. Darunter liegt der Application Server, der anwendungsübergreifende Funktionalität kapselt und Schnittstellen zur obersten und untersten der drei Schichten bereitstellt. Durch Schnittstellen zum Betriebssystem und zur Hardware sorgt er dafür, dass die Präsentationsschicht und die Verarbeitungslogik nicht von der jeweiligen Hardware oder dem Betriebssystem abhängig sind. Der Application Server ist im R/3-System als Middleware implementiert.



Quelle: Buck-Emden/Galimow (1997), S. 191

Abb. 3.3: Architekturschema der R/3-Anwendungen

Die unterste Schicht ist für die Datenhaltung zuständig, wobei hier beliebige relationale Datenbankmanagementsysteme zum Einsatz kommen können (vgl. Rautenstrauch/Schulze (2003), S. 317). In horizontaler Hinsicht ist das R/3-System in funktionale Einheiten oder Komponenten zerlegt, die verschiedenen betriebswirtschaftlichen Bereichen eines Unternehmens zugeordnet sind. Dies sind u. a. (vgl. Rautenstrauch/Schulze (2003), S. 314 f.; Beck-Emden/Galimow (1997), S. 195):

- Rechnungswesen (z. B. Finanzwesen, Controlling)
- Logistik (z. B. Materialwirtschaft, Qualitätsmanagement)
- Personalwirtschaft
- Querschnittslösungen (z. B. Data Warehouse, Bürokommunikation)
- Branchenlösungen (z. B. für öffentliche Verwaltungen, Finanzdienstleister)

Das R/3-System unterstützt eine integrierte Datenhaltung besonders durch zwei Aspekte. Zum einen arbeiten alle Komponenten des R/3-Systems auf derselben Datenbasis. Daten zu einem Vorgang müssen daher nur einmal erfasst werden und stehen dann allen, an der Bearbeitung beteiligten Stellen zur Verfügung (vgl. Buck-Emden/Galimow (1997), S. 192). Zum anderen gewährleistet das R/3-System konsistente Daten in allen

Organisationseinheiten und –ebenen (vgl. Buck-Emden/Gamlimow (1997), S. 109 f.). Das bedeutet, dass z. B. Daten der operativen Ebene zur Entscheidungsunterstützung und -überwachung auf strategischer und administrativer Ebene verwendet werden können. Dadurch ist sichergestellt, dass, bei entsprechender Datenqualität, die aktuellen und tatsächlichen Daten aus dem operativen Geschäft zur Verfügung stehen und nicht erneut gesammelt bzw. eingegeben werden müssen. Eine Komponente des R/3-Systems, die hiervon Gebrauch macht ist z. B. das Data Warehouse, das „eine Gesamtsicht auf alle Daten im Unternehmen ermöglicht.“ (vgl. Buck-Emden/Galimow (1997), S. 205).

4 Entwicklung einer Software zur teilintegrierten Erweiterung des ERP-Systems

4.1 Vorbetrachtung

Das betrachtete Unternehmen setzt für den operativen Bereich ein ERP-System der Firma Microsoft ein. Durch den Wunsch die Anforderungen der Unternehmensleitung und der Benutzer schnell und maßgeschneidert umzusetzen, entschied man sich im Jahre 2003 für die Entwicklung einer eigenen Software, die das ERP-System kostengünstig erweitern sollte. Dazu wurde ein *Framework* entwickelt. Dieses Framework ist ein Entwicklungsrahmen mit klar definierter Schnittstelle und bietet grundlegende Funktionen für die Benutzer- und Rechtverwaltung. In dieses können beliebige Anwendungs-Module eingesetzt werden, die der Schnittstellenanforderung genügen. Das Framework ist über das firmeninterne Netzwerk auf jedem Rechner verfügbar. Es ist jedoch kein integraler Bestandteil des ERP-Systems. Vielmehr stellt es eine eigenständige Applikation dar, die über den Weg des Datenaustausches mit dem ERP-System in Beziehung steht (vgl. Abschnitt 3.2). Dieser Austausch ist jedoch einseitig, d. h. es erfolgen keine Schreibvorgänge in die Datenbanken des ERP-Systems. Vielmehr werden die Daten aus dem ERP-System über *Views* oder Sichten ausgelesen. Eine Sicht ist eine „Folge von Anfrageoperationen [...] die bei jedem Aufruf [...] aus den dann aktuellen Basisrelationen neu berechnet wird.“ (Heuer/Saake (2000), S. 294). Dadurch können bestimmte Daten ausgeblendet, verdichtet oder in einen anderen Zusammenhang gestellt werden. Um Schreibvorgänge in die Datenbank des ERP-Systems zu verhindern, besitzt das Framework lediglich lesenden Zugriff auf diese Datenbank. Alle Daten, die mit dem Framework erfasst werden, sind somit nicht im ERP-System verfügbar. Daher handelt es sich hierbei um eine *teilintegrierte* Erweiterung. Abb. 4.1 verdeutlicht diesen Zusammenhang.

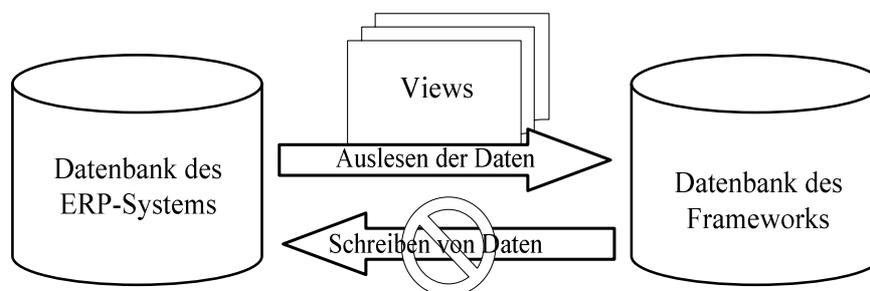


Abb. 4.1: Datenbank-Beziehung ERP-System – Framework

Das erste implementierte Anwendungs-Modul war für die Unterstützung der halbautomatischen Angebotserstellung bestimmt. Inzwischen wurde das Framework um An-

wendungs-Module für die Maschinenbelegungsplanung, für die Erfassung und Verdichtung kundenspezifischer Materialbedarfe, für die Kundendaten-Pflege, für die Bearbeitung von Kundenreklamationen und einige kleinere Module erweitert.

Im Nachfolgenden werden die Entwicklungsschritte beschrieben. Dabei folgt die Beschreibung den Vorgaben der klassischen, *sequentiellen Softwareentwicklung*. Diese schreibt klar abgegrenzte Phasen vor, die vor dem Eintritt in die nachfolgende Phase komplett abgeschlossen sein müssen. Diese sind im Einzelnen (vgl. Dumke (2001), S. 17 ff.; Pomberger/Blaschek (1996), S. 18 ff., Balzert (1996), S. 53 ff.):

- Problembeschreibung
- Anforderungsanalyse
- Spezifikation (oder Definition) und Entwurf
- Implementierung und Test
- Erprobung
- Einführung und Nutzung

Diese Einteilung entspricht jedoch nicht der im Projekt realisierten Reihenfolge. Sie dient lediglich zur Strukturierung der Beschreibung. Eine „rein sequentielle Vorgehensweise [ist] in den seltensten Fällen durchführbar [...]“ (Pomberger/Blaschek (1996), S. 17). Im hier betrachteten Projekt wurde stattdessen eine *prototypische Entwicklungsweise* gewählt. Sie zeichnet sich dadurch aus, dass in einer frühen Phase der Entwicklung lauffähige Teilanwendungen zur Verfügung stehen. Damit steht sie im Gegensatz zur klassischen, sequentiell ausgerichteten Softwareentwicklung, die eine möglichst späte Implementierung lauffähiger Anwendungen, nach Beendigung der vorangehenden Entwicklungsphasen vorsieht. Die Entscheidung für diese Vorgehensweise fiel aus folgenden Gründen. Die Erfahrungen früherer, ähnlicher Projekte hatten gezeigt, dass es den zukünftigen Nutzern leichter fällt über die Anforderungen einer Software zu diskutieren wenn eine konkrete Ausfertigung vorliegt. Ein weiterer Grund war die Größe und die verfügbaren finanziellen Ressourcen dieses Projekts. Dabei führt die prototypische Entwicklungsweise nicht unbedingt zu einer Verminderung der Entwicklungskosten (vgl. Pomberger/Blaschek (1996), S. 175). Die Wartungskosten können jedoch deutlich gesenkt werden und durch frühzeitige Experimente mit der Benutzerschnittstelle entstehen ergonomisch bessere Produkte (vgl. Pomberger/Blaschek (1996), S. 175).

Bei der prototypischen Softwareentwicklung kann man drei grundlegende Formen unterscheiden (vgl. Pomberger/Blaschek (1996), S. 4):

- Exploratives Prototyping
- Experimentelles Prototyping
- Evolutionäres Prototyping

Das explorative Prototyping verfolgt den Zweck, den Entwicklern einen ersten Einblick in den Anwendungsbereich zu geben und verschiedene Lösungsansätze mit den Anwendern zu diskutieren (vgl. Pomberger/Blaschek (1996), S. 4). Die hierbei erstellten Prototypen werden i. d. R. nicht zum Endprodukt weiterentwickelt, sondern verworfen. Gleiches gilt für das experimentelle Prototyping, bei dem eine vollständige Spezifikation von Teilsystemen im Vordergrund steht. Hierbei geht es darum „die Tauglichkeit der Teilspezifikationen, von Architekturmodellen und von Lösungsideen für einzelne Systemkomponenten experimentell nachzuweisen.“ (Pomberger/Blaschek (1996), S. 4). Beim evolutionären Prototyping wird, im Gegensatz zu den erstgenannten Formen, der Prototyp nicht verworfen, sondern dient im Sinne einer inkrementellen Softwareentwicklung als Basis für die weiteren Phasen. Zunächst erfolgt eine Umsetzung der bereits bekannten Anforderungen, die dann schrittweise ausgebaut werden. Ziel hierbei ist die frühzeitige Umsetzung von Benutzeranforderungen und eine Bewertung durch den Nutzer in einer frühen Phase der Entwicklung (vgl. Marciniak (1994), S. 864). Da der Prototyp beim evolutionären Prototyping nicht verworfen wird, sondern als Basis für die weitere Entwicklung dient, wird hierbei von ‚Versionen‘, statt vom 1., 2. usw. Prototypen gesprochen. Eine Unterscheidung zwischen Prototyp und fertigem Produkt ist daher nicht mehr möglich (vgl. Pomberger/Blaschek (1996), S. 5). Die letztgenannte Variante des prototypischen Entwickelns wurde im betrachteten Projekt gewählt.

Dem *Vorteil* der schnellen Erstellung einer ersten lauffähigen Anwendung steht der *Nachteil* gegenüber, dass die schnell und oftmals ohne softwaretechnische Verfahren durchgeführte Entwicklung hinsichtlich der Wartbarkeit, Robustheit und Erweiterbarkeit Defizite aufweisen kann (vgl. Frick (1995), S. 47 ff.).

Da die beiden ersten Phasen der Entwicklung, Problembeschreibung und Anforderungsanalyse, bereits abgeschlossen waren als der Autor in das Projekt einstieg, erfolgt hier lediglich eine Beschreibung der geleisteten Vorarbeiten.

4.2 Problembeschreibung

Nicht implementiert war bisher ein Modul für den Teilbereich der Materialprüfung in der Abteilung Qualitätssicherung. Hier werden alle im Unternehmen erzeugten Mängelberichte erfasst, untersucht und zur weiteren Bearbeitung durch andere Fachabteilungen weitergeleitet. Die wichtigsten Aufgaben sind die Wareneingangsprüfung, die Material- und Produktprüfung vor, während und nach der Produktion und die Überprüfung des Lagerbestandes auf Materialfehler. Geprüft wird dabei die Einhaltung der extern und intern vorgegebenen Standards in Bezug auf Rohmaterial, Zwischen- und Endprodukte. Geprüft werden zum Beispiel die Oberflächenbeschaffenheit des Materials auf Kratzer, Oxidationen, Verfärbungen etc., die Wölbung und die Abmaße. Wird bei dieser Prüfung festgestellt, dass eine Vorgabe nicht erfüllt ist, muss ein sog. *Non Conformance Report* (NCR) erstellt werden. Die weitere Bearbeitung erfolgt dann wie in Abb. A.1 angegeben.

Die Motivation für die Entwicklung einer Software zur Erfassung, Analyse und Wiedergewinnung von Daten in der Materialprüfung ergab sich aus folgenden Gründen. Die Datenerfassung erfolgte in der Vergangenheit mit Hilfe einer Microsoft Access Datenbank und einem sog. Logbuch in Microsoft Excel, in dem zu jedem Non Conformance Report die wichtigsten Eckdaten nochmals aus der Datenbank übernommen wurden. Des Weiteren erfolgte die Erfassung, Änderung und Weitergabe eines Falles zusätzlich auf Formularen aus Papier. Die monatliche und quartalsweise Erstellung von Berichten für die Unternehmensleitung erfolgte manuell, durch nochmalige Aufarbeitung der bereits vorhandenen Daten der Datenbank und des Logbuches.

Durch die Einführung eines elektronischen Erfassungssystems, im Folgenden ‚Software‘ oder ‚Modul‘ genannt, versprach man sich folgende Vorteile, die zum Teil eng mit den Vorteilen einer integrierten Datenhaltung korrespondieren (vgl. hierzu Abschnitt 3.3):

- Beseitigung der manuellen Pflege einer Microsoft Access Datenbank
- Beseitigung der manuellen Pflege eines sog. Logbuches für die Eckdaten der Mängelberichte
- Reduzierung des Einsatzes standardisierter Papierformulare
- Reduzierung des Datenerfassungsaufwandes durch nur einmalige Erfassung
- Erhöhung der Daten-Transparenz im Bereich der Materialprüfung

- Schaffung einer Datenbasis für die halbautomatische Generierung von Berichten für die Unternehmensleitung, mit Hilfe von Microsoft Excel
- Schaffung einer zentralen Anwendung zur Ermittlung aller für den NCR-Prozess benötigten Materialdaten aus dem ERP-System

4.3 Anforderungsanalyse

„Die *Anforderungsanalyse* ist die Phase der Kontrolle von Anforderungen an ein zu entwickelndes Software-System hinsichtlich Korrektheit, Vollständigkeit, Sachgerechtigkeit, Konsistenz und Machbarkeit [...].“ (Dumke (2001), S. 33). Damit schließt sie sich nahtlos an die Problembeschreibung an. Es soll erreicht werden, dass Auftraggeber und -nehmer über dieselben Vorstellungen und Kenntnisse des jeweils anderen Fachgebietes verfügen (vgl. Pomberger/Blaschek (1996), S. 33). Dies kann durch Interviews und Befragungen, Analogieschlüsse und eine Begriffskontrolle geschehen.

Interviews und Befragungen dienen der Ermittlung der Nutzeranforderungen. Anwender können den fachlichen Kontext der zu erstellenden Software besser abschätzen, als es der fachfremde Software-Entwickler kann. Sie können daher die erstellte Liste der funktionalen und qualitativen Anforderungen auf Korrektheit, Vollständigkeit, Sachgerechtigkeit und Konsistenz überprüfen. Diese Prüfung erfolgte im Rahmen von Sitzungen, in denen die ermittelten Anforderungen in Gruppengesprächen geprüft und evtl. berichtigt wurden.

Mit Hilfe von Analogieschlüssen können Aussagen über die Machbarkeit und den Aufwand getroffen werden. Dabei wird versucht durch Vergleich mit bereits abgeschlossenen Entwicklungen Schlüsse über Machbarkeit und Aufwand der Neuentwicklung abzuleiten. Diese Schlüsse sind subjektiv, da es keine allgemeine Vorgehensweise dafür gibt (vgl. Balzert (1996), S. 64). Diese Methode ließ sich im Rahmen des Projektes anwenden, da im Framework bereits ein Modul zur Bearbeitung von Reklamationen externer Kunden bestand. Die Erfahrungen aus der Entwicklung dieses Moduls standen auf Grund der fachlichen und thematischen Nähe als Vergleich zur Verfügung. Eine Prüfung ergab, dass eine Machbarkeit mit ähnlichem Aufwand in personeller, zeitlicher und finanzieller Hinsicht vorlag.

Eine Begriffskontrolle, also eine Überprüfung der korrekten Anwendung von Fachbegriffen (vgl. Dumke (2001), S. 36), fand nicht statt, so dass in der Spezifikation und im Entwurf mehrmals Änderungen der formulierten Anforderungen nötig wurden,

die zu teilweise umfangreichen Änderungen des Prototyps führten. Mit einer gründlichen Begriffskontrolle hätte dies, zumindest teilweise, verhindert werden können.

4.4 Spezifikation und Entwurf

In der *Spezifikation* werden zunächst alle funktionalen und einige qualitative Anforderungen in einem Modell formuliert. Der sich anschließende *Entwurf* ist die Umsetzung dieses Modells in eine Form, die unmittelbar als Implementierungsvorgabe benutzt werden kann (vgl. Dumke (2001), S. 44, S. 61). Für die gewählte, prototypische Entwicklungsweise erschien eine Zusammenlegung der Spezifikation und des Entwurfs als zweckmäßig. Der Grund hierfür liegt in der Eigenschaft des Prototyping, dass Realisierungen sehr frühzeitig erfolgen, die dann von den Nutzern bewertet werden. Um dies zu erreichen wurde in Interviews und Besprechungen die erste Version des Prototyps, der aus den bekannten Anforderungen erstellt wurde, vorgestellt und diskutiert. Erfahrungen aus früheren Projekten im Unternehmen hatten gezeigt, dass die Akzeptanz einer Anwendung stark mit der Bedienbarkeit und Ergonomie der Software zusammenhängt. Deshalb war die erste Version des Prototyps funktionslos und stellte lediglich einen Entwurf der Bildschirmmasken dar. Den zukünftigen Anwendern fällt es leichter über konkrete Realisierungen, also Prototypen bzw. Bildschirmmasken zu diskutieren als über abstrakte, noch zu entwickelnde Software. Somit stellt ein Prototyp eine wesentlich bessere Darstellungsform für funktionale Anforderungen dar als verbale Beschreibungen (vgl. Pomberger/Blaschek (1996), S. 44).

Gleichzeitig mit der Diskussion der ersten Prototyp-Versionen, erfolgte die Ermittlung und Modellierung der abzubildenden *Geschäftsprozesse*. Sie sind „die erfolgsrelevanten, grundlegenden Unternehmenstätigkeiten, die zur Umsetzung der Unternehmensziele und Sicherung des Unternehmenserfolgs dienen [...]“ (Rohloff (1995), S. 84 f.). Die Ermittlung und Modellierung der zu implementierenden Geschäftsprozesse ist wesentlich für die Entwicklung einer Software, da sie implizit die funktionalen Anforderungen enthalten (vgl. Pomberger/Blaschek (1996), S. 33 ff.). Es wurde schrittweise modelliert. Zunächst wurden in Einzel- und Gruppen-Interviews die Funktionen der Geschäftsprozesse ermittelt. Davon ausgehend wurden diese durch Beobachtungen und wiederum Interviews schrittweise verfeinert, d. h. in Teil- und Elementarfunktionen zerlegt. Dabei wurde auf einen möglichst großen Personenkreis zur Ermittlung geachtet, um eine subjektiv gefärbte oder eingeschränkte Sicht auf die Geschäftsprozesse bei der Modellierung zu vermeiden. Abschließend wurden die ermittelten Geschäftsprozess-Modelle in Gruppensitzungen präsentiert, diskutiert und angepasst, um eine möglichst große Über-

einstimmung der Modelle mit der Realität zu erreichen. Für ein Beispiel eines solchen Prozesses vgl. Abb. A.1.

Die Erfassung und Modellierung erfolgte mit Hilfe von Diagrammen der Unified Modeling Language (UML) in Microsoft Visio. Hauptsächlich kamen dabei Aktivitäts- und Zustandsdiagramme zum Einsatz, da in der Vergangenheit bereits eine Modellierung damit erfolgte. Des Weiteren waren die beteiligten Benutzer durch frühere, ähnliche Projekte bereits mit diesen Diagrammen vertraut, so dass keine Umstellung auf andere Modelle, z. B. erweiterte ereignisgesteuerte Prozessketten (eEPK) erfolgte. Für Details zu Grundlagen und Anwendung der eEPK vgl. z. B. SCHEER (2002). Weiterführendes zur UML findet sich z. B. in JACOBSON et al. (2003).

4.5 Implementierung und Test

Implementierung ist die Umsetzung der in der Entwurfsphase erhaltenen Ergebnisse in ausführbare Programme (vgl. Balzert (1996), S. 926; Frick (1995), S. 40). Kodiert wurde die Software in der Sprache Visual Basic .NET. Auf Grund der knappen personellen Ressourcen im Unternehmen, wurde der größte Teil der Kodierung an einen externen Auftragnehmer vergeben. Somit verringerte sich der Implementierungsaufwand auf den Test und die Integration der neuen Software-Versionen in die bereits bestehenden. Ziel des *Testens* ist das Aufdecken möglicher Fehler (vgl. Pomberger/Blaschek (1996), S. 146; Dumke (2001), S. 68). Die Testweise orientierte sich dabei an der prototypischen Entwicklungsweise. Das heißt, dass bereits während der Kodierung getestet wurde. Grobe Software-Fehler oder -Fehlverhalten konnten somit bereits im Vorfeld entdeckt werden. Nach der Ablieferung eines fertigen Softwareteils durch den externen Auftragnehmer bestand die Aufgabe in der *Integration*, also der „[...] Zusammenstellung der akquirierten und selbst erstellten Software-Komponenten [...]“ (Dumke (2001), S. 88). Der Integration folgt der Integrationstest. Damit sollen mögliche Wechselwirkungen der Softwareteile untereinander abgeprüft werden. Der Integrationstest erfolgte nach dem Schema des Sandwich-Integrationstests. Bei dieser Testform wird „[...] von immer größeren Teilbereichen schrittweise zum Test des Gesamtsystems [...]“ übergegangen (Dumke (2001), S. 89). Dadurch ließ sich die Komplexität der möglichen Wechselwirkungen der Software-Komponenten verringern (vgl. Dumke (2001), S. 89).

Wegen der prototypischen Vorgehensweise gab es mehrere Implementierungsphasen. Dadurch erhöhte sich die Komplexität der Entwicklung, da Kodierung, Integration und Test teilweise parallel abliefen.

Gleichzeitig mit der Implementierung der Software erfolgte auch die Implementierung von Berichten für die Unternehmensleitung. Diese enthalten aufbereitete Daten aus der Software, z. B. monatliche Übersichten über Anzahl und Art der Non Conformance Reports. Sie wurden in Microsoft Excel mit Hilfe der eingebauten Datenbankschnittstelle und Pivot-Tabellen erstellt. Durch diese Vorgehensweise erschließen sich zwei Vorteile. Zum einen können die Berichte sehr schnell erstellt und geändert werden, da sie keine weitreichenden Programmierkenntnisse erfordern und größtenteils auf sehr einfachen Datenbankabfragen beruhen. Zum anderen ist Microsoft Excel den meisten Nutzern bekannt und erfordert daher keine Umstellung in der Bedienung. Durch das firmeninterne Netzwerk stehen diese Berichte allen autorisierten Nutzern zur Verfügung.

4.6 Erprobung

„Die Erprobung eines Software-Systems [...] ist der Nachweis seiner Validität auf der Grundlage von Akzeptanzkriterien in einem ausgewählten Anwendungsfeld.“ (Dumke (2001), S. 93). Damit soll sichergestellt werden, dass die Anforderungen der Nutzer und der Auftraggeber erfüllt werden. Die prototypische Entwicklung zwang zu mehrmaligen Erprobungen, da jede Version des Prototyps von den Nutzern und Auftraggebern validiert werden musste. Da eine Erprobung nur in einem ausgewählten Anwendungsfeld stattfinden sollte, wurde jeweils für jeden Erprobungslauf eine repräsentative Nutzergruppe ausgewählt. Diese wurde im Umgang mit dem Prototyp geschult. Da viele der zukünftigen Nutzer bereits an der Spezifikation beteiligt waren, konnte der Aufwand für die Schulungen gering gehalten werden, da Kenntnisse im Umgang mit dem Prototyp vorlagen.

Für die eigentliche Erprobung wurde ein zentrales Dokument zur Verfügung gestellt, das jedem Nutzer von seinem Arbeitsplatz aus zugänglich war. In dieses wurden alle Fehler, Fehlverhalten und Abweichungen von der Spezifikation festgehalten. Nach Ablauf eines im Voraus festgelegten Zeitraums, erfolgte mit den Nutzern eine Überprüfung und Dringlichkeits-Klassifizierung der festgestellten Mängel. Solche, die eine Gefährdung für den Einsatz der Software darstellten, wurden zeitnah behoben. Rechtschreibfehler, Ergonomiemängel etc. wurden erst nach vollständiger Behebung höher klassifizierter Mängel beseitigt.

4.7 Einführung und Nutzung

Als *Einführung* eines Softwareproduktes wird die Phase bezeichnet, in der die Software den Nutzern für den Produktiv-Betrieb zur Verfügung gestellt wird, Schulungen stattfinden und die Software in den Produktiv-Betrieb übernommen wird (vgl. Balzert (1996), S. 964). Da die Software über das firmeninterne Netzwerk jederzeit zur Verfügung steht, erübrigte sich eine gesonderte Installation auf den Arbeitsplatz-Rechnern. Des Weiteren fanden Schulungen für die Nutzer statt, die bis zu diesem Zeitpunkt noch nicht oder nur in geringem Umfang mit der Software in Berührung gekommen waren. Ein weiterer Punkt bei der Einführung war die Umstellung der vorhandenen Datenbestände. Da diese auf verschiedene Medien verteilt waren, entschied man sich aus Gründen der Zeitersparnis dazu, die Altbestände nicht zu übernehmen. Stattdessen wurden die noch schwebenden Vorgänge in der herkömmlichen Art abgeschlossen, also mit Hilfe von Papierformularen, Excel-Dateien etc. Alle Geschäftsvorfälle, die nach dem Einführungstichtag anfielen, durften nur noch mit der neuen Software erfasst und bearbeitet werden.

5 Ergebnisse und Auswertung

5.1 Ergebnisbetrachtung

Um die Ergebnisse des Projekts sichtbar zu machen, wurde im Anschluss an die Einführung im Produktiv-Betrieb eine Umfrage unter den Nutzern durchgeführt. Diese sollte aufzeigen, inwieweit die anfänglich erhofften Vorteile (vgl. hierzu Abschnitt 4.2) durch die neue Software tatsächlich eintraten. Da in einer Umfrage subjektive Aussagen erfasst werden, kann sie eine objektive Ermittlung der abgefragten Aspekte nicht ersetzen. Außerdem erscheint der Zeitraum zwischen Einführung der Software und Umfrage mit ca. zweieinhalb Monaten als zu kurz, um gesicherte Aussagen treffen zu können. Sie soll deshalb an dieser Stelle lediglich als Indiz für die tatsächliche Ausprägung der abgefragten Aspekte gelten. Nachfolgend ist ein Auszug aus der Umfrage und den Ergebnissen aufgeführt.

FRAGE 1: „Ist durch die Einführung der Software eine *Zeitersparnis* bei der Bearbeitung eines Vorfalls vorhanden/spürbar?“

ANTWORT: Ca. 85% der befragten Nutzer bejahten diese Frage. Die Spanne der angegebenen Zeitersparnisse reichte von ca. 0% bis zu ca. 90% der ursprünglichen Zeit. Eine eindeutige Aussage ist hier leider nur fallbezogen möglich, da es strukturell sehr unterschiedliche Fälle zu bearbeiten gibt, die in ihrer Gesamtbearbeitungszeit erheblich voneinander abweichen. Ein geringer Teil der Befragten gab einen erhöhten Zeitaufwand durch die Software an.

FRAGE 2: „Ist eine *Arbeitserleichterung* im Allgemeinen zu spüren? (Keine Mehrfachangaben mehr, bessere Datenverfügbarkeit etc.)“

ANTWORT: Ca. 90% der Nutzer konnten eine deutliche Arbeitserleichterung z. B. durch die Reduzierung der papiergebundenen Arbeiten („Papierkram“), Erhöhung der Abarbeitungsgeschwindigkeit eines Vorfalles und einfachere Verfügbarkeit der benötigten Daten feststellen

FRAGE 3: „Wurde die *Erstellung von Berichten* durch die halbautomatischen Excel-Reports erleichtert?“

ANTWORT: Da lediglich eine sehr kleine Gruppe von Mitarbeitern von zwei bis drei Personen regelmäßig an der Erstellung von Berichten für die Unternehmensleitung beteiligt ist, hatte diese Frage für ca. 90% der Befragten keine Relevanz. Die betroffenen Personen konnten eine Erleichterung feststellen.

FRAGE 4: „Bitte bewerten Sie den *Gesamteindruck* der Software auf einer Schulnotenskala von 1 bis 6.“

ANTWORT: Der ermittelte, gerundete Durchschnittswert ist drei. Viele der Kritikpunkte zielten auf Verbesserungen im Bereich der Bedienerfreundlichkeit und Antwortzeiten der Software ab. Vgl. hierzu auch Abschnitt 5.2.

5.2 Kritik und Ausblick

Wie bereits bei den Umfrageergebnissen in Abschnitt 5.1 angegeben, gab und gibt es Verbesserungspotenzial für die Software. Deshalb soll an dieser Stelle eine kritische Betrachtung der bisher geleisteten Arbeiten und eventueller Verbesserungen erfolgen.

Zunächst muss konstatiert werden, dass die Art der Integration (vgl. Abschnitt 4.1) eine einseitige und *unvollständige Integration* darstellt, da die mit der neuen Software erfassten Daten nicht im ERP-System verfügbar sind. Daher kann nur von einer teilintegrierten Lösung gesprochen werden.

Ein weiterer Kritikpunkt liegt in der *statischen Abbildung* der Prozesse in der Software. Diese sind fest einprogrammiert und daher bei Änderungen im tatsächlichen Ablauf nur schwer zu ändern. Solch eine Änderung bedarf wiederum umfangreicher und aufwändiger Tests und Erprobungen. Eine Lösung, welche die dynamische Konfiguration von Prozessen und Abläufen ermöglicht hätte, wäre jedoch mit den gegebenen Ressourcen nicht möglich gewesen.

Trotz der erwähnten Kritikpunkte hat die vorliegende Lösung ihre Vorteile. Einer liegt in der nun erleichterten Erfassung und Bearbeitung der Geschäftsvorfälle mit der neuen Software. Alle erfassten Daten liegen in kodierter Form vor. Kodierte Daten sind solche, die von Rechnern automatisch weiterverarbeitet werden können (vgl. Schwarze (1997), S. 49 f.). Dies erleichtert in Zukunft den Zugriff auf die Daten und Informationen und die Migration in neue, Daten verarbeitende Systeme. Dieser Vorteil ist insbesondere vor dem Hintergrund einer geplanten Einführung von SAP R/3 zu sehen. Durch die nun kodiert vorliegenden Daten aus der Qualitätssicherung wird die Migration in das R/3-System einfacher als ohne die Software.

Ein weiterer Vorteil ist der nun einfachere Zugriff auf Daten aus dem operativen Geschäft der Qualitätssicherung. Für die Unternehmensleitung bedeutet dies, dass Daten über den Status quo, z. B. Monatsberichte, nicht mühsam beschafft werden müssen, sondern jederzeit abrufbar sind.

Mittlerweile ist die Software seit ca. vier Monaten im Produktiv-Betrieb. Es wurden seitdem ca. 315⁵ Non Conformance Reports erfasst und bearbeitet. Der Schwerpunkt der Aktivitäten liegt jetzt auf der Wartung. Diese beinhaltet Nachbesserungen, vor allem bei der Benutzerschnittstelle, Fehlerkorrekturen und Änderungen infolge neuer Benutzeranforderungen (vgl. Pomberger/Blaschek (1996), S. 172 f.). Die bereits in Abschnitt 5.1 erwähnten Kritikpunkte werden und wurden z. T. bereits beseitigt.

Abschließend kann festgehalten werden, dass die Software aus Sicht der Anwender zu einer Verbesserung der täglichen Arbeit beigetragen hat. Inwieweit eine tatsächliche Verbesserung im Sinne einer Beschleunigung der Abläufe oder Kostensenkung durch Rationalisierung stattgefunden hat, muss eine objektiv durchgeführte Analyse zeigen.

⁵ Stand: Ende Juli 2005

Anhang

A Prozessbeschreibung der Bearbeitung eines Non Conformance Reports

Die Abbildung zeigt die vereinfachte Beschreibung des Prozesses der Bearbeitung eines Non Conformance Reports. Dabei wurden einige Details aus Gründen der Übersichtlichkeit und der Wahrung von Interna weggelassen. Zum Beispiel fehlen hier organisa-

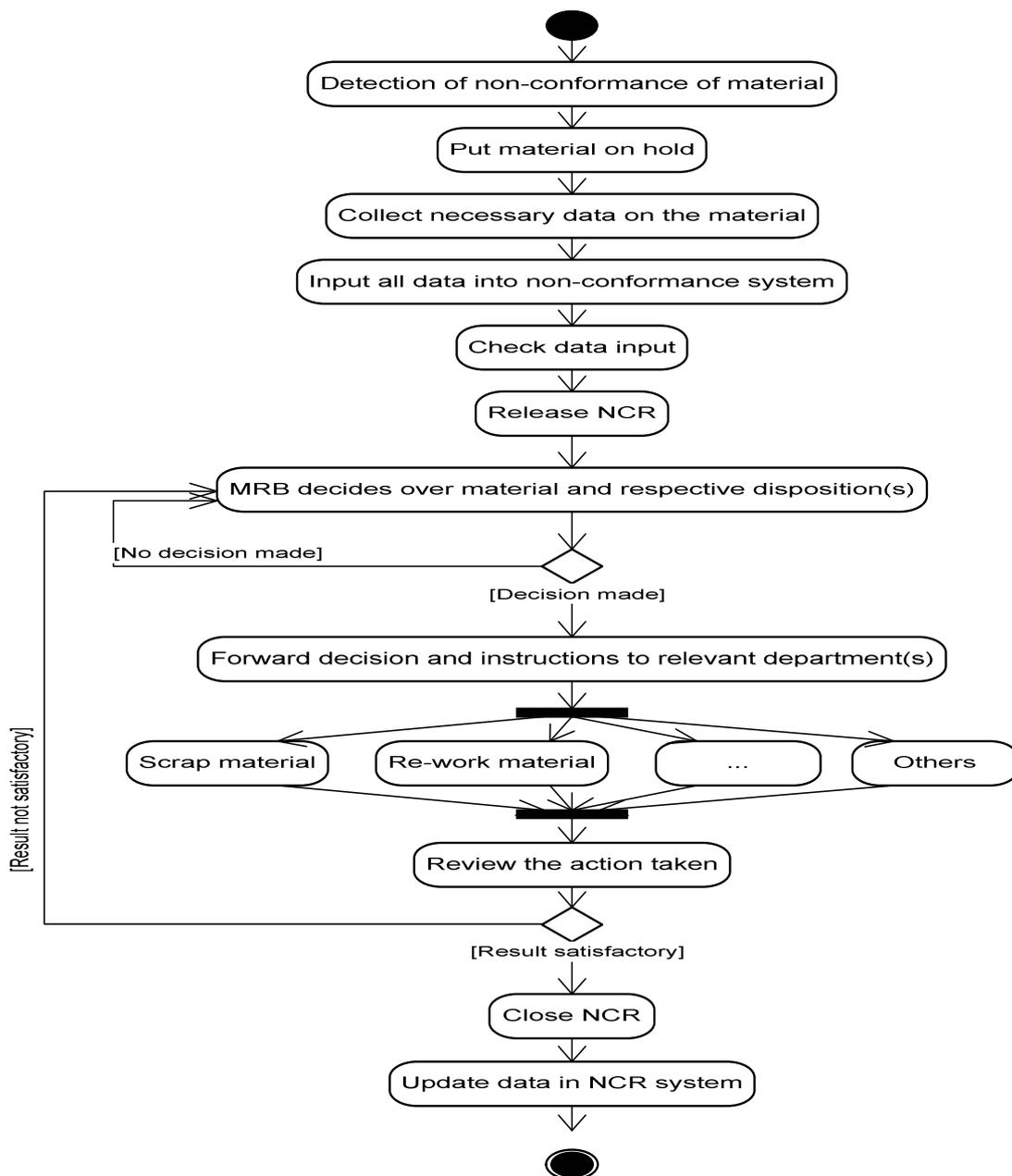


Abb. A.1: Vereinfachte Prozessbeschreibung ‚Non Conformance Report‘

torische Zuständigkeiten völlig, da sie nichts Wesentliches zum Verständnis beitragen würden.

Literaturverzeichnis

- Balzert, H. (1996): Lehrbuch der Software-Technik: Software-Entwicklung. Heidelberg u. a.
- Buck-Emden, R.; Galimow, J. (1997): Die Client/Server-Technologie des SAP-Systems R/3: Basis für betriebswirtschaftliche Standardanwendungen. 3., akt. und erw. Aufl., Bonn u. a.
- Buzzel, D.; Gale, B. T. (1989): Das PIMS-Programm: Strategien und Unternehmenserfolg. Wiesbaden.
- Dumke, R. (2001): Software Engineering. 3. überarb. Aufl., Braunschweig - Wiesbaden.
- Frick, A. (1995): Der Software-Entwicklungsprozess – Ganzheitliche Sicht: Grundlagen zu Entwicklungs-Prozess-Modellen. München – Wien.
- Forst, A. (1999): Information und Wissen (Teil 1): Die neuen betrieblichen Ressourcen. <http://www.doculine.com/news/1999/Februar/infowiss.htm> 01. Juli 2005
- Gutenberg, E. (1983): Grundlagen der Betriebswirtschaftslehre: Band I – Die Produktion. 24., unveränd. Aufl., Berlin u. a.
- Hadeler, Thorsten (2000): Gabler Wirtschaftslexikon. 15., vollst. überarb. und akt. Aufl., Wiesbaden.
- Heuer, A.; Saake, G. (2000): Datenbanken: Konzepte und Sprachen. 2. Aufl., Bonn.
- International Organization for Standardization (2000): Qualitätsmanagementsysteme – Grundlagen und Begriffe (ISO 9000:2000). Genf.
- Jacobson, I.; Booch, G.; Rumbaugh, J. (2003): The Unified Software Development Process: UML. 6th printing, Reading.
- Juran, J. M. (1993): Der neue Juran: Qualität von Anfang an. Landsberg/Lech.
- König, W. (Hrsg.) (1995): Wirtschaftsinformatik '95 – Wettbewerbsfähigkeit, Innovation, Wirtschaftlichkeit. Heidelberg.
- Luft, A. L. (1997): Information – Daten – Wissen. In: Mertens, P. (1997), S. 195-196
- Luft, A. L. (1994): Zur begrifflichen Unterscheidung von „Wissen“, „Information“ und „Daten“. In: Wille, R.; Zickwolff, M. (1994), S. 61-79
- Marciniak, J. J. (Hrsg.) (1994): Encyclopedia of Software Engineering. (Volume II). New York u. a.
- Masing, W. (1999a): Das Unternehmen im Wettbewerb. In: Masing, W. (1999b), S. 3-16
- Masing, W. (Hrsg.) (1999b): Handbuch Qualitätsmanagement. 4., überarb. und erw. Aufl., München – Wien.
- Meffert, H. (1994): Marketing-Management: Analyse, Strategie, Implementierung. Wiesbaden.
- Mertens, P. (Haupt-Hrsg.); Back, A.; Becker, J.; König, W.; Krallmann, H.; Rieger, B.; Scheer, A.-W.; Seibt, D.; Stahlknecht, P.; Strunz, H.; Thome, R.; Wedekind, H. (1997): Lexikon der Wirtschaftsinformatik. 3., vollst. neu bearb. und erw. Aufl., Berlin u. a.
- Mertens, P. (2004): Integrierte Informationsverarbeitung. 14. Aufl., Wiesbaden.

- Mertens, P.; Bodendorf, F.; König, W.; Picot, A.; Schumann, M. (2001): Grundzüge der Wirtschaftsinformatik. 7., neu bearb. Aufl., Berlin u. a.
- Picot, A. (2003): Die grenzenlose Unternehmung – Information, Organisation und Management. 5. Aufl., Wiesbaden.
- Picot, A.; Franck, E. (1988): Die Planung der Unternehmensressource Information (I). WISU, Heft 10 (1988), S. 544-549
- Porter, M. E. (1995): Wettbewerbsstrategie: Methoden zur Analyse von Branchen und Konkurrenten. 8. Aufl., Frankfurt/Main u. a.
- Rautenstrauch, C.; Schulze, T. (2003): Informatik für Wirtschaftswissenschaftler und Wirtschaftsinformatiker. Berlin u. a.
- Rohloff, M. (1995): Integrierte Informationssysteme durch Modellierung von Geschäftsprozessen. In König, W. (1995), S. 83-97
- SAP AG (2005): Konzernlagebericht und Lagebericht der SAP AG. Walldorf.
- Scheer, A.-W. (2002): ARIS - vom Geschäftsprozess zum Anwendungssystem. 4., durchges. Aufl., Berlin u. a.
- Scheer, A.-W. (1990): Wirtschaftsinformatik – Informationssysteme im Industriebetrieb. 3. Aufl., Berlin u. a.
- Schwarze, J. (1997): Einführung in die Wirtschaftsinformatik. 4., völlig überarb. und erw. Aufl., Herne – Berlin.
- Seghezzi, H. D., (2003): Integriertes Qualitätsmanagement. München – Wien.
- Strategic Planning Institute (Hrsg.) (1979): PIMS Data Manual. Cambridge, Mass.
- Strategic Planning Institute (Hrsg.) (1984): The SPI Research Data Base. Cambridge, Mass.
- Timischl, W. (1995): Qualitätssicherung – statistische Methoden. München – Wien.
- Wille, R.; Zickwolff, M. (Hrsg.) (1994): Begriffliche Wissensverarbeitung: Grundlagen und Aufgaben. Mannheim u. a.

Abschließende Erklärung

Ich versichere hiermit, dass ich die vorliegende Studienarbeit selbständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Magdeburg, 19. August 2005

Lars Dankworth

Copyright-Erklärung

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in dieser Arbeit berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.