

# **Prototyping in industriellen Software- Projekten - Erfahrungen und Analysen**

Antoinette Kieback  
Horst Lichter  
Matthias Schneider-Hufschmidt  
Heinz Züllighoven

## **Zusammenfassung**

In vielen ingenieurtechnischen Bereichen ist Prototyping eine Methode, die hauptsächlich dazu eingesetzt wird, neue Projekte, die mit Risiken behaftet sind, kalkulierbar zu machen. In ähnlicher Weise hat Prototyping auch Eingang in die Softwaretechnik gefunden. Bisher fehlen jedoch dokumentierte industrielle Erfahrungen mit diesem Ansatz. Der vorliegende Bericht versucht, diese Lücke zu verkleinern. Er enthält Fallstudien industrieller Software-Projekte, bei denen explizit Prototypen verwendet wurden. Die Bandbreite der ausgewählten fünf Projekte reicht vom Großprojekt mit 240 Personenjahren bis hin zum zwei-Personenjahr-Projekt. Auf der Seite der Entwickler reicht das Spektrum von der DV-Abteilung technischer Großkonzerne über den mittleren bis hin zum kleinen Software-Hersteller. In der zusammenfassenden Analyse haben wir aus unserer Sicht den Nutzen, aber auch die Grenzen von Prototyping formuliert.

# Inhalt

<b>1. Einleitung und Motivation</b>	<b>1</b>
<b>2. Prototyping - Begriffe und Konzepte</b>	<b>4</b>
2.1 Der Prototyp im Software-Entwicklungsprozeß	6
2.2 Ziele des Prototyping	8
2.3 Horizontales und vertikales Prototyping	9
2.4 Das Verhältnis von Prototyp zum Anwendungssystem	10
<b>3. Die Fallstudien der untersuchten Projekte</b>	<b>12</b>
3.1 Ein Beratungssystem	13
3.2 Ein verteiltes Informationssystem	20
3.3 Ein Planungssystem	25
3.4 Ein Dokumentenverwaltungssystem	32
3.5 Ein Prozeßsteuerungssystem	37
<b>4. Analyse der Projekte und Konsequenzen für das Prototyping</b>	<b>42</b>
4.1 Die Grenzen des Prototyping	42
4.2 Die Beteiligung der Anwender bei der Entwicklung	44
4.3 Die Auswirkungen von Prototyping auf die Teamarbeit	45
4.4 Die Planung von Projekten mit Prototyping	46
4.5 Die Dokumentation	47
4.6 Der Stellenwert von Prototypen	48
4.7 Abschließende Bemerkungen	49
<b>Literatur</b>	<b>50</b>

## 1. Einleitung und Motivation

Prototyping ist kein neuer Begriff. In vielen ingenieurtechnischen Bereichen, wie z.B. im Maschinenbau oder im Bauwesen, ist Prototyping eine Methode, die immer dann eingesetzt wird, wenn neue Projekte, die mit Risiken behaftet sind, kalkulierbar gemacht werden sollen. In ähnlicher Weise hat Prototyping auch Eingang in die Software-Technik gefunden.

Aufgrund der Probleme bei der Entwicklung von Software und der Kritik an traditionellen Phasenmodellen, wie etwa dem sog. *Wasserfallmodell*<sup>0</sup>, ist Prototyping in den letzten Jahren zunehmend zu einem Verfahren geworden, um Software-Projekte planbarer und damit auch erfolgreicher zu machen. Ein zentrales Problem bei der Konstruktion von Software ist die Definition der Anforderungen an das geplante System. Dieses Problem wird heute auf die Kommunikationsprobleme zwischen Entwickler und Kunde zurückgeführt: auf der einen Seite kann der Kunde seine Anforderungen nicht so detailliert formulieren, daß sie direkt von den Entwicklern in Software umgesetzt werden können, auf der anderen Seite verfügen die Entwickler nicht über die notwendigen Fachkenntnisse, um die Problemstellung selbst ausreichend analysieren zu können. Ein weiteres nicht zu vernachlässigendes Problem für die Kommunikation ist die unterschiedliche Interpretation von Fachwörtern und scheinbar allgemeinverständlichen Ausdrücken aus dem jeweiligen Arbeitszusammenhang der Benutzer und Entwickler. Mit Hilfe von Prototypen soll dieses Problem gemildert werden, indem sie als Kommunikationsmedium zwischen Entwickler und Kunde dienen.

Doch zunehmend stellt sich heraus, daß Prototyping nicht nur helfen kann, dieses Kommunikationsproblem zu beheben. Die Art und Weise wie Software heute entwickelt und in Organisationen eingesetzt wird, ist fraglich geworden. Dem Ideal eines durchformalisierten, streng arbeitsteiligen und in seinen Schritten und Ergebnissen detailliert zu planenden Software-Projektes, stehen heute die Einsichten und Ideen der evolutionären Systementwicklung gegenüber. Nicht mehr die in sich geschlossene Spezifikation und Implementation eines einzelnen, von grundauf zu entwickelnden Programms steht im Vordergrund, sondern die Weiterentwicklung der informationstechnischen Basis innerhalb einer Organisation.<sup>0</sup>

Prototyping ist aus diesen Gründen auch seit einigen Jahren ein Arbeitsgebiet im Software-Engineering. Aus diesem Arbeitsgebiet entstammen eine Vielzahl von Veröffentlichungen und Studien, die die Vor- und Nachteile dieses Ansatzes

---

<sup>0</sup>vergl. Boehm (1976)

<sup>0</sup>vergl. Budde, Züllighoven (1990)

zeigen<sup>0</sup>, ein Begriffsnetz und eine Klassifikation vorstellen<sup>0</sup>, die Integration von Prototyping in den Software-Entwicklungsprozeß beschreiben<sup>0</sup> oder die Tauglichkeit von Werkzeugen und Sprachen für das Prototyping untersuchen<sup>0</sup>.

Viele der in diesen Veröffentlichungen geschilderten Erfahrungen basieren auf Untersuchungen, die entweder im akademischen Umfeld, z.B. durch studentische Projekte, oder aber durch experimentelle Projekte gewonnen wurden. Bisher fehlen jedoch dokumentierte Erfahrungen aus der industriellen Software-Produktion mit dem Prototyping-Ansatz. Dieser Bericht versucht, diese Lücke zu verkleinern, indem er Fallstudien industrieller Software-Projekte enthält, bei denen explizit Prototypen verwendet wurden.

Zu diesem Zweck wurden Projekte ausgewählt und analysiert, die folgenden Kriterien genügen:

- Das Projekt muß einen industriellen Charakter besitzen; Forschungsprojekte schießen damit aus.
- Entwickler und Anwender müssen verschieden sein. Programmentwicklung für den Eigenbedarf wurde nicht berücksichtigt.
- Die Verwendung von Prototypen muß explizit bei der Projektplanung vorgesehen worden sein. Dadurch schießen die Projekte aus, deren unbefriedigendes Ergebnis nachträglich als Prototyp bezeichnet wurde.

Die Bandbreite der ausgewählten Projekte reicht vom Großprojekt mit einem Gesamtaufwand von 240 Personenjahren bis hin zu kleinen Projekten mit einer Größe von zwei Personenjahren. Die Entwicklergruppen der Projekte waren sowohl DV-Abteilungen technischer Großkonzerne als auch mittlere und kleine Software-Hersteller.

In der Ausarbeitung der Fallstudien haben wir uns bemüht, die Balance zwischen der konkreten Darstellung eines Projektes und der Konzentration auf die wesentlichen Aspekte, die für das Prototyping verallgemeinert werden können, zu wahren. So haben wir grundsätzlich darauf verzichtet, die Namen der einzelnen Projekte oder der daran beteiligten Organisationen zu nennen. Ebenso haben wir nur dann Werkzeuge oder Systeme genannt, wenn sie in ihrer Art typisch für Prototyping-Projekte sind. Es war uns neben den Beschreibungen der Fallstudien wichtig, die in den Projekten mit Prototyping gemachten Erfahrungen zu analysieren und insbesondere den Ansprüchen von Prototyping

---

<sup>0</sup>vergl. Budde et al. (1984), Boehm et al. (1984)

<sup>0</sup>vergl. Budde et al. (1987), Floyd (1984)

<sup>0</sup>vergl. Bischofberger (1989)

<sup>0</sup>vergl. Duncan (1982), Diederich (1987), Doberkat (1989)

gegenüberzustellen. In diese Analyse sind nicht nur Erfahrungen eingegangen, die den fünf beschriebenen Projekten entstammen, sondern auch die Ergebnisse vieler Gespräche mit Praktikern, die wir in unserer eigenen Arbeit und als Dozenten auf diesem Gebiet gesammelt haben. Aufgrund dieser Analyse haben wir aus unserer Sicht den Nutzen, aber auch die Grenzen von Prototyping formuliert.

Dieser Bericht gliedert sich wie folgt: In Kapitel 2 werden die grundlegenden Konzepte und die von uns im Verlauf dieses Berichts benutzte Terminologie eingeführt. Kapitel 3 enthält die Ausarbeitung der fünf Fallstudien. In Kapitel 4 formulieren wir unsere Analyse sowie den Nutzen und die Grenzen von Prototyping.

Dieser Bericht ist das Ergebnis der Arbeit des GI-Arbeitskreises 4.3.2 "Wissensbasierte Systeme für das Prototyping". Der Arbeitskreis hat sich, basierend auf den bisherigen Arbeiten zum Thema Prototyping, zum Ziel gemacht, die Akzeptanz dieses Ansatzes in der Industrie zu untersuchen, Folgerungen daraus zu ziehen und Prototyping im Hinblick auf eine Unterstützung durch wissensbasierte Techniken zu analysieren.

Die hier wiedergegebenen Fallstudien konnten nur durch die Mitarbeit der Personen entstehen, die an den Projekten beteiligt waren. Wir konnten für diesen Bericht auf Erfahrungen der folgenden Firmen zurückgreifen:

*Click Informatik AG, Basel*

*Dialogika Gesellschaft für angewandte Informatik mbH, Saarbrücken*

*Dornier Deutsche Aerospace, Friedrichshafen*

*Interatom GmbH, Bergisch Gladbach*

*InterFace Computer GmbH, München*

*Siemens AG, München*

Wir danken diesen Firmen und insbesondere den entsprechenden Projektmitgliedern für die konstruktive Zusammenarbeit.

## 2. Prototyping - Begriffe und Konzepte

Zunächst wollen wir einen ersten Eindruck davon vermitteln, was wir unter dem Begriff Prototyping verstehen und welche Prinzipien wir damit verbinden.<sup>0</sup>

- PROTOTYPING ist eine Vorgehensweise, die ein bestimmtes *Verständnis* der Software-Entwicklung voraussetzt und die Auswirkungen auf den *gesamten* Entwicklungsprozeß hat.
- PROTOTYPING bedeutet, bei der Systementwicklung *frühzeitig* ablauffähige Modelle ("Prototypen") des zukünftigen Anwendungssystems zu erstellen und mit diesen zu experimentieren.

Prototyping schafft zum einen eine *Kommunikationsbasis* für die Diskussion zwischen allen am Entwicklungsprozeß beteiligten Gruppen. Zum anderen ermöglicht es eine auf *Experiment* und *Erfahrung* gegründete Vorgehensweise bei der Software-Konstruktion.

Damit wir im weiteren ein gemeinsames Verständnis über die aus unserer Sicht am Prototyping beteiligten Gruppen voraussetzen können, erläutern wir einige Begriffe:

- Der AUFTRAGGEBER initialisiert ein Software-Projekt. Meist handelt es sich um eine juristische Person, d.h. eine Firma oder eine Organisation, die in einem ihrer Geschäftsbereiche eine DV-Anwendung einführen oder modifizieren möchte.
- ANWENDER sind diejenigen Personengruppen oder Organisationen, die in ihrer Arbeit von einem Anwendungssystem direkt oder indirekt betroffen sind. Sei es, daß sie die Ergebnisse der Anwendung benutzen oder daß sie die Daten liefern, die für das Anwendungssystem benötigt werden. Oft bezeichnen wir mit Anwender die Organisation insgesamt, in der ein Anwendungssystem verwendet wird oder werden soll.
- Das ANWENDERMANAGEMENT umfaßt die Personen des mittleren Managements, die auf der Fachebene für den Einsatz und die Gestaltung von Anwendungssystemen innerhalb einer Organisation verantwortlich sind.
- Die BENUTZER sind in unserem Kontext nur diejenigen Personen, die ein Anwendungssystem tatsächlich einsetzen. Oft findet man dafür in der Literatur auch die Bezeichnung *Endbenutzer* oder *Endanwender*.

---

<sup>0</sup>Dieser Text ist ein überarbeiteter Auszug aus: "Budde, Kautz, Kuhlenkamp, Züllighoven: Prototyping - an approach to evolutionary system development. Springer Verlag, 1991 (to appear)."

- Ein SOFTWARE-HERSTELLER ist Vertragspartner eines Auftraggebers im Software-Entwicklungsprozeß. Wir bezeichnen damit die Organisationsform, die etwa als Software-Haus oder als DV-Abteilung für die Erstellung und die Integration eines Anwendungssystems verantwortlich ist.
- ENTWICKLER sind diejenigen, die für einen Software-Hersteller den konkreten Entwicklungsprozeß gestalten und dabei mit Benutzern und dem Anwendermanagement zusammenarbeiten.

Die Idee des Prototyping ist entstanden angesichts schlechter Erfahrungen, die Entwickler, Benutzer und Anwender mit traditionellen Software-Entwicklungsstrategien gemacht haben:

Anwender und Benutzer können oft erst beim Einsatz des fertigen Systems ihre Anforderungen an ein solches System deutlich machen --- traditionelle Vorgehensweisen verlangen aber, daß die Anforderungen vollständig zu Beginn der Systementwicklung festzulegen sind.

Entwickler würden am liebsten erst während der Konstruktion eines Systems dessen Spezifikation fertigstellen --- traditionelle Vorgehensweisen lassen die Konstruktion erst nach Abschluß der Spezifikation zu.

Während des gesamten Entwicklungsprozesses ist die gegenseitige Abstimmung der Benutzer und der Entwickler notwendig. Jede Gruppe muß kontinuierlich Wissen über den Arbeitsbereich der anderen Gruppe erwerben --- traditionelle Vorgehensweisen beenden diesen notwendigen Lernprozeß jedoch bereits nach der Anforderungsdefinition.

DV-Abteilungen, die sich gegen die Anwenderabteilungen abschotten, indem sie nur "endgültig fixierte" Spezifikationen annehmen und dann das "fertig" implementierte Produkt "abliefern", sehen sich durch die vielen anwendungsnah arbeitenden DV-Spezialisten und die Dezentralisierung der DV in ihrer Stellung bedroht --- traditionelle Vorgehensweisen halten diese unbefriedigende Abschottung organisatorisch aufrecht.

Im folgenden untersuchen wir vier wesentliche Aspekte des Prototyping innerhalb eines Entwicklungsprojektes:

- *Der Prototyp im Software-Entwicklungsprozeß:* Es gibt verschiedene Arten von Prototypen, die für verschiedene Aktivitäten bei der Software-Entwicklung benötigt werden.
- *Ziel des Prototypenbaus:* Prototypen können zur Klärung verschiedener Fragen eingesetzt werden.
- *Horizontales und vertikales Prototyping:* Prototypen können unterschiedlich aufgebaut werden und für Experimente an verschiedenen Teilen des Anwendungssystems eingesetzt werden.

- *Das Verhältnis von Prototyp zum Anwendungssystem:* Prototypen können unterschiedlich in das Anwendungssystem eingehen.

## 2.1 Der Prototyp im Software-Entwicklungsprozeß

Für die Stellung eines Prototyps im Software-Entwicklungsprozeß sind folgende Merkmale wesentlich:

- Ein Prototyp ist ein *ablauffähiges Modell* des Anwendungssystems; er realisiert bestimmte Aspekte des zukünftigen Systems.
- In der Kommunikation zwischen Entwicklern, Benutzern und Management sind Prototypen ein anschaulicher *Diskussionsgegenstand*. Sie dienen der Diskussion eines Problems, der Klärung einer Frage, oder der Vorbereitung einer Entscheidung. Bei Bedarf werden Prototypen durch schriftliche Systemspezifikationen ergänzt.
- Jeder Prototyp dient als *Spezifikation* für weitere Prototypen oder für das Anwendungssystem.

Damit Prototyping sinnvoll wird, ist es wichtig, daß das Anwendungssystem an der Benutzerschnittstelle nach Funktionalität und Verhalten weitgehend mit den im Prototyp evaluierten Aspekten übereinstimmt.

Drei *zentrale Aktivitäten* im Software-Entwicklungsprozeß können durch die Konstruktion von Prototypen beeinflusst werden: die Projektinitiierung, die Analyse des Anwendungsbereichs und der Entwurf und die Konstruktion des Software-Systems. Um deutlich zu machen, welche Beziehungen wir zwischen Prototyp und diesen Aktivitäten im Auge haben, unterscheiden wir folgende *Arten von Prototypen*:

- Ein DEMONSTRATIONSPROTOTYP unterstützt die Projektinitiierung. Er spielt dort eine bedeutende Rolle, wo ein eigenständiges Software-Projekt zwischen Auftraggebern und Software-Herstellern in die Wege geleitet werden soll, selbst wenn diese beiden Gruppen in einer Organisation beheimatet sind. Bei der Akquisition soll der Demonstrationsprototyp den Auftraggeber überzeugen, daß ein Anwendungssystem entweder prinzipiell gebaut werden kann, oder daß seine Handhabung den Vorstellungen des Anwenders und der Benutzer entspricht.

In der Regel werden solche Demonstrationsprototypen sehr schnell und ohne große Rücksicht auf software-technische Standards gebaut.

- Von einem PROTOTYP IM ENGEREN SINNE sprechen wir, wenn parallel zur Modellierung des Anwendungsbereichs ein ablauffähiges, vorläufiges Software-System erstellt wird. Ein solcher Prototyp soll in der Regel einzelne

Aspekte der Benutzungsoberfläche oder einen Ausschnitt der Funktionalität darstellen und so die Problemstellung klären helfen.

Falls die Fragestellungen inhomogen und offen sind, ist es zweckmäßiger, mehrere "kleine" Prototypen für verschiedene Fragestellungen zu konstruieren, als einen Prototyp immer weiter auszubauen.

- Dient ein Prototyp im wesentlichen der Klärung konstruktionsrelevanter Fragen des Entwicklungsteams, so sprechen wir von einem LABORMUSTER. Ein Labormuster wird aus dem Modell des Anwendungsbereichs und aus einer eventuell bereits vorhandenen Software-Spezifikation abgeleitet. Diese Form des Prototyps findet sich auch in traditionellen Software-Projekten, nur wird die damit verbundene experimentelle Vorgehensweise selten explizit ausgewiesen. Benutzer sind i.a. an der Bewertung von Labormustern nicht beteiligt. Insofern ist der isolierte Einsatz von Labormustern eine eingeschränkte Form des Prototyping.

Ein Labormuster stellt andere Anforderungen an die Konstruktion als ein Prototyp im engeren Sinne: Die modellierten Aspekte sollten technisch mit dem späteren Anwendungssystem vergleichbar sind. Diese Vergleichbarkeit kann sich je nach Fragestellung auf die Architektur oder auf die Funktionalität beziehen.

- Wird ein Prototyp nicht nur zur experimentellen Erprobung einer Konzeption und als "Anschauungsmaterial" verwendet, sondern bereits als Kern des Anwendungssystems im Anwendungsbereich eingesetzt, so nennen wir ihn ein PILOTSYSTEM<sup>0</sup>. Hier ist die strikte Trennung zwischen Prototyp und Anwendungssystem fortgefallen. Ab einer bestimmten "Reife" wird der Prototyp als Pilotsystem praktisch eingesetzt und in Zyklen um weitere Teile ergänzt. Während es i.a. Aufgabe von Anwendermanagement und Entwicklern ist, den Entwicklungsrahmen, d.h. eine Zielvorstellung, für die Software-Entwicklung festzulegen, sollten sich die einzelnen Ausbaustufen des Pilotsystems nach den Prioritäten der Benutzer richten. Technisch erfordert ein Pilotsystem eine wesentlich umfangreichere Gestaltung als ein reiner Prototyp oder ein Labormuster. Denn das Pilotsystem soll im Anwendungsbereich und nicht nur unter Laborbedingungen eingesetzt werden. Selbst bei einer eingeschränkten Benutzung, etwa nur in einer Abteilung oder an einzelnen Arbeitsplätzen, ist eine komfortable und vor allem sichere Bedienbarkeit des Systems und ein Mindestmaß an Benutzerdokumentation unerlässlich.

Prototyping kann also innerhalb eines Entwicklungsprojektes unterschiedliche Aktivitäten unterstützen. Dies wollen wir noch einmal unter weiteren Gesichtspunkten verdeutlichen:

---

<sup>0</sup>vergl. Rzevski (1984)

Demonstrationsprototypen erwecken einen ersten Eindruck bei Auftraggeber und künftigen Benutzern, wie ein Anwendungssystem im Prinzip aussehen könnte. Sie sind noch ausreichend weit von einem tatsächlichen Anwendungssystem entfernt, um daran auch die Begrenzungen eines Prototyps erläutern zu können, was falschen Erwartungen vorbeugen hilft.

Prototypen im engeren Sinne geben den Anwendern und Benutzern eine "greifbare" Vorstellung über die angestrebte Problemlösung und unterstützen dadurch die Analyse und die Einschränkung des Anwendungsbereichs.

Ein Prototyp als Labormuster demonstriert die technische Machbarkeit eines Entwurfs, seine experimentelle Erprobung zeigt die Dynamik und die Restriktionen des Anwendungssystems, ohne daß dieses bereits durchkonstruiert und installiert ist.

Jede Art von Prototyp ist für die Entwickler eine ablauffähige Spezifikation. Er erleichtert die Bewertung der verschiedenen schriftlichen Dokumente und Anforderungen und reduziert den Interpretationsspielraum bei der Software-Konstruktion.

Das Experimentieren mit Prototypen hilft, die organisatorische Integration des Anwendungssystems vorzubereiten, da die Benutzer bereits einen "greifbaren" Eindruck vom System erhalten. Dies gilt verstärkt für Pilotsysteme, die den schrittweisen Ausbau des Anwendungssystems ermöglichen.

## 2.2 Ziele des Prototyping

Prototypen werden im Projektablauf zur *Klärung unterschiedlicher Fragen* verwendet. Wir konzentrieren uns jetzt auf diesen Prototyping-Prozeß und auf die Fragen, die in typischen Situationen im Mittelpunkt stehen und durch Prototyping beantwortet werden sollen. In Anlehnung an Floyd (1984) unterscheiden wir folgende Ziele des Prototyping:

- **EXPLORATIVES PROTOTYPING:** Es wird angewandt, wenn die *Problemstellung unklar* ist. Ausgehend von ersten Vorstellungen werden die Anforderungen der Benutzer und des Managements an das zukünftige System geklärt. Dabei stehen veränderte Arbeitsinhalte genauso zur Diskussion, wie die Frage nach dem Umfang und der Art der DV-Unterstützung. Besonderer Wert wird auf verschiedene Lösungskonzepte gelegt, um den Denkhorizont nicht verfrüht auf einen Ansatz einzuengen (ein Problem, das auch durch Prototyping nicht grundsätzlich behebbar ist). Die Entwickler erhalten einen Einblick in den Anwendungsbereich und in die Aufgaben und Probleme der Benutzer. Diskussionsgegenstand können hier Demonstrationsprototypen und Prototypen im engeren Sinne sein.

- **EXPERIMENTELLES PROTOTYPING:** Die Betonung bei dieser Form des Prototyping liegt auf der *technischen Umsetzung* eines Entwicklungsziels. Einerseits sollen die Benutzer im Experiment ihre Vorstellungen von der DV-Lösung weiter detaillieren, andererseits erhalten die Entwickler eine Grundlage für die Einschätzung der Machbarkeit und Zweckmäßigkeit eines Anwendungssystems. Hier steht die Kommunikation zwischen Benutzern und Entwicklern über technische und software-ergonomische Fragen im Vordergrund: Neben Prototypen im engeren Sinne werden auch Labormuster zur Klärung technischer Fragen der Entwickler konstruiert.
- **EVOLUTIONÄRES PROTOTYPING:** Prototyping wird nicht nur als Hilfsmittel innerhalb eines einzelnen Entwicklungsprojektes eingesetzt, sondern ist ein *kontinuierliches Verfahren*, um ein Anwendungssystem an sich rasch verändernde Randbedingungen anzupassen. Damit verliert Software-Entwicklung auch den Charakter eines abgeschlossenen Projektes und wird zu einem evolutionären Prozeß, der die Anwendung ständig begleitet. Wenn kurze Entwicklungszyklen angestrebt werden, ist es sinnvoll, den Unterschied zwischen Prototyp und Anwendungssystem aufzuheben und Pilotsysteme zu entwickeln. Als eine Konsequenz verändert sich auch die Rolle der Entwickler. Sie sind nicht mehr die "tragenden Figuren" eines abgeschlossenen Projektes, sondern werden zu technischen Beratern, die in Zusammenarbeit mit den Benutzern kontinuierlich an der Verbesserung des Anwendungssystems arbeiten.

### 2.3 Horizontales und vertikales Prototyping

Die Entwicklung von Prototypen ist auch ein technischer Prozeß, der jeweils unterschiedlich gestaltet werden kann. In der Praxis finden wir oft zwei Vorgehensweisen, die durch eine bestimmte Art der Software-Konstruktion nahegelegt werden.

Betrachtet man Software-Entwicklung als Entwurf und Realisierung von Schichten, die von der Benutzungsoberfläche bis zur Basisschicht, i.d.R. das Betriebssystem oder eine Datenbankabfragesprache, reichen, so wird die Unterteilung in horizontales und vertikales Prototyping verständlich.<sup>0</sup>

- Beim **HORIZONTALEN PROTOTYPING** werden nur einzelne Schichten des Systems konstruiert. Beispiele sind die Schicht der Benutzungsoberfläche mit Masken und Menüs oder Schichten des funktionalen Kerns z.B. Datenbanktransaktionen. In der Literatur wird horizontales Prototyping fast immer als Prototyping der Mensch-Maschine-Schnittstelle verstanden.

---

<sup>0</sup>vergl. Mayr et al.(1984)

- Beim VERTIKALEN PROTOTYPING wird ein ausgewählter Teil des Zielsystems vollständig ("in aller Tiefe") implementiert. Diese Technik ist zweckmäßig, wenn die Funktionalität eines Systems und Realisierungsmöglichkeiten fraglich sind. Im Zusammenhang mit dem Bau von Pilotsystemen ist dies gewöhnlich der Fall.

Bei der Betrachtung des Unterschieds zwischen horizontalem und vertikalem Prototyping darf nicht vergessen werden, daß die Methode der Software-Entwicklung die Schichten der Software bestimmt. So ist es keineswegs "natürlich", daß die Bedienschnittstelle die oberste oder unterste Schicht der Software darstellt oder überhaupt in genau einer Schicht zu finden ist. Trotzdem haben wir die Terminologie beibehalten, weil sie unterschiedliche praktisch relevante Vorgehensweisen gut benennt.

## 2.4 Das Verhältnis von Prototyp zum Anwendungssystem

Das Verhältnis von Prototyp und Anwendungssystem ist ein weiteres Kriterium, Prototyping zu klassifizieren. Folgende Herangehensweisen finden wir:

- Ein Prototyp gehört zur Spezifikation des Anwendungssystems:  
 Auf der Grundlage eines akzeptierten Prototyps wird das Anwendungssystem konstruiert. Der Prototyp dient nur als Spezifikation und wird nicht als Baustein für das Anwendungssystem verwendet. Prototyping soll Modellierung und Demonstration ausgewählter Aspekte des gewünschten Systems sein. Die einzelnen Prototypen werden unter Ausnutzung aller technischen Hilfsmittel so rasch wie möglich konstruiert.  
 Die Auswahl der im Prototyp realisierten Aspekte wird meist auf die Benutzerschnittstelle konzentriert, und Fragen wie Ausfallsicherheit oder Effizienz haben kaum Bedeutung. Auch wenn die technischen Realisierungsmöglichkeiten für Prototyp und Anwendungssystem sehr unterschiedlich sind (z.B. Lisp auf einer Lisp-Maschine und Cobol auf einem Großrechner), macht es wenig Sinn, mehr als die Funktionalität im Prototyp zu realisieren. Das Anwendungssystem wird von Grund auf neu konstruiert, um eine software-technisch saubere Lösung zu gewährleisten. Prototypen als Teil der Spezifikation können in Experimenten bewertet werden; für den praktischen Einsatz fehlen ihnen relevante Eigenschaften. Häufig finden wir diese Herangehensweise, wenn Prototyping zur Unterstützung der sog. frühen Phasen eines traditionellen Phasenkonzepts herangezogen wird.
- Prototypen werden zum Anwendungssystem ausgebaut:  
 Mit dem zunehmenden Einsatz von Arbeitsplatzrechnern und der Verbreitung kommerziell einsetzbarer Very High Level Languages vermindert sich der technische Unterschied zwischen Prototyp und Anwendungssystem. Billige Hardware und effiziente Übersetzer oder Interpreter erübrigen oft die

Code-Optimierung. Heute verfügbare Bausteine zur Ein/Ausgabe und Fehlerbehandlung erleichtern es, den Prototyp schrittweise ins Anwendungssystem zu überführen.

Der Prototyp wird zu einem Pilotsystem, das auf der Grundlage weitgehend stabilisierter Benutzer-Anforderungen im praktischen Einsatz erprobt und modifiziert werden kann.

- Prototypen sollen nur Fragen klären:

Ein in der industriellen Praxis selten vorkommender Fall ist, Prototypen nur zum Zweck des Wissenserwerbs zu konstruieren und die Konstruktion eines Anwendungssystem (zunächst) nicht vorzusehen. Diese Art des Prototyping findet man aber in vielen Forschungs- und Entwicklungsabteilungen oder Universitäten. Bedeutung hat dieses Vorgehen aber auch, wenn in einer Vorstudie zunächst Grundsatzentscheidungen über einen notwendigen DV-Einsatz erörtert werden sollen. In derartigen Projekten sollen technische und ökonomische Fragen geklärt oder Ideen ausgearbeitet werden. So kann man z.B. neue Konzepte der Gestaltung von Benutzungsschnittstellen ausprobieren und evaluieren oder die Machbarkeit konstruktiver Ideen prüfen. Die Beziehung zu möglichen Anwendungen bleibt innerhalb des Projektes absichtlich offen.

Oft wird neuartige oder am Markt unübliche Hardware und Software eingesetzt. Derartige Prototypen sind zu unterscheiden von Labormustern, mit denen Fragen im Bezug auf eine geplante Anwendung geklärt werden.

### 3. Die Fallstudien der untersuchten Projekte

Einen zentralen Teil dieses Berichtes bilden die nachfolgend aufgeführten Fallstudien. Es handelt sich dabei um die Aufbereitung der Erfahrungen aus den Projekten, bei denen Prototyping explizit bei der Projektdurchführung vorgesehen war und die wir untersuchen durften.

Wir haben dazu die wesentlichen Erfahrungen, die die Projektmitglieder mit dem Prototyping-Ansatz gewonnen haben, durch Interviews gesammelt. Bei der Auswertung der Interviews haben wir versucht, die erhaltenen Informationen so zu gruppieren und zu strukturieren, daß die Fallstudien weitgehend einheitlich und damit auch vergleichbar formuliert werden konnten.

Im einzelnen haben wir folgende zentrale Aspekte identifiziert, nach denen wir die Fallstudien gegliedert haben:

- das Anwendungsgebiet
- der Entwicklungsprozeß
- die konstruierten Prototypen

Die Beschreibung des Anwendungsgebiets unterteilen wir in die Formulierung der Problemstellung und des ausgewählten Lösungsansatzes. Weiterhin erschien es uns wichtig, die am Projekt beteiligten Personengruppen zu identifizieren.

Bei der Darstellung des Entwicklungsprozesses haben wir versucht, das Projekt zu charakterisieren, die wesentlichen Aktivitäten und Phasen zu identifizieren, den Einsatz von Prototyping zu betrachten und die vorhandenen Werkzeuge aufzulisten.

Bei der detaillierten Betrachtung der konstruierten Prototypen geben wir zuerst eine Klassifikation gemäß der in Kapitel 2 eingeführten Terminologie an und beschreiben anschließend die gefundenen Merkmale der Prototypen.

Abschließend haben wir für jede Fallstudie eine Auswertung der Projekterfahrungen vorgenommen, soweit diese projektspezifisch sind. Die Auswertung enthält sowohl unsere Schlußfolgerungen als auch die Meinung der Projektmitglieder. Erfahrungen, die nicht projektspezifisch sind und auch in anderen Projekten vorgefunden wurden, sind in Kapitel 4 aufgeführt.

## **3.1 Ein Beratungssystem**

### **3.1.1 Das Anwendungsgebiet**

#### **Die Problemstellung**

Zur Unterstützung der Kundenberatung im Dienstleistungsbereich sollte ein Expertensystem entwickelt werden. Es wird von den Kundenberatern eingesetzt, die Privatkunden bei der Auswahl der Serviceleistungen des Unternehmens unterstützen. Das System ist mittlerweile im Einsatz.

Auslösend für das Projekt war die uneinheitliche Beratungsqualität. Ferner erforderte die Änderung der Marktsituation durch das Auftreten von Konkurrenten eine flexiblere Gestaltung des Leistungsumfangs..

Als wichtige Randbedingung waren vom Entwicklerteam zahlreiche technische und organisatorische Regelungen und Verordnungen zu berücksichtigen. Neben sich häufig ändernden Gebühren für die einzelnen Dienstleistungen und Dienstleistungspakete, war bereits bei der Projektinitiierung abzusehen, daß sich Umfang und Art des Serviceangebots im Projektverlauf ändern würden.

Für die Kundenberatung existierte noch keinerlei DV-technische Unterstützung. Die Zusammenstellung und Kalkulation von Serviceleistungen wurden ursprünglich von Hand durchgeführt.

#### **Der ausgewählte Lösungsansatz**

Als Lösung wurde vom Auftraggeber ein regelbasiertes System gewünscht, das auf Arbeitsplatzrechnern dezentral von den verschiedenen Kundenberatern eingesetzt werden kann. Dabei sollen dem Kunden die technischen Kombinationsmöglichkeiten und Einschränkungen bei der Auswahl der gewünschten Serviceleistungen verdeutlicht werden und diese in Einklang mit Regelungen und gesetzlichen Vorschriften gebracht werden. Auf der Grundlage eines vorläufig akzeptierten Servicepakets errechnet das System die anfallenden Gebühren und Kosten und macht selbst Reduktionsvorschläge, wenn der Kunde den Vorschlag als zu teuer ansieht. Jedes Beratungsgespräch kann gespeichert und zu einem späteren Zeitpunkt als Ausgangspunkt einer neuen Beratung genommen werden.

#### **Die beteiligten Personengruppen**

Der Auftraggeber ist ein Dienstleistungsunternehmen der öffentlichen Hand. Das Entwicklerteam setzte sich aus einem Konsortium zweier Software-Hersteller zusammen.

### 3.1.2 Der Entwicklungsprozeß

#### Charakterisierung des Projekts

Das Entwicklerteam bestand aus zwei Gruppen von mehreren erfahrenen Mitarbeitern und wurde besonders in der Endphase durch einzelne Studenten unterstützt. Dazu kam ein vom Auftraggeber ausgewählter Unternehmensberater. Die Benutzer wurden durch einen Beauftragten des Auftraggebers repräsentiert, der zwar nicht unmittelbar mit dem System arbeiten sollte, aber Fachwissen im Anwendungsgebiet mitbrachte.

Für den ersten Prototyp, der zur Projektakquisition diente, wurden ca. sechs Personenmonate in vier Zeitmonaten aufgewendet. Das eigentliche Projekt wurde dann in ca. sieben Personenjahren abgewickelt.

Das Beratungssystem ist in der Programmiersprache Prolog realisiert und umfaßt ca. 3.5 MByte Core, dies entspricht mehr als 4.000 Prolog-Regeln.

Das Projekt wurde vom Auftraggeber auf der Basis eines Festpreises finanziert. Die Auslieferung des zweiten Prototyps erfolgte ca. sieben Monate nach Termin, bei einer geplanten Entwicklungszeit von acht Monaten. Die Verzögerung war insofern auf das evolutionäre Prototyping zurückzuführen, als dadurch bis zuletzt zusätzliche Wünsche des Auftraggebers berücksichtigt wurden. In der Verzögerungszeit mußte weiterhin eine Umstellung der Wissensdomäne vorgenommen werden, da der Auftraggeber eine tiefgreifende Umstellung in der Organisation seines Angebots durchführte. Anhand dieser Domänenänderung wollte er die Tragfähigkeit des wissensbasierten Programmieransatzes testen.

#### Die wesentlichen Aktivitäten und Phasen

Das Projekt wurde aufgrund einer öffentlichen Ausschreibung initiiert. Anschließend wurden einige interessierte Software-Hersteller im Rahmen einer Vorstudie aufgefordert, aufgrund einer schriftlichen Spezifikation einen ersten Prototyp zu erstellen. Nach der Auswertung dieses Prototyps durch Management, DV-Experten und Benutzervertreter des Auftraggebers, wurde vom Management des Auftraggebers die Entscheidung gefällt, den Auftrag an das Konsortium zu vergeben.

Im nächsten Schritt wurde auf der Basis des ersten Prototyps ein zweiter Prototyp entwickelt, der schrittweise zu einem voll funktionalen Pilotsystem ausgebaut wurde.

Dieses Pilotsystem wurde zunächst an einer Stelle eingesetzt und nach einem Jahr Erprobung sukzessive bei den Beratungsstellen des Auftraggebers installiert.

Das System wird seither regelmäßig entsprechend den Anforderungen des Auftraggebers modifiziert. Dies wird durch ein Entwicklerteam geleistet, das personell nicht mehr mit den ursprünglichen Entwicklern identisch ist. Die Dynamik der Wissensbasis ist erheblich: ca. 20% der "Logik"-Regeln und 50% der Objekt-Prädikate mußten im letzten Jahr geändert werden. Diese Änderungen des Systems können als echte Weiterentwicklungen betrachtet werden. Programm- oder Logik-Fehler sind in den letzten Jahren kaum mehr aufgetaucht. Verantwortlich für die meisten Modifikationen sind die Veränderungen der Marktsituation und das modifizierte Serviceangebot des Auftraggebers.

### **Der Einsatz von Prototyping**

Im Projektverlauf wurden mehrere Prototypen erstellt. Die Projektakquisition erfolgte auf der Grundlage eines ersten Prototyps, der ausschließlich einen Ausschnitt der Funktionalität des späteren Anwendungssystems modellierte. An der Entwicklung dieses Prototyps waren keine Benutzer oder sonstige Anwendervertreter beteiligt. Die notwendigen Informationen mußten aus "schriftlichen Unterlagen" (Richtlinien, technische Beschreibungen, Gesetzestexte) extrahiert werden.

Nachdem der Auftrag vergeben war, wurde auf Grundlage des ersten Prototyps der zweite funktionale Prototyp erstellt, der alle wesentlichen Systemfunktionen und die vollständige Benutzungsoberfläche modellierte. Dieser Prototyp war der Ausgangspunkt einer evolutionären Entwicklung hin zum ausgelieferten System, das als Pilotsystem ebenfalls noch Prototyp-Charakter hat.

Um diesen zweiten Prototyp zu erstellen, wurden alle Bereiche analysiert und modelliert, die im ersten Prototyp noch nicht vorhanden waren. Einige Teile des ersten Prototyps konnten unverändert in den zweiten Prototyp übernommen werden, andere mußten angepaßt und abgestimmt werden.

Während der Arbeit am zweiten Prototyp hatten die Entwickler keinen Kontakt mit den späteren Endbenutzern des Systems. Der Experte des Auftraggebers traf mit dem Unternehmensberater alle wesentlichen Entscheidungen für die Gestaltung der Benutzungsoberfläche und definierte auch die Benutzungssituationen.

Neben diesen beiden Prototypen, die für die Kommunikation und die Zusammenarbeit mit dem Auftraggeber von großer Bedeutung waren, wurden im Projektverlauf mehrere kleinere experimentelle Prototypen erstellt, die im Entwicklerteam Fragen der Machbarkeit sowie Designalternativen klären halfen. Es wurden insbesondere Details der Benutzungsschnittstelle als Prototypen realisiert und untersucht.

## **Die verwendeten Werkzeuge**

Die Entwicklung fand insgesamt auf Arbeitsplatzrechnern statt. Die Funktionskomponenten der Prototypen - also auch des Pilotsystems - sind weitgehend in Prolog realisiert. Dazu kommen einzelne systemnahe Teile in C. Die Oberfläche wurde mit dem Fenstersystem Collage von Siemens implementiert.

Während das Prologsystem als stabil und für die Entwicklung als geeignet eingeschätzt wird, war das Fenstersystem in der ersten Projektphase noch nicht stabil und erforderte einigen Aufwand bei der Fehlersuche.

### **3.1.3 Die konstruierten Prototypen**

#### **Klassifizierung der Prototypen**

In diesem Projekt wurden drei verschiedene Arten von Prototypen erstellt.

Der erste Prototyp sollte zum einen die Machbarkeit des Systems zeigen, hatte aber vor allem Akquisitionscharakter. Damit geht er über das hinaus, was wir als reinen Demonstrationsprototyp bezeichnet haben, denn er realisierte bereits wesentliche Aspekte der Funktionalität des Anwendungssystems. Dieser Prototyp diente nicht nur als Spezifikation, sondern bildete auch den technischen Kern zur Weiterentwicklung des Systems.

Der zweite Prototyp ist ein Prototyp im engeren Sinne. Er umfaßt gegenüber dem ersten weitere Teile der Funktionalität sowie die Benutzungsoberfläche. Dieser Prototyp wurde soweit ausgebaut, daß er am Ende der Hauptphase des Entwicklungsprojekts an den Auftraggeber ausgeliefert und dort auch als Pilotsystem eingesetzt werden konnte.

Dazu kommt eine Reihe von kleinen Prototypen, die als Labormuster im Entwicklerteam selbst ausgewertet wurden und zur Klärung technischer Fragen dienten.

#### **Die Merkmale der Prototypen**

Der erste Prototyp war ein reiner Demonstrationsprototyp, der einen Teil der Funktionalität zeigte und einen Vorschlag für die Benutzungsschnittstelle machte, der jedoch vom Auftraggeber später verworfen wurde.

Dies führte zu einem technischen Problem bei der Entwicklung der interaktiven Komponenten des zweiten Prototyps. Die Benutzungsoberfläche konnte erst mit dem zweiten Prototyp realisiert werden, da bei der Konstruktion des ersten Prototyps noch keine verbindliche Entscheidung für ein bestimmtes Benutzungsschnittstellen-Konzept vorgegeben war und die notwendige Hard- und Software noch nicht verfügbar war. Insgesamt wurden mehrere Versionen der

Benutzungsschnittstelle getestet. Da die Benutzungsschnittstelle im ersten Prototyp nicht vorhanden war, mußten grundsätzliche Designentscheidungen mehrfach revidiert werden. Eine Ursache für die verschiedenen Änderungen waren die unterschiedlichen Gestaltungsvorschläge durch den Auftraggeber, die er von Prototypen, die im Rahmen der Vorstudie von anderen Firmen gebaut wurden, ableitete. So wurde vom Auftraggeber grundsätzlich gefordert, daß keine Tastatur verwendet werden sollte. Eine zweite Quelle für Design-Vorschläge waren die Erfahrungen des Unternehmensberaters mit der Macintosh-Oberfläche.

### **Die erstellte Dokumentation**

In diesem Projekt wurden lediglich informelle Beschreibungspapiere und ein Pflichtenheft erstellt, auf ausführliche Codedokumentation wurde "zugunsten einer schnellen Fertigstellung" verzichtet. Allerdings waren die deklarativen Code-Teile bis zu einem gewissen Grad selbstdokumentierend. Dies bedeutete konkret, daß Programmquelltexte in Prolog von einem Anwendungsvertreter im Laufe des Projektes gelesen und kommentiert werden konnten. Szenarien, die von den Anwendern erstellt wurden und typische oder erwünschte Anwendungssituationen zeigten, waren besonders hilfreich, um die Verwendung des Systems zu analysieren. Diese Szenarien wurden den Entwicklern als Anforderungen mitgegeben.

Die Benutzeranleitung wurde in der Endphase des Entwicklungsprojektes von studentischen Hilfskräften erstellt, die erst nach der eigentlichen Entwicklung des Prototyps ins Projekt kamen. Dies wurde von den Entwicklern als sehr positiv bewertet, da das System auf diese Weise aus der Sicht von Laien beschrieben wurde.

Es zeigte sich, daß über diese Dokumente hinaus eine Dokumentation im Sinne einer Systemspezifikation für die folgende Modifikationsphase notwendig wurde.

## **3.1.4 Die Auswertung des Projekts**

### **Die charakteristischen Merkmale**

Die initiale Spezifikation des Systems erfolgte durch die Entwickler auf der Grundlage von schriftlichen Dokumenten ohne direkte Kontakte mit dem Auftraggeber. Dies wird heute als gravierendes Problem gesehen, da die Unterlagen weder vollständig noch ausreichend waren.

Die späteren tatsächlichen Benutzer wurden nicht in den Entwicklungszyklus einbezogen. Das Fehlen der Benutzer im Entwicklungsprozeß scheint aber in diesem Fall charakteristisch, da sich das System mehr an den Bedürfnissen und Anforderungen der Auftraggeber als an den Benutzern orientierte. Wir schließen aus Hinweisen der Entwickler, daß diejenigen Benutzer, die das System

einsetzen, mit dem System zufrieden sind und im Durchschnitt bessere Ergebnisse (d.h. höhere Aufträge/Kunde) zu verzeichnen haben.

Die Domänenänderung während der Konstruktion des zweiten Prototyps hätte nach Meinung der Entwickler zu einer Reimplementierung des Systems führen müssen. Insbesondere hätten die redundante Systemteile entfernt werden müssen und die systeminternen Schnittstellen neu entworfen und auf die neue Domäne angepaßt werden müssen. Dies wurde allerdings, nach unserer Einschätzung typischerweise, aus Kostengründen nicht gemacht. Die Kosten für eine solche Reimplementierung sind nach Einschätzung der Entwickler niedriger zu bemessen, als die Kosten, die für die nachträglichen Änderungen notwendig gewesen wären. Diese Reimplementierung wurde aber erst nach der Auslieferung durchgeführt. Dabei zeigte sich, daß das System sich innerhalb der durchkonstruierten Teile leicht ändern ließ, die gewachsenen Schnittstellen verursachten dagegen überproportionalen Änderungsaufwand.

Prototyping-Projekte erfordern, daß die beteiligten Gruppen in einem höheren Ausmaß zusammenarbeiten und miteinander diskutieren, als es bei konventionellen Software-Projekten der Fall ist. Dies gilt auch für den unmittelbaren räumlichen Zusammenhang eines Projektes. In vorliegenden Fall erwies sich die große Entfernung zwischen Entwicklerteam und Anwenderorganisation als zusätzlicher Aufwand, der schwer kalkuliert werden konnte.

In diesem Projekt wurden der Entwicklungsprozeß und die Arbeitsergebnisse nur minimal schriftlich dokumentiert. Nur die Benutzeranleitung wurde gegen Projektende von nicht unmittelbar an der Entwicklung beteiligten Personen erstellt, was nach Einschätzung des Entwicklerteams allerdings der Qualität der Benutzeranleitung zugute kam. Erst nach dem eigentlichen Projekt wurde die Bedeutung einer Dokumentation des Pilotsystems deutlich, da das System von Personen weiterentwickelt wurde, die im ursprünglichen Projekt nicht beteiligt waren.

Ein Merkmal, das zwar nicht unmittelbar mit Prototyping zusammenhängt, aber als Erfahrung nicht verloren gehen sollte, ist die Einschätzung der Entwickler, daß ein Konsortiumsprojekt von zwei Software-Herstellern zu einem unverhältnismäßig hohen Aufwand bei der Projektleitung und Abstimmung führt. Die hohe Integration der einzelnen Arbeitsschritte bei Prototyping-Projekten erlaubt keine über weite Zeiträume separat und selbständig arbeitenden Teams.

### **Die Bewertung von Prototyping für das Projekt**

Prototyping wird von den Entwicklern als einsetzbares Verfahren bewertet, insbesondere dann, wenn mit dem Auftraggeber Einigkeit über den Zweck des Prototyps und seine Leistungsfähigkeit erzielt werden kann.

Der erste Prototyp, der nur Funktionsaspekte realisierte, hat nach Einschätzung der Entwickler die Planung für das Gesamtsystem sehr erleichtert. Bei der Entwicklung der Benutzungsschnittstelle war die Planung mangels eines derartigen Prototyps wesentlich schwieriger und ungenauer, da die dafür notwendige Hard- und Software erst sehr spät zur Verfügung stand.

Zentrale Probleme bei diesem Prototyping-Projekt resultierten daher, daß der erste Prototyp zwar Aspekte der Funktion, aber keine verbindliche Benutzungsoberfläche modellierte.

Die Kalkulation des Projektes auf der Basis eines Festpreises sowie die Planung und Kontrolle anhand traditioneller Life Cycle Konzepte haben sich als hinderlich für das Prototyping erwiesen. Bei der Planung und der Finanzierung eines solchen Projekts sollte berücksichtigt werden, daß Designalternativen auch tatsächlich erprobt werden können. Aus Kostengründen wurden in diesem Projekt Alternativen fast nicht evaluiert. Wenn dieses doch notwendig war, wie bei der Entwicklung der Benutzungsschnittstelle, dann wurden die Kosten dafür vom Software-Hersteller übernommen.

## 3.2 Ein verteiltes Informationssystem

### 3.2.1 Das Anwendungsgebiet

#### Die Problemstellung

Bei dem untersuchten Projekt forderte ein externer Auftraggeber ein Software-System, das die Kommunikation und den Informationsaustausch für mehrere über Deutschland verteilte Computer sicherstellen sollte. Eine detaillierte Aufgabenbeschreibung konnte von ihm jedoch nicht erstellt werden, sie konnte lediglich rudimentär beschrieben werden.

Teil der Vertragsgestaltung war deshalb ein eher vages und unvollständiges Pflichtenheft, das als Basis für die Entwickler diente.

#### Der ausgewählte Lösungsansatz

Die Aufgabenstellung, so wie sie formuliert wurde, forderte die Entwicklung eines verteilten Informationssystems. Das System sollte aus einer verteilten Datenbasis und der notwendigen Anwendungssoftware bestehen, die ebenfalls auf mehreren Computern verteilt werden sollte. Diese Computer sollten über ein Wide Area Network miteinander verbunden werden. Die technische Basis des Informationssystems sollte ein kommerzielles Datenbanksystem sein. Große verteilte Informationssysteme zeichnen sich dadurch aus, daß sie von mehreren Anwendern benutzt werden. Die Systemanforderungen für ein solches System werden demzufolge von verschiedenen Personengruppen geäußert.

Da die Anforderungen nur vage vorlagen, mehrere Anwendergruppen bei der Fixierung der Anforderungen berücksichtigt werden mußten und der Auftraggeber gefordert hatte, die Systemspezifikation ausgehend von der Benutzungsoberfläche zu erarbeiten, wurde entschieden, Prototyping einzusetzen. Mittels eines Prototyps sollte die Gestaltung der Benutzungsschnittstelle untersucht werden. Begleitend dazu sollten die beim Prototyping-Prozeß identifizierte Funktionalität beschrieben und mit den Anwendern diskutiert werden.

Während die Konstruktion der Benutzungsschnittstelle von einem geeigneten Werkzeug unterstützt werden sollte, sollten die Operationen nur beschrieben und nicht prototypisch realisiert werden.

#### Die beteiligten Personengruppen

An diesem Projekt waren drei unabhängige Parteien beteiligt: Der Auftraggeber hat das Projekt initiiert und ist Vertragspartner des Software-Herstellers. Die

Anwender, die das System später einsetzen sollen, waren nicht identisch mit dem Auftraggeber, wodurch Probleme in der Anforderungsabstimmung entstanden. Das Entwicklerteam des Software-Herstellers bestand aus ca. 80 Personen.

### **3.2.2 Der Entwicklungsprozeß**

#### **Charakterisierung des Projekts**

Das gesamte Projekt wurde auf ca. drei Jahre geplant. Zwei der drei Jahre wurden für die Implementierung angesetzt, ein Jahr wurde für die Analyse und Spezifikation verwendet. Während der Analyse und Spezifikation wurden wiederum zwei bis drei Monate für das eigentliche Prototyping benötigt. Durch den Einsatz von geeigneten Werkzeugen sollte ein effizientes Arbeiten über den kompletten Life Cycle von der Analyse bis zur Implementierung gewährleistet werden.

Aus früheren ähnlichen Projekten hatten die Entwickler Erfahrungen bei der Konstruktion von verteilten Systemen. Doch weder Entwickler noch die Benutzer, geschweige denn der Auftraggeber, hatten Erfahrungen auf dem Gebiet des Prototyping. Die Gruppe, die den Prototyp entwickelte war auch gleichzeitig das Entwicklerteam des Zielsystems.

Der Projektvertrag wurde auf der Basis eines Festpreises erstellt und enthält die erste rudimentäre Grobspezifikation für das zu erstellende System. Der Projektvertrag schreibt weiterhin vor, daß die Systementwicklung durch Prototyping unterstützt werden soll. Aufgrund der Erfahrungen mit dem Prototyp, der eine genauere Systembeschreibung lieferte, wurden später Folgeverträge erstellt, die den Entwicklungsaufwand klarer eingrenzten.

#### **Die wesentlichen Aktivitäten und Phasen**

Auf der Grundlage der Grobspezifikation wurde in der Analyse- und Prototypingphase ein Prototyp der Benutzungsschnittstellen erstellt. Mit diesem und den detailliert formulierten Funktionsbeschreibungen wurde die Grobspezifikation konkretisiert.

Aufgrund der Evaluierung des Prototyps ergaben sich Anforderungen, die nicht im Rahmen des geplanten Budgets, das auf der Grundlage der Grobspezifikation kalkuliert worden war, durchgeführt werden konnten. Daher wurde nach einer Neukalkulation ein Folgevertrag erstellt, der die zusätzlichen Benutzerwünsche enthielt. Ab diesem Zeitpunkt wurden keinerlei wesentliche Änderungen mehr in der Spezifikation vorgenommen.

Nach Abschluß der Prototypingphase wurde das System inkrementell entwickelt. Die erstellten Masken des Prototyps konnten direkt übernommen werden. Die Operationen wurden auf der Basis der erstellten Beschreibung entwickelt.

## **Der Einsatz von Prototyping**

Neben der Problemstellung war ein ausschlaggebender Grund für die Anwendung von Prototyping die Tatsache, daß die gewählte Entwicklungsumgebung die Konstruktion von Prototypen der Benutzungsschnittstellen gut unterstützte. Prototyping sollte hauptsächlich bei der Anforderungsanalyse und Schnittstellendefinition eingesetzt werden. Dazu wurde ein Prototyp konstruiert, mit dem die Benutzer schon kurze Zeit nach Projektbeginn vor dem Terminal die ca. 3000 benötigten Masken mit den Entwicklern in die gewünschte Form brachten.

Bei der Bewertung des Prototyps konnte den Benutzern mit dem Prototyp eine vollständige Oberfläche des Systems präsentiert werden, was als ein Vorteil gewertet wurde, da sich u.a. die Verständnisprobleme zwischen Entwickler und Anwender verringerten. Parallel zu diesem Prototyp wurde die Funktionalität des Systems beschrieben, aber nicht im Prototyp realisiert und evaluiert. Nachdem jeweils eine Prototyp-Entwicklungsstufe abgeschlossen war, wurden die Benutzer in die Evaluierung einbezogen.

Der Prototyp hat Anforderungen und Bedürfnisse bei den Benutzern geweckt, die die Aufgabenstellung drastisch änderten; zum Teil hat sich dadurch das Entwicklungsvolumen wesentlich erweitert. Während der Systementwicklung wurden für spezielle Probleme Laborprototypen gebaut, die z.B. zentrale Operationen realisierten und die Qualität einer Lösung demonstrieren sollten.

## **Die verwendeten Werkzeuge**

Die Systementwicklung wurde mit der im Vertrag vorgegebenen computerunterstützten Umgebung durchgeführt. Dazu zählen das Datenbanksystem Adabas, die Entwicklungs- und Prototypingsprache NATURAL und das Data Dictionary Predict. Alle während der Entwicklung entstandenen Dokumente (z.B. Funktionsbeschreibungen, Masken, Handbuchttexte) wurden zentral in der Datenbank gespeichert und verwaltet. Um die Dokumentation zu erleichtern, wurde ein eigenes Werkzeug entwickelt, das aus der Datenbank nach vorgegebenen Mustern die Dokumente aufbereitet.

### **3.2.3 Der konstruierte Prototyp**

#### **Klassifizierung des Prototyps**

Von Projektbeginn an war geplant, nur einen Prototyp zur Gestaltung der Benutzungsschnittstelle zu erstellen, da nur dieser Teil durch einen verfügbaren Maskengenerator unterstützt wurde. Im Prototyp wurden alle benötigten Bildschirmmasken vollständig erstellt, so daß es sich um horizontales Prototyping handelte.

## **Die Merkmale des Prototyps**

Da das Zielsystem mit derselben Entwicklungsumgebung und derselben Sprache realisiert werden sollte wie der Prototyp, konnten alle Bildschirmmasken in das Zielsystem übernommen werden, was einen erheblichen Teil der Aufwendungen für das Zielsystem darstellte.

### **3.2.4 Die Auswertung des Projekts**

#### **Die charakteristischen Merkmale**

Charakteristisch für dieses Projekt ist unserer Meinung nach die Tatsache, daß Prototyping akzeptiert und damit auch eingesetzt wird, wenn eine ausreichende Werkzeugunterstützung vorhanden ist, die die Konstruktion des Prototyps wesentlich erleichtert. Stehen diese Werkzeuge nicht zur Verfügung, hätten die Entwickler nach eigener Aussage traditionell gearbeitet. Erst die Werkzeuge hatten ihnen Prototyping nahegelegt.

Weiterhin kann festgestellt werden, daß die Benutzer beim Prototyping durch die Integration in den Entwicklungsprozeß motiviert werden und das Gefühl bekommen, am Projekt entscheidend beteiligt zu sein. Allerdings kann diese Euphorie dazu führen, daß sie mit immer neuen Ideen das System erweitern wollen. Aussagen wie "Ja, wenn das so einfach geht, dann wollen wir noch dies und jenes" sind typisch dafür. Auch während der Entwicklung des eigentlichen Zielsystems kamen bei diesem Projekt noch Anregungen von den Benutzern, die jedoch nicht mehr berücksichtigt werden konnten. An diesem Beispiel zeigt sich, daß eine Balance gefunden werden muß zwischen der Modifikation von Systemanforderungen aufgrund sinnvoller Benutzerwünsche und der Konzentration auf das im Rahmen eines Entwicklungszyklus Machbare.

#### **Die Bewertung von Prototyping für das Projekt**

Die Beurteilung des Prototyp-Ansatzes ist sowohl bei den Entwicklern, als auch bei den Anwendern durchweg positiv. Beide würden das eingesetzte Werkzeug noch einmal zum Prototyping verwenden, da es damit möglich ist, bereits frühzeitig die Gestalt und die Arbeitsweise des Systems sowohl dem Nutzer als auch dem Entwickler zu demonstrieren. Das führt auf der einen Seite zu einer Motivationssteigerung bei Entwicklern und Benutzern, auf der anderen Seite eröffnet dies die Möglichkeiten zur Diskussion. Daraus resultiert im Regelfall eine bessere Qualität des Anwendungssystems. Allerdings kann aus der Sicht des Projektmanagements der ebenfalls damit verbundene erheblich höhere Entwicklungsaufwand negativ bewertet werden.

In diesem Projekt zeigte sich durchaus nicht untypisch, daß Benutzer, die etwa einen Heimcomputer besitzen oder auf andere Weise Grundkenntnisse in der

Programmierung haben, über die Art und Weise der Implementierung mitbestimmen wollen, was im Regelfall verhindert werden muß.

Die vertragszwingenden Formalia wurden als unnötiger Ballast bei der Projektdurchführung angesehen. Das Änderungsverfahren sollte so modifiziert werden, daß auch während der Entwicklungsphase die Änderungen mit dem Auftraggeber abgestimmt werden können, damit dadurch die Entwicklung noch stärker durch die Entwickler und Benutzer getragen wird. Jede Änderung der Anforderungen, die durch die Erfahrungen mit dem Prototyp notwendig wurde, mußte im vorliegenden Fall mit umständlichen Änderungsanträgen formal abgestimmt werden. Damit neue Methoden eingesetzt werden können, müssen solche Vertragstechniken dem Entwicklungsmodell angepaßt werden.

Da die Gruppe, die den Prototypen entwickelte, auch gleichzeitig das spätere Entwicklerteam des Endsystems war, gab es kaum Übertragungsprobleme vom Prototyp zum Zielsystem.

## 3.3 Ein Planungssystem

### 3.3.1 Das Anwendungsgebiet

#### Die Problemstellung

Auf dem Gebiet der Heizungs-, Klima- und Sanitärtechnik liegt ein Großteil der Aktivitäten im Bereich der Anlagenplanung. Traditionell werden diese Dienstleistungen arbeitsteilig mit manuellen Geräten und Hilfsmitteln wie Zeichenbrett, Skizzen, Bauzeichnungen, Kalkulationen und Materiallisten erbracht. Die resultierenden Pläne und Listen sind teils ohne vorgeschriebene Formate, teils entlang bestehender Normen oder Vorschriften zu erstellen und dienen als Spezifikation für die zu erbringende Leistung. Da viele Kataloge mit gerätetechnischen Daten das Umfeld sehr komplex werden lassen, werden mehrere Iterationsschritte bereits für die Realisierung der Pläne benötigt.

Neben diesen traditionell manuellen Verfahren verwenden die Ingenieurbüros in verstärktem Maße PC-Programme, die jeweils Lösungen für einzelne Aufgabenstellungen im Bereich der Planung bereitstellen, wie z.B. Programme für Druckverlustberechnungen. Da diese Programme aber nur einzelne Arbeiten unterstützen und für andere Aufgabenstellungen keine Anwendungsprogramme existieren, lassen sie sich nicht so miteinander verbinden, daß die Planung durchgängig computerunterstützt durchgeführt werden kann.

Ein integriertes Planungssystem sollte hier Abhilfe schaffen. Ein weiteres Moment für die Entwicklung des Planungssystems kam daher, daß zunehmend Architektenbüros auf CAD-Systeme umstellen. Diese erwarten dann, daß die ihnen nachgeordneten Ingenieurbüros die Architektenpläne in Form von CAD-Daten übernehmen.

Aus diesen Gründen entstand die Idee, ein integriertes System zur Planung von Projekten der Heizung-, Klima- und Sanitärbranche zu realisieren. Dieses System sollte alle benötigten Design-Werkzeuge in Form eines Werkzeugkastens für die Planung der Projekte bereitstellen. Da für ein Projekt verschiedene Subsysteme - wie Heizung oder Entlüftung - geplant werden müssen, sollte das System zusätzlich die Möglichkeit anbieten, verschiedene Sichten auf die Gesamtplanung darzustellen. Weiterhin wurde gefordert, daß das Planungswerkzeug die Teamarbeit unterstützen muß, also mehrbenutzertauglich sein soll.

#### Der ausgewählte Lösungsansatz

Bei der ersten Analyse der Problemstellung stellte sich heraus, daß das System über eine einheitliche Benutzungsoberfläche, mit der alle Werkzeuge in uniformer

Weise bedient werden können, und über eine einheitliche Datenschnittstelle verfügen muß. Das System sollte so offen konzipiert werden, daß es möglich ist, bestehende Werkzeuge, die bestimmten Qualitätsansprüchen entsprechen, in die integrierte Planungsumgebung aufzunehmen, damit wenigstens ein Teil der bereits in den Ingenieurbüros eingesetzten Programme auch im integrierten Planungssystem weiterverwendet werden kann. Dazu war es notwendig, die verwalteten Daten zu anwendungsnahen Einheiten zusammenzufassen und mit einer entsprechenden Semantik zu versehen, d.h. etwa die graphischen Elemente eines Plans als Rohrleitungen und nicht als Pixel oder als Strichgrafik zu verwalten.

Da die Evaluierung der existierenden CAD-Systeme zu keinem System führte, das als Basissystem verwendet werden konnte, mußte ein solches Basissystem neu entwickelt werden. Dabei sollten möglichst viele existierende Subsysteme verwendet werden. Es wurde vorgesehen, ein kommerziell verfügbares Datenbanksystem einzusetzen, um die zentrale Datenhaltung des Systems zu realisieren. Die Benutzungsschnittstelle sollte mit Hilfe eines speziell dafür geeigneten Werkzeugs erstellt werden.

### **Die beteiligten Personengruppen**

Das Projekt wurde von einem Software-Hersteller durchgeführt, der zu diesem Zeitpunkt keine Erfahrungen mit ähnlichen Projekten oder mit Prototyping hatte. Bei der Durchführung des Projekts wurde die Firma von einem staatlichen Forschungsinstitut finanziell und personell unterstützt und teilweise von einer Hochschulinstitution beraten. Der Auftraggeber bestand aus einem Konsortium von größeren und mittelständischen Branchenfirmen.

## **3.3.2 Der Entwicklungsprozeß**

### **Charakterisierung des Projekts**

Da es sich bei diesem Projekt um ein neues Anwendungsgebiet für die Entwickler handelte, war bei der Gestaltung des Projektteams von vorne herein geplant, daß neben den Informatikern auch Ingenieure aus dem Anwendungsgebiet mitarbeiten sollten. Das Entwicklerteam bestand über die Gesamtdauer des Projekts aus drei Informatikern und einem Ingenieur aus dem Anwendungsgebiet.

Die Entwickler bildeten zusammen mit Vertretern des Auftraggebers einen Projektausschuß. Dieser hatte die Aufgabe, über den Projektfortschritt zu wachen und entsprechende Maßnahmen bei auftretenden Schwierigkeiten, wie z.B. bei der Finanzierung oder bei personellen Engpässen des Entwicklerteams, einzuleiten. Das Projekt wurde 1987 gestartet. Für den ersten Oberflächenprototyp benötigte das Entwicklerteam drei Monate. Das anschließend entwickelte Pilotsystem soll Ende 1991 fertiggestellt sein.

## Die wesentlichen Aktivitäten und Phasen

Da es auf diesem Gebiet keine Vorarbeiten gab, wurde in der ersten Projektphase ein Konzept für die Integration von unterschiedlichen Werkzeugen in die Planungsumgebung entwickelt. Parallel dazu wurde die Einsatzumgebung des Systems, also die Planungsabteilungen in den Branchenfirmen, analysiert. Zu diesem Zweck wurden einige typische Betriebe der Branche ausgewählt und untersucht. Zur Systemanalyse wurde eine computerunterstützte Methode eingesetzt. Sie wurde, nachdem einige in Frage kommende Methoden evaluiert worden waren, mit einem Structured Analysis Werkzeug durchgeführt. Mithilfe dieses Werkzeugs wurde ein Pflichtenheft erstellt. Die Analyse und ihre Darstellung innerhalb der Methode wurden in gewissen Teilen vollständig durchgeführt, in anderen Bereichen nur bis zu einem bestimmten Grad. Bei der Umsetzung der SA-Diagramme nach Structured Design zeigte sich jedoch, daß das vorhandene Werkzeug und das dieser Methode zugrunde liegende Konzept diesen Schritt nicht leisten konnten, sodaß die Arbeit mit der Methode und dem Werkzeug abgebrochen werden mußte.

Am Ende der Analysephase zeigte sich, daß das geplante System gegenüber den ursprünglichen Vorstellungen umfangreich und komplex werden würde und nur auf Arbeitsplatzrechnern zu realisieren wäre. Diese neuen Anforderungen an den Systemumfang und die Hardware führten dazu, daß die Auftraggeber die Durchführung des gesamten Projekts in Frage stellten.

In dieser Situation kamen die Entwickler zu der Einschätzung, daß die Abstimmung und die Kommunikation zwischen den Entwicklern, den Anwendern und weiteren potentiellen Software-Herstellern ein zentrales Problem war. Um dieses Problem zu beseitigen, wurde ein Prototyp erstellt, der als Kommunikationsgrundlage diente. Da zu diesem Zeitpunkt auch für die Entwickler noch nicht klar war, wie das geplante System realisiert werden konnte, wurde ein einfacher Oberflächenprototyp gebaut. Aufgrund der positiven Aufnahme der Prototypdemonstration wurde von den Auftraggebern entschieden, das Projekt fortzusetzen.

In der Hauptphase des Entwicklungsprojektes wurde ein voll funktionstüchtiger Prototyp erstellt. Zur Klärung von technischen Designfragen wurden begleitend experimentell kleinere Prototypen realisiert.

Der funktionale Prototyp wird so weit entwickelt, daß er als Pilotsystem in einigen ausgewählten Ingenieurbüros eingesetzt werden kann, damit Erfahrungen mit dem Planungssystem gewonnen werden können. Der Prototyp soll Ende 1991 diesen Ausbaustand erreicht haben. Anschließend ist geplant, das Pilotsystem bis zur Produktreife weiterzuentwickeln.

## **Der Einsatz von Prototyping**

In diesem Projekt wurden Prototypen unter zwei Aspekten eingesetzt. Zum einen dienten sie der Erleichterung der zeitweise problematischen Kommunikation mit späteren Anwendern und den Auftraggebern. Da andererseits das Entwicklerteam über keine vergleichbaren Projekterfahrungen verfügte, wurden Prototypen zum anderen auch dann erstellt, wenn Risiken nicht abgeschätzt werden konnten. Im Verlauf des Projekts wurden daher mehrere Prototypen entwickelt.

## **Die verwendeten Werkzeuge**

Die Entwicklung der Prototypen und des Pilotsystems fand auf einem UNIX-Arbeitsplatzrechner mit Bitmap-Display und X-Windows statt. Der Oberflächenprototyp wurde mit dem Werkzeug Apollo-Dialog erstellt. Dieses enthält eine spezielle Sprache, um Benutzungsschnittstellen beschreiben zu können. Weiterhin wurde ein kommerzielles CAD-System benutzt, um die Pläne, die der Prototyp zeigte, zu erstellen.

Sowohl der funktionale Prototyp als auch das Pilotsystem werden in Ada entwickelt, wobei Fremdsoftware-Komponenten für die Datenhaltung und für die Benutzungsschnittstelle verwendet werden. Eine eigentliche in sich geschlossene Entwicklungsumgebung existiert nicht.

### **3.3.3 Die konstruierten Prototypen**

#### **Klassifizierung der Prototypen**

Der erste einfache Oberflächenprototyp war ein reiner Demonstrationsprototyp. Seine Funktion bestand darin, den Vertretern des Auftraggebers einen nachvollziehbaren Eindruck von den Möglichkeiten einer software-technischen Lösung für ihre Problemstellung vorzuführen. Dieser Prototyp zeigte die Arbeit und die Funktionsweise des Planungssystems anhand eines speziell für den Prototyp entwickelten Planungsszenarios, mit jeweils simulierter Funktionalität. Dazu wurden lediglich Daten aus einem CAD-System in Fensterrahmen verschiebbar montiert und einige vorgefertigte Dialoghülsen abgerufen. Damit vermittelte der Prototyp zwar das Flair der möglichen Umgangsformen; er sollte aber noch kein Modell des tatsächlichen Aufbaus des späteren Systems sein. Das unterscheidet diesen Demonstrationsprototyp von einem horizontalen Prototyp.

Der Prototyp konnte nicht von den Anwendern benutzt werden, er wurde ihnen in Form einer geführten Demonstration von einem Mitglied des Entwicklungsteams präsentiert. Da der Oberflächenprototyp nur eine Attrappe war, gab es außer den subjektiven Äußerungen der Anwender keine Messungen im eigentlichen Sinne. Der Prototyp bewirkte jedoch, daß die Anwender ein Gefühl für die Funktionsweise und die Vorteile des Planungssystems bekamen und die Fortsetzung des Projekts befürworteten, er hatte somit hauptsächlich

Akquisitionscharakter. Nachdem er seine Aufgabe erfüllt hatte, wurde er nicht mehr benötigt und war insbesondere kein technischer Ausgangspunkt für die weitere Systementwicklung.

Diese einfache Attrappe wurde in drei Monaten erstellt, wobei zwei Monate für die Definition des Prototypszenarios und ein Monat für die Realisierung benötigt wurden.

Nach der Präsentation des Oberflächenprototyps wurde die Entwicklung eines funktionalen Prototyps begonnen. Dieser wird inkrementell bis zum auslieferbaren Pilotsystem entwickelt. Der Prototyp dient als das wesentliche Kommunikationsmittel zwischen den Entwicklern und dem Projektausschuß. Zu diesem Zweck wird der Prototyp am Ende eines jeden Entwicklungsschritts - etwa alle 3-4 Monate - dem Projektausschuß präsentiert. Der Projektausschuß hat dadurch die Möglichkeit, einerseits den Arbeitsfortschritt zu kontrollieren und andererseits den Prototyp zu evaluieren. In die Bewertung des Prototyps werden auch vereinzelt weitere potentielle Benutzer einbezogen.

Die weitere Planung sieht vor, daß das Pilotsystem "vor Ort" zusammen mit den Entwicklern ausgewertet wird. Die dabei auftretenden Probleme werden diskutiert und bei der weiteren Entwicklung des Pilotsystems berücksichtigt.

Um die technische Realisierbarkeit für bestimmte Probleme, wie z.B. im Bereich der Datenhaltung oder der Benutzungsschnittstellengestaltung, zu zeigen, wurden in kleineren internen Projekten Labormuster erstellt. Dabei konnten sowohl Erfahrungen mit den verwendeten Fremdsoftware-Komponenten und der gewählten Programmiersprache gesammelt werden als auch die Machbarkeit unterschiedlicher Lösungsansätze gezeigt oder verworfen werden. Zu erwähnen ist dabei insbesondere ein internes Projekt, das das entwickelte Konzept für die Realisierung der Datenhaltung erproben sollte. Aufgrund der Erfahrungen mit diesem Labormuster mußte eine andere Realisierungsmöglichkeit gefunden werden.

Die globale Aufgabenstellung selbst hat sich, da sie am Beginn des Projekts eher allgemein formuliert war, durch das Prototyping nicht geändert. Dagegen wurden insbesondere durch die Erfahrungen mit den Labormustern und dem Pilotsystem neue Möglichkeiten für die Realisierung gewonnen. Da die Prototypen bisher nur zur Demonstration und Bewertung dienten, ist ein direkter Einfluß auf den Anwendungsbereich noch nicht festzustellen. Allerdings zeigen punktuelle Erfahrungen, daß die tägliche Arbeit mit dem Pilotsystem zu neuen Zusammenhängen und Sichten auf das Arbeitsmaterial des Planers führen wird.

Der Oberflächenprototyp diente dazu, die Auftraggeber vom gewählten Konzept zu überzeugen. Die Labormuster und die funktionalen Prototypen unterstützten im wesentlichen die Entwurfs- und Codierungsphase. Das Pilotsystem wird die weitere Anpassung des Systems an die Anforderungen der Benutzer erleichtern.

### **Die Merkmale der Prototypen**

Der Oberflächenprototyp diente lediglich zur Demonstration. Er zeigte zwar die bei der Planung notwendigen Arbeitsmaterialien und einige denkbare Werkzeuge, sollte aber keine Vorgabe für das angestrebte Planungssystem sein.

Bei der Realisierung des Pilotsystems wurden verschiedene kommerzielle Subsysteme, wie ein Datenbanksystem und ein Software-Baukasten für Benutzungsschnittstellen verwendet. Die Integration dieser Subsysteme mit den anderen in Ada implementierten Teilen bereitete den Entwicklern allerdings erhebliche Schwierigkeiten.

Das Pilotsystem wird in drei- bis viermonatigen Schritten erweitert, bis die geplante Funktionalität erreicht ist. Das Pilotsystem wird allerdings nicht alle gestellten funktionalen Anforderungen erfüllen, so wird es z.B. keine Recovery-Mechanismen besitzen und keine automatische Konsistenzüberprüfung anbieten.

### **3.3.4 Die Auswertung des Projekts**

#### **Die charakteristischen Merkmale**

Bei diesem innovativen Projekt haben wir ein Beispiel gefunden, wie die Aufgabenstellung und die Reaktionen der Anwender zu einem evolutionären Entwicklungsprozeß führten. Dabei ist bemerkenswert, daß das Projekt zunächst ganz konventionell konzipiert war. Doch die Systemanalyse mittels Structured Analysis führte zu einer Spezifikation, die zwar das Problemfeld beschrieb, aber nicht als Kommunikationsgrundlage zwischen den Entwicklern und dem Auftraggeber geeignet war.

Der Nutzen der verwendeten Methode bei der Systemanalyse bestand aus Sicht der Entwickler darin, daß sie dadurch gezwungen wurden, die im System modellierten Strukturen im Detail auszuformulieren und dabei zu einer einheitlichen Begriffsbildung zu gelangen. Sie haben dadurch das Aufgabengebiet leichter und besser verstanden. Gleichzeitig wurde den Entwicklern dadurch klar, daß sich die anstehenden technischen und kommunikativen Probleme rein analytisch und spezifizierend nicht lösen ließen.

So konnten erst aufgrund der Demonstration des einfachen Oberflächenprototyps die vorher etwas zögerlichen Vertreter des Auftraggebers von dem geplanten Konzept überzeugt werden. Der Prototyp war unserer Meinung nach deswegen erfolgreich, weil er den Anwendern als Abbild genau die Arbeitsmaterialien, wie z.B. Pläne oder Kostenblätter, präsentierte, die sie aus ihrer täglichen Arbeit gewohnt sind. Allerdings glauben wir, daß die frühzeitige Festlegung auf eine möglichst direkte Abbildung der traditionellen Arbeitsmaterialien und Hilfsmittel die Ausschöpfung DV-spezifischer Möglichkeiten eingeschränkt hat.

Bei den zyklischen Evaluierungen des Prototyps wurde von den Entwicklern sehr bald festgestellt, daß ein Großteil der Benutzer nur sehr beschränkt Einfluß auf die Entwicklung nehmen kann, da sie die Anforderungen nicht abstrakt darstellen konnten. Damit kreative Vorschläge für zukünftige Weiterentwicklungen gemacht werden können, müssen die Nutzer das System auch technisch verstehen. Als charakteristisch für die Evaluierung von Prototypen konnte festgestellt werden, daß es den Benutzern schwerfällt, konstruktive Kritik am Prototyp zu äußern, und daß für die Kommunikation mit dem Benutzer neben dem Prototyp selbst auch erhebliches Know-How aus dem Anwendungsgebiet erforderlich ist. Bei diesem Projekt war ein Experte direkt im Entwicklerteam.

Der Prototyp ist nach Meinung der Entwickler jedoch das einzige sinnvolle Medium, um überhaupt mit den Anwendern über die Problemlösung diskutieren zu können.

### **Die Bewertung von Prototyping für das Projekt**

Prototyping wurde von den Entwicklern und von den Auftraggebern, die im Projektausschuß beteiligt sind, als positiv und förderlich bewertet.

Für die Entwickler besteht der Hauptnutzen darin, daß sie aufgrund der inkrementellen Vorgehensweise die einzelnen Entwicklungsschritte einfacher planen können und ihre Ergebnisse dem Projektausschuß präsentieren können.

Als problematisch hat sich jedoch der Grad der Mitarbeit der Nutzer gezeigt. Die Nutzer sind in der Regel nur in der Lage, das System negativ zu bewerten ("Das ist es aber nicht, was wir uns darunter vorstellen, das ist aber umständlich."). Sollte das Projekt mit veränderten, rein kommerziellen Randbedingungen noch einmal durchgeführt werden, könnte ein System wie der Apple Macintosh die Gestaltung der Oberfläche einfacher machen. Bei einer Neuentwicklung sollte eine durchgängige objektorientierte Modellierung und Entwicklung angestrebt werden.

## 3.4 Ein Dokumentenverwaltungssystem

### 3.4.1 Das Anwendungsgebiet

#### Die Problemstellung

Ein Verlag veröffentlicht im jährlichen Turnus umfangreiche Textdokumente. Kennzeichnend für diese Dokumente ist, daß die Struktur der Texte und zum Teil auch deren Inhalt über mehrere Ausgaben eines Dokumentes konstant bleiben und daß diese Dokumente parallel in mehreren Sprachen erscheinen. Die Verwaltung, die Überarbeitung und die Erstellung der Druckvorlagen ist ein Anwendungsgebiet, das kaum software-technisch erschlossen ist.

Bisher wurde die Datenerfassung und -korrektur im wesentlichen manuell am Satzrechner durchgeführt. Nur in der Produktion wurde eine Lichtsatanlage verwendet. Vergleichbare Systeme in anderen Anwendungsgebieten sind nicht bekannt.

Bei der Entwicklung eines Systems zur rechnergestützten Erstellung von Texten mit den oben beschriebenen Eigenschaften sind unterschiedliche technische Aspekte zu berücksichtigen, beispielsweise:

- Wie wird die Mehrsprachigkeit behandelt? Können identische Textteile in den verschiedenen Sprachen identifiziert und eventuell auf Konsistenz überprüft werden?
- Wie wird das Layout vereinheitlicht? Kann sichergestellt werden, daß gleiche Textteile in verschiedenen Sprachen an der gleichen Stelle im Dokument erscheinen?
- Kann die Datenmenge bewältigt werden? Ist das vom Auftraggeber gewünschte Datenbanksystem geeignet?

Dazu kommen strategische und unternehmenspolitische Fragen wie:

- Der Auftraggeber möchte feststellen, ob sich die Dokumentproduktion bis in die Druckerei rationalisieren läßt.
- Es bleibt zu klären, wie das Datenbanksystem sich durch geeignete Benutzungsoberflächen in die Arbeitsabläufe von Texterfassung und Korrektur bei unterschiedlichen Autoren, beim Verlag und in die Dokumentproduktion in der Druckerei einpassen läßt.

### **Der ausgewählte Lösungsansatz**

Von einem Software-Hersteller wurde ein Prototyp eines Systems entwickelt, das die oben beschriebenen Vorgänge unterstützen soll. Das System sollte primär die funktionalen Eigenschaften besitzen, an die Erstellung einer geeigneten Benutzungsoberfläche war nicht gedacht.

### **Die beteiligten Personengruppen**

Die Realisierung des Prototyps wurde von einem Software-Hersteller, der ca. 30 Mitarbeiter beschäftigt und sich auf diese Art von Anwendungsproblematik spezialisiert hat, durchgeführt. Am Projekt selbst arbeiteten maximal vier Personen gleichzeitig. Die Bewertung des Systems erfolgte in erster Linie durch den Auftraggeber, die späteren Benutzer waren nicht in die Projektentwicklung integriert.

## **3.4.2 Der Entwicklungsprozeß**

### **Charakterisierung des Projekts**

Ausgehend von einer für einen Verlag erstellten Vorstudie erhielt der Software-Hersteller den Auftrag, den Prototyp einer Datenbank für die rationelle Verwaltung und Produktion von Textdokumenten zu entwickeln. Diese Datenbank soll sowohl beim Verlag als auch in der Druckerei erprobt werden, bei welcher die fertigen Textdokumente erstellt werden. Auf Grundlage dieses Prototyps soll dann entschieden werden, ob ein derartiges Anwendungssystem in der Lage ist, die Dokumentproduktion schneller und kostengünstiger abzuwickeln. Der Gesamtaufwand für den Prototyp lag bei ca. zwei Personenjahren. Das Projekt wurde auf der Basis eines Festpreises abgewickelt. Allerdings sah der Vertrag vor, zusätzliche, unvorhergesehene Aufwendungen zu honorieren.

Den Entwicklern wurden im Laufe des Projektes einige wesentliche Probleme und denkbare Lösungen deutlich. Dabei wurde auch klar, daß die Vorstudie, obwohl teils von den Entwicklern selbst erstellt, einige Fragen des Ablaufs, der Funktionalität sowie der Produktionssteuerung offengelassen hatte.

Aus Sicht der Entwickler wurden die Anforderungen, die an den Prototyp gestellt werden, erfüllt. Aus Benutzersicht konnte bisher keine Evaluierung des Systems erfolgen. Sollte der Prototyp aus der Sicht des Auftraggebers die gestellten Anforderungen erfüllen, ist er als Basis für ein zukünftig zu entwickelndes Zielsystem vorgesehen, ein direkter Einsatz des Prototyps als Zielsystem ist hingegen nicht geplant.

Zunächst ist nur die Akzeptanz des Prototyps beim Auftraggeber wichtig. Die Benutzer in der Druckerei werden erst dann in Erscheinung treten, wenn der Auftraggeber den Prototyp akzeptiert hat. Der Prototyp wird sowohl technisch als

auch insbesondere hinsichtlich seines Rationalisierungseffekts getestet. Spezielle Kriterien für die Evaluierung sind den Entwicklern nicht bekannt.

Das Projekt stand zum Zeitpunkt der Berichtabfassung kurz vor der Fertigstellung des Prototyps.

### **Die wesentlichen Aktivitäten und Phasen**

Im Vorfeld des Projekts wurde vom Software-Hersteller eine Studie im Bereich der Dokumentenproduktion ausgearbeitet. Sie umfaßte eine Strukturanalyse sowie einen konzeptionellen Lösungsvorschlag. Die Studie war dafür ausschlaggebend, daß der Auftrag zur Entwicklung des Prototyps erteilt wurde.

Eine eigentliche Erhebungsphase fand zu Beginn der Prototypingaktivitäten nicht statt. Die notwendigen Anforderungen wurden den Aussagen des Auftraggebers und den vorliegenden Dokumenten entnommen.

Im Projektverlauf wurde zunächst nur ein Prototyp geplant. Ein ursprünglich als Frontend geplanter Struktureditor für die Beschreibung von Textdokumenten wird in der Projektlaufzeit vermutlich nicht zur Verfügung stehen.

Der Prototyp sollte helfen, die Systemanforderungen zu klären, vor allem aber diente er zur Spezifikation des Anwendungssystems. Zu einem kleinen Teil wird mit seiner Hilfe die organisatorische Integration des Systems in die Anwendungsumgebung vorbereitet.

Die verarbeiteten Daten wurden in Form von verschiedenen Zwischenergebnissen des Dokuments vom Auftraggeber bereitgestellt; die implementierte Funktionalität wurde von den Entwicklern spezifiziert und vom Auftraggeber bestätigt. Der Auftraggeber hatte feste Vorstellungen, welche Datenbank und welcher Struktureditor verwendet werden sollte.

Die Bewertung des Prototyps findet zunächst beim Auftraggeber und dann auch bei der Druckerei statt, wobei noch unklar ist, ob es dort zu einem wirklichen Piloteinsatz oder nur zu Eignungstests kommen wird.

Der Prototyp dient im wesentlichen zur Überprüfung der oben beschriebenen Fragestellungen. Da der Projektrahmen rs aber gelegentlich zuließ, ließen sich an ihm auch neue Ideen und alternative Lösungsansätze erproben.

Die eigentlichen Probleme der Kommunikation zwischen Benutzern, Entwicklern und Auftraggeber bei der Erprobung oder beim Piloteinsatz des Prototyps werden als noch offen betrachtet.

## **Die verwendeten Werkzeuge**

Den Entwicklern standen eine gut ausgebaute UNIX-Umgebung auf Sun-Arbeitsplatzrechnern sowie MS-DOS-Rechner zur Verfügung. Verwendete Systeme waren die Programmiersprache C, Werkzeuge zur lexikalischen Analyse und zur Compilerkonstruktion, wie LEX und YACC, und die üblichen Serviceleistungen eines lokalen Netzes.

Die Werkzeuge wurden im wesentlichen zur Entwicklung des Prototyps verwendet. Erste Ansätze einer technischen Unterstützung der Projektsteuerung und -kontrolle sind vorhanden.

### **3.4.3 Der konstruierte Prototyp**

#### **Klassifizierung des Prototyps**

Da der Prototyp im wesentlichen die Umsetzbarkeit technischer und organisatorischer Anforderungen demonstrieren soll, handelt es sich hier um einen Prototyp im engeren Sinne. Bezogen auf seinen Ausbau kann man von einem rein funktionalen Prototyp sprechen, der im engeren Sinne alle wesentlichen Aspekte der Funktionskomponente demonstriert, aber keine eigene Benutzungsoberfläche hat. Zu Demonstrationszwecken wird eine Interaktionskomponente dazugesetzt, die aber nicht Gegenstand der Bewertung ist.

#### **Die Merkmale des Prototyps**

Der Prototyp integriert ein vorhandenes relationales Datenbanksystem und soll in einer UNIX-Umgebung ablauffähig sein. Da ein extern erstellter Struktureditor für Textdokumente nicht zur Verfügung steht, wird ein provisorisches Frontend implementiert, das die Steuerung des Prototyps ermöglichen soll. Der Prototyp selbst besteht aus einem Ensemble von Modulen, die auf der Basis eines Konzepts für die Spezifikation von Schnittstellen entwickelt wurden. Die einzelnen Komponenten, wie Eingabe, Datenkorrektur, Datenbankzugriffe und Editiermasken, wurden arbeitsteilig entwickelt.

Gegenüber dem geplanten Anwendungssystem ist der Prototyp eingeschränkt in Datenformaten und Datenmengen sowie in einigen funktionalen Aspekten. Der Prototyp verfügt zusätzlich über eine Ausgabeschnittstelle, damit er einfacher erstellt und getestet werden kann.

#### **Die erstellte Dokumentation**

Die Dokumentation des abzuliefernden Prototyps umfaßt neben den Quelldateien die technische Entwicklungsdokumentation, Schnittstellenbeschreibungen, Konfigurationsdateien, Benutzerhandbuch und Installationsanweisungen.

Schnittstellenbeschreibungen, Konfigurationsdateien und Installationsanweisungen sind allerdings kein Vertragsgegenstand und werden je nach verfügbarer Zeit realisiert. Meilensteindokumente im eigentlichen Sinne existieren nicht.

### 3.4.4 Die Auswertung des Projekts

#### Die charakteristischen Merkmale

Auffällig an der hier beschriebenen Prototyp-Entwicklung erscheinen uns folgende Punkte:

- Das Projekt wurde ohne Anforderungsanalyse im üblichen Sinne initiiert. Statt einer Erhebung der Probleme und Vorstellungen "vor Ort" waren die Entwickler auf schriftliche Dokumente angewiesen.
- Die potentiellen Benutzer des geplanten Systems sind nicht in die Prototyp-Entwicklung bzw. in dessen Bewertung integriert. Ausschlaggebend für die Qualität des Prototyps ist vielmehr einzig die Akzeptanz beim Auftraggeber. Dieses Vorgehen erscheint jedoch folgerichtig, wenn man bedenkt, daß der Prototyp auch auf mögliche Rationalisierungen in der beteiligten Druckerei untersucht werden soll.
- Das Anwendungsproblem stellt weitgehend Neuland dar. Vergleichbare Systeme sind nicht bekannt. Auch die verwendete "Basissoftware" ist in dieser Kombination noch nicht erprobt. Unter diesen Umständen konnte nur eine explorative und experimentelle Vorgehensweise erfolgversprechend sein.
- Der Prototyp ist rein funktionaler Natur. Über die Oberfläche läßt sich anhand des entwickelten Systems keine Aussage treffen, da diese nicht zu den bewerteten Teilen des Prototyps gehört.

#### Die Bewertung von Prototyping für das Projekt

Die Entwickler des Prototypen waren mit dem gewählten Ansatz bis auf Marginalien sehr zufrieden. So würden sie bei der Auswahl von konventionellen Software-Komponenten, die in den Prototyp integriert werden sollen, mehr Vorsicht walten lassen.

Die positive Einschätzung des Verfahrens resultiert sicherlich auch daraus, daß die beteiligten Gruppen, die den Prototyp bisher bewertet haben, sich über die Einschränkungen und die Aussagekraft des Systems im klaren waren. Ob diese positive Einschätzung auch vom Auftraggeber geteilt wird, kann aus heutiger Sicht der Dinge nicht beurteilt werden.

## 3.5 Ein Prozeßsteuerungssystem

### 3.5.1 Das Anwendungsgebiet

Für einen Auftraggeber soll eine Realzeit-Steuerung eines chemischen Prozesses zur Qualitätssicherung entwickelt werden. Die Entwicklung des Überwachungssystems wird von einem mittelgroßen Software-Hersteller als Unterauftrag bearbeitet. Die beteiligten Mitarbeiter verfügen über Erfahrungen im Bereich der Entwicklung von Prozeßsteuerungssystemen und Expertensystemen.

Das Gesamtprojekt war zwischen Auftraggeber und einem Software-Hersteller so angelegt, daß das Überwachungssystem in einzelnen Komponenten erstellt wird. Für die Entwicklung dieser Komponenten war die Vergabe von Unteraufträgen vorgesehen. Als die ersten Vorgespräche mit den von uns interviewten Entwicklern geführt wurden, hatte das Gesamtprojekt bereits begonnen. Das Überwachungssystem wird als Expertensystem entwickelt, wobei der Unterauftragnehmer die Aufgabe hat, die regelbasierte Komponente des Steuerungssystems zu erstellen. Als technische Unterstützung wird dazu eine für solche Steuerungsaufgaben zugeschnittene Entwicklungsumgebung eingesetzt, in der regelbasierte Bausteine bereits enthalten sind.

### 3.5.2 Der Entwicklungsprozeß

#### Charakterisierung des Projekts

Der erste Schritt zur Lösung des oben beschriebenen Problems bestand darin, den Auftraggeber davon zu überzeugen, daß der vorgeschlagene Lösungsansatz tragfähig ist. Zu diesem Zweck wurden zwei kleine Prototypen entwickelt, die den prinzipiellen Aufbau und die Art der Handhabung eines derartigen Überwachungssystems darstellen sollten.

Nach Vertragsabschluß wurde mit einem evolutionären Entwicklungsprozeß mit Prototypen begonnen, die zum Zielsystem ausgebaut werden. Eines der Hauptprobleme in dieser Entwicklungsphase besteht darin, die Arbeit der verschiedenen Projektgruppen so aufeinander abzustimmen, daß die Integration der Komponenten des entstehenden Systems in die gesamte Prozeßsteuerung gewährleistet werden kann.

#### Die beteiligten Personengruppen

Die Entwicklung des Überwachungssystems wird von einer kleinen Gruppe durchgeführt, die Erfahrungen im Bereich des Expertensystembaus und der Prozeßsteuerung mitbringen. Die für die Systemmodellierung notwendigen

Informationen werden von den Entwicklern des Gesamtsystems übermittelt. Unmittelbarer Kontakt zu späteren Systembenutzern findet also nicht statt.

### **Die wesentlichen Aktivitäten und Phasen**

Das Zielsystem wird mittels einer Reihe von Prototypen entwickelt, die in Abschnitt 3.5.3 genauer beschrieben werden. Der letzte Prototyp wird zum Anwendungssystem ausgebaut.

Die einzelnen Entwicklergruppen haben anhand der Prototypen die Schnittstellen der von ihnen entwickelten Systemkomponenten und deren Funktionalität besser aufeinander abstimmen können. Das von uns interviewte Entwicklerteam konnte durch die Prototypen erst den Auftrag gewinnen. Anschließend konnten sich die Entwickler schrittweise ein Problemverständnis erarbeiten.

### **Die verwendeten Werkzeuge**

Ein für die Zwecke der Anlagen- und Prozeßsteuerung spezialisiertes System, basierend auf Lisp und dem Fenstersystem X-Windows, wird eingesetzt. Das System läuft auf einem größeren UNIX-Arbeitsplatzrechner und integriert alle Komponenten (einschließlich einer Dokumentverwaltung) zu einer Entwicklungsumgebung. Diese Entwicklungsumgebung erlaubt schnelle interaktive Design-, Konstruktions- und Bewertungszyklen.

## **3.5.3 Die konstruierten Prototypen**

### **Klassifizierung der Prototypen**

Im Laufe des Projekts wurden mehrere Prototypen entwickelt, von denen der letzte zum Zielsystem ausgebaut werden soll.

#### *Der Demonstrationsprototyp*

Der erste Prototyp hatte hauptsächlich Akquisitionscharakter. Gezeigt werden sollte, wie ein Benutzer interaktiv einen Prozeß überwachen kann. Dazu wurde eine Oberfläche demonstriert, die nichts mit der eigentlichen Problemstellung zu tun hatte, wohl aber einen Eindruck von der prinzipiellen Funktionalität des Systems und seiner Handhabung gab. Da die verwendete Entwicklungsumgebung bereits vorgefertigte Demonstrationssoftware enthielt, lag der Aufwand für diesen Prototyp unter einem Personentag.

#### *Der eigentliche Prototyp*

In rund zweitägiger gemeinsamer Arbeit mit zwei Anwendervertretern erstellte ein Entwickler einen Prototyp, der alle wesentlichen Komponenten des Zielsystems (d.h. Benutzungsschnittstelle, Datenbanksystem, Wissensbasis, Anlagenmodell, Expertenwissen) in minimaler Form realisierte. Das Anlagenmodell wurde durch den Simulator der Entwicklungsumgebung dargestellt. Neben den Diskussionen

mit den Anwendervertretern standen dem Entwickler Protokolle des Anwenders mit typischen Anwendungssituationen zur Verfügung, die als Ist-Analyse gelten konnten. Hier wurde also ein Prototyp im engeren Sinne im Rahmen einer explorativen Analyse verwendet.

#### *Das Pilotsystem*

Nach endgültigem Vertragsabschluß wurde auf der Basis des vorliegenden Prototyps die erste Version eines einsatzfähigen Prototyps entwickelt. Dieser stellt keinen radikalen Bruch zum Prototyp im engeren Sinne dar, da wesentliche Konzepte für die Regelbasis und einige Funktionen übernommen werden konnten. Die Auswertung der explorativen Aktivitäten ergab aber, daß das Anlagenmodell wesentlich verbessert werden mußte. Wichtigste Informationsquellen für diesen Prototyp waren Protokolle von Experteninterviews und die Analyse von Problemfällen mit ihren Folgerungen. Da dieser Prototyp in die Anwendung überführt wird, hat er den Charakter eines Pilotsystems.

Technisch bedeutet die Überführung des Pilotsystem in das Anwendungsgebiet, daß eine Version generiert werden muß, in der die Entwicklungsumgebung weitgehend außer Funktion gesetzt ist. Die Art der verwendeten Entwicklungsumgebung macht es nicht möglich, ein selbständiges Anwendungssystem völlig ohne Laufzeitkomponenten der Entwicklungsumgebung zu generieren.

Aus Zeitgründen wurde keine Spezifikation des Zielsystems vorgenommen, sondern "sofort in die Realisierung" gesprungen.

#### *Die Labormuster*

Bei der Realisierung des Pilotsystems erwies es sich immer wieder als notwendig, im Entwicklungsteam Labormuster zu bauen, um technische Fragen, die sich auf die Struktur der Regelbasis bezogen, beantworten zu können.

### **Die Bewertung durch die Entwickler**

Es zeichnet sich ab, daß die derzeitige Version des Prototyps noch verbesserungswürdig ist. Bisher hat der Auftraggeber noch nicht entschieden, ob deshalb ein weiterer vollständiger Prototyping-Zyklus durchlaufen werden soll, oder ob das Anwendungssystem durch "Nachbesserung" des Prototyps erstellt werden wird.

Dennoch haben nach Einschätzung der Entwickler die Prototypen ihren Zweck erfüllt. Zum einen waren sie, wie gesagt, notwendig, um den Auftrag zu beschaffen, zum andern dienten sie dazu, das Systemverständnis der Entwickler zu verbessern sowie die Problematik der Systemintegration in den Griff zu bekommen.

Entscheidend für die positive Einschätzung war nach Auffassung der Entwickler vor allem die Existenz einer Entwicklungsumgebung, die den Entwicklern bereits vor Projektbeginn vertraut war.

Im Projekt wurde der Schwerpunkt auf die rasche Realisierung des Systems gelegt, hierbei ließen sich die Prototypen sinnvoll einsetzen. Eine systematische Requirementsanalyse fehlte allerdings. Dieses wurde von den Entwicklern negativ bewertet.

Mit geeigneten Werkzeugen ist Prototyping ein Verfahren, das konventionellen Techniken vorgezogen wird. Allerdings müssen nach Ansicht der Entwickler die Randbedingungen stimmen. So sollte beispielsweise ein Entwicklungsvertrag für ein Prototyping-Projekt nicht strikt nach Meilensteinprinzipien organisiert sein.

### **3.5.4 Die Auswertung des Projekts**

Im Bereich der Entwicklung regelbasierter Systeme ist Prototyping schon seit längerem weit verbreitet. Da das hier beschriebene System auch in diese Kategorie fällt, erscheint es uns nur folgerichtig, daß hier verschiedene Arten von Prototypen erfolgreich eingesetzt wurden. Die Art der Problemstellung und der Mangel an vergleichbaren Systemen lassen Prototypen als den einzig gangbaren Weg erscheinen, die Anforderungen an das Zielsystem zu definieren. Zu diesem Zweck wurden sie in diesem Fall auch eingesetzt. Von großem Vorteil war, daß eine zugeschnittene Entwicklungsumgebung eingesetzt werden konnte. Dadurch haben sich die Entwicklungszyklen gegenüber konventionellen Projekten wesentlich verkürzt. Die Anforderungen an die Benutzungsschnittstelle und die Funktionalität des Systems erfordern zudem, daß ein spezialisierter Arbeitsplatzrechner in das Anwendungssystem integriert wird. Bei diesen hohen Anforderungen an die Benutzbarkeit des Systems ist allerdings bemerkenswert, daß die späteren Benutzer nicht unmittelbar in die Entwicklung integriert waren.

Charakteristisch ist auch die Tatsache, daß der Vertrag für die System-Entwicklung nach dem Meilenstein-Prinzip organisiert war, obwohl die Entwickler Erfahrungen mit dem Bau von Prototypen hatten und wußten, daß eine solche Vertragsgestaltung nur sehr eingeschränkt sinnvoll ist. Allerdings entspricht die Vorgehensweise insgesamt der Entwicklungsphilosophie des Software-Herstellers. Generell wird dort entlang eines Phasenmodells mit Meilensteinen entwickelt, wobei in den einzelnen Phasen Prototypen verwendet werden können.

Interessanterweise fällt das hier analysierte Projekt partiell aus diesem Entwicklungskonzept heraus. Wir betrachten dies als einen Hinweis dafür, daß zum einen derartige Richtlinien zur Systementwicklung innerhalb eines Software-Herstellers von der Art und Aufgabenstellung abhängig gemacht werden sollten, zum anderen, daß bestimmte Probleme im Anwendungsgebiet und bei der technischen Realisierung von Software zu prototypischem Vorgehen zwingen.

## 4. Analyse der Projekte und Konsequenzen für das Prototyping

Die untersuchten Projekte zeigen jeweils spezifische Eigenschaften und Auswirkungen von Prototyping auf die Software-Entwicklung. Wir haben versucht, die Projekte so zu analysieren, daß die erhaltenen Ergebnisse verallgemeinert werden können und somit auch für andere Projekte relevant sind. Die Ergebnisse der Analyse haben wir logisch zusammengefaßt. Wir haben dabei festgestellt, daß existierende Ansprüche teils erhärtet werden, teils aber auch relativiert werden müssen. Weiterhin haben wir Aspekte erkannt, die bisher entweder kaum oder gar nicht bei der Diskussion von Prototyping berücksichtigt wurden. Nachfolgend werden die Ergebnisse unserer Analyse wiedergegeben.

### 4.1 Die Grenzen des Prototyping

Als grundlegendes Problem erweist sich, daß vor allem die Auftraggeber nur sehr wenig über den Nutzen und die Leistungen von Prototypen wissen. Die Erwartungen, die sich an einen ersten Prototyp mit reinem Demonstrationscharakter knüpfen, sind oft überhöht. Insbesondere wenn der Prototyp die Benutzungsschnittstelle zusammen mit einer rudimentären Funktionalität zeigt, kann der Eindruck entstehen, daß bereits der größte Teil der Arbeit geschafft ist. Wenn die Leistung des Prototyps nicht von vorneherein klargestellt ist, bleibt häufig unklar, welche Aspekte an einem Prototyp jeweils bewertet werden können. So muß den Anwendern und Benutzern verdeutlicht werden, daß es nicht sinnvoll ist, über die Effizienz eines Prototyps zu reden, der nur den Dialogablauf in seiner Struktur modellieren soll. Ein häufiges Problem dieser Art ist die Bewertung eines reinen Oberflächen-Prototyps. Da keine Arbeitshandlungen im eigentlichen Sinne erprobt werden können, sind Aussagen über die Verwendbarkeit des Anwendungssystems nur begrenzt möglich.

Es hat sich weiterhin gezeigt, daß die Entwickler über ausreichende Kenntnisse im Anwendungsgebiet und in der Einsatzumgebung verfügen müssen, damit das Verhalten des Prototyps nicht soweit von den Zielvorstellungen entfernt ist, daß er für eine Weiterentwicklung ungeeignet ist. Eine Systemanalyse ist somit die Basis für ein erfolgreiches Prototyping. An dieser Analyse muß ein Fachmann des Anwendungsgebiets beteiligt sein. Die Analyse sollte ebenfalls eine einheitliche Terminologie für Entwickler und Anwender schaffen.

Wir finden gelegentlich die Situation, daß ein Entwicklerteam ausschließlich auf der Basis vorliegender Dokumente und schriftlicher Anforderungen in ein Prototypingprojekt einsteigt. Zu den Problemen, die sich aus der mangelnden

Einarbeitung in die tatsächliche Anwendungssituation ergeben, kommt noch das Problem der impliziten Einschränkung der Begriffsbildung und der Lösungsvorstellungen. Derartige Dokumente enthalten bereits eine Vorauswahl der Terminologie und geben oft eine Richtung für die technische Realisierung vor. Diese Vorgaben sind aber meist ohne Beteiligung der Benutzer gemacht worden, wodurch das Potential des Prototyping eingeschränkt wird. Gelegentlich gehen diese Vorgaben soweit, daß konkrete Entscheidungen für die Auswahl von Hardware und Software bereits vor dem eigentlichen Projektbeginn gefallen sind. So verständlich dies aus Sicht der Auftraggeber sein mag, so problematisch erweist sich diese Fixierung meist mit Blick auf die technischen Anforderungen, die erst im Prototypingprozeß deutlich werden.

Sind diese Punkte nicht berücksichtigt, kann auch Prototyping den Mißerfolg eines Projektes nicht verhindern, wie das folgende Beispiel zeigt:

Für einen externen Auftraggeber sollte eine Überwachungssteuerung für Werkzeugmaschinen entwickelt werden. Diese Steuerung sollte mittels eines regelbasierten Systems auf einer vorgegebenen Zielmaschine realisiert werden. Die Werkzeugmaschinen wurden bereits durch ein DV-System gesteuert. Im Projekt war eine Folge von Prototypen vorgesehen, deren erster sowohl die Benutzungsoberfläche als auch wesentliche Teile der Funktionalität zeigen sollte. Zu seiner Realisierung wurde eine Expertensystem-Shell auf einem Arbeitsplatzrechner eingesetzt.

Die Auswertung des Prototyps machte eine Reihe von Problemen deutlich:

- Der Prototyp war ohne ausreichenden Kontakt zwischen Auftraggebern, Benutzern und Entwicklern entstanden. Dies führte dazu, daß Anforderungen der Benutzer an das System nicht ausreichend berücksichtigt wurden. Auch das Ausbildungsniveau der späteren Benutzer wurde bei der Entwicklung des Prototyps vernachlässigt.
- Die Zielvorstellungen der Auftraggeber konnten den Entwicklern nicht hinreichend genau vermittelt werden. Deshalb klaffte eine Lücke zwischen den Leistungen des Prototyps und den Wünschen der Auftraggeber.
- Die Frage der Integration des Anwendungssystems in die vorhandene Arbeitsumgebung war nicht ausreichend bearbeitet worden. Im Laufe der Prototyp-Entwicklung stellte sich heraus, daß die vom Auftraggeber verlangte Hardware-Umgebung die Anforderungen an die Benutzungsschnittstelle nicht erfüllen konnte.

Die Auswertung des Prototyps führte zu der Entscheidung, das gesamte Projekt abzubrechen. Der bis zu diesem Zeitpunkt bereits investierte Aufwand belief sich auf ungefähr die Hälfte des geplanten Gesamtaufwands.

Nach unserer Analyse lag ein wesentlicher Grund für die beobachteten Probleme in der fehlenden Kommunikation zwischen den beteiligten Gruppen. Es wurde keine Verständigung erreicht und wesentliche Vorstellungen des Auftraggebers (Einbindung des Systems in ein CIM-Gesamtsystem) wurden während der Entwicklung nicht erkannt.

Obwohl das Projekt abgebrochen wurde, hatte der Prototyp einen wichtigen Zweck erfüllt, da er zu einem relativ frühen Zeitpunkt die grundlegenden Fehler im Ansatz offenbarte. Diese Erfahrung trug mit dazu bei, daß die betroffene Firma ihre Strategie der Systementwicklung revidierte und vor allem die Einbindung der späteren Systemnutzer in die Systementwicklung forcierte.

## 4.2 Die Beteiligung der Anwender bei der Entwicklung

Ein weiteres durchgehendes Mißverständnis ist mit der Frage verbunden, wie und inwieweit der Benutzer in den Entwicklungsprozeß einbezogen werden kann. Viele Anwender scheuen sich, die tatsächlichen Endbenutzer eines Anwendungssystems in die Auswertung von Prototypen oder gar die Diskussion um Gestaltungsalternativen mit einzubeziehen. Dies läßt sich oft auf zwei Gründe zurückführen

- Häufig ist das implizite (d.h. vom Anwendermanagement nicht ausgesprochene) Ziel eines Entwicklungsprojektes die Reduktion von Arbeitsplätzen durch Rationalisierung. Um entsprechende Informationen darüber so lange wie möglich von den Betroffenen fernzuhalten, werden sie erst gar nicht in den Entwicklungsprozeß einbezogen.
- Unabhängig davon besteht gerade bei Vertretern des mittleren Anwendermanagements die Ansicht, die Benutzungssituation und mögliche Designalternativen besser einschätzen zu können, als die späteren Benutzer des Systems. Dies mag im Einzelfall durchaus zutreffen. Aber oft erscheint uns diese Haltung als eine Variante der traditionellen Entwicklereinstellung - "Wir wissen, was die Benutzer eigentlich wollen".

Andererseits haben viele Entwickler zu hohe Erwartungen an die Kreativität und die innovativen Ideen der Benutzer, was die technische Gestaltung eines Prototyps anbetrifft. Dies gilt vor allem für diejenigen Prototypen, die unter "Laborbedingungen" von den Benutzern bewertet werden. Da den Benutzern die tägliche Erfahrung im Umgang mit dem System ja noch fehlt, sind sie selten in der Lage, Vorschläge für die Gestaltung noch nicht vorhandener technischer Aspekte zu machen. Ihre Anregungen werden sich in der Regel auf eine Kritik des bereits Vorhandenen beziehen. Dies ändert sich meist, wenn ein Pilotsystem am Arbeitsplatz eingesetzt wird. Dann werden neben störenden und funktionsuntüchtigen Merkmalen eines Prototyps auch solche Aspekte auffällig, die zur Erledigung einer Arbeitsaufgabe fehlen. In dieser Frage sollten die Entwickler

ihrerseits nicht in den Fehler verfallen, die Benutzer bei der Bewertung eines Prototyps zur Formulierung beliebiger Wunschvorstellungen zu ermuntern, sodaß alle nur irgendwie vorstellbaren Funktionen oder Oberflächengestaltungen als Anforderungen aufgenommen werden. In diesem Zusammenhang soll auch erwähnt werden, daß Benutzer aufgrund von Grundkenntnissen der Programmierung gelegentlich dazu neigen, sich bereits als Software-Experten einzuschätzen und den Entwicklern Vorschläge zur Systemimplementierung machen zu wollen. Dies ist so wenig sinnvoll, wie die verbreitete Entwicklerhaltung, den Benutzern ihre "eigentlichen" Anforderungen klarmachen zu wollen.

Einen wichtigen Beitrag zur Klärung der Anforderungen an ein Anwendungssystem können Szenarien spielen, die von den Benutzern erstellt werden. Darin werden relevante gegenwärtige oder erwünschte Arbeitssituationen so beschrieben, daß Handlungen, Arbeitsmittel und Motivation deutlich werden. Wir haben mehrfach festgestellt, daß sich Entwickler durch solche Szenarien einen detaillierten Einblick in Anwendungsprobleme verschaffen konnten.

### **4.3 Die Auswirkungen von Prototyping auf die Teamarbeit**

Bei der Analyse erfolgreicher Entwicklungsprojekte zeigt sich, daß trotz der verbesserten Kommunikationsmöglichkeiten durch das Prototyping die Einarbeitung der Entwickler in die Probleme und Arbeitsweisen des Anwendungsfeldes von zentraler Bedeutung ist. Hier bestätigt sich erneut, daß Software-Entwicklung ein Lernprozeß aller beteiligten Gruppen ist. Dieses Lernen findet auf Seiten der Entwickler dann unter optimalen Bedingungen statt, wenn im Entwicklerteam Personen mit einer Doppelqualifikation als Software-Techniker und als Anwendungsspezialisten sind. Ähnliches gilt umgekehrt für Benutzer, die neben ihrem Arbeitsgebiet auch Erfahrungen und Kenntnisse in Computeranwendungen besitzen. Während es heute allgemein gängig ist, von einer notwendigen "Computeralphabetisierung" der Benutzer zu sprechen, ist noch keine systematische Förderung der Einarbeitung der Entwickler in wechselnde Anwendungsbereiche abzusehen. Schritte in diese Richtung bleiben Ausnahmen.

Wenn wir die aktive Einbeziehung aller relevanten Gruppen - Entwickler, Benutzer, Fachingenieure, Anwendermanagement - als essentiell für ein wirklich erfolgreiches Prototyping-Projekt ansehen, dann bedeutet dies auch, daß der räumliche Zusammenhang zwischen diesen verschiedenen Gruppen angestrebt werden sollte. Unsere Erfahrungen besagen, daß trotz aller Möglichkeiten der Telekommunikation (wie Fax, Electronic Mail oder Telefon) der persönlichen Diskussion und Abstimmung herausragende Bedeutung für die Herausbildung einer Projektidentität und Projektkultur zukommt.

#### 4.4 Die Planung von Projekten mit Prototyping

Wir haben den Eindruck gewonnen, daß ein schrittweises oder evolutionäres Prototyping die Planbarkeit eines Entwicklungsprozesses gegenüber traditionellen Vorgehensweisen merklich erhöht. Dies gilt auch in den Bereichen, die durch den innovativen Charakter des angestrebten Anwendungssystems allgemein als sehr schwer abschätzbar gelten. Einzelne gegenläufige Erfahrungen dürfen nicht ignoriert werden. Allerdings sollte man genauer hinsehen. Oft sind Preise und Termine in Software-Projekten eher "unternehmensstrategisch" kalkuliert, da Anschlußprojekte oder die Ausschaltung von Konkurrenten das Kalkül beeinflußt haben.

Prototypingprojekte werden oft in Gebieten von technischem und anwendungsbezogenen Neuland durchgeführt. Dabei sollen häufig neue Geräte oder Software-Systeme integriert werden. Dies kann ein ausdrückliches Projektziel sein ("Wir wollen Laptops für die Kundenberatung ausprobieren"), dies kann aber auch organisatorisch-technische Gründe haben ("Wir wollen kein eigenes Fenstersystem entwickeln"). Damit machen sich die Beteiligten - vor allem die Entwickler - aber von projektexternen Anbietern abhängig. Leider fanden wir häufig die Situation vor, daß Liefertermine nicht eingehalten wurden oder daß das gelieferte Produkt nicht seiner Beschreibung in der Ankündigung oder sogar im Liefervertrag entsprach. Dies soll kein Plädoyer für eine konservative Hardware- und Software-Strategie sein. Wichtig scheint uns aber, daß solche Faktoren bei der Projektplanung berücksichtigt werden. So kann beispielsweise ein Prototyp einer Datenbankanwendung zunächst ohne die persistente Datenhaltung im gewünschten Datenbanksystem erstellt werden und dann nach erfolgreicher Installation an dieses Datenbanksystem angeschlossen werden. Ebenso kann die Schnittstelle zu einem Fenstersystem von vorneherein so allgemein sein, daß eine Alternative gewählt werden kann, wenn das gewünschte Fenstersystem doch nicht den Anforderungen entspricht.

Der durchgängige Einsatz von Prototypen harmonisiert schlecht mit den Meilenstein-Dokumenten eines klassischen Phasenkonzepts. Eine Planung und Projektsteuerung entlang sog. Meilensteindokumente, die sich im wesentlichen an das Management richten, erweist sich oft als hinderlich. Einzelne bezahlte und dokumentierte Aktivitäten, die im Entwicklungsprozeß selbst wesentlich sind, wie ein Prototyp im Rahmen der Anforderungsanalyse, ein formalisiertes Begriffslexikon, etc. erscheinen angemessener. Aber auch Prototyping-spezifische Aktivitäten, wie die Reimplementierung des initialen Prototyps, soweit er Akquisitionscharakter hat und nicht auf Wiederverwendbarkeit hin konstruiert worden ist, sollten im Vertrag vorgesehen werden.

In einer solchen schrittweisen Vertragsgestaltung, die sich am Projektstand und den klarer werdenden Vorstellungen und Lösungen orientiert, liegen neue Chancen für die Auftraggeber von Entwicklungsprojekten. Dadurch werden

nämlich bereits zu frühen Zeitpunkten im Projektverlauf bessere Entscheidungsgrundlagen für die Projektsteuerung an die Hand gegeben. In vielen Fällen läßt sich mit vertretbarem Aufwand ein Projekt abbrechen, daß nicht die gewünschte Richtung nimmt.<sup>0</sup>

## 4.5 Die Dokumentation

Die Dokumentation ist nach wie vor das Stiefkind der Software-Entwicklung. Daran ändert auch das Prototyping nichts. Auch wenn es zunächst so scheint, als mache Prototyping schriftliche Dokumente weitgehend überflüssig, da der Prototyp als funktionstüchtiges Objekt eine Beschreibung ersetzt, so werden die Folgen einer mangelhaften Dokumentation auch in der Praxis rasch deutlich. Da ist zum einen die unverzichtbare Benutzungsdokumentation, ohne die ein System praktisch kaum eingesetzt werden kann. Dazu kommt noch die Entwicklungsdokumentation, welche die Architekturbeschreibungen und relevante Designentscheidungen festhält, ohne die eine sinnvolle Weiterentwicklung von Prototyp oder Anwendungssystem sehr aufwendig wird.

Die Haltung der Auftraggeber zu Fragen der Dokumentation ist recht unterschiedlich. Wir finden Fälle, in denen die Auftraggeber im wesentlichen auf einen lauffähigen Prototyp als Projektergebnis Wert legen und bei knapp werdenden Ressourcen dann eher auf die Dokumentation verzichten. Andere Auftraggeber fordern die Einbindung der gesamten Projekt- und Produktdokumentation in bestehende Standards und ggf. vorhandenen Data Dictionaries. Während der erste Fall von einer mangelnden software-technischen "Reife" des Auftraggebers zeugt, bringt der zweite Fall oft ebenfalls Probleme. Wir haben zum einen darauf hingewiesen, daß Meilensteindokumente vielfach der Idee des Prototyping widersprechen. Zum anderen erzwingen viele Data Dictionaries eine "datengetriebene" Systembeschreibung, die unverträglich mit der tatsächlich gewählten Software-Architektur des Prototyps sein kann.

Die höhere Benutzerbeteiligung bei Prototyping-Projekten ermöglicht gelegentlich, daß die Benutzerdokumentation von den Benutzern selbst oder unter ihrer Federführung erstellt wird. Dies wird im Ergebnis von allen Beteiligten positiv bewertet. Hier liegt auch ein Aufgabenbereich, der von solchen Benutzervertretern gut bearbeitet werden kann, die einem Entwicklerteam fest zugeordnet sind.

---

<sup>0</sup>siehe dazu das in Kapitel 4.1 beschriebenen Projekt.

## 4.6 Der Stellenwert von Prototypen

In der Literatur ist der Rolle von Demonstrationsprototypen für die Projektakquisition bisher kaum Bedeutung geschenkt worden. Dem stehen die einschlägigen praktischen Erfahrungen gegenüber. Solche Demonstrationsprototypen können je nach Möglichkeiten und verfügbarer Zeit sehr unterschiedlich gestaltet sein

- als reine funktionslose Fassade (mock up) mit simulierten Benutzerinteraktionen,
- als Demonstration der Handhabung des künftigen Systems an einem Beispiel aus einem völlig anderen Anwendungsbereich,
- als erster Prototyp, der als funktionstüchtige Skizze des Anwendungssystems betrachtet werden kann, aber software-technisch eher ein Wegwerfprodukt ist.

Es ist nicht nur begrifflich wichtig, diese vorläufigen Demonstrationsprototypen von den weiteren Versionen zu trennen. Erfahrungen zeigen, daß die Übernahme solcher Prototypen als technische Bausteine im weiteren Entwicklungsprozeß problematisch, wenn nicht sogar fatal für das gesamte Projekt ist, da sie aufgrund ihres Charakters keine solide Grundlage für eine qualitativ hochwertige und problemangemessene Software-Architektur sein können. Demonstrationsprototypen sollten deshalb explizit als Wegwerfprodukte eingeplant werden, damit aus den Vorteilen des Prototyping kein Nachteil für das spätere Anwendungssystem wird.

Ebenso wenig ist bisher aufgearbeitet worden, welche Bedeutung Labormuster für kreative Lösungen und Designalternativen besitzen. Die Erfahrungen zeigen, daß in diesem Bereich bisher ein großes Potential für innovative Ideen unberücksichtigt blieb. In allen Projekten, in denen die Entwickler über das Tagesgeschäft hinaus die Möglichkeiten hatten, mit Labormustern experimentell und "spielerisch" umzugehen, sind Entwürfe und Konzepte entstanden, die zwar oft nicht in die aktuelle Systemversion eingegangen sind, die aber Grundlage für weitere Revisionen sind oder sein können. Wir gewannen zudem den Eindruck, daß die Konstruktion und Bewertung von Labormustern den Entwicklern viel mehr Übersicht und Maßstäbe für die Beurteilung der jeweils im Anwendungssystem realisierten Lösungen gibt.

Unsere Untersuchung zeigt, daß eine einheitliche Vorgehensweise beim Prototypenbau - etwa von der Systemoberfläche "in die Tiefe"- nicht sinnvoll ist. Ausschlaggebend scheint uns die Art der Problemstellung. Wenn die Funktionalität eines Anwendungssystems weitgehend vorgegeben ist und bereits Erfahrung im Einsatz von Computern für ähnliche Fragestellungen vorliegt, dann empfiehlt sich diese oft propagierte Form des "horizontalen Prototyping".

Andererseits zeigt eine Fallstudie, daß bei weitgehend unklarer Funktionalität eines Systems diese der Ansatzpunkt für das Prototyping sein wird, wobei gelegentlich auf die Modellierung der Systemoberfläche im Rahmen eines solchen Projektes ganz verzichtet werden kann.

#### **4.7 Abschließende Bemerkungen**

Die Untersuchung und die Analyse der Projekte haben ergeben, daß Prototyping, insbesondere wenn es mit einer evolutionären Entwicklungsstrategie gekoppelt ist, für die Qualität des Produkts, aber auch für die Qualität des Entwicklungsprozesses förderlich ist. Es hat sich weiterhin gezeigt, daß der positive Effekt von Prototyping dann gesteigert werden kann, wenn alle am Entwicklungsprozeß beteiligten Personengruppen die Leistungen und die Grenzen dieses Ansatzes kennen und wenn die organisatorischen Rahmenbedingungen sowie die Vertragsgestaltung auf dieses Entwicklungsmodell zugeschnitten sind.

Von grundlegender Bedeutung ist schließlich der Einsatz geeigneter und von den Entwicklern beherrschter Werkzeuge zur Unterstützung des Prototyping. Erst dadurch sind in vielen Fällen die technischen und organisatorischen Randbedingungen gegeben, um Prototyping durchführen zu können. Sicherlich gibt es Situationen, in denen auch mit vorhandenen traditionellen Hilfsmitteln Prototypen mit großem Erfolg erstellt werden, doch ermöglicht erst eine entsprechende Entwicklungsumgebung die schnellen Entwicklungszyklen, die Prototyping attraktiv und effizient werden lassen.

## Literatur

- Bischoffberger, W., Keller, R. (1989): Enhancing the Software Life Cycle by Prototyping. **Structured Programming**, Vol. 10, No. 1, pp 47-59.
- Boehm, B.W. (1976): Software Engineering. **IEEE Transactions on Computers**, Vol. 25, No.12), 1976, pp 1226-1241.
- Boehm, B.W., Gray, T.E., Seewaldt, T. (1984): Prototyping versus Specification: A Multiproject Experiment. **IEEE Transactions on Software Engineering**, Vol. SE-10, No. 3, pp 290-302.
- Budde, R., Kuhlenkamp, K., Mathiassen, L., Züllighoven, H. (eds) (1984): **Approaches to Prototyping**, Springer-Verlag.
- Budde, R., Kuhlenkamp, K., Sylla, K.H., Züllighoven, H. (1987): Konzepte des Prototyping. In **Requirements Engineering '87**, GMD-Studien Nr. 121, Gesellschaft für Mathematik und Datenverarbeitung mbH, Sankt Augustin, pp 75-94.
- Budde, R., Züllighoven, H. (1990): Prototyping Revisited. In **Computer Systems and Software Engineering**, IEEE Computer Society Press, Tel Aviv, Israel.
- Diederich, L.G., Milton, J. (1987): Experimental Prototyping in Smalltalk-80. **IEEE Software**, May 1987, pp 50-64.
- Doberkat, E., Fox, D. (1989): **Software Prototyping in SETL**. Teubner-Verlag, Stuttgart, 226 S.
- Duncan, A. (1982): Prototyping in Ada: A Case Study. **ACM SIGSOFT Software Engineering Notes**, Vol. 7, No. 5.
- Floyd, C. (1984): A Systematic Look at Prototyping. In: Budde et al. (eds) **Approaches to Prototyping**, Springer-Verlag, pp 1-19.
- Mayr, H.C., Bever, M., Lockemann, P.C. (1984): Prototyping Interactive Application Systems. In: Budde et al. (eds) **Approaches to Prototyping**, Springer-Verlag, pp 105 - 122.
- Rzevski, G. (1984): Prototypes versus Pilot Systems: Strategies for evolutionary Information System Development. In: Budde et al. (eds) **Approaches to Prototyping**, Springer-Verlag, pp 341-356.

## Die Autoren

### **Antoinette Kieback**

30 Jahre alt. 1980 bis 1986 Studium und Diplom in Informatik an der Universität Stuttgart.

Seit 1986 Mitarbeiterin der Dornier GmbH in Friedrichshafen. Von 1986 bis 1990 Leiterin und Mitarbeiterin in den ESPRIT-Projekten TODOS und HECTOR. Seit 1988 Leiterin des GI Arbeitskreises 4.3.2 "Wissensbasierte Systeme für das Prototyping".

Arbeitsgebiete: Prototyping in Entwurfswerkzeugen, Zugriffssicherheit in Informationssystemen.

### **Horst Lichter**

30 Jahre alt. 1986 Diplom in Informatik an der Universität Kaiserslautern.

Von 1986 bis 1988 Mitarbeiter an der Eidgenössischen Technischen Hochschule Zürich, seit 1988 Mitarbeiter am Institut für Informatik, Abteilung Software-Engineering, Universität Stuttgart.

Arbeitsgebiete: Software-Engineering, Prototyping, objektorientierte Programmierung.

### **Matthias Schneider-Hufschmidt**

33 Jahre alt, Studium an der Universität Stuttgart und am M.I.T., Cambridge, USA. 1981 Diplom in Informatik, 1986 Promotion zum Dr. rer. nat an der Universität Stuttgart.

Seit 1987 Mitarbeiter in der Abteilung "Zentrale Forschung und Entwicklung" ZFE IS SOF der Siemens AG in München.

Arbeitsgebiete: Software Engineering-Methoden, Mensch-Computer-Kommunikation, Prototyping, verteilte Systeme.

### **Heinz Züllighoven**

41 Jahre alt. 1977 Staatsexamen Mathematik und Germanistik an der Universität Bonn, 1989 Promotion zum Dr.-Ing. an der TU Berlin.

Seit 1974 Mitarbeiter der Gesellschaft für Mathematik und Datenverarbeitung (GMD Bonn), Institut für Systemtechnik. Seit 1986 Lehrbeauftragter an der TU Berlin.

Arbeitsgebiete: Entwicklung von Programmierumgebungen, Methodik des Software-Entwurfs und der Programmierung, Prototyping.