

# Informatik und Arbeitsumgebungen

## *Szenario*

Ich liege hier am Strand von Salalah in tropischer Sonne und schaue aufs Meer. Dieser Ort, für mich ein Urlaubsparadies, ist für die Inder, Pakistanis und Omanis, die zum Servicepersonal des Hotels gehören, ihr Arbeitsplatz. Vor knapp einer Woche wurde hier im Hotel ein neues Computersystem zusammen mit einem neuem Kassen- und Abrechnungssystem eingeführt. Für einen Gast hat das folgende Auswirkungen:

Ich wollte meine bisher angefallenen Rechnungen bezahlen. Ich ging also zur Rezeption and brachte meinen Wunsch vor. »You gave something to the laundry ?« fragte der junge Omani nach einem Blick auf den Bildschirm. »Yes and I got it back this afternoon.« war meine Antwort. So druckte mir der

Mann eine Rechnung aus, die nur den Wäschereiposten enthielt. Ich hatte aber abends noch Bier getrunken und die eine oder andere Wasserpfeife geraucht. Es mußten noch Rechnungen vom Restaurant da sein. Aus irgendeiner Ecke klaubte dann der Mann einen Stapel Rechnungsbelege hervor. Glücklicherweise hatte ich meine arabische Unterschrift benutzt und so waren meine Belege leicht herauszufischen. Nun wechselte der Mann irritierte Blicke zwischen Bildschirm und Zetteln. Irgend etwas schien da nicht zu stimmen. Schließlich bat er mich, in einer halben Stunde wiederzukommen.

Eine knappe Stunde später wagte ich mich wieder zur Rezeption. Inzwischen saß der junge Omani zusammen mit einem anderen Herrn vor dem Rechner. Die ausgedruckte Rechnung, die er jetzt

in der Hand hielt, enthielt die Wäsche, die Biere, die Wasserpfeifen und drei Dinner. Ich hatte aber Halbpension gebucht und somit gehörten die Dinner da nicht hin. »Why are these meals on it ?« rief der Omani verzweifelt. Der Spezialist neben ihm tat noch ein paar Handgriffe und endlich hatte ich nach über einer Stunde die korrekte Rechnung in der Hand und konnte bezahlen.

## **Systementwicklung und Arbeitsprozesse**

Was zeigt uns nun dieses Beispiel? Das System, das im Hotel installiert wurde, arbeitet sicher korrekt. Korrekt in dem Sinne, daß das System die Funktionalität ausführt, die in der Spezifikation beschrieben wurde. Wer schreibt aber

diese Spezifikation und was beschreibt sie?

In dem Hotel in Salalah sind fast alle Mitarbeiter in den Abrechnungsprozeß eingebunden. Fast jeder muß entweder Belege ausfüllen, sammeln, bestätigen, weiterreichen oder sortieren. Die Arbeitssprachen im Haus sind Englisch und Arabisch, die aber nicht unbedingt alle Mitarbeiter des Hauses verstehen. Zudem ist das Bildungsniveau der Mitarbeiter sehr inhomogen. So ist es wohl vorstellbar, wie schwierig es ist, diesem Team die Handhabung der Belege für das neue System zu erklären, sofern überhaupt jemand im Vorfeld über mögliche notwendige Änderungen nachgedacht hat. So entsteht also eine lange Kette von Trials und Errors, bis alles reibungslos abläuft.

Solch ein Anwendungssystem soll in der Regel einen Arbeitsprozeß unterstützen. Das System wird sogar zu einem Teil dieses Arbeitsprozesses und muß daher in diesen eingegliedert werden. Wird ein neuer Mitarbeiter im Hotel eingestellt, so wird dieser in die bestehenden Prozesse eingewiesen. Wird aber eine Computeranwendung eingeführt, so müssen sich die Mitarbeiter mit ihrem Arbeitsprozeß der Anwendung anpassen.

Es stellt sich also die Frage, warum die IT-Anwendungen nicht den bestehenden Prozessen angepaßt werden. Diese Frage ist sehr vielschichtig. Standardanwendungen, die nur mit minimalsten Anpassungen überall laufen sollen, können gar nicht Spezialanforderungen eines einzelnen Kunden erfüllen. Häufig werden Anwendungen auch gerade eingeführt, weil man den bestehenden Arbeitsprozeß effektiver gestalten und damit verändern möchte. Bis zu einem gewissen Grad ist eine Änderung des Arbeitsprozesses mit der Einführung einer neuen IT-Umgebung also geplant.

Es gibt diverse Möglichkeiten, wie es zu der Entscheidung, etwas Neues einzuführen, kommt. Der Impuls kann von den Endbenutzern (z.B. der Planungsabteilung) oder auch von der Geschäftsführung des Kunden kommen. Als nächstes muß grundsätzlich entschieden werden, ob man Standardsoftware einführen und den Arbeitsprozeß ggf. darauf abstimmen will, oder eine Speziallösung in Auftrag geben möchte. In letzterem Fall muß eine Spezifikation für die neue Anwendung erstellt werden, die den Einsatzbereich und die äußere

Funktionalität beschreibt. In beiden Fällen kann ein eigenes Beratungs- oder Studienprojekt nötig sein, das eine Analyse der Geschäftsprozesse beinhaltet. Nicht immer ist den Beteiligten (Endbenutzer, Management des Kunden, Entwickler/Designer des Systems, IT-Bereich des Kunden und ggf. Berater) bei der Auswahl eines Systems oder der Erstellung der Spezifikation für das System bewußt, wie weitreichend der Einfluß auf den Arbeitsprozeß ist. So kommt es, daß der Arbeitsprozeß bei der Einführung eines neuen Systems häufig an mehr Stellen angepaßt werden muß als geplant. Typische Beispiele dafür sind:

Das System braucht einen neue Belegtypen oder Daten, die in dem alten Prozeß bisher nicht benötigt wurden. Niemand überlegt sich, wie diese Belege/Daten im bestehenden Prozeß erstellt und behandelt werden. Das neue System im Hotel in Salalah z.B. braucht für die interne Kostenrechnung der Restaurants auch Belege für die Dinner der Halbpensionsgäste, um die ausgegebenen Essen gegenüber dem Hotel abzurechnen. Diese Belege sind vielleicht eingeführt worden, um die Restaurants vom Hotel stärker zu entkoppeln und damit deren Wirtschaftlichkeit transparenter zu machen oder die Restaurants wirtschaftlich ausgliedern zu können. Diese Belege sind aber vom Personal anders als andere Restaurantrechnungen zu behandeln. Insbesondere sollte der Gast diese Belege nicht unterschreiben müssen, da er sie ja nicht extra bezahlt.

Das System gibt eine zu starre Arbeitsabfolge vor. Bei der Erstellung einer Spezifikation versuchen die Entwickler oder Berater herauszufinden, was typische und häufige Arbeitsfolgen sind. Entsprechend werden dann die Bildschirmmasken erstellt. Die Softwareentwickler bekommen aber selten ein Gefühl dafür, wie häufig von einer Standardabfolge abgewichen wird oder wie stark sich die Abfolge bei verschiedenen Sachbearbeitern unterscheidet. Häufig wird die Abfolge, in der Masken auszufüllen sind, zu stark fixiert. Das macht es dem Sachbearbeiter später unmöglich, seine individuelle Reihenfolge beizubehalten. Die Sachbearbeiter bewerten dann das System als umständlich, was die Akzeptanz neuer Systeme erschwert.

Ein großes Problem kann die Datenqualität sein, insbesondere wenn ein

Arbeitsvorgang zum ersten Mal vom Rechner unterstützt wird. Stellen wir uns einen Produktionsbetrieb vor, in dem die Maschinenbelegung heute noch per Hand geplant wird. Die Produktionsplaner haben ein sehr umfangreiches Wissen über die internen Prozesse und Kapazitätsdaten. Sollte in der Datenbank mal ein fehlerhaftes Datum stehen, so kann es im Verlauf der Planung per Hand korrigiert werden. Die Firma entschließt sich nun, die Planung der Maschinenbelegung durch eine IT-Anwendung zu unterstützen. Häufig wird übersehen, daß die IT-Anwendung nicht alle Korrekturen an den Daten vornehmen kann und darf, die der Planer aufgrund seiner jahrzehntelangen Erfahrung intuitiv und aus dem Stegreif macht. Damit sind die Anforderungen an die Konsistenz der Daten in den Datenbanken, die dieses System mit Input versorgen, nun sehr viel höher als es vorher für die ›Handplanung‹ nötig war. Die Korrekturen, die die Produktionsplaner bisher während der Planung nebenher machten werden nun ein extra Arbeitsschritt vor dem Starten der Planungsanwendung im Rechner. Selten ist den Kunden bewußt, wieviel kleine Korrekturen nebenher vorgenommen werden. Eine Bereinigung der Daten en bloc durchzuführen ist zudem sehr viel schwieriger als parallel zur Planung, da keine Checkliste für diese Korrekturen existiert, sondern die Planer bisher nach Intuition eingreifen. Wird nun die neue IT-Anwendung eingeführt, so rechnet diese am Anfang mit inkonsistenten Daten. Eine lange Reihe von Analyse- und Debugzyklen ist nötig, diese Inkonsistenzen zu finden und ggf. eine Checkliste für die Konsistenzprüfung zu erstellen.

Eine andere Gefahr ist, die Endbenutzer unzureichend an der Planungs- und Analysephase für ein neues System zu beteiligen. Werden während der Analysephase die zukünftigen Benutzer einer IT-Anwendung kaum, zu spät oder gar nicht befragt, so wird das System auf der Grundlage unzureichender Informationen entwickelt.

Die Designer und Architekten von Geschäftsanwendungen beeinflussen mit ihrem Design weite Teile der Arbeitsabläufe. Die Beispiele oben veranschaulichen, daß sich das volle Ausmaß dieser Einflüsse meist erst nach der Einführung eines Systems herausstellt. Nicht selten dauert die Einführung einer Anwendung fast genauso lange

wie deren Entwicklung. Die Einführung der Anwendung umfaßt den Zeitraum von der ersten Auslieferung an den Kunden bis zum Produktionseinsatz beim Kunden. In dieser Zeit können sowohl Änderungen an der Anwendung aber mehr noch an den Arbeitsprozessen notwendig werden.

Bei großen Umstellungen, wie Datawarehouse-Projekten, SAP-Einführung etc., wird im Vorfeld eine Geschäftsprozessanalyse durchgeführt. Diese wird aber meist von Unternehmensberatern durchgeführt – also selten von den Leuten, die später die Anwendung entwickeln. Häufig sind sogar verschiedene Firmen am Werk. Dadurch wird sowohl der Kommunikations- als auch der Wissensfluß unterbrochen. Der Geschäftsprozeß liegt in der Verantwortung der Berater und die IT-Anwendung in der Verantwortung der IT-Fachleute. Das einzige Verbindungsglied ist dann bestenfalls eine Grobspezifikation für die IT-Anwendung. Hat die Unternehmensberatungsfirma, die den neuen Geschäftsprozeß erarbeitet hat, das Projekt vor der Entwicklung der IT-Anwendung verlassen, so können die Entwickler der IT-Anwendung auch keine Rückfragen mehr stellen. Die Grobspezifikation wird dann zu einer unveränderlichen Konstante, auf der sogar der Vertrag mit dem Kunden basiert. Während der Entwicklungsphase fallen aber sowohl dem Kunden als auch den Entwicklern genug Dinge ein, die zu ändern sind. Meist werden diese Änderungen eingebaut, ohne zu prüfen, ob die beteiligten Geschäftsprozesse noch konsistent sind. Geprüft wird dabei in der Regel, ob benötigte Daten erstellbar sind und welche Zusatzkosten entstehen. Selten wird aber geprüft, welche Auswirkungen die Änderung auf die Arbeitsabfolge des Sachbearbeiters/Planners, die Effektivität des Arbeitsablaufes, oder die Kommunikationspfade hat.

Ist die Spezifikation in einer Analysephase von den Entwicklern der IT-Anwendung erstellt worden, so ist zwar ein gewisses Verständnis der Geschäftsprozesse bei den Entwicklern vorhanden. Es wird aber selten in der Entwicklungsphase darauf zurückgegriffen. Das heißt, daß auch hier bei Änderungen der Spezifikation kaum die Konsistenz mit den Arbeitsprozessen geprüft wird.

## **Kommunikations- und Ausbildungsdefizite**

Software wird heute fast immer nach einem iterativen Entwicklungsprozeß durchgeführt. In jedem Fall werden die diversen Designebenen, Coding und Test iteriert. Die Analysephase zählt meist zum prä-iterativen Teil des Prozesses, zumal sie häufig ja auch zum großen Teil von einer anderen Firma durchgeführt wird. Allerdings kann, im Fall des Rapid Prototyping, die Analysephase die Erstellung eines Prototypen beinhalten. Dieser wird dann, nach der Verifikation durch den Kunden, Teil der Spezifikation. In der Softwareentwicklung hat sich der iterative Entwicklungsprozeß eingebürgert, weil die Softwareingenieure gelernt haben, daß man im Laufe der Entwicklung dazu lernt und nicht alles in der Analysephase vorhersehen kann. Das gilt genauso für den Kunden. Dieser bekommt durch Rückfragen der Entwickler auch neue Impulse und erkennt Lücken bei der Eingliederung der IT-Anwendung in den Arbeitsprozeß, die durch nachträgliche funktionale Erweiterungen der Spezifikation entstanden sind oder von Anfang an übersehen wurden. Selten bekommen aber die Softwareentwickler während dieser Phase einen Ansprechpartner in den Anwendungsabteilungen oder bei den Unternehmensberatern. Sie müssen häufig alle Fragen über die IT-Abteilung des Kunden klären. Dadurch wird es dem Entwickler fast unmöglich, eine Vorstellung von der Arbeitsumgebung, in die die Anwendung später eingebettet wird, zu erhalten. Hier sind also auch die Kunden gefragt, den Fachabteilungen mehr Zeit für Rückfragen während der Entwicklungsphase einzuräumen.

Da die Entwickler meist Informatiker ohne großes Fachwissen und Erfahrung im Fachgebiet des Kunden sind, ist es ihnen nur mit großem Zeitaufwand möglich, die Arbeitsumgebung des Kunden voll zu verstehen. Die Fachabteilung setzt dann leicht in den Interviews mit den Entwicklern Hintergrundwissen voraus, das bei den Informatikern nicht da sein kann. Als Beispiel sei hier die recht umfangreiche Fachsprache der Stahlplanung genannt. Ein Entwickler in einem Projekt in der Stahlindustrie braucht einige Zeit, um die nötige Fachsprache für die Gespräche mit dem Kunden zu lernen. Die Projektpläne sind aber aufgrund des Kos-

tendrucks sehr eng. Daher ist das Interesse der Entwickler, viel Zeit in das Verstehen der Arbeitsumgebung des Kunden zu investieren, relativ gering. Einen tieferen Einblick in die Probleme der Arbeitsumgebung des Kunden zu erhalten, erfordert dann schon Erfahrung aus vielen Projekten mit Kunden der gleichen Branche. Das ist ein Grund, warum zum einen in der IT-Branche immer mehr Informatiker mit Zusatzkenntnissen in anderen Bereichen (z.B. Betriebswirtschaft) gesucht werden und zum anderen man hier und da auch versucht, Leute mehrfach in Projekten aus einer Branche einzusetzen, damit sie Branchenwissen aufbauen können. Das gelingt aber natürlich nicht immer. Denn nicht immer hat ein Softwarehersteller genügend Aufträge aus einer Branche. Durch diesen Mangel an Branchenwissen und Zeit ist den Entwicklern meist kaum bewußt, wie weit sie die Arbeitsabläufe anderer wirklich beeinflussen.

In den Programmierübungen an Universitäten sind die Informatikstudenten gewöhnt, sehr genaue Spezifikationen zu bekommen. Diese wirken fast als ›Konstanten‹. Gegen diese ›Konstanten‹ muß korrekt programmiert werden. So entsteht leicht die Gewohnheit, daß eine Spezifikation nicht zu hinterfragen ist. Die Realität sieht aber anders aus. Selbst wenn eine Grobspezifikation vorliegt, kostet es häufig lange Interviews beim Kunden, eine genaue Spezifikation zu erstellen. Ein paar Studentinnen in meiner Software-Engineering-Vorlesung (während der Informatica Feminale) waren sehr verwirrt, als sie in einer Übung keine detaillierte Spezifikation bekamen, sondern erst herausfinden mußten, was genau zu lösen war. Viele versuchten das eher zu raten, anstatt den ›Kunden‹ zu fragen.

Viele Informatiker neigen dazu, zu schnell eigene Lösungen vorzuschlagen, anstatt erst mal dem Kunden genau zuzuhören. Das geschieht teilweise auch unbewußt aus Zeitmangel. Nehmen wir an, dem Entwickler fällt ein Sonderfall auf, für den in der Spezifikation nicht beschrieben wird, wie das System reagieren soll. Solch ein Sonderfall kann z.B. das Fehlen von Daten für einen Kunden in der Datenbank o. ä. sein. Häufig ist es weniger Aufwand, eine relativ bequem zu implementierende Lösung vorzuschlagen, als beim Kunden genau nachzufragen, was benötigt wird. Der Entwickler implementiert ein-

fach einen Abbruch mit Fehlermeldung. Der Endbenutzer hätte aber lieber die Möglichkeit, die fehlenden Daten per Hand einzugeben. Oder dem Entwickler fällt auf, daß er eine Regel oder Strategie braucht, fehlende Daten in der Datenbank zu suchen. Statt den Kunden zu fragen, ob die Suchstrategien Auswirkungen auf das Ergebnis haben, wird einfach eine Standardstrategie implementiert. Der Endbenutzer würde aber diese Suchstrategie für verschiedene Fälle gern selbst definieren. Die Gefahr, zu wenig Fragen zu stellen, besteht auch dann, wenn die Entwickler schon ähnliche Projekte gemacht haben. Voraussetzungen, die beim Kunden A gegeben waren, werden dann beim Kunden B ohne Rückfrage ebenfalls vorausgesetzt. Das kann ein schwerwiegender Irrtum sein.

So manches Projekt scheitert dann auch bei der Einführung der Anwendung, weil man dann erkennt, daß die Funktionalität, die in der Spezifikation beschrieben wurde, das Kundenproblem nicht löst oder nicht in die bestehenden Arbeitsprozesse paßt. Das Erstellen der Spezifikation und mehr noch die Prüfung der Arbeitsabläufe müssen also mit in die Iterationen des Entwicklungsprozesses eingegliedert werden.

## **Der Informatiker der Zukunft**

Der Informatiker der Zukunft muß daher vor allem auch Zuhören und gute Interviewtechniken lernen, um in der Kommunikation mit Menschen aus ihm fremden Arbeitswelten möglichst wenig zu übersehen. Er braucht ein stärkeres Bewußtsein dafür, daß er wirklich für Menschen arbeitet und zu einem Teil deren Arbeitsumgebungen gestaltet. Indirekt wird damit auch der Softwareentwickler zum Berater und sollte sich auch wirklich in dieser Rolle sehen. Zudem arbeiten heute schon viele Informatiker als Berater. Dem sollte auch bei der Strukturierung der Studiengänge für Informatiker Rechnung getragen werden.

Consulting- und Softwareentwicklungsmethoden müssen miteinander verknüpft werden. Der iterative Softwareentwicklungsprozeß darf nicht erst bei der fertigen Grobspezifikation beginnen. In Projekten, denen eine Studie einer Unternehmensberatung vorangeht oder die von Unternehmensberatern begleitet werden, müssen Berater und Informatiker mehr zu einem Team zusammenwachsen. Die Entwickler sollten in Teilen der Geschäftsprozessanalyse mitbeteiligt sein. Dadurch erhalten sie einen besseren Einblick in die Umgebung, in die später die Anwendung eingegliedert wird. Die Berater erhalten auf der anderen Seite frühzeitig Informatio-

nen über technische Machbarkeit und Lösungsmöglichkeiten. Ebenso muß eine Kontaktperson zu den Anwendern während der Entwicklung und Anwendungseinführung zur Verfügung stehen, um bei Änderungen die Auswirkungen auf die Prozesse zu prüfen. Dieses kann je nach Projekt ein Berater oder Entwickler sein. Nur so ist der Bruch zwischen der Voranalyse des Geschäftsprozesses und der Analysephase des Softwareprojektes zu überwinden.

Das erfordert natürlich auch zusätzliche Schwerpunkte in der Ausbildung der Informatiker. Nicht jeder Informatiker muß ein voll ausgebildeter Unternehmensberater sein. Er sollte aber in Kommunikations- und Repräsentationstechniken gezielter geschult werden, als dies heute der Fall ist. Der Informatiker der Zukunft muß ja nicht nur die Probleme des Kunden verstehen und Lösungen implementieren. Oft genug muß er die Lösungsansätze auch dem Kunden oder Endbenutzer erläutern und ihm gegenüber begründen. Dies muß dann in der Sprache des Kunden und nicht des Informatikers geschehen.

Vielleicht wird das neue Millenium nicht nur eine Fortführung des Informationszeitalters, in dem wir uns nach Neil Postman zu Tode informieren und amüsieren, sondern eines der Kommunikation und des Verstehens – nicht nur zwischen Informatikern, Beratern und Anwendern.