

Nicht falsch, sondern das Falsche geschätzt.

Wann wird man je versteh'n, wann wird man je versteh'n?

Gehört es zum Wesen der Software, daß ca. 50% aller Softwareentwicklungen nicht zur geplanten Zeit, nicht zu den geplanten Kosten und nicht im geplanten Umfang ausgeliefert werden und ca. 30% aller Projekte ohne Ergebnis abgeschlossen werden? ¹

In der Tat ist es so, daß die Mehrzahl aller Softwareanwender über die vorstehend beschriebenen Symptome klagt, unabhängig davon, ob sie Standard- oder individuell entwickelte Software nutzen. Erstaunlich ist, daß zwar dieser Mißstand bekannt ist, aber dennoch nicht zu Anstrengungen im IT-Bereich führt, diesen zu analysieren und zu reduzieren.

Ein anderes Phänomen, von dem die Standish Group berichtet, ist, daß ca. 50% der IT-Manager überzeugt sind, daß heute mehr Projekte scheitern als vor 5 und 10 Jahren. Hier stellt sich die Frage, wissen die IT-Manger nicht was sie tun, oder warum lernen sie nicht über die Jahre, erkannte Schwächen zu verbessern?

Daß die IT-Manager wissen, was die Gründe für den nicht zufrieden stellenden Projektverlauf sind, stellte die Standish Group in ihren Erhebungen fest. Als die wichtigsten Ursachen wurde von den IT-Manager fast einstimmig genannt: mangelnde Einschaltung der Anwender, unvollständige und/oder ungenaue Anforderungen und zu wenig Beachtung durch das Management des Unternehmens. Es mangelt nicht primär an den technischen und methodischen Kenntnissen der Mitarbeiter im IT-Bereich und einer unzureichenden Ausrüstung der Arbeitsplätze.

Erstaunlich ist, daß die oben aufgeführten Ursachen für die Mängel im IT-Bereich seit langem bekannt, aber ernsthafte Bestrebungen zur Abhilfe nicht erkennbar sind. Warum ist dies so und warum bleibt es so?

Mangelnde Einschaltung der Anwender.
Am Anfang gab es keine Programmierer.

Der Grund " mangelnde Einschaltung der Anwender" ist historisch bedingt. Als Anfang der 60-er Jahre die ersten elektronischen Anlagen verkauft wurden, gab es keine Programmierer. Die Anwender mußten Mitarbeiter aus dem eigenen Unternehmen zur Programmierausbildung zu den Lieferanten der Anlagen schicken. (Die Begriffe Hard- und Software waren noch nicht geprägt.) Sinnvollerweise wurden damals Mitarbeiter aus den Bereichen des Unternehmens zur Schulung geschickt, deren Arbeiten auf die Anlagen umgestellt wurden. Da es bei ersten den Anwendungen um reine Automatisierung bestehender manueller Abläufe ging, verfügten die ersten Programmierer über die betriebswirtschaftlichen Kenntnisse der Anwendung, die für die Umstellung notwendig waren. Die Programmierer waren in der Realität groß geworden, die sie in der Software

¹ Sh. Standish Group International, 1996, www.standishgroup.com
dasfalsche 05.01.06

abbilden sollten. Die Planung der Druckausgaben wurden mit der gleichen Akribie ausgeführt, mit der damals die Organisatoren zu arbeiten pflegten. Generell lag die Verantwortung für die Umstellungen bei den Organisationsabteilungen der Unternehmen. Aus heutiger Sicht kann man feststellen, daß damals die Anwender vertreten durch die Organisationsabteilungen die Verantwortung für Umstellung ausübten.

In den letzten 40 Jahren hat sich seitdem das Bild um 180 ° gedreht. In den 70er Jahren fand man in den Unternehmen noch Org-DV-Abteilungen. D.h. man sah noch einen Unterschied zwischen der Organisations- und DV-Kompetenz.

Mit dem Einzug der EDV in die Lehrpläne der Universitäten und technischen Hochschulen verselbständigte sich die Informatik. Software wurde immer mehr als eine Reihenfolge von Befehlen an den Computer gesehen und weniger als die Abbildung eines realen Vorganges durch eine Abbildung auf dem Computer. Die IT entwickelte sich zu einer selbständigen Disziplin, obgleich sie stets interdisziplinär genutzt wird, um Aufgaben und Probleme anderer Bereiche zu lösen. Sicherlich sollte der IT-Bereich (Hardware und Betriebssysteme) selbst Gegenstand von Forschung und Lehre sein. Die interdisziplinäre Anwendung jedoch ist in den Anwendungsbereichen selber zu vermitteln. (Z.B. haben das Bauen, Fahren und Nutzen von Kraftfahrzeugen unterschiedliche Aspekte zu berücksichtigen.)

In IT-Kreisen hört man immer wieder das Argument, daß der Anwender selber nicht weiß, was er macht und was er benötigt. - Daß die IT-Spezialisten es aber auch nicht besser wissen, bestätigen nicht nur die Untersuchungen der Standish Group International, sondern die tägliche Erfahrung. - Abgesehen von der Arroganz, die in dem Ausspruch liegt, beachtet er nicht die Tatsache, daß Anwender und Informatiker zwei völlig verschiedene Sprachen sprechen und unterschiedliche Erfahrungshintergründe haben. Jeder kann an sich selbst beobachten, daß Beschreibungen von Aktivitäten um so gröber ausfallen je selbstverständlicher sie einem sind. (Beschreiben Sie einmal den Vorgang des Parkplatzsuchens und des Einparkens so, daß ein Auto-Roboter danach programmiert werden kann.)

Die stärkere Einschaltung der Anwender in den Prozeß der Softwareentwicklung ist zwar notwendig aber alleine nicht ausreichend, um unvollständige und/oder ungenaue Anforderungen zu vermeiden. Wir benötigen in den Unternehmen und Verwaltungen Betriebsorganisatoren, die die geforderten und notwendigen Arbeitsabläufe/Prozesse erkennen und so beschreiben können, daß sie von Softwareentwicklern eindeutig umgesetzt werden können.

Unvollständige und/oder ungenaue Anforderungen.

- Nimm Dir Zeit, um Zeit zu sparen-

Unumstritten ist die Aussage, daß mindestens 60% aller beim Testen gefundene Fehler ihre Ursache in der Anwender Anforderung haben.

Die Softwarebranche befindet sich heute noch in einem Stadium der Fertigung, das die Autoindustrie mit großen Anstrengungen vor mehr als 10 Jahren verlassen hat: Die IT-Spezialisten bringen die Qualität beim Abnahmetest in ihre Produkte, statt eine Qualitätssicherung in ihre Arbeitsprozesse einzuplanen. Am Anfang jeder Qualitätssicherung steht eine genaue, vollständige und verbindliche Produkthanforderung. Der GIGO, der vor Jahren in der Datenverarbeitung oft zitiert wurde, sollte heute vor allem bei der aufwendigen und folgewirksamen Softwareentwicklung beachtet werden: Garbage In,

Garbage Out. Ein Softwareprodukt wird selten vollständiger und besser werden als die Anwenderanforderung. Planung und Fertigung von Softwareprodukten muß im Sinne des oben Beschriebenen wieder eigenverantwortlich arbeiten und sich über eine definierte Schnittstelle abstimmen.

Die wichtigste Voraussetzung für jede Verbesserung in der Softwareentwicklung ist jedoch eine fundierte und verantwortliche Terminsetzung. Termine dürfen sich nicht mehr an der gewünschten Verfügbarkeit des Produktes ausrichten, sondern an dem Aufwand, der für eine vorliegende Produkthanforderung notwendig erscheint.

Für die Praxis bedeutet dies:

Planung und Realisierung von Softwareprodukten sind in getrennten Projekten aufzusetzen. Die Verantwortung für die Projekte sind im Sinne von Architekt und Bauunternehmen zu trennen. Der Termin für die Planung wird durch den Planungsverantwortlichen festgelegt. Der Termin für die Realisierung wird auf der Grundlage der vorliegenden Planung und der zur Verfügung stehenden Ressourcen ermittelt.

Die Vergangenheit zeigt, daß die inhaltlichen Vorgaben konstanter sind als deren technische Realisierung. Der Inhalt eines Auftrages bleibt der gleiche unabhängig davon, ob er per Post, Telefon, Fax oder Internet erteilt wird. Layout und Medium dagegen werden sich an der aktuellen Technik orientieren.

Neue Organisationsstrukturen und Zielsetzungen werden nicht aus sich selbst heraus wachsen. Sie vorzugeben und durchzusetzen ist Aufgabe des Management.

Zu wenig Beachtung durch das Management des Unternehmens

- Mut, kritische Frage zu stellen -

In den Chefetagen der Unternehmen findet man leider allzu häufig ein Bildungsrelikt, das sich wie folgt zeigt: man schämt sich nicht zu sagen, daß man sich in Mathematik und den Naturwissenschaften in der Schule schwer getan hat. Was man nicht zugibt, ist, daß man die Werke von Händel und Purcell nicht unterscheiden kann. Die Folge ist, daß die Manager sich nicht die Kompetenz zutrauen, Entscheidungen im IT-Bereich gegen den Willen der Betroffenen zu treffen. Dadurch entwickelten sich die IT und darin vor allem der Softwarebereich zu allzu selbständigen und selbstbestimmten Bereichen.

Das Management muß erkennen, daß auch für den IT-Bereich Organisationskompetenz und damit verbundene Zielsetzungen Managementaufgaben sind. Für durch das Management gesetzte Ziele sind erst in einem zweiten Schritt die technischen Lösungen zu finden. Organisatorische Vorgaben bestimmen die technische Realisierung und nicht umgekehrt.

Das Management sollte einige einfache Fragen an die Verantwortlichen im Softwarebereich stellen und Antworten verlangen:

- Warum ist trotz beachtlichen Investitionen in Tools und Methoden über die Jahre keine Produktivitätsverbesserung nachzuweisen; oder doch?
- Warum ist trotz beachtlichen Investitionen in Tools und Methoden über die Jahre keine Qualitätsverbesserung nachzuweisen; oder doch?

- Können sie nachweisen, daß die Softwarekosten zwangsweise mit dem Geschäftsvolumen wachsen müssen? (Die Erstellungskosten eines Buches wachsen nicht mit der Zahl der Leser!)
- Wie haben sie die Kapazitäten verwendet, die nach der Lösung des 2000-Problems doch in größerer Zahl werden mußten?
- Wie hat sich die Beschaffung von Standardsoftware und die Einschaltung externer Mitarbeiter auf das Software-Budget ausgewirkt? Warum und wieviel ist es gesunken/gestiegen?
- Warum haben sie nie gegen unsere Terminvorgaben massiv berechtigten Einspruch eingelegt?
- Warum beginnen Sie mit der technischen Realisierung der Software, bevor eine verbindliche Anwenderanforderung vorliegt?

(Das Management sollte die Fragen stellen, auch wenn es befürchten muß, daß einige Antworten es selber treffen könnte!)

Mit Sicherheit werden derartige Fragen einen gedanklichen Erdrutsch auslösen, der nicht nur vorhandene Probleme, sondern auch Ziele für deren Lösung freilegt. Welche Wege zur Erreichung der Ziele man wählt, um mit welchen Mittel man sie angeht, muß das Management vorgeben und überwachen.

© Robert Hürten, Blankenheim