

Auszug aus **Eine Million oder ein Jahr**
http://www.ephorie.de/eine_million_oder_ein_jahr.htm

Dieser Auszug wird Ihnen präsentiert von:



ephorie.de – Das Management-Portal
<http://www.ephorie.de>

Mit freundlicher Unterstützung von und alle Rechte bei:
SmartBooks Publishing

Computer sind phantastisch: In wenigen Minuten können sie einen so großen Fehler produzieren, wie Menschen ihn nur in vielen Monaten machen könnten.

(Merle Meacham)

Eine Million oder ein Jahr

Warum IT-Projekte so lange dauern, so kompliziert und so teuer sind

Ein oft gebrauchter *Running Gag* lautet: Wenn du die IT um die Änderung eines Programms oder um die Installation eines neuen Systems bittest, so erhältst du entweder die Antwort «Es kostet eine Million», oder «Es dauert ein Jahr». Manchmal bekommt man auch beide Antworten gleichzeitig.

Nun will ja niemand den Mitarbeitern in der IT fehlenden Willen unterstellen, und man kann sicher davon ausgehen, dass sie nach besten Kräften helfen wollen. Dennoch sind diese Antworten meistens korrekt.

Für die Antwort «Es dauert ein Jahr» könnte die einleuchtende Erklärung aufgeführt werden, dass zu viele Projekte und Aktivitäten gleichzeitig betreut werden: Es ist schlicht keine Zeit vorhanden. Aber das wäre zu kurz gegriffen.

Bis zum Ende der 80er-Jahre war IT aus technischer Sicht noch recht überschaubar. Eine große Maschine, die alles konnte und wusste, kein mit den heutigen Möglichkeiten vergleichbares Netzwerk, keine PCs und nur eine minimale und rein theoretische Gefahr durch vereinzelt Hacker. In den 90ern wurde es dann unübersichtlich und kompliziert und der Begriff *IT-Landschaften* wurde geprägt, um die Ansammlung von vielen unterschiedlichen Bausteinen zu bezeichnen. Das Wort *Landschaft* hat etwas Beruhigendes, es erinnert an grüne Wiesen, prachtvolle Bäume, grasende Kühe und zwitschernde Vögel. Jeder mag es, jeder findet es gut. Aber es ist nicht richtig, von Landschaften zu sprechen. IT entspricht eher einem Ökosystem, einem Biotop. Es ist fragil und kompliziert, es gibt unbekannte Zusammenhänge und vor allem: Keiner vermag auf Anhieb zu sagen was passiert, wenn man eine Änderung darin vornimmt. Ich ziehe es daher vor, von IT-Biotopen zu reden anstatt von IT-Landschaften.

Die Anschaffung eines neuen Computersystems oder die Installation von neuer Software kann man in etwa mit dem vergleichen was passiert, wenn man in einem Biotop eine neue Tier- oder Pflanzenart aussetzt. Die Risiken sind nicht so ohne weiteres einzuschätzen, keiner weiß ob es funktionieren wird, ob dadurch ein Mehrwert geschaffen oder ob es das Gleichgewicht zerstören wird.

Um also die Risiken zu kontrollieren, muss eine Impaktanalyse durchgeführt werden. Wird dieses System oder Software sich benehmen wie ein Krokodil und alles andere fressen, oder ist es eine Ente, die vom Krokodil gefressen wird? Wie wird es seine Nahrung bekommen, sprich, wie wird es mit anderen Systemen kommunizieren? Es ist nicht schwer, sich vorzustellen, dass dies sehr aufwendig

und ausgesprochen teuer ist und dass es keine Garantien für das Ergebnis der Analyse gibt.

Nun könnte man ja auf die Idee kommen, den Produzenten (Provider) eines Systems oder einer Software zu bitten, eine Verträglichkeitsgarantie abzugeben. Der wird sich jedoch weigern dies zu tun, und zwar aus sehr gutem Grund: Es gibt nämlich keine zwei identischen IT-Biotope auf der Welt!

Jedes IT-Biotop besteht aus diversen Hardwarekomponenten von verschiedenen Providern, die wiederum eigene Kommunikationskomponenten und verschiedene Steuerungssoftware haben. Allein dieser Teil füllt bereits Bibliotheken an Fachliteratur. Dann gibt es Hunderte, in größeren Betrieben Tausende von Einzelsoftwarekomponenten. Wenn ich es mir recht überlege, sollte ich eigentlich von einem Dschungel reden, anstatt von einem Biotop. Für jede einzelne Komponente gibt es verschiedene Versionen, die entweder von der IT-Abteilung auf den letzten Stand gebracht worden sind oder noch hinterher hinken (Windows XP, Windows 2000, Windows NT ...). Damit das alles miteinander *reden* kann, gibt es Brücken zwischen den einzelnen Komponenten, so genannte Interfaces. Es ist genauso unwahrscheinlich, zwei Menschen mit den gleichen Genen anzutreffen wie zwei IT-Biotope mit absolut identischen Komponenten. Vor diesem Hintergrund soll ein Provider eine Verträglichkeitsgarantie abgeben? Das wäre geschäftlicher Selbstmord.

Die nächste Schwierigkeit: Passt das einzuführende System überhaupt zu unseren Bedürfnissen, das heißt erfüllt das System die funktionalen Anforderungen des Betriebes?

Wollte der Geschäftsbereich etwas Schilf an den Teich? Oder war es in Wirklichkeit eine Seerose, die gewünscht wurde? Mitunter stellt sich dies erst im Laufe eines Projektes heraus, wenn bereits große Summen ausgegeben wurden. Es ist enorm schwer für die Geschäftsbereiche präzise zu entscheiden, ob ein einzuführendes System den Anforderungen entspricht oder um etwaige Abweichungen festzustellen. IT-Abteilungen aber müssen sich auf die Entscheidung des betroffenen Geschäftsbereichs verlassen. Letztendlich werden ja auch alle Kosten vom Geschäft getragen. Präsentationen der Provider sind immer perfekt, Feedback von befreundeten Firmen – sofern dies zu bekommen ist – ist mit Vorsicht zu genießen, da deren Aussage nur deren Biotop betrifft. («Es funktioniert doch bei denen, dann werdet ihr das wohl doch auch hibekommen!»). Die Provider erleichtern die Aufgabe auch nicht, wenn sie behaupten, dass das Schilf in 10 Minuten eingepflanzt werden kann. Wir werden in diesem Kapitel sehen, dass die Implementierung (Installation) sehr kompliziert und keine 10-Minuten Aufgabe ist.

Bei der Anschaffung eines Systems wird in der Regel ein Unterprojekt *Implementierung* gestartet. Es beginnt mit einer Trockenanalyse, ob das System ein Krokodil oder eine Ente ist, um dann, wenn das Ergebnis vorliegt, das System in ein Testbiotop zu werfen, um verschiedenste Tests durchzuführen. Dabei kann es vorkommen, dass die vermeintliche Ente sich dann doch noch als Krokodil entpuppt. Solche Implementierungen gehen massiv ins Geld, da hierfür hoch bezahlte Spezialisten eingesetzt werden. Tagessätze von 1.000 - 1.500 Euro sind

noch nicht einmal exzessiv. Der Zeitrahmen für eine Implementierung wird nicht mit Wochen, sondern in der Regel mit Monaten angesetzt.

Ein wichtiger Punkt, der während der Implementierung enorm viel Zeit in Anspruch nehmen kann, ist die Definition der Installationsparameter. Jede Software, selbst das einfachste Produkt, hat welche. Mit ihnen wird festgelegt, wie die Software benutzt werden soll und wie sie an die individuellen Bedürfnisse der Firma oder des Benutzers angepasst wird. Ungewünschte oder nicht benötigte Funktionen können dadurch abgeschaltet werden. Bei Produkten für die breite Masse sind die Parameter der Software fast immer bereits vorgegeben und niemand muss sich darüber speziell Gedanken machen. Schauen Sie mal bei Word unter *Optionen* nach und Sie werden feststellen, dass es recht viele sind. Bei Fachsoftware, wie Buchungsprogrammen, sind diese Parameter in aller Regel undefiniert und es gibt Hunderte davon, die allesamt definiert werden müssen. Dabei muss der Sinn eines jeden einzelnen Parameters zuerst verstanden sein. Dann muss geprüft werden, welchen Einfluss er auf die Abläufe in der Organisation der Firma hat. Dabei kann man sehr leicht schwerwiegende Fehler begehen, die erst im Produktionsbetrieb auffallen und nur sehr schwer zu korrigieren sind. Besonders aufpassen muss man auch darauf, dass nur sehr genau definierte Personen Änderungsrechte an diesen Parametern erhalten.

Um die Sache nicht zu einfach zu machen, ändert sich oft während der Impaktanalyse oder interessanter noch, während der Implementierung selbst, ein entscheidender Baustein. Es kann sich zum Beispiel herausstellen, dass ein anderes Projekt eine Umstellung auf die nächste Windows-Version plant und hierdurch irgendeine kleine Unterkomponente des uns interessierenden Projektes inkompatibel zum Rest wird. Der Provider stellt dann vielleicht eine Lösung für «in 6 Monaten» in Aussicht. Bei diesem Thema kann man wunderbar monatelang Karussell fahren, ohne jemals wirklich voranzukommen.

Der Appetit kommt bekanntlich beim Essen. So ist es die Regel, dass die Geschäftsbereiche im laufenden Projekt die Prioritäten ändern («Wenn wir das auch noch hineinpacken, dann können wir unseren Profit um x steigern. Time ist schließlich Money») und zusätzliche Bedürfnisse («Wir haben da noch etwas vergessen») formulieren. Die beste Planung wird dann zur Makulatur. Der Druck auf den Provider, die zusätzlich benötigte Funktionalität in sehr kurzer Zeit zu entwickeln, wird extrem hoch und die Qualität wird dadurch automatisch reduziert. Zusätzlich kann es natürlich auch sein, dass völlig andere, neue Projekte eine höhere Priorität bekommen und das aktuelle Projekt auf *Halten* gesetzt werden muss.

Es ist gar nicht so einfach, wie man glaubt, die Zielsetzung eines Projektes einem Provider klar zu machen. Der Auftraggeber benutzt seine eigene Terminologie und ist möglicherweise schon mehrere Wochen intern mit dem Thema beschäftigt, bevor er versucht, einen Provider für sein Projekt zu finden. Manche Projekte sind so groß, dass mehrere Provider ausgesucht werden. Man kann sehr oft beobachten, dass der gewählte Provider am Ende nicht genau weiß, was alles in dem Projekt steckt, welche Wichtigkeit es hat und vor allem, welche Teile des Projektes für ihn unsichtbar sind, weil ein anderer Provider dafür ausgewählt wurde.

Jeder Provider will seinen Auftrag unbedingt bekommen und verspricht oft das Blaue vom Himmel. Später kommt er dann in zeitliche Bedrängnis und vielleicht reicht das Budget nicht aus.

Mit den von der Geschäftsseite definierten Prioritäten ist das auch so eine Sache. Jeder Bereich sieht nur seine eigenen Wünsche und bei der IT laufen diese alle auf. Jeder Bereich im Betrieb definiert seine Anforderungen als Top-Priorität. Es ist klar, dass nicht alles und jedes sofort gemacht werden kann. Aus diesem Grund werden Entscheidungsgremien installiert, die die Aufgabe haben, die globalen Prioritäten festzulegen. Kleinere Anfragen, die nicht als Projekt klassifiziert werden, sind selbstverständlich vom Anfragenden auch als Priorität 1 deklariert worden. Jeder Geschäftsbereich weiß, dass es keinen Sinn macht, Anforderungen als Priorität 2 zu deklarieren, da dann klar ist, dass in den nächsten zehn Jahren bei diesem Thema nichts passieren wird. Wenn aber alles als Priorität 1 deklariert wird, dann verliert das Ganze seinen Sinn. Also geht die IT hin und klassiert klammheimlich intern die kleineren Anforderungen um in Priorität 1a, 1b, 1c und so weiter ...

Sehr unbeliebt sind auch plötzliche Änderungen bei legalen und regulatorischen Anforderungen. Dieser Aspekt der Arbeit hat in den letzten Jahren stark zugenommen und wird weiter an Bedeutung gewinnen und damit die Realisierung von Projekten weiter erschweren (siehe Kapitel *Worst Case*).

Projekte können noch durch weitere Gründe scheitern oder verzögert werden. In der Regel werden Projektleiter eingesetzt, jeweils einen im eigenen Betrieb und einen beim Provider. Man beobachtet in den letzten Jahren verstärkt einen Trend dahin, sehr junge Projektleiter mit nur wenig Erfahrung einzusetzen. Diese haben zwar in der Regel eine gute Ausbildung, besitzen aber keine ausreichende Erfahrung. So ist es unglaublich schwer, diesen jungen Leuten zu vermitteln, dass es durchaus in Ordnung sein kann, wenn ein Projektpunkt nicht rechtzeitig erledigt worden ist. Projektpläne haben wichtige und unwichtige Punkte; bei den unwichtigen Punkten dienen die vereinbarten Termine lediglich als Richtlinie. Die Unterscheidung zwischen wichtig und unwichtig haben diese Jungspunde noch nicht gelernt und wollen es auch nicht begreifen. Ich wäre sehr dafür, generell nur erfahrenen Mitarbeitern die Leitung eines Projektes zu übertragen, doch junge Mitarbeiter kosten den Betrieb weniger Gehalt, dafür fehlt ihnen dann die notwendige Erfahrung. Ein teurer Irrweg.

Die Provider befinden sich in einem extrem konkurrierenden Markt und die neuen Produkte müssen so schnell wie möglich ausgeliefert werden. Dass dies der Qualität nicht gerade förderlich ist, liegt auf der Hand. (Mehr Details zu der Problematik der Softwarequalität im Kapitel *Fällt in China ein Fahrrad um*). Um sich gegen das Problem der mangelnden Softwarequalität abzusichern, verpflichtet sich der Käufer in einem betrieblichen Umfeld dazu, per Wartungsvertrag jedes Jahr 20 Prozent des Kaufpreises (in seltenen Fällen kann es weniger als 20 sein) an den Provider zu zahlen, damit dieser die Fehler verbessert, die er selbst erschaffen hat. Ohne Frage ein wunderbares Geschäftsmodell. Zugegeben, ein Teil dieses Geldes wird benutzt, um die Software laufend an neue externe Gegebenheiten, wie zum Beispiel eine neue Windows-Version, anzupassen. Der Löwenanteil jedoch wird für das Verbessern der Fehler aufgewendet. Manche Softwareprovider gehen hin und erhöhen den zu

zahlenden Betrag jedes Jahr um einige Prozent um einen vermeintlichen Inflationsverlust auszugleichen. Nach nur fünf Jahren haben sich die 20 Prozent bereits auf 25 erhöht. Es gibt auch Beispiele, bei denen Provider die Nutzung der ursprünglichen Lizenz zeitlich begrenzen. Nach Ablauf der Frist (zum Beispiel 10 Jahren), wird der volle ursprüngliche Betrag erneut fällig. Die Wartungsgebühren lagen in den 80er-Jahren bei 10 Prozent und haben sich mittlerweile verdoppelt – sicherlich auch ein Hinweis darauf, dass Software zwar mehr kann, aber auch störanfälliger geworden ist. Je mehr Fehler drin sind, umso mehr Wartungsgebühren werden verlangt. (Allerdings sollten die Provider hier nicht übertreiben, da sonst das Produkt sich gar nicht mehr verkauft.) Und jede neue Version eines Programms bringt neue Fehler.

Eine noch größere Problematik besteht in der Realisierung von so genannten *Interfaces*. Dies sind Brücken zwischen Systemen, die in den meisten Fällen von der IT, zumindest aber im direkten Auftrag der IT realisiert werden. Bildlich gesprochen besteht die Aufgabe darin, einen Text von Russisch auf Chinesisch zu übersetzen, wobei man weder die eine, noch die andere Sprache beherrscht. Da es wie gesehen keine zwei identischen IT-Biotope gibt, ist die Chance gering, ein derartiges Interface schlüsselfertig einzukaufen. Es muss in den meisten Fällen speziell programmiert werden, wobei wir wieder bei der obigen Problematik der Softwarequalität sind. Das Schreiben von Interfaces ist in den IT-Abteilungen außerordentlich unbeliebt, da es eine der aufwendigsten und riskantesten Aufgaben ist, der eine IT sich stellen muss. Seit jeher herrscht bei diesem Thema ein eklatantes Missverständnis zwischen den Geschäftsbereichen und der IT. Lapidare Aussagen wie: «Dann schreibt unsere IT ein Interface» hört man auch heute noch regelmäßig. Jedes Mal wenn ich dies höre, kommen wieder ein paar graue Haare dazu. Es wird unterstellt, dass dies eine einfache und risikolose Aufgabe ist. Ich denke, die IT-Experten haben es bis heute nicht verstanden, zu vermitteln, dass Interfaces kompliziert sind und einen lebenslangen Aufwand bedeuten. Möglicherweise herrscht hierbei der Hintergedanke vor, sich unentbehrlich zu machen – was besonders für externe Softwarefirmen gelten dürfte. Interfaces sind einer der Hauptgründe, warum die Anzahl der Mitarbeiter in den IT-Abteilungen so enorm gewachsen ist.

Ein kleines Beispiel zur Veranschaulichung: Softwaresysteme speichern ihre Daten in Datenbanken. Zumeist sind diese Datenbanken für die IT bei einer Neuanschaffung *Blackboxes* und auch nur unzureichend dokumentiert. Der Provider ist aus Gründen des Geschäftsgeheimnisses nicht geneigt, dies zu ändern. Ein Interface hat die Aufgabe, Daten von einer *Blackbox* zur nächsten zu liefern. System A (zum Beispiel Auftragsverwaltung) hat eine Kundennummer, die 10 Stellen lang ist. Diese Information muss in das System B (zum Beispiel Rechnungserstellung) transferiert werden, in dem die Kundennummer nur 8 Stellen hat. Das kann nicht gehen. Dieses Beispiel ist extrem einfach, verdeutlicht aber bereits die Schwierigkeiten, die bei Interfaces zu meistern sind. Fragen Sie Ihren CIO im Betrieb, er kann Ihnen sicherlich Beispiele liefern, bei denen man sich die Zähne ausbeißt.

Ein in Betrieb befindliches Interface hat spezielle Anforderungen an seine Überwachung. Eine so genannte *Reconciliation* (Abstimmung) muss implementiert werden. Deren Aufgabe ist es, sicherzustellen, dass System A und B auf dem gleichen Informationsstand sind. Fällt das Interface einmal aus oder

arbeitet es inkorrekt, dann muss die *Reconciliation* Alarm schlagen. Die Entwicklung einer solchen Funktion ist fast genauso komplex wie das Interface selbst.

Wenn die IT dann irgendwann mit ihrer Arbeit fertig ist, fangen die Tests in den Fachbereichen an. Dies ist ebenfalls eine sehr komplexe Thematik: Die Biotope sind so groß, dass von einem Anwender nicht erwartet werden kann, dass er alle Zusammenhänge kennt. Zugleich wird auch in den Fachbereichen die Spezialisierung auf einzelne, punktuelle Themen immer stärker. Dies passt nicht zusammen. Die Testergebnisse sind dann per natura nur bedingt aussagekräftig. Hier braucht es Leute mit übergreifendem IT-Wissen und Fachwissen aus dem Geschäftsbereich. Solche Kräfte zu finden ist außerordentlich schwierig. Das Testen von Programmänderungen gewinnt von Jahr zu Jahr an Bedeutung und stellt in sehr vielen Fällen die wichtigste Aufgabe bezüglich Zusammenarbeit mit der IT in einem Fachbereich dar. Es ist ein Element, das jeder Fachbereich in einem Betrieb bei seiner Ressourcenplanung berücksichtigen muss. Erschwerend kommt bei den durchzuführenden Tests hinzu, dass jede Software Fehler hat und deshalb eine präzise Projektplanung zu einem Ding der Unmöglichkeit werden kann. Diese Unplanbarkeit führt regelmäßig zu großer Frustration bei allen Projektbeteiligten und ist ein entscheidender Grund dafür, warum sich IT-Projekte verzögern.

Richtig schön kompliziert wird es, wenn in einem System gleichzeitig mehrere Projekte durchgeführt werden, die sich gegenseitig beeinflussen, aber von der Thematik eigentlich unabhängig sind. Zum Beispiel muss das Buchungsprogramm wegen der Euro-Einführung umgeschrieben werden. Da man schon ein kritisches Programm abändert, möchte der Geschäftsbereich auch gleichzeitig das Gebührenmodell ändern. Aus reiner IT-Sicht wäre es ratsam, zuerst die Euro-Einführung abzuschließen und dann das Gebührenmodell anzugehen. Aus geschäftlichen Gründen (Top-Priorität) mag dies anders aussehen und deshalb wird entschieden, beides gleichzeitig durchzuführen. Beide Projekte müssen zwingend gleichzeitig abgeschlossen werden, da man das Buchungsprogramm nicht für die Euro-Einführung in Produktion nehmen kann, wenn die Tests für das Gebührenmodell noch nicht erfolgreich abgeschlossen wurden. Da IT-Projekte zeitlich nie so verlaufen, wie man das plant, ist diese Situation ein sehr großes Risiko und sie ist der Auslöser für so manche Krisensitzung.

Selbst wenn alles so weit gut gelaufen ist, bleibt eine weitere Gefahr: Ihr Provider ändert seine Strategie (oder geht in Konkurs), und Sie haben Ihre IT-Strategie auf der des Providers aufgebaut. Klar, Verträge sichern dies ab. Aber es passiert nun mal, dass Lieferanten trotz gültiger Verträge entscheidende Dinge ändern und man steht plötzlich im Regen. Es hilft wenig, wenn Schadensersatz geleistet wird, wenn Sie danach während einer nicht absehbaren Zeit keine Lösung für die Anforderungen aus dem Geschäftsbereich haben. In meiner Berufskarriere ist mir eine fundamentale Strategieänderung des Providers gleich zweimal passiert mit äußerst unangenehmen Konsequenzen für die betroffenen Geschäftsbereiche. Es kann aber auch schon ausreichen, damit Projekte in Schwierigkeiten geraten, wenn bei den beteiligten externen Firmen Schlüsselpersonen kündigen.

Das bis hierher Gesagte gilt für die Einführung von völlig neuen Lösungen. Es gibt aber noch eine andere Kategorie von Projekten, die bei den Projektteilnehmern weniger beliebt ist und die eine eigene Problematik besitzt: die Umstellung (Migration) einer bestehenden Lösung auf eine neue. Hierbei kann man zwischen technischer und fachlicher Umstellung unterscheiden. Bei technischen Migrationen (zum Beispiel die Umstellung von Windows 2000 auf Windows XP oder der Wechsel des E-Mail-Systems) gibt es keinen erkennbaren zusätzlichen Nutzen für die Benutzer, dafür aber unübersehbare Unannehmlichkeiten: Neues Erlernen der Software mit voraussehbaren Fehlbedienungen zu Beginn, sowie Testen aller mit diesem System kommunizierenden Komponenten (bei Windows XP wären das im Grunde alle benutzten Programme). Für die IT sind derartige Projekte oft eine technische Notwendigkeit, verbunden mit der Schwierigkeit, die Budgets hierfür genehmigt zu bekommen, da der Nutzen für den Betrieb nicht erkennbar ist, sowie ein erheblicher personeller und zeitlicher Aufwand. Dann gibt es die fachlichen Migrationen, bei denen zum Beispiel ein Buchungssystem abgelöst und durch ein neues ersetzt wird. Solche Projekte können je nach Komplexität des Systems viele Monate, manchmal sogar Jahre dauern und erzeugen viele Probleme:

- Die Daten des alten Systems müssen in das neue transferiert werden. Große Systeme haben Tausende von Dateien, wobei die Bedeutung mancher Datei nicht unbedingt klar ist (schon gar nicht bei dem neuen System). Es müssen spezielle Programme eigens für solche Datenmigrationen geschrieben werden, da man auf dem Markt nur ganz selten Software hierfür findet. Es gibt quasi eine Garantie dafür, bei dieser Aufgabe Fehler zu begehen.
- Die Akzeptanz bei den Benutzern kann sehr variabel sein, da nicht für jeden mit der Umstellung auf ein neues System ein Vorteil verbunden ist.
- Der Testaufwand in den Fachbereichen kann alle Grenzen sprengen und durch die lange Projektphase geht irgendwann die Lust, das *Momentum*, verloren. Wenn dann das neue System auch noch verhältnismäßig viele Fehler aufweist (jede Software hat Fehler), dann wird es kritisch für das gesamte Projekt.
- Während der Umstellungsphase wird selbstverständlich im alten System nichts mehr geändert. Dies bedeutet dann eine Periode des Stillstands für den Betrieb, was in Zeiten der Notwendigkeit zu schnellen Reaktionen am Markt schlimme Folgen haben kann.
- Das alte Know-how wird vernichtet, und man startet mit einem System, das weder von der IT noch von den Benutzern ausreichend beherrscht wird. Man ist gut beraten, Leute einzustellen, die dieses Know-how mitbringen. Das wirkliche Beherrschen eines großen Systems, wie einer Bankenbuchungssoftware, kann Jahre dauern. Es ist vergleichbar mit dem Erlernen einer neuen Sprache: Auch beim besten Willen und mit der besten Schulung wird man erst die notwendige Gebrauchssicherheit haben, wenn man über einen längeren Zeitraum praktisch damit gearbeitet hat. Mich erstaunt gelegentlich, mit welcher Leichtfertigkeit manche Firmen entscheiden, ihre Hauptsysteme auszutauschen und sich nachher naiv wundern, dass sie Probleme über Probleme haben. In vielen Fällen hat man dann die Pest durch die Cholera ersetzt.

Als letzter Schritt bei der Einführung eines Systems muss dann noch ein so genannter Stresstest durchgeführt werden. Hierbei geht es darum, durch

Simulation sicherzustellen, dass das System große Datenmengen in dem vorgegebenen Zeitrahmen verarbeiten kann. Wer hat nicht schon in den Medien gehört, dass die Systeme zu langsam oder zu instabil wurden, wenn alle Anwender das System zu nutzen begannen (Software für Hartz 4, LKW-Maut, ...)? Da in diesen Tests die Wirklichkeit (die man ja noch nicht kennt) simuliert werden muss, sind solche Tests ausgesprochen schwierig zu organisieren und eben oft nicht erfolgreich.

Wenn dann irgendwann, nach langer Zeit und viel Ärger die Implementierung abgeschlossen ist, geht es in den Livebetrieb (Produktion). Und dort passiert, was immer passiert: Das System benimmt sich nicht so, wie das in den Tests der Fall war. Die ersten Tage und Wochen mit einem neuen System sind meistens ein Alptraum mit vielen Nachtschichten, Urlaubssperre und rebellierenden Anwendern.

Wie ist das möglich?

Nun, die Tests wurden in einer Testumgebung durchgeführt und diese ist lediglich ein kopiertes Biotop. Die Produktionsumgebung ist in vielen Punkten total anders. Es sind mehr Fische im Teich, es gibt ein Kommen und Gehen an der Futterstelle (entspricht externen Anbindungen von Kunden und Lieferanten), und überhaupt ist das Hauptbiotop viel größer.

Und dann gibt es noch das psychologische Element, das zum Scheitern von IT-Projekten führen kann. Die Fachzeitschriften sind voll mit Beispielen. Wenn Schlüsselpersonen nicht von der Einführung eines Systems überzeugt sind («Das Alte funktioniert doch noch», «Ich habe das immer so gemacht, warum sollte ich das jetzt nach so vielen Jahren plötzlich ändern?», «Das ist wieder so ein Ding, das braucht doch kein Mensch»,...), dann wird das Projekt mit absoluter Sicherheit scheitern, und zwar spätestens dann, wenn der Produktionsbetrieb beginnt. Die Einbindung aller Beteiligten ist von allergrößter Wichtigkeit. Aber es wird in viel zu vielen Fällen einfach nicht gemacht. Das beste System der Welt wird es dann nicht schaffen, sich durchzusetzen. Manchmal versuchen auch Beteiligte am Anfang des Projektes eine andere Lösung zu bekommen als die, die dann letztendlich entschieden wird. Nachdem sie den Kampf verloren haben, verstecken sie ihre Ablehnung, um dann beim Produktionsbetrieb bei der ersten Schwierigkeit die gewählte Lösung zu boykottieren («Seht Ihr! Ich habe es von Anfang an gesagt!»). Dagegen ist man zunächst einmal machtlos.

Nach einiger Zeit im Livebetrieb werden Dinge wichtig, die vorher nur eine untergeordnete Rolle spielten, nämlich: Wie einfach oder kompliziert ist die tägliche Benutzung eines Systems (Benutzerfreundlichkeit) und wie einfach lässt es sich an veränderte Begebenheiten anpassen? Wenn die Antworten auf diese Fragen negativ ausfallen, dann kann es durchaus zu einer zeitverzögerten Ablehnung des Systems kommen. Es ist eigentlich immer so: Diese beiden Elemente spielen keine entscheidende Rolle bei einer Systemauswahl, und dies ist ein Fehler.

Regelrecht verhasst in der IT sind Fusionen zwischen zwei Firmen.

Neben den bekannten generellen Problemen einer Fusion (wenig Motivation bei

den Mitarbeitern, Angst vor dem Arbeitsplatzverlust, kulturelle Differenzen,...) gibt es noch zusätzliche rein IT-bezogene Schwierigkeiten. In einer derartigen Situation müssen sich zwei existierende IT-Biotope gegenüber ihrem jeweiligen Konkurrenten messen lassen. Dabei wird ein IT-Zielbiotop bestimmt. Es kann sein, dass ein Biotop komplett aufgegeben wird, oder dass ein neues Biotop als Mischung aus Teilen der beiden existierenden Biotope angelegt wird. Die Schwierigkeiten, ein neues und funktionierendes IT-Biotop anzulegen sind monströs und gehen an die Substanz und die Belastbarkeitsgrenze aller IT-Mitarbeiter. Bei diesem Vorgang wird eine ganze Reihe von Systemen umgestellt oder stillgelegt und die Datenbestände werden in neue Systeme migriert. Einige Systeme werden gar ganz aufgegeben. Das Mitarbeiten in einem derartigen Projekt ist ein extremes und intensives Erlebnis, das man so schnell nicht vergisst, und jeder IT-Mitarbeiter sollte daher einmal in seiner Karriere eine solche Situation mitmachen. Einmal reicht dann aber auch wirklich für das ganze Leben.

Und warum kostet es eine Million (oder noch mehr)? Obige Ausführungen belegen, dass eine Implementierung Kosten generiert. Es gibt aber noch weitere Kosten: Software und Hardwaresysteme sind bereits sehr teuer in der Anschaffung. Dass jedes Jahr für Software bis zu 20 % des Anschaffungswertes an Wartung aufgebracht werden muss, haben wir bereits gesehen. Darüber hinaus müssen sich Mitarbeiter um die Systeme kümmern. Es ist nicht wie bei einem Auto, das einmal im Jahr in die Werkstatt muss. Nein, jedes System braucht tägliche Pflege und Überwachung durch eine entsprechend ausgebildete Mannschaft. Da Mitarbeiter auch in Urlaub gehen, müssen immer mindestens zwei Mitarbeiter geschult werden und als Administratoren die Verantwortung übernehmen. Dass IT-Mitarbeiter nicht die billigsten Mitarbeiter in einem Betrieb sind, ist bekannt.

Weniger bekannt ist die Tatsache, dass bei der Anschaffung einer Softwarelösung mehr als nur eine Maschine angeschafft werden muss. Jeder Privatmann hat normalerweise einen PC, und das war's. Nicht so im Betrieb: Benötigt wird eine Produktionsmaschine, eine Notfallmaschine (siehe Kapitel *Nach ersten Schätzungen geht der Schaden in die Millionen*), eine Testmaschine und vielleicht auch noch eine Entwicklungsmaschine (um Software zu entwickeln). Bei jeder Neuanschaffung erhöht sich außerdem der Bedarf an die Kapazität der Klimaanlage und an das Notstromaggregat. Dokumentationen müssen nachgeführt, Prozesse und Datensicherungsstrategien müssen angepasst werden.

In diesem Kapitel werden die zu lösenden Probleme und die anfallenden Kosten, um das neue System sicher zu machen, nicht behandelt. Hierfür wurde ein eigenes Kapitel reserviert (siehe Kapitel *IT-Sicherheit*).

Die Thematik der Abhängigkeiten zwischen den einzelnen Komponenten wurde oben kurz beschrieben. Um es besser zu verdeutlichen, hier eine konstruierte Übertragung in den privaten Bereich (etwas überspitzt), parallel zur IT (sehr real).

In Ihrem Haus tropft der Wasserhahn. Die Verbuchungssoftware hat einen Sie analysieren die Ursache und finden Fehler, der nur bei einem bestimmten

heraus, dass die Dichtung kaputt ist. Sie gehen in den Baumarkt um eine Neue zu kaufen, aber der Verkäufer teilt Ihnen mit, dass es diese Größe seit Kurzem nicht mehr gibt. Er zeigt Ihnen die Möglichkeit auf, eine auf Maß anfertigen zu lassen, rät aber gleichzeitig aufgrund der Kosten und der langen Wartezeit davon ab. Er schlägt Ihnen vor einen neuen Wasserhahn zu kaufen, was Sie auch tun.

Zu Hause versuchen Sie den Wasserhahn zu installieren, stellen aber schnell fest, dass er eine unterschiedliche Größe hat und nicht ins bestehende Waschbecken passt. Sie gehen zurück zum Baumarkt und möchten den neuen Wasserhahn zurückgeben. Der Verkäufer lehnt ab mit dem Hinweis, dass die Größe auf der Packung gestanden hätte.

Sie beratschlagen sich mit Ihrem Partner und kommen zum Schluss, dass das Waschbecken eh alt ist und eine modernere Farbe der Sache nicht abträglich wäre. In der Zwischenzeit tropft der Wasserhahn munter weiter.

Sie bestellen das neue Waschbecken, warten eine gewisse Zeit und machen sich selbst an die Installation. Bei diesem Vorgang brechen ein paar Fliesen ab. Einen Weg zurück gibt es jetzt nicht mehr. Sie entscheiden sich, neu zu fliesen. In der Zwischenzeit waschen Sie sich beim Nachbar oder wenn es regnet.

Kaum haben Sie die Fliesen installiert, stellen Sie fest, dass die Tür sich jetzt nicht mehr schließen lässt, da die Fliesen zu dick sind. Sie entscheiden, die Tür aufzulassen. Ist reine Gewöhnungssache.

Ihrer Kunden auftritt. Sie wenden sich an den Provider, der Ihnen mitteilt, dass die eingesetzte Version veraltet ist, und bietet Ihnen als einzige Möglichkeit ein kostengünstiges Upgrade auf die aktuelle Version an. Sie folgen seinem Ratschlag.

Bei den Tests der neuen Version stellen Sie fest, dass plötzlich die Zinsbeträge mit einer neuen (für den Betrieb nicht brauchbaren) Formel berechnet werden. Der Provider gibt daraufhin zu bedenken, dass dieses Problem bekannt ist und dies in der Produktbeschreibung dokumentiert ist. (Beim Verkauf war hiervon natürlich keine Rede).

Sie beauftragen einen teuren Berater, um die Situation zu analysieren. Er rät Ihnen, ein teures Zusatzmodul in der Software anzuschaffen, um das neue Problem zu beheben. Die neuen Funktionalitäten kann der Betrieb eh gebrauchen. Die Kundenbuchungen sind nach wie vor falsch.

Bei den Tests des neuen Moduls bricht der Computer unter der zusätzlichen Last zusammen. Sie haben keine Wahl und kaufen einen größeren Server. Dem Kunden helfen Sie bei seinem Problem, indem ein paar Mitarbeiter manuelle Korrekturen durchführen.

Als Sie die Tests fast abgeschlossen haben, merken Sie, dass die Datensicherung nicht mehr einwandfrei funktioniert. Sie installieren eine temporäre Lösung, die sich aller Voraussicht nach mehrere Jahre halten wird.

Der große Tag ist da und alle Arbeiten sind abgeschlossen. Sie drehen den neuen Wasserhahn auf. Alles funktioniert bestens, aber nach 2 Tagen fängt das Telefon an zu klingeln, wenn Sie den Wasserhahn aufdrehen. Als Sie bei der Telefongesellschaft nachfragen, wird Ihnen gesagt, dass dieser Typ Wasserhahn eine neuartige Metalllegierung hat und deshalb mindestens 1 Meter von der nächsten Telefonleitung entfernt sein muss.

Ansonsten

kann es zu einer Interferenz kommen.

Sie erwägen die Möglichkeit, den Wasserhahn erneut auszutauschen, lassen davon ab.

Die IT erwägt, die neue Software zu deaktivieren, lässt aber davon ab, da die Risiken zu groß sind.

Ihr Partner hat daraufhin genug von der Baustelle und schlägt vor ein neues Haus zu kaufen, bei dem diese Probleme nicht bestehen.

In dieser Situation schlägt der Geschäftsbereich vor eine komplette neue Software zu kaufen, die auf einen Schlag alle Probleme lösen wird.