

# Messbarer Wertbeitrag und schlanke Software (-Entwicklung) -

## Zwei Aspekte eines „Business Alignment der IT“

---

### 1 Einleitung

Wir haben uns daran gewöhnt, dass die Entwicklung der Märkte den auf den Unternehmen lastenden Kosten- und Wettbewerbsdruck beständig erhöht. Gerade unter den aktuellen Rahmenbedingungen sind Kostensenkungsprogramme an der Tagesordnung und der Rechtfertigungsdruck für Investitionen jeder Art ist größer denn je. Unternehmen sind gezwungen, kontinuierlich alle geschäftlichen Aktivitäten auf den Prüfstand zu stellen, unter dem ausschließlichen Licht ihres Wertbeitrags zum Unternehmenserfolg zu bewerten und konsequent auf dessen Maximierung auszurichten. Umso bemerkenswerter ist es da, dass gerade die bedarfsgerechte Gestaltung des Wettbewerbsfaktors IT in vielen Fällen noch immer auf der Betrachtung technikdominierter Zusammenhänge und Messgrößen basiert, die Unternehmensführung und Geschäftsfunktionen (im Weiteren in Gegenüberstellung zur IT als **Business** bezeichnet) kaum verstehen und noch weniger wirklich bewerten können. Fragestellungen wie: „Geben wir für unsere IT zu viel aus?“, „Welchen Gegenwert erhalten wir dafür?“, „Tun die wirklich, was wir brauchen?“ oder „Was genau tun die eigentlich?“ sind so kaum zu beantworten. Die IT bleibt oft einer der letzten zentralen Wettbewerbsfaktoren, der sich einer wirklich geschäftsorientierten Betrachtung entzieht.

Wirtschaftlicher Erfolg aber ist nicht nur davon abhängig, dass wir das, was wir tun, auch gut tun (also „Operational Excellence“ erreichen), sondern vor allem von der gezielten Konzentration der Aktivitäten auf die Handlungsfelder und Maßnahmen, die für den Unternehmenserfolg am wichtigsten sind. Erst dies führt zur allein entscheidenden „Business Excellence“. Business Alignment ist die geeignete Disziplin zur Erreichung dieses Ziels auch in der IT. Business Alignment zielt unter anderem darauf ab, dass die an den IT-Erstellungsprozessen beteiligten Personen (insbesondere also auch diejenigen, die Anwendungen entwickeln) wissen, worum es geht und was die zukünftigen Anwender brauchen, und dass dadurch fachliches Domänenwissen auf direktem Weg z.B. in die zu erstellenden Anwendungen einfließen kann. Die Forderung, dass genau diese Kenntnis der Zusammenhänge eine erfolgskritische Komponente für die erfolgreiche Anwendungsentwicklung ist, findet sich in zahlreichen Ansätzen der Domänenmodellierung und der domänenspezifischen Sprachen [EVA06] und genauso in den Ansätzen der Produktliniengestaltung [Poh08]. Nimmt man an, dass eine „business-aligned“ IT über ausreichend Domänenkenntnisse verfügt, um die Fachbereiche zu verstehen und in ihrer Anforderungspriorisierung zu unterstützen, dann lassen sich Softwareprozesse schlank gestalten. Vollständigkeit und komplette Konsistenz von Anforderungs- und Spezifikationsdokumenten sind nicht mehr notwendig, stattdessen lassen sich auf abstrakter Ebene Klärungen von Inhalten vorantreiben, Prioritäten definieren und schlanke Lösungen entwerfen. Dies kann mindestens dann ein sinnvoller Ansatz sein, wenn zukünftiger Anwender und Entwickler Hand in Hand arbeiten können, ohne dass sie von vertraglichen Konstrukten separiert werden. Ideen dieser Art werden im No-Frills-Ansatz formuliert [GS09].

## 2 Business Alignment

Seit mehr als 15 Jahren werden Alignment-Konzepte [Hen92] und die eng damit verbundene Serviceorientierung bzw. IT-Service-Management (ITSM) [ITIL1] als Lösungsmodelle diskutiert, und spätestens mit ITIL Version 3 [ITIL3] sollten diese eigentlich als etabliert gelten. Eine Ursache für die bisher dennoch geringen Umsetzungserfolge sieht etwa Peter Hinssen [Hin08] in der Tatsache, dass Business Alignment im Kern keinen mit einer möglichst geringen Anstrengung zu erreichenden Endzustand beschreibt, sondern vielmehr für die andauernde Verpflichtung für IT und Business gleichermaßen zu einem besseren gegenseitigen Verständnis steht. Zur Charakterisierung des heutigen Zustands zieht er dabei einen vermutlich nicht ganz unzutreffenden Vergleich: „IT is much like the dentist. People need dentists, but they’re never going to be our friends” [Hin08].

Will man vor diesem Hintergrund eine kleine phänomenologische Betrachtung anstellen, muss man vor allem zwei Faktoren Beachtung schenken. Zum einen der simplen Tatsache, dass die IT gegenüber ihren Mutterunternehmen in den weitaus meisten Fällen branchenfremd ist, ohne natürlichen Zugang zum jeweiligen Geschäftsgegenstand und mit einem völlig anderem Sprachrepertoire ausgestattet. Hinzu kommen, geprägt durch Ausbildung und Mentalität, beinahe gegensätzliche Denkweisen: Dem vorwiegend objektiv analytischen Ansatz der „IT’ler“, der von möglichen Funktionalitäten ausgeht und sich dann vor allem mit deren technischer Beherrschbarkeit beschäftigt, steht die eher intuitive Herangehensweise von Business-Leuten gegenüber, bei der zunächst das Vertrauen in eine Lösung und deren erkennbare Tauglichkeit für das Geschäft dominieren.

Schon durch existierende Sprachbarrieren begründete Kommunikationsprobleme und ein latentes beiderseitiges Fremdsein dürfen also nicht grundsätzlich verwundern – angesichts der Bedeutung der IT für den Unternehmenserfolg wohl aber die anhaltend geringe Entschlossenheit zur nachhaltigen Beseitigung dieser Kluft. Dabei steht Business Alignment doch eigentlich für eine einfache und sowohl im Alltag als auch im Umgang mit den primären Geschäftsfunktionen allgemein unbestrittene Handlungsmaxime, nämlich die zur Erreichung eines Ziels (in diesem Fall des Unternehmenserfolgs) erforderlichen Mittel (in diesem Fall

die IT) aktiv, vorausschauend und nachhaltig am Ziel selbst auszurichten und bewusst auf maximale Zweckdienlichkeit hin zu gestalten. Vielleicht liegt es an einer in der öffentlichen Diskussion oft zu beobachtenden Sichtverengung beinahe ausschließlich auf Strategie Alignment, dass Business Alignment trotzdem allzu häufig als eine eher akademische Übung gilt, und damit

zum Streichposten gegenüber meistens höher priorisierten operativen Maßnahmen wird. Unserer Überzeugung nach umfasst Business Alignment jedoch ein deutlich weiteres Aufgabenfeld, dessen einzelne Gestaltungsbereiche als ganzheitliches IT-Handlungsmodell in **Abb. 1** dargestellt sind. Für die weitere Betrachtung wollen wir uns dabei auf zwei Bereiche von besonderer Bedeutung konzentrieren: Serviceorientierung und Projekt- bzw. Softwareentwicklung.



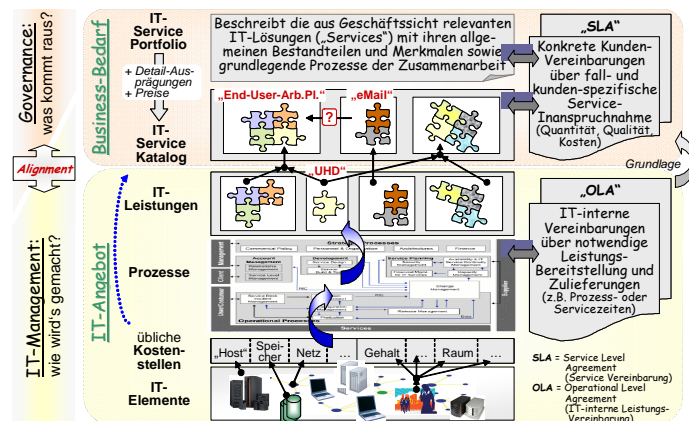
### 3 Serviceorientierung

Wie einleitend dargestellt entziehen sich die IT und ihr Leistungsangebot traditionell einer wirklich fundierten, d.h. geschäftsbezogenen, Bewertung durch das Business. Fehlender Zugang zur branchenfremden Disziplin, ein technikgetriebenes Vokabular der „IT’ler“ und oft ebenfalls eher technikorientierte Kennzahlensysteme machen Bewertungen z.B. der Preiswürdigkeit des IT-Leitungsangebots oder seiner Relation zum generierten Nutzen für Business-Leute kaum möglich. Die Folge ist häufig eine Art Kapitulation und Sichfügen in bestimmte die IT betreffende Entscheidungen („die versteht man ja doch nicht“), verbunden mit latenter Unzufriedenheit und Misstrauen. Eine Situation, die keineswegs nur für das Business unbefriedigend ist. Auch für die IT ist das daraus entstehende Klima der Zusammenarbeit oft nur mäßig motivierend, vor allem aber bedeutet die resultierende Zähigkeit von Genehmigungsprozessen und ganz besonders die häufiger auftretende Notwendigkeit, getroffene Entscheidungen auch nachträglich immer wieder rechtfertigen – oder sogar revidieren – zu müssen, erhebliche Reibungsverluste.

Serviceorientierung ist das geeignete Mittel, diesen Kreislauf zu durchbrechen und eine tragfähige Plattform zu etablieren, auf der beide Seiten gleichberechtigt an der Ausrichtung und Gestaltung der IT mitwirken können, ohne vom Business technologisch bestimmte Erörterungen oder von der IT die Aufgabe ihrer technologischen Gestaltungskompetenz verlangen zu müssen. Im Übrigen wird hier ganz bewusst der Begriff „Serviceorientierung“ verwendet und dem Terminus „Service-Management“ vorgezogen. Denn für unser Verständnis von Business Alignment, das den Wirkungsgrad der IT nachhaltig steigern hilft, bedarf es nicht nur eines Prozesses zum Management von Services, sondern eben einer in die Kultur des Miteinanders von IT und Business eingreifenden umfassenden Serviceorientierung.

Im Gegensatz zu ansonsten üblichen Optimierungen der IT, die viel zu oft aus der Sicht von innen nach außen vorgenommen werden, verfolgt die Serviceorientierung den strikten Ansatz, alles aus der Sicht des Business – also des IT-Kunden – zu betrachten. In den Vordergrund gerückt wird damit der konkret formulierte Bedarf des Business an IT-Leistungen, dem sich die Angebotsgestaltung der IT anzupassen hat. Die beiden zentralen Elemente bei der Etablierung von Serviceorientierung sind dabei die Erarbeitung eines Service-Portfolios, in dem die zu businessrelevanten Lösungen (den „IT-Services“) gebündelten IT-Leistungen beschrieben werden, sowie ein Kostenmodell, das die IT-Kosten auf diese vom Business bewertbare Lösungsebene aggregiert, statt sie auf rein technische Kostenstellen abzubilden. Alle weiteren Maßnahmen bauen mehr oder weniger direkt auf diesen beiden Kernelementen auf. **Abb. 2** zeigt ein IT-Service-Modell, dessen Elemente im Folgenden kurz behandelt werden.

Im **IT-Service-Portfolio** wird die Gesamtheit der IT-Leistungen beschrieben, aber eben nicht als eine Menge IT-spezifischer Prozesse („Availability Management“, „Change Management“ etc.) oder technologischer Dienstleistungen (z.B. „Server-Administration“), sondern als für das Business aussagefähige Lösungen, die in dessen Sprache formuliert und von ihm in ihrem Wert zu beurteilen sind (z.B. „Bereitstellen eines IT-Arbeitsplatzes“ oder „Betreiben einer fachlichen Anwendung“). Dabei empfiehlt es sich, bei der For-



mulierung der Services auf eine gewisse Marktgängigkeit zu achten, damit später auch ein Vergleich mit externen Serviceanbietern möglich ist und die Wettbewerbsfähigkeit einzelner Leistungsangebote objektiv festgestellt werden kann. Hierdurch kann einerseits das Business die Preiswürdigkeit empfangener Leistungen und andererseits die IT besondere Optimierungspotenziale (im Einzelfall bis hin zum sinnvollen Bezug externer Teilleistungen) erkennen.

Der **IT-Service-Katalog** stellt die Operationalisierung des IT-Service-Portfolios dar, in dem er für die Inanspruchnahme von Services wichtige Detailinformationen hinzufügt (z.B. Rechnertyp und Ausstattung eines Arbeitsplatzrechners) und außerdem eine Handhabungsplattform für den konkreten Abruf einzelner Services bereitstellt (im Idealfall ein elektronisches Bestellsystem). Daneben werden für die meisten Services mehrere Ausprägungen angeboten, die mit unterschiedlichen Leistungsniveaus („Service-Level“) und Kosten verbunden sind. Zum Beispiel: Der Betrieb einer Anwendung im Service-Level „Gold“ bedeutet maximal zwei Stunden Ausfall je Monat und besonders gute Antwortzeiten. Service-Level „Silber“ und „Bronze“ entsprechen niedrigeren Leistungsversprechen bei niedrigeren Preisen.

Dadurch setzt Serviceorientierung äußerst hilfreiche Selbstregulierungsmechanismen in Gang, die Kosten- und Leistungseffizienz der IT steigern. So gestalten sich in traditionellen Konzepten Diskussionen darüber, welche Leistungsmerkmale (z.B. Betriebszeiten Nachts- oder an Wochenenden, Wiederanlaufzeiten im Störfall) für bestimmte Anwendungen benötigt werden, oft schwierig bis fruchtlos und enden deshalb häufig notgedrungen in der Implementierung von Maximalforderungen. Auf der Grundlage eines etablierten Servicekatalogs kann nun das Business selbst und in eigener Abwägung von Bedarf und Kosten den für die bestellte Anwendung gewünschten „Service-Level“ wählen. Nicht selten ergeben sich allein dadurch kostenoptimierende Regulierungseffekte insbesondere in der Infrastruktur. Darüber hinaus wird die IT aus ihrer Spielverderber-Rolle befreit, weil anstelle von für das Business oft nicht nachvollziehbarer Ablehnung bestimmter Lösungsanforderungen nun eine vom Kunden (also dem Business) ausgehende Selbstregulierung auf Kosten-/Nutzen-Basis tritt.

Um dies zu ermöglichen, bedarf es auch eines für das Business transparenten **Kostenmodells**. Hier finden sich heute vielfach technologielastrige Ausprägungen, deren geschäftsorientierte Aussagefähigkeit entsprechend begrenzt ist. So ist die Erkenntnis, dass etwa der Großrechner oder das Netzwerk besondere Kostentreiber sind, von nur geringem Wert und kaum zur Ableitung wirksamer Maßnahmen für die Optimierung des Wirkungsgrads der IT geeignet. Ziel muss sein, die einzelnen Kostenelemente auf die definierten IT-Services abzubilden, weil nur dort eine Bewertung im geschäftsrelevanten Kontext und in der Abwägung von Kosten und Nutzen für das Business möglich ist. In der Praxis wird dies nicht im ersten Schritt vollständig gelingen. Durch schrittweise Abbildung zunächst von der technischen Ebene auf Prozesse und dann weiter auf aggregierte IT-Leistungen und schließlich IT-Services erreicht man mit überschaubarem Aufwand und in vertretbarer Zeit jedoch schon recht gute Annäherungen.

Unter strategischem Gesichtspunkt bedeutet die Serviceorientierung für das Business qualifizierte Governance-Möglichkeiten, ohne die Gefahr, sich in technologischem Sprachwirrwarr zu verirren. Über die Gestaltung des IT-Service-Portfolios kann die IT effektiv auf die Ziele und Bedürfnisse des Unternehmens ausgerichtet werden. Aus der laufenden Betrachtung der Wirtschaftlichkeit in der Bereitstellung der Services können der IT geeignete Vorgaben gemacht und über ihren Einfluss auf die Wettbewerbsfähigkeit der Services schließlich sogar eigentlich technologische Investitionsentscheidungen anhand ihrer Business-Relevanz getroffen werden. Im Gegenzug kann die IT weiterhin ihre spezifische Technologiekompetenz ohne Einmischung seitens des Business ausspielen und die IT-dominierten Domänen IT-Prozesse und Technologie weitgehend autonom gestalten – allerdings unter Befolgung der gegebenen Rahmenvorgaben.

Die Formulierung von IT-Services gestattet übrigens einen relativ großen Gestaltungsfreiraum, der unter Berücksichtigung nicht nur objektiven Bedarfs, sondern auch kultureller oder unternehmenspolitischer Aspekte, im gemeinsamen Gestaltungsprozess mit dem Business auszufüllen ist. So kann durchaus ein Service-Portfolio mit nur 5 oder 6 IT-Services den gesamten Leistungsumfang eines IT-Bereichs – aus Businesssicht (!) – abdecken, je nach Umfeld und konkreter Zielsetzung kann jedoch auch ein 20 – 30 Services umfassendes Portfolio sinnvoll sein. Um den Unterschied zu veranschaulichen, betrachten wir noch einmal den IT-Service „Bereitstellen eines IT-Arbeitsplatzes“. Im ersten Fall eines hochkomprimierten Portfolios würde man diesen Service typischerweise so definieren, dass er immer aus einer großen Zahl von Einzelleistungen besteht (Rechner + Drucker + Internet-Zugang + E-Mail-Anschluss + Speicherplatz für persönliche Daten + Unterstützung durch Helpdesk + ...), was vor allem bei einer homogenen Anwendergruppe durchaus empfehlenswert ist und zu einem hohen Standardisierungsgrad führt. Die einzelnen Leistungsbausteine „E-Mail“, „Internet-Zugang“ etc. würden in unser obiges Servicemodell dann als gebündelte „IT-Leistungen“ einfließen. Demgegenüber kann man den Leistungsumfang des Service im anderen Extrem auch auf die ausschließliche Versorgung mit einem Arbeitsplatzrechner beschränken und die restlichen Bausteine als separate IT-Services anbieten. Während die Vorteile des ersten Ansatzes in hoher Standardisierung und Kosteneffizienz liegen, räumt der zweite Ansatz dem Business freiere Gestaltungsmöglichkeiten und mehr Flexibilität ein.

Weiter oben wurde auf die Erwartung hingewiesen, dass das IT-Service-Portfolio, aus Sicht des Business, das gesamte IT-Leistungsspektrum abdecken sollte. Diese Aussage muss zumindest für den Handlungsbereich Projektentwicklung ein wenig relativiert werden. Zwar umfassen IT-Services üblicherweise auch die direkt mit Projekten zusammenhängenden IT-Leistungen, können dies – wegen der geringen inhaltlichen Bestimmbarkeit von Projekten – in der Praxis jedoch nur formal schaffen, z. B. indem sie differenzierte Verrechnungspreise für unterschiedliche IT-Rollen oder IT-Tätigkeiten festschreiben. Die inhaltliche Gestaltung von Projekt- und insbesondere Softwareentwicklung, dabei im Sinne des Business Alignment vor allem die Gestaltung der Zusammenarbeit von Business und IT, bleibt jedoch eine eigenständige Aufgabe und wird im folgenden Kapitel ausführlicher behandelt.

## 4 No Frills Software Engineering

Im vorigen Abschnitt haben wir erörtert, dass die strikte Orientierung an den Services, die aus Geschäftssicht erforderlich sind, ein wesentliches Element des Business Alignment ist. Innerhalb der Softwareentwicklung wird die Fokussierung auf den Geschäftsnutzen (als wesentliches Merkmal des Business Alignment) durch die Grundsätze des "No Frills Software Engineering (NFSE)" unterstützt [GS09].

Die Grundidee des NFSE ist, dass traditionelle Vorgehensweisen und Aufgaben in der Softwareentwicklung im Hinblick auf ihren Nutzen und ihre Effizienz hinterfragt werden. Bei Billigfliegern, Direktversicherern und bestimmten Angeboten von Telekommunikationsanbietern haben solche Herangehensweisen zu schlanken Dienstleistungen und neuen Preismodellen geführt. Unsere Annahme ist, dass die schnickschnackfreie Softwareentwicklung zu schlanker Software und niedrigen Herstellungskosten führt. Im Folgenden erörtern wir zwei der Grundelemente des NFSE detaillierter. Weitere Elemente sind in [GS08] diskutiert. Zu den hier nicht näher diskutierten Elementen gehören die promi-

nente und durchgängige Verankerung des Anwendungswissens im gesamten Entwicklungsprozess [GPR07], die Unterstützung von Abstraktion in allen Dokumenten und die explizite Repräsentation und das Management von Ungewissheit [LR02].

## 4.1 Wertorientierte Softwareentwicklung

Die Idee der wertorientierten Unternehmenssteuerung ist es, dass alle Aufgaben auf den Unternehmenswertzuwachs ausgerichtet werden [Koc05]. Die Idee der wertorientierten Softwareentwicklung ist es, dass alle Aktivitäten der Softwareentwicklung daraufhin untersucht werden, ob sie zu (geschäfts-) nützlicher Software beitragen und solche, die das nicht tun, neu zu orientieren oder wegzulassen [BAB06].

Viele Projekte finden unter der Rahmenbedingung eines Anwendungsstaus statt. D. h., Änderungs- und Erweiterungswünsche der Anwender sind über einen längeren Zeitraum nicht erfüllt worden, weil eine neue Software in irgendeiner Weise absehbar war. Die Entscheidung für Beschaffung oder Entwicklung einer Software dauert dann doch länger als geplant und schon haben sich Anforderungen der Anwender gestaut. Werden dann die Schleusen geöffnet und die Anwender nach ihren Anforderungen befragt, kommen Wünsche ganz unterschiedlicher Dringlichkeit und Wichtigkeit zum Vorschein, und das in der Regel nicht in sinnvoller Priorisierung. Sehr oft ist es so, dass Details, die am aktuell im Einsatz befindlichen System als fehlend oder störend empfunden werden, überhöht werden. Aktuelle Anforderungen werden verhältnismäßig viel zu hoch bewertet. Und – am allerschlimmsten – Detailanforderungen für Neuerungen werden in epischer Breite erörtert und mit dem Hinweis, dass das neue System natürlich alles können soll, was das alte auch schon konnte, versehen. Selbst wenn man diese Vielfalt ordnet und priorisiert (was nur vor dem Hintergrund von Anwendungswissen aus der betroffenen Domäne funktionieren kann), ist eines klar: Anwendungen werden von Generation zu Generation immer komplexer, minimale Rückschritte hinter den Automatisierungsgrad werden in ursächlichen Zusammenhang mit einem drohenden Weltuntergang gebracht, detaillierte Erweiterungen werden als wesentlicher Funktionalitätsgewinn überbewertet. Auf diese Weise wuchert Software selbst dann, wenn keine Zeile Code wiederverwendet wird. Ein Instrument um dem Wuchern entgegen zu treten ist der beherrzte Einsatz von so genannten, mit guten Nerven ausgestatteten Business Analysten, die die tatsächlich zu unterstützenden Geschäftsprozesse kennen und im Crescendo der Beteuerung der Unverzichtbarkeit von Anforderungen der allerhöchsten Priorität die Ruhe bewahren und Wichtiges von Unwichtigem trennen. Die entlarvende Frage (und damit das wichtigste Instrument) dabei ist: „Was passiert, wenn die Funktionalität A nicht oder später realisiert wird?“ Abgesehen von weiteren Illustrationen der bereits diskutierten Weltuntergangsszenarien, ist die Antwort – wenn sie denn erst von rhetorischem Gestrüpp und Eitelkeit befreit ist – oft genug ganz einfach: nichts. Dieses Muster findet sich besonders oft im Hinblick auf zu erstellende Berichte und Statistiken. In der Entwicklung eines Bestandsführungssystems für eine Versicherungsgesellschaft drohte ungeheures Ungemach, weil ein spät geforderter Bericht nicht pünktlich zum Produktivbetrieb am Jahresanfang bereitgestellt werden konnte. Nach Eskalation über mehrere Hierarchiestufen und Drohszenarien bis hin zur Verschiebung des Produktivtermins stellte sich heraus, dass dieser Bericht zwei- (in Zahlen 2) mal jährlich benötigt wird und dass es gar keinen Berichtsweg gab, der ihn tatsächlich erforderte. Der Bericht war auf die Anforderungsliste geraten, weil das neue System ja alles können sollte, was das alte konnte.

Wertorientierung bedeutet in diesem Zusammenhang, Anforderungen auf ihre Substanz zu untersuchen, zu hinterfragen, welchen Geschäftsnutzen ihre Umsetzung stiftet, wieviele Anwender in welchen wie kritischen Geschäftsprozessen mit gewünschten Funktionalitäten zu tun haben und immer wieder

zu erörtern, ob es andere, billigere Umsetzungswege gibt. Hierdurch wird Softwareentwicklung lästig, potenzielle Anwender müssen Gewohnheiten zur Disposition stellen, nach schlanken Lösungen suchen und sich auf das Wesentliche konzentrieren. Aber die Mühe lohnt. Sehr häufig entpuppen sich Anforderungen als vermeidbar und Funktionalitäten als überflüssig. Und selbst wenn all das nicht eintritt, dann führt die Wertorientierung dazu, dass die gewünschte Funktionalität tatsächlich verstanden wird, dass ihre Rolle für die Anwender geklärt ist und dass Abweichungen davon risikobewertet werden können.

Den Grundsatz der Wertorientierung in der Softwareentwicklung setzen wir folgendermaßen um:

- Anwendung des Pareto-Prinzips: Wenn mit 20% des Entwicklungsaufwands 80% des Tagesgeschäfts unterstützt werden können (und wir glauben, dass das oft so ist), dann müssen die verbleibenden 20% des Tagesgeschäfts allein aus sich heraus rechtfertigen, dass Softwareunterstützung nötig ist.
- Automatisierung mit Augenmaß: Software erlaubt Automatisierung. Sinnvoll ist Automatisierung, wenn fehleranfällige Vorgänge oder solche mit hoher Wiederholungszahl softwareunterstützt werden können.
- Kontinuierliche Auseinandersetzung mit den Anwendern: Die immer wieder gestellte Frage "Was passiert ohne diese Funktionalität?" verbunden mit Anwendungswissen, das gewährleistet, dass die Antworten verstanden werden, führt immer wieder zu schlanken Lösungen und sinnvollen Priorisierungen.

## 4.2 Varianten von Softwareprozessmodellen

Software ist nicht gleich Software. Druckertreiber werden anders entwickelt als Informationssysteme, realzeitkritische Anwendungen anders als solche ohne Zeitanforderungen. Softwareprodukte erfordern andere Vorkehrungen als Individualsoftware. Technische Komplexität und Management-Komplexität prägen die Art und Weise, wie Software entwickelt werden sollte, ebenfalls erheblich. In der Dimension der technischen Komplexität reicht die Bandbreite von Anwendungen zur einfachen Verarbeitung wenig komplexer strukturierter Informationen bis hin zu dauerhaft zu betreibenden, hochskalierenden Anwendungen wie Telekom Switches. In der Dimension der Management-Komplexität reicht die Bandbreite von einfachen Anwendungen, die für wenige Benutzer vorgesehen sind bis hin zu Anwendungen, die von vielen heterogenen Anwendergruppen bei womöglich unterschiedlichen Unternehmen zum Einsatz kommen sollen. Allein diese schematische Betrachtung zeigt, dass Software nicht gleich Software ist und dass, völlig unabhängig von der funktionalen Ausgestaltung, wahrscheinlich unterschiedliche Entwicklungsansätze sinnvoll, wenn nicht sogar erforderlich sind, um die einzelnen Arten von Software zu entwickeln.

Selbst in der Softwareklasse "individuelle Informationssysteme" wie sie typischerweise in Versicherungsunternehmen anzutreffen ist, gibt es eine Bandbreite, die den Einsatz unterschiedlicher Softwareprozessmodelle nahe legt. Unter einem Softwareprozessmodell verstehen wir dabei die Aktivitäten, Ergebnistypen und Rollen, die in der Softwareentwicklung auftreten. Im Unterschied zu einem Vorgehensmodell ist ein Softwareprozessmodell konkret und legt Aktivitäten operational fest, definiert Muster von Ergebnissen und umfasst Prüflisten, mittels derer Ergebniseigenschaften festgelegt werden.

Nehmen wir an, dass eine Anwendung aus der genannten Klasse von Softwaresystemen erstellt werden soll, die gut verstanden ist und für die die Anforderungen vergleichsweise klar sind. Dann ist ein

an den Management-Erfordernissen (Kontrollierbarkeit, Meilenstein-Prüfbarkeit) orientiertes Prozessmodell ein sinnvolles Instrumentarium. Natürlich werden im Laufe der Entwicklung auch hier späte Anforderungen auftreten und natürlich werden auch hier die vereinfachenden Annahmen über die Qualität der zu migrierenden Daten und die intuitive Benutzbarkeit des zu erstellenden Systems während der Entwicklung leiden, aber letztlich lässt sich eine gut verstandene Anwendung gemäß eines Prozessmodells entwickeln, das sich an den Grundzügen des Wasserfallmodells orientiert. Es gibt aber auch Anwendungen, die weniger gut verstanden sind. Typische Beispiele sind neuartige Anwendungen (neue Anwendergruppen, neue Geschäftsprozesse/neue Geschäftsmodelle, neue Arten der Verteilung und Mobilität) und solche, die sehr oberflächenintensiv sind (Masken und/oder Reports). In solchen Fällen sorgen unklare, womöglich widersprüchliche Anforderungen dafür, dass initiale Anforderungsanalyse und Spezifikation meist keine abschließende Klärung herbeiführen können. Stattdessen sollten agile Prozesselemente, Timeboxing-Ansätze und stark auf den Einsatz von Prototypen abzielende Modelle zur Anwendung kommen.

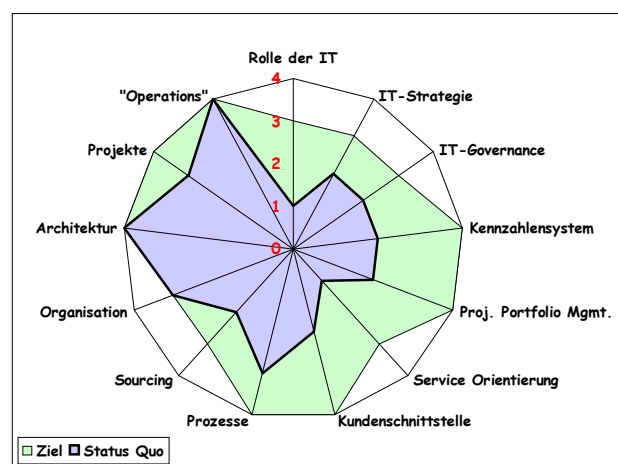
Weitere Prozessmodellvarianten lassen sich herleiten für die Wartung älterer Anwendungen, für die Anpassung besonders risikobehafteter Anwendungsteile und für die Herstellung von Anwendungsteilen, die als releasefähige Kerne dienen sollen.

NFSE besagt nun, dass nicht alle Anwendungsteile (geschweige denn alle Anwendungen) mit dem gleichen Prozessmodell entwickelt werden können. Vielmehr kommt es darauf an, die Bandbreite der Kritikalität und Unklarheiten mit einer Menge von Prozessmodellvarianten abzudecken. Von entscheidender Bedeutung ist es dabei, die Weiche zwischen den Varianten so zu gestalten, dass die Wahl der Varianten objektiviert wird und letztlich nicht den Präferenzen des einzelnen Projektleiters überlassen wird.

## 5 Erfahrungen

Da Business Alignment das Ziel einer möglichst wirkungsoptimierten Ausrichtung der IT an den Belangen des Gesamtunternehmens verfolgt, sind die resultierenden Alignment-Lösungen im Detail zwangsläufig ebenso vielfältig wie die konkreten Ausgangssituationen der betroffenen Unternehmen selbst. Dennoch kann man aus Business-Alignment-Projekten natürlich eine Reihe allgemeingültiger Erfahrungen ableiten, die insbesondere für Projektplanung und Vorgehensweise hilfreich sind.

So empfiehlt sich als Einstieg häufig eine Vorstudie (oder „Business Alignment Assessment“) von nicht mehr als 3 Wochen Aufwand, in die das Business bereits einbezogen wird und so von Anfang an Schwerpunkte und Richtung mitgestalten kann. In dieser Vorstudie sollten alle relevanten Handlungsfelder gemäß des IT-





Handlungsmodells aus Kapitel 2 zumindest grob betrachtet und – am besten getrennt – aus den Blickwinkeln beider Parteien bewertet werden. Aus den Abweichungen zwischen Status quo und angestrebtem Zielzustand, aber auch aus den oft sehr unterschiedlichen Wahrnehmungen von IT und Business können dann die gewünschten Schwerpunkte abgeleitet werden.

Als Handlungsfeld mit dem größten gefühlten Unwohlsein entpuppt sich dabei häufig die Schnittstelle zwischen Business und IT im Rahmen der Projektentwicklung, das mit der umfangreichsten Wirkungsbreite ist sicher die Serviceorientierung.

Zur Etablierung von Serviceorientierung hat sich ein dreistufiges Vorgehen bewährt, bei dem jede Phase konkrete Nutzeneffekte generiert und damit auch jederzeit mögliche Ausstiegsszenarien bietet. In einer **ersten Phase** werden im Dialog zwischen Business und IT zunächst die benötigten IT-Services festgelegt und als „Service Portfolio Version 1.0“ beschrieben. Das Ergebnis stellt die gemeinsame Identifikation des vom Business benötigten IT-Bedarfs dar und ist sowohl arbeitsfähige Kommunikationsgrundlage als auch gültige Handlungsvorgabe zur weiteren Gestaltung und Optimierung der IT, selbst wenn das Vorhaben an dieser Stelle abgebrochen würde. In **Phase zwei** wird das Service-Portfolio weiter verfeinert und durch weitere Detaillierung der Servicekatalog entwickelt. Einen Schwerpunkt bildet dabei die Abstimmung konkreter Leistungsmerkmale und ggf. benötigter Serviceklassen. Ein weiterer Schwerpunkt besteht in der Aggregation der IT-Kosten auf die Service-Ebene zur Gestaltung eines ersten Kostenmodells mit geschäftsrelevantem Aussagewert. In der **dritten Phase** schließlich erfolgen die operative Etablierung des Servicekatalogs und der schrittweise Abschluss konkreter Servicevereinbarungen. Dabei sollte man sich auf die Festlegung wirklich messbarer oder zumindest objektiv nachprüfbarer Leistungsmerkmale konzentrieren und gerade anfangs Verbindlichkeit und Zuverlässigkeit höher bewerten als überambitionierte Zielgrößen. Zur dauerhaften operativen Etablierung bedarf es letztlich immer einer Automatisierung des Servicekatalogs, in dem zumindest die Ebenen „IT-Service-Katalog“ und „IT-Leistungen“ des Service-Modells aus Kapitel 3 abgebildet werden. Inzwischen sind hierfür auch Werkzeuge verfügbar, die teilweise sogar über vordefinierte Service-Referenzmodelle verfügen und damit auch schon während der früheren Projektphasen nützlich sein können.

In jeder Phase sollte man sich immer wieder klar machen, dass nicht alles regelbar ist und vor allem nicht im ersten Durchlauf geregelt werden muss. Die Ergebnisse der Vorstudie und die erste Version des Service-Portfolios aus Phase 1 bilden einen roten Faden, unter dessen Orientierung die beiden folgenden Phasen auch in mehreren Zyklen durchlaufen werden können. Dies eröffnet die Möglichkeit zu einer Art Releasekonzept, nach dem jeweils nur die Bereiche mit besonderem Handlungsdruck oder besonders schnell zu schöpfendem Nutzenpotenzial bis zum operativen Einsatz entwickelt werden.

Im Hinblick auf das No Frills SE bietet es sich an, zunächst zu untersuchen, wie kompliziert die durchgeführten Softwareprozesse sind. Indizien für Verschlangungspotenziale sind umfangreiche Dokumente in den frühen Phasen (womöglich inklusive Vermischung von Abstraktionsebenen), zahlreiche, sehr ähnliche Versionen von Dokumenten, große Aufwände für die Konsistenzsicherstellung zwischen abhängigen Dokumenten, schlecht handhabbare Modelle und mangelhafte Konzentration auf die wirklich wichtigen Anforderungen.

## 6 Schlussfolgerungen

Auch mit Business Alignment und NFSE bleiben Business und IT zwei Partner mit unterschiedlicher Ausgangsposition, aber dem Unternehmenserfolg als gemeinsamem Ziel. Business Alignment gibt jedoch die Mittel in die Hand, mit denen sich die Partner an allen Schnittstellen gleichberechtigt und ohne Sprachbarrieren über den besten Weg zu diesem Ziel verständigen und abstimmen können. Zwei aus unserer Sicht besonders zentrale Aufgabenbereiche haben wir hier ausführlicher dargestellt. Schon eingangs wurde aber darauf hingewiesen, dass Business Alignment auch eine dauerhafte Gestaltungsaufgabe für IT, Geschäftsfunktionen und Unternehmensführung auf der Grundlage eines anhaltenden und expliziten Dialogs darstellt. Insofern ist es kein Zufall, dass in vielen Business Alignment Projekten schon der tatsächliche Beginn dieser oft als Reise empfundenen Maßnahme sowie diese Reise selbst als beinahe ebenso wichtig empfunden wurden wie die dabei erzielten Ergebnisse.

---

**Die Autoren:** Prof. Dr. Volker Gruhn ist Inhaber der Stiftungsprofessur für Angewandte Telematik / e-Business der Universität Leipzig. Jürgen Geisel war bis 2008 CIO und stv. Mitglied des Vorstandes der DEVK-Versicherung. Er arbeitet seit 2009 als selbständiger Berater für IT-Management-Fragen.

### Literatur

- [BAB06] Stefan Biffel, Aybüke Aurum, Barry Boehm, Hakan Erdogmus, Paul Grünbacher: Value-based software engineering, Berlin, Heidelberg: Springer, 2006.
- [Eva06] Eric Evans: Domain-driven design, Boston, Munich: Addison-Wesley, 2006.
- [GPR07] Volker Gruhn, Daniel Pieper, Carsten Röttgers: MDA: Effektives Software-Engineering Mit UML2 und Eclipse, Berlin, Heidelberg: Springer, 2006.
- [GS08] Volker Gruhn, Clemens Schäfer: Runtime Properties of Mobile Systems. In: H. Fujita, I. Zualkernan (Eds.): New Trends in Software Methodologies, Tools and Techniques, Sharjah, UAE: IOS Press, 2008.
- [GS09] Volker Gruhn, Clemens Schäfer: No-Frills Software Engineering for Business Information Systems, Proceedings of the 8th International Conference on Software Methodologies, Tools and Techniques, SoMeT\_09, Prague: IOS Press, 2009 (to appear).
- [Hen92] J. Henderson, N. Venkatraman: Strategic Alignment: A model for organisational transformation through information technology. In T. Kochan, M. Unseem (Eds.), Transforming Organisations, New York: Oxford University Press, 1992.
- [Hin08] Peter Hinssen: Business/IT Fusion - \*How to move beyond Alignment and transform IT in your organization, Belgien: MachMedia, 2008.
- [ITIL1] Office of Government Commerce: Introduction to ITIL, Norwich, UK: Stationery Office Books, 2005.
- [ITIL3] Majid Iqbal, Michael Nieves: Service Strategy: Office of Government Commerce, Norwich, UK: Stationery Office Books, 2007.
- [Koc05] Gottfried Koch: Wertorientierte Steuerung von Versicherungsunternehmen im Lichte des normativen Managements, ZVersWiss(95) 2006, H.1, S.125-148.
- [LR02] Manny M. Lehman, J. F. Ramil: Software Uncertainty, Proceedings of the Soft-Ware 2002: Computing in an Imperfect World, Belfast, Northern Ireland, 2002, p. 477-514.
- [Poh08] Klaus Pohl, Andreas Metzger: Variabilitätsmanagement in Software-Produktlinien. In: Korbinian Herrmann, Bernd Bruegge (Hrsg.), Lecture Notes in Informatics (121), Software Engineering 2008, München: 2008.