

# Der Weg zur richtigen Software-Lösung

## **Aspekte zur Entscheidungsfindung zwischen standardisierten oder individuellen Softwarelösungen.**

**Gerhard Schubert, IGS EDV-Systeme**

Betrachten wir die typische Implementation eines beliebigen Software-Systems, folgt der Ablauf dem - vereinfachten – Phasenmodell:

- a) Funktionale Festlegung
- b) Systemauswahl
- c) Funktionale Realisierung
- d) Test
- e) Einführung
- f) Dauerbetrieb

Diese Phasen gelten unabhängig davon, ob es sich um eine Standard- oder Individuallösung handelt. Unterschiedlich sind lediglich die Anteile an den jeweiligen Phasen.

Die Startphase ‚*Funktionale Festlegung*‘ (a) formuliert die Aufgabenstellung (und deren optimale Lösung) möglichst präzise, oft in Form eines oder mehrerer Pflichtenhefte. Die Beschreibung ist unabhängig vom später gewählten Lösungsmodell.

In Phase ‚*Systemauswahl*‘ (b) wird normalerweise über den Lösungsweg entschieden. Falls dieser nicht von vornherein zugunsten einer Individuallösung fällt, wird mittels einer Marktanalyse nach verwendbaren und bewährten Standard-Lösungen gesucht. Finden sich hier geeignete Kandidaten, werden diese einer genaueren Analyse ihrer Verwendbarkeit auf die geplante Lösung hin unterzogen. Testsysteme werden installiert, geprüft und diskutiert, Kosten werden verglichen, Schulungs- und Supportfragen erörtert und schließlich ein Anbieter ausgewählt. Diese Phase kann, besonders bei Betrachtung mehrerer Systeme, zeit- und kostenintensiv werden.

Im Fall der Individuallösung werden die Anbieter auf Basis der in Phase ‚*Funktionale Festlegung*‘ (a) erstellten Ausarbeitungen verbindliche Angebote abgeben, die verglichen und einer Entscheidung zugeführt werden. Dieser Aufwand ist hier u.U. deutlich geringer als der einer Standard-Lösung.

Die dritte Phase ist die der ‚*funktionalen Realisierung*‘ (c). Dies bedeutet bei einer Individuallösung die Codierung der in Phase ‚*Funktionale Festlegung*‘ (a) festgelegten Funktionen.

Im Fall einer Nicht-idealen Standard-Software bedeutet das, die Funktionen zu realisieren, die zum Erreichen der 100%-igen funktionalen Abdeckung erforderlich sind.

Dies geschieht in der Regel durch Anpassung des Standard-Produktes an die individuellen Forderungen des Kunden seitens des Anbieters bzw. durch ein von ihm beauftragtes weiteres Unternehmen.

Falls die softwaretechnische Realisierung nicht möglich ist, da der Anbieter der Standard-Software nicht willens oder in der Lage ist, sein Produkt individuell abzuwandeln, ist der Kunde wohl oder übel gezwungen, organisatorische bzw. administrative Änderungen seines Workflows vorzunehmen. Bei solchen Systemen fällt in dieser Phase die Entwicklung und Auswertung von Formularen, der Verteilungs- und Sammelmechanismen und die Archivierung an.

Im ungünstigsten Fall (Standard-Software ist in ihren Strukturen nicht auf individuelle Erweiterungen ausgelegt) entsteht in Phase (c) vergleichbar viel oder sogar noch höherer Aufwand als bei einer Individuallösung.

Darüber hinaus sind folgende Vorteile einer Standardlösung durch individuelle Anpassungen gefährdet:

- I) Update-Fähigkeit auf das Standardpaket
- II) Betriebssicherheit des Gesamtsystems

Im idealen Fall einer 100% passenden Standardlösung hat der Kunde keinen weiteren Aufwand.

Die vierte Phase ‚*Test*‘ (d) verläuft in etwa analog zur vorhergehenden. Eine ideal passende Standardlösung generiert hier keinen oder nur minimalen Aufwand, während individuell angepasste Standard-Software und Individuallösungen den Testzyklen unterworfen werden müssen.

Wurden Aktivitäten administrativer bzw. organisatorischer Art zur Erreichung einer 100%-igen Funktionsabdeckung vorgenommen, werden in dieser Phase die Schulungen des beteiligten Personals sinnvollerweise durchgeführt werden.

In Phase ‚*Einführung*‘ (e) trifft die Planung erstmals auf die Realität. Jeder gewählte Lösungsansatz wird hier noch einmal an den realen Bedingungen im Unternehmen verifiziert. Eine Standard-Software muß hier genauso wie eine Individuallösung beweisen, ob sie der ‚*funktionalen Festlegung*‘ in Phase (a) entspricht.

Tiefer liegende Software-Fehler einer unverändert übernommenen Standard-Lösung treten erst in dieser Phase auf, da keine oder nur eine oberflächliche Testphase voranging. Dies gilt auch für organisatorische bzw. administrative Anpassungen zur Erreichung der 100%-igen Funktionalität. Diese werden hier erstmals im Normalbetrieb eingesetzt und können erst hier vollständig verifiziert werden.

Alle anderen Lösungsarten wurden schon vorher in der Phase ‚*Test*‘ (d) auf Fehler geprüft.

Nach Ablauf der vorher festgesetzten Laufzeit und etwaiger Korrekturen in Phase ‚*Einführung*‘ (e), wird die Lösung für Phase ‚*Dauerbetrieb*‘ (e) freigegeben.

Für diese Phase sind weitere Überlegungen anzustellen:

I)

Der Bundesminister für Finanzen definiert Laufzeiten, in denen Anlagegüter, also auch Software-Systeme, betriebswirtschaftlich abgeschrieben werden können. Leider sind diese angegebenen Zeiten nicht immer stabil und unterliegen, je nach Lage der Konjunktur und Einschätzung der Verantwortlichen, gewissen Schwankungen.

Ein angeschafftes System sollte aber mindestens über den jeweils gültigen Abschreibungszeitraum betrieben werden, um dessen Kosten vollständig abrechnen zu können.

II)

Ein Unternehmen muß sich zu jeder Zeit den Anforderungen seines Marktes stellen. Es wird bestehende Produkte und Dienstleistungen den Markterfordernissen anpassen und/oder neue entwickeln.

Diese Anforderungen sind nicht immer planbar und können in kurzen Zeiträumen – sogar mehrfach – entstehen (z.B. Innovationen des Wettbewerbs).

Bietet das eingesetzte Software-System die Flexibilität, um auch geänderte Prozesse abzubilden, entsteht kein weiterer Handlungsbedarf.

Ist dies nicht der Fall, muß die Software, bzw. die administrativen oder organisatorischen Abläufe, an die veränderten Prozesse angepasst und damit ein zusätzlicher Aufwand getrieben werden (der mit weiteren Kosten verbunden ist) um wieder die 100%-ige funktionale Deckung herzustellen.

Solche Zusatzkosten müssen ab dem Entstehungszeitpunkt für die restliche Laufzeit des Systems (mindestens den Abschreibungszeitraum) geleistet werden.

Diese Kosten sind bei Erstellung der TCO-Analyse (total cost of ownership) in Phase ‚funktionale Festlegung‘ (a) oft noch nicht absehbar und werden daher gerne ignoriert, obwohl sie auftreten können und u.U. bemerkenswerten Einfluß auf die Gesamtkostenbilanz der Lösung haben.

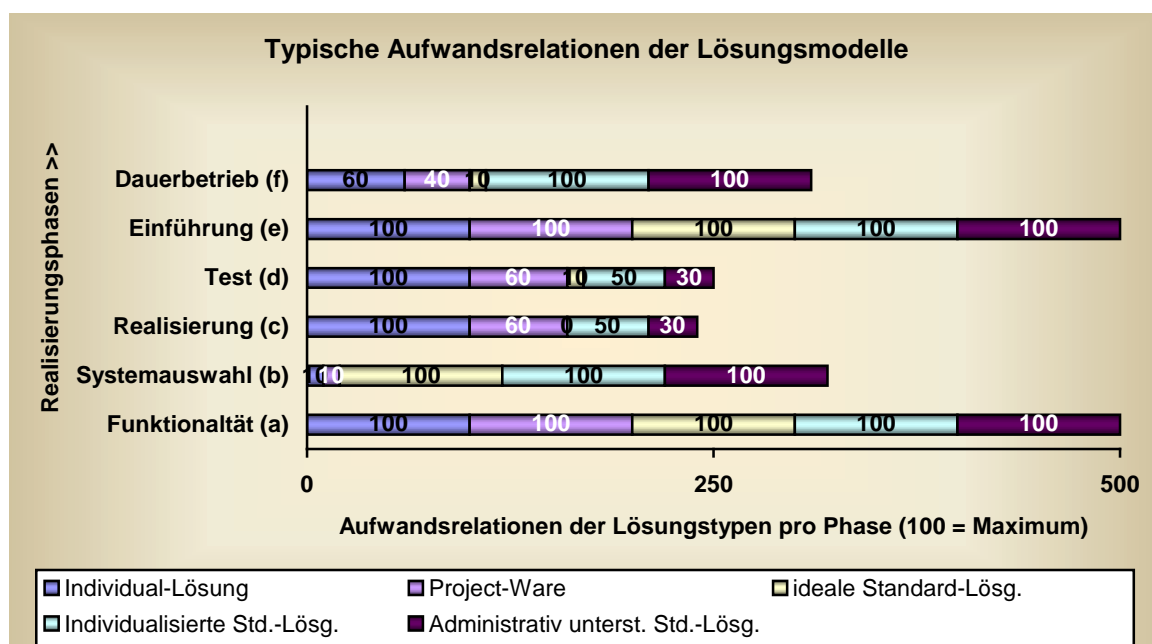
Hier zeigt sich eine der Schwächen von Standard-Software gegenüber einer Individuallösung: Mangelnde Flexibilität.

Bei beträchtlichem Änderungsaufwand über den geplanten Nutzungszeitraum eines Software-Systems können in Phase ‚Dauerbetrieb‘ (f) erhebliche Kostenunterschiede zwischen den Lösungsmodellen entstehen.

Individuallösung sind hier im Vorteil, da sie jederzeit an geänderte Anforderungen passgenau angeglichen werden können – auch mehrfach. Dies ist natürlich auch mit Kosten verbunden, aber die Integrität des Systems und die 100% funktionale Abdeckung kann in jedem Fall wieder hergestellt werden.

Wohlgemerkt: Änderungen im Anforderungsprofil können, müssen aber nicht entstehen. Allerdings wächst aufgrund kürzerer Produktzyklen und schärfer werdenden Konkurrenzsituationen die Wahrscheinlichkeit von Änderungsanforderungen, besonders, wenn über längere Betriebszeiträume betrachtet.

In der folgenden Tabelle sind die typischen Aufwandsrelationen der Lösungsmodelle über die einzelnen Realisierungsphasen dargestellt.



Die Werte auf der X-Achse (Aufwandsrelationen...) geben die typischen relativen Aufwände der einzelnen Lösungstypen für jede Phase bezogen auf 100 (= 100%) an.

Absolut gesehen können 100% des Zeitaufwandes verschiedener Phasen stark voneinander abweichen.

Die Aufwandsrelationen können sich durch besondere Gegebenheiten des jeweiligen Einzelprojektes und durch Unterschiede in der Flexibilität der jeweiligen Standard-Software natürlich gegeneinander verschieben.

Die Aufwände für Auswahl und Beschaffung von Hardware-Komponenten, wie z.B. zusätzliche Server und Workstations, Netzwerk-Komponenten etc., sind nicht Gegenstand dieser Betrachtung und wurden in den Aufwandsrelationen nicht berücksichtigt. Diese müssen im jeweiligen Projekt betrachtet werden.

Kann in Phase ‚Systemauswahl‘ (b) eine Standardlösung gefunden werden, die 100% der Anforderungen über die projektierte Laufzeit abzubilden vermag, so wird diese Lösung jeder anderen vorzuziehen sein. Ist diese Bedingung nicht erfüllt, so müssen die Vor- und Nachteile jedes Lösungsmodells sorgfältig gegeneinander abgewogen werden.

Wie aus der Tabelle hervorgeht, liegen die Kostenverursacher und damit Schwächen von Individuallösungen in den Phasen ‚Codierung‘ (c) und ‚Test‘ (d). Dies ist besonders dann der Fall, wenn eine Individuallösung vollständig neu entwickelt wird.

Ein weiterer, schwer zu quantifizierender Punkt, gesellt sich hinzu:

Der Benutzer wird Einbußen im Bedienungskomfort hinnehmen müssen, da die Codierung einer benutzerfreundlichen und komfortablen Benutzeroberfläche einer Individuallösung Aufwand erfordert und damit Kosten produziert.

Um diesen typischen Nachteilen einer Individuallösung entgegenzuwirken, ist folgender Ansatz denkbar:

Nahezu jede Software auf PC-/Workstation-Basis arbeitet heute unter einer grafischen Benutzeroberfläche, die meisten Pakete mit einer Datenbank-Anbindung.

Funktionen wie Login, Benutzerberechtigung, Menüsteuerung, Buttonverwaltung, Listendrucke usw., sind von den allermeisten Paketen zu leisten.

Daher liegt es nahe, diese Funktionen aus der eigentlichen Individualentwicklung herauszuziehen, so weit als möglich zu standardisieren und separat zu codieren. Diese Funktionalität wird mit dem Aufwand und der Sorgfalt einer Standard-Software entwickelt und ist damit in puncto Betriebssicherheit und Bedienungskomfort mit ihr vergleichbar.

Alle Funktionen, die nichts mit der individuellen Aufgabenstellung zu tun haben, werden separiert und in einem eigenen Programmrahmen realisiert. Analog dazu werden die Teile des Datenbank-Schemas, die nicht individualisiert sein müssen, standardisiert und in einer Basisdatenbank realisiert.

Dadurch wird ein System geschaffen, das strukturell eine Standard-Software ist, in Optik und Bedienungskomfort einem Standard-System entspricht und doch durch Entwicklung spezifischer Masken ein vollständiges Individualsystem mit allen seinen Vorteilen ist.

Durch diesen Ansatz verringert sich der Aufwand für die Phasen ‚Codierung‘ (c) und ‚Test‘ (d) deutlich. Da weniger individueller Code erzeugt wird, ist dieser auch weniger fehleranfällig, als ein von Grund auf individuell entwickeltes System. Diesem Ansatz folgt unsere Lösung ‚Project-Ware‘.

Neben den eher technischen Aspekten der Lösungsauswahl gibt es noch weitere Argumente, die für oder gegen ein bestimmtes Modell sprechen. Ein sehr wichtiges ist: Ist der Systemanbieter mindestens über die geplante Laufzeit des Software-Systems existent, um Support zu leisten und ggf. Änderungen am System vorzunehmen? Die Größe eines Unternehmens war früher ein wichtiges Kriterium, um dessen Überlebensfähigkeit am Markt einzuschätzen. In der augenblicklichen wirtschaftlichen Lage fallen allerdings Unternehmen jeglicher Größenordnung der ‚Insolvenzwelle‘ zum Opfer. Große Anbieter sind häufig in ‚business units‘ segmentiert. Jede unit ist für ihren Erfolg verantwortlich und hat ‚schwarze‘ Zahlen zu liefern. Ist dies nicht der Fall, kann sie aufgelöst oder veräußert werden. Durch diese Prozesse ist das Kriterium ‚Größe‘ nicht mehr alleine ausschlaggebend.

Ein weiterer Einflussfaktor ist die Konzentration von Unternehmen. Mittlerweile werden auch große Unternehmen aufgekauft. Damit ist die Geschäfts- und Produktpolitik den Intentionen des neuen Eigentümers unterworfen. In den seltensten Fällen bleiben diese ohne Einfluß auf das gekaufte Unternehmen. Die Folgen für die Bestandskunden sind dann nicht mehr kalkulierbar. Software-Systeme können zugunsten den Systemen des Erwerbers sogar eingestellt werden, ohne dass der Kunde einen Einfluß hierauf hat. Von daher kann von der automatischen Präferenz des großen Anbieters über den kleinen nicht mehr zwingend ausgegangen werden. Die Zeitdauer und Konstanz der Marktpräsenz eines kleineren Anbieters (und damit der Erfahrung) sollte stärker in der Bewertung der Anbieterreignung berücksichtigt werden.

Große Anbieter tendieren zum Anbieten von Standard-Systemen, während kleinere Anbieter flexibel genug sind, um ihre Lösungen zu individualisieren.

Und schließlich ist die Motivation kleinerer Unternehmen (die meistens durch die Inhaber geführt werden), eine für den Kunden zufriedenstellende Lösung zu realisieren, in der Regel höher als die großer Unternehmen.