

Business Software

Das Thema Software stellt Unternehmen immer wieder vor neue Herausforderungen: Soll eine bestehende Softwarelösung her oder lässt man sich eine spezifische Lösung schneiden? Wie kann man die Sicherheit der IT-Infrastruktur gewährleisten? Und wie lassen sich die Ressourcen im Unternehmen optimal einsetzen?



Entscheidend für den Projekterfolg ist, dass Kosten, Zeit und Funktionsumfang unter Kontrolle bleiben.

Massgeschneiderte Business Software als Ingenieursaufgabe

Bei der Einführung einer Business Software stellt sich dem beschaffenden Unternehmen häufig die Frage, ob ein bestehendes System eingekauft und angepasst oder ob eine massgeschneiderte Lösung realisiert werden soll. Wann lohnt sich die Realisierung einer massgeschneiderten Lösung? Und was muss dabei beachtet werden?

Auf den Punkt gebracht: Die Entwicklung einer massgeschneiderten Business Software lohnt sich genau dann, wenn diese Software Kernprozesse betrifft, in denen sich das Unternehmen massgeblich von seinen Mitbewerbern unterscheidet. Einige konkrete Beispiele aus den letzten zehn Jahren:

- Für den Detailhandel wurden eine Personaleinsatzplanung und eine Filiallogistik entwickelt. Die Planung des Personals und die Feinlogistik in den Verkaufsstellen sind kritische Kernprozesse jedes Einzelhandelsunternehmens.

Selbst kleine Optimierungen haben aufgrund des Multiplikationseffekts einen signifikanten Einfluss auf das Ergebnis.

- Für die Telecom-Industrie wurde eine Service-Factory realisiert. Diese Factory ermöglicht es dem Telecom-Unternehmen, seinen Geschäftskunden stark individualisierte Leistungen anzubieten, die aber genau so effizient produziert werden können wie standardisierte Leistungen. Die Lösung wurde dazu tief in die Prozesse des Unternehmens integriert.

- Für ein kantonales Strasseninspektorat wurde die Software für die Planung und Überwachung des Winterdienstes von Grund auf entwickelt. Die Software ermöglicht einen effizienten Einsatz von Geräten und Streumitteln und erhöht die Sicherheit auf den Strassen.

Diesen Beispielen ist gemeinsam, dass sie Kernprozesse betreffen, mit denen sich die Organisationen differenzieren. Die Differenzierung ist typischerweise in der Tiefe der Abdeckung und dem Auto-

matisierungsgrad eben dieser Kernprozesse begründet oder auch in der Einzigartigkeit des Geschäftsmodells.

Anders gestaltet sich die Situation in Bereichen, die ausserhalb des Kerngeschäfts liegen, oder in denen keine ausgeprägte Individualisierung erforderlich ist. Beispiele sind die Logistik in einem Nicht-Handelsunternehmen oder die Lohnbuchhaltung. In der Praxis ist die Situation nicht immer eindeutig. Für das beschaffende Unternehmen geht es bei einer solchen Make-or-Buy-Entscheidung um die Beurteilung: Können wir mit dem Standard leben oder nicht? Natürlich ist jedes Unternehmen anders. Die entscheidende Frage ist, ob ein Unternehmen genügend anders oder ob ein Prozessbereich zentral genug ist, um die Entwicklung einer massgeschneiderten Lösung zu rechtfertigen.

Am Entscheid festhalten

Wichtig ist, dass man sich an den einmal getroffenen Entscheid hält. Im Falle einer eingekauften Lösung bedeutet dies zu akzeptieren, dass die Anpassungsfähigkeit Grenzen hat und sich das Unternehmen in gewissem Mass der Software anpassen muss. Im Falle einer massgeschneiderten Lösung bedeutet es, dass sich die Lösung in einem festgelegten Funktionsumfang bewegt und man nicht «alles» bekommen kann. Die Realisierung individueller Business-Lösungen stellt eine anspruchsvolle Aufgabe dar, wie die folgenden Herausforderungen zeigen.

Funktionsumfang im Griff behalten

Die Erstellung einer massgeschneiderten Business Software ist ein Projekt.

Entscheidend für den Erfolg ist, dass Kosten, Zeit und Funktionsumfang unter Kontrolle bleiben. Besonders wichtig ist, den Rahmen, der von der Individuallösung abgedeckt wird, bereits in der Vorbereitung klar zu bestimmen. Die Versuchung ist gross, den Umfang im Verlauf des Projekts auszudehnen. Der Appetit kommt sprichwörtlich mit dem Essen, sobald die Freiheiten einer individuellen Lösung klar werden. Es ist die vermeintliche Chance, «alles» zu bekommen, die lockt. Geben die Verantwortlichen in diesem Moment nach, steigen die Projektrisiken stark. Gegenmassnahmen sind ein klar definierter Projektauftrag, eine starke Projektleitung, die für diesen Auftrag einstehen kann, sowie im Voraus festgelegte Um Systeme und Schnittstellen. Die Zeit, die dafür in der Vorbereitungsphase aufgewendet wird, ist gut investiert.

Ein sinnvolles Vorgehen besteht häufig darin, von Beginn an ein weiteres Release einzuplanen. Dies nimmt Druck aus dem ersten Release – und das zweite Release kann mit den Erkenntnissen des ersten angegangen werden. Weiter nimmt es auch Druck vom Kunden, alles hier und jetzt entscheiden zu müssen.

Flexibilität durch domänenspezifische Sprachen

Die Lösung zu entwickeln ist eine Sache, sie nach erfolgter Einführung zu betreiben und unterhalten eine andere. Der grösste Teil der Kosten einer Business Software fällt erst während der Betriebszeit an. Beim Erwerb eines Produkts profitiert der Käufer gegen einen fixen Wartungsbetrag von den Anpassungen, die der Hersteller im Rahmen der Produktpflege vornimmt. Insbesondere die Anpassungen an regulatorische Veränderungen und allgemeine Marktveränderungen sind damit typischerweise abgedeckt. Dies gilt allerdings nur so lange, wie man sich durch eigene Anpassungen (Customizing) nicht allzu weit vom Standard des Herstellers wegbewegt hat.

Wird eine eingekaufte Software-Lösung im Anpassungsprozess stark verändert, weil man eben doch nicht mit dem Standard leben kann oder will, erhöhen sich der Aufwand und das Risiko für ein späteres Upgrade erheblich.

Bei einer Individualsoftware trägt der Käufer die Gesamtkosten für Anpassungen und Erweiterungen. Damit sich diese Kosten in einer wirtschaftlich sinnvollen Grössenordnung bewegen, bietet sich in vielen Fällen die Einführung einer domänenspezifischen Sprache an. Eine solche Sprache ermöglicht es, die sich verändernden Anforderungen des Kunden business-nah auszudrücken. Der Softwareingenieur betrachtet die heute konkreten Anforderungen des Kunden als Spezialfall allgemeiner Anforderungen, die sich mit Blick auf die Zukunft abzeichnen.

Konkrete Anwendungen solcher Sprachen sind beispielsweise Regeln zur Bewertung von Arbeitszeit oder ein Modell für einen Servicekatalog inklusive zugehöriger Prozesse. Der Begriff «Sprache» ist weitläufig zu verstehen und reicht von universellen Regeln und Prozessbeschreibungen über eine Menge von Parametertabellen bis hin zu XML-Dokumenten mit passenden Werkzeugen für die Generierung von Code. Gemeinsam ist allen diesen Ansätzen, dass sie die Softwareentwicklung auf eine höhere Ordnung und damit näher ans Business bringen. Dies vermeidet, dass schwer wartbarer Code entsteht, der starken Veränderungen ausgesetzt ist. Die langfristigen Kosten sinken. Die Herleitung geeigneter domänenspezifischer Sprachen ist nicht einfach. Es ist selbst für qualifizierte Software-Ingenieure eine Herausforderung, sich so weit in das Geschäft des Kunden einzudenken, dass geeignete Modellierungskonstrukte evident werden. An dieser Stelle ist gutes Software-Design gefragt. Der französische Schriftsteller Antoine de Saint-Exupéry hat einmal gesagt: «Ein Text ist nicht dann vollkommen, wenn



Eine domänenspezifische Sprache bringt die Softwareentwicklung auf eine höhere Ordnung und damit näher ans Business.

man nichts mehr hinzufügen, sondern nichts mehr weglassen kann.» Das gilt ebenso für gute Software. Die Modellierungssprache ist minimal und zugleich ausreichend mächtig für die bestehenden und kommenden Anforderungen des Kunden.

Design Matters

Wir leben in einem Zeitalter der Konsumerisierung der IT-Industrie. Dank Geräten wie Smartphones und Cloud-Services wie Dropbox verfügen heute viele Arbeitnehmer privat über eine bessere IT-Infrastruktur als jene, die ihnen am Arbeitsplatz geboten wird. Es genügt deshalb für massgeschneiderte Business Software nicht, die funktionalen Anforderungen des Auftraggebers abzudecken. Das Ziel muss zusätzlich sein, die Endbenutzer zu begeistern. Die Benutzeroberfläche spielt dabei eine wichtige Rolle. Es ist deshalb sinnvoll, die Benutzeroberfläche mit der Unterstützung von Design- und

Usability-Experten zu gestalten. Eine ansprechende Gestaltung und hohe Bedienbarkeit sind wesentliche Erfolgsfaktoren für eine Business Software. Der Schlüssel zu einer guten Benutzeroberfläche ist häufig die Einfachheit: weniger ist mehr. Es ist im Allgemeinen sinnvoller, wenige Funktionen zu unterstützen, die dafür flexibel eingesetzt werden können, anstatt deren viele, die nur sehr spezifisch anwendbar sind. «Simple things should be simple, complex things should be possible», um es mit den Worten des amerikanischen Informatikers Alan Kay zu sagen. Auch an dieser Stelle ist gutes Software-Design gefragt, um gemeinsam mit dem Kunden die minimale Lösung herzuleiten, die heute und in der absehbaren Zukunft ausreichend ist. ☛

André Naef, Informatik-Ingenieur und Business Development Manager Ergon Informatik AG