

Umbau oder Neubau?

Bevor ein Software-Gebilde abgerissen und neu aufgebaut wird, sollte jeweils die Tragfähigkeit der vorhandenen Lösung beurteilt werden. Oft ist eine sanfte Renovation sinnvoller. Hier den richtigen Entscheid zu treffen, ist allerdings nicht trivial.

→ VON PETER GASSMANN & BEAT FLURI



Applikationen sind immer ein Kind ihrer Zeit. Dies zeigt sich bei diversen Aspekten von der Performance und der Verfügbarkeit bis zu den unterstützten Endgeräten. Die Strategien und Geschäftsabläufe der Unternehmen ändern sich jedoch in immer kürzeren Zeitabständen. Mit der Einführung neuer Technologien und der zunehmenden Computerisierung des Privaten verändern sich zudem die Erwartungen der Benutzer bezüglich User Experience. Auch vonseiten des Betriebs können neue Anforderungen kommen,

und kontinuierlich anzupassen. Applikationsverantwortliche ihrerseits müssen immer wieder Entscheide bezüglich der Umsetzung von Änderungsanfragen herbeiführen. Doch auf welcher Basis können solche Entscheidungen überhaupt gefällt werden?

ENTSCHEIDUNGSKRITERIEN FESTMACHEN Beschlüsse werden letztlich auf der Basis von Nutzen und Kosten bzw. Chancen und Risiken getroffen. Bei den Kosten muss primär das Verhältnis zum Nutzen betrachtet werden. Sie

sind abhängig von der Komplexität der Anforderung und vom aktuellen Zustand des Systems. Die Applikationsarchitektur kann durch eine unflexible Struktur gewisse Veränderungen verunmöglichen. Die Ausgaben für

den Betrieb einer Lösung, aber auch die Opportunitätskosten – wenn zum Beispiel wegen einer schlechten Benutzeroberfläche Kunden abwandern – sind oft schwieriger festzustellen und werden daher selten berücksichtigt.

Bezüglich der Risiken sind ebenfalls verschiedene Aspekte zu betrachten. Ein tiefer Eingriff in ein laufendes System ist riskant, vor allem, wenn eine zuverlässige Teststrategie und -infrastruktur fehlen. Es kann aber auch riskant sein, nichts zu tun, etwa vom jeweiligen Hersteller empfohlene Upgrades nicht durchzuführen.

Der Zustand der Applikation beeinflusst somit sowohl die Kosten als auch die Risiken: Eine gut unterhaltene Software kann einfacher und schneller angepasst werden.

WIE SETZT SOFTWARE ROST AN?

Wertschöpfende Software-Entwicklung ist mehr als die einmalige Produktion funktionierender Codes. Software braucht angemessene Pflege. Im Idealfall wird sie gesund in den produktiven Betrieb aufgenommen. Zu diesem Zeitpunkt verwendet sie aktuelle Technologien und passt in die IT-Landschaft des Unternehmens. Danach sollten regelmässig Wartungsarbeiten durchgeführt werden, um Technologien zu erneuern und die Software an Änderungen der IT-Landschaft oder an neue Anforderungen anzupassen. Dabei ist auch die Modularisierung der Software immer wieder neu zu beurteilen. Denn

bei intelligenter Modularisierung können z. B. unterschiedliche Release-Zyklen für die einzelnen Komponenten die rasche Umsetzung vereinfachen. Wegen fehlender Zeit, ändernder Priorisierung oder mangelndem Budget werden die Wartungsarbeiten jedoch häufig vernachlässigt. Damit setzt die Software Rost an.

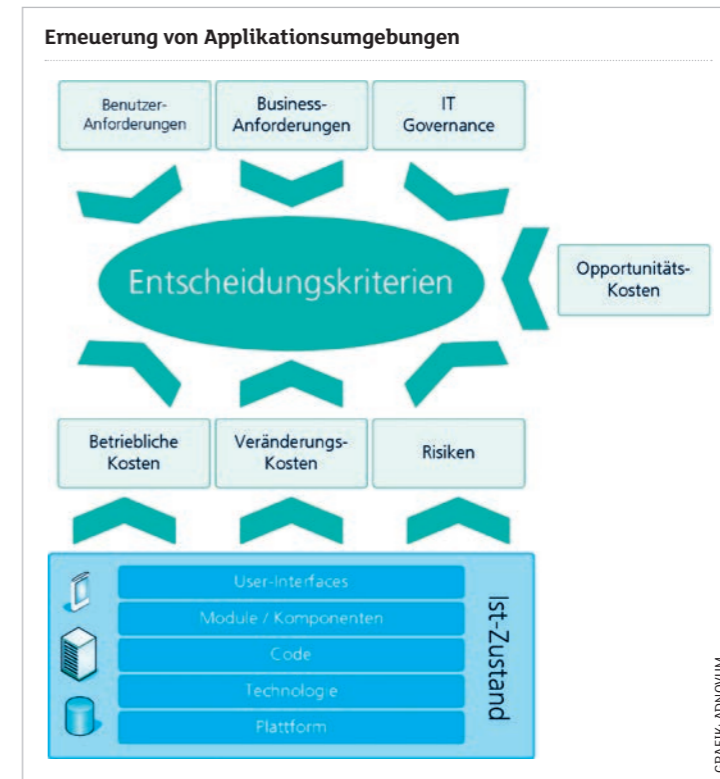
SYMPTOME FÜR EVOLUTIONSBEDARF

Mangelnde Pflege wirkt sich hauptsächlich im Betrieb und im Business aus. Die eingerosteten Lösungen können Upgrade- oder Migrationsvorhaben blockieren, Teilauslagerungen von Komponenten für Outsourcing von Prozessschritten verunmöglichen oder Know-how und Ressourcen für alte Hardware und Betriebssysteme binden. Dies führt zu hohen Betriebskosten, die durch eine stetige Modernisierung der Technologie vermieden werden könnten.

Auch aus geschäftlicher Sicht sind die möglichen Auswirkungen substanzial: massive Herausforderungen bei einem anstehenden Wechsel zu einer integrierten Kunden- oder Produktsicht, erhöhte Time to Market, hohe Kosten für die Realisierung neuer funktionaler oder regulatorischer Anforderungen, zunehmende Fehlerhäufigkeit sowie ein für die Kunden nicht nachvollziehbarer erhöhter Zeitbedarf, bis ein Fehler analysiert, behoben und der Fix in die Produktion eingespielt ist. Solche Probleme können dazu führen, dass Kunden abwandern und die Wettbewerbsfähigkeit beeinträchtigt ist. Die Pflege der Software ist somit aus geschäftlicher wie betrieblicher Sicht vital.

VORGEHEN ZUR ENTSCHEIDUNGSFINDUNG

Für die Verantwortlichen geht es darum, in diesem Setting den Überblick zu behalten und die Weiterentwicklung ihrer Applikationen gezielt voranzutreiben. IT-Consulting kann dazu einen zentralen Beitrag leisten. Besonders wirksam ist Beratung in der Konzeptionsphase oder wenn es um die Planung und Bewertung der Evolution einer Applikation geht. In einer Architekturanalyse kann die Problemstellung im Kontext der Gesamtarchitektur aus der Perspektive verschiedener Stakeholder beleuchtet und eine sinnvolle Zielarchitektur definiert werden.



Evolution oder Revolution? Bei den Entscheidungskriterien fallen die Opportunitätskosten oft unter den Tisch, da sie kaum messbar sind

werden. Mit diesem Vorgehen lässt sich sicherstellen, dass Stakeholder wie Business und Betrieb frühzeitig involviert werden und massgeblich zur Lösungsfindung beitragen können.

Um eine Zielarchitektur vorzuschlagen, braucht es neben der rein geschäftlichen und betrieblichen Betrachtung auch eine technologische Beurteilung der Software auf verschiedenen Ebenen:

■ **Plattform:** Welche Plattform wird eingesetzt, um die Software zu betreiben?

■ **Technologie:** Kann die Software mit dem verwendeten Technologie-Stack mit vernünftigen Kosten weiterentwickelt werden oder ist der Stack derart veraltet, dass das Know-how fehlt und moderne User-Interaction-Anforderungen nicht umgesetzt werden können?

■ **Code:** Falls der Technologie-Stack nicht ersetzt werden muss, wie ist die Code-Qualität der Software? Wie hoch ist ihre Komplexität? Werden gängige Designmuster und Codier-Richtlinien eingesetzt? Wie steil ist die Lernkurve

für neue Entwickler? Wie hoch ist die Testabdeckung und welche Qualitätsmetriken werden bei der Entwicklung angestrebt?

■ **Modul/Komponenten:** Welche Modul-

ularisierung wurde gewählt? Können mit der Modularisierung unterschiedliche Release-Zyklen für Komponenten unterstützt werden? Ist ein unterschiedliches Deployment der Komponenten möglich?

■ **User-Interfaces:** Welche Kanäle und Benutzerschnittstellen können unterstützt werden? Wie komplex ist die Anpassung für Responsive Design?

Für Business, Benutzer und Betrieb resultiert so je eine Ist- und Sollarchitektursicht mit den entsprechenden Gaps und Prioritäten. Diese Informationen dienen als Basis für die Abschätzung von Kosten und Risiken. Aus den verschiedenen Architektursichten lässt sich eine Roadmap für die Applikationsumgebungen des Unternehmens ableiten, die als Basis für Entscheide über die Weiterentwicklung oder den Ersatz von Komponenten dient.

FAZIT: REVOLUTIONEN VERMEIDEN

Um die Zukunftsaussichten von geschäftskritischen Software-Lösungen zu bewerten, ist sowohl die technische als auch die betriebswirtschaftliche Perspektive zu berücksichtigen. Ein Architekturassessment schafft die nötige Grundlage. So lässt sich entscheiden, ob die Software revolutioniert werden muss oder ob einzelne Evolutionsschritte genügen.

Generell empfiehlt es sich, von Anfang an für eine nachhaltige und wertschöpfende Software-Entwicklung zu sorgen, um die sich verändernden Anforderungen möglichst lang mit vernünftigem Aufwand erfüllen zu können. So kann eine Revolution der Applikationsumgebung längerfristig vermieden werden. ←

Peter Gassmann ist CCO und Head of IT-Consulting bei AdNovum; Dr. Beat Fluri ist Senior Solution Architect bei AdNovum → www.adnovum.ch



«Die technische und die betriebswirtschaftliche Sicht sind gleichermaßen zu berücksichtigen»

Peter Gassmann

zum Beispiel, wenn eine Betriebsplattform aktualisiert oder ausgetauscht werden soll.

In diesem dynamischen Umfeld haben CIOs die komplexe Aufgabe, geschäftskritische Systeme und Applikationen am Laufen zu halten