



## Sabotage für Könner

Mitarbeiter) aus dem Weg räumen können? Oder anders gefragt, wie schaffen wir es trotzdem immer wieder, Projekte in aller Regelmäßigkeit zu versenken?

Keinesfalls sollten wir jetzt „den Sand in den Kopf stecken“ [Lothar Matthäus]. Um dem Geheimnis unserer Misserfolge auf die Spur zu kommen, ist eine Analyse real existierender Projekte und Strategien äußerst nützlich. Was schon auf den ersten Blick erstaunt, ist die Beharrlichkeit von Experten, konsequent eigene Fehler oder die anderer zu wiederholen. Schon Douglas Adams wusste anzumerken: „Menschliche Wesen, deren Einzigartigkeit darin besteht, von anderen lernen zu können, sind ebenso bemerkenswert in ihrer offensichtlichen Abneigung, dies zu tun“. Möchten Sie also gezielt ein Projekt boykottieren, sollten Sie sich das Zitat schon mal unauffällig hinter die Ohren schreiben.

Äußerst hilfreich ist in diesem Zusammenhang, den verwendeten Technologien blind zu vertrauen. Man nehme also eine (ordentliche) Prise BPM, SOA, Cloud Computing, dazu Java EE 6, einen Enterprise Service Bus sowie möglichst viele andere Zutaten aus dem allgegenwärtigen Schlagwort-Bingo und verrühre alles, äußerst agil und gleichmäßig, zu einer einheitlichen Masse. Als geeignete Mixer für alle Lebenslagen fungieren dabei Eclipse, IntelliJ oder Netbeans. Das resultierende Konglomerat glänzt nicht nur auf Vertriebsbrochüren, sondern verschafft Entwicklern eine exzellente Lebensgrundlage aufgrund des, gefühlt lebenslänglichen, Refactoring-Potenzials.

Dass „Softwarearchitektur allenfalls eine kurzfristige Dienstleistung“, gemessen am Lebenszyklus einer Anwendung, darstellt, zeugt von praktischer Erfahrung aus berufenem Munde – in diesem Fall dem eines hochrangigen Managers. Zwar führt kein Instrument effektiver zu einem stabilen Rahmen, um Anforderungen adäquat zu berücksichtigen, als Architektur. Allerdings entstehen dadurch angeblich „erhebliche“ Kosten, denen als Ergebnis fragwürdige Diagramme aus reichlich verzierten Kästchen und Linien gegenüber stehen. Für den echten Kreativitätsfanatiker unter den Softwareentwicklern ist ein starres Gerüst ohnehin eher Schreckgespenst als Hilfe. Warum sich also um unnötigen Ballast scheren?

Selbst an paradiesischen Orten, die Zachmans Jünger oder andere Schatzenboxer unter ihre Fittiche genommen haben, präsentieren sich bisweilen Geschäftsarchitekturen wie komplexe Schnittmuster. Ein altbewährtes Anti-Pattern besteht aus den Prinzipien „Data Hiding“ und „Separation of Concerns“. Dabei gewähren Teilarchitekten ihren Kollegen keinerlei Einsichten in die eigenen Funktionalitäten, weil sich jeder ausschließlich um die eigenen Angelegenheiten kümmern sollte. Wer jetzt noch Dezentralismus mangels eines Chefarchitekten hinzufügt, kann sich zurücklehnen und genüsslich des kommenden Ungemachs harren. Immerhin sind Querschnittsbelange die treibende Kraft hinter Unternehmensarchitekturen und lassen sich so gezielt vernachlässigen. Besonders eindrucksvoll erscheint ein Projektergebnis übrigens, wenn es nicht unter der schier grenzenlosen Vielzahl von Lösungsalternativen selektiert, sondern sich immer für mehrere Optionen entscheidet. Ein probates Mittel, um dem Hass auf Servicetechniker zu frönen.

Gegner brachialer Gewalt können sich auch subtilerer Mittel bedienen. Wer gibt die Anwendung in Auftrag? Und wer muss sie hinterher ausbaden, sprich nutzen? Falls sich die Antworten unterscheiden, lassen sich sämtliche Folterinstrumente undurchsichtiger Geschäftsprozesse einsetzen, speziell, wenn gleichzeitig komplizierte Formulare im Spiel sind. Fehlende Usability fördert speziell die unerschrockenen Abenteurer unter den Anwendern. Aber das Thema hatten wir schon im letzten Editorial.

Ich bin mir sicher, Sie könnten uns aus Ihrem Erfahrungsschatz noch weitere Anregungen nennen, um gezielte und raffinierte Fallstricke bei der Entwicklung von Geschäftsanwendungen einzuflechten. Falls dem so ist, lassen Sie uns doch bitte an Ihren diabolischen Ideen teilhaben.

In diesem Sinn viel Spaß mit der vorliegenden Ausgabe

Ihr Michael Stal

► Früher haben IT-Abteilungen die meisten Geschäftswendungen „ad hoc“, also aufgrund sachdienlicher Hinweise und eigener Mutmaßungen, entwickelt. Immerhin lässt sich ohne ausreichende Kenntnis der Anwendungsdomäne gänzlich unvorbelastet entwickeln. Wie wir wissen, waren die Resultate nicht immer kompatibel mit den Erwartungen der Anwender. Aber bekanntlich können Informatiker Leidenschaft und Sturheit, ja geradezu Autismus an den Tag legen.

Natürlich ist heute alles besser – wie könnte es auch anders sein in Zeiten von Clouds, SOA und BPM. Während die Softwareentwickler der IT-Frühgeschichte nur in eng abgesteckten Sandkästen spielen durften, kann sich ihr Einsatzgebiet in unserer postmodernen Ära nahezu unbegrenzt und ungebremst entfalten – der vollständigen IT-Durchdringung sei Dank. Alles ist mit allem vernetzt; nur auf diese Weise lassen sich integrierte Geschäftsabläufe ermöglichen. Leider schaukeln sich dadurch auch Fehler mittels Dominoeffekten bisweilen zum Ausmaß eines Tsunamis hoch, dem überwiegend unschuldige Anwender zum Opfer fallen. Aber wer interessiert sich schon für das Ende der Nahrungskette.

Die eigentliche Frage muss daher lauten: Warum schlagen so viele Projekte in der Unternehmens-IT fehl, verschlingen Unsummen an Budgets oder verzögern sich erheblich, wenn doch heute eine derartige Vielzahl an Technologien, Methoden, Experten und Werkzeugen existiert? Haben wir nicht geradezu ein gewaltiges Arsenal von „Silver Bullets“ zur Verfügung, mit denen wir alle Produktivitätshemmnisse (synonym für: