

Simplifikation im Softwarebau

Liebe Leserinnen und Leser,

bekanntlich soll die Modellierung den Softwarebau vereinfachen, dies erkaufen wir uns durch eine stärkere Formalisierung, die Sprachen und Architekturen zu vereinheitlichen ja sogar zu standardisieren.

Auf der anderen Seite wird die IT-Welt immer komplizierter und generiert fortlaufend neue Frameworks, Sprachen und Begriffe wie z.B. Rails, LINQ, AOP, SOA und MDA als jüngere Beispiele.¹

Was läuft hier am Markt?

Ganz einfach, der Markt oder die Gemeinde produziert neue Bedürfnisse und Techniken, die im Wettbewerb meistens zu einem abnehmenden Nutzungsgrad stehen. Das Management ist überfordert und die Entwickler verunsichert. Die Lösung sollte man doch eher in der Simplifikation von den Anforderungen über die Algorithmen bis hin zur Architektur sehen und nicht ständig neue Hypes generieren. Ein erster Schritt dazu ist sicher die Visualisierung. Eine einfache Darstellung hilft, Anwendungen zu gliedern, Komplexität zu reduzieren und so robuste Systemarchitekturen zu bauen. Im Einsteinjahr sei hier ein Zitat angebracht "Das Problem zu erkennen ist wichtiger, als die Lösung zu erkennen, denn die genaue Darstellung des Problems führt zur Lösung".

Simplifikation heisst auch Sicherheit. Eine durchdachte und verständliche Ergonomie produziert weniger Fehler beim Anwender. Muss ein Passwort (Passwortsatz) nun einfach oder kompliziert sein? Der Idealfall besteht natürlich darin, überhaupt nirgends Paßwörter niederzuschreiben; mit einfachen Eselsbrücken läßt sich so etwas auch bei einem schlechten Gedächtnis verwirklichen. Sie sehen, Simplifikation kann Schwieriges verbergen.

Wir stehen immer noch vor der Frage, wie Prozesse der Simplifikation zu erklären sind. Und welche technischen Strukturen sind in besonderer Weise "anfällig" für Vereinfachungen? Um einer Beantwortung dieser Fragen näher zu kommen, sind die Anforderungen der Schlüssel. Die Simplifikation, also eine Verschlinkung und Vereinfachung der IT-Landschaft, sowie die Konsolidierung der Daten und Funktionen auf einer einheitlichen Plattform stehen seit Jahren im Interesse vieler Firmen, doch fehlt dann der Mut die Prozesse auch wirklich zu vereinfachen. Anforderungen oder Designfragen lassen sich heutzutage mit Standards wie Patterns, Profile oder Referenzen einfacher lösen sofern man vom "not invented here" Syndrom wekommt.

Programmieren ist eine konstruktive Kunst und diese besteht auch darin, mittels Refactoring oder einem Redesign den Code zu verbessern. Simplifikation kostet, das ist klar. Denn ein Ausdruck wie (if a not <> b) mit (if a = b) zu vereinfachen, bedingt eine gehörige Portion Hirnschmalz.

Auch eine Bibliothek kann schlank daherkommen. Jede .net-Sprache hat eine Sprachbindungsbibliothek. Die VB.dll hat eine Größe von 300 KB. C#'s Bibliothek ist zwischen 3 und 20 MB groß, sofern man den Strich zwischen mscorlib.dll und System.dll zieht oder alles vom .net-Framework zu C#'s benötigter Minimalinstallation zählt. Delphis Borland.Delphi.dll ist weniger als 64 KB groß.

Um noch einmal Einstein zu zitieren: "Alles sollte so einfach wie möglich gemacht sein, aber nicht einfacher". Investieren Sie also künftig mehr in die Techniken der Vereinfachung. UML, Patterns oder Refactoring leisten mit zugehörigen Tools einen Beitrag dazu. Zu Empfehlen sei auch das Buch über The School of Niklaus Wirth - "The Art of Simplicity".

In diesem Sinne wünsche ich Mut zur Vereinfachung und Spass beim Lesen.

Max Kleiner, November 2005

¹ (Wobei ich AOP und MDA eher zum Entwicklungsprozess zähle, aus dem mit SOA dann eine serviceorientierte Zielplattform entsteht! Tja, man kann auch komplizierte Begriffe einfach erklären;)).