

## Von der Anwendungsentwicklung zur Anwendungsmontage

Autor: Dieter E. Jenz

Moderne Komponententechnologien eröffnen den Weg zu komponentenbasierten Anwendungssystemen. Damit einhergehend kann der Schwerpunkt von der Anwendungsentwicklung zur Anwendungskonfiguration verschoben werden. Anwendungen werden aus vorgefertigten Komponenten zusammengesetzt, wobei die einzelnen Komponenten bei Bedarf über Adapter miteinander verbunden werden. Adapter sind kleine Code-Fragmente, die mit einer Skript-Sprache erstellt werden und auch als "Komponenten-Klebstoff" bezeichnet werden können.

Der Nutzungsbereich von Komponenten muß so breit wie möglich sein. Dem Komponentenentwickler eröffnet sich dadurch ein höheres Absatzpotential, dem Komponentennutzer hingegen die Möglichkeit, Komponenten mehrfach in unterschiedlichen Einsatzfeldern zu nutzen. Die Konfigurierbarkeit von Komponenten über extern einstellbare Parameter spielt somit eine wichtige Rolle.

Durch eine ausgeprägte Parametrisierbarkeit von Komponenten rückt wiederum ein Fernziel etwas näher: die Assemblierung von Anwendungen durch Mitarbeiter von Fachabteilungen. Komponenten werden in diesem Szenario durch Anpassen von Parametern konfiguriert und mit anderen Komponenten zu ausführbaren Anwendungen "zusammengestöpselt". Woher die Komponenten stammen, ob diese selbst entwickelt oder von Software-Herstellern lizenziert wurden, spielt keine Rolle.

An dieser Stelle stellt sich die Frage, in welchem Umfang diese Ziele heute bereits erreichbar sind und welche wesentlichen Hindernisse noch entgegenstehen. An einigen Punkten soll der aktuelle Stand aufgezeigt werden.

### Technologieabhängigkeit

Derzeit dominieren drei unterschiedliche Komponentenmodelle: COM/ActiveX (Microsoft), JavaBeans und Enterprise JavaBeans (Sun Microsystems) und CORBA Components (Object Management Group). Die genannten Modelle unterscheiden sich hinsichtlich ihrer Semantik. Dadurch bedingt entsteht eine technologische Abhängigkeit erstellter Komponenten, die über Technologiebrücken mühsam gemildert werden muß und in manchen Fällen beseitigt werden kann.

Eindeutiger Marktführer ist COM/ActiveX mit einem mehrjährigen Marktvorsprung im Hinblick auf konkrete Implementierungen. JavaBeans (Client-Komponenten) und Enterprise JavaBeans (Server-Komponenten) holen auf. CORBA Components sind hingegen derzeit lediglich spezifiziert, jedoch noch nicht konkret implementiert. Über zukünftige Marktanteile mit Sicht auf 5 Jahren ist keine seriöse Vorhersage möglich, da sich Marktanteile sehr schnell verschieben können. Es ist davon auszugehen, daß für viele Aufgaben immer mehrere Implementierungen als Alternative verfügbar sein werden. Gibt es z. B. eine am Markt erfolgreiche COM/ActiveX-Komponente für eine bestimmte Aufgabe, wird mit Sicherheit eine funktionsäquivalente Enterprise JavaBean verfügbar werden.

Auf absehbare Zeit ist wohl kaum damit zu rechnen, daß die technologischen Gräben zwischen den Komponententechnologien eingeebnet werden. Dem steht das Interesse von Software-Herstellern entgegen, jeweils eigene Spezifikationen zu verankern, Schnittstellen zu kontrollieren und auf diesem Weg Kundenbindung zu bewirken.

Wie zieht sich das Anwenderunternehmen aus der Affäre, das nicht Spielball von Herstellerinteressen sein möchte? Als einzige Möglichkeit bleibt, auf weitestgehende Herstellerunabhängigkeit zu setzen und die Plattform mit der geringsten "Herstellernanfälligkeit" zur strategischen Plattform zu erklären. Nach Lage der Dinge kommt lediglich die CORBA Components-Spezifikation in Frage, die allerdings die Interoperabilität mit Enterprise JavaBeans bereits berücksichtigt.

Die strategische Plattform gilt als Referenzplattform. Nur wenn für eine bestimmte Anforderung kein aktuelles Komponentenangebot verfügbar ist, kann fallweise ausgewichen werden. Dabei muß jedoch gewährleistet sein, daß die Interoperabilität durch Technologiebrücken herstellbar ist. Im Idealfall ist eine aus taktischen Gründen "eingebaute" Komponente später wieder substituierbar, wenn die Angebotslücke ausgefüllt ist.

Die Technologieabhängigkeit wirkt auch auf anderen Ebenen weiter. Beispiele sind die in der Plattform für verteilte Anwendungen enthaltenen Dienste. Aber auch Entwicklungs-Werkzeuge unterstützen jeweils nur bestimmte

Basistechnologien (z. B. Entwicklungs-Werkzeuge für Java-Anwendungen). CASE-Werkzeuge sind hingegen wesentlich technologieneutraler. Die Technologieabhängigkeit ist noch immer sehr stark ausgeprägt und ein wesentlicher Hemmschuh auf dem Weg zu komponentenbasierten Anwendungssystemen.

### **Der Software-Entwicklungsprozeß**

Gegenwärtig ist der Software-Entwicklungsprozeß noch sehr stark auf Software-Fachleute ausgerichtet. Mitarbeiter von Fachabteilungen sind nur an frühen Aktivitäten im Projektverlauf beteiligt, so vor allem der Anforderungsanalyse (Requirements Analysis). Sie verstehen kaum die spezifische Terminologie der IT-Fachleute und sind es gewohnt, ihre Anforderungen über die Bewertung von Prototypen der Dialogoberfläche deutlich zu machen.

Die komponentenbasierte Anwendungsentwicklung moderner Prägung muß qualifizierte und gut ausgebildete Mitarbeiter der Fachabteilungen in die Lage versetzen, geeignete Komponenten zu finden und diese in einer geeigneten Weise zu Anwendungen zusammenzusetzen. Die Software-Entwicklungsprozesse, die Vorgehensmodelle, sind darauf nicht ausgerichtet.

Obwohl gegenwärtig die fachabteilungsgetriebene komponentenbasierte Anwendungsentwicklung noch in keiner Weise bzw. höchstens in ersten Ansätzen verfolgt wird, ist es sinnvoll, ein erstes Pilotprojekt aufzusetzen. Die Fachabteilung kann von einem oder mehreren IT-Mitarbeitern unterstützt werden, die die Rolle eines Coach übernehmen. Die intensivere Einbindung der Fachabteilungen in die Anwendungserstellung ist ein Prozeß, der mehrere Jahre in Anspruch nehmen und die bisherige "Entwicklungskultur" im Kern verändern wird. Deshalb ist es kein Fehler, derartig tiefgreifende Änderungen bereits zu einem frühen Zeitpunkt zu initiieren.

### **Die Entwicklungs-Infrastruktur**

Die komponentenbasierte Anwendungsentwicklung ist ohne eine mächtige Entwicklungs-Infrastruktur von vornherein zum Scheitern verurteilt. Ein wesentlicher Aspekt der komponentenbasierten Anwendungsentwicklung ist die Wiederverwendbarkeit vorhandener Komponenten. Was wiederverwendbar sein soll, muß zuerst wiedergefunden werden können. Deshalb muß ein Komponenten-Repository vorhanden sein, das sämtliche Komponenten aufnehmen kann und Werkzeuge bereitstellt, mit deren Hilfe Komponenten nach bestimmten Profilen gesucht und aufgefunden werden können.

Die heute übliche Entwicklungs-Infrastruktur verfügt noch nicht über die geforderten Möglichkeiten. Produkte werden erst nach und nach verfügbar werden und weisen in den ersten Produktversionen noch wenig Funktionalität auf.

### **Zusammenfassung und Ausblick**

Bis die Möglichkeit, Anwendungen aus vorgefertigten Komponenten zusammenbauen zu können, tatsächlich genutzt werden kann, ist es noch ein sehr weiter Weg. Dabei steht nicht die Technologie im Vordergrund, sondern eindeutig Fragen der Organisation und der "Software-Produktionskultur".

Selbst entwickelte und von Software-Herstellern lizenzierte Komponenten müssen zusammenwirken können. Bei lizenzierten Komponenten hat das Anwenderunternehmen in der Regel keine Möglichkeit, auf deren zugrundeliegende Architektur Einfluß zu nehmen. Bei selbst entwickelten Komponenten ist jedoch eine gemeinsame Architekturbasis unverzichtbar. Nicht nur das Fehlerrisiko wird vermindert, auch der Entwicklungsprozeß wird beschleunigt.

Natürlich stellt sich die grundsätzliche Frage, ob die intensivere Einbeziehung der Fachabteilungen in die Software-Produktion überhaupt sinnvoll ist. An Gegenargumenten herrscht kein Mangel, jedoch wird in aller Regel mit auf die Gegenwart bezogenen Argumenten operiert. Wenn jedoch vollherzig kürzere Entwicklungszyklen und eine kürzere time to market angestrebt werden, führt an einer Verlagerung von Software-Entwicklungsaufgaben in die Fachabteilung kein Weg mehr vorbei. Selbstverständlich kann auch der Software-Entwicklungsprozeß nicht in der heutigen Form beibehalten werden und die Fachabteilung kann auch nur auf der Grundlage der komponentenbasierten Anwendungsentwicklung einbezogen werden.

Die Rolle der IT-Abteilung muß sich wandeln. Sie bleibt damit beschäftigt, die IT-Infrastruktur des Unternehmens weiterzuentwickeln und erstellt Komponenten auf Anforderung von Fachabteilungen. Die Fachabteilungen bauen Anwendungen aus bereits getesteten Komponenten zusammen und erstellen die Dialog-Oberfläche. Die Betonung liegt auf Konfiguration, kaum auf Entwicklung. Der Produktionsprozeß wird zeitlich verkürzt und außerdem werden Ressourcen sparsamer eingesetzt. Es kann nicht ausbleiben, daß Fachabteilungen die Möglichkeiten moderner Technologie immer stärker nutzen. Dies wird auch aus anderen Anlässen ohnehin erzwungen. Im Zeitverlauf werden Fachabteilungen und IT-Abteilung mehr und mehr eine gemeinsame Sprache sprechen.

Die komponentenbasierte Anwendungsentwicklung steht noch ganz am Anfang. Es wird noch mehrere Jahre dauern bis sämtliche Voraussetzungen geschaffen sind, auch Fachabteilungen in der oben kurz skizzierten Weise mit einzubeziehen. Ganz wesentlich über den Erfolg des Transformationsprozesses entscheidet mit, ob es gelingt, die erforderliche Kulturänderung erfolgreich zu bewerkstelligen. Einen Wettbewerbsvorteil werden in jedem Fall die Unternehmen erzielen, denen es gelingt, den Veränderungsprozeß schon frühzeitig zu beginnen.