

# Business Agility – Vermeidung von Guerilla-Software



## Die Fachabteilungen der Finanzdienstleister

Die Fachabteilungen der Finanzdienstleister weichen zur Umsetzung von Fachapplikationen oft auf Excel und Access aus, da sie auf diesen Plattformen ihre Applikationen flexibel ändern und erweitern können. Diese Form der Umsetzung bietet meist die Möglichkeit, sehr nah an der fachlichen Begriffswelt zu bleiben. Business Agility bezeichnet einen Ansatz, Software zu bauen, der die beschriebenen Vorteile mit den Anforderungen an einen geordneten Entwurf und Betrieb von Software-Systemen verbinden möchte.

## Momentaufnahme aus dem Projektgeschäft

**Insbesondere** in den Fachabteilungen von Banken und Kapitalanlagegesellschaften sind Microsoft Excel und Microsoft Access beliebte Plattformen, um Fachapplikationen zu erstellen. Sie werden aus den folgenden Gründen Individualentwicklungen (wie beispielsweise in Java) vorgezogen:

- Die Ergebnisse sind unmittelbar verwendbar und müssen auf keine Release- oder Beauftragungszyklen achten.
- Die Anwendung wird von Fachleuten für Fachleute entwickelt. Das bedeutet, dass die aufwändige Aufbereitung der Anforderungen für die Kollegen der IT-Abteilung oder eines Dienstleisters entfällt und die Applikationslogik in der Sprache der Fachabteilung (z. B. in Formeln) und nicht im Source Code

oder in technischen Konfigurationen beschrieben ist.

- Die fachliche Logik lässt sich inkrementell und auf Basis realer Daten definieren und muss nicht rein theoretisch vorab auf dem Reißbrett entworfen werden. Zudem sind kleinere, meist zeitkritische Änderungen jederzeit möglich – z. B. aufgrund gesetzlicher Anforderungen.

Trotz der beschriebenen Vorteile für die Fachabteilungen handelt es sich aus Sicht der IT-Abteilung um nicht wartbare Anwendungen. Die Gründe dafür: Bei der Realisierung der Anwendung wurde ausschließlich auf fachliche Aspekte geachtet, in den wenigsten Fällen existiert eine Dokumentation, das Anwendungswissen existiert nur in einigen wenigen Köpfen und die Änderungen an der Applikation sind in der Regel nicht versioniert und können nachträglich weder nachvollzogen noch rückgängig gemacht werden.

Business Agility versucht durch den Entwurf technisch wartbarer Anwendungen, die eine flexible Konfiguration in der Sprache der Fachabteilung bieten, beiden Parteien gerecht zu werden.

## Flexibilisierung der Konfiguration

Moderne Fachapplikationen sind in der Regel heute bereits konfigurierbar. Diese Konfigurationen werden allerdings meist gemeinsam mit der Anwendung ausgeliefert – beispielsweise in Form von INI- oder XML-Dateien (siehe linke Seite der nachfolgenden Abbildung).

Eine flexible Konfiguration wird nicht gemeinsam mit der Anwendung bzw. dem Anwendungskern aus-



geliefert, sondern befindet sich beispielsweise in einer Datenbank. Sie ist damit zur Laufzeit änderbar und kann gleichzeitig historisiert werden.

Die häufig zur Konfiguration verwendeten XML- oder INI-Dateien haben die Restriktionen, dass sich nur statische Werte konfigurieren lassen, aber Formeln oder Regeln nur schwer abzubilden sind. Um nicht an diese Restriktionen gebunden zu sein, finden für eine flexible Konfiguration mächtige Skript-Sprachen wie Groovy oder Scala Verwendung. Dadurch ändert sich das Zusammenspiel von Anwendung und Konfiguration: Verfügt eine Anwendung über eine klassische Konfiguration, liest der im Anwendungskern implementierte Prozess einzelne Werte aus der Konfiguration, in einem Szenario mit flexibler Konfiguration stellt der Anwendungskern den Rahmen für alle fachlichen Konfigurationen dar. Der darin implementierte Prozess sucht sich in den einzelnen Schritten passende Konfigurationsbausteine und delegiert die Ausführung des Prozessschrittes direkt an diese Bausteine.

## Nutzen, Vorteile, Besonderheiten

**Die Auslagerung** der Konfiguration der Anwendung in eine externe Quelle und die Verwendung von Skriptsprachen für mächtige Konfigurationen lösen das Problem der mangelnden Flexibilität – Anwendungen lassen sich kurzfristig, flexibel, nachvollziehbar und kontrolliert ändern. Eine herkömmliche

Groovy-Klasse ist aber für einen Fachexperten ähnlich schwierig zu verstehen wie eine komplexe XML-Konfiguration oder ein Java Source Code.

Damit die Definition der Konfigurationsbausteine in der Fachabteilung erfolgen kann, wird das Konzept der flexiblen Konfiguration ergänzt durch Domain Specific Languages (DSLs). Die Sprache, in der die Konfiguration „geschrieben“ ist, kann technisch dann zwar Groovy, Scala oder Xtext sein, wird inhaltlich aber an den Sprachgebrauch der Fachabteilung angelehnt. Das folgende fiktive Beispiel zeigt eine einfache Reduktion um Steuern für eine Ausschüttung:

```
Wenn (
    Land(„Deutschland“),
    Dann {
        Ausschüttung
        - Kapitalertragsteuer
        - Solidaritätszuschlag
    }
)
```

DSLs können entweder wie im Beispiel in textueller Form oder grafisch formuliert werden – beispielsweise zur Modellierung komplexer Finanzinstrumente. Des Weiteren existieren Mischformen, in denen die modellierten Finanzinstrumente beispielsweise um Formeln und Regeln zur Berechnung von Kennzahlen angereichert werden.

Fragen zu diesen oder anderen Themen beantworten wir Ihnen jederzeit gerne.

#### **Cofinpro AG**

Untermainkai 27–28, 60329 Frankfurt am Main

Tel: +49 (0) 69 - 2 99 20 87 60

Mail: [welcome@cofinpro.de](mailto:welcome@cofinpro.de)

[www.cofinpro.de](http://www.cofinpro.de)

**COFINPRO**  
Finest Processes in Finance

**Cofinpro unterstützt Deutschlands führende Finanzdienstleister** bei der Verbesserung von Geschäftsprozessen. Wir transformieren Strategien in Prozesse und implementieren diese Prozesse in der IT. Durch Business Transformation schaffen wir für unsere Kunden entscheidende Wettbewerbsvorteile in einem Finanzmarkt, der sich immer noch sehr schnell verändert und vor spannenden Herausforderungen steht. Unser Erfolgskonzept ist dabei ein ganz besonderes: Wir kombinieren hohe Fach- und Technologiekompetenz mit den Methoden unseres Business Engineering Frameworks – durch diese einzigartige Kompetenzbündelung erreichen wir für unsere Kunden exzellente Ergebnisse: „Finest Processes in Finance“.