

# Das Projekt „Blue Eagle“ – ein Beispiel für die Anwendung objektorientierter Methoden in der Analyse-Phase

von Michael Wrede, Olaf Merkert, Constantin Szallies

Bereits in der Analyse-Phase eines Projektes können objektorientierte Technologien erfolgreich eingesetzt werden. In Großprojekten müssen dabei spezielle Bedingungen beachtet werden. Der Beitrag beschreibt die Entwicklungsmethodik in einem Projekt der Immobilienwirtschaft. Darin kommen projektspezifische Generatoren, Anforderungs-Datenbanken sowie die UML zum Einsatz. Die mit der Methodik sowie den Werkzeugen gemachten Erfahrungen werden geschildert.

## Projekthintergrund

Unter dem Namen GES wurde das ERP-System der DePfa IT Services seit den 50-er Jahren zum führenden EDV-Dienstleistungssystem für die Immobilienwirtschaft in Deutschland. Durch organisches Wachstum und Übernahmen wurde ein Produkt-Portfolio entwickelt, das seit 1990 drei vollwertige Systeme enthält.

Die Wartung und Weiterentwicklung von drei ähnlichen Systemen steht strategischen Zielen wie dem Wachstum in Europa und der Erschließung neuer Marktsegmente entgegen. Die Systeme wurden in Teilen bereits in den 70-er Jahren konzipiert. Neue Module wurden um die „alte“ Fachlogik herum entwickelt.

1999 faßte DePfa IT Services den Beschluß, ein neues System unter dem Arbeitstitel „Blue Eagle“ zu entwickeln. „Blue Eagle“ soll die ganze Breite der Geschäftsvorfälle der Immobilienwirtschaft in Europa unterstützen. Es wird in der Endversion über alle Funktionen der Altsysteme hinaus alle derzeit bekannten Anforderungen abdecken.

Die personellen Ressourcen für dieses Projekt sind über fünf Länder mit ca. 12 Standorten verteilt. Aus den Systemhäusern und den konzernverbundenen Unternehmen wurde ein Projektteam aus ca. 100 Personen zusammengestellt. Dieses Team konzipiert und realisiert in 15 Teilprojekten das neue Produkt.

## Das Fachkonzept des „Blue Eagle“

Das Ziel ist eine vollständige Dokumentation des benötigten Leistungsumfangs. Dies ist insbesondere bei der Größe des Projektes zwingend erforderlich. Es darf keine Abhängigkeit von „Wissen in den Köpfen“ geben.

Im Projekt wird die Methode Objects 9000 von Rösch Consulting eingesetzt. Im folgenden werden die einzelnen Teilschritte der Analyse-Phase im Projekt „Blue Eagle“ beschrieben. Diese Schritte werden iterativ durchlaufen, wenn in einem späteren Schritt neue Erkenntnisse gewonnen werden.

### Schritt 1: Anforderungen und Akzeptanzkriterien

Anwendungsexperten beschreiben die erforderliche Systemfunktionalität in einer Menge von *Anforderungen*. Eine Anforderung beschreibt einen kleinen Teil der Fachlogik in Textform. Zu einer Anforderung müssen *Akzeptanzkriterien* erfaßt werden. Erst dadurch wird die Erfüllung einer Anforderung messbar. Dies entspricht der ISO 9000 Definition von Qualität: Eine Anforderung ist fehlerfrei umgesetzt, wenn das System alle Akzeptanzkriterien erfüllt (siehe dazu auch [Roe98]).

**Anforderung:** Eine Person kann gelöscht werden, wenn sie in keinem Vertrag mit dem Zustand „aktiv“ verwendet wird. Kann die Person nicht gelöscht werden, wird die Meldung „Die Person #... kann nicht gelöscht werden...“ erzeugt. Sie wird in aktiven Verträgen verwendet.“

**Akzeptanzkriterium „Löschung wird abgewiesen“**

**Situation:** Es existiert eine Person #4665. Diese Person wird im Vertrag #A3336 verwendet. Dieser Vertrag ist aktiv.

**Aktion:** Der Benutzer versucht, Person #4665 zu löschen.

**Erwartetes Ergebnis:** Die Meldung „Die Person #4665 kann nicht gelöscht werden....“ wird erzeugt. Die Person #4665 existiert unverändert.

Abbildung 1 zeigt ein vereinfachtes Beispiel für eine abgesicherte Anforderung. Wie zu erkennen ist, ist ein Akzeptanzkriterium ein fachlicher Testfall. Mindestens eines muß zwingend zu jeder Anforderung erstellt werden. Wie viele Akzeptanzkriterien aufgestellt werden, liegt im Ermessen des Anforderungsstellers. Damit trägt er die Verantwortung für die Qualität der Anforderung.

Neben den Anwendungsexperten werden die späteren Systembenutzer sowie deren fachliche Berater in die Konzeption einbezogen. Eine repräsentative Anzahl von Kunden prüft und erweitert die vorhandenen Konzepte aus Anwendersicht. Die Ergebnisse werden ebenfalls als Anforderungen und Akzeptanzkriterien direkt den Teilprojekten zugeordnet.

Diese Vorgehensweise hat entscheidende Vorteile:

- DePfa IT Services weiß vor der Erstellung des Produktes, dass es marktfähig wird.
- Der zukünftige Kunde weiß, dass er ein brauchbares Produkt erhalten wird.
- Die Anpassung an den Markt vor der Einführung erspart nachträglichen Änderungsaufwand.

Hersteller und Kunde tragen so zu einer kostengünstigen Entwicklung bei.

Für die Erfassung der Anforderungen und Akzeptanzkriterien wird Lotus Notes verwendet. Die dort erfaßten Texte können mit geringem Aufwand in Handbücher umgewandelt werden. Die Texte müssen dazu formal den Kriterien genügen und den geforderten Leistungsumfang vollständig beschreiben. Die verwendete Sprache muß verständlich und – für die Internationalisierung – leicht übersetzbar sein. Kurze Sätze und definierte Begriffe erleichtern dies.

Daraus erstellen technische Autoren die Anwendungsbeschreibungen vor der Erstellung der Software. In der Regel ist nur eine sprachliche Aufbereitung erforderlich. Die fachlichen Inhalte müssen in den Anforderungen enthalten sein.

### Schritt 2: OOA-Modell

Die Anforderungen werden in ein objektorientiertes Analyse-Modell (OOA Modell) integriert. Einzelne Anforderungen werden auf Modellierungselemente wie Klassen, Attribute usw. abgebildet. Im Modell bewähren sich die Anforderungen zum ersten Mal. Unvollständige und widersprüchliche Beschreibungen der Fachlogik werden entdeckt.

Die Modellierung kann zeitlich versetzt oder parallel durchgeführt werden. Es hat sich als sinnvoll erwiesen, komplexe Themen in kleinen Einheiten darzustellen. Dazu werden im Modellierungswerkzeug Sichten verwendet. Diese enthalten einen fachlich zusammenhängenden Ausschnitt des gesamten Modells.

Es hat sich bewährt, Teams in Anforderungssteller und Modellierer aufzuteilen. So kann sich jeder Mitarbeiter auf seine Kernkompetenzen konzentrieren. Für den Anforderungssteller sind dies praktische Erfahrungen aus der Immobilienwirtschaft, für den Modellierer die Fähigkeit zu abstrahieren. Zusätzlich wird ein 4-Augenkonzept realisiert. Häufig werden gerade Basisfunktionalitäten nicht ausreichend in den Anforderungen beschrieben, sondern stillschweigend vorausgesetzt. Diese werden aber unbedingt benötigt. Im Modell stellt sich schnell heraus, was fehlt. Es kommt zu Fragen und Rückmeldungen an die Anforderungssteller.

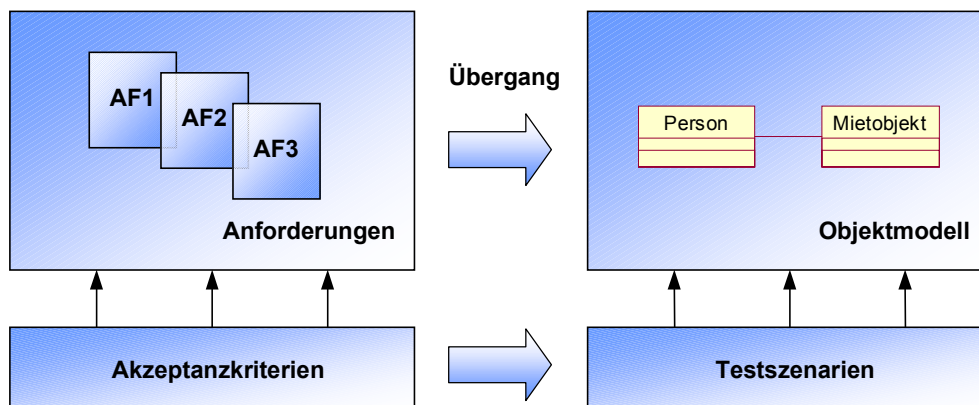
Das OOA-Modell wird mit Hilfe von Rational Rose erstellt. Ein projektspezifisches Werkzeug überprüft automatisch, ob das Modell konform zu den Modellierungsrichtlinien ist.

### Schritt 3: Simulationsmodell

Das OOA-Modell wird in ein Simulationsmodell umgesetzt. Ein Simulationsprogrammierer implementiert die Fachlogik ohne Beachtung technischer Aspekte wie Persistenz oder Benutzeroberfläche. Ein projektspezifischer Codegenerator beschleunigt die Entwicklung, indem er das Modell auf C++ Code abbildet. Danach müssen die fachlichen Methoden manuell programmiert werden.

Bei der Umsetzung entdeckt der Programmierer unvollständige Angaben im OOA-Modell. Dies führt zu einer weiteren Überprüfung der Anforderungsmenge. Im Gegensatz zu anderen Vorgehensweisen liegt die Implementierung der Fachlogik vollständig in der Verantwortung des Analyse-Teams. Dadurch sind kurze Kommunikationswege zwischen Anforderungssteller und -umsetzer gegeben.

Im Projekt werden fachliche und technische Teile des Systems vollständig getrennt voneinander erstellt. Es wurde eine Simulationsarchitektur mit einer sogenannten Business-Objekt Schicht entwickelt [Met00]. Diese Schicht ist die einzige Stelle, an der sich fachlicher und technischer Code berühren. Der fachliche Code kann später unverändert im produktiven System eingesetzt werden, da die Produktivarchitektur eine äquivalente Business-Object Schicht bereitstellt.



#### **Schritt 4: Testszzenarien**

Die Akzeptanzkriterien werden als ausführbare Programme implementiert. Diese Testszzenarien arbeiten mit den Objekten des Simulationsmodells. Sie überprüfen, ob die Klassen des Simulationsmodells die Anforderungen korrekt und vollständig wiedergeben. Wie in Abbildung 2 zu sehen ist, sichern sie den Übergang von textuellen Anforderungen zum formalen Objektmodell.

Die Erstellung der Testszzenarien wird ebenfalls durch einen Generator unterstützt.

### **Komponenten-Technologie im Blue Eagle**

Im Projekt „Blue Eagle“ arbeiten mehr als 100 Personen aus verschiedenen Unternehmen zusammen. Das Projekt wurde daher in 15 Teilprojekte unterteilt. Der initiale Projektschnitt erfolgte nach fachlichen Themen. Jedes Thema entspricht mindestens einer Komponente in der späteren Anwendung. Es wird durch den Projektauftrag jedes Teilprojektes umrissen.

#### **Analyse-Komponenten**

Mit der Aufteilung ist die Komplexität des Gesamtsystems beherrschbar. Ausserdem sollen einzelne Teile des Systems austauschbar und Fremdsysteme leicht zu integrieren sein. Diese Projektziele erzeugten jedoch ein bekanntes Problem: Wie können die Teile anschließend zu einem Ganzen zusammengesetzt werden? Im „Blue Eagle“ wird zur Lösung dieses Problems Komponenten-Technologie verwendet.

Normalerweise konzentriert sich die Analyse auf die äußere Schnittstelle des Systems. Durch die Aufteilung in Komponenten war es erforderlich, die internen Schnittstellen explizit zu planen und zu modellieren. Als Konsequenz mussten bereits in der Analyse-Phase des Projektes Konzepte der Komponenten-Technologie Berücksichtigung finden.

Im Projekt wird dazu die Methode Components 9000 verwendet. Components 9000 erweitert Objects 9000 um Komponentenaspekte und beinhaltet das Fachkonzept und die fachliche Implementierung. Zur exakten Beschreibung und Absicherung der Komponentengrenzen werden ebenfalls Anforderungen und Akzeptanzkriterien eingesetzt. Da sich die Analyse-Teams auf fachliche Schnittstellen konzentrieren, spricht man von Analyse-Komponenten.

#### **Schnittstellenfindung**

Jedes Teilprojekt bearbeitet ein fachliches Thema. An den Grenzen jedes Themas ergeben sich Schnittstellen, an denen Daten oder Dienstleistungen verwendet werden, die von einem anderen Teilprojekten bereitgestellt werden.

Am Anfang des Projektes waren die genauen Schnittstellen zwischen den Komponenten noch unscharf. Sie werden im Laufe der Analyse zwischen den einzelnen Teilprojekten verhandelt. Dies erfolgt in direkter Absprache mit anschließendem Austausch von Anforderungen und Akzeptanzkriterien.

Auch wenn der initiale Projektschnitt ein scheinbar vollständiges Bild ergibt, finden sich im Laufe des Projektes diverse Lücken. Diese werden erkannt, wenn es für Anforderungen keine akzeptierenden Teilprojekte gibt.

Die Verhandlung und Dokumentation von Komponentengrenzen ist ein integraler Bestandteil von Components 9000 und wird durch das Konzept von „Initiator“ und „Bearbeiter“ realisiert.

#### **Initiator und Bearbeiter**

Zu jeder Anforderung gehören zwei Teilprojekte: Das I-TP (Initiator Teilprojekt) und das B-TP (Bearbeitendes Teilprojekt). Jede Anforderung wird vom I-TP formuliert und vom B-TP umgesetzt.

Entspricht das I-TP dem B-TP, ist die Anforderung eine interne Anforderung. Ist das I-TP ungleich dem B-TP, handelt es sich um eine Schnittstellen-Anforderung. Mit ihr wird dokumentiert, daß das I-TP etwas vom B-TP benötigt, um seine Aufgaben zu erfüllen. Das B-TP kann die Schnittstellen-Anforderung akzeptieren oder zurückweisen. Wird sie akzeptiert, verpflichtet sich das B-TP die geforderte Funktionalität zu implementieren. Das I-TP prüft die korrekte Umsetzung der Anforderung mittels der von ihm aufgestellten Akzeptanzkriterien. Das B-TP kann sein Arbeitsergebnis bereits vor der Auslieferung an das I-TP mit diesen Akzeptanzkriterien kontrollieren.

Besonders zu Beginn des Projektes haben B-TPs viele Schnittstellenanforderungen abgewiesen. Meist fühlte sich das B-TP für die Anforderung nicht zuständig. Eine Absprache zwischen den Beteiligten führt meist zu einer der folgenden Lösungen:

- Die Anforderung wird fallengelassen, da sie nicht mehr sinnvoll erscheint.
- Die Anforderung wird nach einer gemeinsamen Überarbeitung vom B-TP akzeptiert. Besonders wichtig ist hier die Entfernung von synonymen und homonymen Fachbegriffen. Dafür existiert ein Glossar mit allen Fachbegriffen und deren Definitionen.
- Ein anderes Teilprojekt wird als B-TP identifiziert. Dies kann möglicherweise auch ein neues Teilprojekt sein.

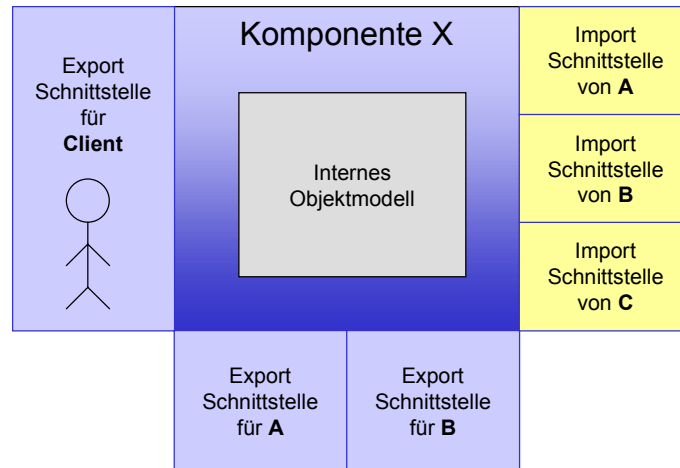
- Die Anforderung ist eine interne Anforderung des I-TP.

*Beispiel: In jedem Teilprojekt werden Anforderungen zu Auswertungen gesammelt. Anwender möchten vorgegebene sowie frei konfigurierbare Auswertungen, die häufig Varianten vorgegebener Auswertungen sind. Da sich kein Teilprojekt für die Umsetzung dieses Anforderungspaketes fand, wurde das neue Teilprojekt „Reportgenerator“ initiiert.*

## Modellierung der Schnittstellen

Technische Komponentengrenzen sollten den fachlichen Teilprojektgrenzen entsprechen. Dadurch erhält man klare Verantwortungsbereiche. Jedes Teilprojekt beschreibt und implementiert eine oder mehrere Komponenten vollständig. Es gibt keine Komponente, die von zwei Teilprojekten bearbeitet wird.

Wie in Abbildung 3 dargestellt, werden Schnittstellen-Anforderungen explizit als Komponenten Schnittstellen modelliert. Man unterscheidet die Schnittstellen vom internen Objektmodell, das nur innerhalb der Komponente sichtbar ist.



Aus Sicht einer einzelnen Komponente sind Schnittstellen entweder Import- oder Export-Schnittstellen. Eine Import-Schnittstelle berücksichtigt Anforderungen an andere Komponenten. Anforderungen von fremden Komponenten werden dagegen in den Export-Schnittstellen berücksichtigt. Durch diese explizite Kennzeichnung der beiden Schnittstellen werden spätere Wartungs- und Erweiterungsarbeiten vereinfacht. Der Austausch sowie der optionale Einsatz von Komponenten ist möglich.

Der Übergang zur Implementierung wird ebenfalls durch den Generator unterstützt. Dabei werden Schnittstellen auf abstrakte C++ Klassen abgebildet.

## Notes Tool

In der Analysephase werden Informationen in zahlreichen Dokumenten gesammelt. Diese müssen von allen beteiligten Personen gelesen und bearbeitet werden. Um die Verwaltung dieser Dokumente zu vereinfachen, wird im Projekt „Blue Eagle“ eine spezielle Notes Datenbank verwendet. Dadurch sind Anpassungen der Datenbank an die projektspezifischen Bedürfnisse möglich.

## Verteiltes Arbeiten

Die vollständige Projektdokumentation wird an jedem der Projektstandorte (mehrere Städte in vier Ländern) benötigt. Die Notes-Infrastruktur unterstützt verteiltes sowie Offline Arbeiten. In Workshops werden Repliken der Serverdatenbestände verwendet. Eine Replik umfasst den gesamten Serverdatenbestand und kann mit diesem abgeglichen werden. In der Replik gemachte Änderungen sind nach dem Abgleich in den Datenbestand des Servers integriert. Dieser Mechanismus wird auch von Mitarbeitern genutzt, die im privaten Arbeitszimmer tätig werden.

In den Systemhäusern arbeiten permanent mehrere Teams. Durch die Verwendung der Serverdatenbank sind alle Ergebnisse stets direkt verfügbar. Geänderte Dokumente sind jederzeit identifizierbar. Werden bestimmte Dokumente geändert, kann eine Person automatisch via EMail benachrichtigt werden.

## Strukturierung der Anforderungsmenge

Alle Dokumente sind nach den diversen Kriterien in Sichten darstellbar. Anforderungen können z.B. nach Status, nach Thema und nach Anforderungssteller und –bearbeiter angezeigt werden. Jeder Benutzer kann so seinen bevorzugten Blickwinkel auf die Anforderungsmenge finden. Er kann Schwerpunkte bilden und Engpässe erkennen. In den Sichten sind unter anderem auch Statusinformationen der einzelnen Anforderungen zu finden (vgl. Abb 4). Durch deren grafische Darstellung fallen überfällige Statusveränderungen direkt ins Auge.

| ▼ Vertragsverwaltung |   |   |
|----------------------|---|---|
| 001-001              | ▶ | Mustervertrag anlegen                           |
| 001-002              | ▶ | Mustervertrag verwenden                         |
| 001-003              | ▶ | Umwandlung von Musterverträgen und Vorverträgen |
| 001a-002             | ▶ | Zuordnung der Vertragsarten löschen             |
| 001a-004             | ▶ | Leistungen der Vertragsart zuordnen             |
| 001a-005             | ▶ | Leistungen zum Rechnungskreis                   |
| 002-001              | ▶ | zulässige Rollen der Vertragsart                |

Die interne Verknüpfung der Dokumente erlaubt die zusammenhängende Bearbeitung von Themen. Weitere Informationen können „angesprungen“ werden, ohne den aktuellen fachlichen Kontext zu verlassen. Alle Anforderungen und Probleme sind offenbar für Dritte und die Projektleitung. Konfliktmanagement kann direkt an fachlichen Themen ansetzen.

### **XML-Schnittstelle und Auswertungen**

Die Anforderung sowie alle übrigen Objects 9000 Dokumente sind als XML-Dokumente ex- und importierbar. Auf diese Weise können sie vom Übersetzungswerkzeug der DePfa Systems verarbeitet werden. Ausserdem ist so ihre Integration in das Werkzeug ClearCase möglich, das für das Konfigurationsmanagement eingesetzt wird.

Statusinformationen zu den Anforderungen geben Aufschluss über ihre aktuelle Position im Entwicklungsprozess. Auf Basis dieser Statusinformationen werden diverse quantitative Metriken zum Projektfortschritt automatisch erzeugt. Sie liegen als Excel-Tabellen und –Diagramme vor. Die Arbeitsleistung der Teilprojekte kann so verglichen werden.

Die Zuordnung der Dokumente zu den Teilprojekten gibt jederzeit Auskunft über Abhängigkeiten zwischen ihnen. Sie sind einem automatisch erstellten Abhängigkeitsdiagramm entnehmbar (vgl. „Abhängigkeits-Matrix“ in [Wöh00]).

### **Fazit**

Erst mit dem Einsatz von Komponententechnologie wird die Anwendung von OO in Großprojekten möglich. Dies sollte bereits in der Analysephase berücksichtigt werden. Die Verwendung einer Anforderungsdatenbank ist unumgänglich, um mit vielen Mitarbeitern effizient ein gemeinsames Ergebnis zu produzieren. Es hat sich gezeigt, dass individuelle Einflussmöglichkeiten auf die eingesetzten Werkzeuge überaus hilfreich sind.

### **Literatur**

[Met00] P. Metzen, B. Paffenholz, C. Szallies, F. Töpler: Architekturgetriebene Generierung für objektorientierte Softwaresysteme, in OBJEKTSpektrum 2/2000

[Roe98] M. Rösch, Muß Software wirklich Fehler haben?, in OBJEKTSpektrum 1/1998

[Wöh00] C. Wöhrle: Komponentenbildung in großen Softwareprojekten, in OBJEKTSpektrum 4/2000

### **Autoren**

Michael Wrede (E-mail-Adresse: [mwrede@depfa-it.com](mailto:mwrede@depfa-it.com)) arbeitet als Teilprojektleiter im Bereich Produkt Services der DePfa Systems GmbH in Mainz. Er ist für das fachliche Konzept und die Implementierung der Komponente „Vertragsverwaltung“ im Projekt „Blue Eagle“ verantwortlich.

Olaf Merkert und Constantin Szallies ([olaf@merkert.de](mailto:olaf@merkert.de) und [cms@wohnklo.de](mailto:cms@wohnklo.de)) arbeiten als freie IT Berater.