



Von Anforderungen bis zur fertigen Anwendung:

Schneller und einfacher zum Ziel
mit Eclipse, UML und TREND

Timo Krauss



Technologie beherrschen ► Prozesse gestalten ► Lösungen schaffen ►►

Was soll vermittelt werden?



- **Integration von Anforderungen und Implementierung zu einem durchgängigen Entwicklungsprozess sorgen für „das richtige“ Ergebnis im Projekt**
- **Model Driven Development sorgt für eine hohe Umsetzungsgeschwindigkeit**
- **Beides Integriert führt zu einem effektiven**

„Requirements Driven Development“

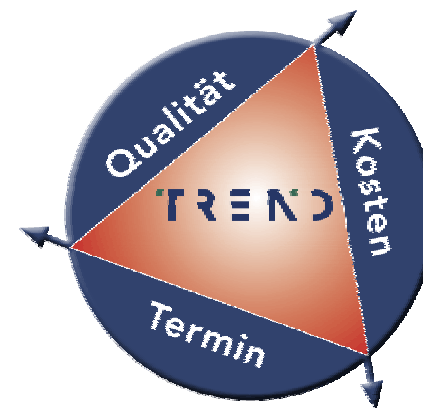
- **Live Demo des Ansatzes und Eindruck über das TREND Project Development Environment**



Herausforderungen in der AW-Entwicklung (Auszug!!)



- Gute Kommunikation im Team und zwischen Fach- und IT-Seite
 - Verstehen was der Kunde will
 - Der IT vermitteln was man will
- In Time und Budget bleiben
 - Dem Kunden schnell und günstig frühe Ergebnisse liefern
- verlässliches Projektcontrolling
- Motiviertes Team (Entwickler und Kunden)
- u.v.a.m.



Sind Fach- und IT-Abteilung kompatibel?



Kunde/Fachbereich

- Benutzer
- Sponsor
- Auftraggeber
- Management



IT-Abteilung

- Analytiker
- Designer
- Implementierer
- Tester



Verbesserung der Kommunikation



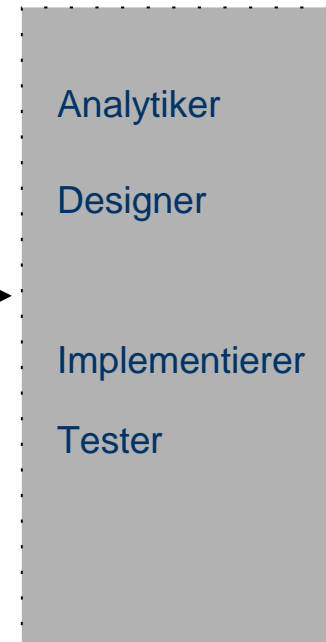
Kunde/Fachbereich



BA/RE



IT-Abteilung



Das Optimum



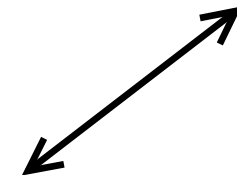
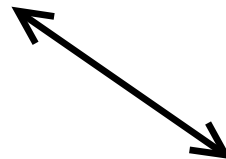
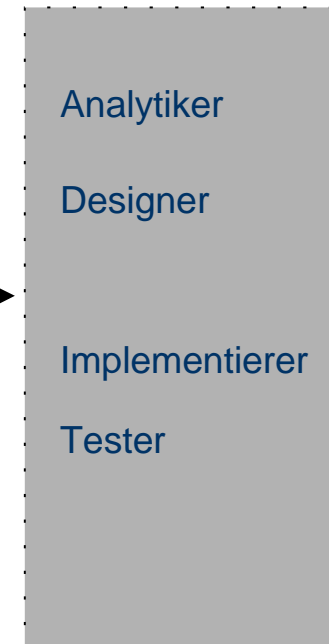
Kunde/Fachbereich



BA/RE



IT-Abteilung



Ziel: Das gemeinsame „Modell“ Integration von Requirements und Implementierung



- **„Modell“ umfasst:**
 - Denk-, Sprach- und Vorgehensweise
 - zahlreiche „Artefakte“
- **Beispiele für Artefakte (VGM-abhängig):**
 - Requirements-seitig:
 - Analyse Use Cases, Use Case Beschreibungen
 - Nicht funktionale Anforderungen
 - ...
 - Implementierungs-seitig:
 - Design Use Cases
 - Klassendiagramme
 - Aktivitätsdiagramme
 - ...



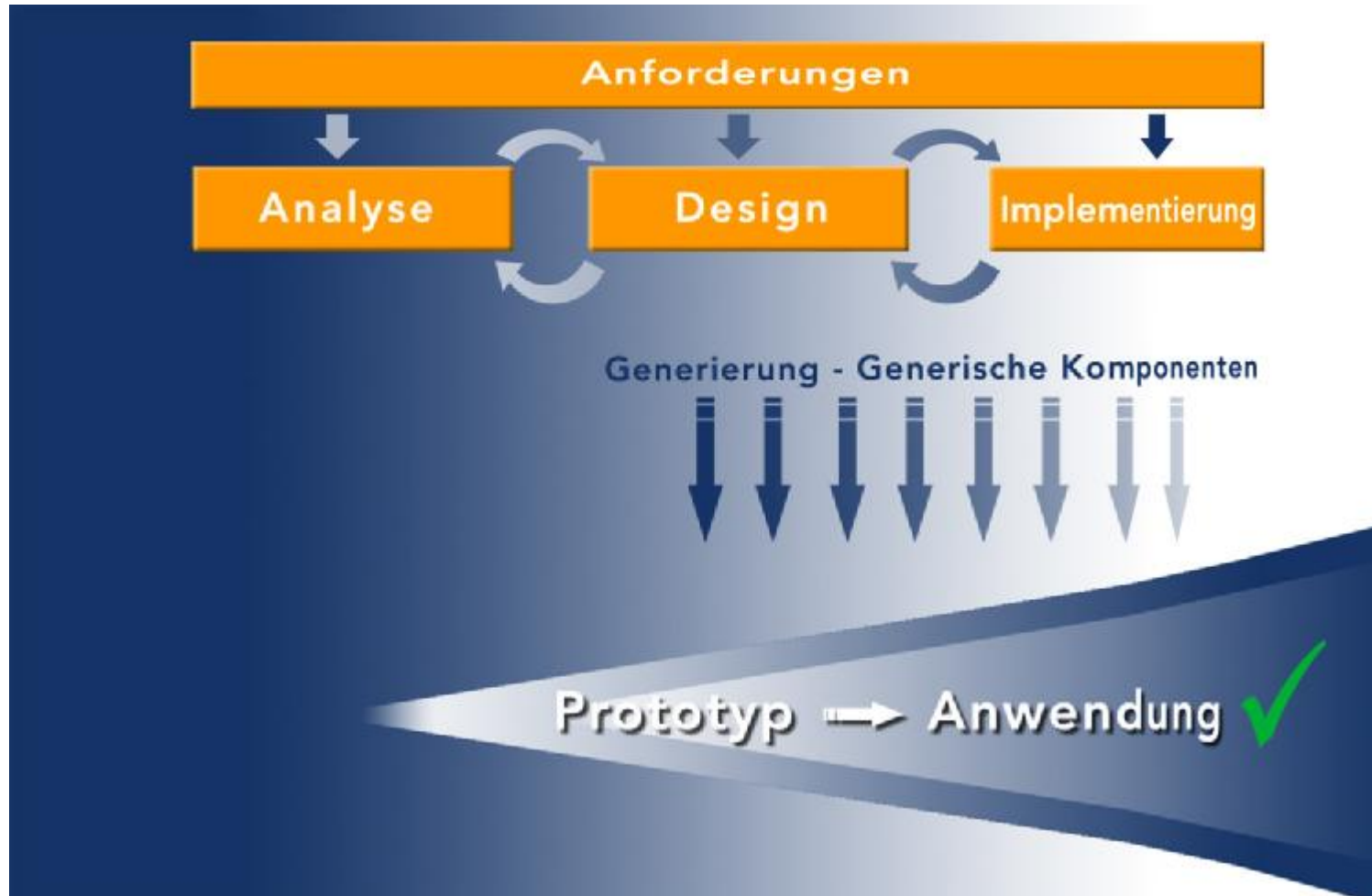
Integration von Requirements und Implementierung Am Use Case hängt's



- **Anforderungen und Implementierung dichter zusammen bringen**
- **Suche nach einem gemeinsamen Nenner**
- **Die 7+ Rollen von Use Cases...**
 - Gemeinsame Sprachbasis von Fach- und IT-Seite
 - Basis für die Festlegung funktionaler Anforderungen
 - Basis für die Projektplanung (Feature-Release Planung)
 - Basis für die Implementierung
 - Basis für das Testen
 - Basis für die Abnahme
 - Basis für das Projektcontrolling und die Fortschrittsüberwachung
- **UC Ausgangspunkt für unser MDD Vorgehen**



Der MDD-Entwicklungsprozeß mit Integration der Anforderungen



Model Driven Development



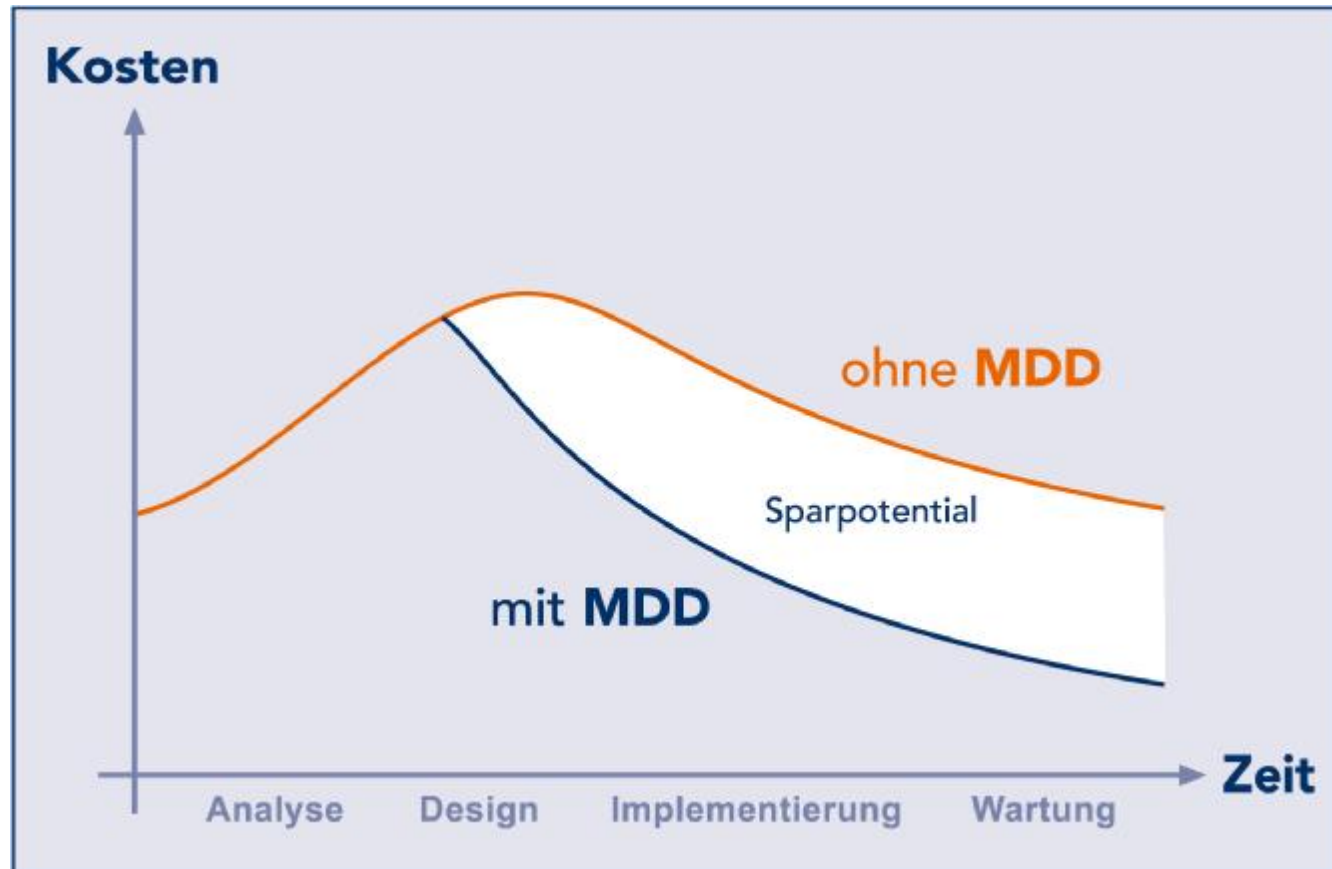
Eine vereinfachte Definition:

Model Driven Development ist gekennzeichnet durch einen **Entwicklungsprozess ausgehend vom UML Design Modell im möglichst engen Round-Trip mit der Implementierung.**

Ein wichtiges Merkmal von modellbasierter Entwicklung ist die **Definition und Ableitung wesentlicher Anwendungs-komponenten sowohl auf fachlicher als auch auf technischer Ebene direkt mit bzw. aus UML Modellen.**

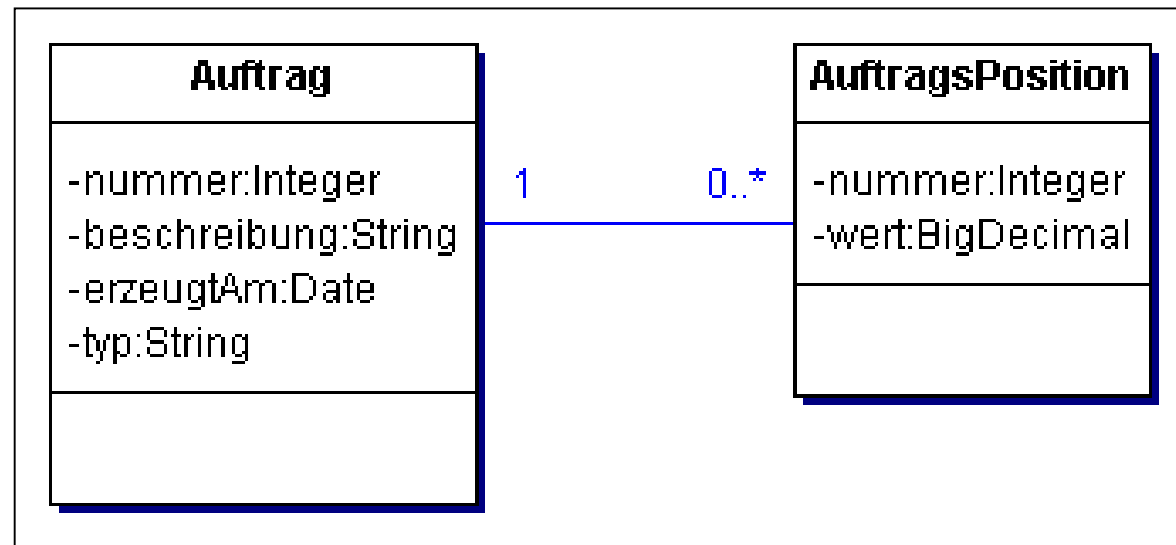


Auswirkungen von MDD auf die Projektkosten



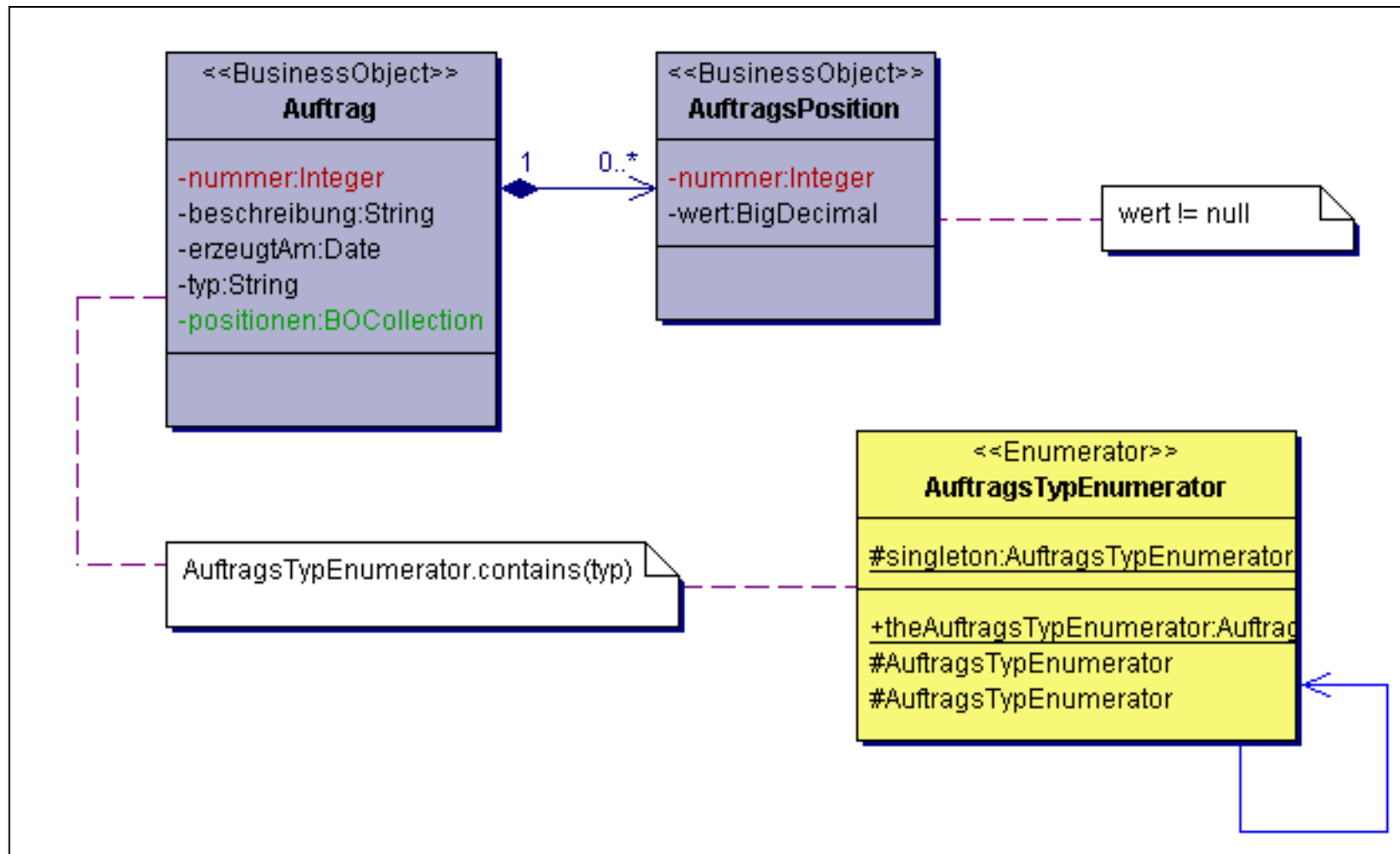
MDD – einige Prinzipien

„unvollständig“ beschriebenes fachliches



MDD – einige Prinzipien

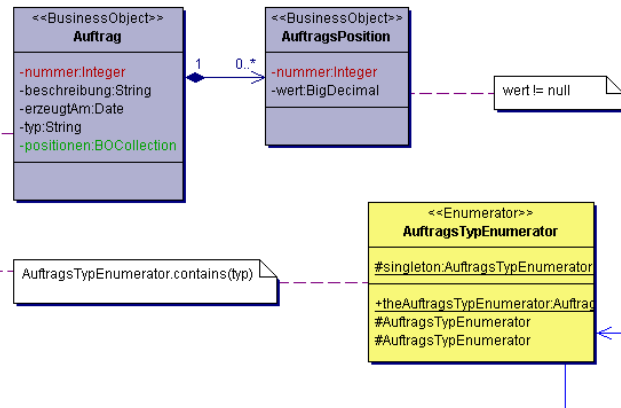
„vollständig“ beschriebenes fachliches Modell



Anwendungskomponenten mit UML-Mitteln definieren



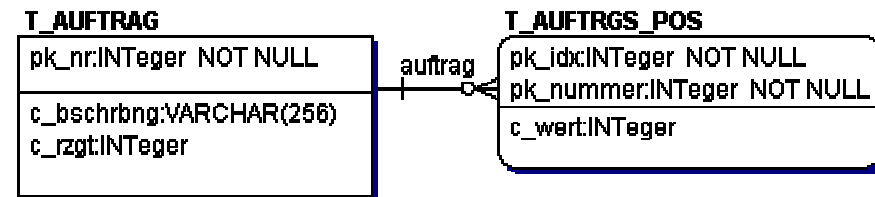
Klassendiagramm



Beispiel 1

Dateneingabelogik in der Präsentationsschicht

Persistenz auf Basis des Klassenmodells (AccessLayer)



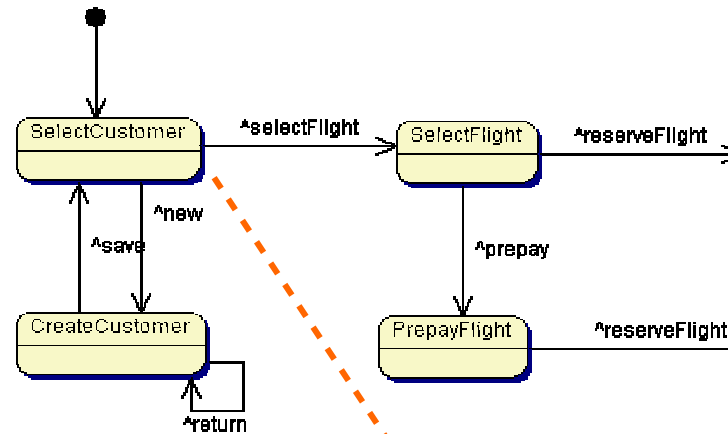
Anwendungskomponenten mit UML-Mitteln definieren



Beispiel 2

Aktivitätendiagramme

Ablaufsteuerung von Anwendungen
(transiente und persistente Workflows)



Create Customer

Customer | Reservations

First Name:

Last Name:

Address:

City:

Birth Day:

Select Customer

	First Name	Last Name	Address	City	Birth Day
<input type="radio"/>	Hans	Ottfried	Schlunz...	Oldenburg	Mon Feb 05 00:00:00 CET 1968
<input type="radio"/>	Hans	Müller	Cicerostrasse 32	10202 Berlin	Sat Jul 10 00:00:00 CEST 1982
<input type="radio"/>	Daniela	Sauter	In den Reben 4	Enzberg	Thu Apr 21 00:00:00 CET 1977
<input type="radio"/>	Peter J.	Broxton	677 Huntington Ave	Boston MA 02138	Fri Apr 17 00:00:00 CET 1970
<input type="radio"/>	Lasse	Widmann	Am Ring 12	Konstanz	Mon Apr 29 00:00:00 CEST 2002



Design und Implementierung „in Sync“

Alternative Lösungsmöglichkeiten

- Code Generierung aus dem UML-Model
 - Template Programmierung
 - Vorwärtsgenerierung
 - Erlaubt alternative „Zielplattformen“ und insbesondere alternative Zielprogrammiersprachen

- Repository-basierter Ansatz
 - Verwendung des Modells zur Laufzeit
 - Integrierter Entwicklungsprozeß
 - Schneller Round-trip zwischen Modellierung und Test
 - Dynamische Modelländerungen für hochkonfigurierbare Anwendungen



Trends in modellbasierter Entwicklung



- Formalisierung des code-generierenden Ansatzes
 - MDA Standards
 - UML Profile: Definierter Weg, UML in seinen Ausdrucksvarianten anzureichern

- UML Actions
 - Modellierung von „Verhalten“

- Sprachen integrieren sich mit dem Modell
 - Metatags in .NET-Sprachen
 - „Metadata Facility for Java“ (JSR 175)

- Verschmelzung von Modellierungswerkzeugen, Entwicklungsumgebungen und Requirements-Tools

- „Single Source“ Technologie
- JBuilder, Eclipse und Together
- UML Plug-ins für WSAD/Eclipse
 - Rational XDE
 - Together Edition for Eclipse
 - Omondo, v.a.m.
- CaliberRM, Doors

TREND: Framework und Project Development Environment



TRENDS

„Anwendungsdomäne“:

- Transaktionsorientiert
- Verteilt
- Komponenten-basiert
- Rich Clients und HTML-basiert

TRENDS

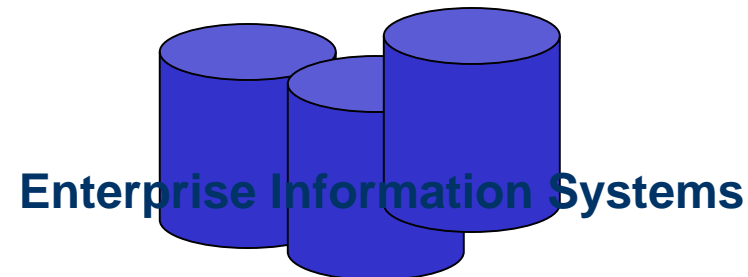
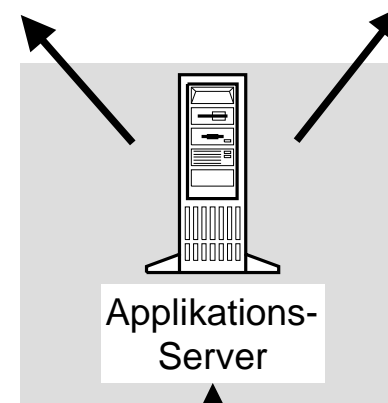
„Keyfeature“

- komplett integrierte SEU von Anforderungen bis zur Anwendung
- TREND verfolgt beide Integrationsansätze von Modell und Code
 - Repository
 - Generierung

Java Clients



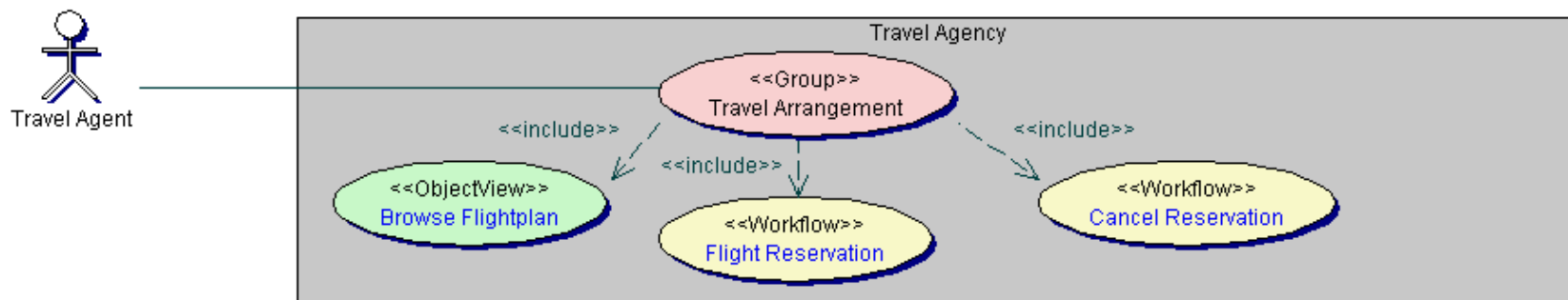
Web Clients



Demotime ... Von den Requirements bis zur Anwendung



- Demonstriert wird ein **durchgängiger Entwicklungsprozeß** von einem **Analysemodell bis hin zur lauffähigen Endanwendung**
- Unter Einsatz von **Eclipse, UML und TREND**
 - **Direktes Ausführen des Modells mit dem TREND UML-Runner**
 - **Alternative Zielplattformen am Beispiel Rich-Clients und Web-Clients**
- **Darlegung des Projektcontrollings**





Fazit

Unser Fazit aus dem Einsatz von Requirements- und Model Driven Development in diversen Projekten:

- Fach- und IT-Abteilung/Entwicklern entwickeln besseres Verständnis füreinander
- Der prototypische Ansatz mit kurzen Iterationszyklen **deckt Probleme der Anforderungs-Analyse frühzeitig auf**
- Die **Produktivität** bei der Anwendungsentwicklung wurde entscheidend **verbessert**
- Die **Wartungskosten** konnten auf Grund der (zwangsläufigen) Dokumentation **gesenkt** werden
- Projekte mit MDD können **einfacher und präziser kalkuliert** und gesteuert werden





Fragen und Kontakt

Fragen?!

**Besuchen Sie uns gerne
am Stand im Foyer**

.... und gewinnen eine Sony PlayStation

Weitere Informationen unter www.gebit.de
oder senden Sie eine Mail an trend@gebit.de

