

objectiF[®]

Model-Driven Development – die maßgeschneiderte Entwicklungstechnik für SOA

Herzlich willkommen



Benedikt Funk
microTOOL GmbH, Berlin



Die Themen

- ▲ Was steckt hinter modellgetriebener Entwicklung?
- ▲ Wie funktioniert MDD?
- ▲ Wie entwickle ich Modelltransformationen?

live

microTOOL

making IT better

Der Erfolg Ihrer
IT-Projekte
ist unser **Ziel**

Qualität

der Entwicklungsprozesse
durch

**prozessbasiertes
Projektmanagement**

mit

in-Step®

Effizienz

der Entwicklungstechnik
durch

**Automatisierung per
Model-Driven-Development**

mit

objectiF®

ABUS Pfaffenhain GmbH
achermann consulting ag
ADCURAM
Industrieberatung GmbH
Aeromaritime Systembau GmbH
AFL Europe GmbH
Allianz Deutschland AG
ASIC - Allianz Shared Infrastructure
Autostadt GmbH
AVM Computersysteme
AWD Allgemeiner Wirtschaftsdienst
Baumüller Anlagen-Systemtechnik
Bayer Business Services
Bayer Healthcare AG
Bayer Technology Services GmbH
Bayerisches Landesamt für Steuern
Bayerisches Landesamt für Umwelt
Bijou Brigitte
Biocomfort Diagnostics
BKC GmbH
Blaupunkt GmbH
Bosch Engineering GmbH
Bundesamt für den Zivildienst
Bundesamt für Migration und
Flüchtlinge
Bundesanstalt für Wasserbau
Bundesdruckerei GmbH
Bundesfinanzdirektion Nord
BWI Systeme GmbH
Carl Zeiss IMT GmbH
Cirquent GmbH
Collogia AG
CompuGROUP Software GmbH
Computer Zentrum Strausberg

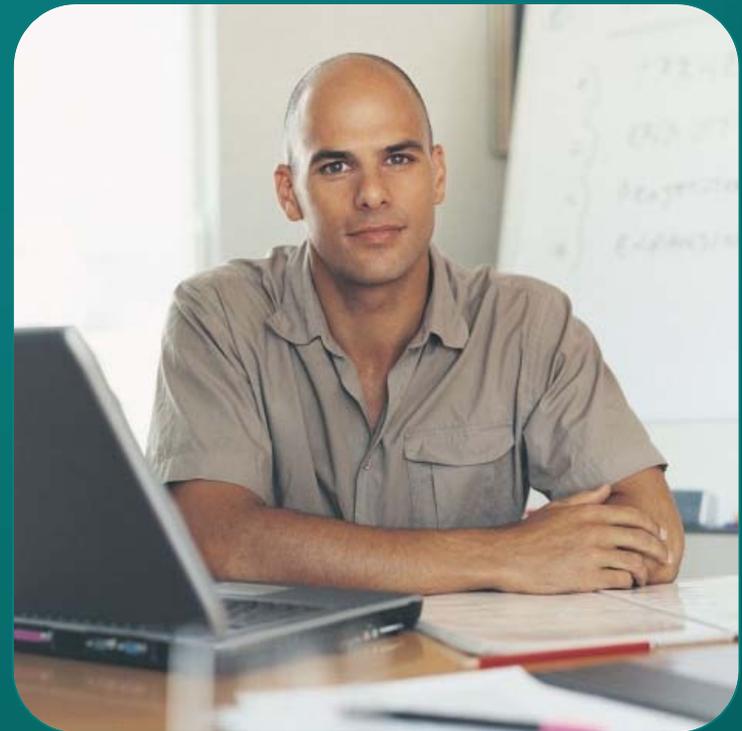
COMSOFT GmbH
Cooper Power Tools
CORa-IT GmbH
CSC Deutschland Solutions GmbH
Currenta GmbH & Co. OHG
Daimler AG
Dallmeier electronic GmbH & Co. KG
Datenzentrale Baden-Württemberg
DATEV eG
Deutsche Bank AG
Deutsche Flugsicherung GmbH - DFS
Deutsche Post Direkt GmbH
Deutsche Rentenversicherung Bund
Deutsche Telekom AG
Deutsches Zentrum für Luft- &
Raumfahrt
DOMCURA AG
Dr. Männer und Partner
DRS Systemberatung GmbH
Duales System Deutschland GmbH
EADS Deutschland GmbH
eBankingServices Nord GmbH
Eckelmann AG
ECOMAL Deutschland GmbH
Eidgenössisches Finanzdepartment
Elektrobit Automotive GmbH
EPLAN Software & Service GmbH & Co.
EWE TEL GmbH
FEV Motorentchnik GmbH
Fujitsu Technology Solutions GmbH
GAF AG
GEA Wiegand GmbH
GfK Retail and Technolgy GmbH
Global Draw Austria GmbH

HOPPE AG
IAV GmbH
icubic AG
Insight Technology Solutions
Internetdienste der Stadt Zürich
ITERGO GmbH
JobScout24.de
Jomo Großhandel
Kapsch CarrierCom AG
Kassenärztliche Vereinigung
Kassenzahnärztliche Vereinigung
Knowledge Department
Kuhnke Support GmbH & Co. KG
Lachmann & Rink GmbH
Landis+Gyr AG
lastminute.com GmbH
Layer 2 GmbH
lesswire AG
Logica Deutschland
Marineamt
MEN Mikro Elektronik GmbH
Mettler & Fuchs AG
MicroContact AG
MSC Vertriebs GmbH
msgGILLARDON AG
ND SatCom Defence GmbH
North IT-Service
Nuon Deutschland GmbH
OPTIGEM GmbH
Plan International Deutschland e.V.
Process Relations GmbH
Proximity Technology GmbH
Prüftechnik AG
Raytheon Anschutz GmbH

RENK AG
Robert Bosch GmbH
Robert Koch-Institut
robomotion GmbH
RSN Risk Solution Network AG
Sächsisches Staatsministerium
Salzgitter Flachstahl GmbH
SHD Einzelhandelssoftware GmbH
SHI Elektronische Medien GmbH
Smurfit Kappa GmbH
SOPTIM AG
Staatsbetrieb Sächsische Informatik
Statistisches Bundesamt
Statistisches Landesamt Sachsen
STP Informationstechnologie AG
Sumitomo Electric Bordnetze
GmbH
tdb Software Service GmbH
Thales Group
Thomas Cook AG
TI Automotive Technology Center
Trianel GmbH
TRW Automotive Safety Systems
T-Systems Enterprise Services
GmbH
Ultratronik GmbH
Unternehmensberatung H&D
Vattenfall Europe
viadico AG
Wasser- und Schifffahrtsamt
Wave Developments
Werum Software & Systems AG
Westdeutsche Landesbank
Xavo AG

Unsere Kunden

Wann würden Sie gern
modellieren



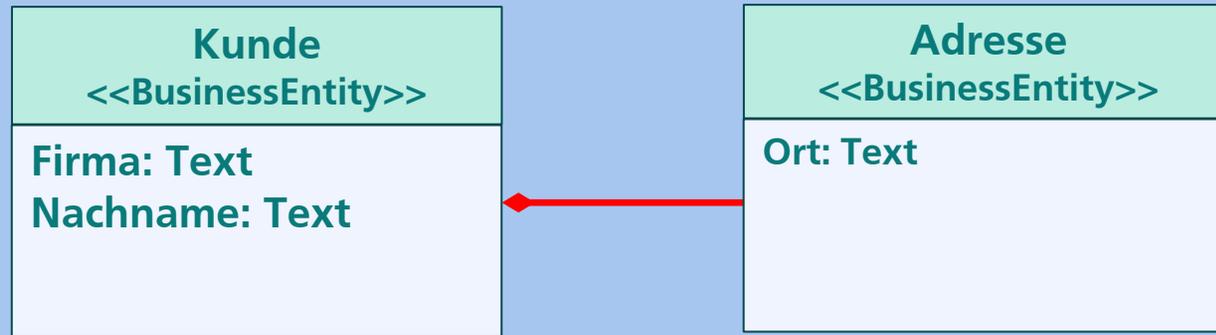
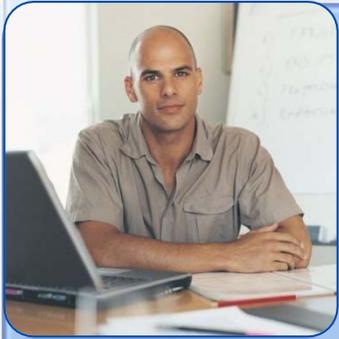
?

Wenn sicher wäre, dass:

- ▲ Ihre Modelle über den Softwarelebenszyklus aktuell sind
- ▲ Ihre Modelle und Ihr Code zueinander passen
- ▲ Sie aus Modellen etwas generieren können

Modellgetriebene Entwicklung (MDD) mit *objectiF*[®]

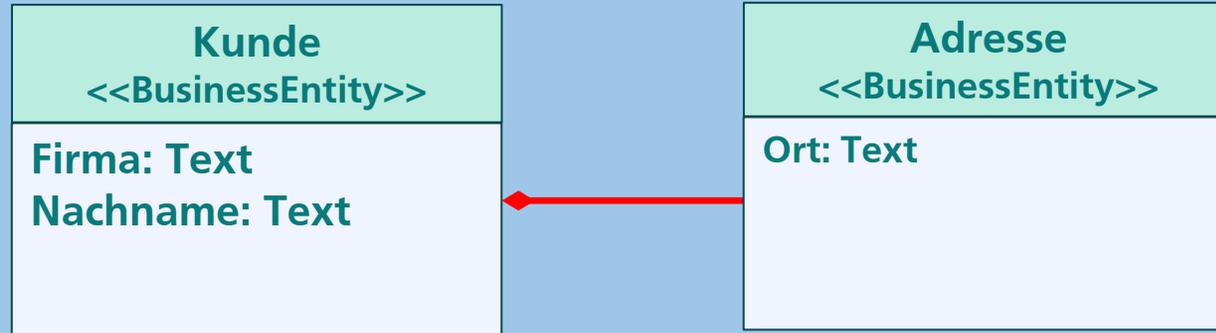
MDD setzt beim fachlichen Modell in UML/BPMN auf:



- ▲ angepasst an die fachliche Domäne
- ▲ frei von Technologie
- ▲ durch den Softwareentwickler/Facharchitekt erstellt

Automatisch...

Fachliches Modell

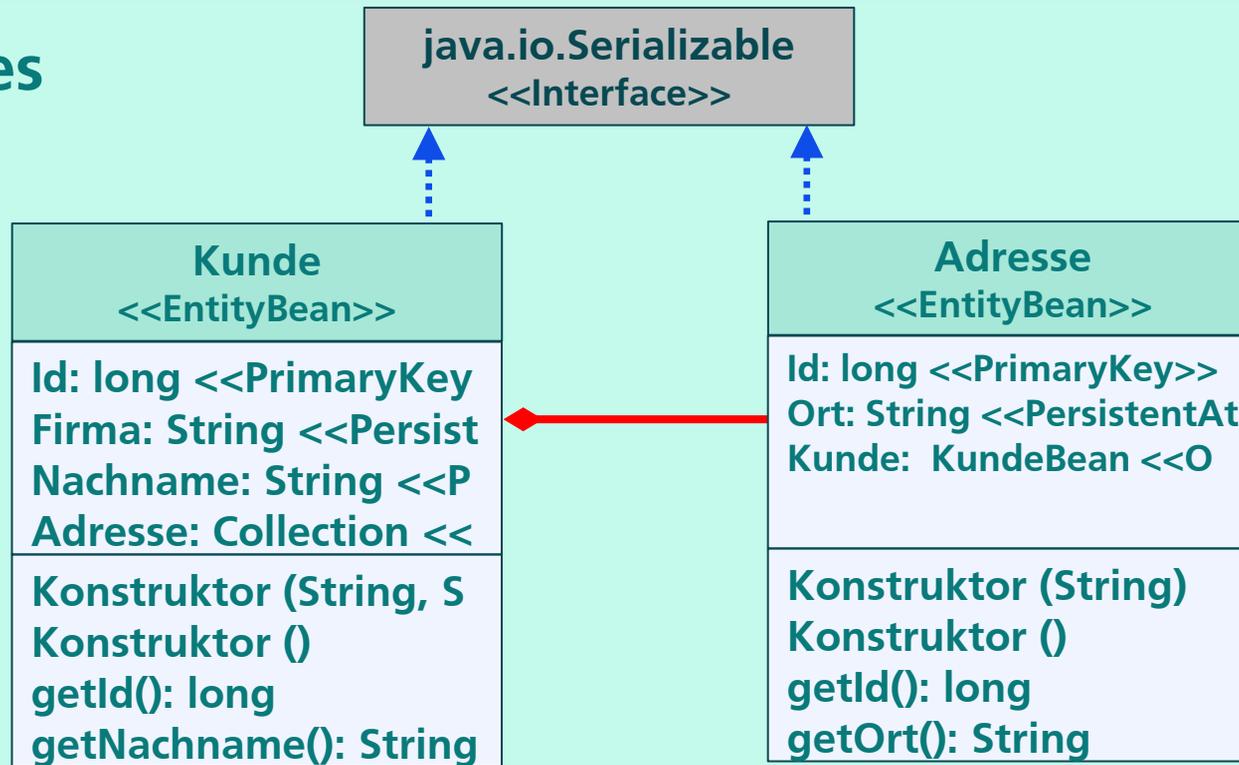


Modelltransformation



Ergebnis: das technische Modell in UML und Source-Code

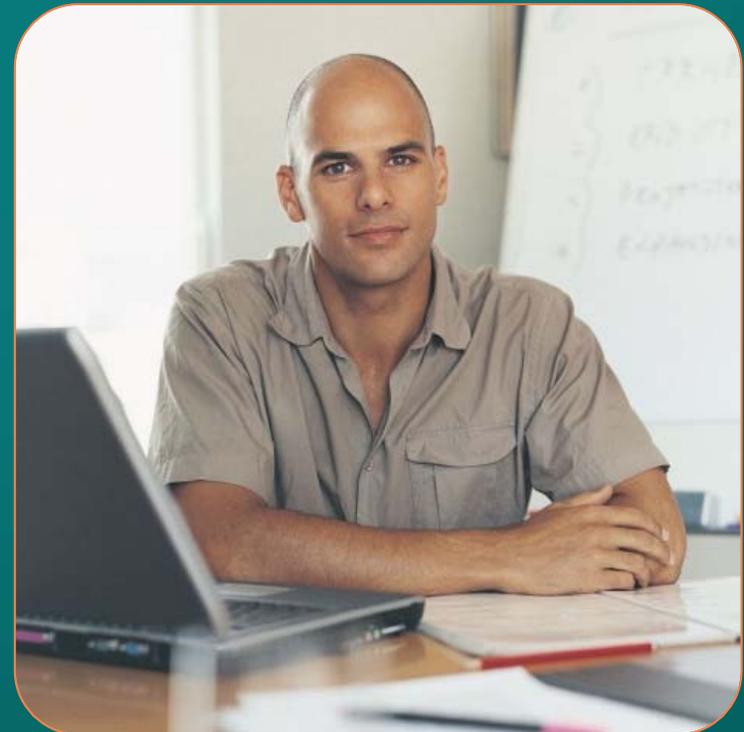
Technisches Modell



- ▲ spiegelt Architektur der Software wider
- ▲ enthält viel generierten Code
- ▲ ist Ausgangspunkt der Implementierung der Business Logik

Unser fachliches Modell
ändert sich

aber **oft**



Modelltransformationen werden iterativ angewendet

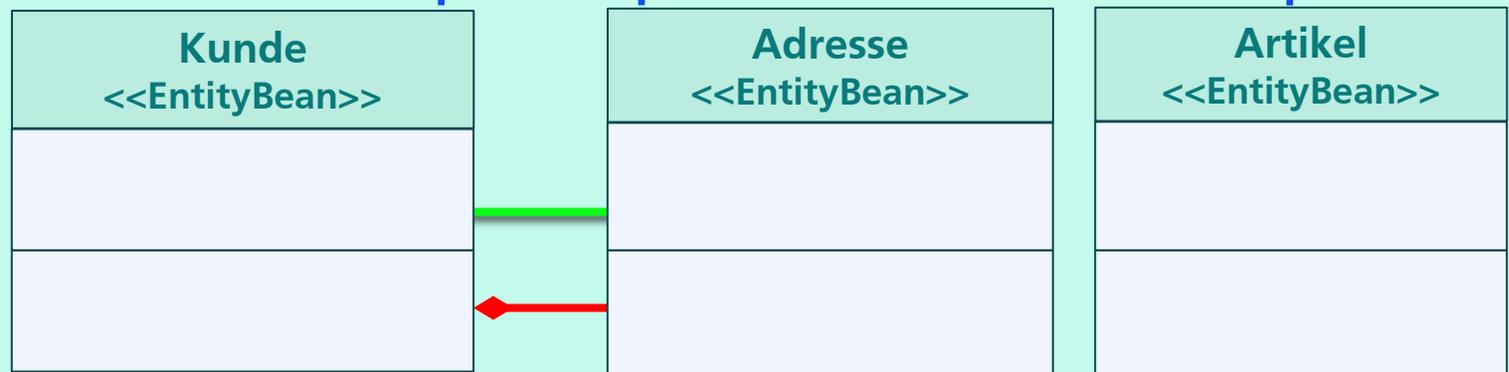
Fachliches Modell



Modelltransformation (Aktualisierung)

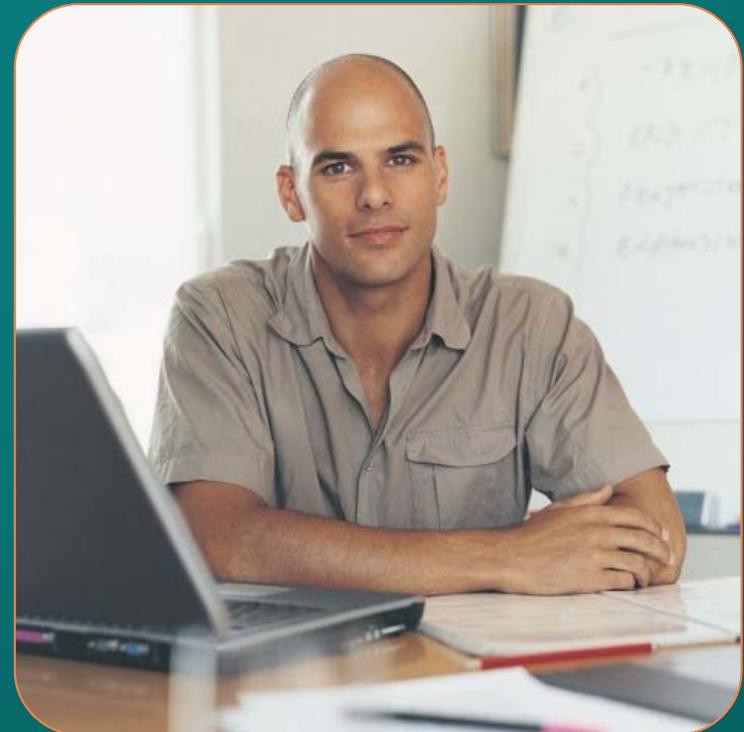


Technisches Modell



Kann ich **nachvollziehen**,
was woraus
entstanden ist?

?



Klar...

Fachliches Modell

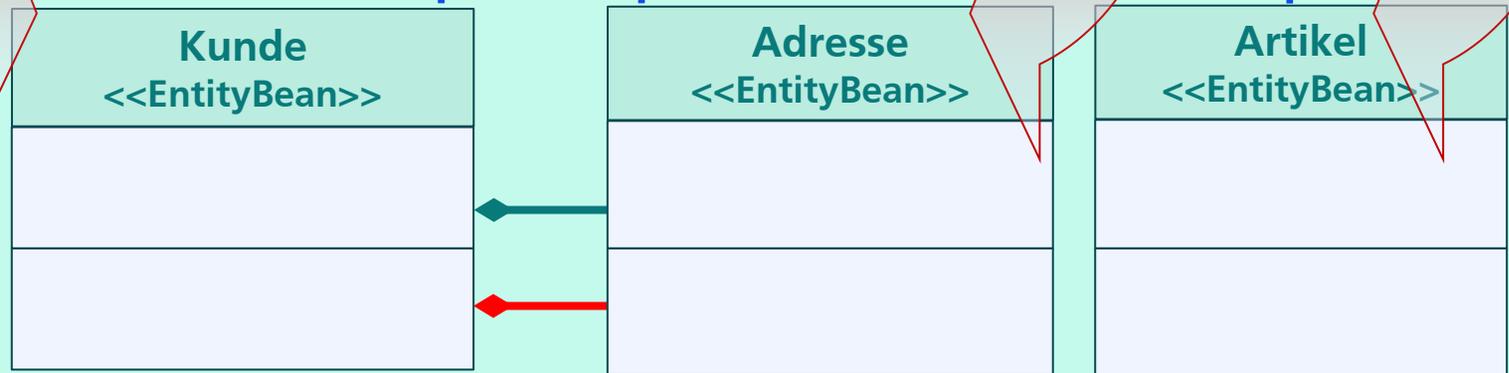


Modelltransformation (Aktualisierung)



Navigieren

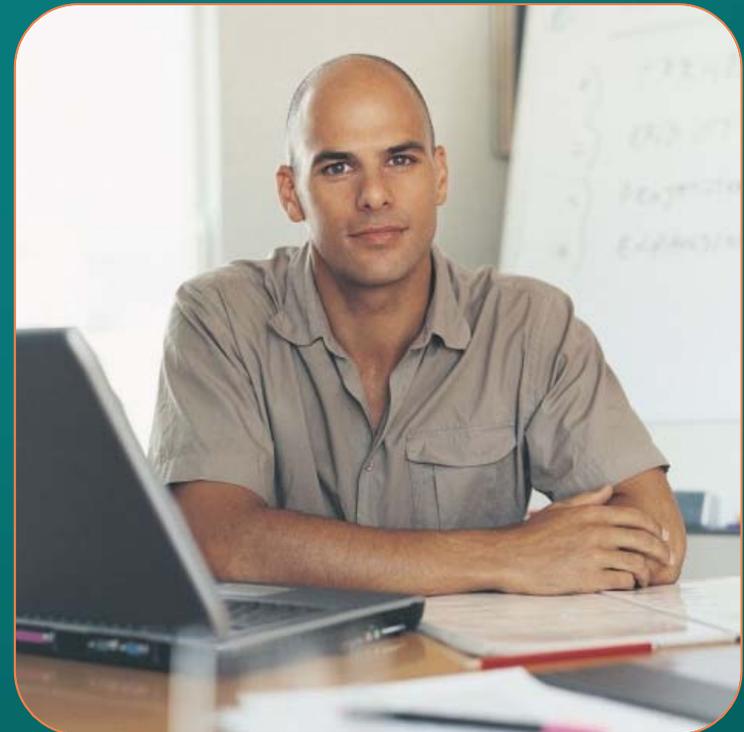
Technisches Modell



Und **was** passiert dabei
mit meinem

Code

?



Nur Gutes...

Element im fachlichen Modell



Element im technischen Modell



Generierte

Editierbare

Generierte Code-Blöcke

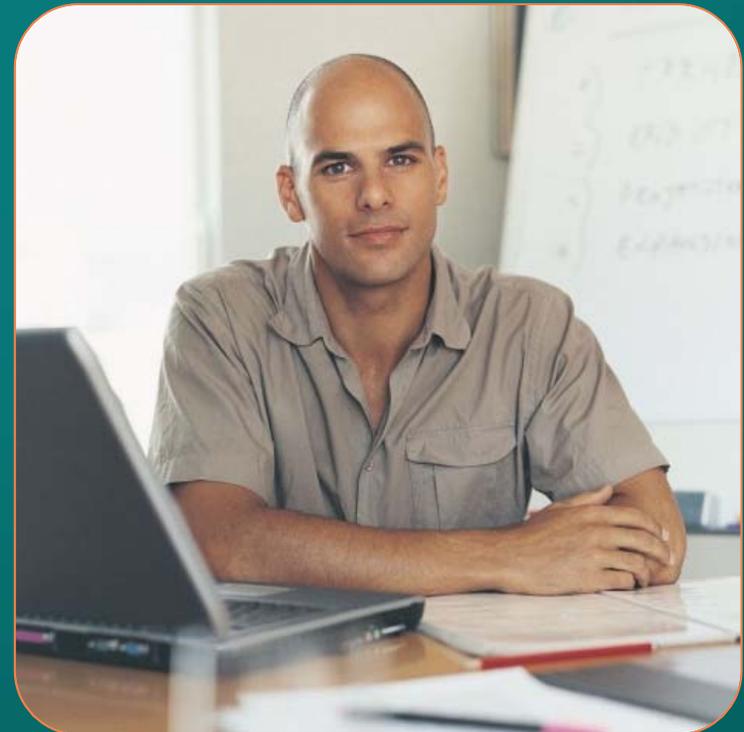
Editierbare Code-Blöcke

Generierte Code-Blöcke

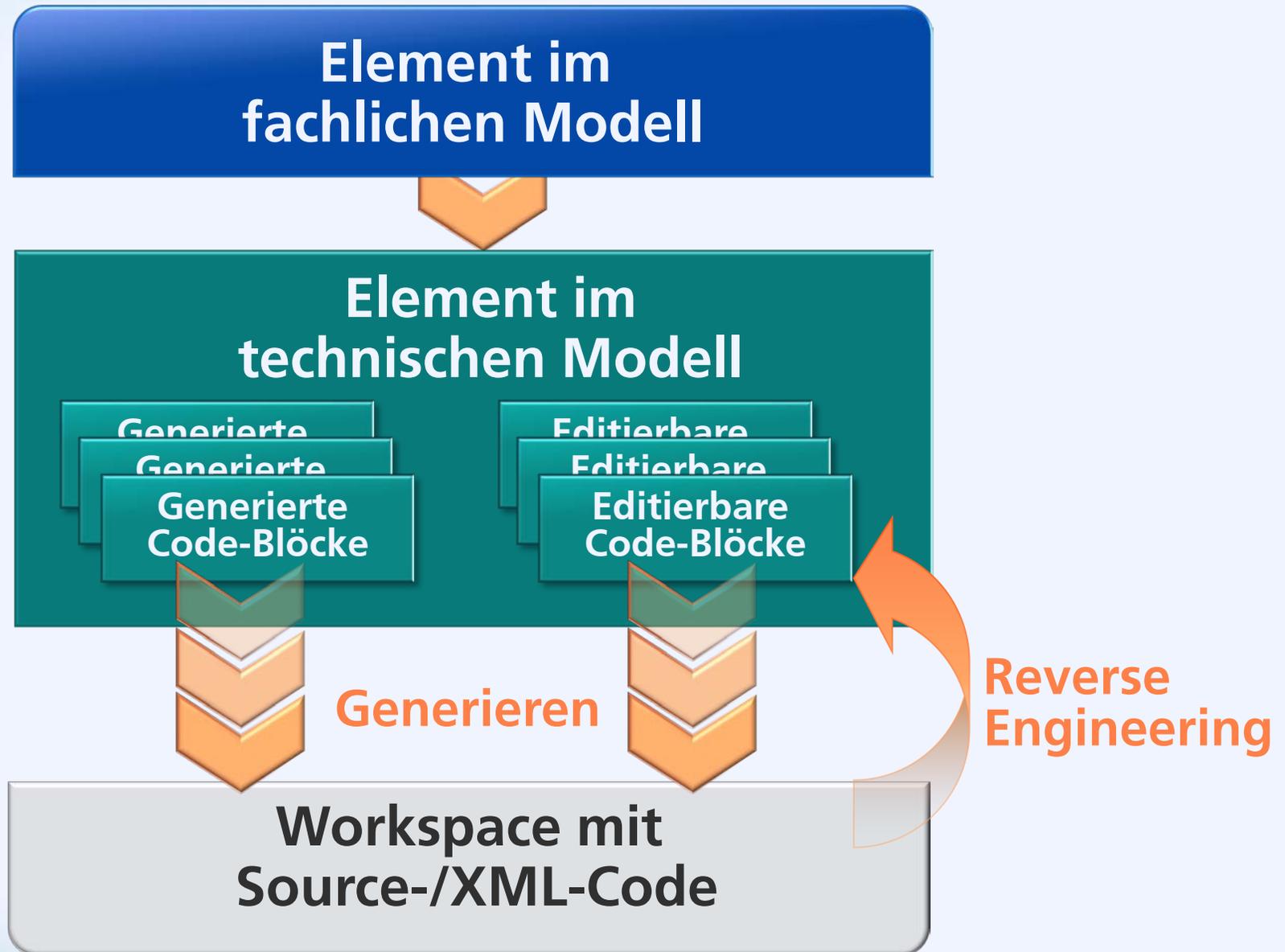
Editierbare Code-Blöcke



Aber ich **tippe** lieber
in **Eclipse**



Tun Sie's...



Was bringt Ihnen MDD mit *objectiF*[®]?

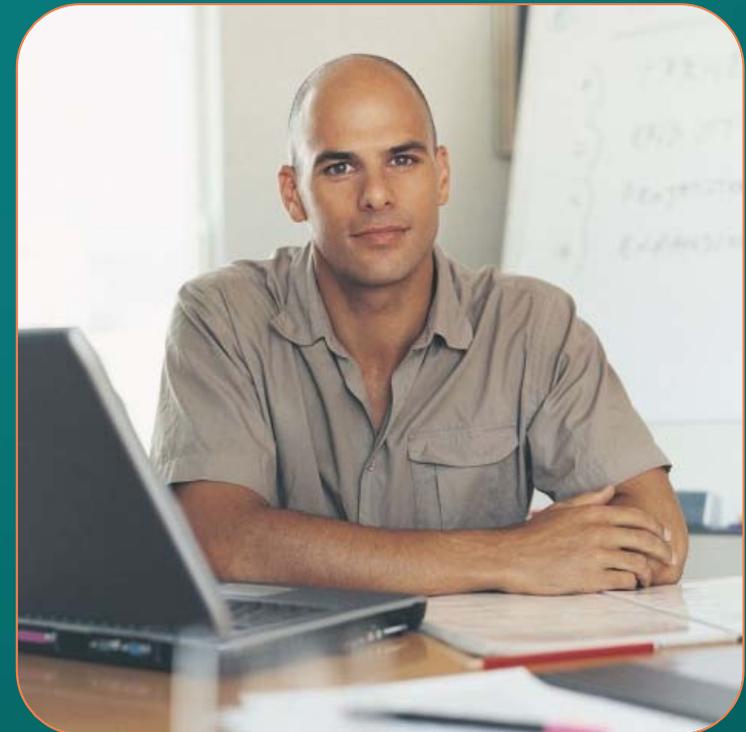
- ▲ Fachliche Modelle bleiben aktuell
- ▲ Technische Architektur und Source-Code werden automatisch erzeugt

Mehr Zeit

Mehr Qualität

Investitionsschutz

Was muss ich **tun**, um
modellgetrieben
zu entwickeln



A Ich verwende das, was *objectiF*[®]
mitbringt:
... **Modelltransformationen**
für gängige
Java- und .NET-Technologien
oder

B Ich erstelle **eigene Modell-**
transformationen mit der in *objectiF*[®]
integrierten Technik

Wir brauchen dazu 3 Dinge...

**eine fachliche Architektur und
Ausdrucksmittel zur fachlichen Modellierung**

**Regeln zur Umsetzung fachlicher Modelle
in technische Modelle und Source-Code**



**eine technische Architektur und Ausdrucksmittel zur
technischen Modellierung und Implementierung**

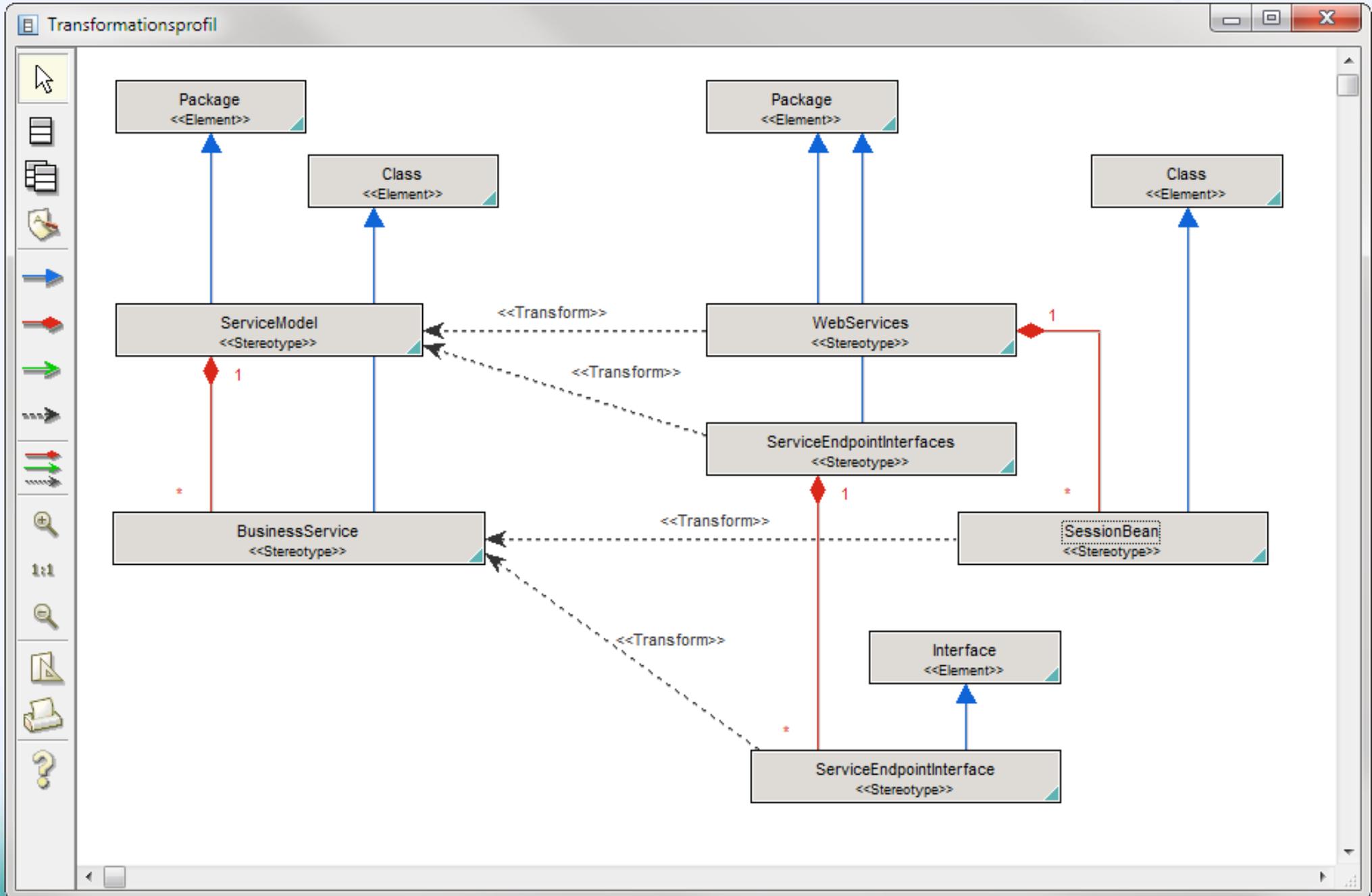
Also erzeugen wir mit *objectiF*[®] ...

eine fachliche Architektur und
ein Metamodell für fachliche Modelle

eine Transformation, die fachliche in
technische Modelle und Source-Code überführt

eine technische Architektur und
ein Metamodell für technische Modelle

So sieht das mit **objectiF**® aus...



Konkret: Wir entwickeln modellgetrieben

- ▲ Transformation mit der UML spezifizieren
- ▲ Technisches Modell und Source-Code der Transformation mit *objectiF*[®] erzeugen
- ▲ Transformationslogik in C# vervollständigen

Eigene Modelltransformationen in *objectiF*[®]

- ▲ Sie bestimmen die Transformationslogik

Mehr Zeit

Mehr Qualität

Investitionsschutz

Sie sehen jetzt **live** mit **objectiF**[®]

am Beispiel der Erstellung
einer **SOA-Anwendung**:

- ▲ wie eine Transformation das technische Modell und Java-Code für EJB 3.0 erzeugt
- ▲ wie MDD iterativ angewandt wird
- ▲ wie die Transformationslogik angepasst wird

Fazit:

Einfache fachliche Modellierung

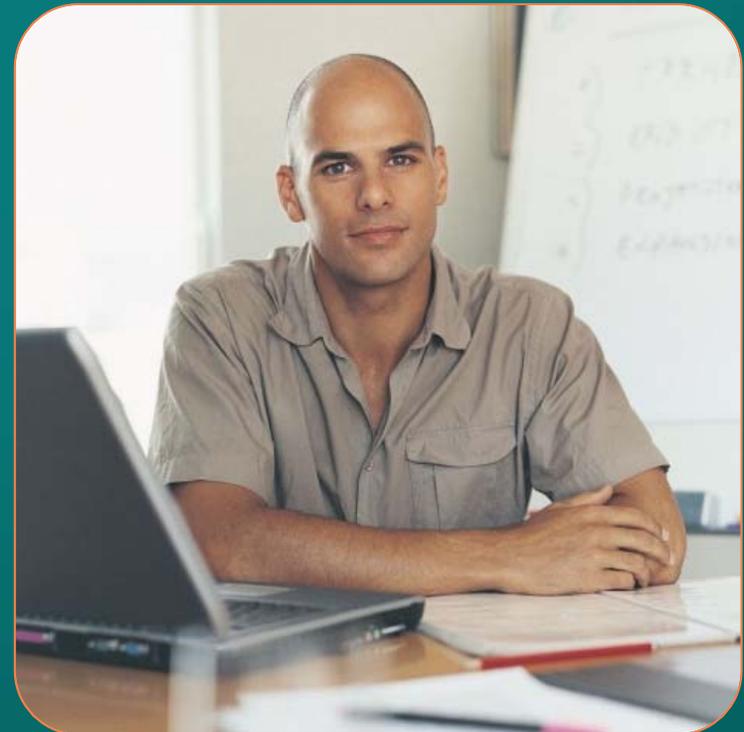
Einhaltung der Architekturstandards

**Vollständige und automatische
Implementierung**

**Einfache Anpassung der
Transformationslogik**

Welche Transformationen **bringt** *objectiF*[®] **mit**

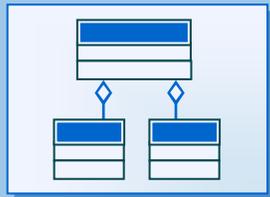
?



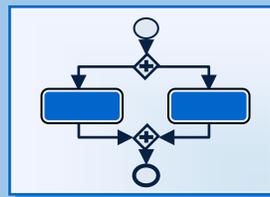
Diese Modelltransformationen bietet **objectiF**[®]

Fachlicher Modellierungsrahmen

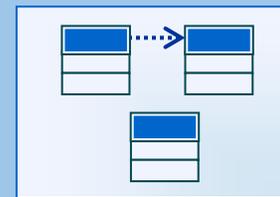
Präsentationsmodell



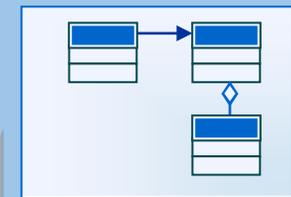
Prozessmodell



Service-Modell



Entity-Modell



Geschäftsprozess zu BPEL

Geschäftsprozess zu WSDL

Business Services zu Session Beans

Business Services zu WSDL

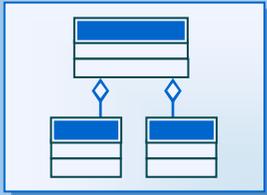
Business Message zu Java

Business Message zu XSD

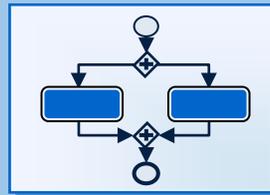
Diese Modelltransformationen bietet **objectiF**[®]

Fachlicher Modellierungsrahmen

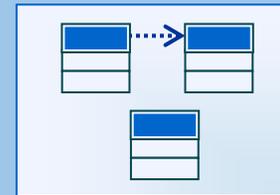
Präsentationsmodell



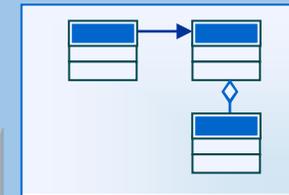
Prozessmodell



Service-Modell



Entity-Modell



Business Entity zu Entity Bean

Business Entity zu NHibernate

Business View zu ViewData und ViewDataServices

WSDL zu Business Service

XSD zu Business Message

Profil zu Modelltransformation (Metatransformation)

**Vielen Dank
für Ihre Aufmerksamkeit!**



Mehr zu

objectiF[®]

und **in-Step**[®]

Stand 12