

# Architecturecentric MDSD

How to get rid of that  
Monkey Code?



# Agenda

- MDA/MDSD Einführung
- MDSD Ziele und Mehrwert
- AC-MDSD mit oAW
- AC-MDSD Demo
- Fazit



# MDA/MDSD Einführung

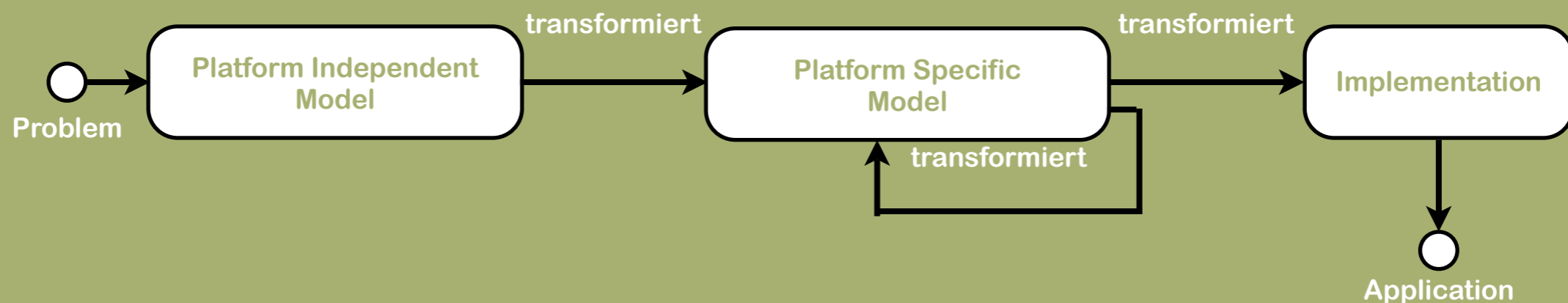
Standardisierungsinitiative und pragmatische  
Umsetzbarkeit

# Rückblick

- CASE und 4GLs
- OO-Development
- UML Standard
- MDA/MDSD

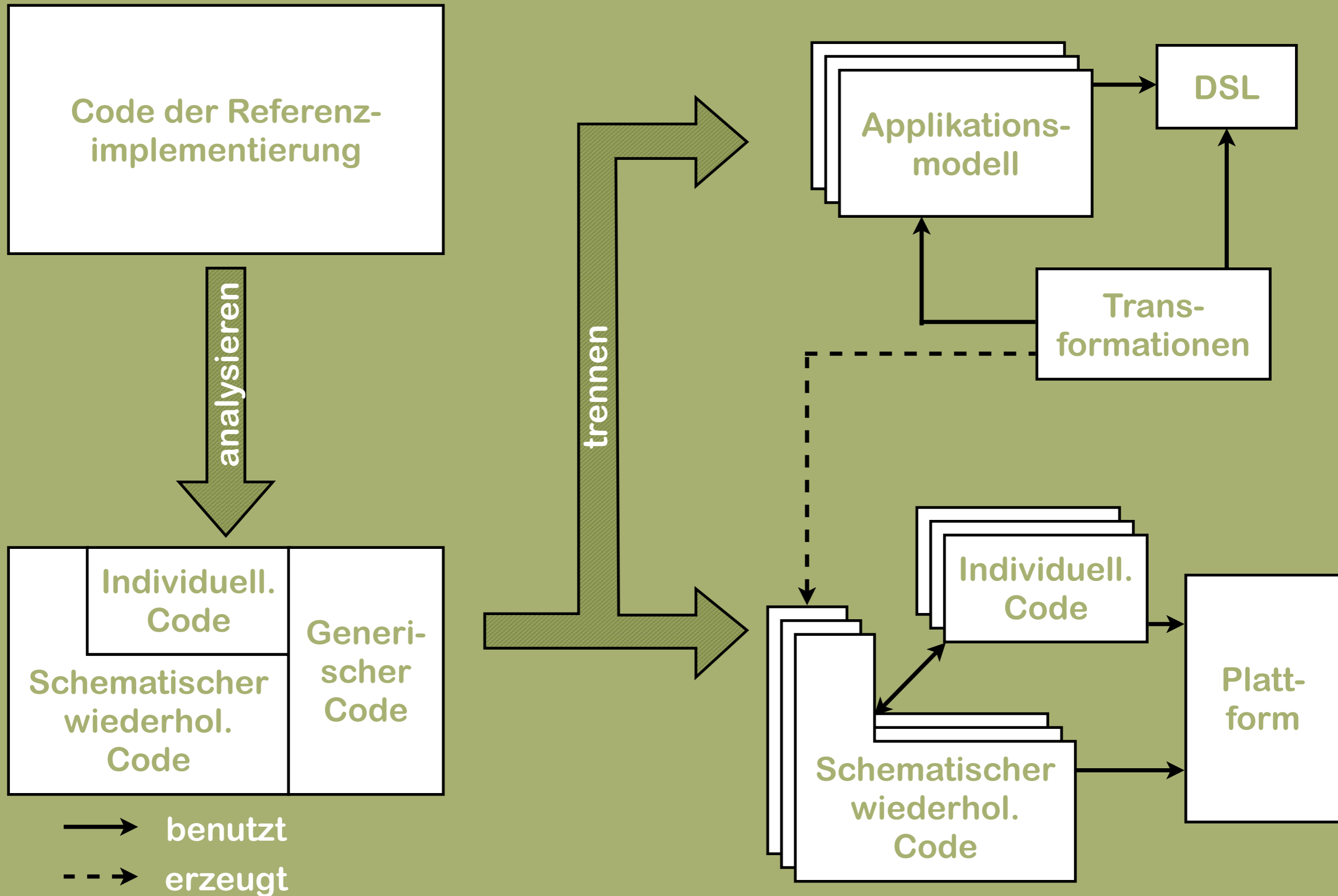
# MDA

- Standardisierungsinitiative der OMG
- PIM → PSM (Modell zu Modell Transform.)
- Interoperabilität von MDA-Tools
- Metamodelle



# MDSD

- pragmatische Interpretation der MDA
- Modell zu Code Transformation
- Referenzimplementierung als Basis
- DSL, Plattform, Generierung



# AC - MDSD

- Domäne „Softwarearchitektur“
- Infrastrukturcode wird generiert (Monkey Code)
- Fachlogik wird ausgeklammert

MDA

MDSD

AC-MDSD



# MDSD Ziele und Mehrwert

„Der Intellekt hat ein scharfes Auge für  
Methoden und Werkzeuge,  
aber er ist blind gegen Ziele und Werte.“

Albert Einstein

# MDSD Ziele

- Steigerung der Entwicklungsgeschwindigkeit
- Höhere Softwarequalität
- Höherer Grad an Wiederverwendbarkeit
- Produktiveres Umfeld schaffen
- Handhabbarkeit von Technologiewandel

# MDSD Mehrwert

- Dokumentation ist Sourcecode
- Architektur Vertrag
- Multiplattformssysteme

# AC - MDSD mit oAW

# Geschichte von OpenArchitectureWare

**1999**

Framework  
b+m Informatik

**2000**

Produkt  
b+mAW

**-2003**

Version 2.0  
Komm. Nutzung

**2003**

OpenSource  
oAW

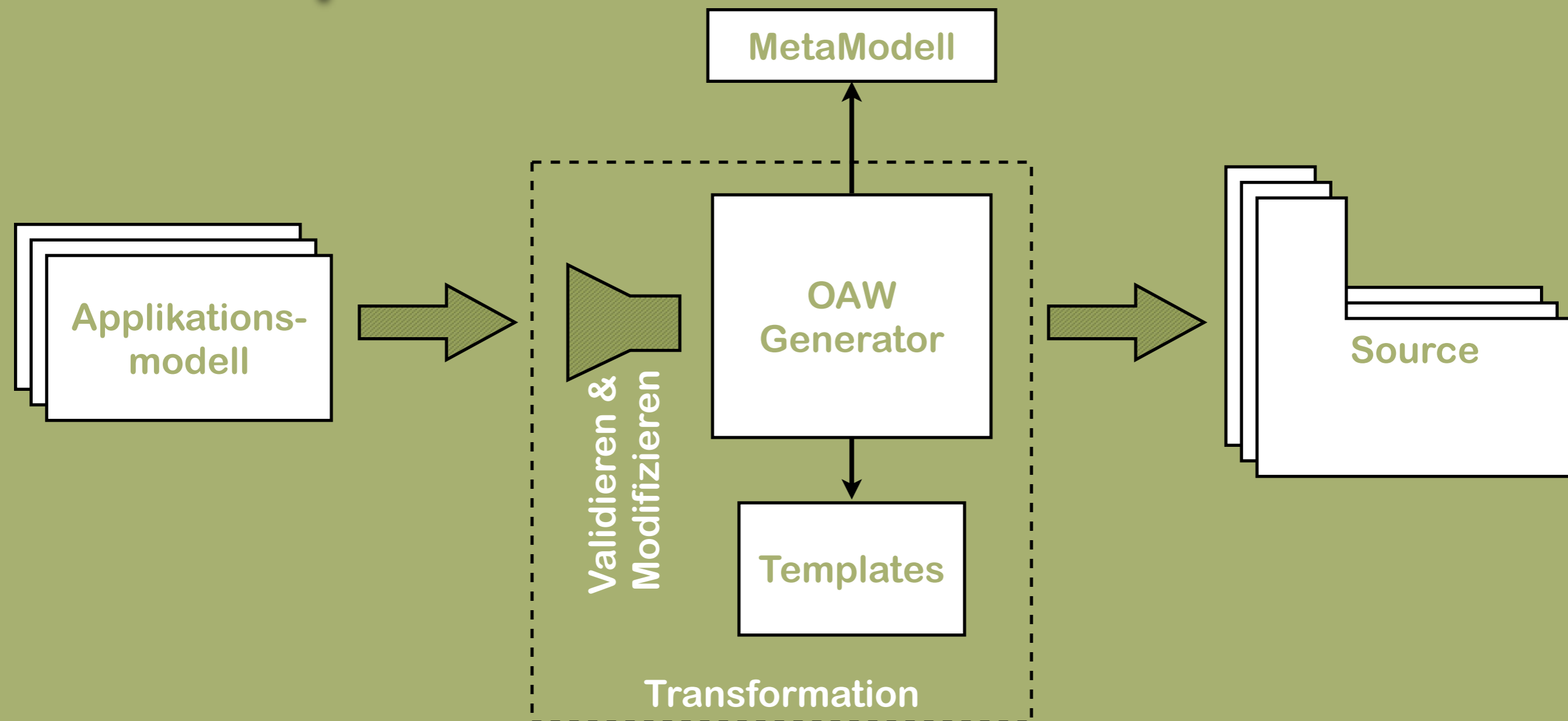
**-2005**

Version 3.0  
oAW

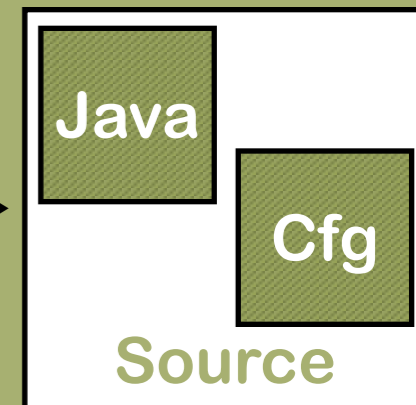
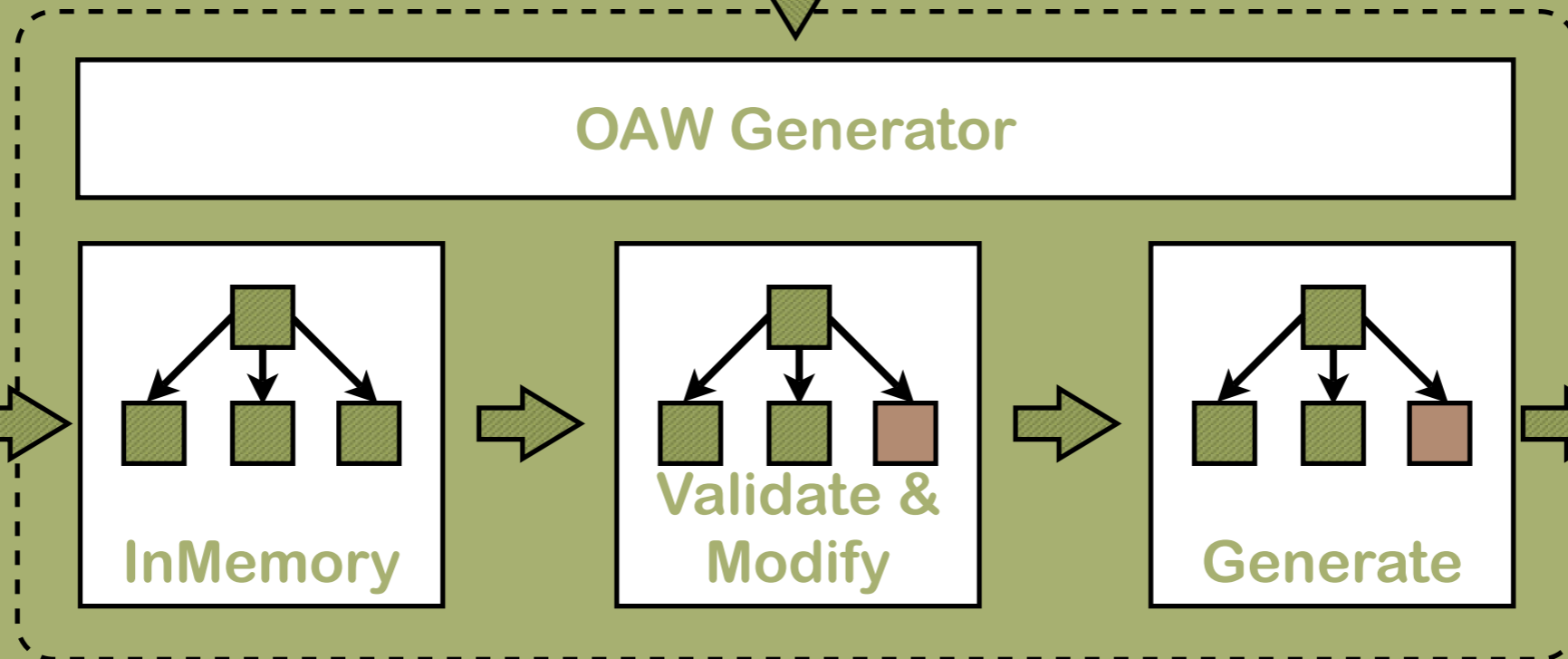
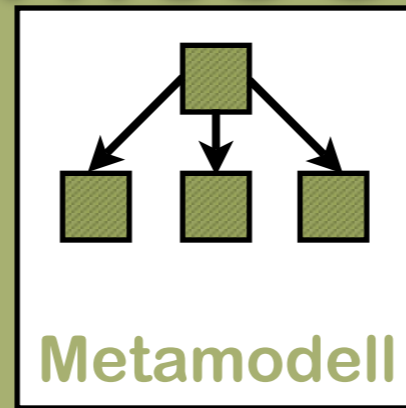
**-2006**

Version 4.1  
oAW

# MDSD mit openArchitectureWare



# MDSD mit openArchitectureWare





# MDSD Blueprint

- Architektur/Design festlegen
- Referenzimplementierung erstellen
- Referenzimpl. analysieren
- „Schemat. wiederhol. Code“ finden und in einem Metamodell definieren
- Schablonen zum Metamodell erstellen
- Applikationsmodell erstellen
- Code mit Generator und Applikationsmodell erzeugen

# DEMO

# Fazit

# AC - MDSD

**Faster, Better, Cheaper!**

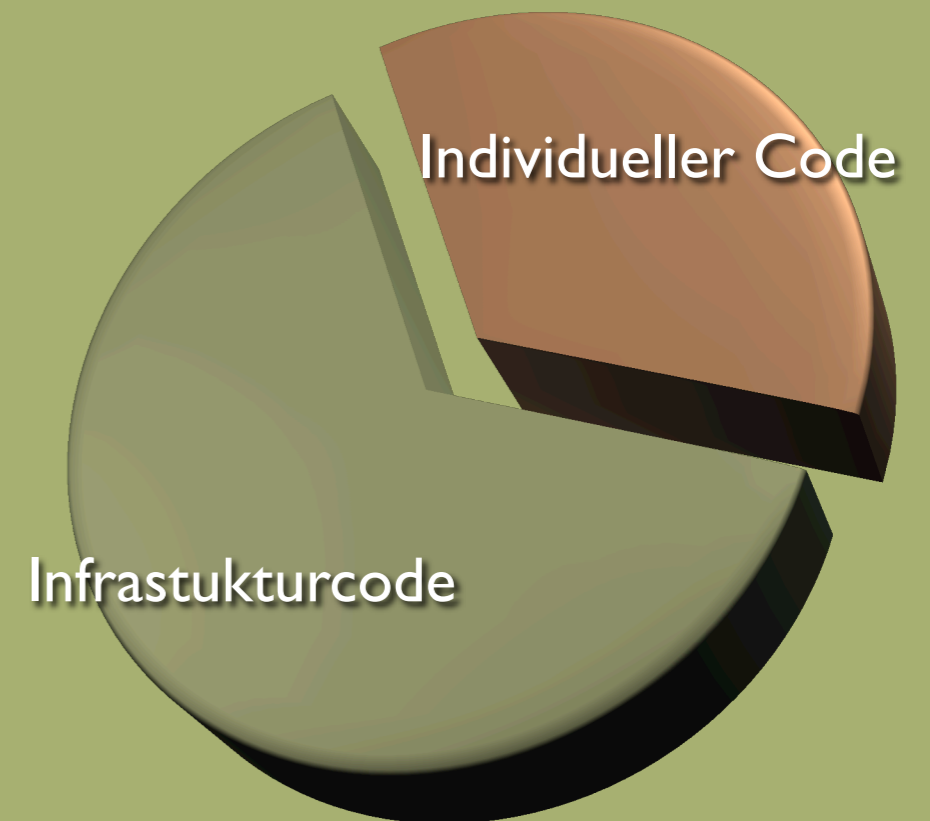


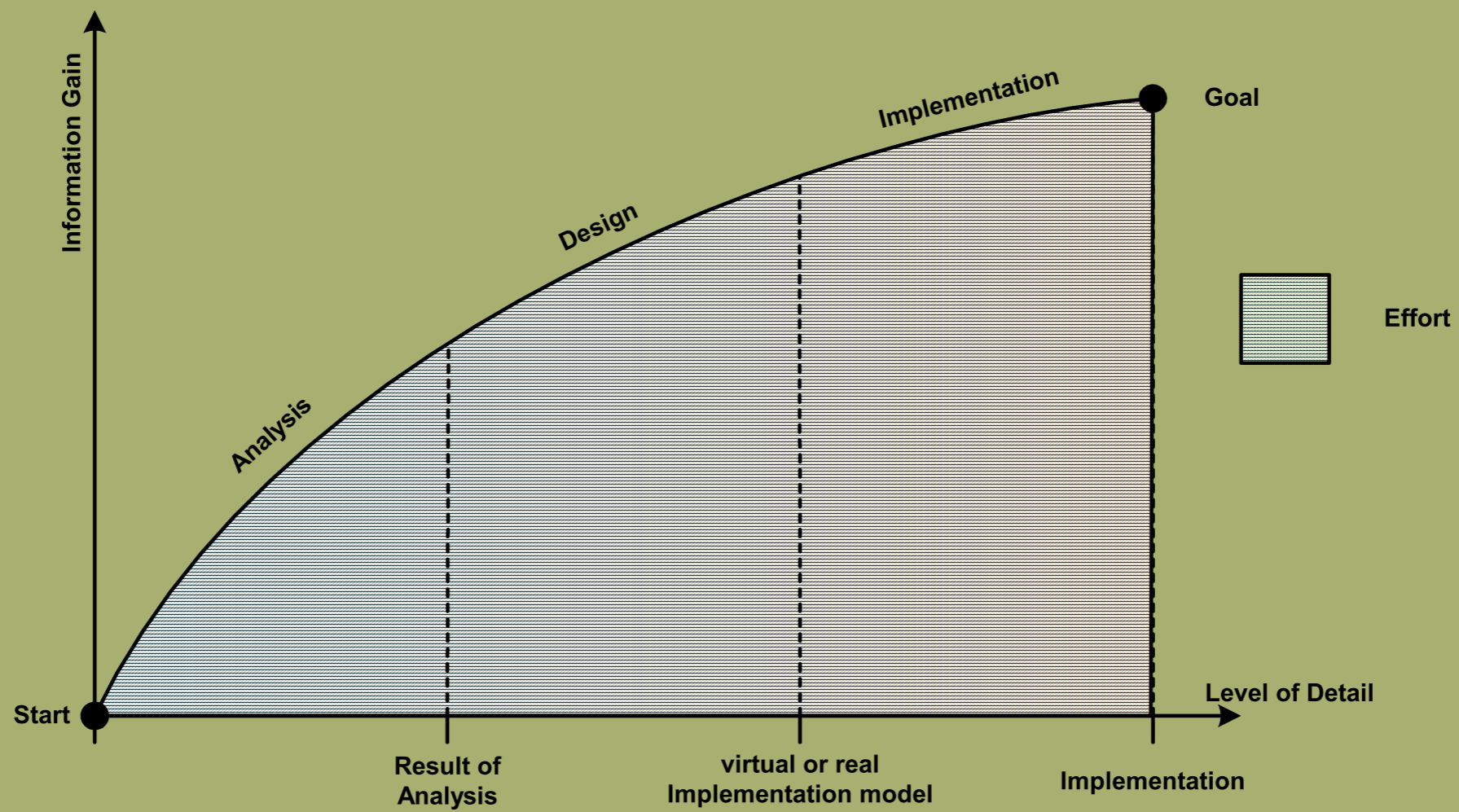
# Traditionelle Software- entwicklung vs. MDSD

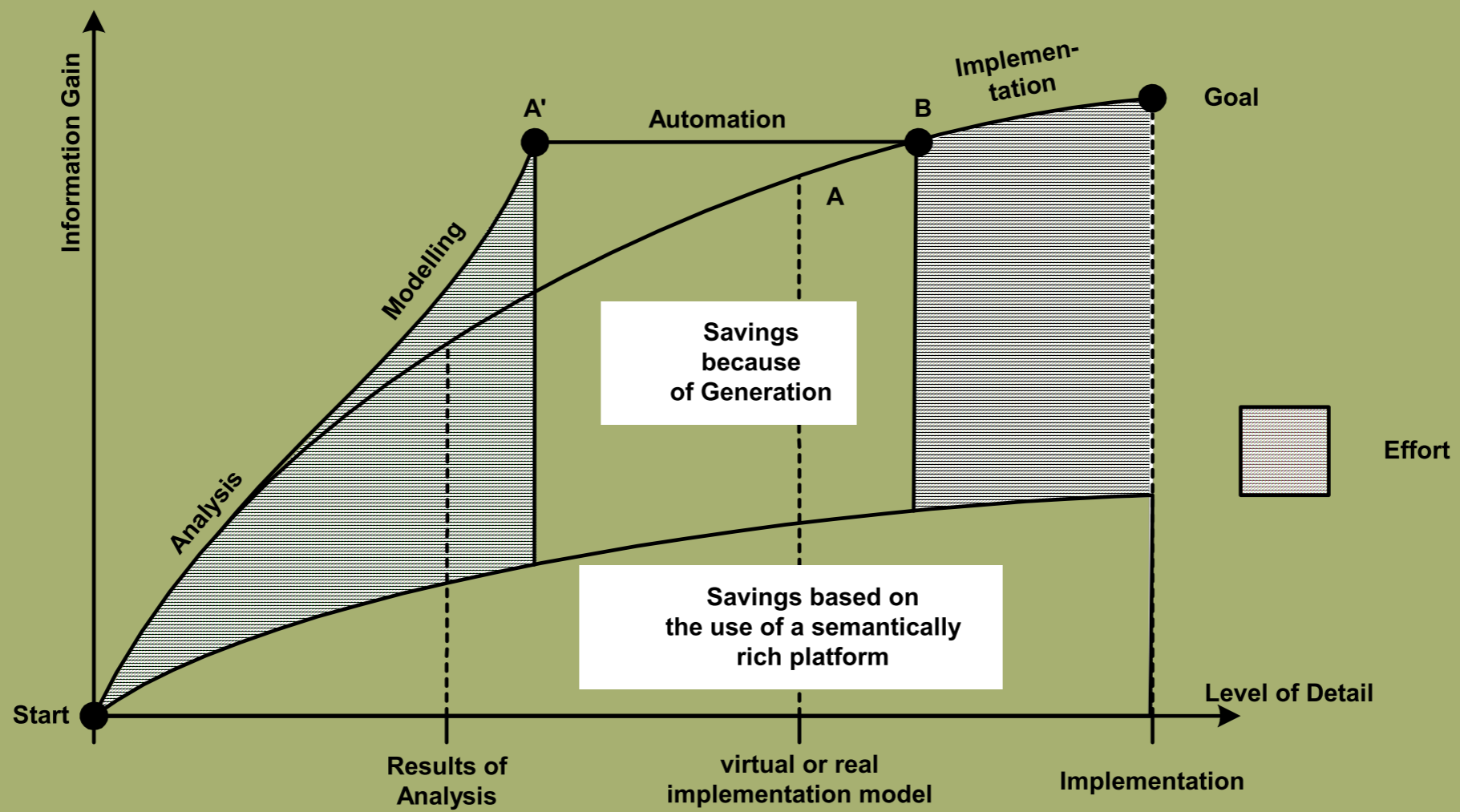
- DDL anpassen
  - Entity erweitern
  - Hibernate-Config
  - DAO
  - ....
  - Fachliche Logik
- Model anpassen
  - generieren
  - Fachliche Logik

# Betriebswirtschaftliches Potenzial

- 60-70% Infrastrukturcode
- Qualität konstant hoch







**Vielen Dank  
für Ihre  
Aufmerksamkeit**



Grafikverweis:

Tom Stahl, Markus Völter

## **Modellgetriebene Softwareentwicklung** Techniken, Engineering, Management

dPunkt 2005, Februar 2005  
ISBN 3-89864-310-7