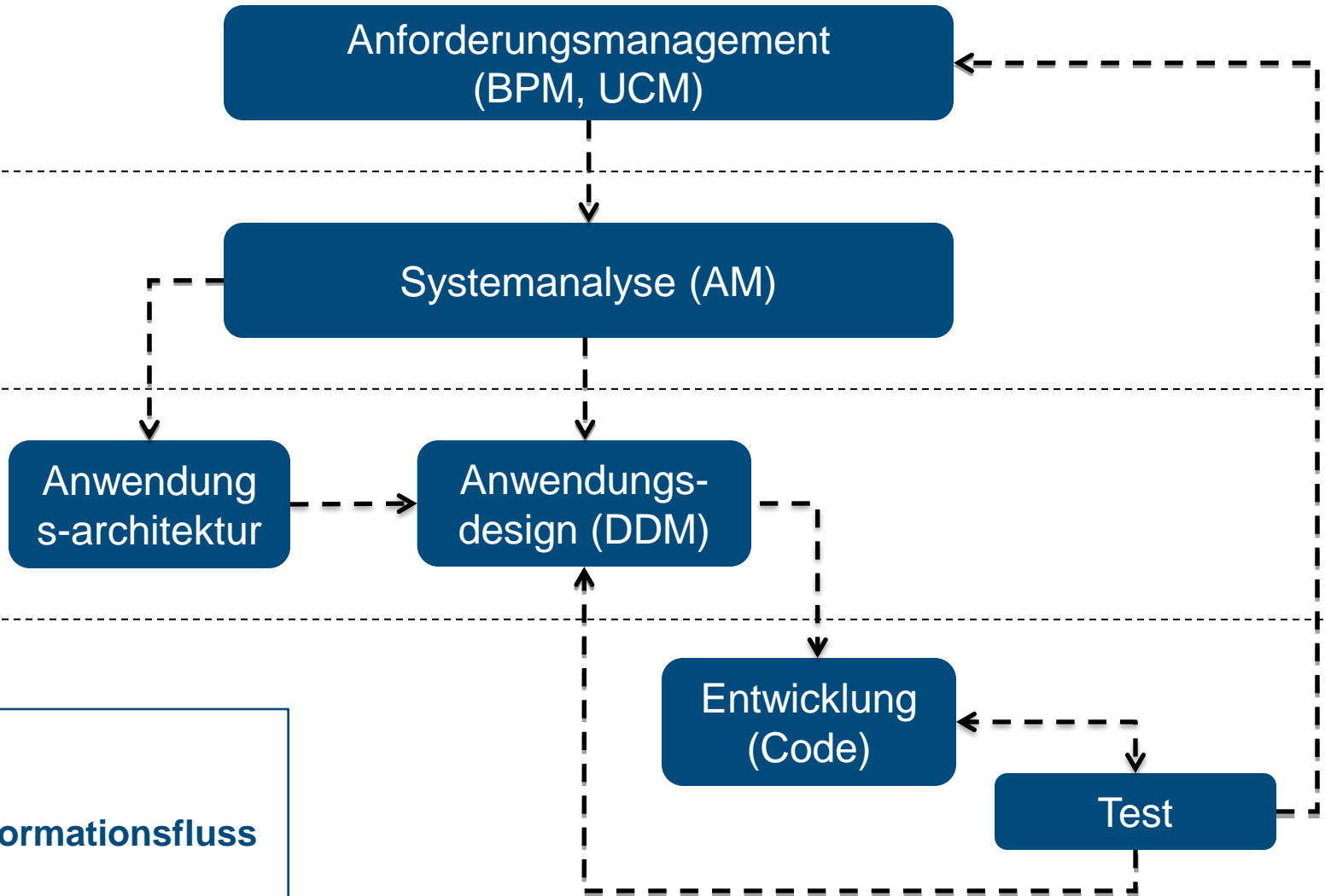


Beispielhaft MDSD in der Praxis

Dr. Shota Okujava
shota.okujava@isento.de
www.isento.de

- Einführung
- Softwareentwicklungsprozess und MDSD
- Technologien und Werkzeuge
- Demo
 - Entwicklung der Metamodelle
 - Entwicklung des PIM
 - Entwicklung eines Model2Model-Transformators
 - Bearbeitung des PSM
 - Entwicklung des Code-Generators
 - Code-Generierung

- Model Driven Software Development – erzeugen eines lauffähigen Codes aus formalen Modellen.
- Im Laufe eines Softwareentwicklungsprozesses können mehrere Modelle entstehen:
 - Fachliches Modell
 - Technisches Modell
 - Geschäftsprozessmodell
 - Business Domain Modell
 - ...
- Wichtig ist es, die Modellelemente aus unterschiedlichen Modellen aufeinander abbilden zu können und diese Abbildung persistent zu halten



- In einem Softwareentwicklungsprozess gestalten sich die Übergänge zwischen Disziplinen besonders schwer
- Mögliche Problemfelder sind:
 - Informationsbasis ist verstreut auf viele Dokumente/Artefakte
 - Die von einer Disziplin generierten Inhalte sind
 - nicht eindeutig – es gibt zu viele Interpretationsmöglichkeiten
 - für andere Disziplinen nicht verständlich
- Durch den generativen Übergang können viele dieser Problemfelder adressiert werden.

CIM

Anforderungsmanagement
(BPM, UCM)

PIM

Systemanalyse (AM)

PSM


Anwendung
s-architektur

Anwendungs-
design (DDM)

Entwicklung
(Code)

Test

Legende

-  Informationsfluss
-  (Teil-)Automatisierter Übergang



- Ein Modell ist ein vereinfachter Abbild der Realität
- Ein Modell hat eine bestimmte Zielsetzung. In einem Softwareentwicklungsprozess kann diese Zielsetzung je nach Disziplin unterschiedlich aussehen:
 - Anforderungsmanagement: Grobes Modell zur Veranschaulichung der Zusammenhänge und der Abläufe → BP-Modell
 - Systemanalyse: Fachlich bedingte Elemente, Beziehungen und Abläufe → PIM
 - Anwendungsdesign: Technische Gegebenheiten des Anwendungssystems – Persistenzmodell, Servicemodell → PSM

Metameta-Ebene
(entsp. OMG M3)

ecore

Meta-Ebene
(entsp. OMG M2)

PIM-
Metamodell

PSM-
Metamodell

Modell-Ebene
(entsp. OMG M1)

PIM-Modell

PSM-Modell

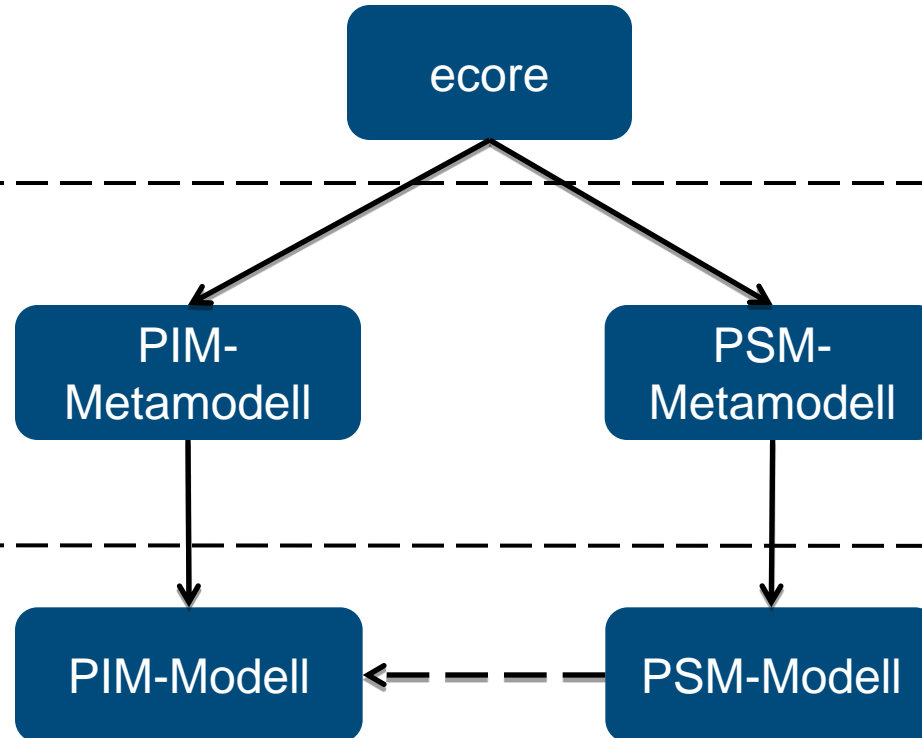
Legende



Abstraktion



Spezifikation



- IDE
 - Eclipse & Eclipse Modeling Framework
- Metametamodell
 - Ecore
- Model2Model-Transformation
 - ATL
- Code-Generierung
 - openArchitectureWare & expand
- Modellbearbeitung
 - Live-entwickelte eigene Plug-Ins für Eclipse

- ATL - Atlas Transformation Language – Entwickelt als Alternative zu QVT vom OMG
- Spezialisiert sich auf die Model2Model Transformation
- Der Transformationsskript besteht aus
 - Modulen bestehend aus
 - Rules – bilden Quellmodelle auf Zielmodelle ab
 - Helper-Strukturen – wiederverwendbare Operationen auf Modellelemente
 - Queries – berechnen primitive Werte aus Modellen und
 - Bibliotheken – in Module importierbare gekapselte Module
- Typische Einsatzbereiche:
 - Analysemodell in ein (technisches) Designmodell transformieren
 - Aus Klassenmodell ERM erstellen
 - ...

- Metamodell für Systemanalyse definieren
- PIM entwickeln
- Metamodell für Anwendungsdesign definieren
- PIM in PSM transformieren, PSM weiterentwickeln
- Aus PSM den Code generieren, Code vervollständigen

Livedemo

- Ist CASE doch nicht gestorben? – Ist MDSD die Neugeburt der CASE und CASE-Werkzeuge?
- Wie ist der Reifegrad von MDSD?
- Was sind die Herausforderungen?
- Gibt es „typische“ Projekte für MDSD?
- Hat MDSD Zukunft? Wie sieht diese aus?

Fragen?

Vielen Dank!

shota.okujava@isento.de