

Model Driven Software Development

Potentiale modellgetriebener Softwareentwicklung

Referent:
Hartwig Tödter

Inhaltsverzeichnis

1. Grundideen modellgetriebener Softwareentwicklung

2. Vorteile von MDSD-Ansätzen

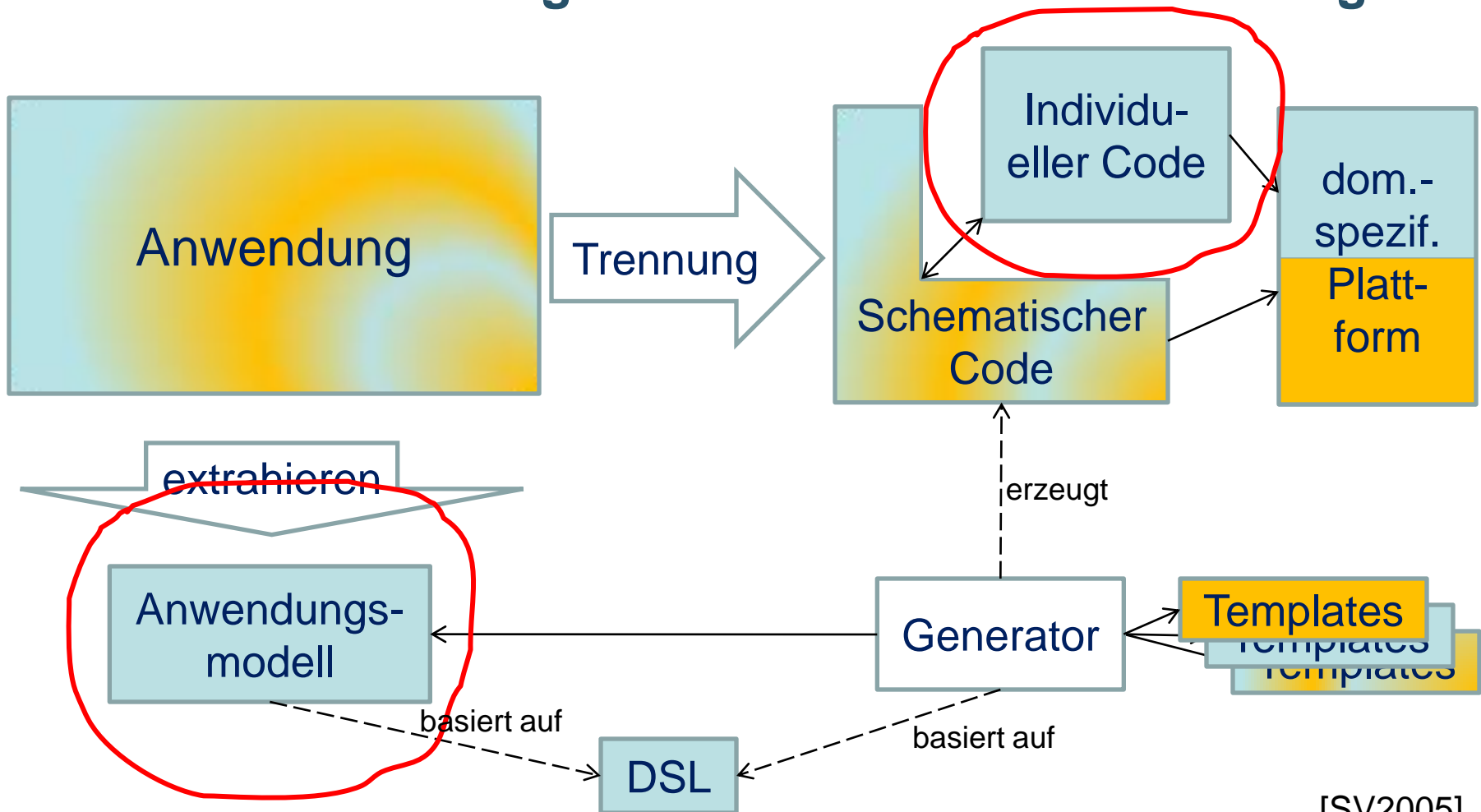
- Verwendung einer fachlichen Modellierungssprache
- Trennung von Verantwortlichkeiten
- Automatisierung und Wiederverwendung

3. Aufwandsvergleiche

4. Zusammenfassung

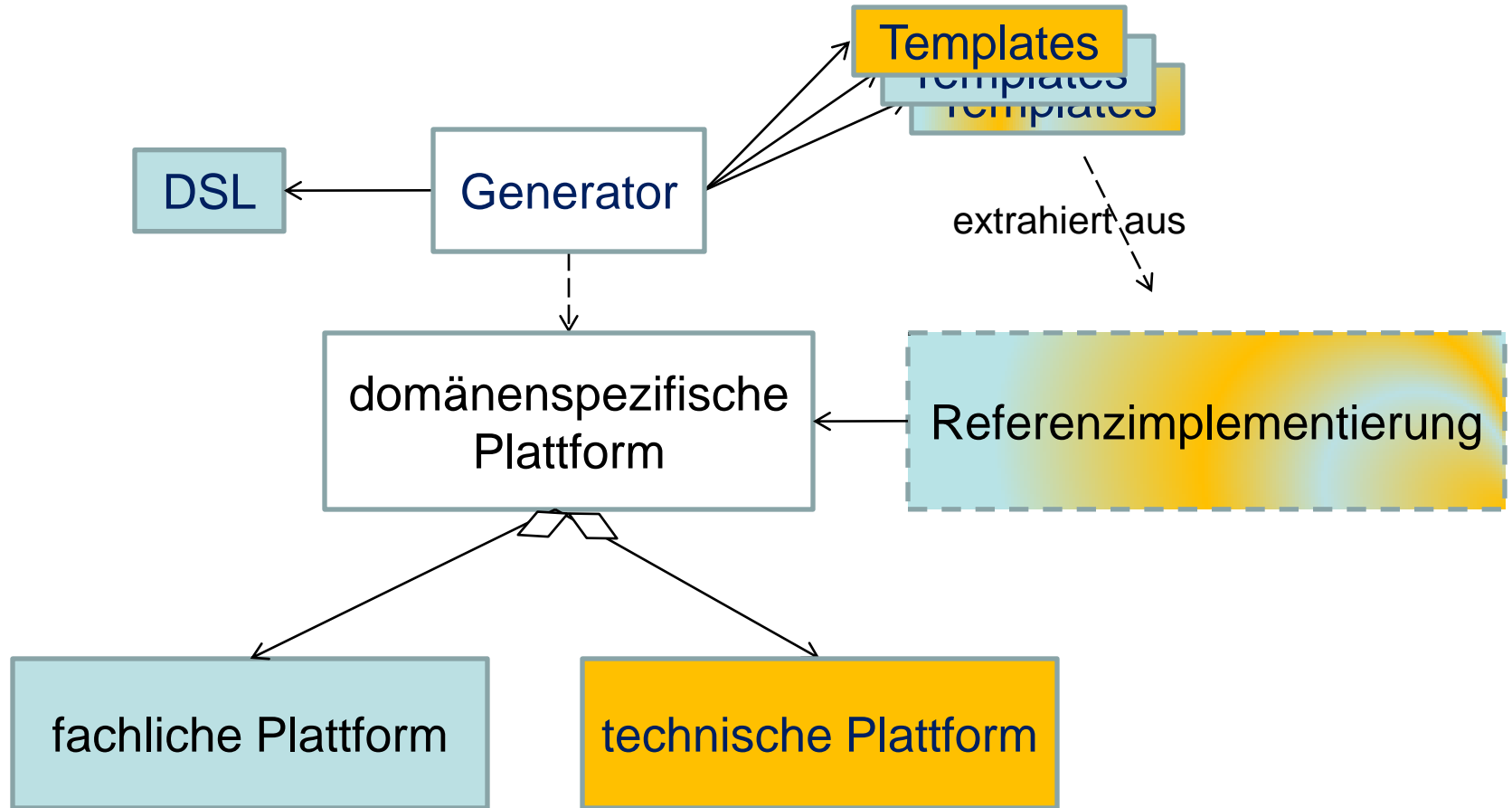
1. Grundideen modellgetriebener Softwareentwicklung

Grundidee modellgetriebener Softwareentwicklung



[SV2005]

Domänenarchitektur

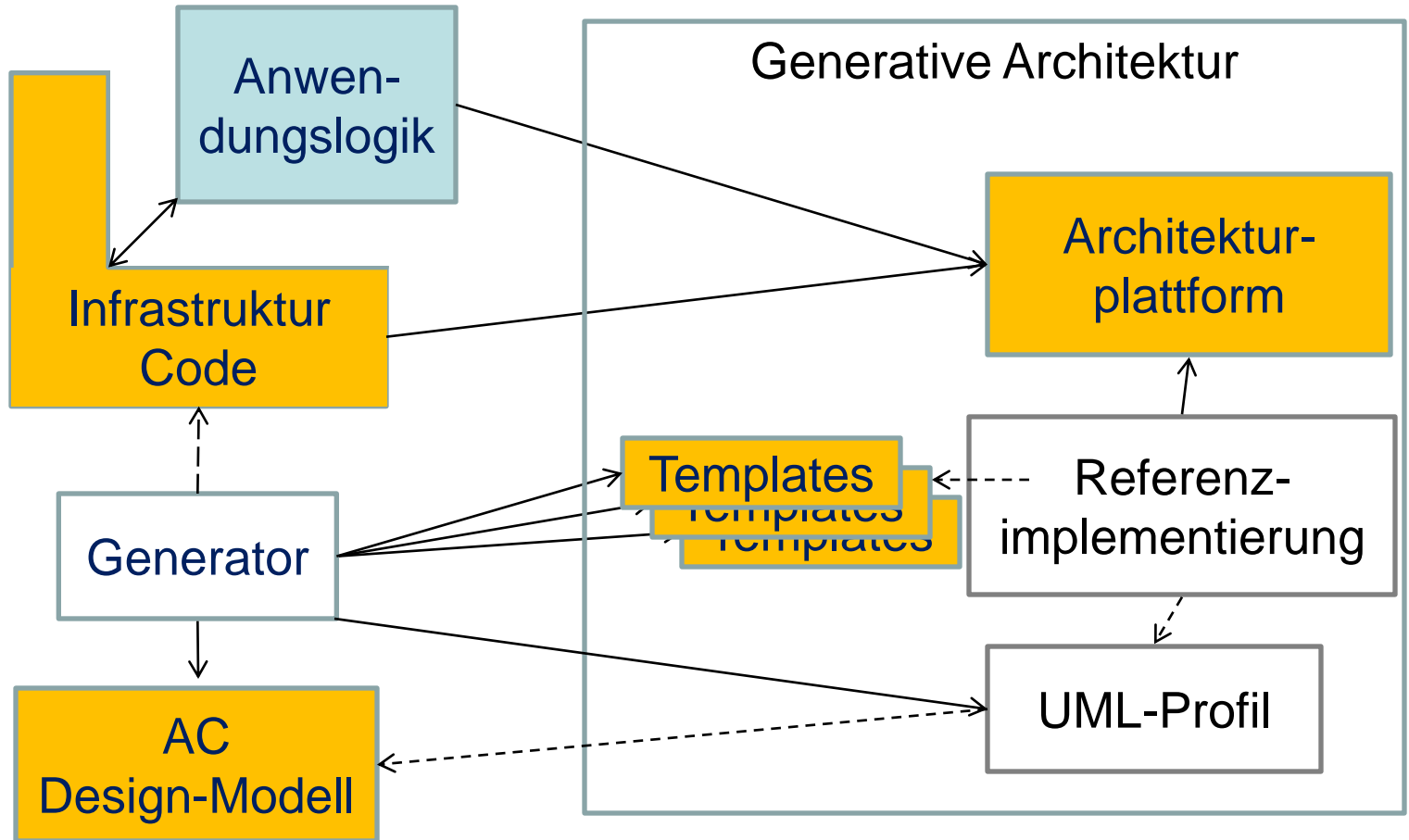


Architekturzentrierte MDSD

Steigerung von

- **Entwicklungseffizienz**
- **Softwarequalität**
- **Wiederverwendbarkeit**
- **ganzheitliche automatisierte Erstellung von Infrastrukturcode**
- **Minimierung der manuellen Erstellung von redundanten, technischen Code**

Architekturzentrierte MDS



2. Vorteile von MDSD-Ansätzen

Verwendung einer fachlichen Modellierungssprache

- Das Modell ist zentraler Bestandteil der Entwicklung
 - ➔ **Die Qualität der Anwendung wird deutlich höher**
- Technische Details werden in den Modellen vermieden
 - ➔ **Die Modelle werden einfacher und aussagekräftiger**
- Modellelemente orientieren sich an fachlichen Konzepten und Begriffen
 - ➔ **Das Modell kann von Fachabteilungen verstanden werden**
 - ➔ **Das Modell wird zusammen mit der Fachabteilung erstellt**
 - ➔ **Deutlich verbesserte Kommunikation zwischen Fachabteilung und IT**

Trennung von Verantwortlichkeiten

Technische Experten

- Aufbau der Architektur und Plattform
- Generator-Konfiguration
- Referenzimplementierung und Templates

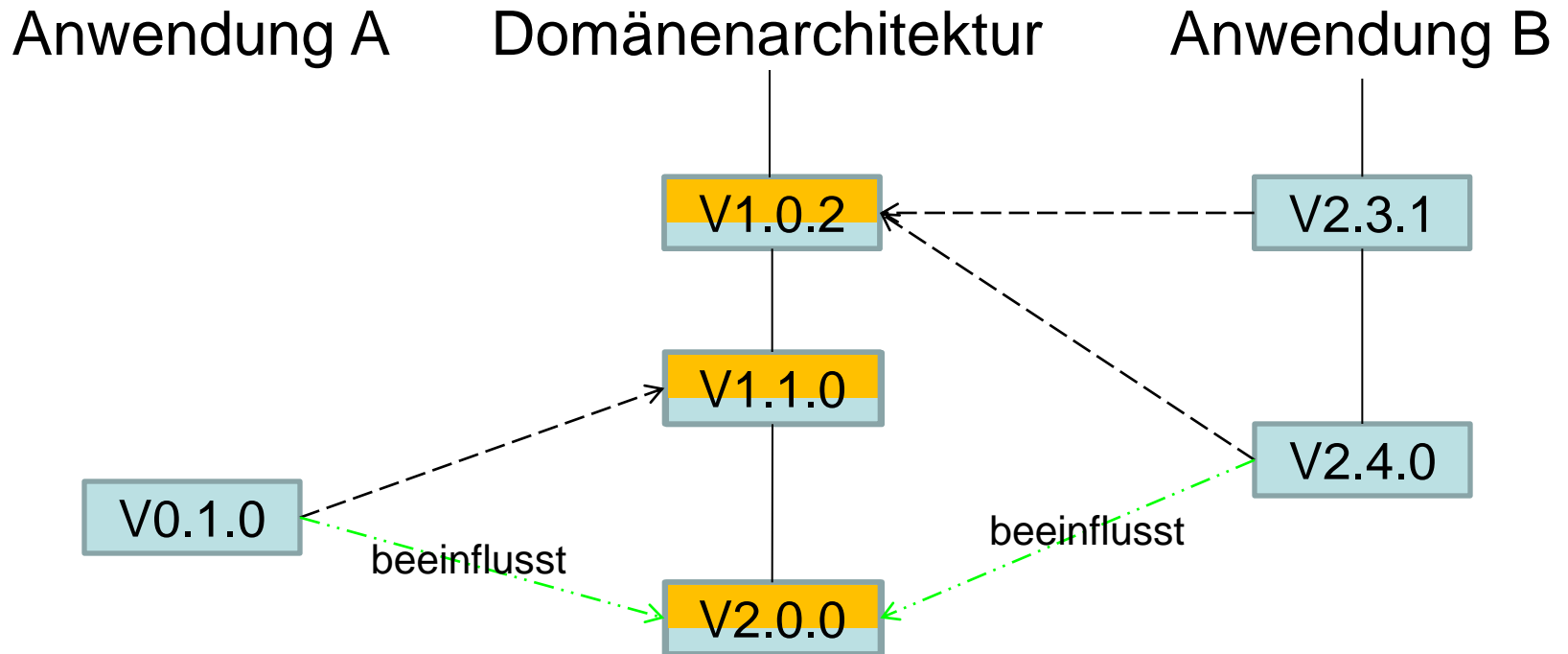
Konzentration auf Architektur
und Technik

Domänen Experten (der IT-Abteilung)

- Kommunikation
- Aufbau der DSL
- Anwendungsentwicklung
Implementierung der Fachlogik

Konzentration auf DSL, funktionale
Anforderungen und Kontakt zur
Fachabteilung, Anwendungsentwicklung

Getrennte Entwicklungsstränge

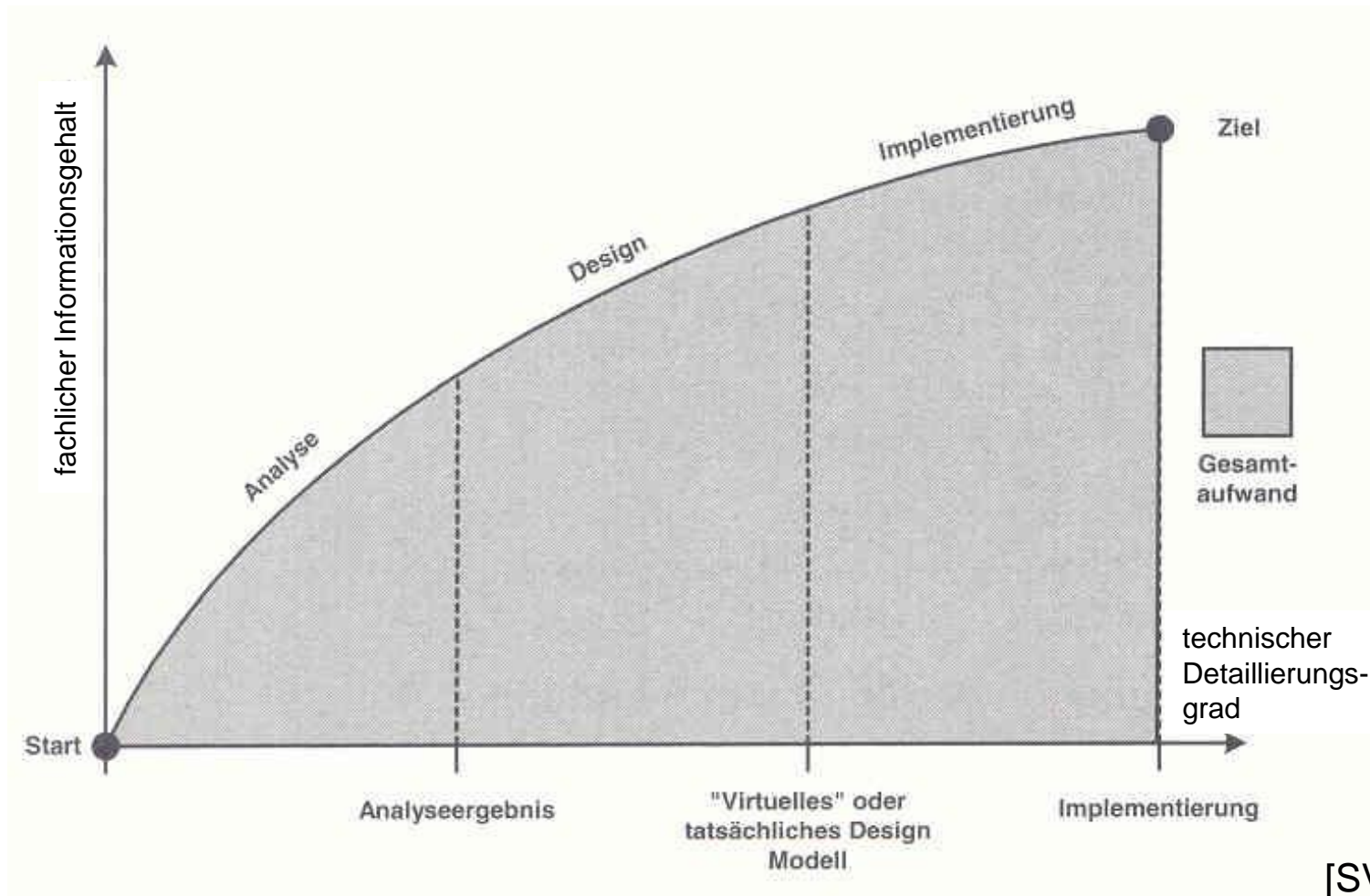


Automatisierung und Wiederverwendung

- Automatisierung wiederholter Abläufe durch Generierung
 - **Steigende Qualität der generierten Teile**
 - **Verkürzung von Entwicklungszeiten**
- Wiederverwendung der domänenspezifischen Plattform
 - **Stabile, einheitliche Softwarearchitektur**
 - **Zentralgesteuerter Technologiewandel**

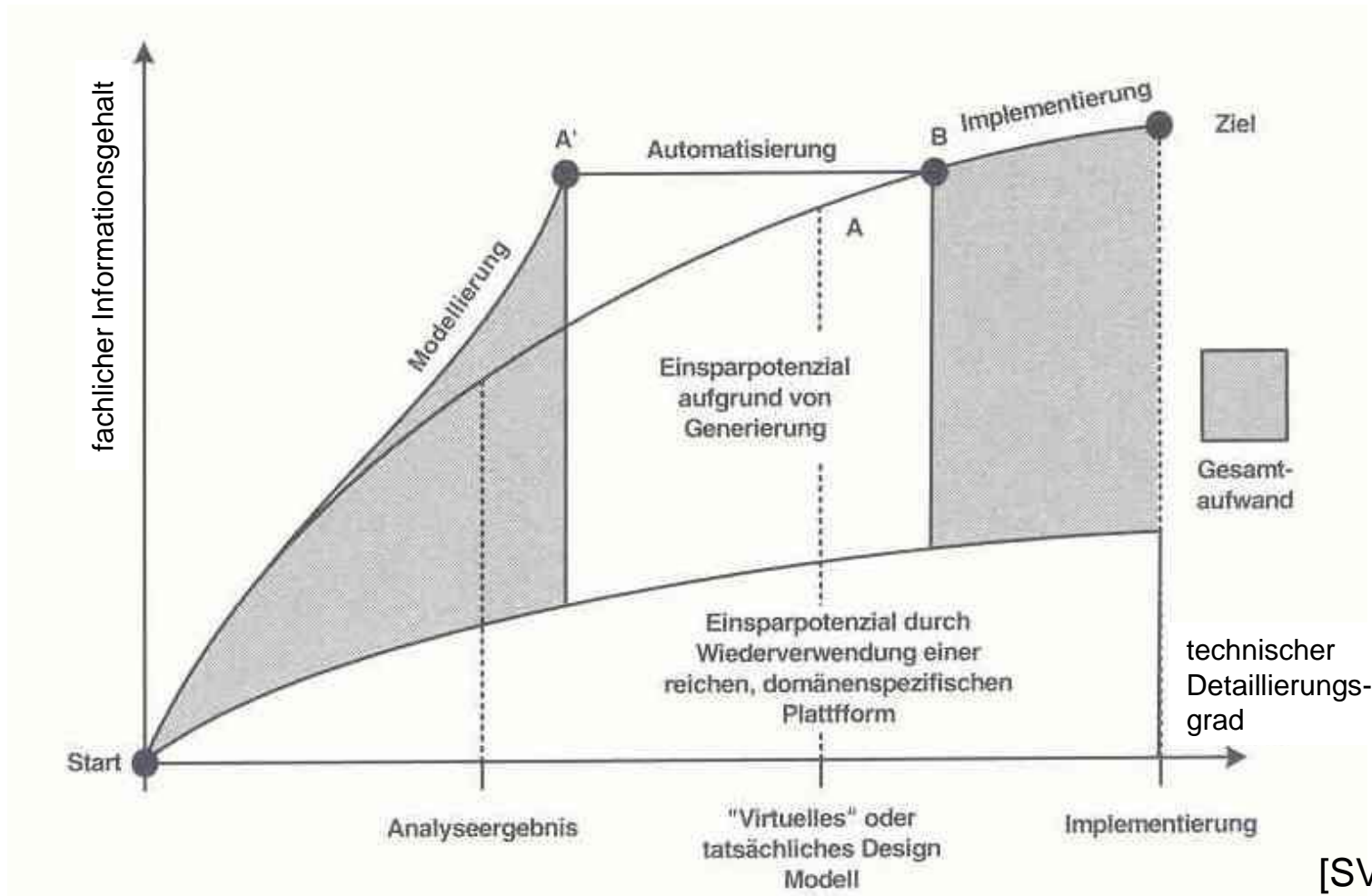
3. Aufwandsvergleiche

Aufwand bei weitgehend manueller Entwicklung



[SV2005]


Aufwand bei MDSD mit teilw. manueller Entwicklung



[SV2005]

Vergleich: manuelle Entwicklung / MDSD

	manuelle Entwicklung	MDSD
Referenzimplementierung (kB)		1000
manuell geschriebener Code (kB)	18800	2200
Modelle (kB)		3400
Transformationen (kB)		200
Summe	18800	6800
In % zur manuellen Entwicklung	100%	36%



[SV2005]

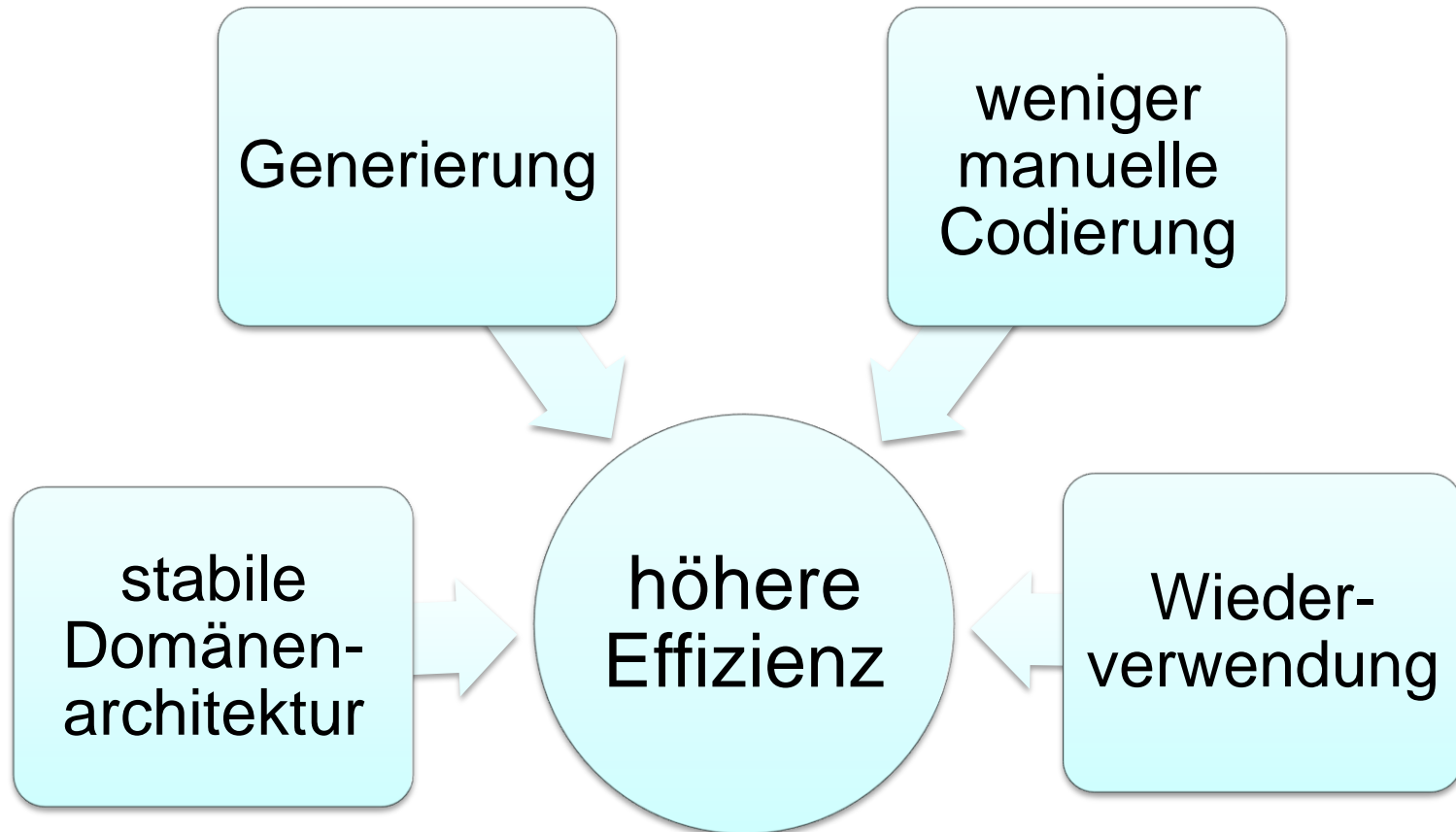
Vergleich: manuelle Entwicklung / MDSD

	manuelle Entwicklung	MDSD
Manuell codiert (LOC)	126	20
Generierter Code (LOC)	0	106
Summe LOC	126	126
Statische Modelle (AME)	0	21
Dynamische Modelle (AME)	0	20
Summe AME	0	41
Summe man. LOC und AME	126	62
In % zur man. Entwicklung	100%	49%

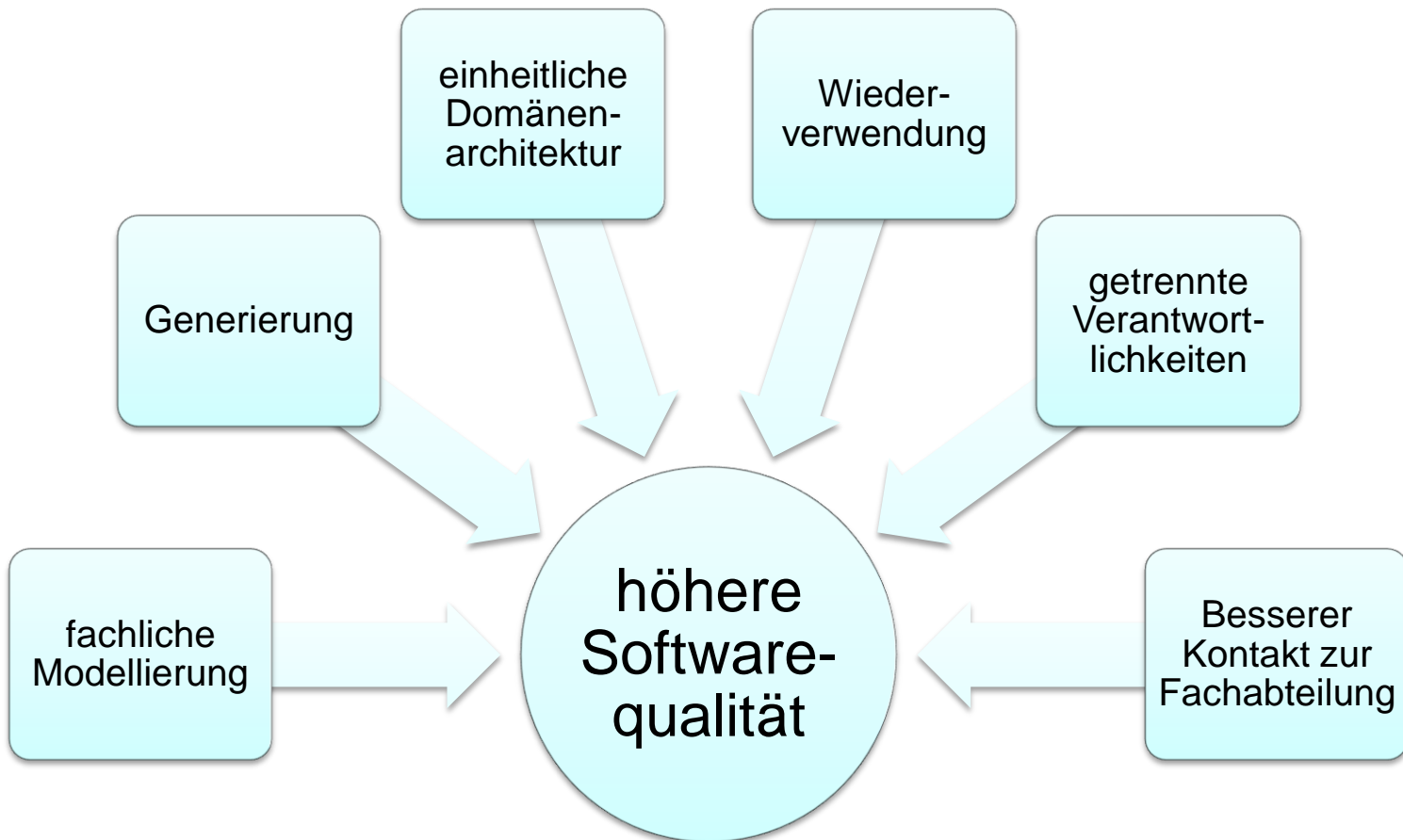
[B2003]

4. Zusammenfassung

Vorteil der MDSD: Entwicklungseffizienz



Vorteil der MDSD: Softwarequalität



Literaturverzeichnis

- [SV2005] - Stahl, Thomas; Völter, Markus: Modellgetriebene Softwareentwicklung: Techniken, Engineering, Management. 2005**
- [PT2007] - Pietrek, Georg; Trompeter, Jens (Hrsg.): Modellgetriebene Softwareentwicklung. 2007**
- [B2003] - Bettin, Jorn: Model-Driven Architecture – Implementation & Metrics V1.1. 2003**

www.iks-gmbh.com