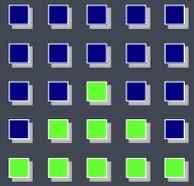


**DELTA**  
software  
technology





**DELTA**  
software  
technology

Karlsruhe, 21. Mai 2014

Softwareentwicklung - Modellgetrieben und trotzdem agil

Daniela Schilling

Delta Software Technology GmbH



The Perfect Way to Better Software

---

# Modellgetriebene Entwicklung

---

Garant für Effizienz und Qualität

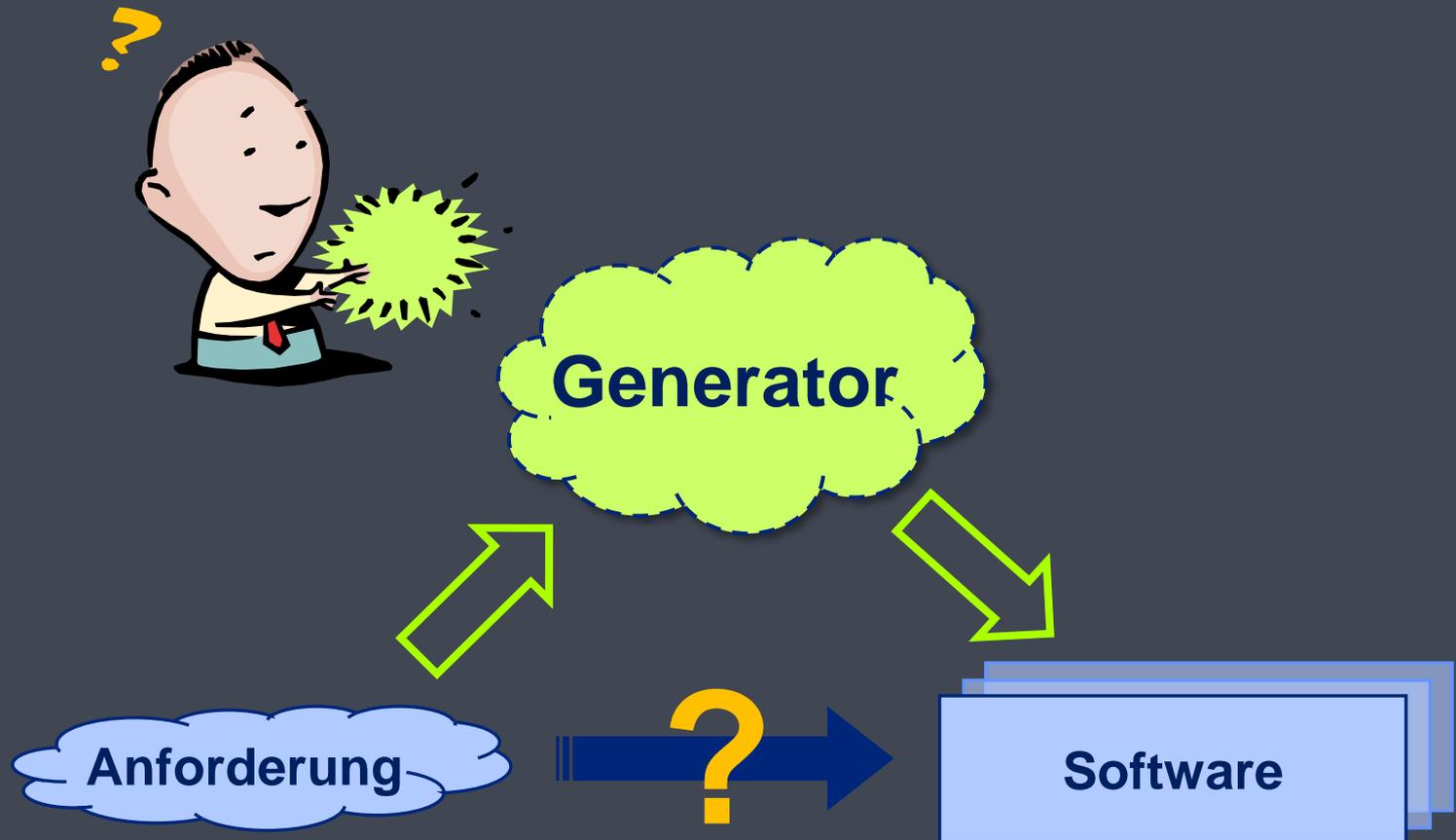
Grundlage sind Modelle, Generatoren und DSLs

Vorbehalte

- Anforderungen an Generator und DSL müssen frühzeitig und vollständig bekannt sein
  - Großer Initialaufwand bis eine erste Version verfügbar ist
  - Hoher Aufwand bei Erweiterungen und Refakturierungen
- ⇒ Modellgetriebene und agile Softwareentwicklung widersprechen sich – oder ...?

# And then ... a miracle occurs

---



# Generator - Universell vs. spezialisiert

---

## Universelle Generatoren

- Oft Syntaxtransformatoren, z.B. UML-> Java
- Wenig spezialisiert
- Keine Optimierung des Codes
- Keine Wiederverwendung von existierendem Code

## Spezialisierter Generatoren

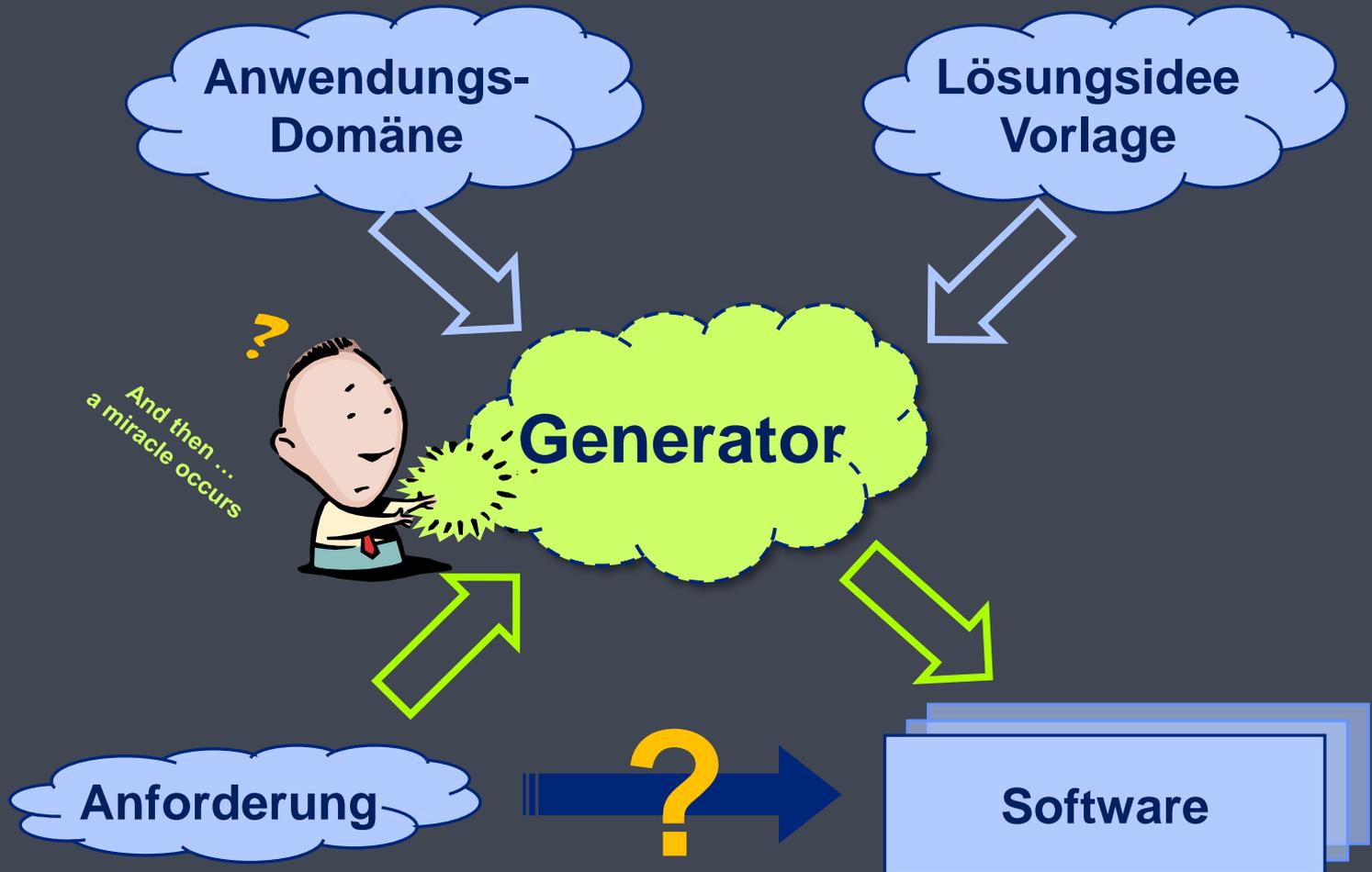
- An die vorliegende Aufgabe angepasst
- Generierter Code optimiert
- Wiederverwendung von existierendem Code möglich

## Generator und DSL Entwicklung

- Agil
  - ❖ Mit skalierbarer Schrittweite
  - ❖ Nach jedem Schritt funktionierender Generator und DSL
- Modellgetrieben
  - ❖ Mit inkrementeller Modellentwicklung
- Zu jedem Zeitpunkt in die Entwicklung integrierbar
  - ❖ Wiederverwendung von existierendem Code
  
- Im Rahmen des Anwendungsprojekts!

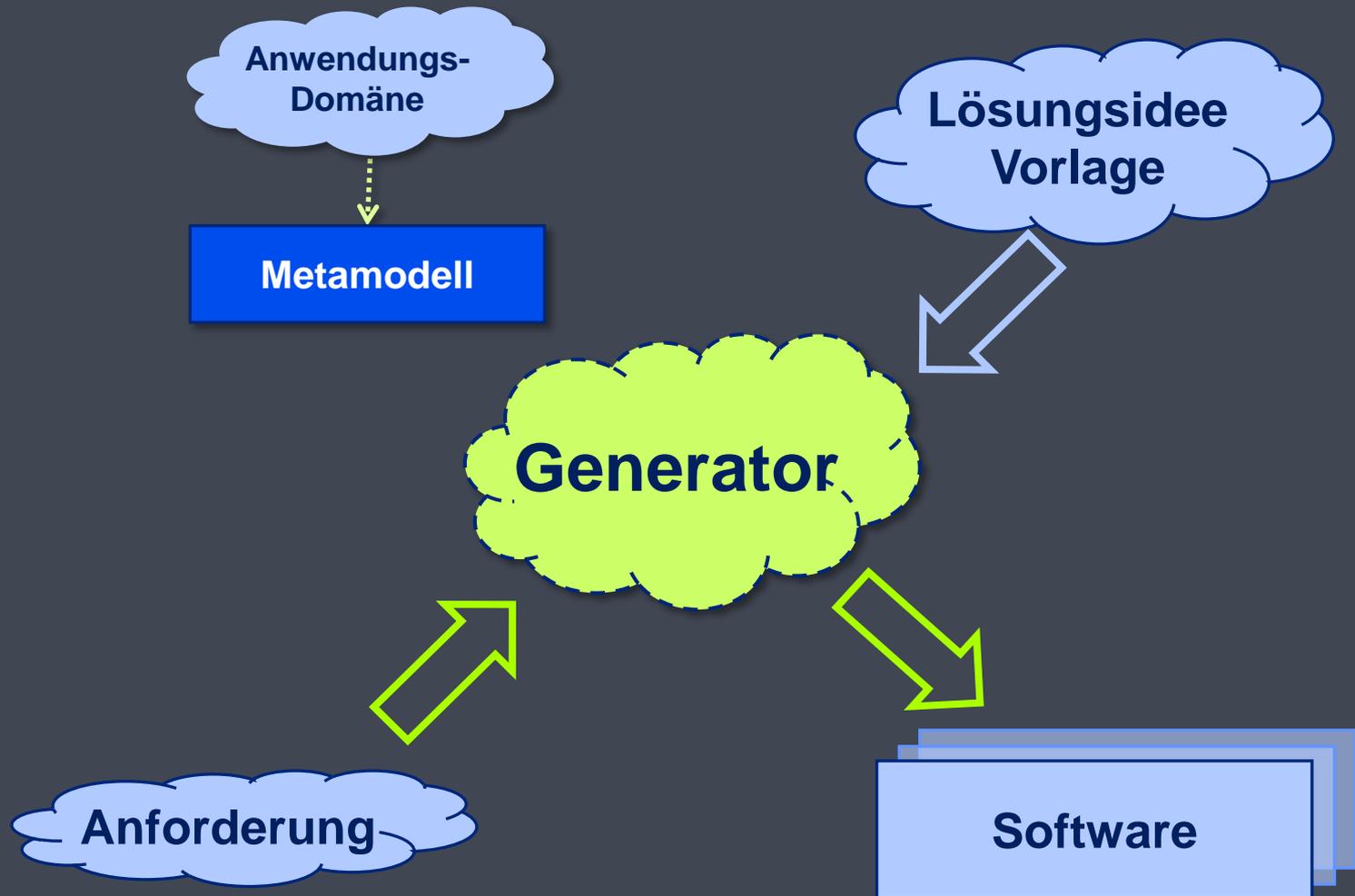
# Ein Generator?

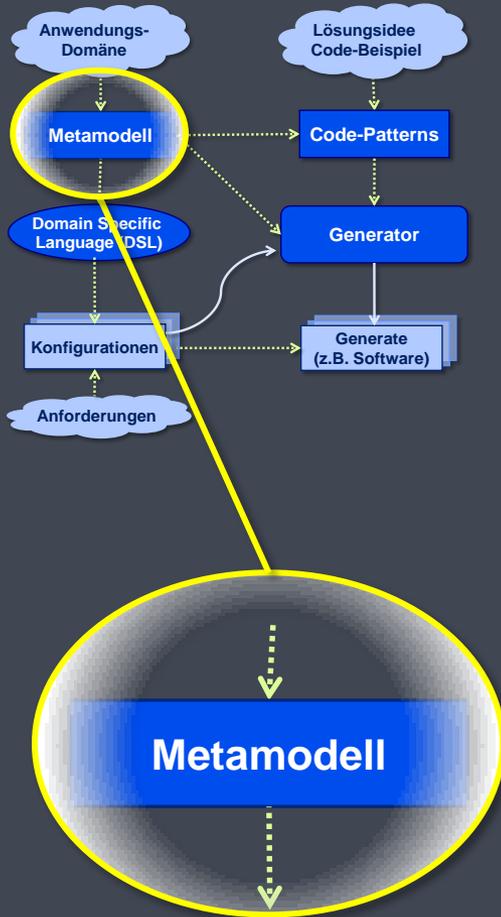
---



# Ein Generator?

---





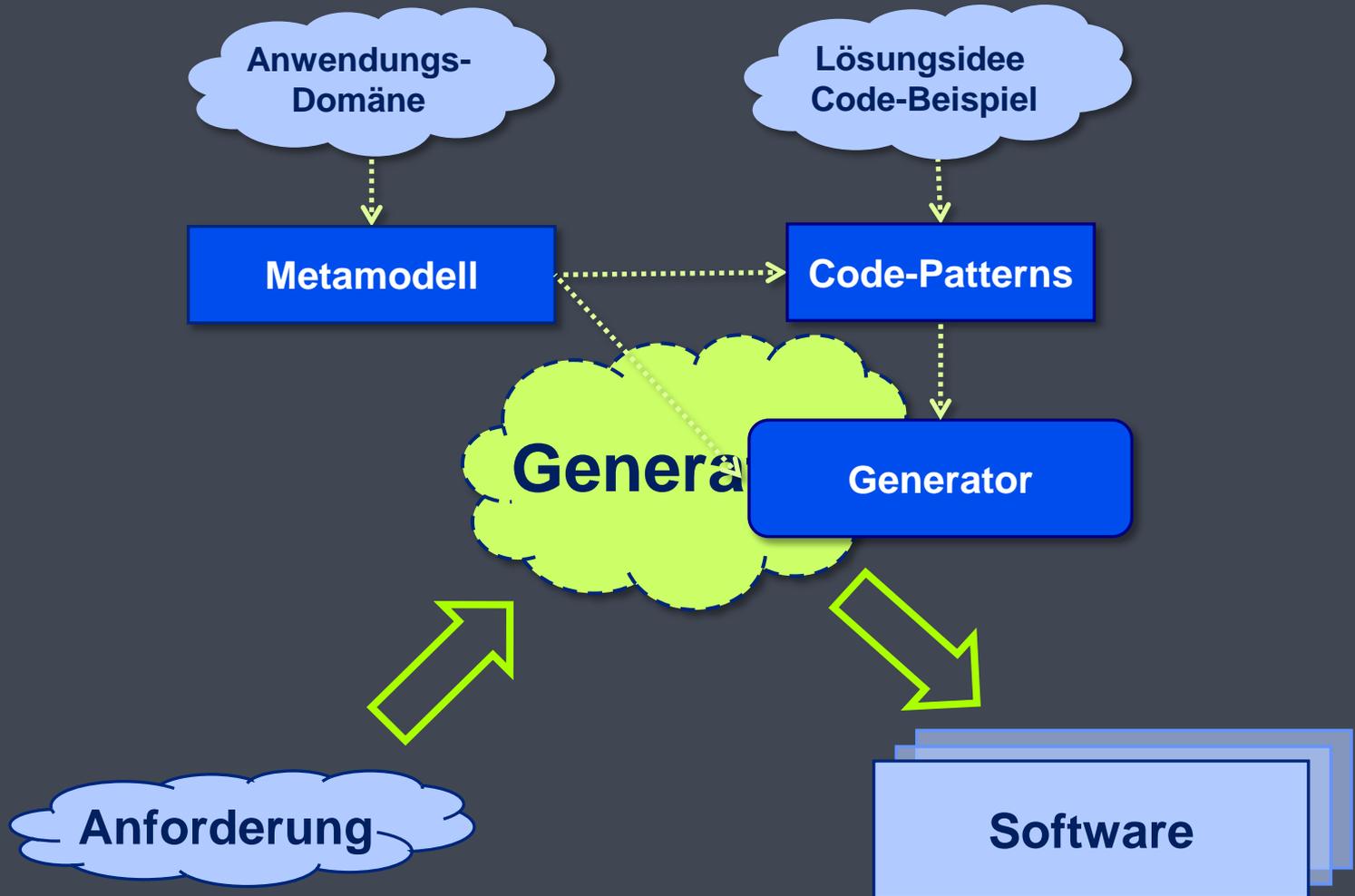
## Modell der Variabilität

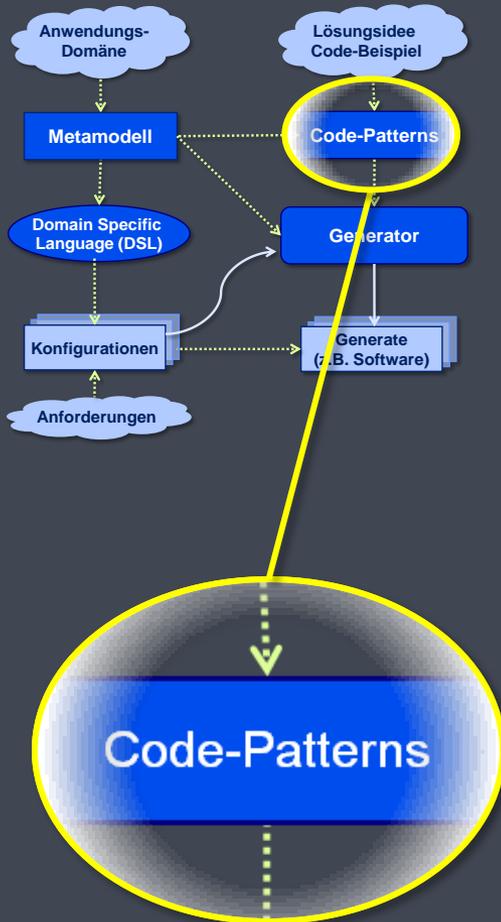
- Was unterscheidet die Mitglieder einer Systemfamilie?
  - ❖ Invariante Eigenschaften sind für einen Generator nicht relevant
- ≈ Feature Model (Merkmalmodell)

## Agilität

- Nur aktuell benötigte Variabilität wird modelliert
- Erweiterungen, Anpassungen und Restrukturierungen in späteren Entwicklungszyklen

# Ein Generator?





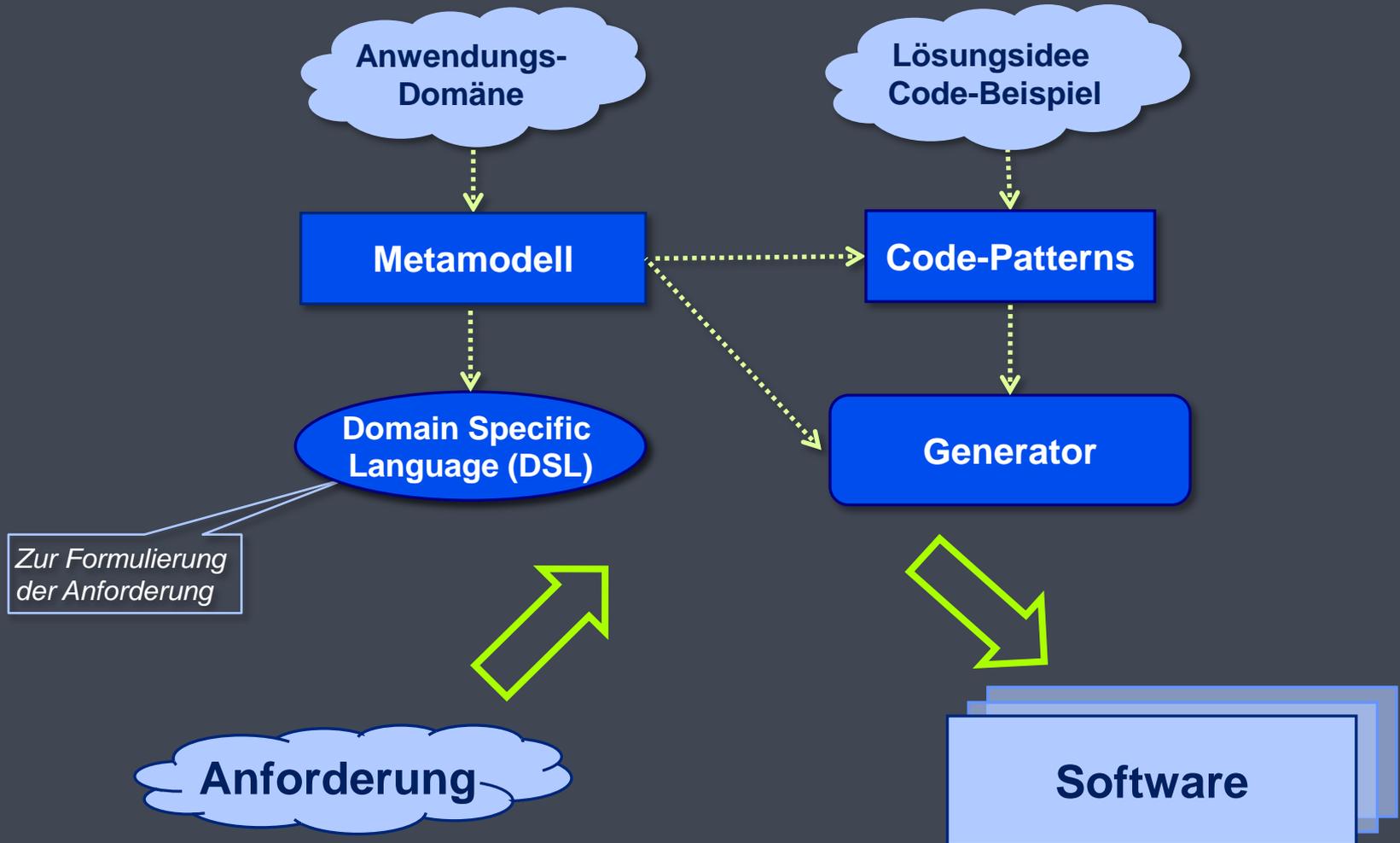
## Wiederverwendbare, parametrisierte Code-Bausteine

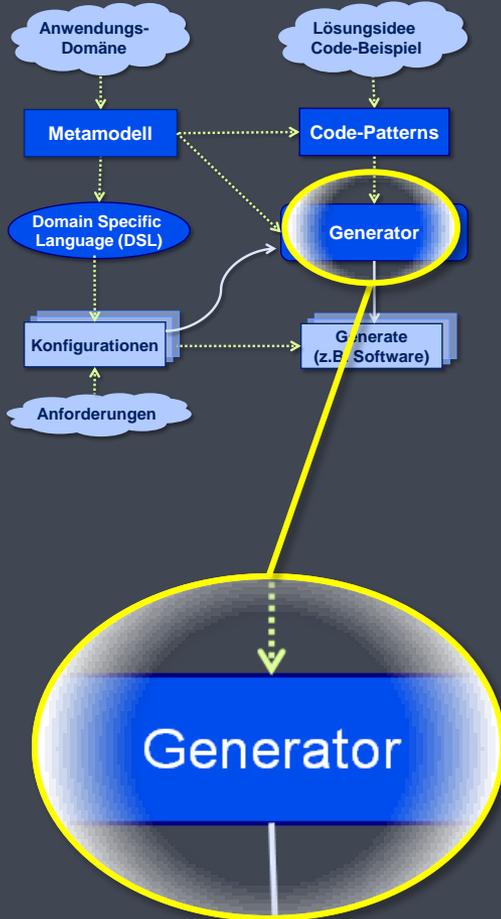
- Basierend auf Frame/Slot-Konzept
- Verknüpfung mit dem Metamodell
  - ❖ Automatische Navigation in der Konfiguration
- Schrittweise Ableitung von Code-Pattern aus Vorlagen und Prototypen

## Agilität

- Innerhalb eines Zyklus werden nur
  - ❖ Pattern für Codefragmente abgeleitet, die bis zu diesen Zyklus variable Stellen enthalten
  - ❖ Variable Stellen in den Pattern identifiziert, die bis zu diesem Zyklus relevant sind
- Weitere Pattern und variable Stellen werden in späteren Entwicklungszyklen abgeleitet

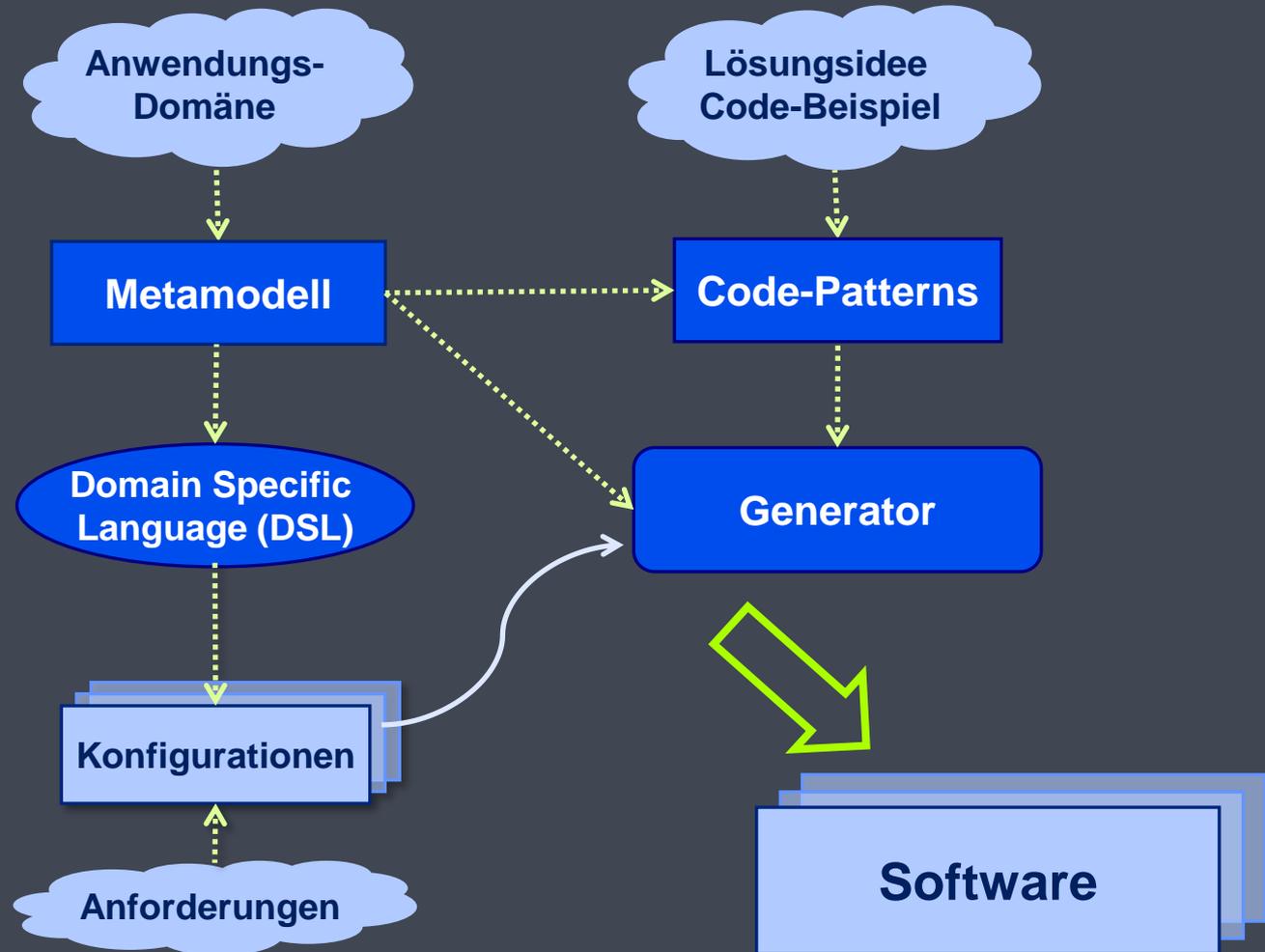
# Ein Generator?



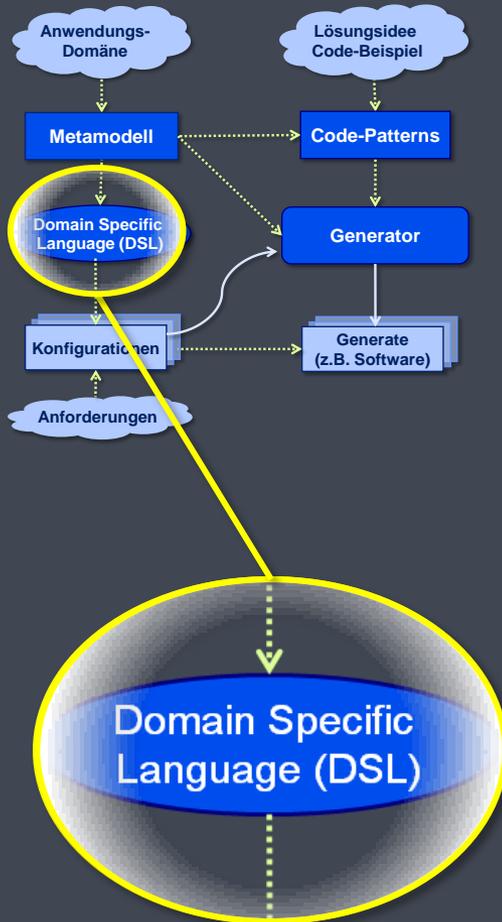


Der Generator wird automatisch erzeugt  
➤ aus Meta-Modell und Code-Pattern

# Ein Generator?



# Domain Specific Language



## Variantenmodell (Metamodell)

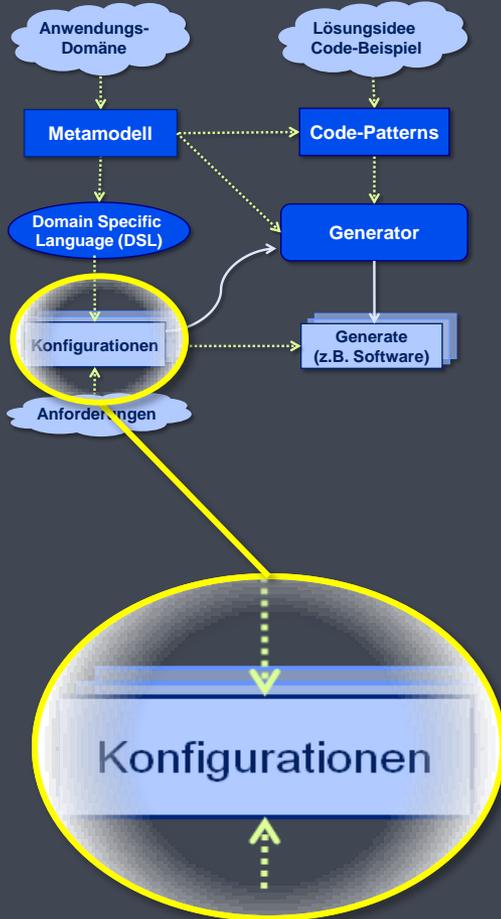
- Grammatik implizit und vollständig

## DSL → Konkrete Syntax

- automatisch ableitbar  
= Variantenmodell + Konfigurationsmuster
- Konfigurationsmuster ähnlich wie Code-Pattern

## Agilität

- Konfigurationsmuster werden nur für bis dahin vorhandene Variabilität erstellt
- Neue Muster und Erweiterung existierender Muster erfolgt zu späterem Zeitpunkt



Eine Konfiguration legt die Parameter für ein bestimmtes Produkt fest

- Produkt = 1 oder mehrere Generate

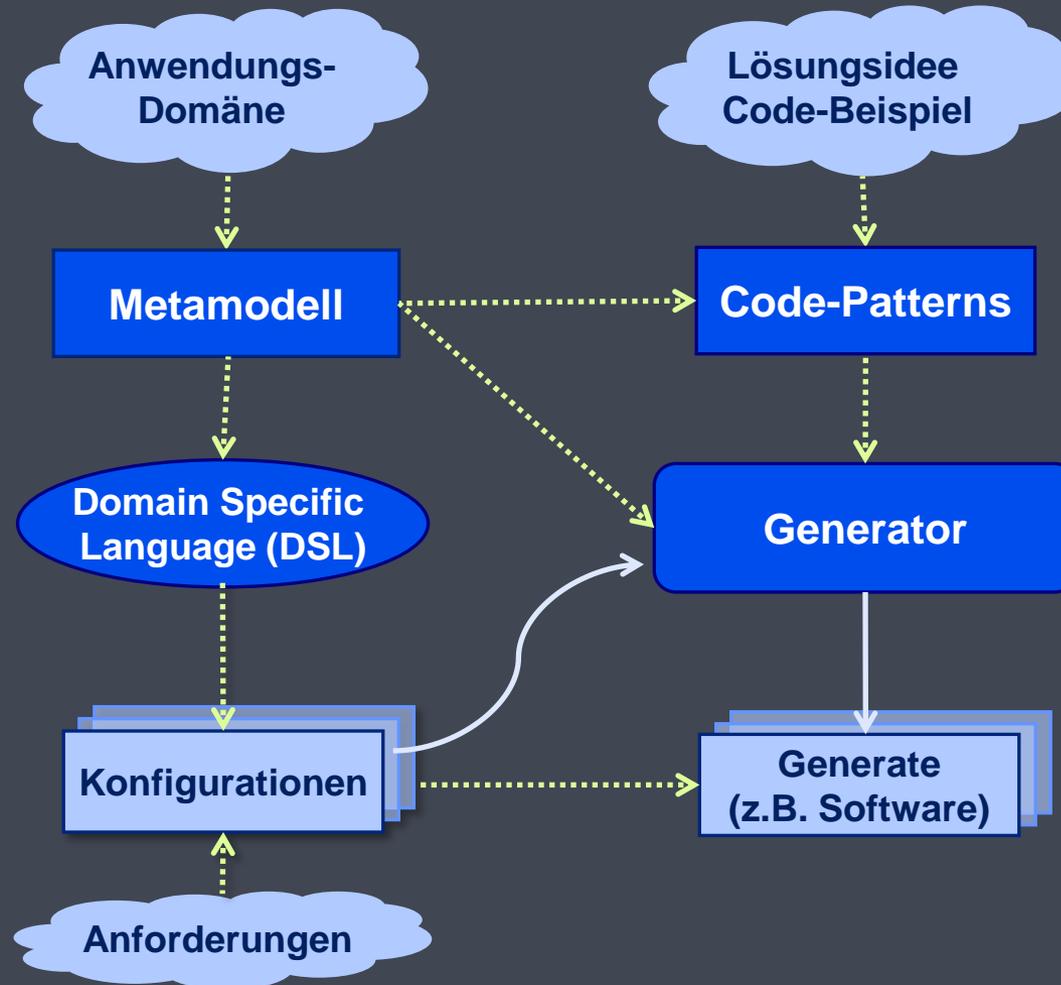
Konfiguration = Instanz Metamodell

Eine Konfiguration kann als Input für mehrere Generatoren dienen

Agilität

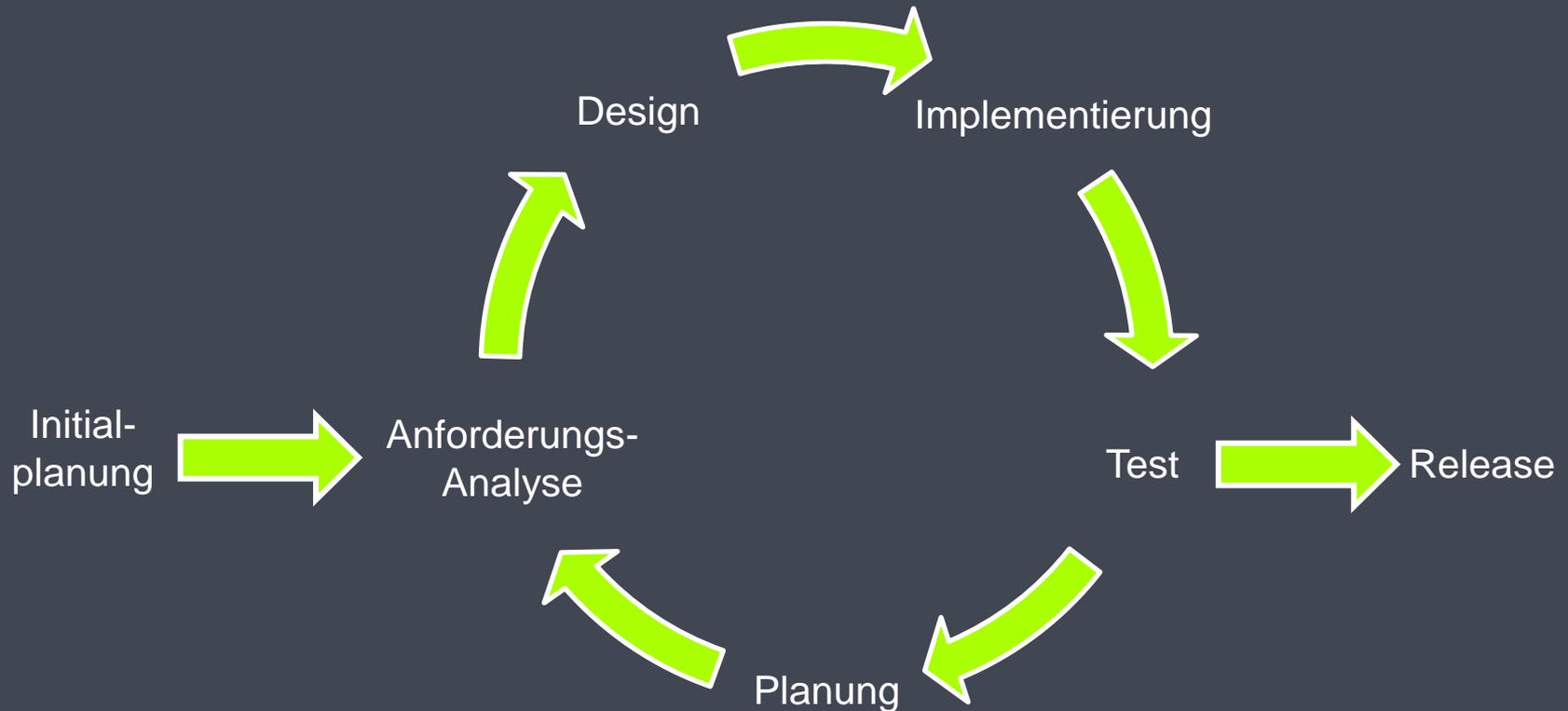
- Konfiguration entspricht dem aktuellen Zustand der DSL
- Erweiterung der DSL = Erweiterung der Konfigurationsmöglichkeiten

# Ein Generator?



# Was ändert sich durch agiles MDSD?

---



Welchen Einfluss haben modellgetriebene  
und generative Techniken auf den agilen Prozess?

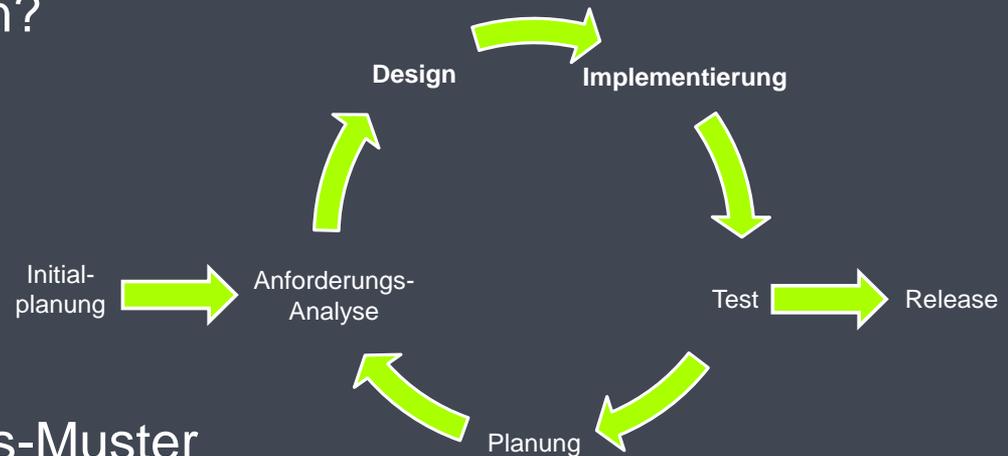
---

## Design

- Welche Komponenten sollen modellgetrieben (weiter)entwickelt werden?
  - ❖ Neu oder weiterhin?
  - ❖ Welche Vorlagen?

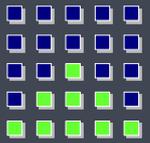
## Implementierung

- Modell anpassen
- Code- und Konfigurations-Muster
  - ❖ Anpassen und/oder neue ableiten
- Konfigurieren und generieren



## Kombination der Vorteile von modellgetriebener und agiler Entwicklung

- Varianten (technische/funktional) können parallel entwickelt werden
- Wiederverwendung von existierendem Code
- Generatoren und DSLs können zu beliebigem Zeitpunkt in die Entwicklung integriert werden
- Entwicklung in Schritten
  - ❖ Schrittweite skalierbar
  - ❖ Nach jedem Schritt steht ein funktionsfähiger Generator samt passender DSL zur Verfügung
- Verteilte Entwicklung von Generator und DSL möglich



**DELTA**  
software  
technology



Vielen Dank

Mehr Infos: [www.delta-software.com](http://www.delta-software.com)



ENTWICKLERTAG

# meet the **SPEAKER** **@speakerlounge**



1. OG DIREKT ÜBER DEM  
EMPFANG