

# Order Management

## Anforderungen

Eine Universalbank entwickelt ein neues Order Management System für den Zahlungsverkehr. Die existierenden Transaktionssysteme auf COBOL-Basis können aber nicht ersetzt werden, da einerseits die schiere Menge an Code nicht in vernünftiger Zeit zu bewältigen und andererseits ein Teil des Wissen über die Legacy verloren gegangen ist. Ein erster Anlauf für die Neuentwicklung scheitert: Dem Projektteam fehlen erfahrene COBOL-Entwickler und die Architektur leidet unter starken Kopplungen zwischen den einzelnen Modulen. In der Not entscheiden die Projektverantwortlichen, die Entwicklung mit zwei Massnahmen zu verbessern:

- Aufbau einer Service-orientierten Architektur (SOA) als Integrationsplattform zwischen bisherigen und neuen Systemen
- Automatisierung durch modellgetriebene Methoden (MDD/MDA): Die Service-Implementierung wird mit Hilfe von Klassen- und Activity-Diagrammen modelliert. Aus dem bestehenden Legacy-Code werden entsprechende Code-Templates extrahiert.



## Highlights

- Die Modellierung von Business Workflows erfolgt mittels State Diagrammen. Damit kann der logische Ablauf von Business Prozessen durch Business-Experten ohne IT-Hintergrund definiert werden.
- COBOL-Code wird direkt aus Klassen- und Activity-Diagrammen generiert. Damit entschärft sich der Bedarf für hochqualifizierte COBOL-Programmierer.

## Erfolge

- Beschleunigung der Entwicklung um Faktor 3
- Verfolgbarkeit von Requirements bis zum Code
- Plug-and-Play-Funktionalität für neue Komponenten
- Jederzeit aktuelle Dokumentation
- Durchsetzung von Best Practices bez. Architektur und Design
- Optimiertes Testing und einfache Integration