



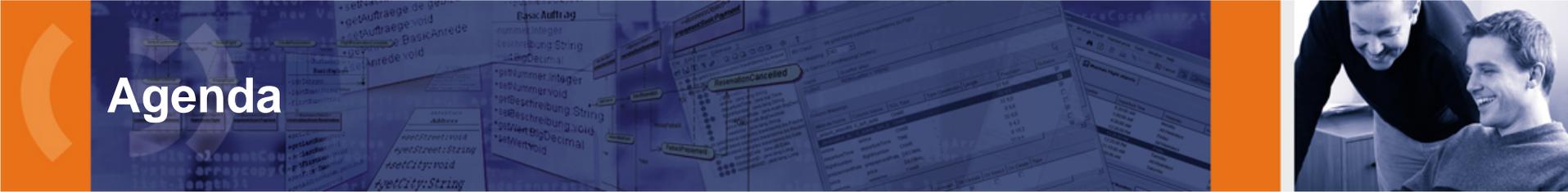
MDRE

die nächste Generation des
Requirements Engineerings

Tom Krauß, GEBIT Solutions GmbH



Technologie beherrschen ► Prozesse gestalten ► Lösungen schaffen ►►



Agenda

- **Requirements Engineering heute – eine Bestandsaufnahme**
- **Modell-Driven Requirements Engineering – die nächste Generation des Requirements Engineering**
- **Vorzüge des MDRE-Ansatzes und praktische Erfahrungen von GEBIT**





Requirements Engineering heute

Eine Bestandsaufnahme

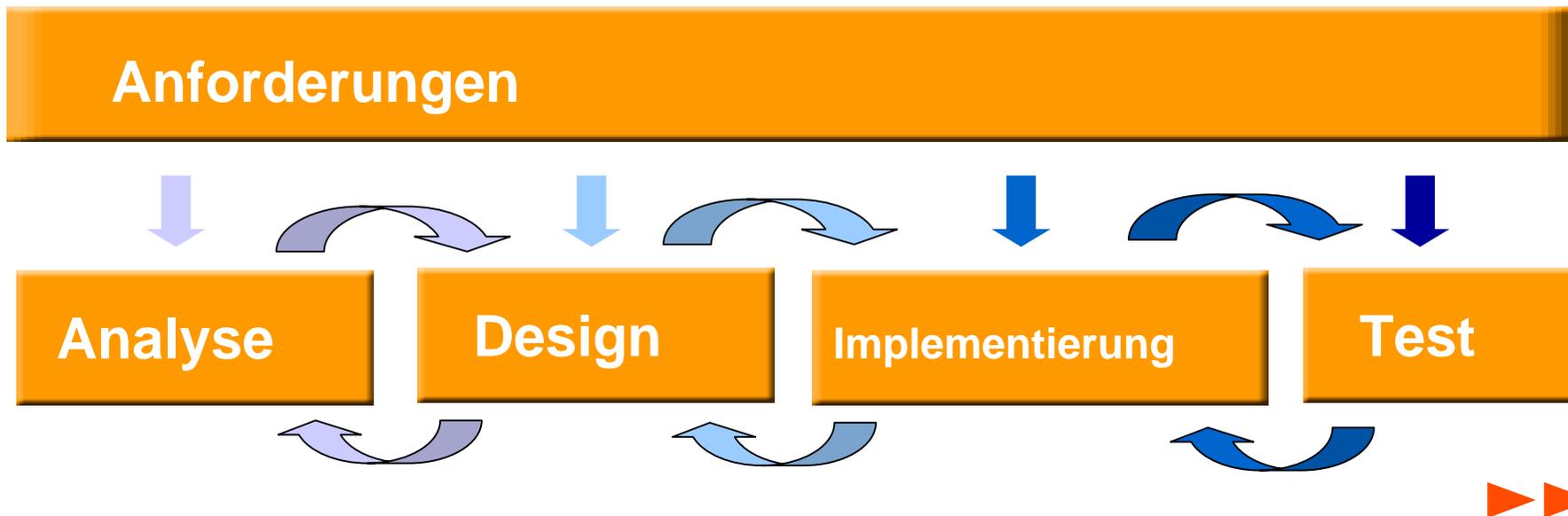


Technologie beherrschen ► Prozesse gestalten ► Lösungen schaffen ►►

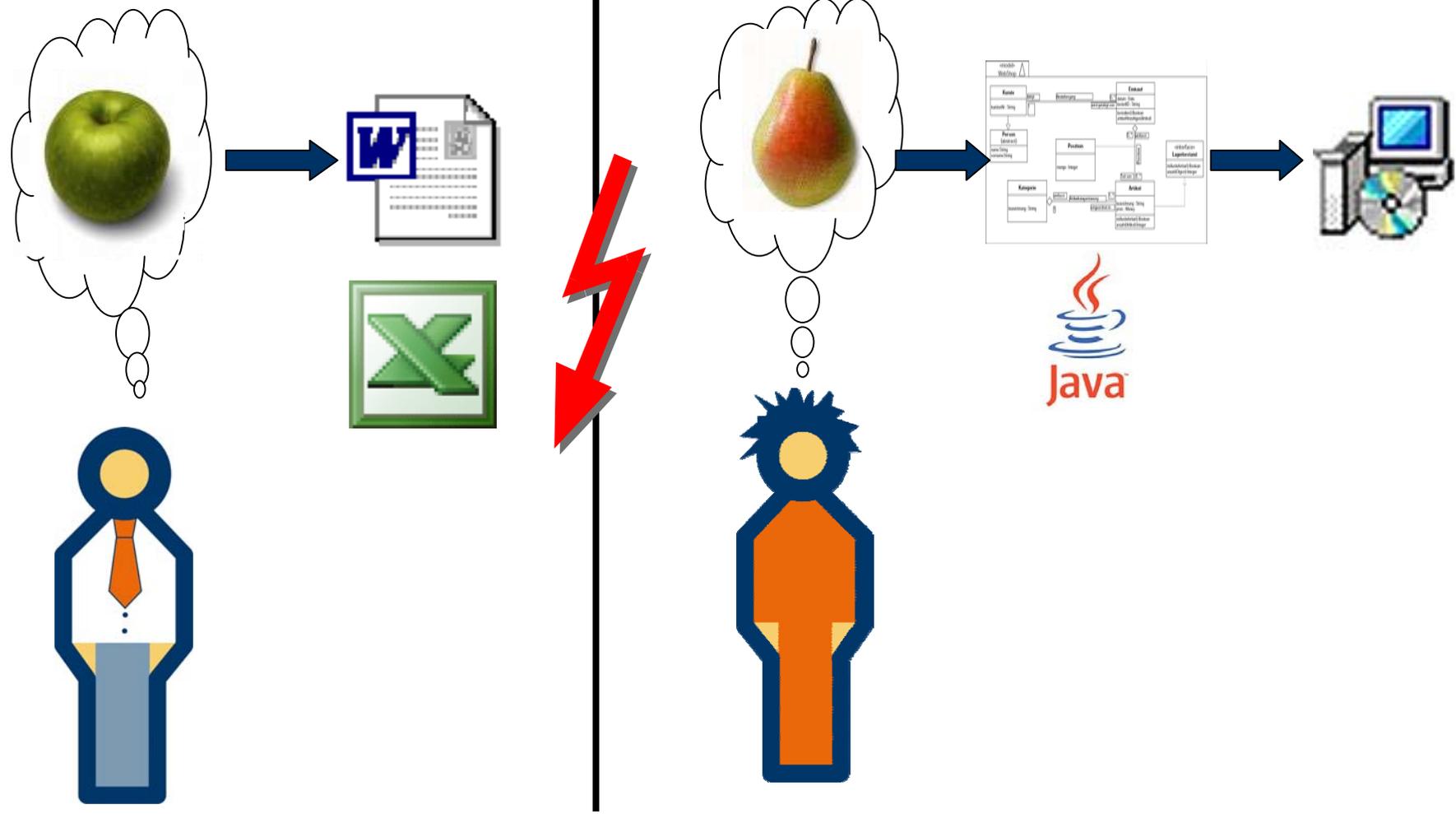
Einordnung der Anforderungsanalyse in den SW-Entwicklungsprozeß



- **Anforderungsanalyse und –Management haben einen wesentlichen Anteil am SW-Entwicklungsprozeß**
- **Projekte scheitern wegen unklaren Anforderungen: 70%, wegen technischer Probleme nur 7% (Nach Untersuchung der PA Consulting Group)**



Von den Anforderungen zum fertigen System - der „klassische Ansatz“





Probleme bei Verwendung des „klassischen Ansatzes“

- **Fehlende gemeinsame Sprache zwischen Entwicklung und Fachbereich**
- **Mangelnde Strukturierung der Anforderungen**
 - Kaum Möglichkeit der formalen Überprüfung
 - Schwieriges Versionshandling
 - Zuviel Freiheitsgrade bei der Ablage von Anforderungsdokumenten
- **Umständliche Erstellung von Anforderungsdokumenten wg. schlecht integrierter Werkzeuge**
- **Bezug zwischen Anforderungen und Implementierung fehlt (keine Nachverfolgbarkeit)**
 - Change Requests nur schwer umsetzbar
 - Projektplanung auf Basis der gegebenen Anforderungen schwierig
 - Enger Round-Trip zwischen Entwicklung und Anforderungsermittlung kaum möglich





Modell-Driven Requirements Engineering

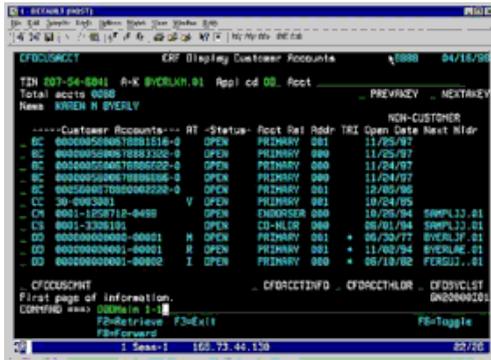
die nächste Generation des Requirements Engineering



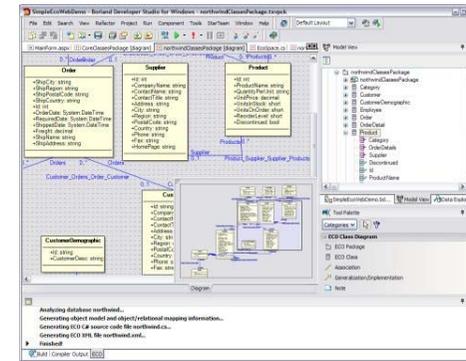
Technologie beherrschen ► Prozesse gestalten ► Lösungen schaffen ►►

Evolution in der Softwareentwicklung

Implementierung

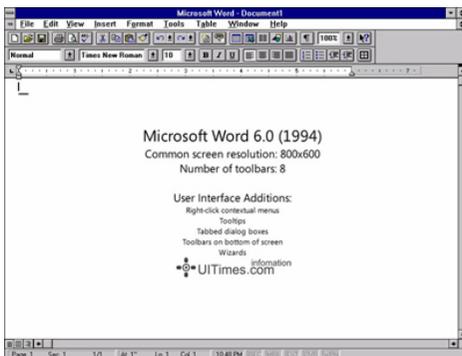


COBOL+Co.



Java + MDD

Anforderungsanalyse



Word 6.0



Word 2007

Wie kann man Requirements Engineering modernier gestalten?



- **Implementierungswerkzeuge als Vorbild:**
 - Integration (Integrierte Entwicklungsumgebungen)
 - Standardisierte Modellierung (UML, BPMN, ...)
 - Abstraktion und Struktur (Model Driven Development)
- **Wichtig: Balance zwischen Bewährtem und Neuem bewahren**
 - Alte Methoden nicht über Bord werfen
 - Textuelle Beschreibungen als zentraler Bestandteil
 - Änderungsfreundlich durch flexible Dokumente
 - Den RE nicht mit zu viel Neuem überfordern

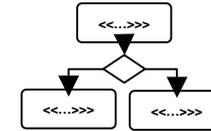
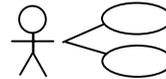
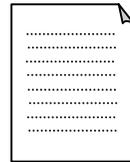
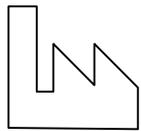


Kommunikationskompetenzen



fachlich

technisch



```
if{...}
```

Spezifikation

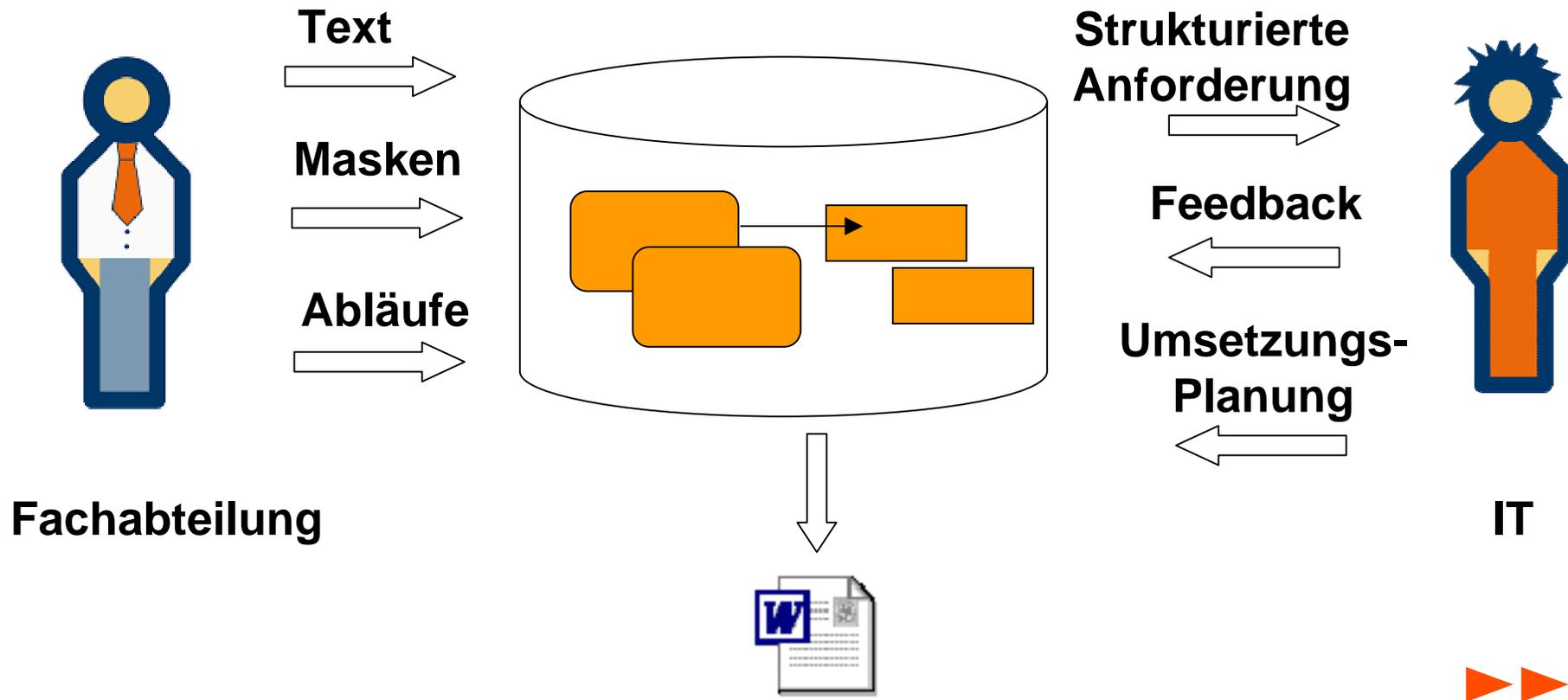
UML



Kommunikation über ein gemeinsames Modell



Anforderungen



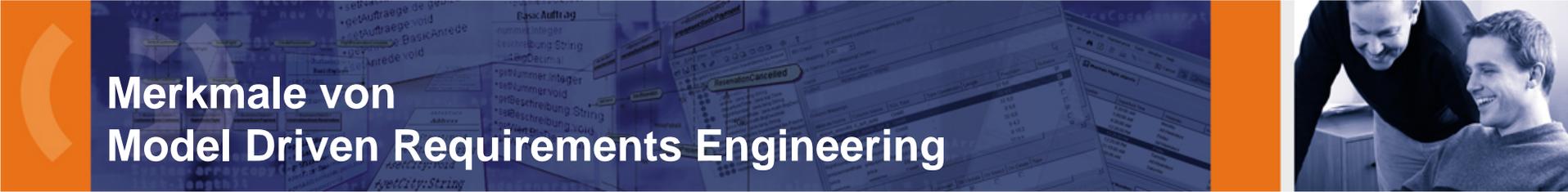


UML als gemeinsame Sprache zwischen Entwicklung und Fachbereich?



- **UML ist oft zu technisch**
 - Gehört nicht zur Kernkompetenz im Fachbereich
 - Textuelle Beschreibung kommt zu kurz
 - Vollständigkeit schwer zu erreichen
- **UML bietet aber auch gute Elemente für ein gemeinsames Verständnis**
- **Unser Vorschlag:**
 - UML + andere Diagramme (SysML)
 - Erweitert um formale Möglichkeiten Freitexte zu erfassen
 - -> MDRE



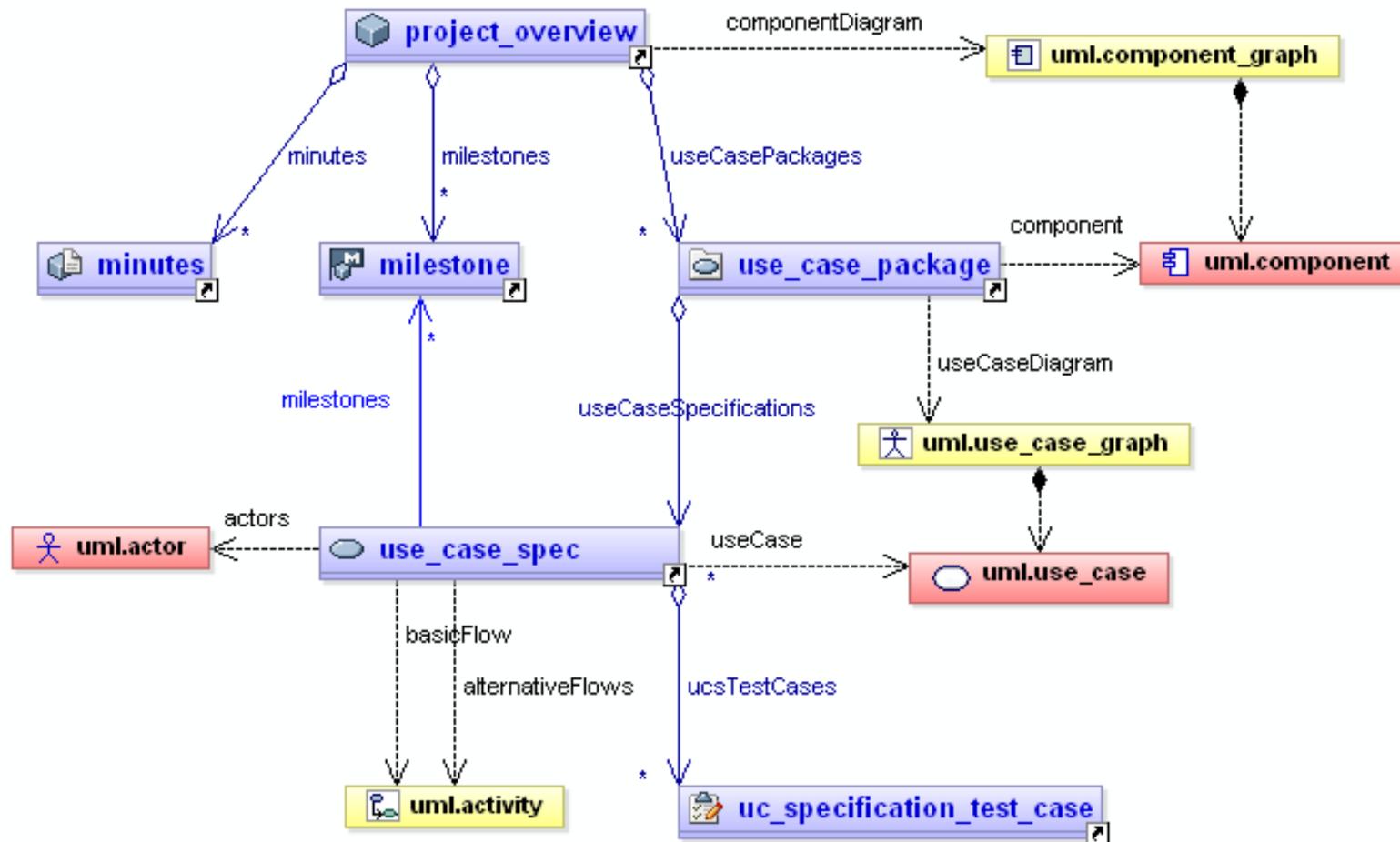


Merkmale von Model Driven Requirements Engineering

- **Modell/Metamodell**
Kombination UML & Text
Exakte Beschreibung von Dokumenttypen und Beziehungen (auch zu UML)
- **Strukturierte Erfassung / Methodische Unterstützung**
Führung des Benutzers durch Struktur
- **Flexible Erzeugung von Ausgabedokumenten**
Erzeugung von beliebigen Dokumenten (Pflichtenheft,...)
Reports und Metriken auf Basis der erfassten Artefakte
- **Integration**
Werkzeugintegration für direkten Austausch



Metamodell (Beispiel)





Möglichkeiten für eine Werkzeugunterstützung des MDRE-Ansatzes

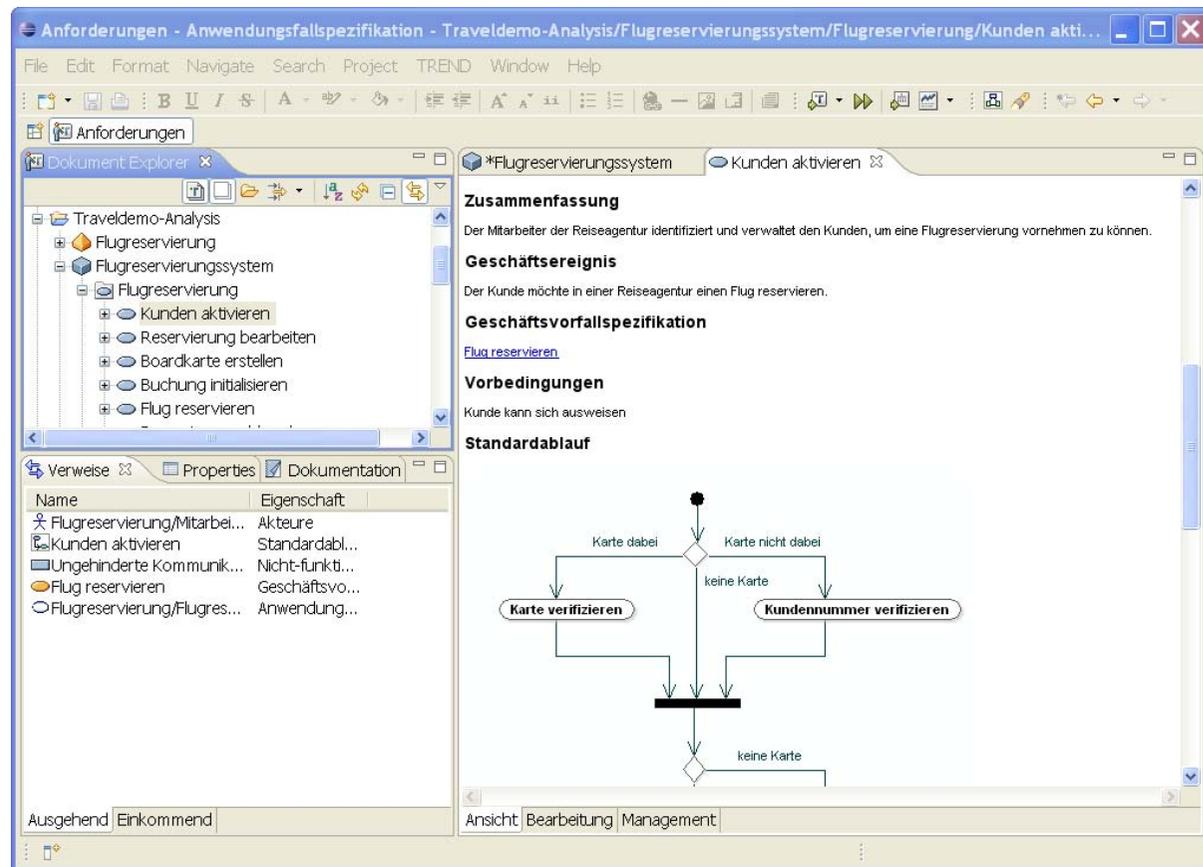
- **Option 1: Anpassung eines UML Werkzeuges**
 - Basierend z.B. auf Use Cases und Komponenten
 - Zusätzliche Eigenschaften
 - Alternativ: UML Erweiterungen (neue Begriffe und Bezeichner)
 - Generierung von Dokumenten aus den an den UML-Elementen hinterlegten Texten
- **Option 2: Spezialisierte Werkzeuge**
 - Beliebige Metamodelle (Modellierung)
 - Formularbasierte Erfassung
 - Verknüpfung von Text und Diagrammen
 - Integriert in Entwicklungsumgebungen



Beispiel für ein spezialisiertes MDRE – Werkzeug



- Eingebettet in Eclipse
- Gemeinsames Repository mit der Entwicklung (CVS, ...)
- Flexibel Anpassbare Dokumentenstruktur
- Kombination UML und Text
- ...



TREND/Analyst Community Edition
<http://www.gebit-community.de>





Vorzüge beim Einsatz von MDRE

.. und praktische Erfahrungen beim Einsatz

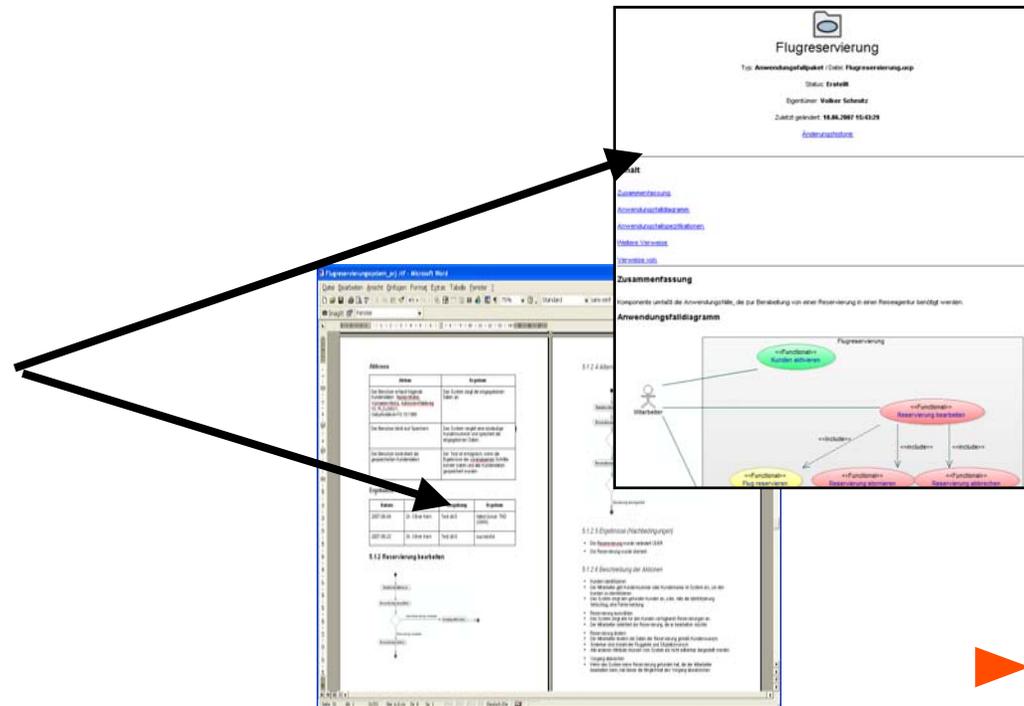
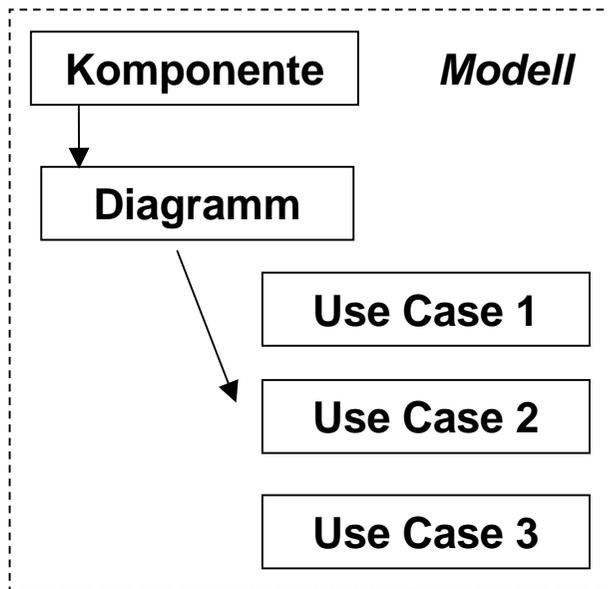


Technologie beherrschen ► Prozesse gestalten ► Lösungen schaffen ►►

Flexible Dokumenterzeugung



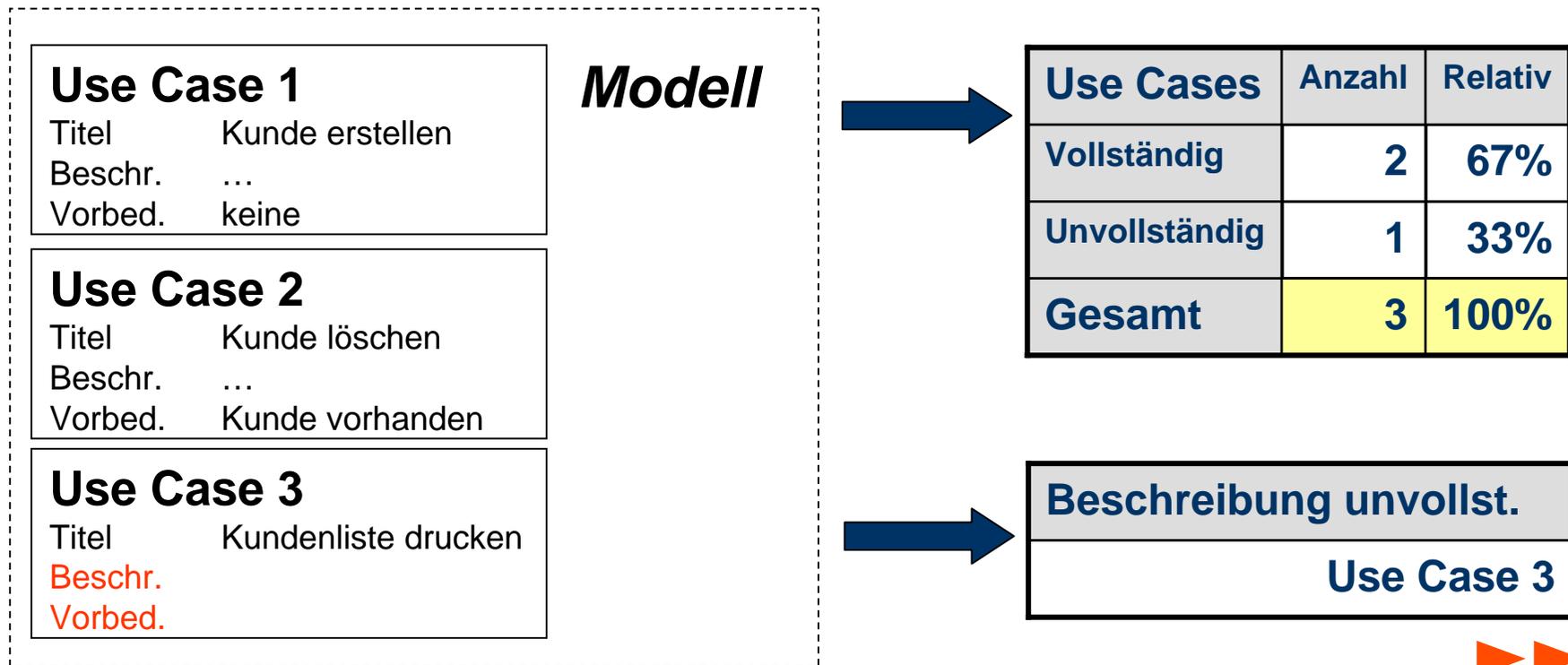
- Unterschiedliche Ergebnisdokumente aus gegebenen Anforderungen generierbar
- Einfache Integration von Bildern in Text
- Automatische Erstellung von Index und Verlinkung
-



Qualitätskontrolle Anforderungen



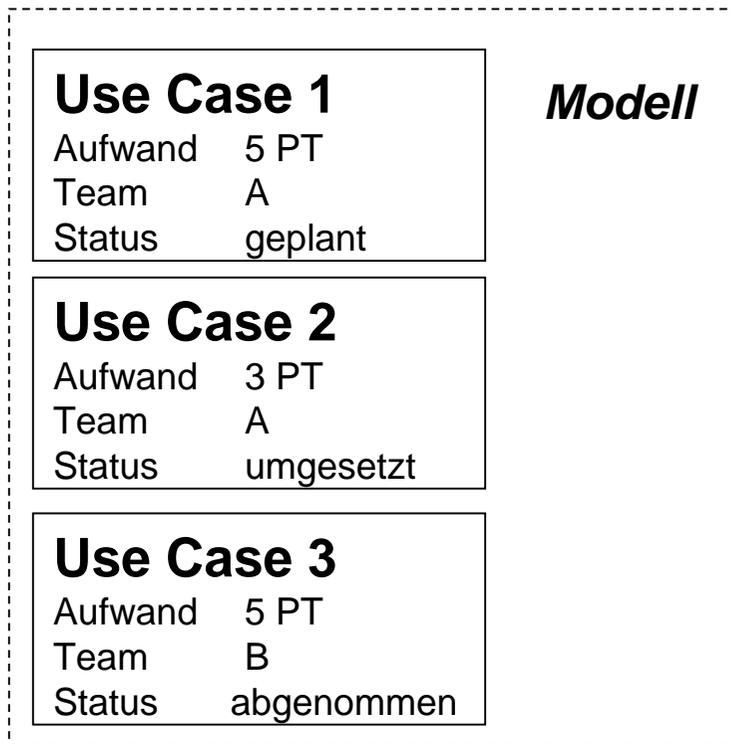
- **Metriken auf Basis des Modells einfach generierbar**
- **Auswertungsbeispiel 1: Qualitätskontrolle der erfassten Anforderungen**



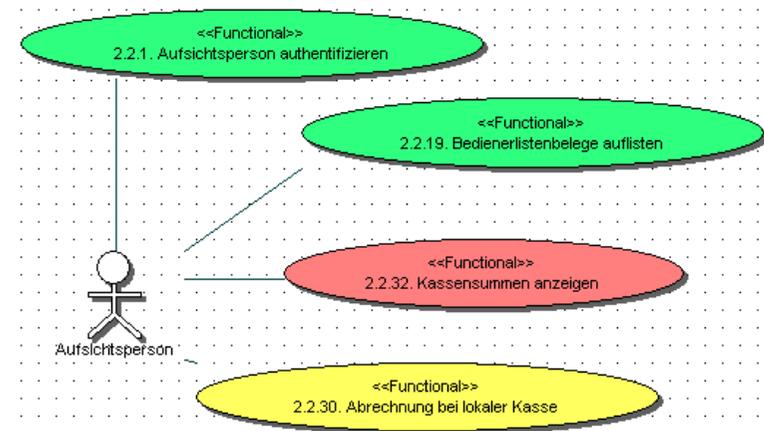
Projektstatusverfolgung direkt an Hand des Modells



- Auswertungsbeispiel 2: Projektstatus am Modell ermitteln**



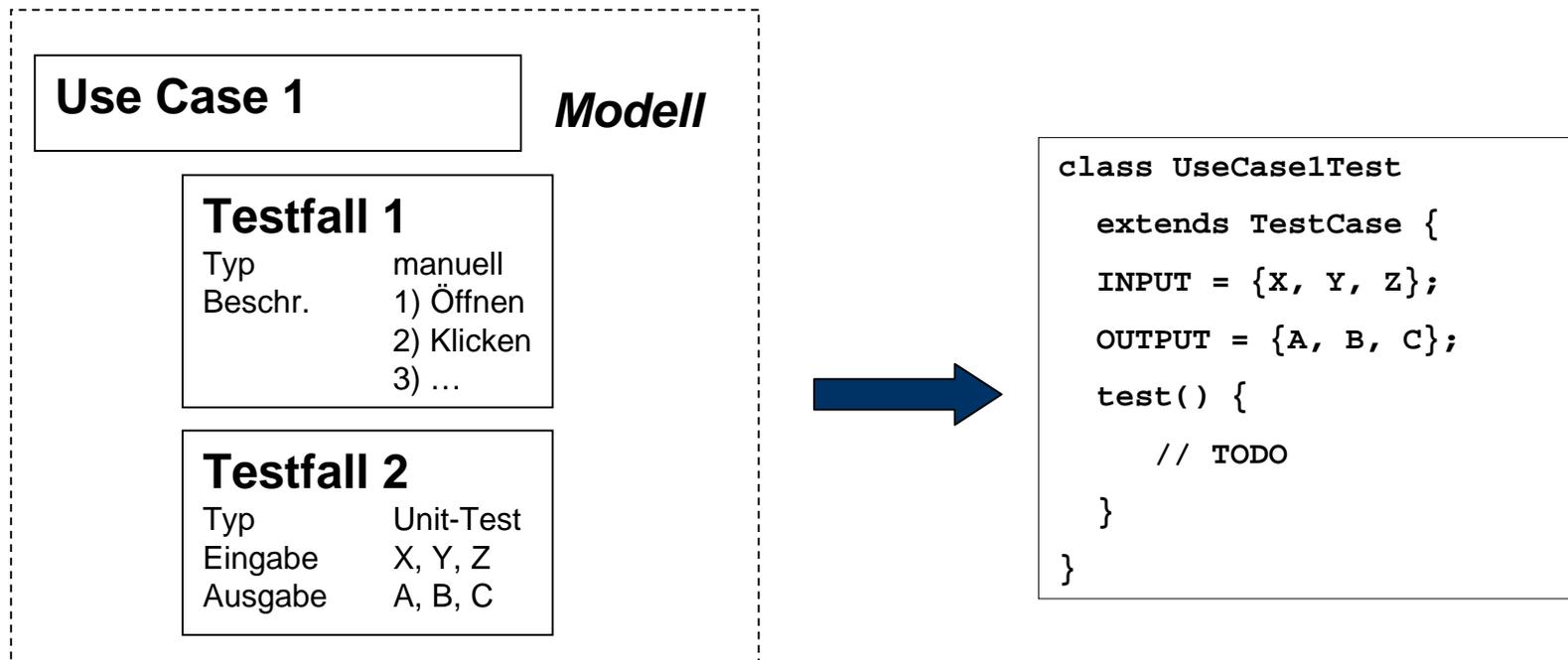
	Offen	Umges.	Abgen.
A	63%	37%	0%
B	0%	0%	100%
	38%	23%	38%



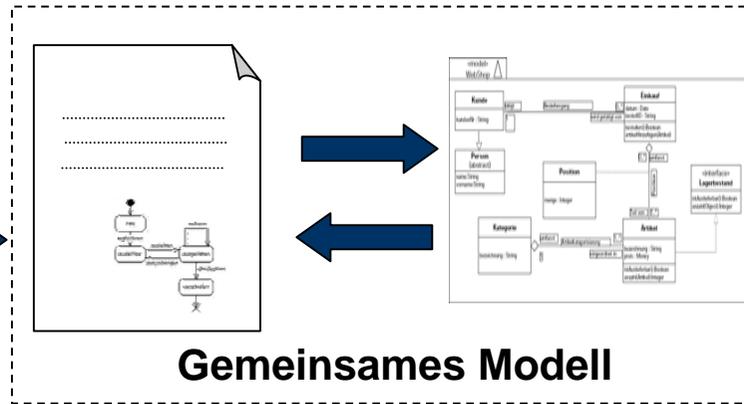
Integration von Anforderungsanalyse und Test



- Generierung von Testrahmen (JUnit, FitNesse, ...) direkt aus den Anforderungen



Fazit



- **Einheitlichere Dokumente**
- **Verbesserter Kommunikationsfluss**
- **Immer aktuelle Dokumentation**
- **Prozessneutral**

Fazit 2: Erfahrungen aus der Praxis

- **MDRE wurde von GEBIT Solutions bei mehreren Projekten eingesetzt**
- **Projektgrößen: 1 PM – 70 PJ**
- **Vorzüge:**
 - Auf Änderungswünsche konnte schnell reagiert werden
 - Projektdokumentation war auch am Ende noch aktuell
 - Alle Projekte „in-time & budget“ abgeschlossen
 - Initialer Einarbeitungsaufwand für den Fachbereich hat sich ausgezahlt!

