

IBM Rational Software Developer Platform  
Zürich, 16. September 2009

## Entwickeln Sie schon effizient oder codieren Sie noch?

Philip Zollinger, Geschäftsführer, EVOCEAN GmbH

Blaise Rey-Mermet, Principal Consultant – Requirements Engineering & Management, EVOCEAN GmbH

Johannes Scheier, Principal Consultant – Model Driven Development, EVOCEAN GmbH

## AGENDA

1. Einführung

2. Schlüsselprozesse für Effektivität und Effizienz

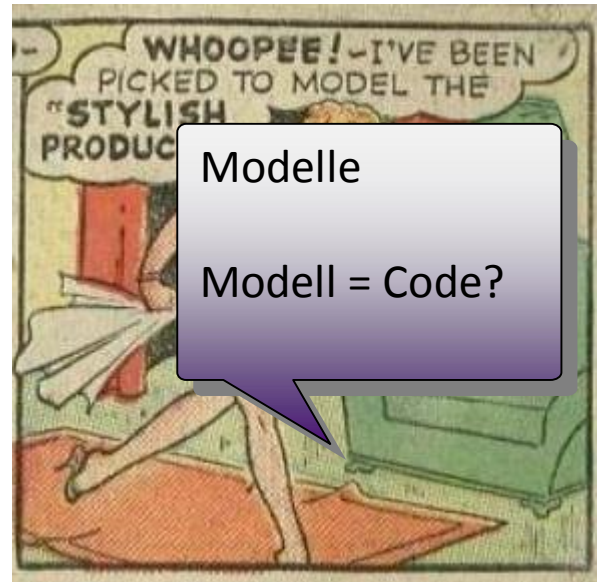
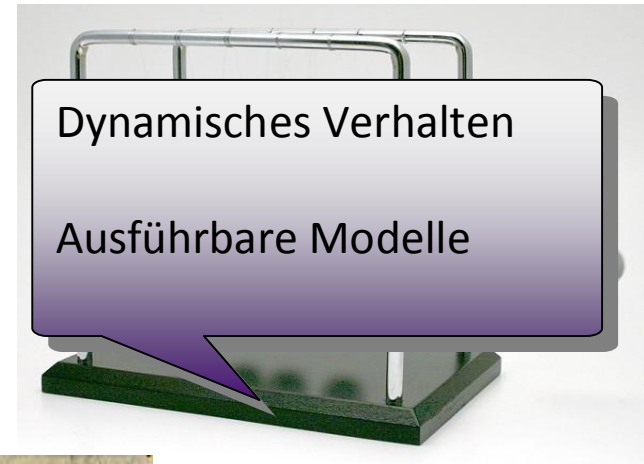
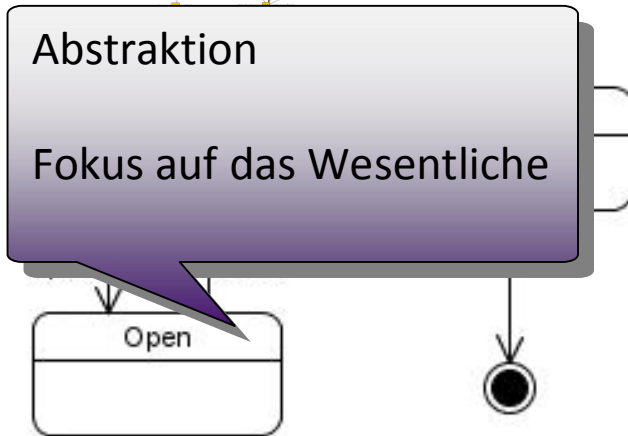
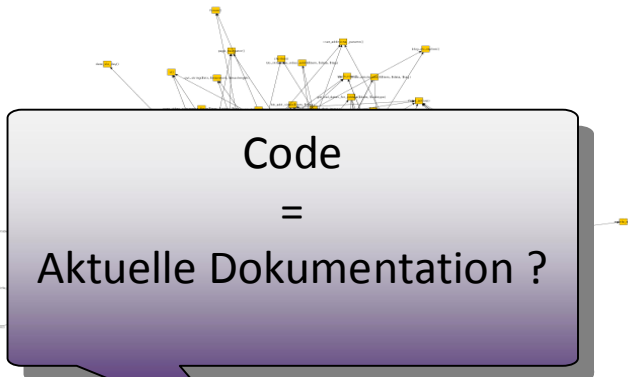
3. Anforderungsmanagement im agilen Kontext – Roadmap

4. Ausführbare Modellierung als Schlüsseltechnologie

5. Zusammenfassung

## 1. EINFÜHRUNG

## WAS STEHT HEUTE IM ZENTRUM ... ODER EBEN NICHT?



## BEWÄHRTE LÖSUNGEN AUS DER PRAXIS



PHONAK

life is on

Schmidhauser



Schindler

HAMILTON  
MEDICAL

Das richtige Produkt zum richtigen Zeitpunkt zu den geplanten Kosten und geforderten Qualität auf den Markt zu bringen.

## HERAUSFORDERUNGEN



### Wechsel

- Anforderungen
- Technologien
- Mitarbeiter



### Kommunikation

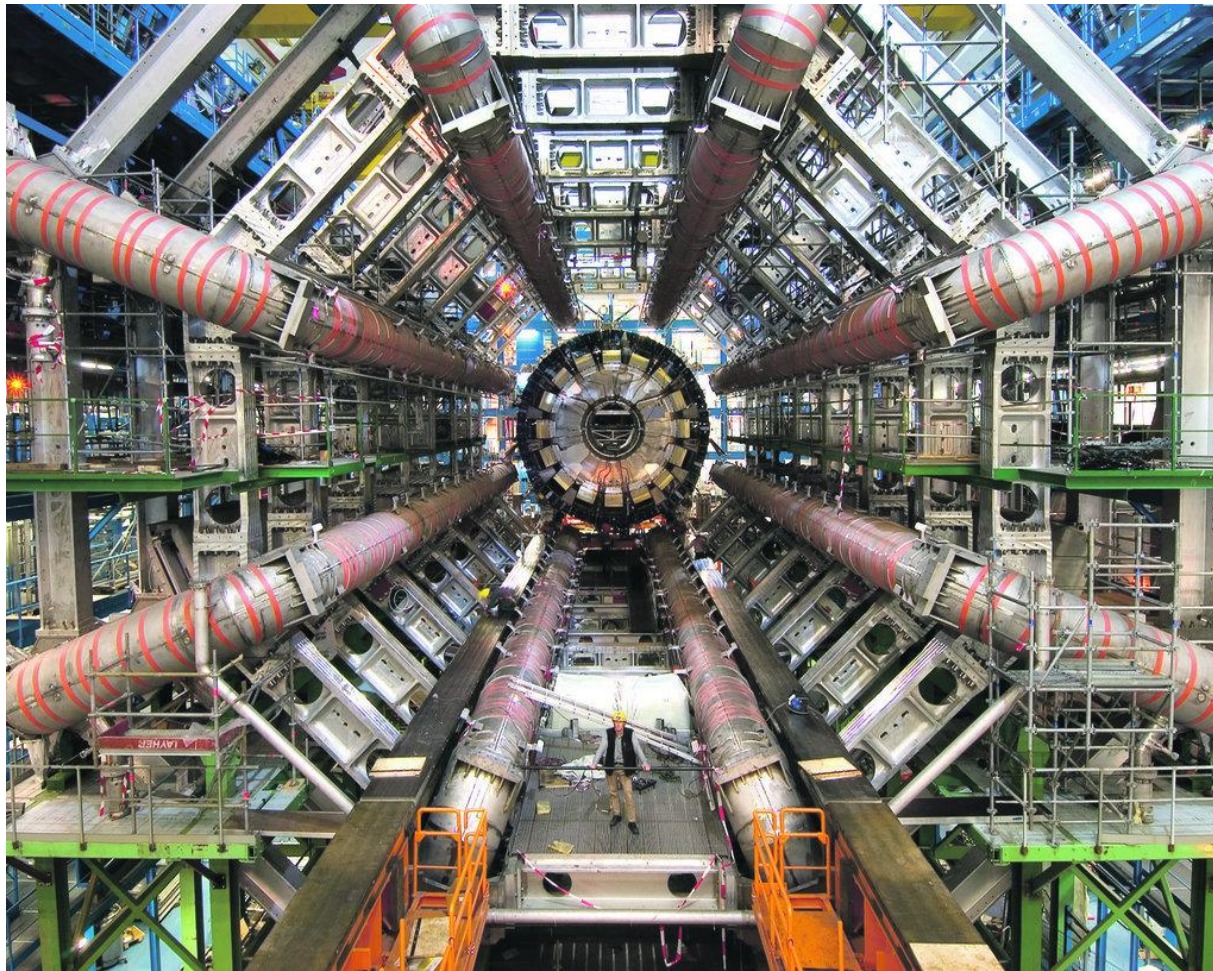
- Verstehen wir was der Kunde will?
- Verstehen wir die Dokumentation, das Design, den Code?
- Erwartungen an die Zusammenarbeit?



### Kosten & Zeitdruck

- Time to market
- Integration von neuen Funktionalitäten
- Wartung, Fehlersuche

## PLUS STEIGENDE KOMPLEXITÄT

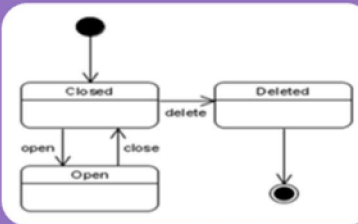


## KONSEQUENZEN



### Schnelles Feedback

- Kurze Iterationen
- Lauffähige Iterationen
- Agile Prozesse sind wichtig



### Konzentration auf das Wesentliche

- Abstraktion sinnvoll einsetzen
- Effektivität steigern
- Modellieren (Graphik) statt Codieren (Text)



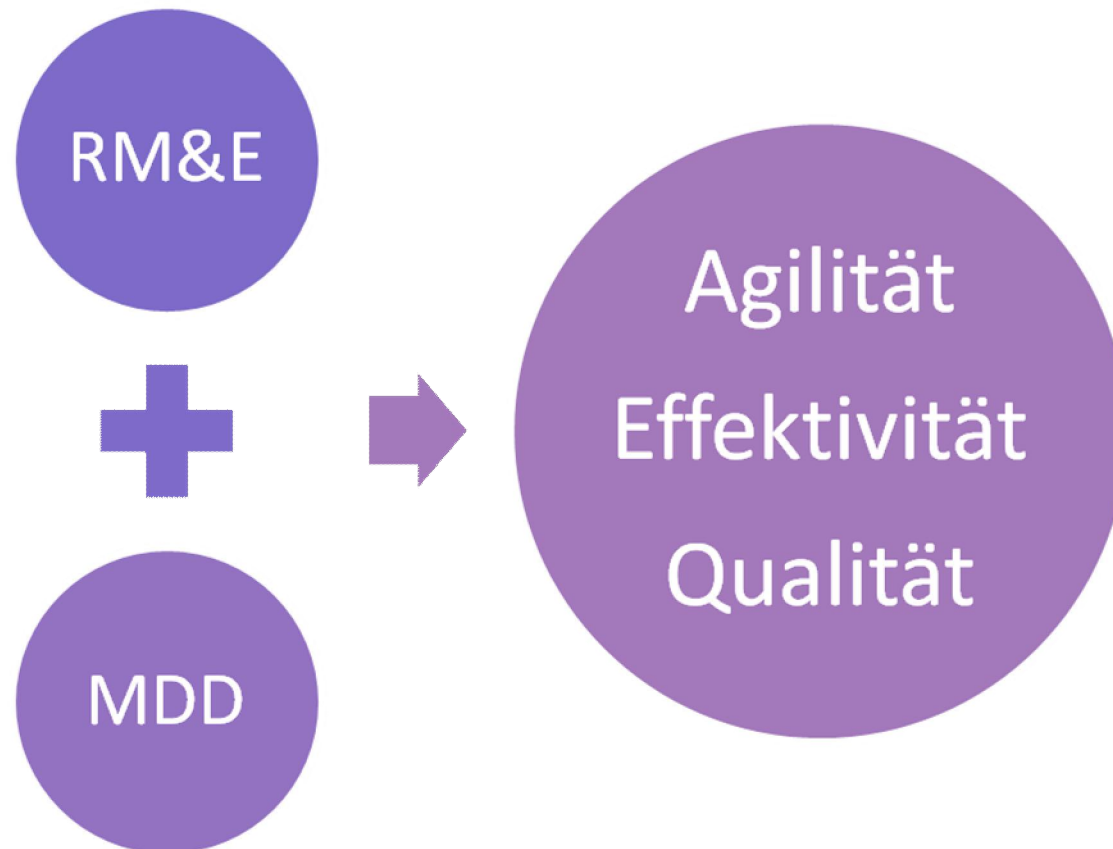
### Automatisation

- Durchgängigkeit in der Entwicklungsumgebung eliminiert Medienbrüche
- Modell und Code sind synchron (Codegenerierung)
- Ausführbare Modelle um dynamisches Verhalten zu verstehen



## 2. SCHLÜSSELPROZESSE FÜR EFFEKTIVITÄT UND EFFIZIENZ

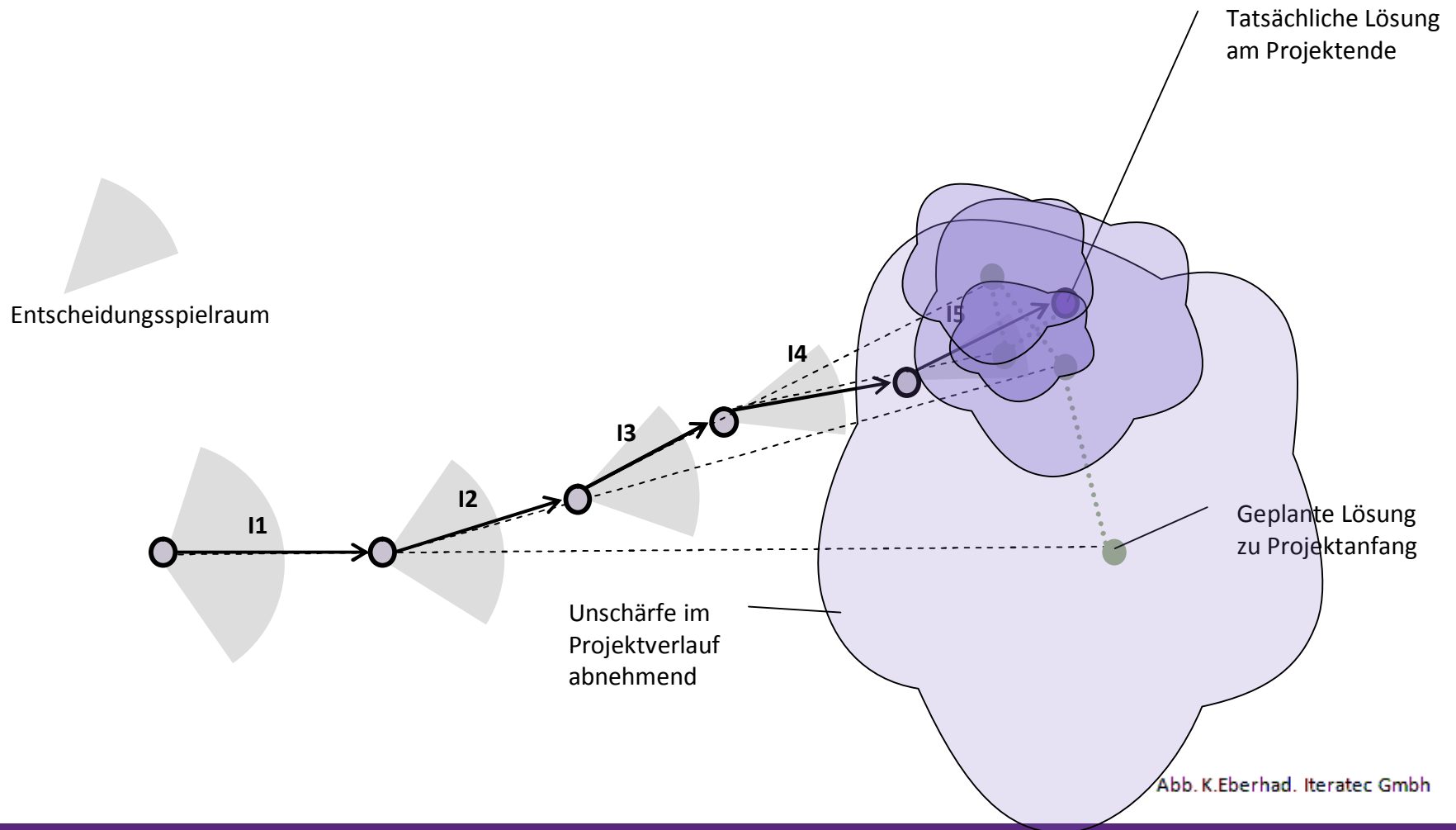
## UNSER LÖSUNGSANSATZ



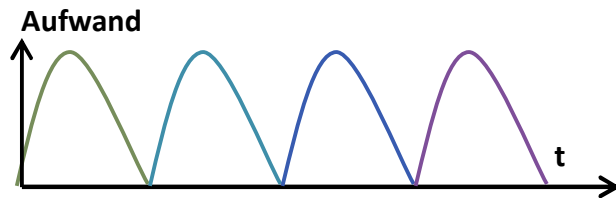
- RM&E – Requirements Management & Engineering
- MDD – Model Driven Development

## LÖSUNG TEIL 1: ITERATIONEN JAGEN BEWEGLICHE ZIELE

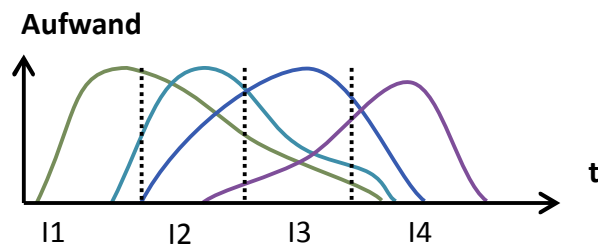
- Frühes Feedback – passt sich an bewegliche Ziele und Änderungen an



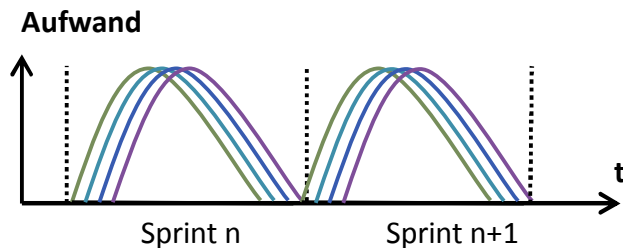
## LÖSUNG TEIL 2: MÖGLICHT AGILE PROZESSE



- **Im V-Modell** werden die diversen Entwicklungsdisziplinen mehrheitlich sequentiell durchgeführt.

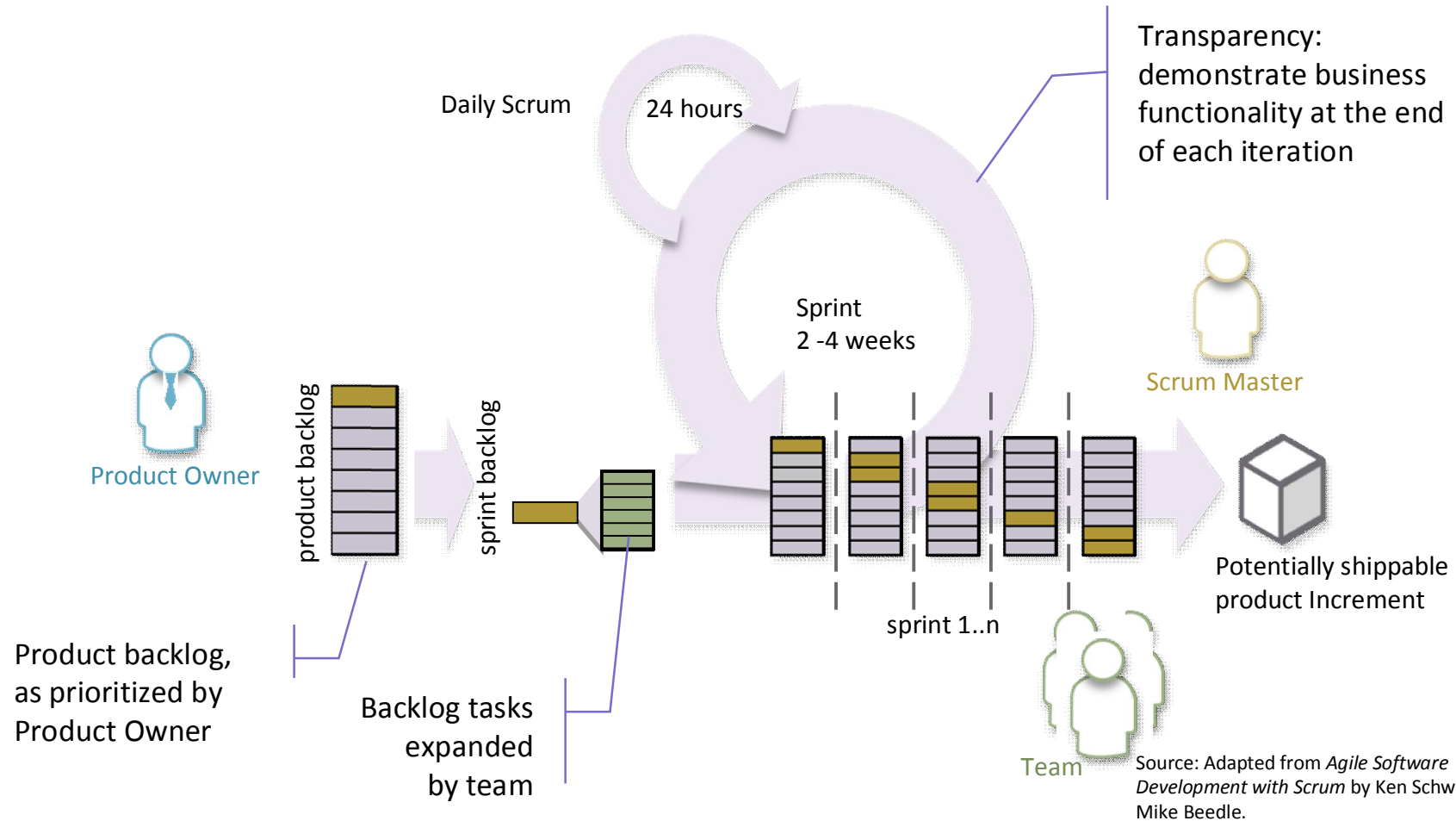


- **In iterativen Prozessen**, wie z.B. Unified Process, werden die Disziplinen überlappend während der ganzen Projektzeit durchgeführt. Das Schwergewicht ändert sich vom Anforderungsmanagement bis zur Verifikation.

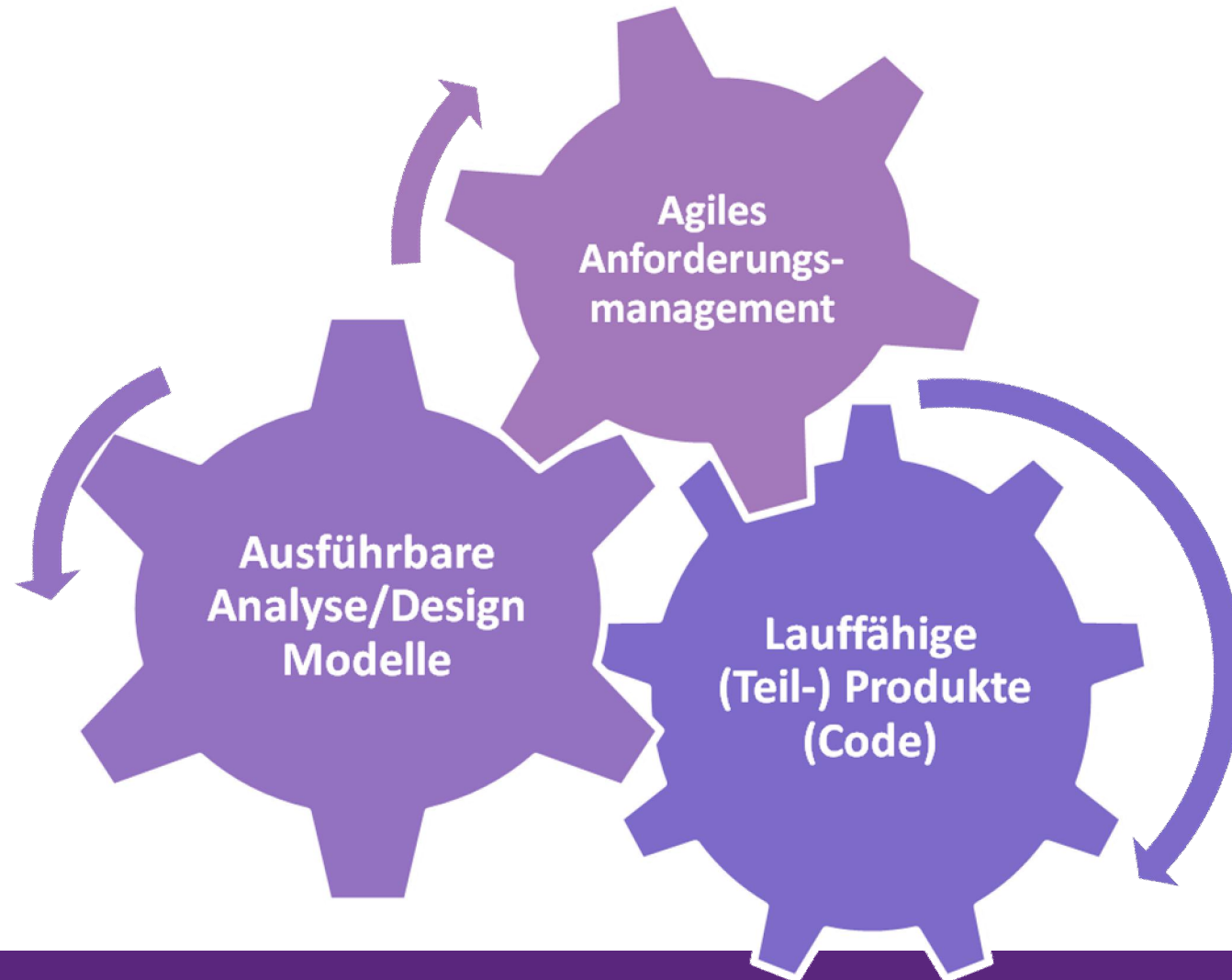


- **In agilen Prozessen** werden alle Disziplinen während dem gleichen Sprint vom Team gemeinsam durchgeführt.

## AN AGILE PROCESS EXAMPLE: SCRUM (PROCESS OVERVIEW)



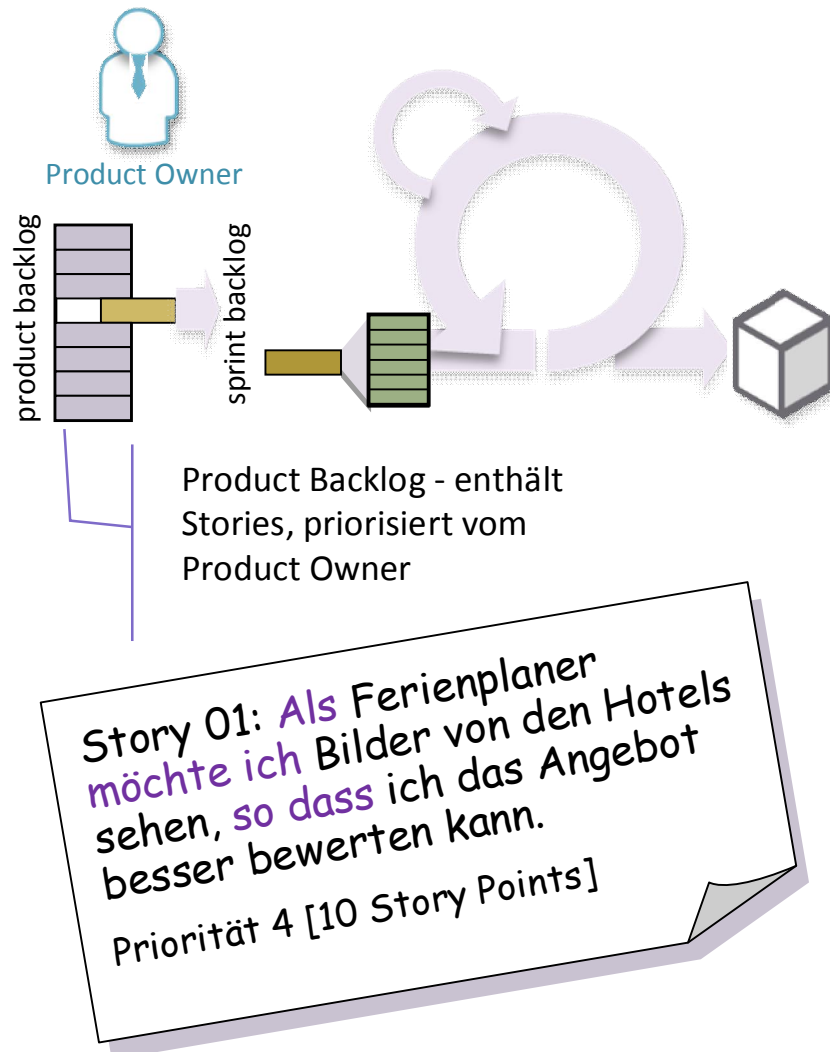
## LÖSUNG TEIL 3: INTEGRATION ANFORDERUNGEN – ANALYSE – DESIGN – CODE – TEST



## 2. ANFORDERUNGSMANAGEMENT IM AGILEN KONTEXT – ROADMAP

- Anforderungen als Stories aufnehmen
- Anforderungen priorisieren
- Anforderungen detaillieren – darf man das in agilen Projekten?
- Anforderungen mit Prototypen und Modellen detaillieren

# 1. ANFORDERUNGEN ALS USER STORIES AUFNEHMEN



- In SCRUM gibt es nur ein **Product Backlog** mit allen Aufgaben für das Team. Es enthält **User Stories**.
- Eine **User Story** ist eine kurze, in Alltagssprache formulierte Beschreibung von etwas Wertvollem, das das System für den Anwender machen muss.
- Stories sind normalerweise so beschrieben, dass der Wert für den Anwender klar ist, zum Beispiel in der Form:

**Als** <Role> **möchte ich** <Tätigkeit> **so dass** <Wert>



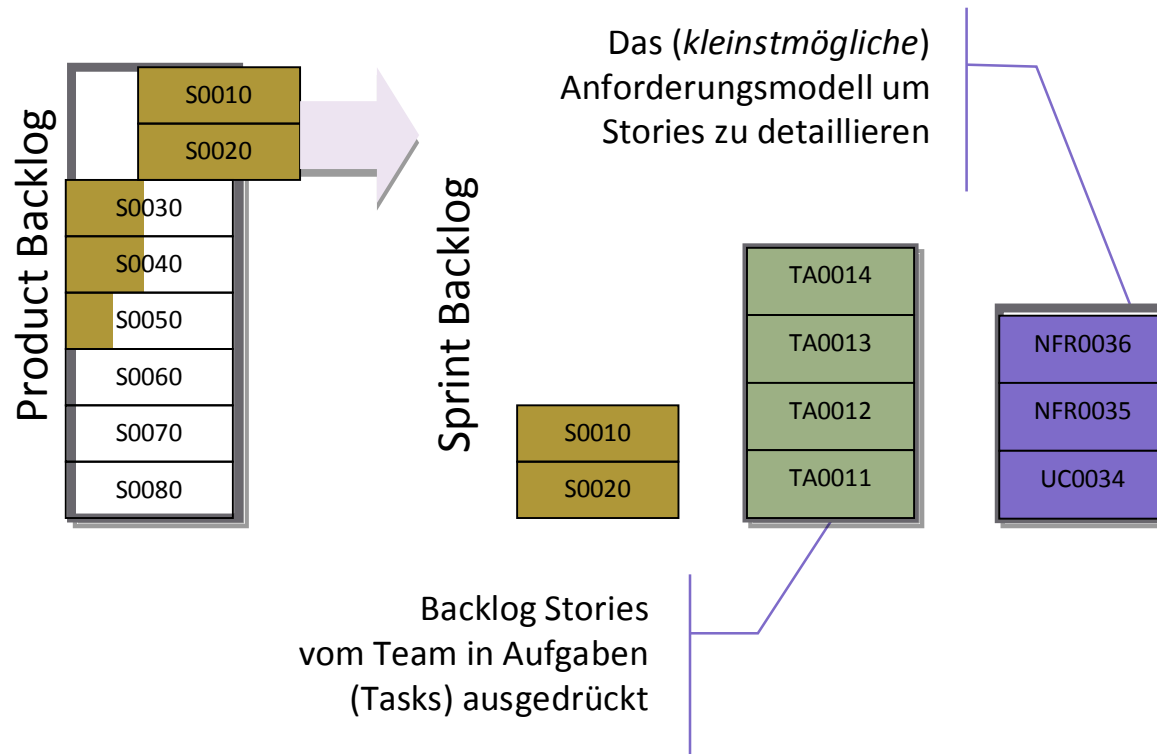
## 2. ANFORDERUNGEN WERDEN VOM PRODUCT OWNER PRIORISIERT

- Die Stories werden vom **Product Owner** priorisiert.
- Vor jedem Sprint werden die Stories bei Bedarf neu priorisiert.
- Man verwendet am besten relative Schätzmethode. Die Schätzwerte werden schrittweise verbessert.

Beispiel: Product Backlog:

Priorität	Stories	Story Point Schätzung
1	Als Gast möchte ich ein Reiseangebot reservieren.	20
2	Als Gast möchte ich ein Reiseangebot retournieren.	5
3	Als Gast möchte ich die Reservationstermine ändern.	3
4	Als Hotelmitarbeiter kann ich RevPAR Berichte (revenue-per-available-room) erstellen.	8
5	...	13

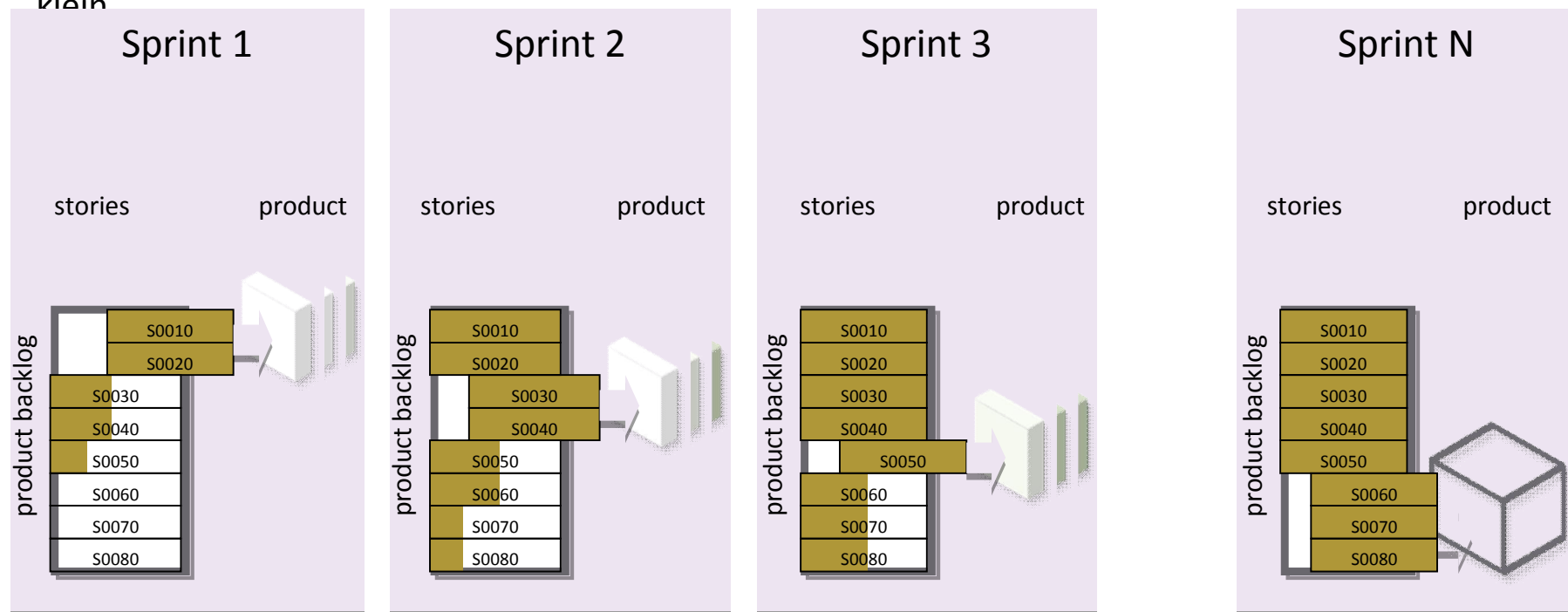
### 3. ANFORDERUNGEN MIT USE CASES UND NICHT-FUNKTIONALEN ANFORDERUNGEN DETAILLIEREN



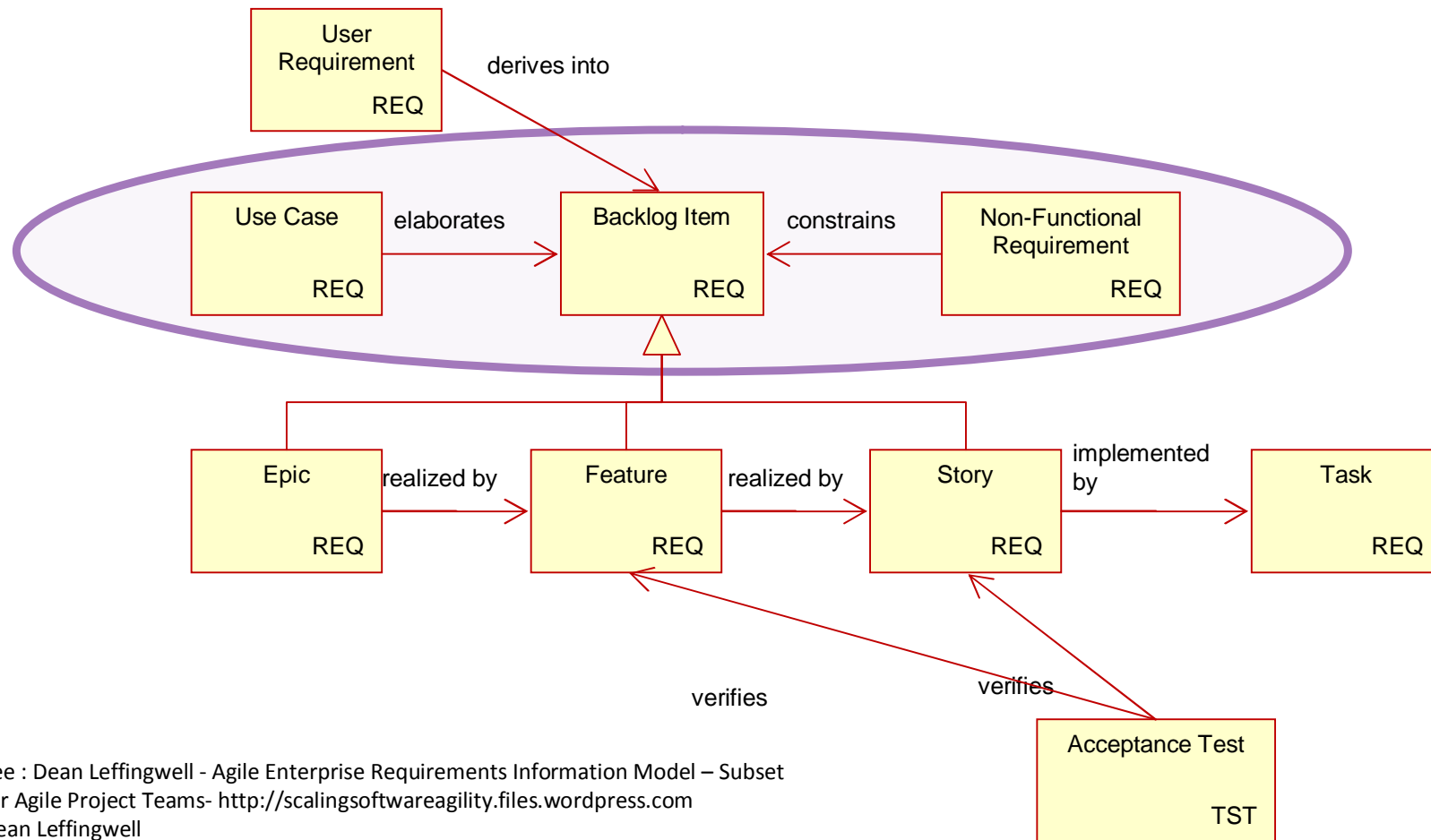
# ANFORDERUNGEN DETAILLIEREN – DARF MAN DAS IN AGILEN PROJEKTEN?

**NUR WAS IM SPRINT ENTWICKELT WIRD, WIRD IM DETAIL SPEZIFIZIERT**

- Detaillieren Sie nur diejenigen Anforderungen, welche sich in der Entwicklung befinden.
- Die Spezifikation ist erst am Projektende vollständig.
- Der Kunde bekommt oft und früh Feedback vom Projekt.
- Anforderungen werden oft geändert. Für die, welche nicht in Entwicklung sind, ist die Auswirkung *klein*.

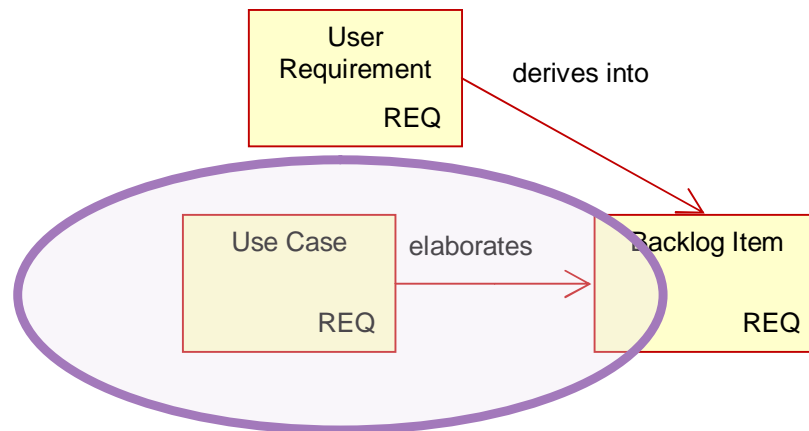


## AGILEN ANFORDERUNGEN: EIN MODELL



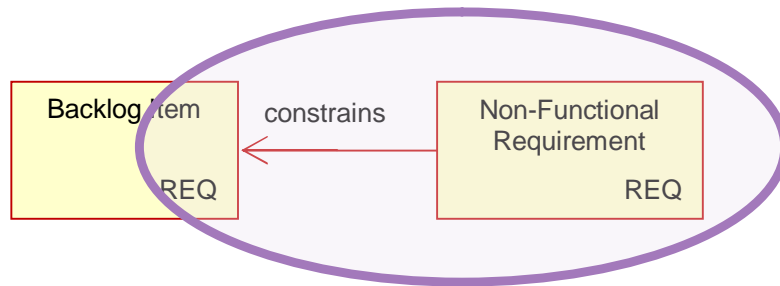
See : Dean Leffingwell - Agile Enterprise Requirements Information Model – Subset for Agile Project Teams- <http://scalingsoftwareagility.files.wordpress.com>  
Dean Leffingwell

## WIESO USE CASES? AGILE ANFORDERUNGEN SIND KUNDENORIENTIERT



- Die Anforderungen müssen die Kundenbedürfnisse genau widerspiegeln, insbesondere
  - Mehrdeutigkeit und Unklarheiten vermeiden,
  - der **Wert** muss bekannt sein.
- Ein Use Case erlaubt es, uns **auf die Funktionalität zu fokussieren**, bevor man die technischen Details des Produkts formuliert.
- Es verbessert die Kommunikation zwischen dem Product Owners und dem Entwicklungsteam.

# DIE STORIES MIT NICHT-FUNKTIONALEN ANFORDERUNGEN PRÄZISIEREN



## Nicht-funktionale Anforderungen

- definieren eine **Qualität**, welche das Produkt für die Anwender haben muss.
- Beschreiben **“wie gut”** (in welcher Qualität); funktionale Anforderungen definieren, welche Funktionen das Produkt hat.
  - Performance, security, reliability
  - Norms, standards
  - Design constraints

## 4. WEITER MIT FEATURES, MODELLEN UND PROTOTYPEN

Working modes for

appliance group (audio, video, etc..)

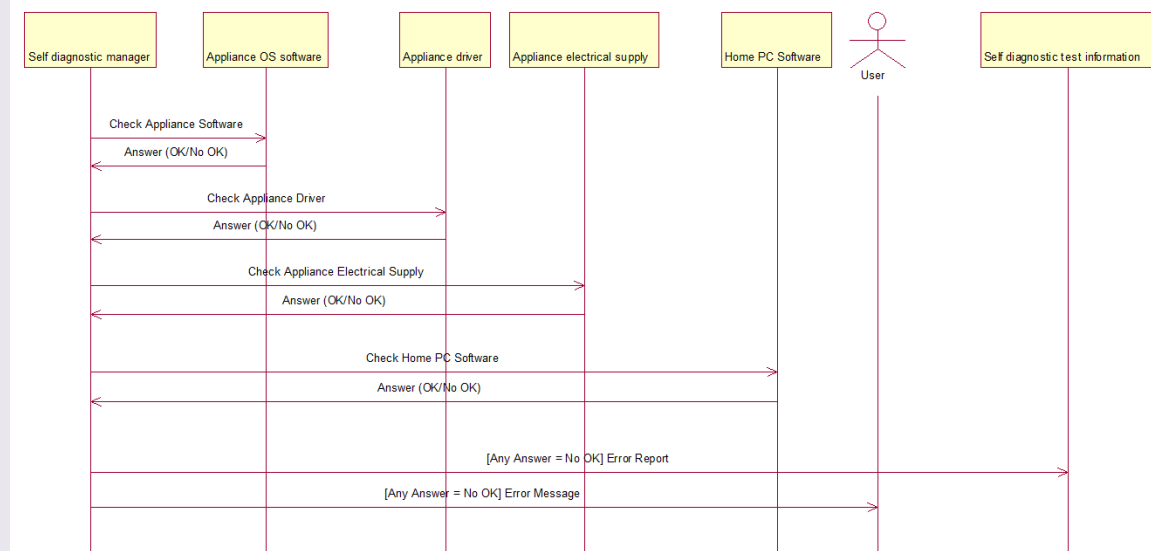
Select appliance

state (on, off, sleep)

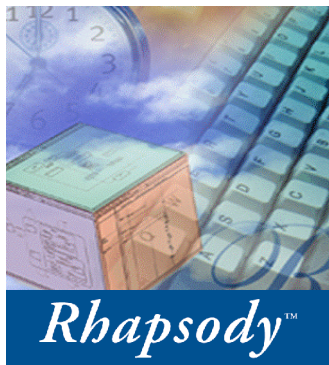
Request detail view

List of selected appliances with detailed working modes inc.  
- actual state  
- IP address  
- availability  
- access restrictions

- Zum Beispiel einen Prototyp der Benutzerschnittstelle skizzieren, um ein Szenario zu erläutern.
- Prüfen ob die Abläufe in den Anforderungen mit dem Prototyp realisierbar sind.
- Ein Modell erlaubt eine detaillierte Überprüfung der Interaktion mit dem Produkt.



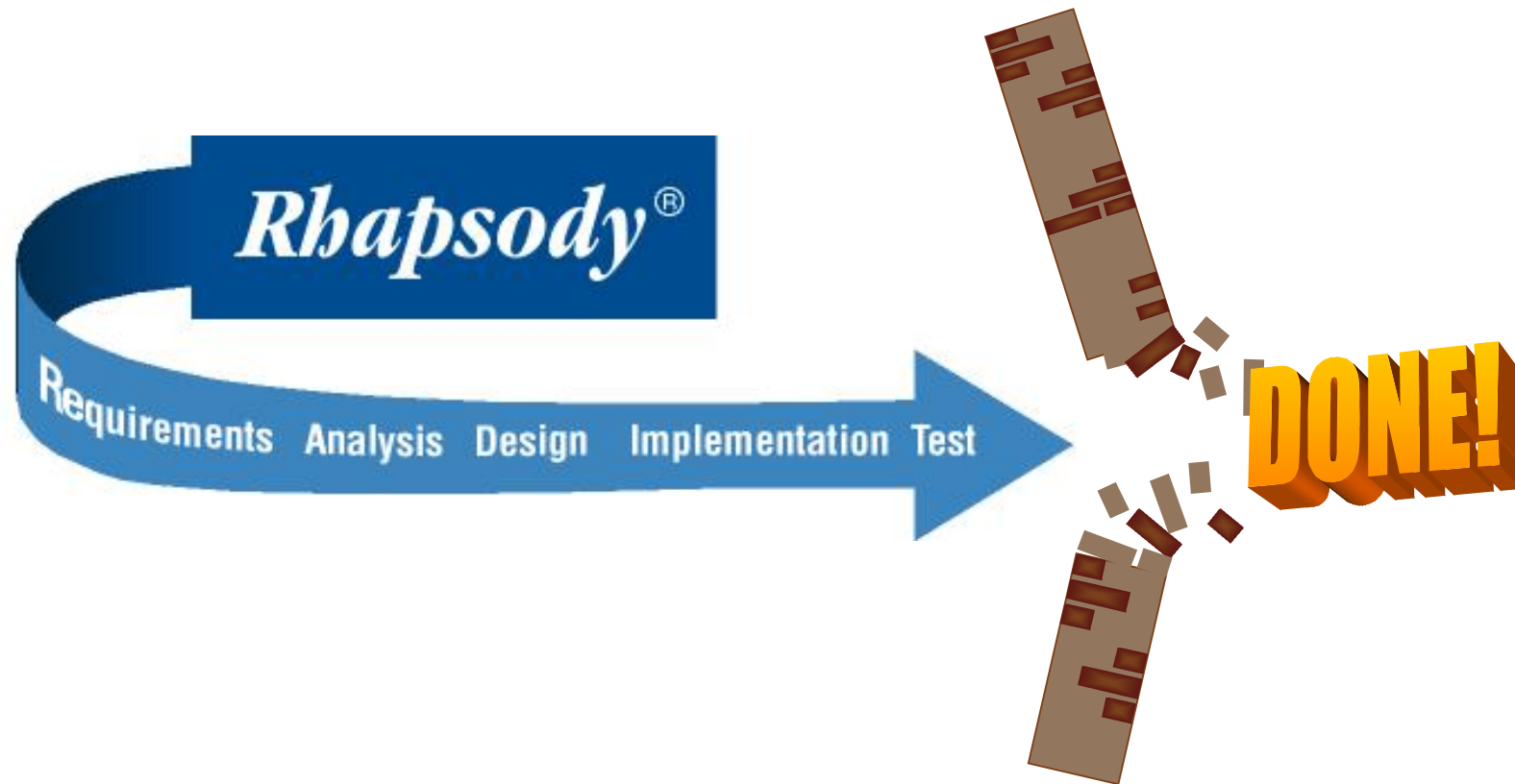
## AUSFÜHRBARE MODELLIERUNG ALS SCHLÜSSELTECHNOLOGIE



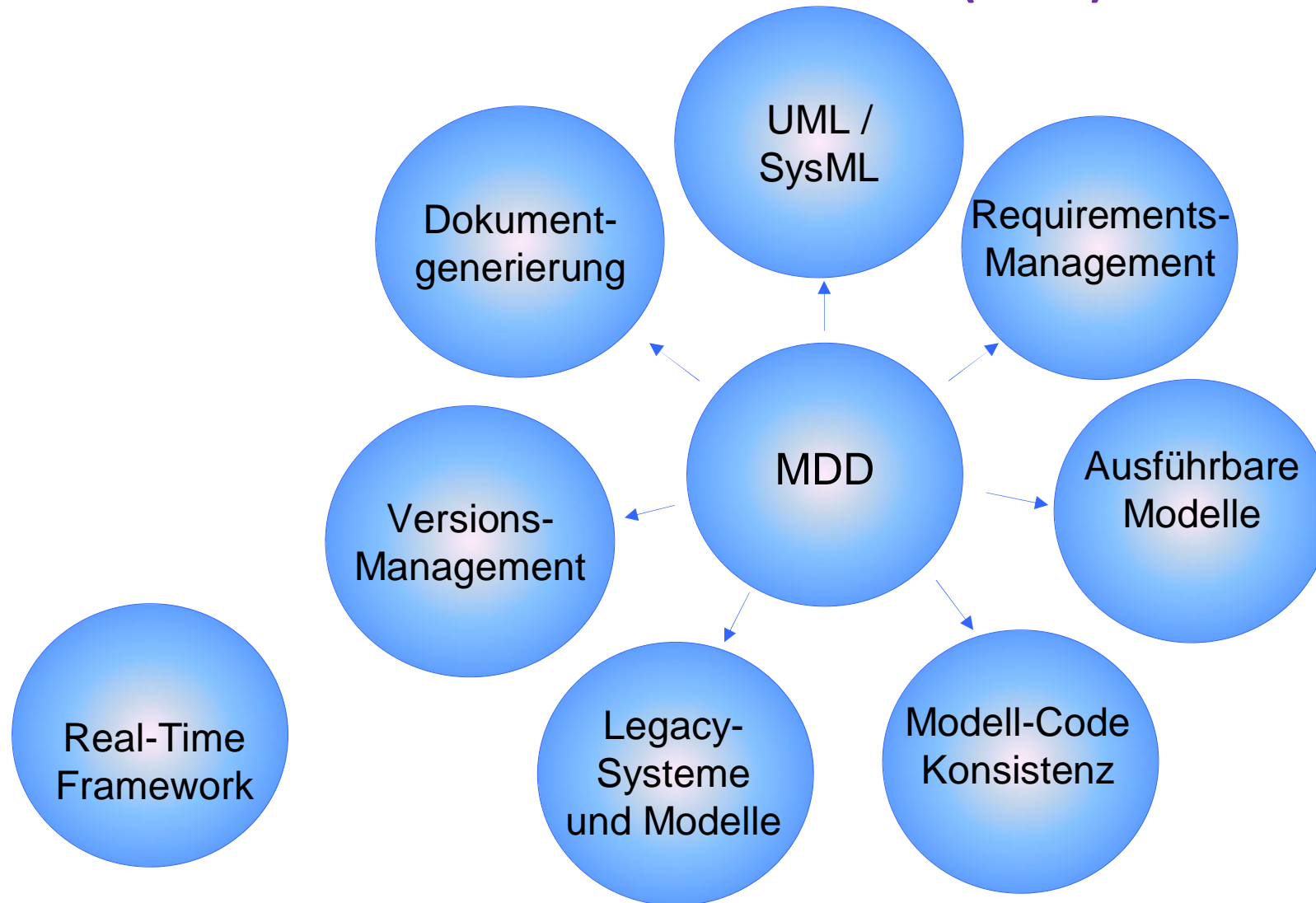
- Einstieg anhand von einem Beispiel mit Rhapsody
- Wie Rhapsody Model Driven Development unterstützen

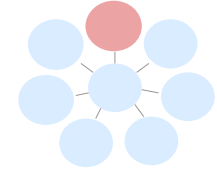


## DEMO TIME



## ÜBERSICHT : MODEL DRIVEN DEVELOPMENT (MDD)



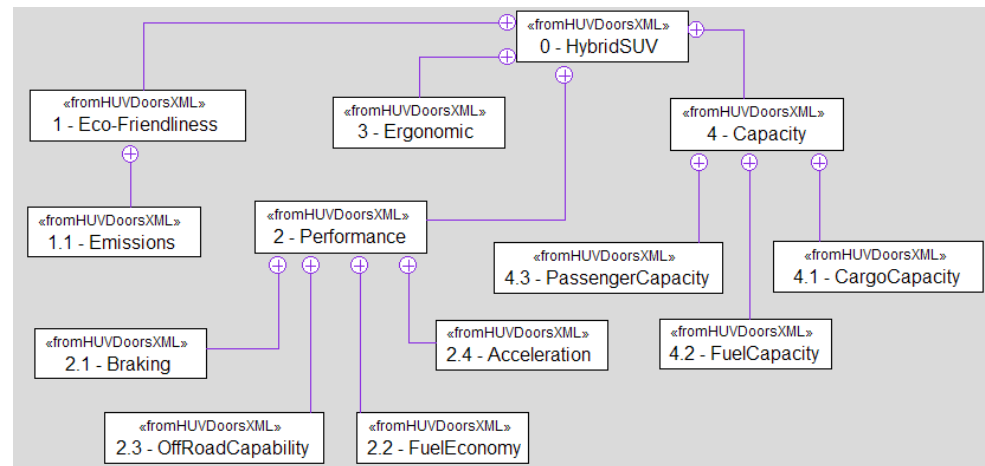


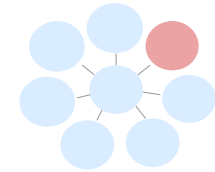
## UML / SysML

- Rhapsody unterstützt UML und SysML
- SysML ist eine Anpassung von UML für Systems-Engineering (ein Ansatz um komplexe technische Systeme zu entwickeln)
- Unterstützung von SysML-Diagrammen
  - Requirements: **Requirements- und Use-Case-Diagramm**
  - Struktur: **Block-Definition und Internal-Block-Diagramm**
  - Verhalten: State-, Activity- und Sequence-Diagramm
  - Constraints: **Parametric-Diagramm**
- Integrierte Umgebung zur Modellierung von Requirements und Design
- Mehr als Modellierung...
  - Simulation von SysML-Modellen
  - Systemtests für SysML

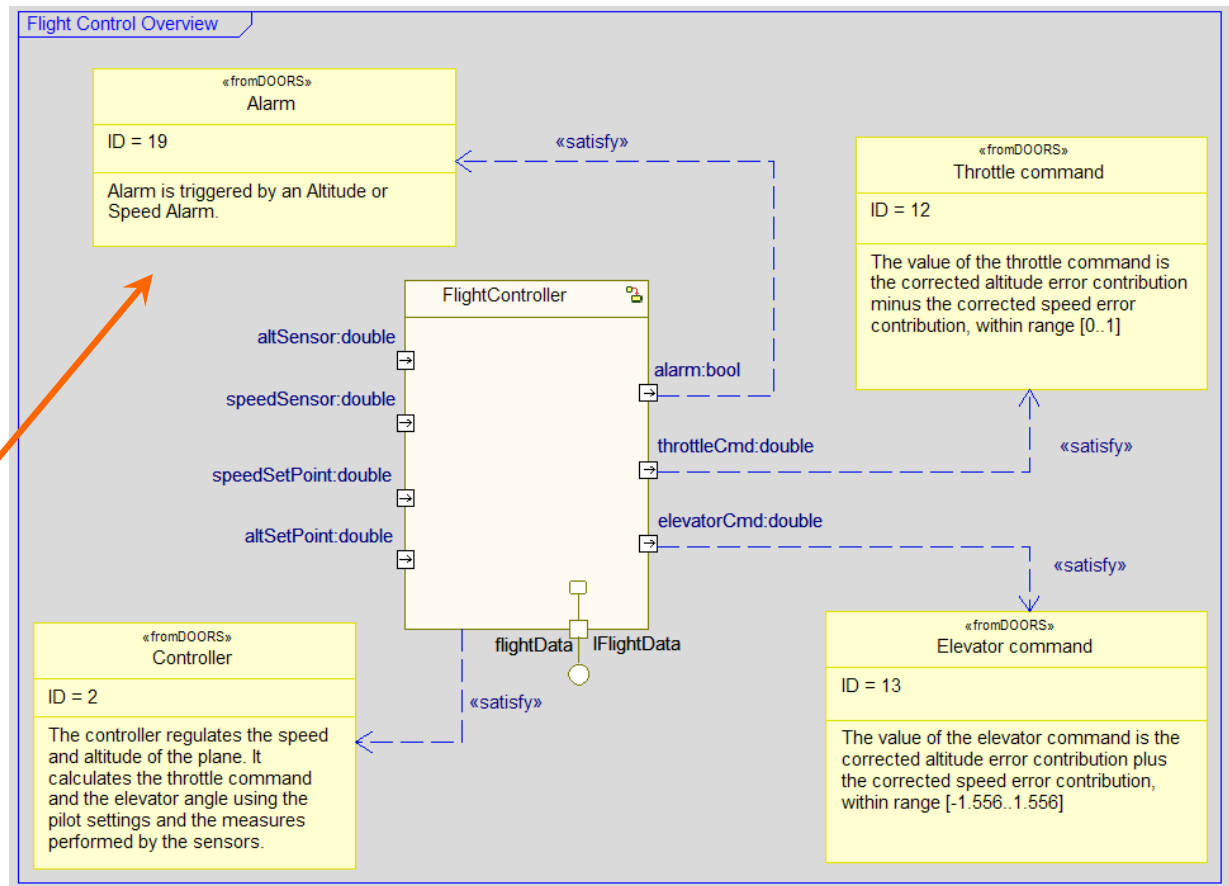
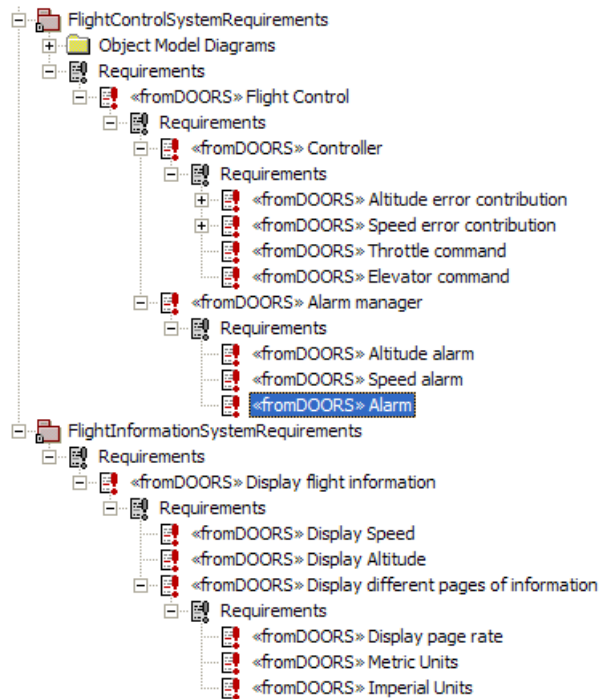
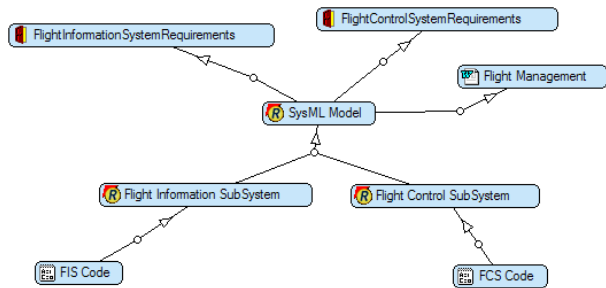


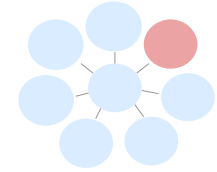
OMG  
SYSTEMS  
MODELING  
LANGUAGE





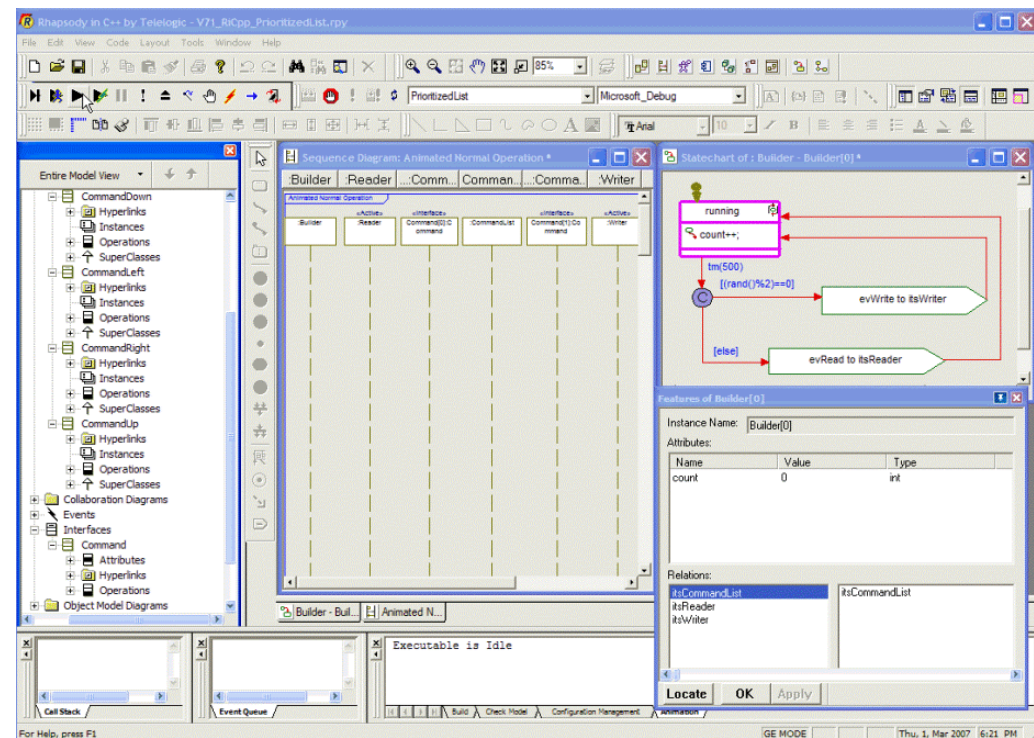
# REQUIREMENTS MODELLIEREN, ERFASSEN UND TRACEN MIT RATIONAL Rhapsody Gateway

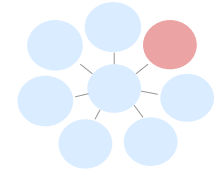




## AUSFÜHRBARE MODELLE

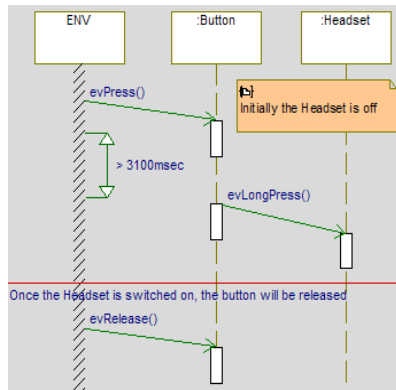
- Der beste Weg um Fehler zu vermeiden ist, diese gar nicht einzuführen.
- Modelle sind jederzeit ausführbar.
- Analyse-Modelle
  - Entsprechen die Sequence Diagramme den Zustandsdiagrammen?
  - Ist die Beschreibung der Abläufe widersprüchlich?
- Design-Modelle
  - Iterativ inkrementell entwickeln
  - Fehler so früh wie möglich erkennen und eliminieren
- Tests automatisieren
  - Die Standard UML Requirements-Artefakte verwenden
  - Analyse-Szenarios definieren die Test-Cases



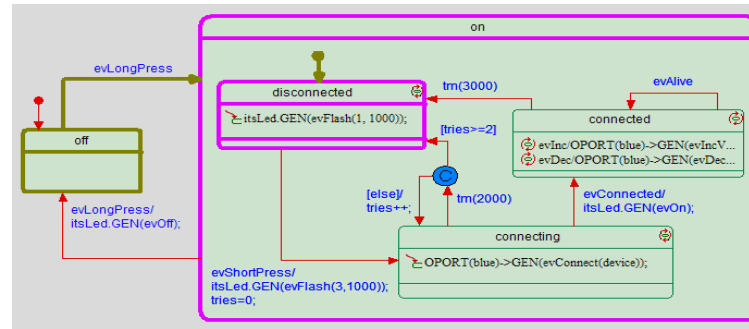


# REQUIREMENTS-BASIERTES TESTEN

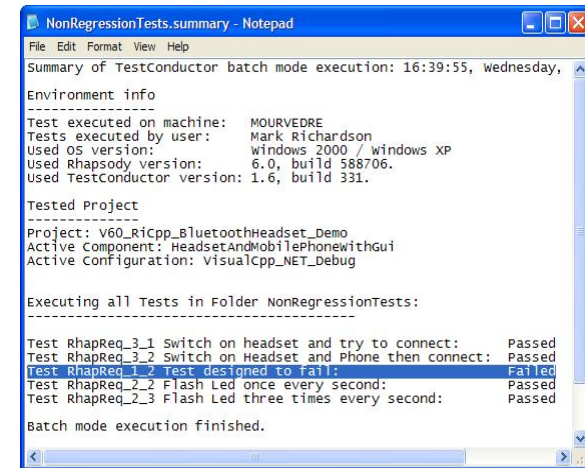
## Sequenz Diagramme



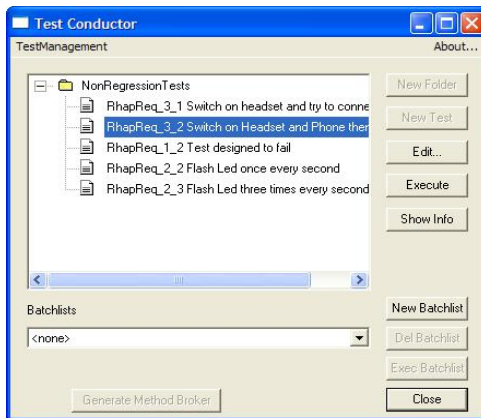
## Modell simulieren & überwachen



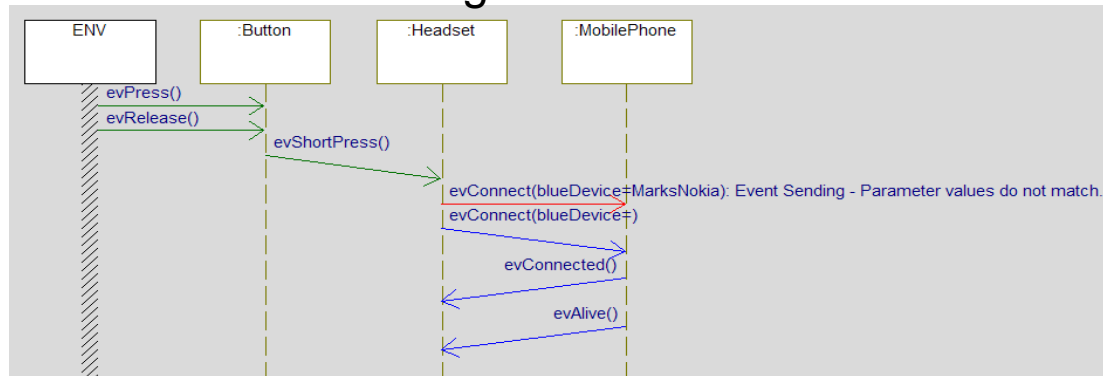
## Test Resultate

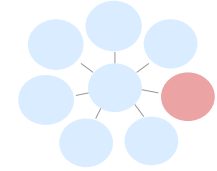


## Test Konfiguration



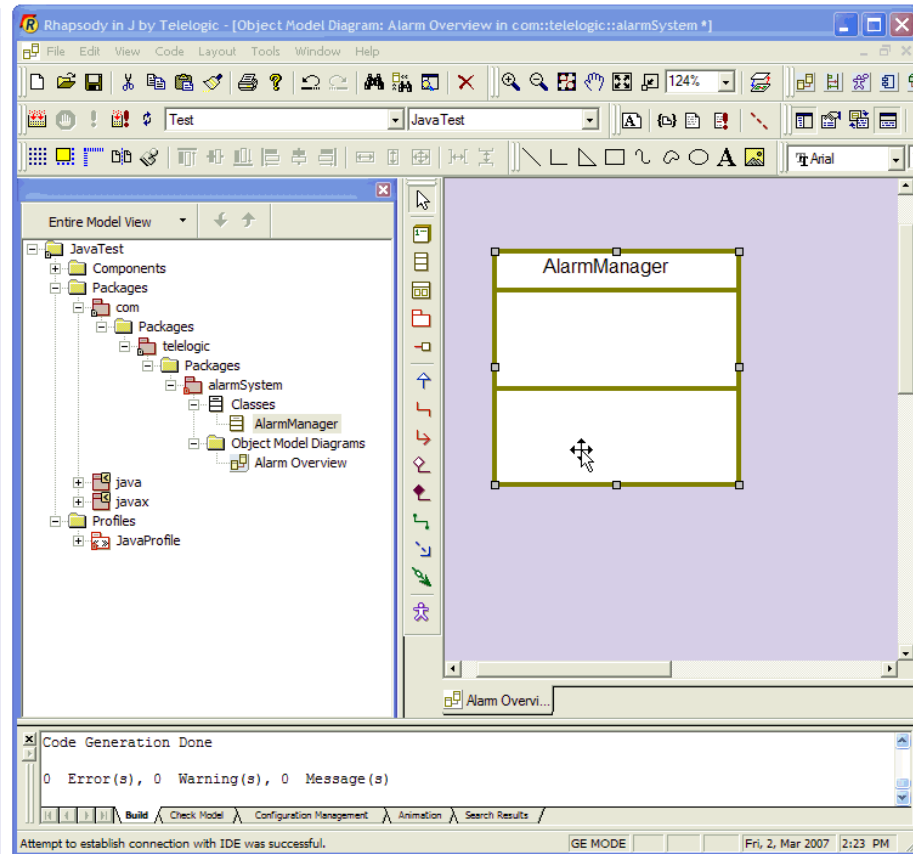
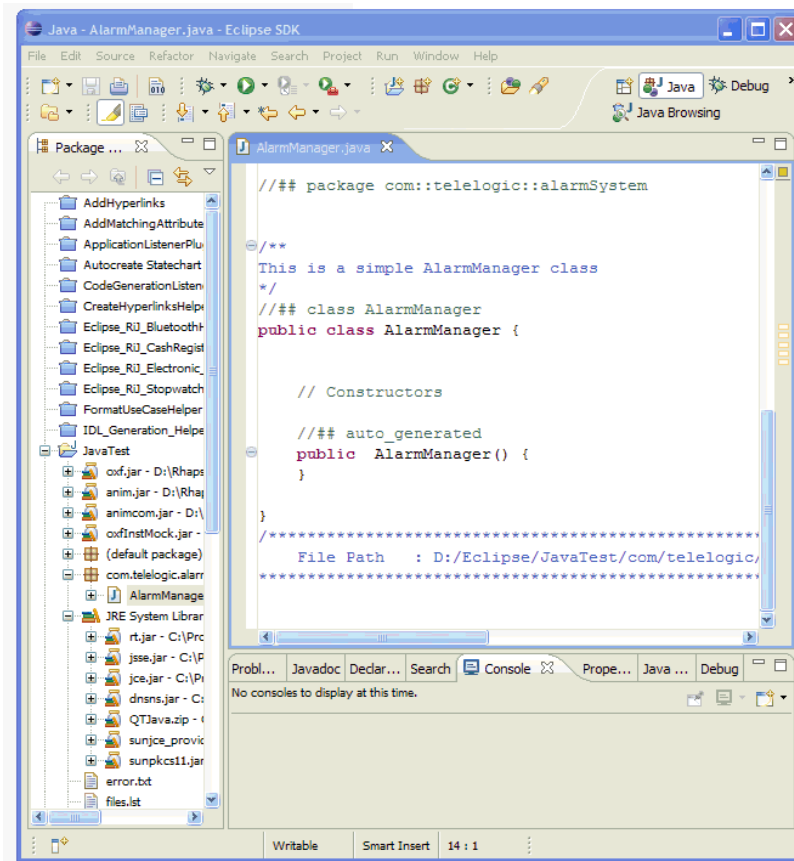
## Fehler finden & korrigieren

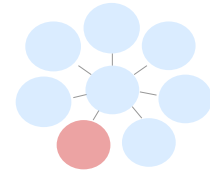




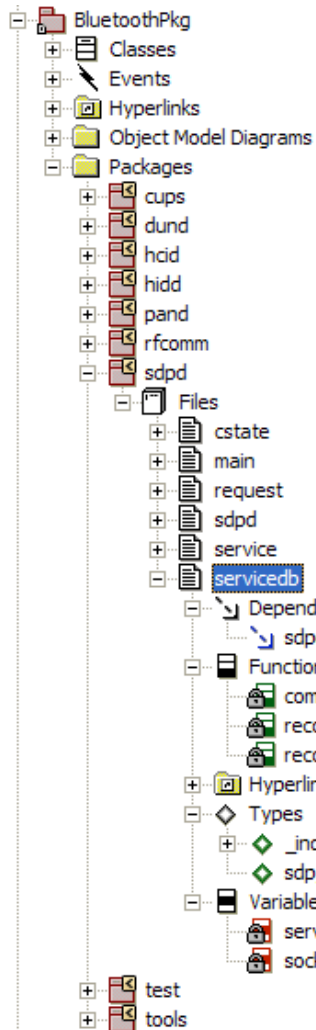
## DYNAMISCHE MODELL ZU CODE KONSISTENZ

- Änderungen in einer Sicht werden **automatisch nachgeführt**
- Modell und Code sind immer konsistent

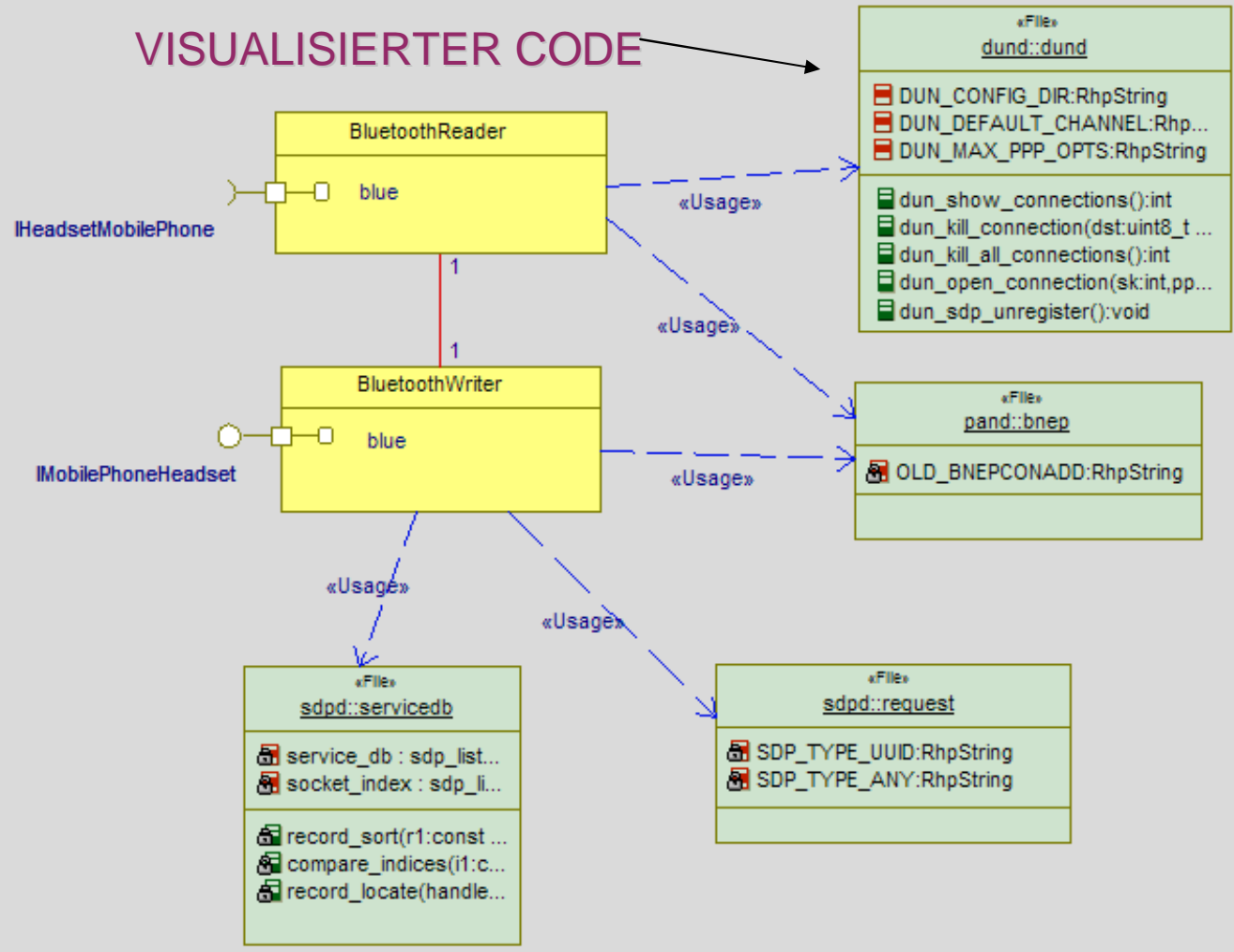




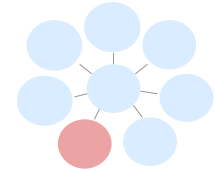
# EINBINDUNG VON LEGACY-ARTEFAKTEN



## VISUALISIERTER CODE

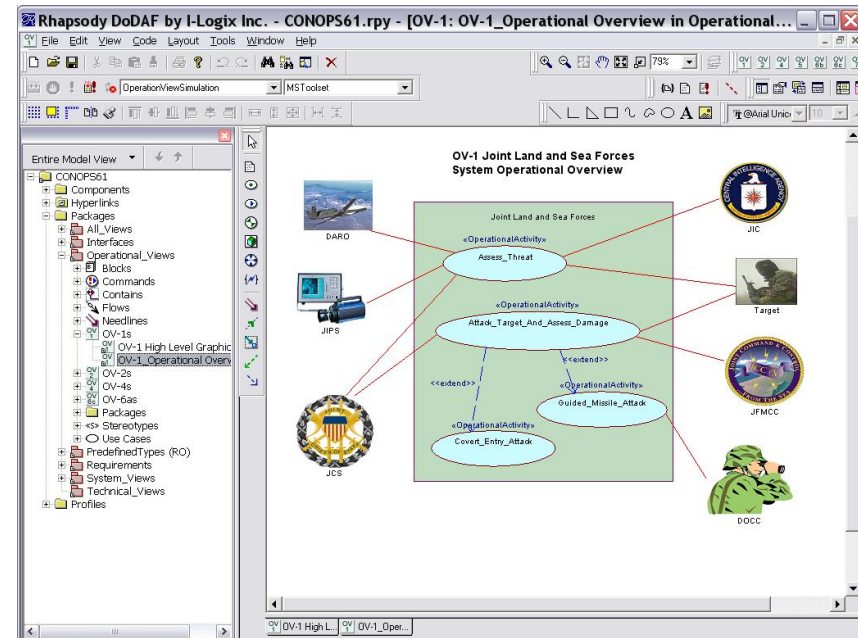
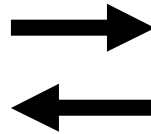






## WIEDERVERWENDUNG VON LEGACY CODE UND MODELLEN

Verschiedenste Modellformate können direkt in Rhapsody importiert werden: XMI-Modelle von Rational® Rose, No Magic, Inc.'s MagicDraw® oder Sparx Systems Enterprise Architect



## ZUSAMMENFASSUNG

## ZUSAMMENFASSUNG

```

<<DOCUMENTL.CXX
//parse C++ file and produce grammar graph
void exDocment::analyse(const ygTrngs file)
{
    exDocStream *tokenStream;
    tokenStream = new exDocStream(file.data());
    //for each token in the file
    while (tokenStream->next(token))
    {
        //if begins with "/*" then
        if (token.substring(0,2) == "/*")
        {
            //create a new exComment object
            currObj->addChild(new exComment(token));
        }
        //else if one of "if,else,while,for,do"
        else if (state.find(tokenStream->last()) != sta
        {
            //create a new exStatement object
    }
}
    
```

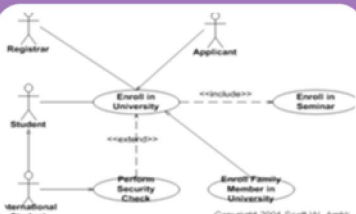
### Arbeiten auf Code Ebene genügt nicht für Effizienz

- Code notwendig aber ungenügende Sicht für Struktur und Verhalten
- Code nicht änderungsfreudig, Abhängigkeiten schlecht ersichtlich
- Fehlersuche auf Code Ebene sehr zeitaufwändig



### Ausführbare Modellierung als Basis für Agilität

- Graphische Methoden sehr wichtig für Abstraktion und Entkopplung
- Ausführbare Modelle essenziell für frühes Feedback
- Modell / Code Assoziation grundlegend für Nachhaltigkeit



### Anforderungsmanagement ausschlaggebend für Effektivität

- Anforderungsmanagement entscheidend für Priorisierung und Planung
- Agiles Anforderungsmanagement unterstützt Produktentwicklung optimal
- Tools unterstützen „Impact Analyse“ von Änderungen optimal

## EVOCEAN GMBH – FIRMENPORTRAIT

## STRATEGISCHE TOOL PARTNER



EVOCEAN GmbH ist Advanced IBM Business Partner sowie IBM Rational Training Partner und bietet Value Added Packages an basierend auf IBM Rational Produkten, spezifisch Rhapsody und DOORS.



EVOCEAN GmbH ist Partner von Visure Solutions und bietet Beratung bei der Einführung von Requirements Management & Engineering in den Bereichen IT und Produktentwicklung an (z.B. Medizinaltechnologie, Industrie etc. ).



EVOCEAN GmbH ist Partner von Willert Software Tools und bietet ergänzende Beratung bei der Einführung von Model Driven Development auf speicherarmen Embedded Systemen mit UML für C an.



EVOCEAN GmbH ist Partner von BigLever und bietet ergänzende Beratung bei der Einführung von Product Line Engineering Lösungen an, welche die IBM Produkte ergänzen.

FRAGEN?

## DANKE FÜR IHRE AUFMERKSAMKEIT!

### Kontakt

EVOCEAN GmbH  
Grundstrasse 8  
CH-6343 Rotkreuz

Tel. +41 41 790 78 88  
FAX +41 41 790 78 93

[www.evocean.com](http://www.evocean.com)

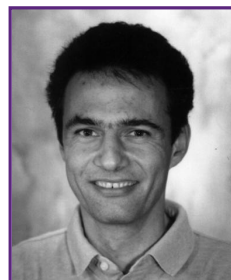
[info@evocean.ch](mailto:info@evocean.ch)



Philip Zollinger  
Geschäftsführer, Gründer  
Dipl. El. Ing. FH  
Mobile: +41 78 800 03 88  
Email: [philip.zollinger@evocean.ch](mailto:philip.zollinger@evocean.ch)



Blaise Rey-Mermet  
Principal Consultant  
Dipl. Natw. ETHZ  
Mobile: +41 79 276 64 19  
Email: [blaise.rey-mermet@evocean.ch](mailto:blaise.rey-mermet@evocean.ch)



Johannes Scheier  
Principal Consultant  
dipl. math.  
Mobile. +41 79 253 46 45  
Email: [johannes.scheier@evocean.ch](mailto:johannes.scheier@evocean.ch)