

# Einführung modellbasierter Technologien im industriellen SW-Engineering Umfeld

Hubert B. Keller

Software-Workshop "Effiziente Entwicklung zuverlässiger Software  
und methodisches Instrumentarium"  
Technologiepark Karlsruhe, 24.01.2008

# Übersicht

- Situation und Ziel
- Modellbasierte Technologien
- Technologieeinführung
- Prozessaspekte
- Projekt konkret
- Bewertung und Ausblick

# Situation und Ziel

## Existierendes System

- Altsystem 300 PA (eng gekoppeltes und verteiltes Gerätesystem)
- Alter > 10 Jahre
- Innovation < 10%, Wartung > 90%
- Monolithische Struktur, prozedural, HW-orientiert

## Ziel

- Flexibel auf Requirements reagieren (Skalierung, ...)
- Hohe Qualität (Architektur, Weiterentwicklung, Doku, ...)
- Effizienz (Entwicklung, Portabilität, time-to-market, ...)
- Zertifizierbarer Kernel mit in-house und externen Add-Ons
- Security Aspekte berücksichtigen

# Situation und Ziel - Fehler und Phasen

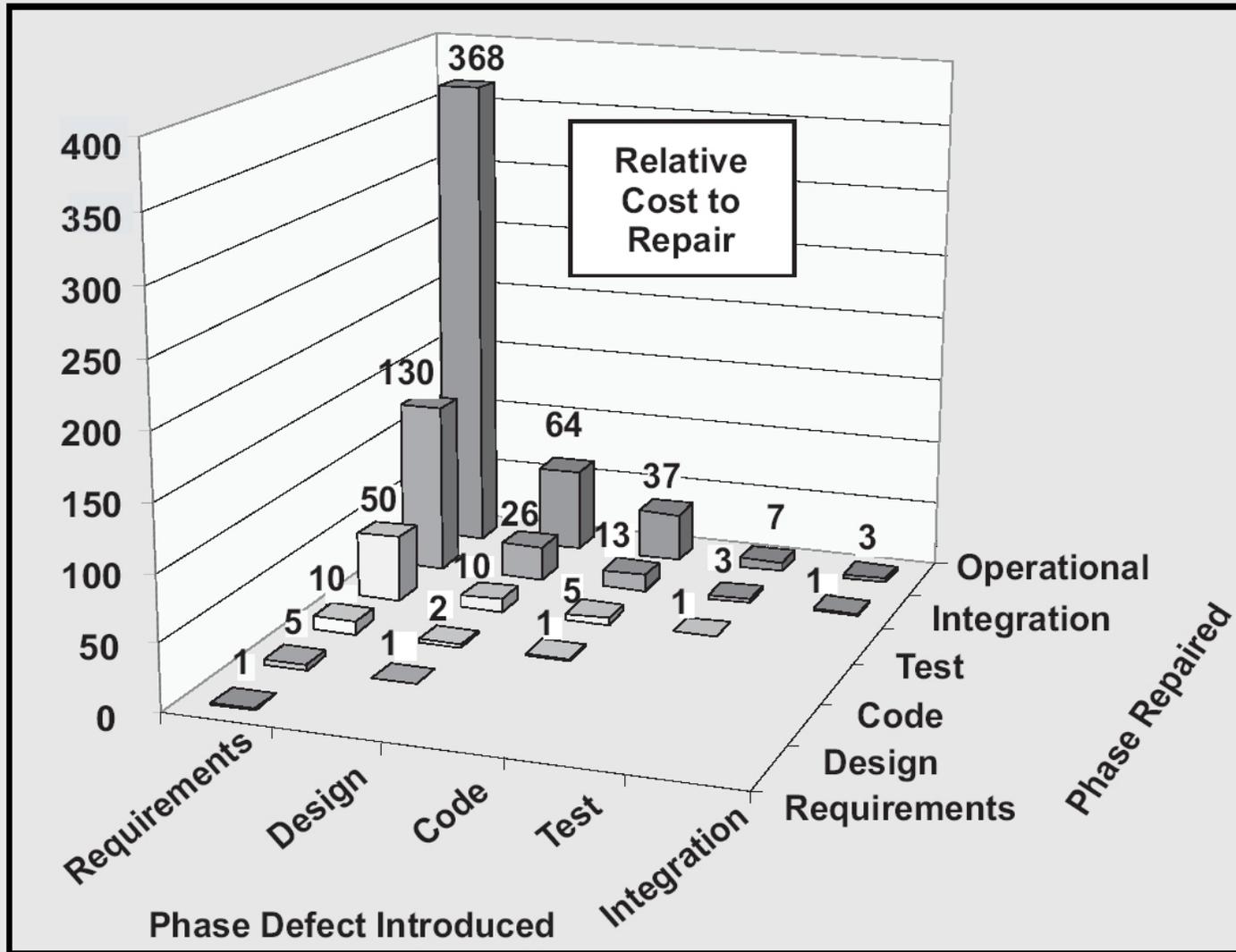
- HSE Out of Control – Analyse (2003)
  - 44% had inadequate specification as their primary cause
  - 15% design and implementation
  - 6% installation and commissioning
  - 15% operation and maintenance
  - 20% changes after commissioning
- 3/5 of control system failures are built-in before operation commences

Analysephase wird deutlich unterschätzt  
→ Mehr Fokus auf Analysemodell!

(Out of control - Why control systems go wrong and how to prevent failure. Health and Safety Executive (2003), No 238)

# Situation und Ziel - Kostenpropagierung

Figure 4: *Relative Cost of Software Fault Propagation*



Einführung modellbasierter Technologien im industriellen SWE Umfeld

# Situation und Ziel - Fehler und Kosten

- Tom DeMarco / Nosek und Palvia:
  - 75% der Gesamtkosten müssen z. Zt. für die Wartungsphase eingesetzt werden (Fehlerbehebung statt Neuentwicklung)
  - Bis zu 50% dieser Kosten resultieren aus Verständnisproblemen
- Ramberger (Distribution of Error Types):
  - 49% sind Dokumentationsfehler
  - 54% „value never returned“ und
  - 31% „value not documented“

# Modellbasierte Technologien - Warum Modelle?

## Mensch als Entwickler

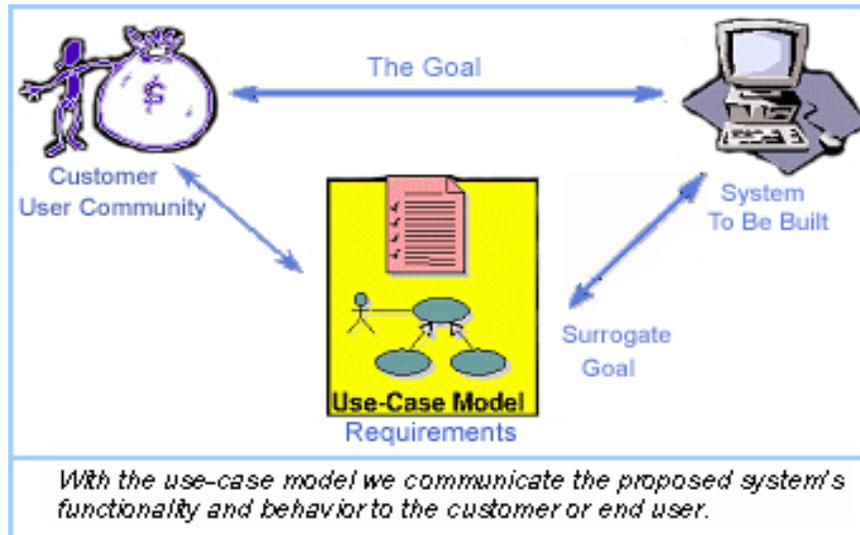
- Software Engineering erfordert
  - Kognitionsfähigkeiten [Denkfehler]
  - Explizierungsfähigkeit [Beschreibungsfehler]
  - Kommunikationsfähigkeit [Verständnisfehler]
  - Teamfähigkeit (Fachkompetenz ++)
- Modelle als Grundlage zur Entwicklung
  - Terminologie und Nomenklatur als Grundlage
  - Syntax, Semantik und Pragmatik von Modellbeschreibungen
  - Know How / Wissensmanagement

# Modellbasierte Technologien - Engineering

## Engineering von SW Systemen

- SW Engineering auf Modellbasis
  - UML als Modellierungssprache (Struktur und Verhalten)
  - Konzentration auf logische Struktur
  - Verständlichkeit
  - Transformationsgedanke
- SW ReEngineering auf Modellbasis (Migration)
  - Modell als Anforderungsdefinition
  - Modell als Maß für Wiederverwendbarkeit Altsystem
  - Modell als Zieldefinition (Machbarkeit)

# Modellbasierte Technologien - MDA und RUP



## Projekt und Ziel

← Organization along time →

## Weg und Prozess

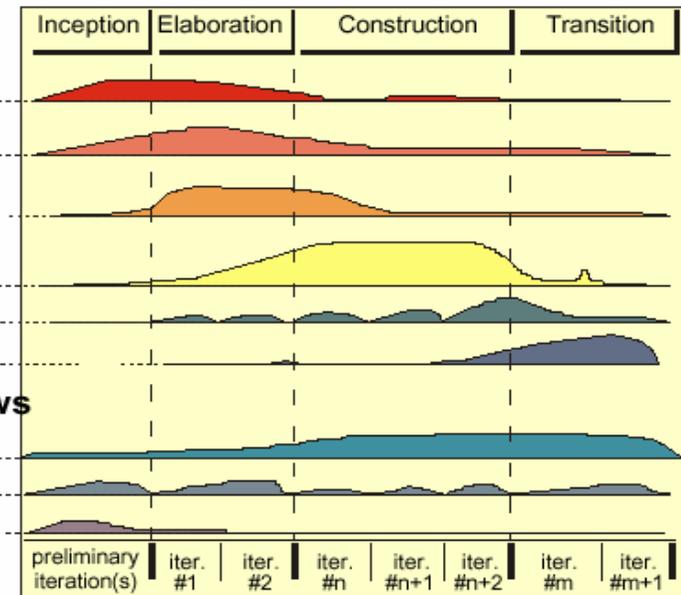
Organization along content

### Core Process Workflows

- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment

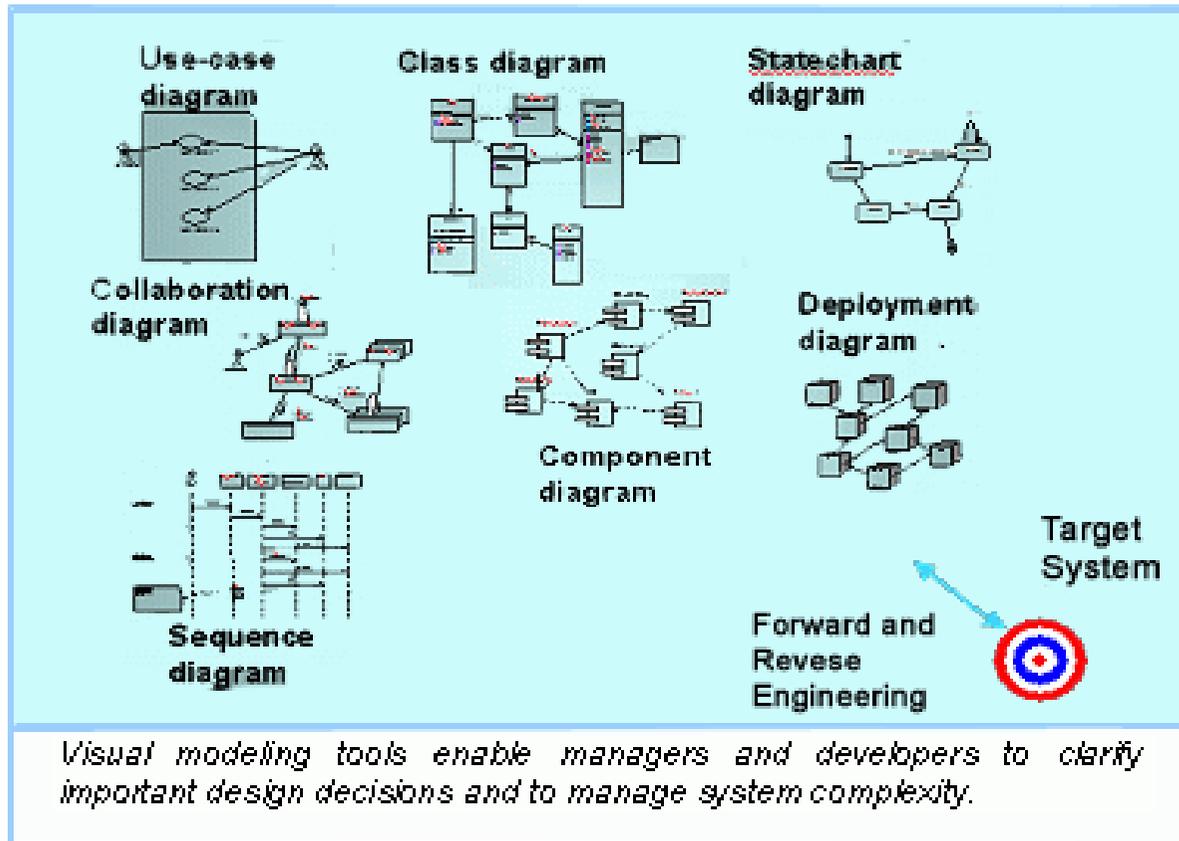
### Core Supporting Workflows

- Configuration & Change Mgmt.
- Project Management
- Environment



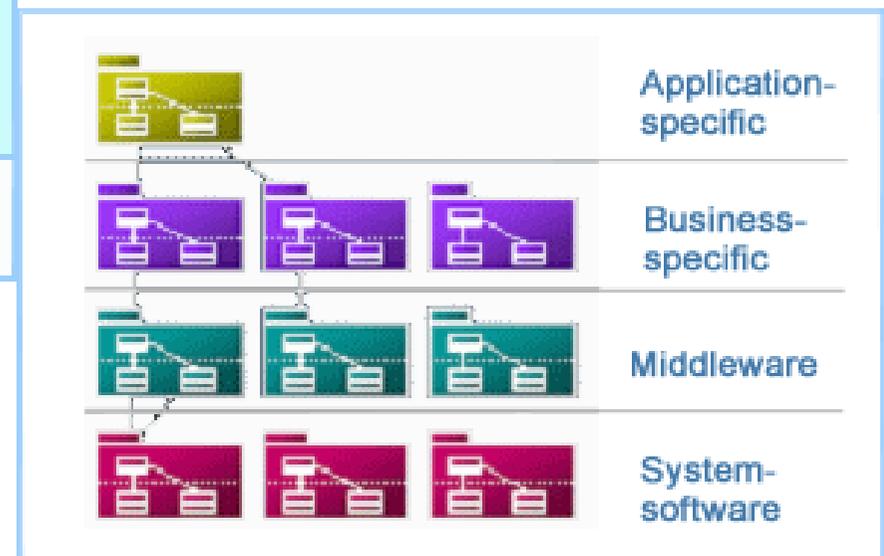
Iterations

# Modellbasierte Technologien - Iterativ, Inkrementell, OO



## Konzepte und UML Modellelemente

## Architekturziel

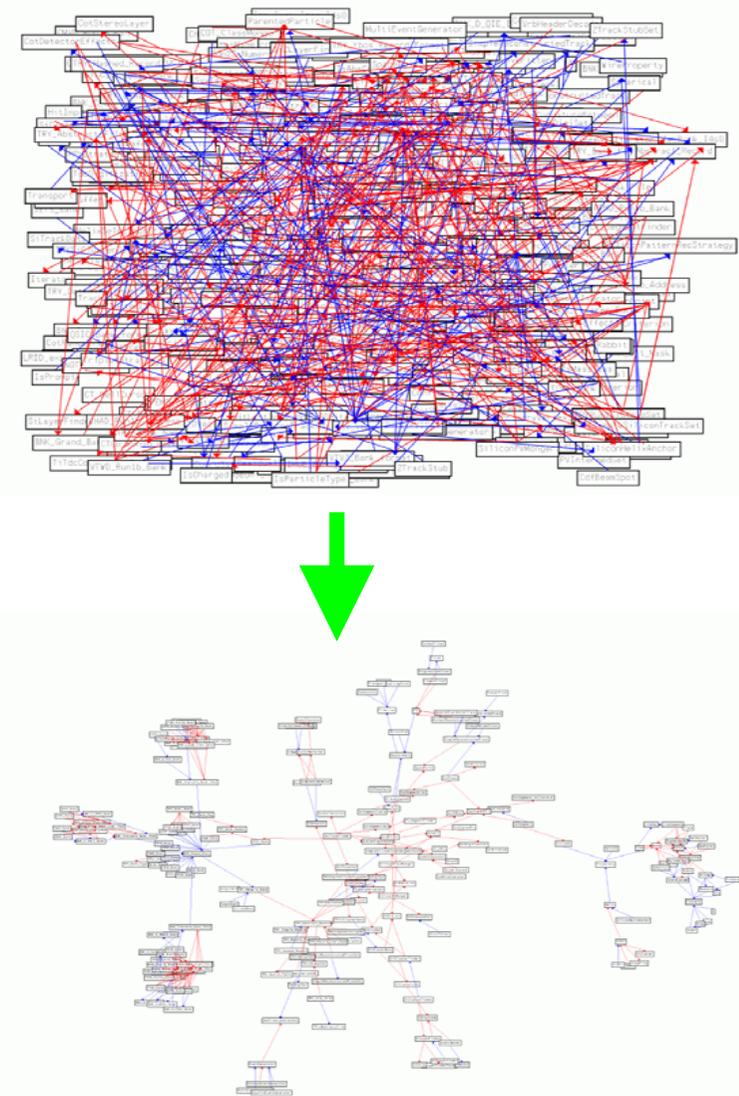


Component-based architecture with layers

# Modellbasierte Technologien - ReEngineering

## ReEngineering – Strategische Bedeutung

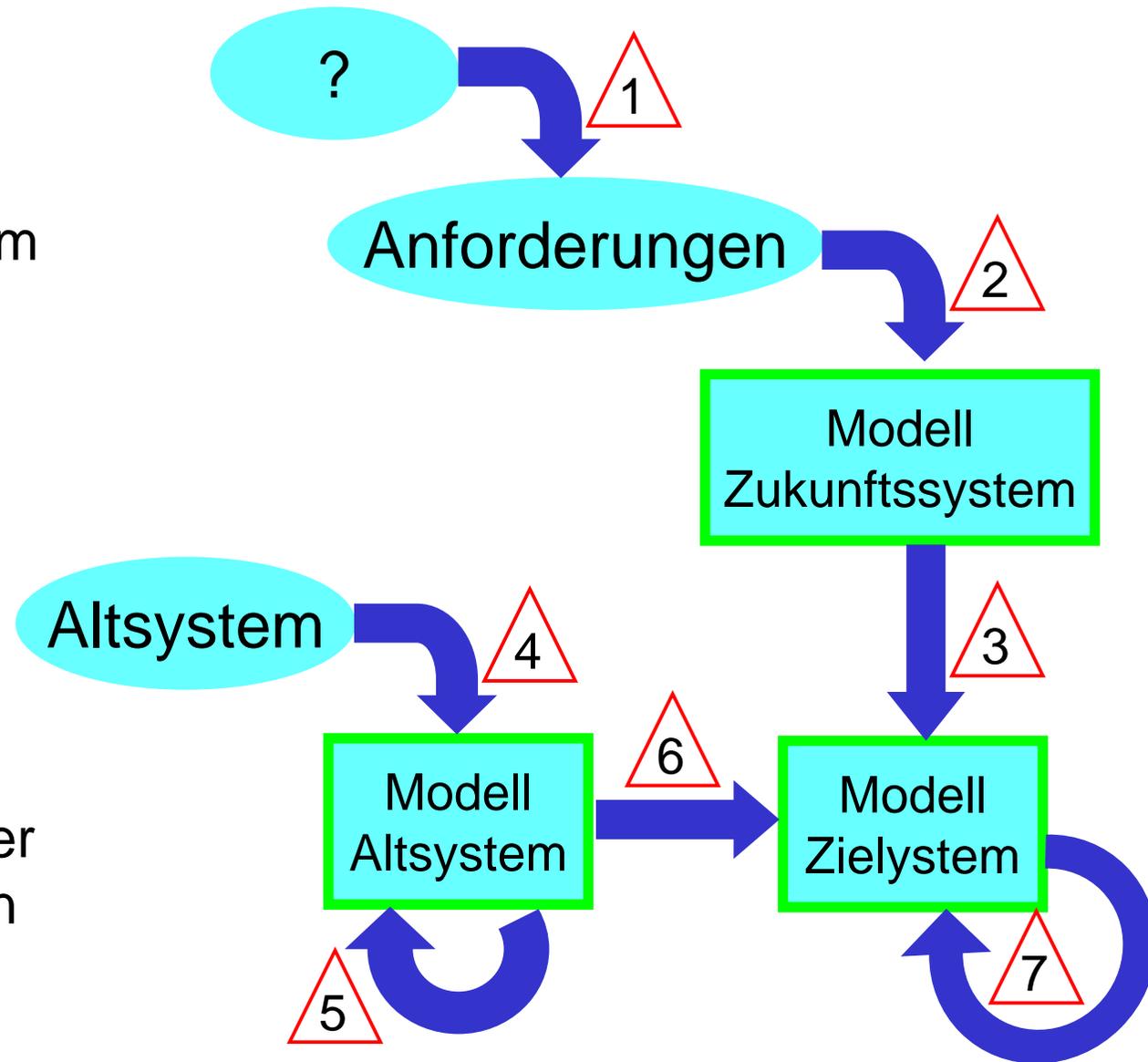
- Investition in ein Software-System, mit den Zielen:
  - flexiblere Architekturen
  - bessere Strukturierung
  - kostengünstige Funktionserweiterungen, Fehlerbehebungen und Technologiewechsel
  
- Investitionsschutz durch Wiederverwendung
  - Extraktion von Know-How und Code in Form von Komponenten aus Altsystemen



Einführung modellbasierter Technologien im industriellen SWE Umfeld

## Aktivitäten

1. Anforderungsanalyse
2. Modellierung Zukunftssystem
3. Entwicklung Kernsystem
4. Modellbildung, Reverse Engineering
5. Modellanalyse/ReUseability
6. Modell Migration
7. Iterative, inkrementelle Erweiterung des Zielsystemkerns mit iterativer und inkrementeller Migration des Altsystems



Einführung modellbasierter Technologien im industriellen SWE Umfeld

# Technologieeinführung - Vorgehen

- Vermittlung der Technologien vor Projektbeginn (Schulung)
- Anwendung und Vertiefung projektbegleitend (Grundverständnis + learning bei doing)
- Vermittlung des Prozessgedankens und der Vorgehensweise (RE → Kern-Requirements → Architektur → Vorgehen inkrementell, iterativ, OO)
- Vermittlung neuer / bewährter Methoden
- Konzept zur Integration von Altsystemen (vorhandene Werte) bzw. Restrukturierung

# Technologieeinführung - ... Vorgehen

- Freistellen von Personal und Motivation für Ziel (Zeitdruck, Lernbereitschaft, ...)
- Definition von Leitprojekten (Kernsysteme, Leitlinien, gesteuerte Evolution)
- Zusammenführung von
  - Geschäftsprozess-gesteuerte Requirements-Analyse
  - Architekturierung und Verhaltensmodellierung
  - mit funktionalen Komponenten des Altsystems
- Freischaufeln von Wartung – hin zu modellbasierter Entwicklung und automatischer Codegenerierung

# Technologieeinführung - weitere Aspekte

- Requirements Engineering für Kernanforderungen  
→ Zielsystemarchitektur (Kern-Anforderungen, Zukunftsträger)
- Risikoanalyse und –bewertung  
→ frühes Abklären, Prototyp, Zeit, Aufwand
- Funktionale und nichtfunktionale Anforderungen  
→ Zielsystemeinbettung / Umfeld / Zukunftsfähigkeit / Priorität HW
- Projektplanung / zeitliche Randbedingungen  
→ Egoismus (StoryCard), echte Personalverfügbarkeit, Reviewprozesse, ...

# Prozessaspekte – SW als Produkt

- Aktivitäten (Schritte) im Entwicklungsprozess und zugehörige Methoden
- Produkte als Ergebnis von Aktivitäten
- Verwaltung der Produkte
- Personen und Rollen
- SWE + KM + QS + PM
- Definition von Rückkopplungen im Prozess (einschließlich Kundeneinsatz) – Ziel CMMI

## Prozessaspekte - Phasen

- Analysephase – Requirements Engineering / Use Case Modellierung / Nebenläufigkeit und Verteilung
- Designphase – Komponenten, Interaktion, Verhalten, Architekturen, Client-Server, Kommunikation
- Automatische Codegenerierung – Transformation von Modellen mit Anreicherung und Prüfung / Templates / UML Profil
- Test- und Integrationsphase – Modellgetriebenes Testen in Planung
- Versions- und Variantenverwaltung - Standardtool

# Prozessaspekte - Rollenverständnis

- Rollen, z.B.
  - Architekt des Systems
  - Business Process Analyse/Requirements Management
  - Begriffsweltkonsistenz
  - MDA und UML Profilierung (Modell und Bedeutung)

# Prozessaspekte - Review

- Kernanforderungen darstellen, durchgehen und verifizieren
- Semantik Modellkomponenten und Konsistenz Modell
- Architekturierung-Ableitung
  - Architektur und Grundprinzipien des Systemkonzepts (Client-Server, ...)
- Systemkomponenten (wesentliche) darstellen
  - Teilsystemstrukturierung und Interaktion
  - Anbindung an Altsystem
- Prinzip: Anforderung, Konzeption, Modell, Alternativentscheidung

# Projekt konkret – Use Cases

Aus Vertraulichkeitsgründen sind die Folien aus dem konkreten Projekt im Netz nicht verfügbar.  
Für Rückfragen zu Details stehe ich gerne zur Verfügung.

Ich bitte um Verständnis.  
Danke.

# Bewertung und Ausblick

- Machbarkeitsstudie
  - Technologievermittlung
  - Projektbegleitende Technologievertiefung
  - Projektbegleitende Beratung
  - Projektbegleitende Analysen (Risiken, Schwachstellen)
- Definition von Leitprojekten
- Schaffung einer kritischen Masse (Projekte/Menschen) zur gesteuerten Evolution der Technologie bei Industriepartner
- Modelle geeignet für
  - Transparenz von Know How, In-Team Kommunikation
  - Kommunikation zu Vertrieb, Produktmanagement, GF, Kunden

## ... Bewertung

- Use Case für Requirements Engineering
- Erarbeitung einer konsistenten Begriffswelt
- Modelle für Architektur und funktionalem Verhalten
  
- Werkzeugeinsatz **OpenAmeos** (MDA basierte Codegenerierung, GUI Layout, Versionsverwaltung im Sinne Continuous Integration,...)
- MDA für Effizienz- und Qualitätssteigerung (Ausschluss zufälliger Fehler, Systemgenerierung über UML Profilierung und Templates)
  
- Prozessdefinition als Rückgrat der Software Entwicklung mit klarer Rollenfestlegung
  - Architekturverantwortlicher
  - MDA Verantwortlicher

# Ausblick

- Fokuswechsel von HW-orientierter zu SW-orientierter Strategie
- Mehrwert transparent und erkannt
- Leitprojekt als Keimzelle für umfassende Technologieeinführung
  
- Gruppe Leitprojekt zur Abteilung aufgewertet
- Produkt aus Leitprojekt zukünftig Kernprodukt
  
- Umfeldentwicklungen müssen sich an Leitprojektvorgehen orientieren
- CMMI Zertifizierung angestrebt / SW Prozessdefinition in Vorbereitung
  
- Modellbasierte Technologien bringen definitiv Mehrwert  
(30-35 PA Umfang mit ca. 20 PA trotz MDE + RE \ Vorwissen)