

Grosse Systeme im Griff

Ein Konzept für V-Modell konformes Anforderungsmanagement und Systemarchitekturmodellierung mit
UML und RE/RM für komplexe Systeme

Teil:
Methodisches Vorgehen

This document contains proprietary information belonging to EADS Intellectual Property GmbH. It may not be distributed or copied and its contents may not be used or disclosed in any third party without the written permission of EADS. Violation will lead to immediate prosecution. All rights in respect of patents or other intellectual property rights reserved.

Vorstellung

EADS European Aeronautic Defence and Space Company
Bereich: Defence & Civil Systems
Firma: Lenkflugkörpersysteme GmbH

Aufgabe: Entwicklungsleiter für eine Waffenführungssystem
Name: D. Wagner

CoCOO – Competence Centre ObjectOrientation
(www.cocoo.de)

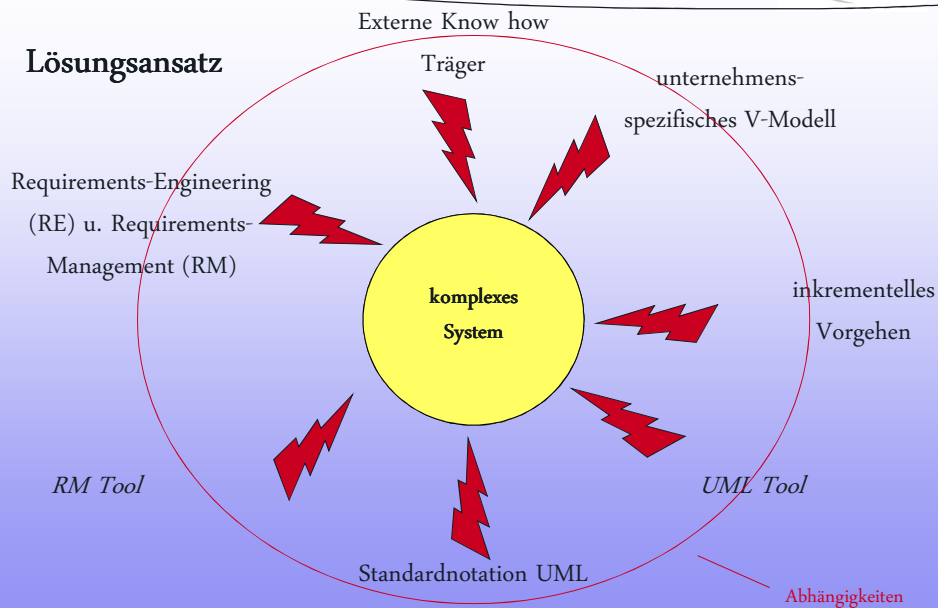
Aufgabe: Methodenberater für OO/UML und
V-Modell'97

Name: M. Reinhold (reinhold@cocoo.de)



Rahmenbedingungen

- Entwicklung eines komplexen Systems
 - System besteht aus n Anwendungssystemen
 - Systementwicklung mehr als „nur“ Anwendungsentwicklung
 - Entwicklung/Integration unterschiedlicher Anwendungssysteme, HW, Betriebssysteme, Netzwerktechnologien
- Berücksichtigung beim Vorgehen, Teile des System als Unterauftrag zu vergeben
- Berücksichtigung des Unternehmensstandards zur Erstellung von Systemen (V-Modell)
- Realtime System



Lösungsbausteine

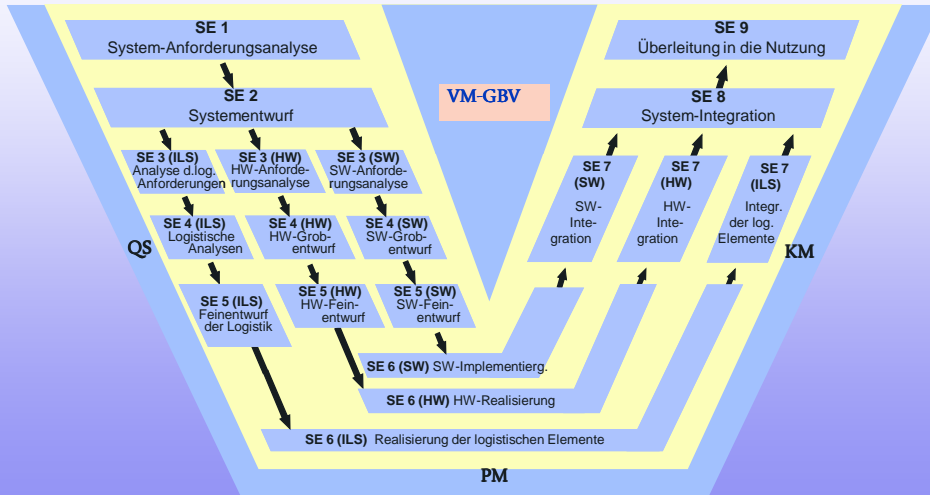
- Unternehmensspezifisches V-Modell (erweitertes V-Modell'97)
- Inkrementelles Vorgehen auf Systemebene
- Standardnotation UML

Verbindung mit der Toolumgebung (siehe Teil 2)

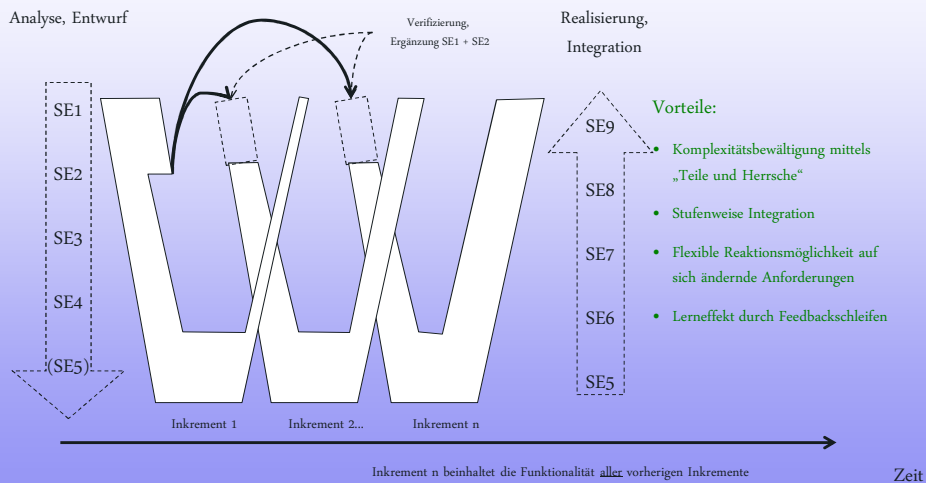
- *Einsatz der Rational Suite*
- *Erweiterungen der Rational Suite durch Skripts*

Lebenszyklus komplexer Systeme

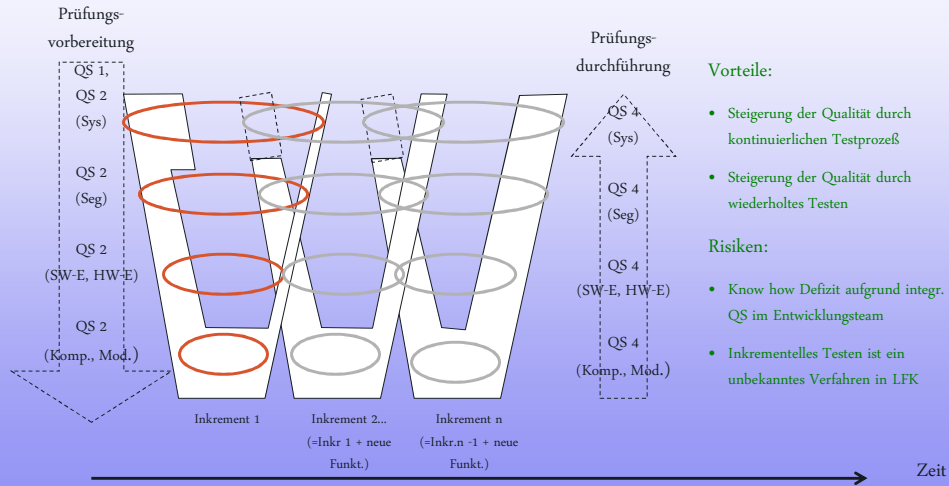
Unternehmensspezifisches V-Modell



Inkrementelle Vorgehensweise aus SE Sicht

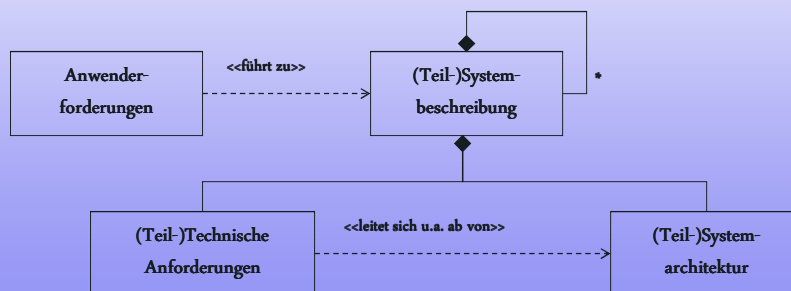


Inkrementelle Vorgehensweise aus QS Sicht

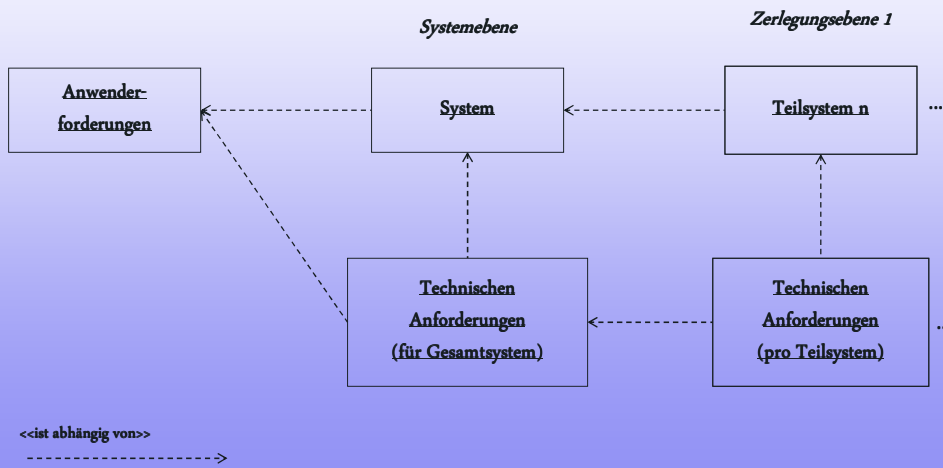


UML Modellierung - Grundidee (1)

- Kundenanforderungen bestimmen das System
- Firmenanforderungen bestimmen die Realisierung
- Hierarchischer Aufbau
 - Jedes komplexe System zerfällt *hierarchisch* in Teilsysteme

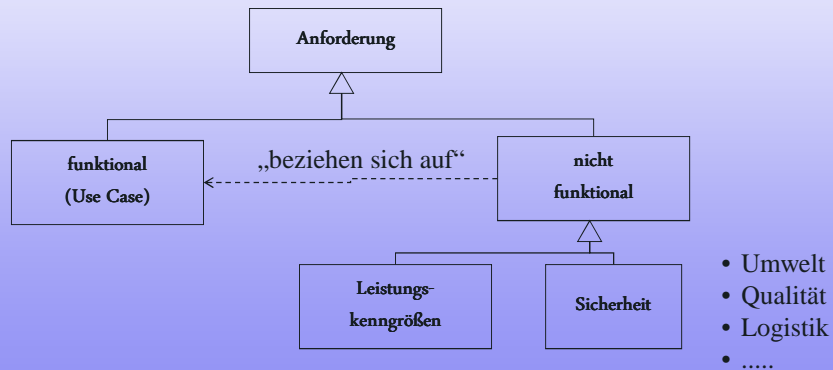


UML Modellierung – Grundidee (2)



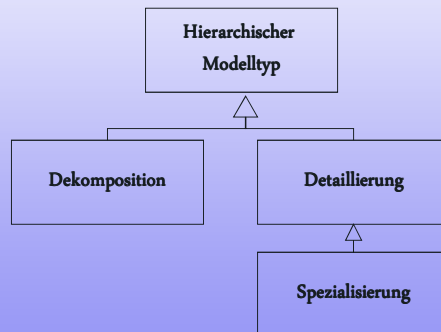
Anforderungsarten

- Fragestellung
 - Welche Anforderungsarten gibt es?



UML Modellierung - Hierarchisierung (1)

- Fragestellung
 - Welche „hierarchischen“ Modelltypen kennt die UML



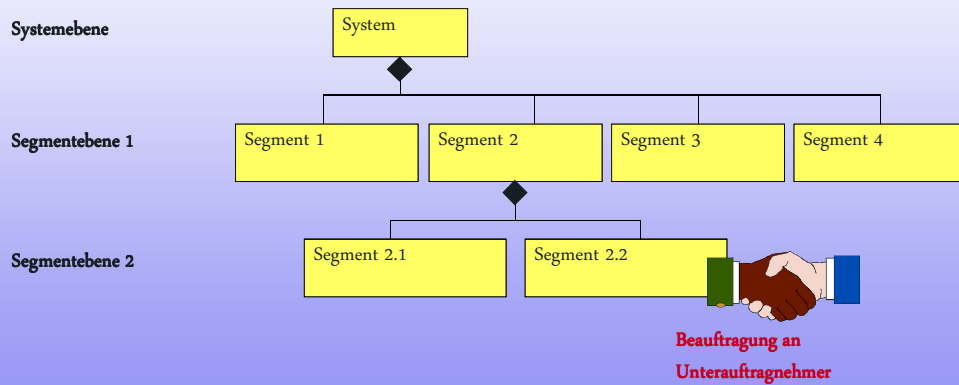
Bei einer hierarchischen Zerlegung des Systems wird pro Ebene noch ein **integratives** Element benötigt, welches den dynamischen Zusammenhang erläutert

Modellierungslösung – Hierarchisierung (2)

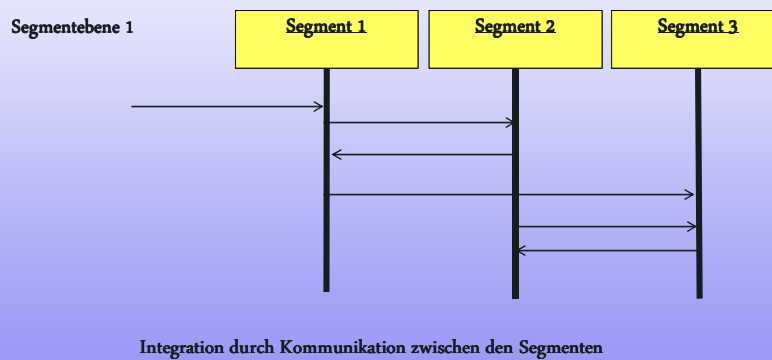
In konkreten UML-Modellelementen bedeutet dies:

- **Architektur**
 - Klassendiagramm, Paketdiagramm (Dekomposition)
 - Sequenzdiagramme zur Schnittstellenbeschreibung - *Integrativ*
- **Technische Anforderungen (pro Dekompositionsebene in Architektur)**
 - (hierarchisierbare) Use Cases (Detaillierung)
 - Zustandsdiagramm (Spezialisierung)
 - Sequenzdiagramme und Aktivitätsdiagramme - *Integrativ*

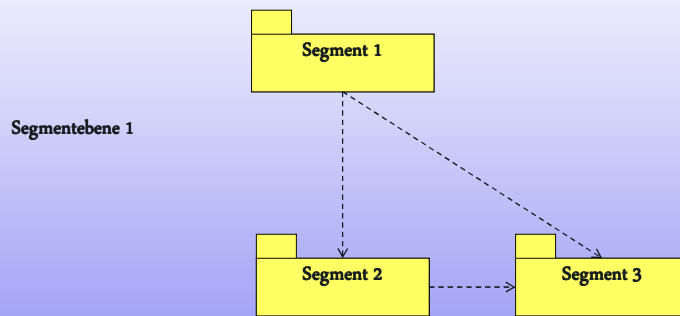
Hierarchische Systemarchitektur durch Dekomposition



Dynamisches Zusammenspiel der Architekturbausteine

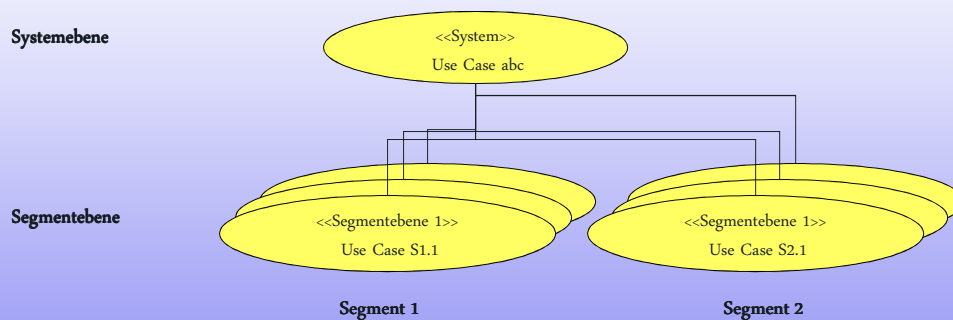


Abhängigkeiten zwischen den Architekturbausteinen



Abhängigkeit durch Kommunikationskanäle zwischen den Segmenten

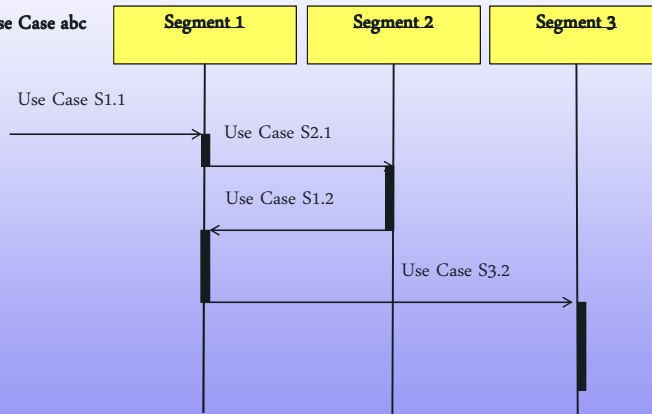
Hierarchische Anforderungsbeschreibung durch Detaillierung



Rekursives Zerfallen der Use Cases auf jeder Segmentebene

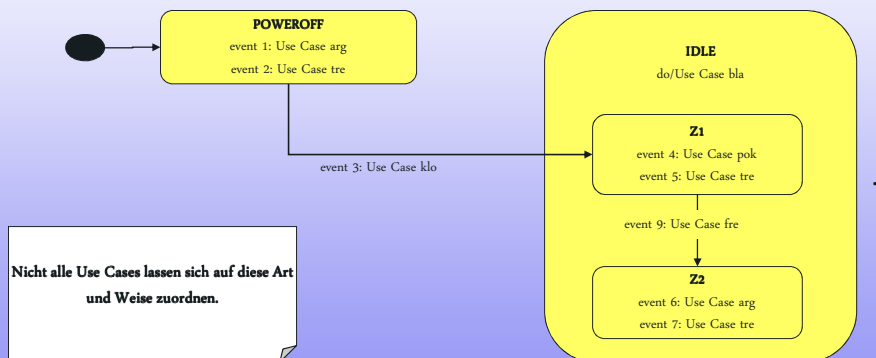
Zusammenspiel der Use Cases

z.B. System Use Case abc



Realisierung der System Use Cases durch Segment Use Cases

Zusammenhang der Use Cases zu Systemzuständen



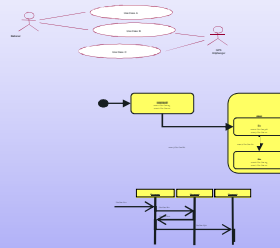
Nicht alle Use Cases lassen sich auf diese Art und Weise zuordnen.

Zuordnung der Use Cases zu den Systemzuständen

Zusammenfassung Modellelemente

- Technische Anforderungen

- Use Cases
- Zustandsdiagramm
- Sequenzdiagramme und Aktivitätsdiagramme



- Architektur

- Klassendiagramm
- Sequenzdiagramme

