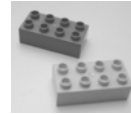
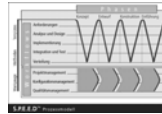


# Modellbasiertes Requirements Engineering

## 2. Requirements Engineering Tagung 2003

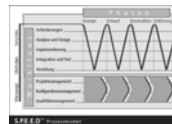
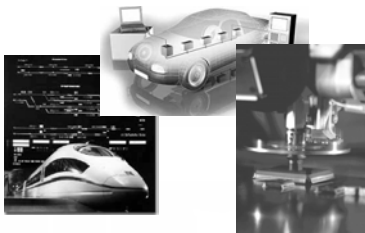


Dr. Rolf Ebert  
 rolf.ebert@bms.de, www.bms.de  
 Berner & Mattner Systemtechnik GmbH

## Berner & Mattner Systemtechnik GmbH

### Leistungsschwerpunkte

- >20 Jahre kundenspezifische Entwicklung von technischen Softwaresystemen (Real Time, Embedded)
- >10 Jahre Leading Edge Technologien für die modellbasierte Entwicklung von Software und Systemen
- Beratung zu Prozessen und Methoden



**SPEED**  
 Berner & Mattner

*Rhapsody*  
 Statemate MAGNUM



## Übersicht

- **Problemstellung**
- **Anforderungs-Modelle**
- **Praktische Umsetzung**
- **Erfahrungen**

# Problemstellung

## Problemstellung

- Anforderungen sind noch unklar
- Anforderungen sind oft schwer verständlich
- Anforderungen sind nicht eindeutig
- Anforderungen sind nicht überprüfbar
- Systeme sind sehr komplex
- Anforderungen sind nie vollständig

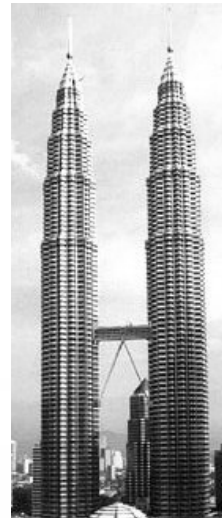


## Anforderungs-Modelle

## Warum sind Modelle sinnvoll?

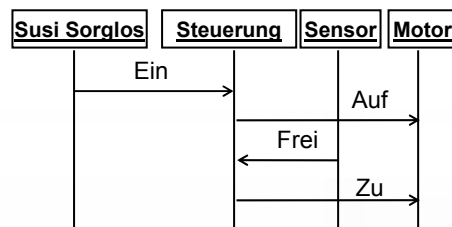
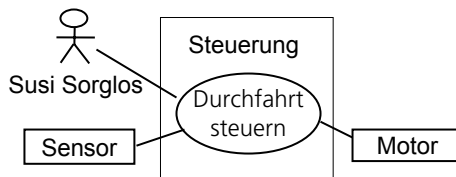
**„Ein Bild sagt mehr als 1000 Worte“**

**„Ein Modell sagt mehr als 1000 Bilder“**



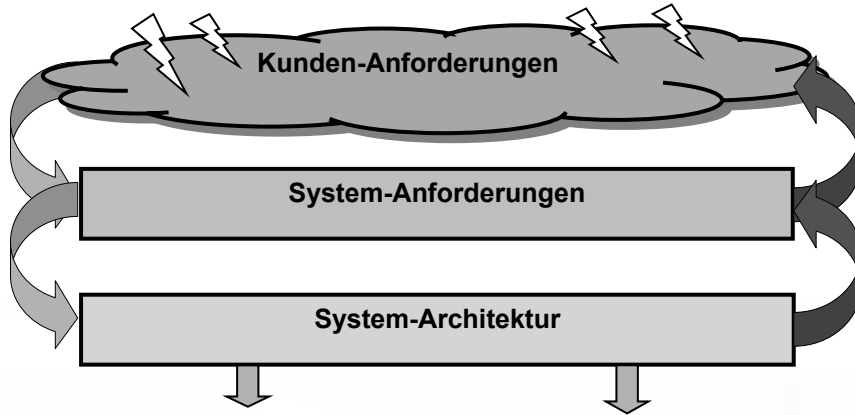
## Anforderungs-Modelle

- Use Cases sind der zentrale “Auhänger”
- Diagramme + Text
- Detaillierung der Use Cases
  - Sequenz-Diagramme
  - Zustands-Diagramme
  - Aktivitäts-Diagramme



## Anforderungs-Modelle

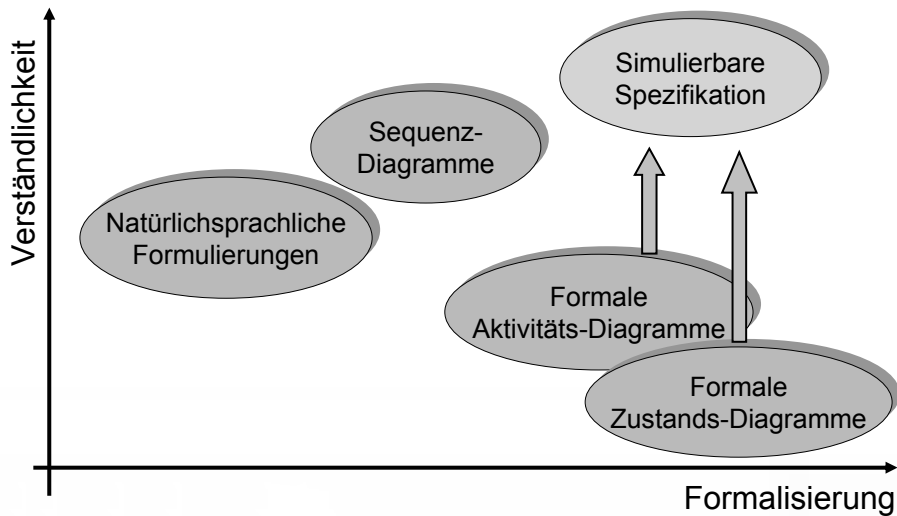
### Requirements Engineering und Architektur



## Simulierbare Anforderungs-Spezifikation

- **Simulation erfordert eine formale Beschreibung des Systemverhaltens**
  - mit Zustands-Diagrammen
  - mit Aktivitäts-Diagrammen
  - mit Block-Diagrammen
- **die Modellbildung führt zu einer hochgradigen Durchdringung des Systems**
- **primär für kritische Use Cases sinnvoll**

## Verständlichkeit und Formalisierung



## Verifikations- und Validierungs-Methoden

- **Verifikation von Anforderungen**
  - statische Prüfungen mit Inspektionen und Reviews
  - dynamische Tests mit simulierbaren Modellen
  - Model-Checking
- **Validierung von Anforderungen**
  - nur simulierbare/ausführbare Anforderungs-Spezifikationen können validiert werden



You can't test what you can't execute

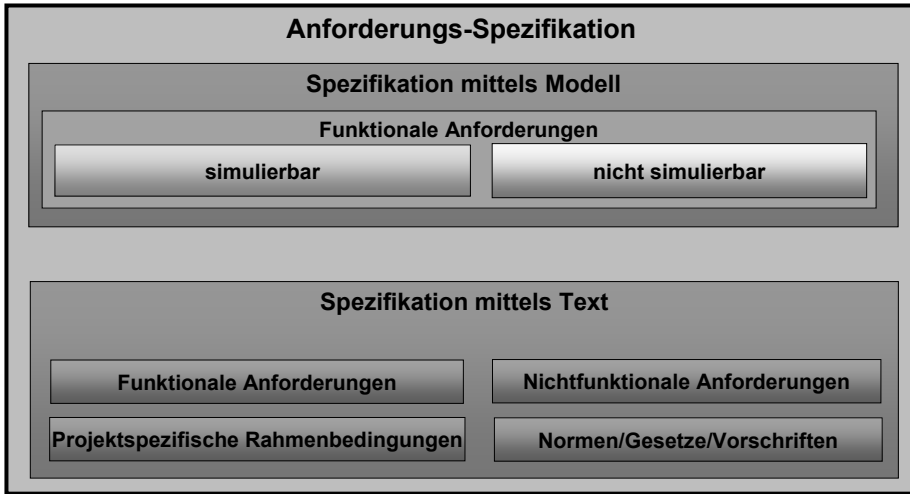
(Bruce P. Douglass)

## Anforderungs-Modelle

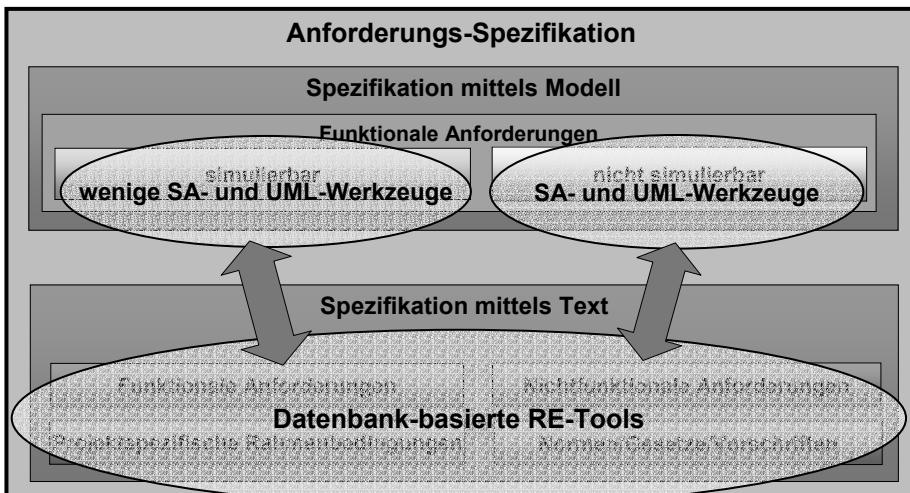
- verschiedene Darstellungen
  - Text
  - Modell
  - ausführbares Modell
- nicht Implementierung vorwegnehmen
- Verständnis und Überblick schaffen
- zu Fragen anregen
- Simulation „beweist“
  - Freiheit von Widersprüchen
  - Machbarkeit

## Praktische Umsetzung

## Anforderungen strukturieren

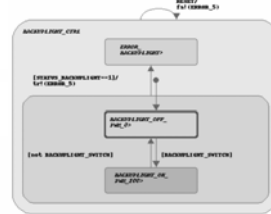
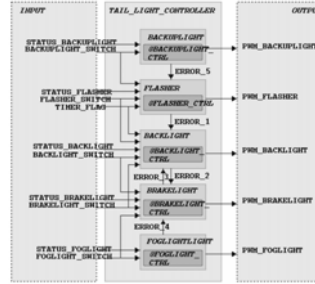


## Werkzeuge gezielt einsetzen



# Praktische Umsetzung

ID	Anforderung	Status	Priorität
1	<b>1 Allgemeine Beschreibung</b>	Offen	Normal
2	<b>2 Globale Einschränkungen</b>	Offen	Normal
3	2.1 Zweck des Systems	Offen	Normal
4	2.2 Stakeholder	Offen	Normal
5	2.3 Grenzungen des Systems	Offen	Normal
6	<b>3 Funktionale Anforderungen</b>	Offen	Normal
7	<b>4 Nichtfunktionale Anforderungen</b>	Offen	Normal
8	4.1 Performance	Offen	Normal
9	4.2 Interface	Offen	Normal
10	4.3 Verfügbarkeit	Offen	Normal
11	4.4 Wartung	Offen	Normal
12	4.5 Umgebungsbedingungen	Offen	Normal



# Erfahrungen

## Erfahrungen

- Modellerstellen erscheint als Mehraufwand
- richtiges Modell und Simulation benötigen Zeit
- Implementierungs-Schwierigkeiten, um Simulation zu erreichen
- welche Modell-Elemente werden validiert, welche werden benötigt zur Ausführbarkeit?



## Erfahrungen

- viele offene Punkte durch Modelle aufgedeckt
- viele Widersprüche durch Modelle aufgedeckt
- Modellierungsaufwand reduziert Nacharbeiten
- formale Modelle ermöglichen formale Nachweise



# Fazit

## Modell-basiertes Requirements Engineering

### Best Practices

- Iterativ vorgehen
- Werkzeuge mit offenen Schnittstellen
- Texte + Modelle + Ausführbare Modelle einsetzen
- Den Faktor Mensch beachten – er entscheidet alles!