



---

# AP1: Lernmaterial

SPEDiT Abschlussveranstaltung, 12.04.2019

Dr. Andreas Froese  
(Universität Duisburg-Essen)

---



(2009-2012)



(2012-2015)

- **Vielzahl** von Konzepten, Techniken und Methoden zur modellbasierten Entwicklung eingebetteter Systeme

**Aufbereitung** der Ergebnisse als **geeignetes Lernmaterial**  
(für die Schulungsplattform)



- **Definition potentieller Schulungsteilnehmer** die SPEDiT adressieren soll
  - Typische Lernende und ihr Fokus
  - Lernmaterial schneiden
  - **Strukturierung nach Personas**
  
- **Persona**
  - Aufgabengebiet, Vorkenntnisse, Wünsche bzgl. Schulungsangebot
  - Geben Leitlinien für Aufbau und Umfang der Lerneinheiten vor
  - 14 typische Lernende durch SPEDiT-Industriepartner beschrieben
  - **Konsolidierung in 5 Personas**
  
- **Personas** im Entwicklungsprozess für „Eingebettete Systeme“
  - System Architect (Sebastian Nitsch)
  - System Requirements Engineer (Kaiwen Li)
  - Test Engineer (Cornelius Cover)
  - Software Architect (Christian Strätz)
  - Software Developer (Hedwig Hacker)

**Sebastian Nitsch**, 35 Jahre alt, ledig (System Architect)

## Beruflicher Hintergrund

Herr Nitsch hat bereits als **Maschinenbau-Student** bei seinem jetzigen Arbeitgeber begonnen. Er wurde aus einem großen Kreis von Kandidaten ausgewählt und durfte die Entwicklung und den Bau eines Rennwagens begleiten und schließlich selbst Rennen fahren, die von seinem Arbeitgeber gesponsert wurden. Er gilt in seinem Unternehmen als "High Potential" und wurde intensiv und vielfältig ausgebildet.

## Skills/Vorwissen

- **Vorwissen Systems Engineering und MBSE**
- Enterprise Architect / PTC Integrity Lifecycle Manager
- Deutsch / Englisch

## Interesse zur Weiterbildung/Psychographie

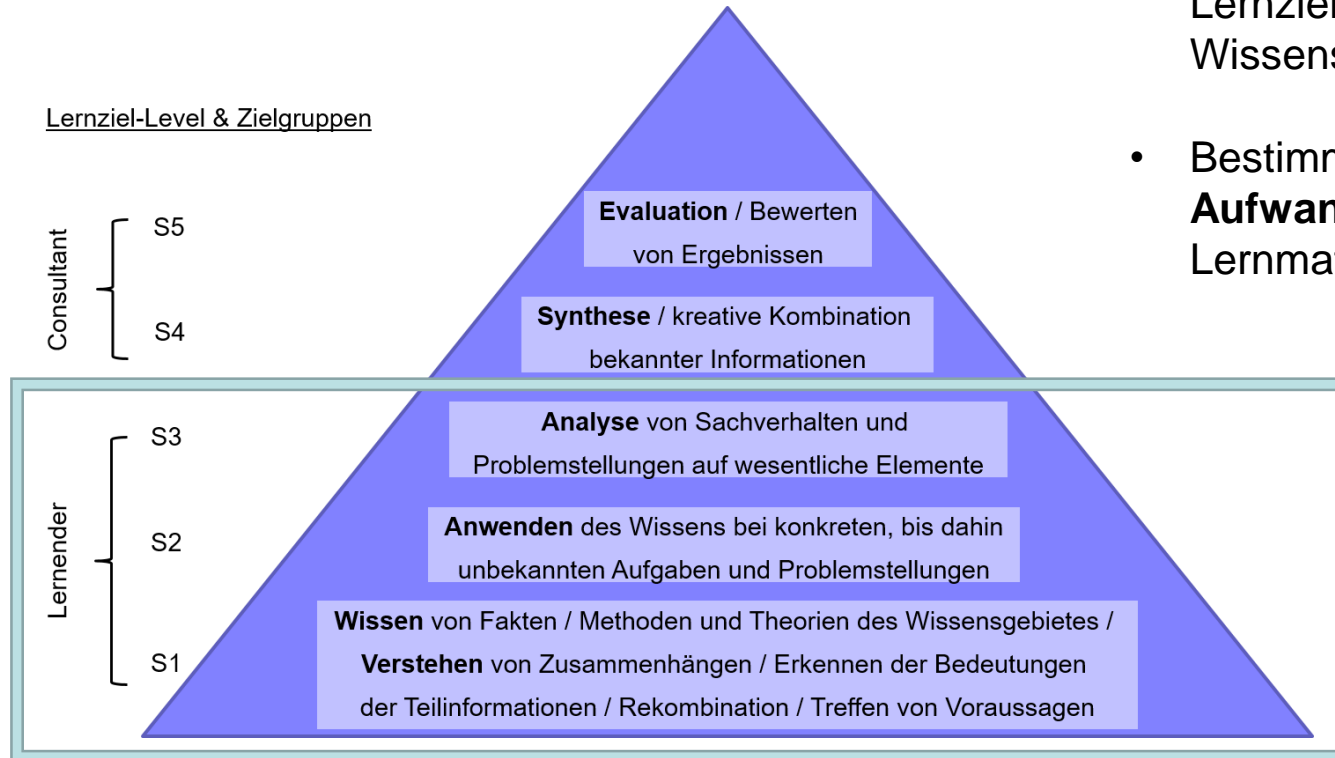
Herr Nitsch ist immer bestrebt, seine **Effektivität** und **Effizienz** zu steigern. Im Bereich Systems Engineering sucht er nach **methodischen Weiterentwicklungen**. Dabei ist er sehr anspruchsvoll: **er hinterfragt sehr hartnäckig** Sinn, Nutzen und Anwendbarkeit von neuen Methoden.

Aufgrund seiner ungewöhnlich hohen Auffassungsgabe ist er sehr an "**self-paced learning**" interessiert. Er will sich in seinem Lerntempo nicht bremsen lassen. Er **mag daher weder Class-Room-Tutorials** noch Videos. **Interaktionen mit anderen Teilnehmern sind ihm nicht wichtig**. Er möchte zu definierten Zeitpunkten sehr konkrete und tiefgehende **Fragen an einen Experten** stellen.

Außerdem sind ihm klare organisatorische Strukturen sehr wichtig. Er möchte sein Aufgabenfeld eindeutig abgrenzen gegen Disziplinen-spezifische Architekturen.

# Lernzielpyramide (\*)

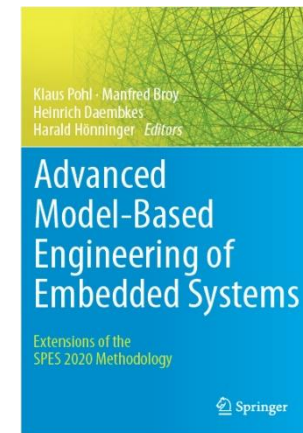
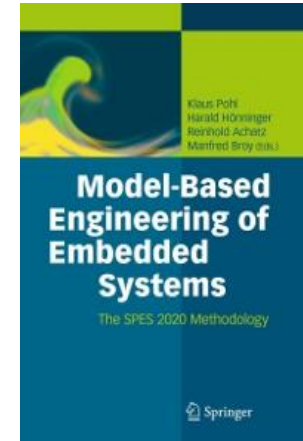
- **Klassifikationsschema** für Lernziele und Wissensvermittlung
- Bestimmung des **Niveaus**, **Aufwand**, **Anspruch** des Lernmaterials



(\*) Bloom, Benjamin S., Krathwohl, David R., Masia, Bertram B.: *Taxonomy of educational objectives – The classification of educational goals*. Longman, New York (1984)



- Verfügbare Quellen untersucht
- Resultate in **Deliverables**,  
**wissenschaftlichen Publikationen** sowie  
2 Bücher
- **Ca. 240 Konzepte, Techniken, Methoden**
- **Klassifizierung:**  
konstruktiv, analytisch, Building Blocks





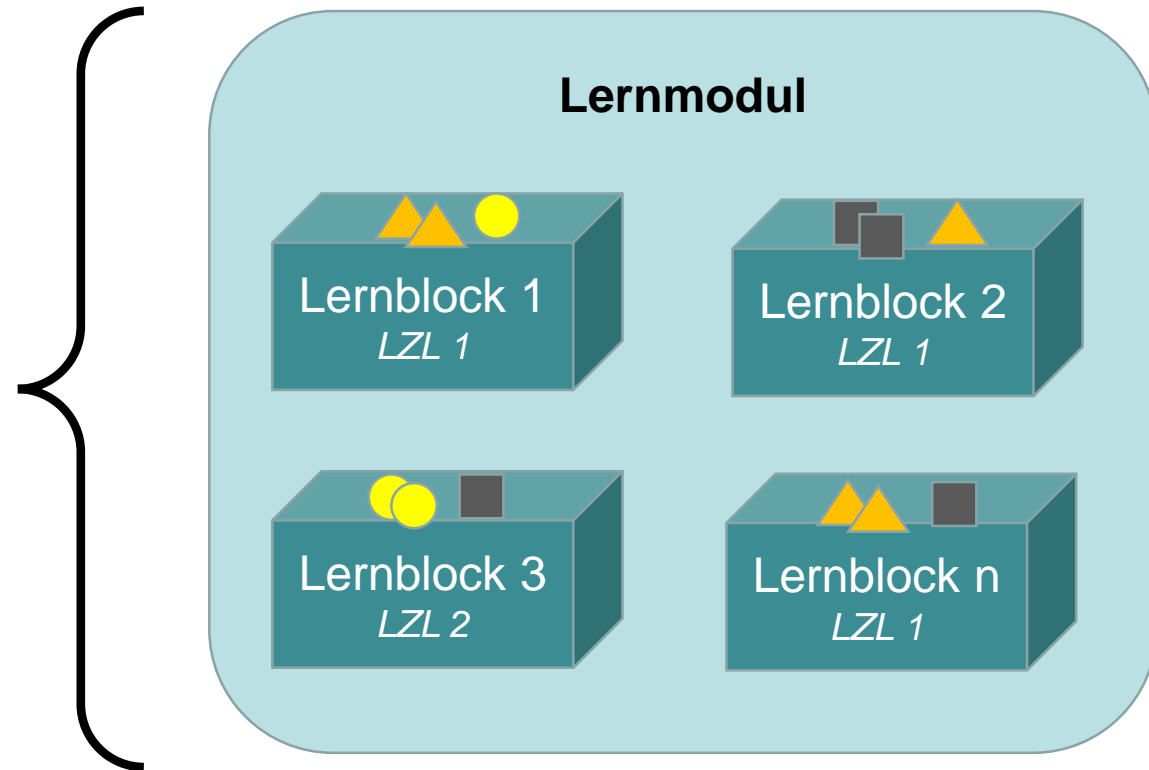
# Methodenlandkarte (Auszug)

ID	Resultat	Modul	Konzept	Modell	Methode / Technik			Tool Support		Kurzbeschreibung
					Konstruktiv	Analytisch	Building Block	PTC	AF3	
2.1.3.5	Verhaltens-Kontext	M01, M02, M03, M04	X	X						Der Verhaltens-Kontext modelliert das Verhalten der relevanten Entitäten und Funktionen im Kontext.
2.1.4	Integration in das SPES MF	M01, M02, M04	X	X						Alle SPES VPs auf allen Granularitätsebenen nutzen die Kontextmodelle, die in Form von Kontext-Diagrammen dokumentiert werden. In den unteren Granularitätsebenen ist die Kontext-Information eines Subsystems in dem Modellen der anderen Subsysteme enthalten.
2.1.5	Erweiterungen des SPES CMF für spezifische Themen		X	X						Safety, Validierung, Variabilität. Das SPES Blütenmodell
2.1.6	Kontextmodellierung mit PTC (SysML Blockdiagramme)	M01, M02, M03, M04	X		X			X		Siehe Kontextmodellierung
2.1.7	Kontextmodellierung mit AutoFOCUS3	M01, M02, M03, M04	X		X				X	Kontextmodellierung kann in AF3 mit Hilfe der logischen Architektur als Workaround durchgeführt werden, wobei dabei der funktionale operationelle Kontext im Vordergrund steht.
2.1.8	Spezifikation der Anforderungsquellen in AutoFOCUS3	M02	X	X					X	Templates mit vordefinierten Attributen für Anforderungsquellen unterstützen die Spezifikation in AutoFOCUS3. Die Quellen können im Tool mit den Anforderungen verlinkt werden. (siehe auch Toolvergleich AP2.D2.1)
2.1.9	Kontextmodellierung mit UML-Blockdiagrammen		X		X					Siehe Kontextmodellierung
2.1.10	Anwendung von Kontext Modellen	M01, M02, M03, M04			X	X				Beschreibung der Anwendungsmöglichkeiten von Kontextmodellen, e.g. Änderungsmanagement und Analyse der Auswirkungen, (semi-) automatische Artefaktgenerierung, Validierung und Verifikation, Variabilität.
2.2	Zielmodellierung	M01, M02	X		X					Im Requirements Engineering sind Ziele eine sehr abstrakte Beschreibung dessen, welche Funktionen das zu entwickelnde System erfüllen soll.
2.2.1	Zielmodellierung mit KAOS		X		X					Siehe Zielmodellierung
2.2.2	Zielmodellierung mit PTC (SysML-Requirements Diagramme)	M01, M02	X		X			X		Siehe Zielmodellierung
2.2.3	Zielmodellierung mit i*model		X		X					Dient der Modellierung der Abhängigkeiten zwischen Akteuren in den Fällen, in denen die Zielerfüllung eines Akteurs von einem anderen Akteur abhängt.
2.3	Szenariomodellierung	M01 - M05	X		X					Szenarien sind mögliche Realisierungen eines Ziels in einem Kontext.
2.3.1	Szenariomodellierung PTC (SysML)	M01 - M05	X		X			X		Siehe Szenariomodellierung
2.3.2	Szenariomodellierung mit AutoFOCUS3	M01 - M05	X		X				X	In AF3 existiert die Szenariomodellierung als spezielle Form der Use Case Modellierung. (siehe auch Toolvergleich AP2.D2.1)
2.3.3	Szenariomodellierung mit ITU-MSD		X		X					Siehe Szenariomodellierung



Auf **eine** Rolle  
zugeschnitten

z.B. System Architect



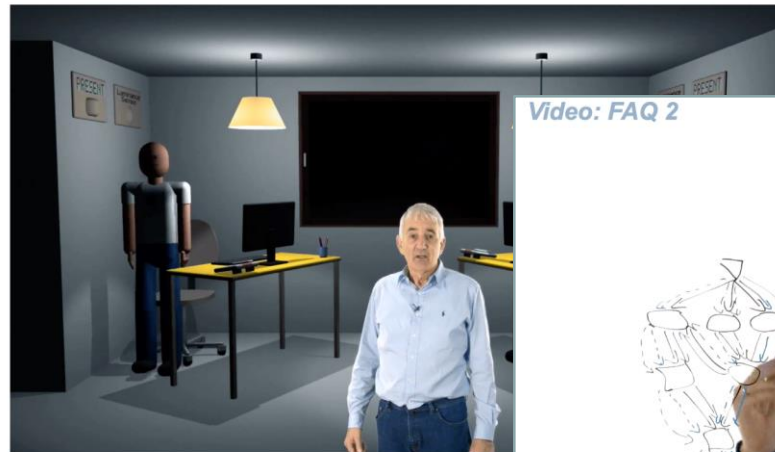
Medienbezogene Bausteine

- Stellt eine **inhaltliche Einheit** dar
- Kann aus mehreren **medienbezogenen Bausteinen** bestehen
  - **Skript**
  - **Video**
    - Tagesschaumodus (Folienpräsentation mit Vortrag)
    - Plexiglas-Präsentation
    - Diskussion (z.B. Webinar)
    - Tool-Demo
  - **Praktische Übungen**
    - Multiple Choice Fragen
    - Pen & Paper Übungen
    - Tool basierte Übungen
- Zeit für **Lernzielerreichung** (inkl. Übungen) **ca. 1-3 Stunden**

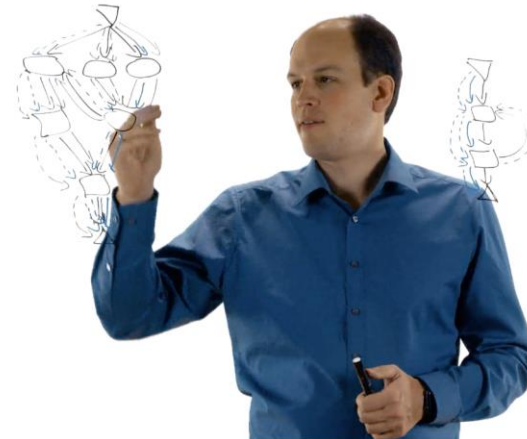
Video: Kontextmodellierung

Wissenskontext

Video: Home-Übergreifende Beispiele



Video: FAQ 2

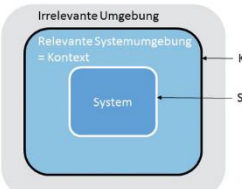


Um den Kontext insgesamt zu strukturieren und einen besseren Überblick behalten zu können, bietet es sich an, Kontexte bzw. deren Modelle in verschiedene Typen zu unterteilen. Durch Fokussierung auf einzelne Aspekte einfacher, die Fülle an Informationen angemessen zu strukturieren – wodurch insbesondere auch Komplexität des Systemlebenszyklus des betrachteten Systems besser beherrschbar ist.

Der Kontext eines Systems lässt sich in verschiedene Aspekte kategorisieren, die sich in der Anwendung der SPES-Methode unterscheiden zunächst zwei Kontextarten: den **operationalen Kontext** und den **systemischen Kontext**. Beim operationalen Kontext wiederum unterschiedliche Perspektiven. Mit diesen Strukturierungsmitteln zur Komplexitätsverringering werden wir uns im weiteren Verlauf dieser Veranstaltung näher befassen.

### 1.1 Systemgrenze und Kontextgrenze: Leit als Kriterium

Der Kontext eines Systems hat zwei Grenzen: zum System hin (Systemgrenze) und zur irrelevanten Umgebung (Kontextgrenze) hin (vgl. Abbildung 1).



Bei der Modellierung des Kontextes ist genauestens darauf zu achten, dass die Grenzen des Kontextes in den allermeisten Fällen eindeutig sind. Ein gegebenes „Ding“ (a) dem System, (b) dem Kontext zugeordnet ist, so ergeben sich doch auch Zweifelsfälle, bei denen es hilfreich sein kann, Entscheidungen zu treffen.

Der Prozess der Festlegung der Systemgrenze wird „**Scoping**“ genannt, der Prozess der Festlegung der Kontextgrenze wird als „**Contexting**“ bezeichnet.

geschieht nicht willkürlich. Man benötigt objektive Kriterien, nach denen man entscheidet, ob dieses oder jenes Element noch im Skopus liegt oder eben nicht. Daher wollen wir im Folgenden jeweils näher auf die Faktoren eingehen, die bei der Festlegung der System- und Kontextgrenze relevant sind.

Der Prozess des Bestimmen einer Systemgrenze, die das Kontext abgrenzt, heißt „**Scoping**“. Das Innerhalb der Systemgrenze liegt im Skopus („**Scope**“) des Entwicklungsprozesses.

[Video: Kontextmodellierung – „Grauzonenvideo“]

Als Erstes müssen Sie sich klarmachen: Weder System- noch Kontextgrenzen sind unveränderbar oder umkehrbar. Alles im Zuge des Entwicklungsprozesses Veränderbare gehört zum System, alles Unveränderbare gehört zum Kontext. Nun ist es aber zweierlei, ob etwas unveränderbar ist, so **bleiben wird**, oder ob es nur von Projektdetails abhängt, dass es als unveränderbar betrachtet wird.

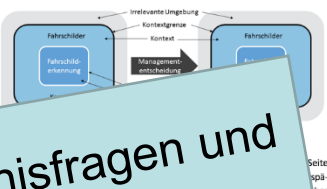
Der Kontext eines Systems, des sogenannten „**Context**“, ist das Kontextsubjekt selbst Teil des Kontextes. Das Kontextsubjekt dient Verändern der Kontextgrenze. Veränderbare gehören zum System, unveränderbare zum Kontext.

Den Unterschied zwischen Systemgrenze und Kontextgrenze verdeutlicht das Beispiel der Fahrschildererkennung. Die Fahrschildererkennung ist ein System, das die Fahrschildererkennung in der Kamera verarbeitet. Die Fahrschildererkennung ist ein System, das die Fahrschildererkennung in der Kamera verarbeitet. Die Fahrschildererkennung ist ein System, das die Fahrschildererkennung in der Kamera verarbeitet.

Der Unterschied zwischen den Fahrschildern und den Kameras ist, dass die Fahrschildererkennung ein System ist, das die Fahrschildererkennung in der Kamera verarbeitet. Die Fahrschildererkennung ist ein System, das die Fahrschildererkennung in der Kamera verarbeitet.

System ist, das Kamera nutzt. In einem solchen Fall würde man sie zum System hinzuzählen. Letztlich liegt es immer im Ermessensspielraum des Managements, zu entscheiden, ob etwas Teil des Systems wird oder nicht.

Allgemein kann man sagen, dass Sie in der Praxis in allen drei Bereichen – System, Kontext und irrelevanten Umgebung – mit Veränderungen zur Laufzeit eines Entwicklungsprojektes rechnen sollten.



Seite 10 über Kontextgrenze

Angewandt auf das ESP-Beispiel würden Sie sich also zum Beispiel fragen:

- Kann das ABS (in einem Auto) ohne das ESP die gleiche Funktion erfüllen?
- Ja, offensichtlich, also ist das ABS Kontextobjekt.
- Kann (z.B.) das ESP-Hydroaggregat ohne das ESP die gleiche Funktion erfüllen?
- Nein, also ist das Hydroaggregat eine Komponente des Kontextsubjektes.

Relevante Systeme, die für das Funktionieren des betrachteten Systems notwendig sind, sind entweder Kontextobjekte oder Systemkomponenten. Wenn die betreffende Entität in einem Produkt verbaut werden kann, das nicht über das Kontextsubjekt verfügt, und die Funktion, für die es vorgesehen ist, dort erfüllen kann, so handelt es sich in diesem Sinne um ein eigenständiges System. In diesem Fall ist es ein **Kontextobjekt**.

Das Gegenstück zum Kontextobjekt ist die **Systemkomponente** (oder Subsystem). Wenn die betreffende Entität in einem Produkt verbaut werden kann, das nicht über das Kontextsubjekt verfügt, und die Funktion, für die es vorgesehen ist, dort nicht erfüllen kann, so handelt es sich um ein System, dessen Funktionen vom Kontextsubjekt abhängen. In diesem Fall ist es eine Systemkomponente.

Die obigen Ausführungen betreffen allesamt die Systemgrenze. Die „äußere“ Grenze des Kontextes, die Kontextgrenze, ist manchmal noch viel weniger eindeutig zu bestimmen. Alles, was für das betrachtete System relevant ist, wird dem Kontext zugerechnet. Neben Erfahrung und fachlichem Verständnis helfen bei der Beurteilung der Relevanz einer gegebenen Entität die in diesem Skript enthaltenen Hinweise.

### Kontrollfragen

1. Kann sich der Kontext im Laufe eines Projektes verändern?
2. Kann sich die Systemgrenze im Laufe eines Projektes ändern?
3. Nach welchen Kriterien entscheiden Sie, ob ein System als Subsystem/Komponente oder als Kontextentität modelliert wird?
4. Welchem Zweck dient die Strukturierung des Kontextes?
5. Wie kann man den Kontext sinnvoll strukturieren?

1.3 Qualitätsmerkmale eines Kontextmodells

Ziel dieses Skriptes ist es, wie gesagt, Sie dabei zu unterstützen, gute Kontextmodelle zu erstellen. Diesem Zweck dienen einerseits die im Folgenden versammelten Hilfestellungen, die Sie bei der Erstellung geeigneter Modelle unterstützen. Andererseits benötigen Sie auch **Qualitätskriterien**, die Ihnen im Nachhinein die Möglichkeit geben, ein von Ihnen erstelltes Modell hinsichtlich seiner Qualität zu bewerten.

Verständnisfragen und kleine Tests

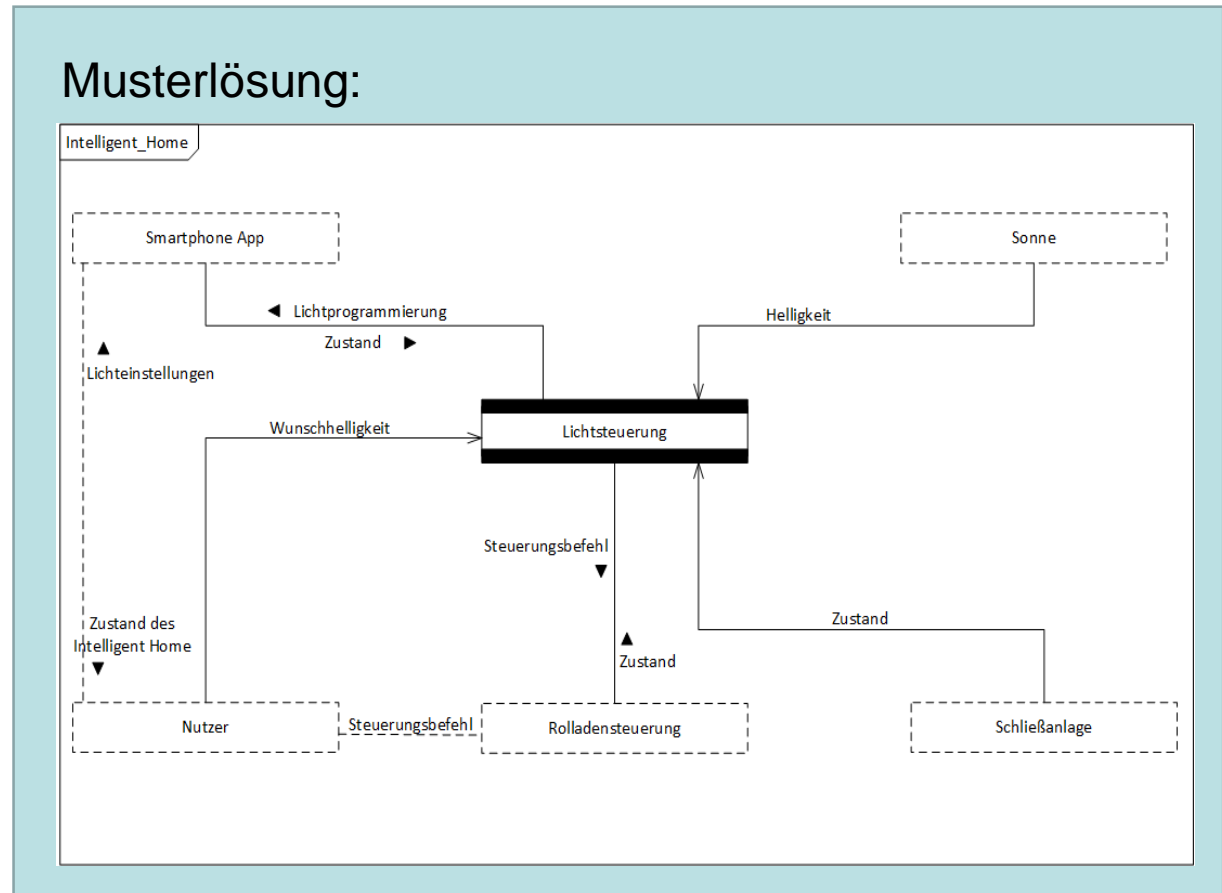
1. Kann sich der Kontext im Verlaufe eines Projektes verändern?
  - a. Ja. Er kann sowohl anwachsen als auch schrumpfen. Sowohl die System- als auch die Kontextgrenze können sich ändern.
  - b. Ja. Das genaue Ausmaß des Systemkontextes lässt sich nie exakt bestimmen.
  - c. Nein. Die Festlegung von System- und Kontextgrenze ist eine der frühesten Tätigkeiten des Requirements Engineering und ist abgeschlossen, bevor Arbeiten in nachgeordneten Viewpoints beginnen.
  - d. Nein. Da sich sowohl System- als auch Kontextgrenze unmittelbar aus der Systemvision ableiten lassen, ist eine spätere Veränderung des Kontextes ausgeschlossen.
2. Kann sich die Systemgrenze im Verlaufe eines Projektes ändern?
  - a. Ja.
  - b. Nein.
3. Nach welchen Kriterien entscheiden Sie, ob ein System als Subsystem/Komponente oder als Kontextentität modelliert wird?
  - a. Das hängt davon ab, ob die Funktion des infrage stehenden Systems vom Kontextsubjekt abhängig ist.
  - b. Das hängt davon ab, ob die Integration als Systemkomponente die Komplexität des statisch-strukturellen Modells negativ beeinflusst.
  - c. Das hängt davon ab, ob die Entwicklung des betreffenden Systems außer Haus durch Dritte erfolgt.
  - d. Das hängt davon ab, ob das Kontextsubjekt auch unabhängig von der potentiellen Kontextentität funktionieren würde.



## Aufgabenstellung:

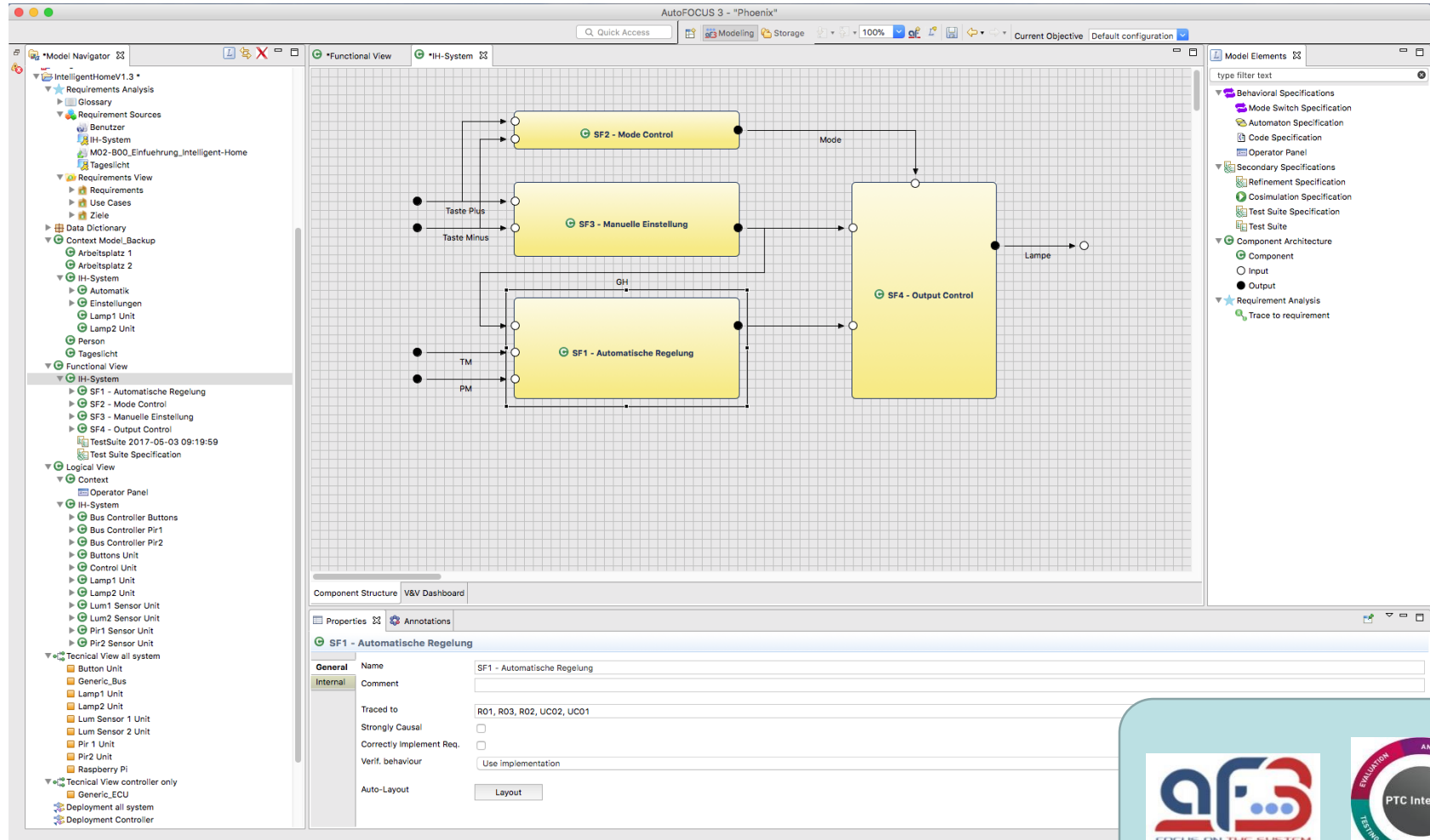
Modellieren Sie anhand der Ihnen vorliegenden Unterlagen zur Kontextmodellierung und dem Intelligent Home Beispiel, die **statisch-strukturelle Perspektive des operationellen Kontextes**.

## Musterlösung:





# Praktische Übungen: z.B. Tool-Übungen

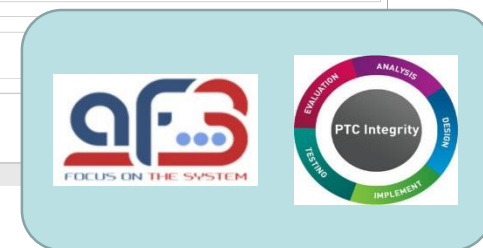


The screenshot displays the AutoFOCUS 3 software interface for a project named "Phoenix". The main workspace shows a functional view diagram with four components: SF2 - Mode Control, SF3 - Manuelle Einstellung, SF1 - Automatische Regelung, and SF4 - Output Control. SF2 and SF3 are grouped together, and SF1 and SF4 are grouped together. Inputs include "Taste Plus", "Taste Minus", "TM", and "PM". Outputs include "Mode" and "Lampe".

The left sidebar shows a hierarchical model structure under "IntelligentHomeV1.3", including "Requirements Analysis", "Data Dictionary", "Functional View", and "Logical View".

The bottom panel shows the "Properties" view for "SF1 - Automatische Regelung".

Category	Property	Value
General	Name	SF1 - Automatische Regelung
	Comment	
Internal	Traced to	R01, R03, R02, UC02, UC01
	Strongly Causal	<input type="checkbox"/>
	Correctly Implement Req.	<input type="checkbox"/>
	Verif. behaviour	Use implementation
	Auto-Layout	Layout



## Modul 0: SPEDiT Übersichts- und Einführungsmodul

Einführung in MBSE und SPES-Methodik  
Schulungsplattform und Lernmaterial  
Kurzbeschreibung der Lernmodule  
Rollen  
Werkzeuge  
Entwicklungsprozess

## Modul 1: System Architect

Anforderungsmodellierung	L1
Funktionsarchitektur	L2-3
Logische Architektur	L2-3
Deployment	L1

## Modul 2: System Requirements Engineer

Kontext- und Zielmodellierung	L2
Szenariomodelle	L2
Spezifikation / Analyse von Anforderungen	L2-3
Verfeinerung von Anforderungen	L2-3

## Modul 3: Test Engineer

Anforderungsmodellierung	L1-2
Modellierung von Testfallspezifikationen	L2-3
Testausführung	L3
Simulation	L3

## Modul 4: Software Architect

Kontext- und Zielmodellierung	L2
Szenariomodelle	L2
Spezifikation / Analyse von Anforderungen	L2-3
Funktionsarchitektur	L2-3
Logische Architektur	L2-3

## Modul 5: Software Developer

Anforderungsmodellierung	L1
Spezifikation von Softwarekomponenten	L2
Formale Analyse	L3
Simulation	L3
Scheduling Synthese	L3

	Block Bezeichnung	Lernziel-Level
<b>Anforderungsmodellierung</b>	Der SPES Modeling Framework	1
	Kontextmodellierung	1
	Zielmodellierung	1
	Szenariomodellierung	1
	Spezifikation von Anforderungen	1
	Reflexion der Spezifikation von Anforderungen im funktionalen Systementwurf	
<b>Funktionsarchitektur</b>	Zerlegung des Systems in Systemfunktionen (1)	2
	Zerlegung des Systems in Systemfunktionen (2)	2
	Spezifikation von Systemfunktionen (1)	2
	Spezifikation von Systemfunktionen (2)	2
	Abbildung von Anforderungen auf Systemfunktionen (1)	2
	Abbildung von Anforderungen auf Systemfunktionen (2)	2
	Analyse der Funktionsarchitektur gegen Anforderungen (1)	3
	Analyse der Funktionsarchitektur gegen Anforderungen (2)	3
	Reflexion der Spezifikation und Analyse von Funktionsarchitekturen	
<b>Logische Architektur</b>	Zerlegung des Systems in Teilsysteme (1)	2
	Zerlegung des Systems in Teilsysteme (2)	2
	Analyse der logischen Architektur gegen die funktionale Architektur (1)	3
	Analyse der logischen Architektur gegen die funktionale Architektur (2)	3
<b>Technische Architektur</b>	Erstellung einer technischen Architektur (1)	2
	Erstellung einer technischen Architektur (2)	2
	Verteilung der logischen Softwarekomponente auf Hardwarekomponenten	1
	Reflexion Architektur und Deployment	



Gedrucktes  
Exemplar liegt im  
Marketplace aus

- Unterstützung der **Entwicklungsingenieure** bei der **Anwendung** der SPES Konzepte, Techniken und Methoden
- Unterstützung der **Prozessverantwortlichen** bei der **Anpassung** von Konzepten, Techniken und Methoden für einen spezifischen Entwicklungskontext

**Vielen Dank für Ihre Aufmerksamkeit!**

