

GEFÖRDERT VOM



Bundesministerium  
für Bildung  
und Forschung



# SPEDiT Methoden verstehen ... an einem praktischen Beispiel



Dr. Helmut Lagger

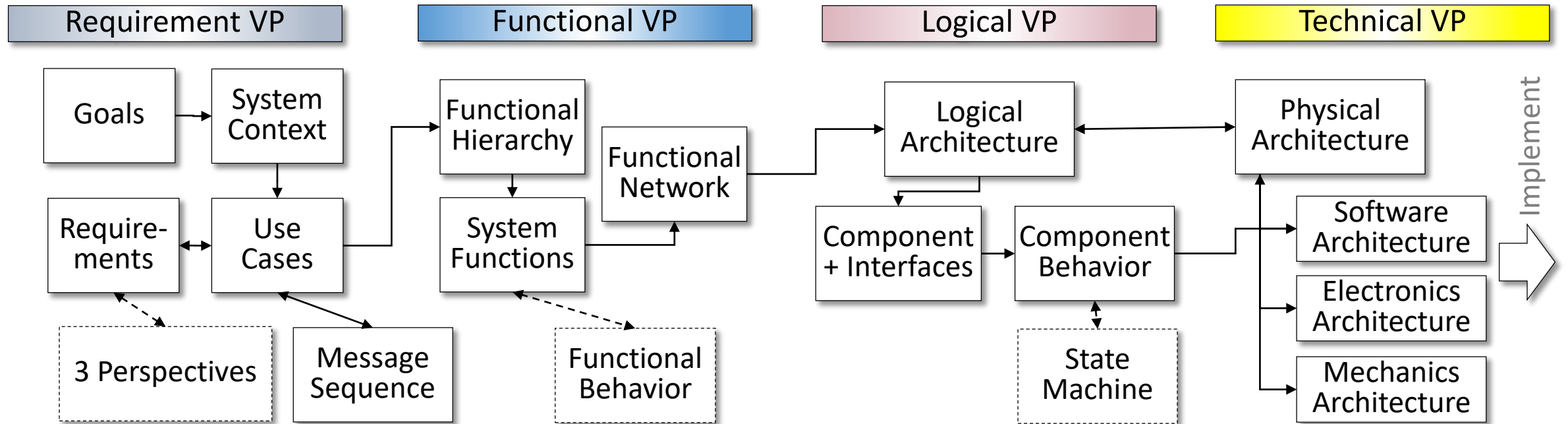
GPP Communication GmbH & Co KG  
Kolpingring 18  
Oberhaching

SPEDiT Abschlussveranstaltung 12.04.2019

# SPES System Modellierung - Prozess Beispiel

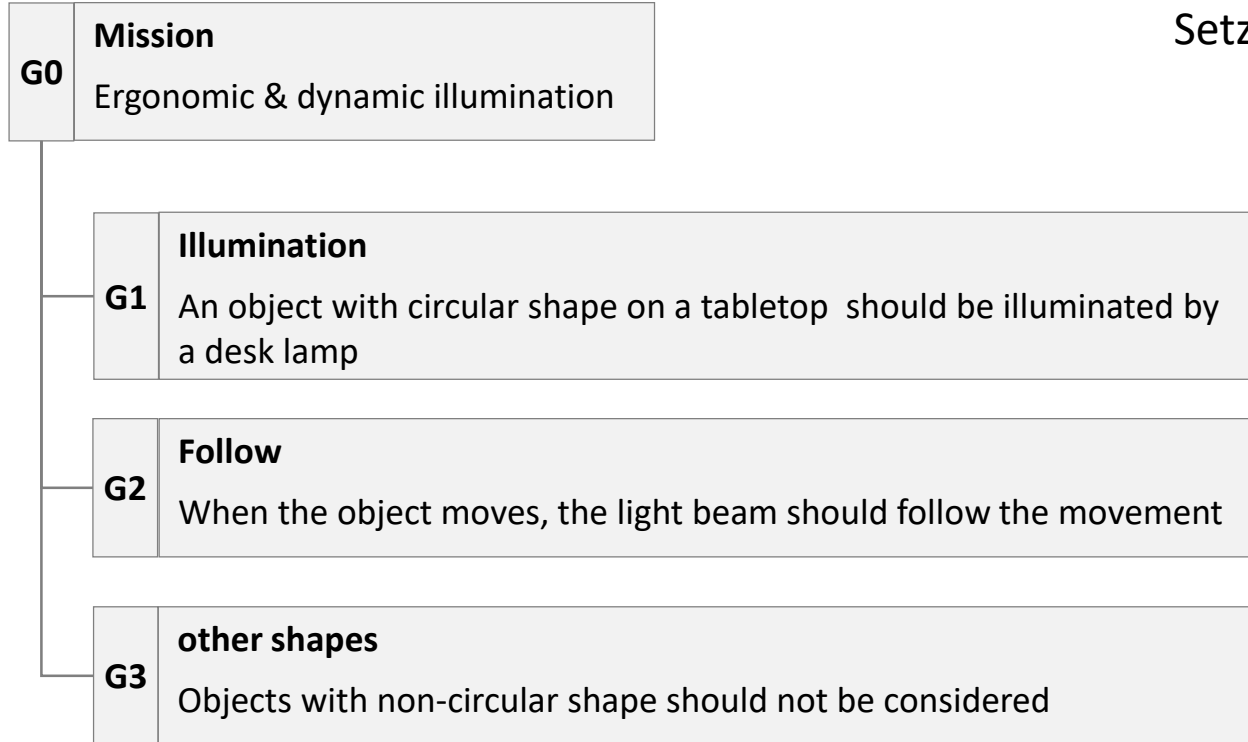


- View Points (VP) gruppieren die Anwendung von Methoden um Modelle schrittweise zu erstellen und zu verbessern
- Details der Sub-Systeme werden in der nächsten Granularitätsebene adressiert

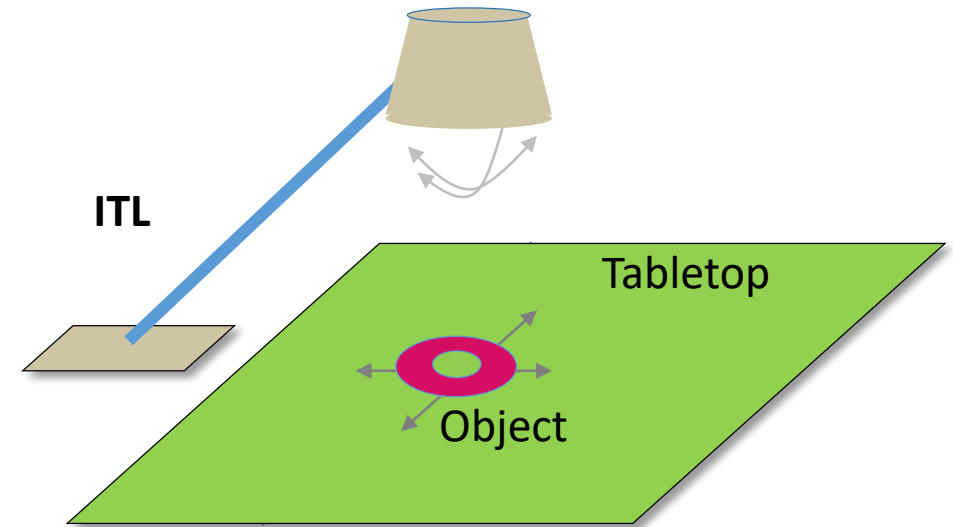


Each box represents a method producing a SPES artifact according to the name of the box.

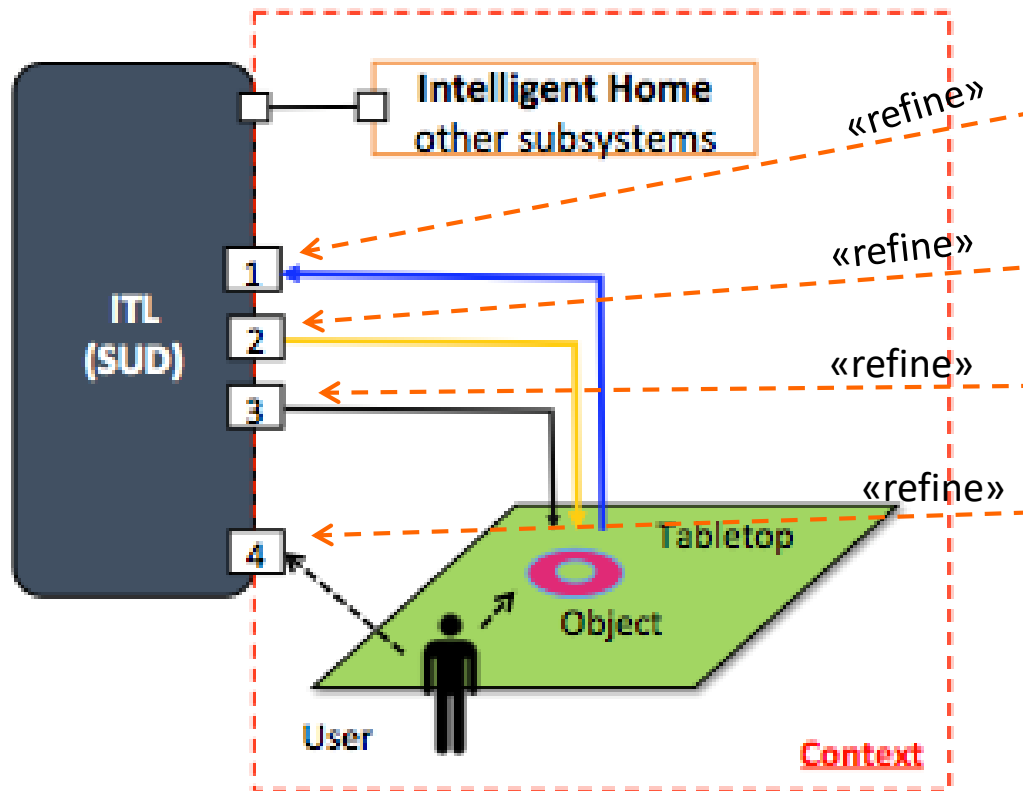
# Intelligent Table Lamp (ITL) – Ziele



Setze high-Level Ziele für das Projekt - aus Benutzer Sicht  
Ziele werden auch für das “Scoping” genutzt



# ITL Requirements – in Bezug auf das Kontext Modell



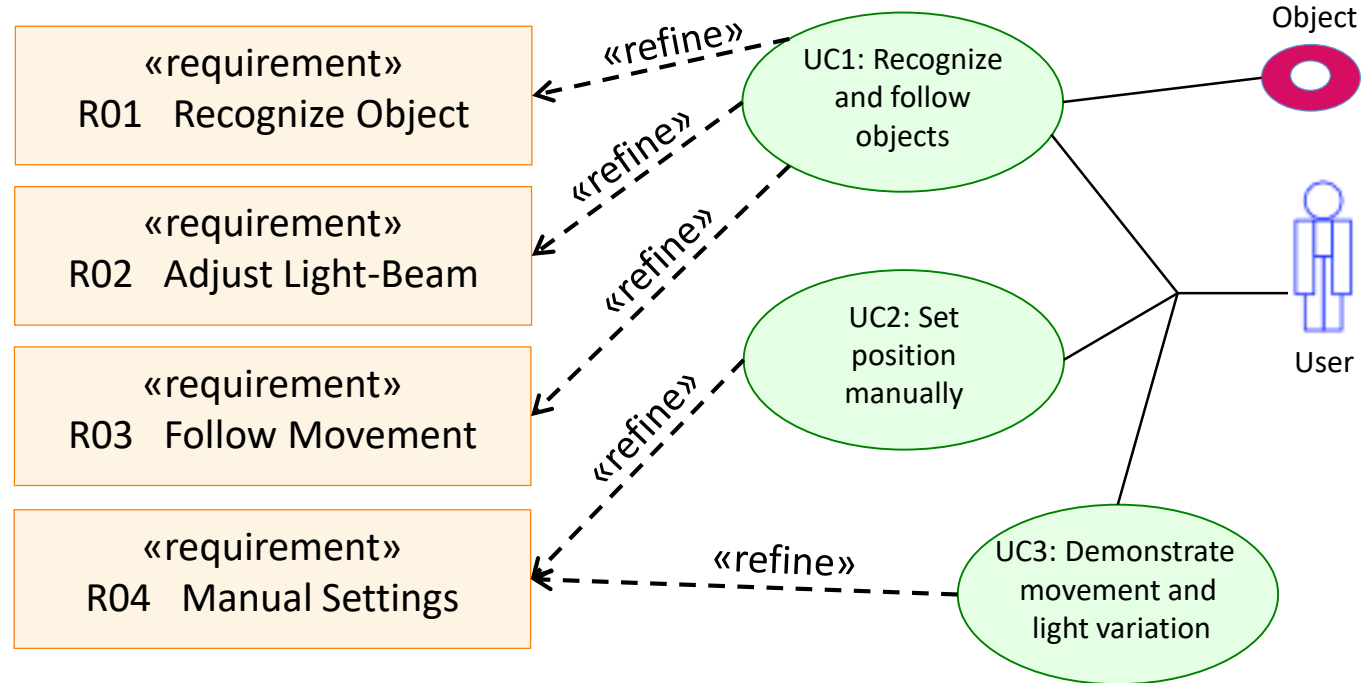
ID	name	txt
R01	Recognize Object	The lamp should recognize a round object on a homogeneous tabletop surface
R02	Adjust Light-Beam	The lamp should align the center of the light beam with the center of the round object
R03	Follow Movement	If the detected object moves, the lamp should follow the movement. R02 continues to apply (the center of the light beam should align with the center of the round object)
R04	Manual Settings	The user should be able to set the lamp manually via a web interface. Position, brightness and color of the light should be adjustable.

**Check** ob alle Schnittstellen des Kontextmodells und die Anforderungen konsistent sind

**Check** ob die Anforderungen das erwartete vollständige SUD-Verhalten beschreiben

# System Requirements und Szenarien / Use Cases

- Use Cases enthalten eine Reihe von Szenarien
- Szenarien verfeinern die Systemanforderungen (in diesem Beispiel)
- Für die Rückverfolgbarkeit sollte die Beziehung sichtbar gemacht werden (hier: Use Cases "verfeinern" Requirements)



**Check** ob die Anforderung erfüllt ist, wenn das System den Anwendungsfall im angenommenen Kontext ausführt

# Use Case/Szenarien – spezifiziert durch ein Sequence Diagram

## Use Case UC1 “Recognize and follow object” Scenario 1.1

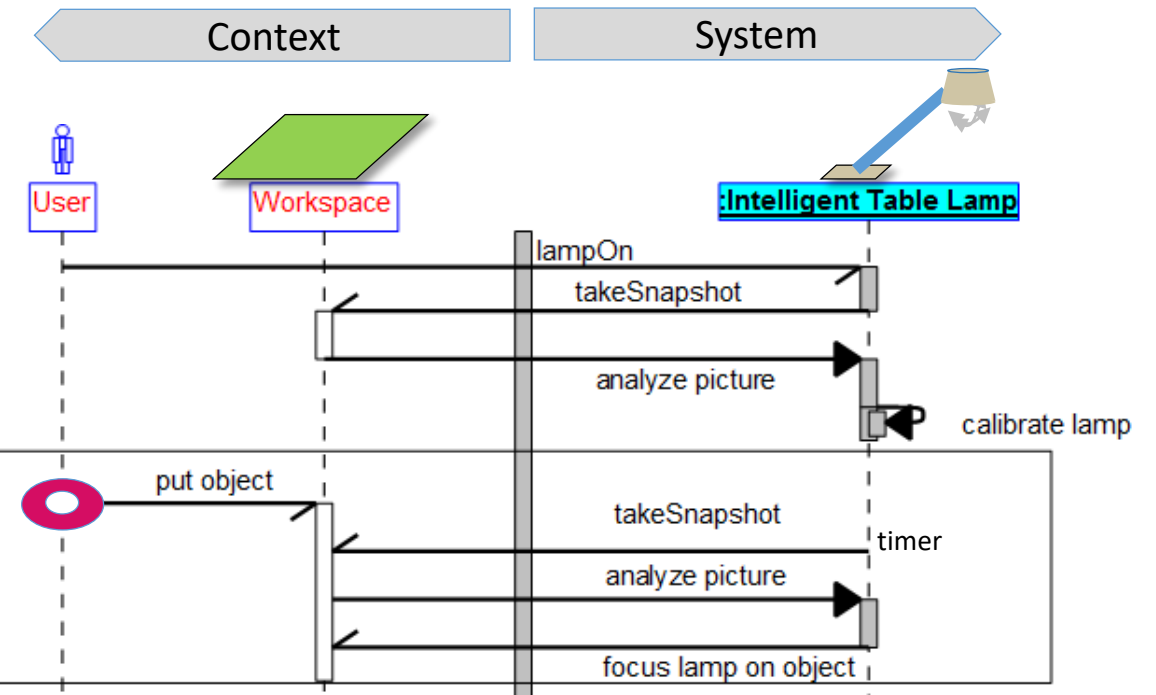
### Observation Mode

#### Description

Lamp on  
take snap shot  
analyze picture  
calibrate lamp

#### UC1, Scenario 1.1

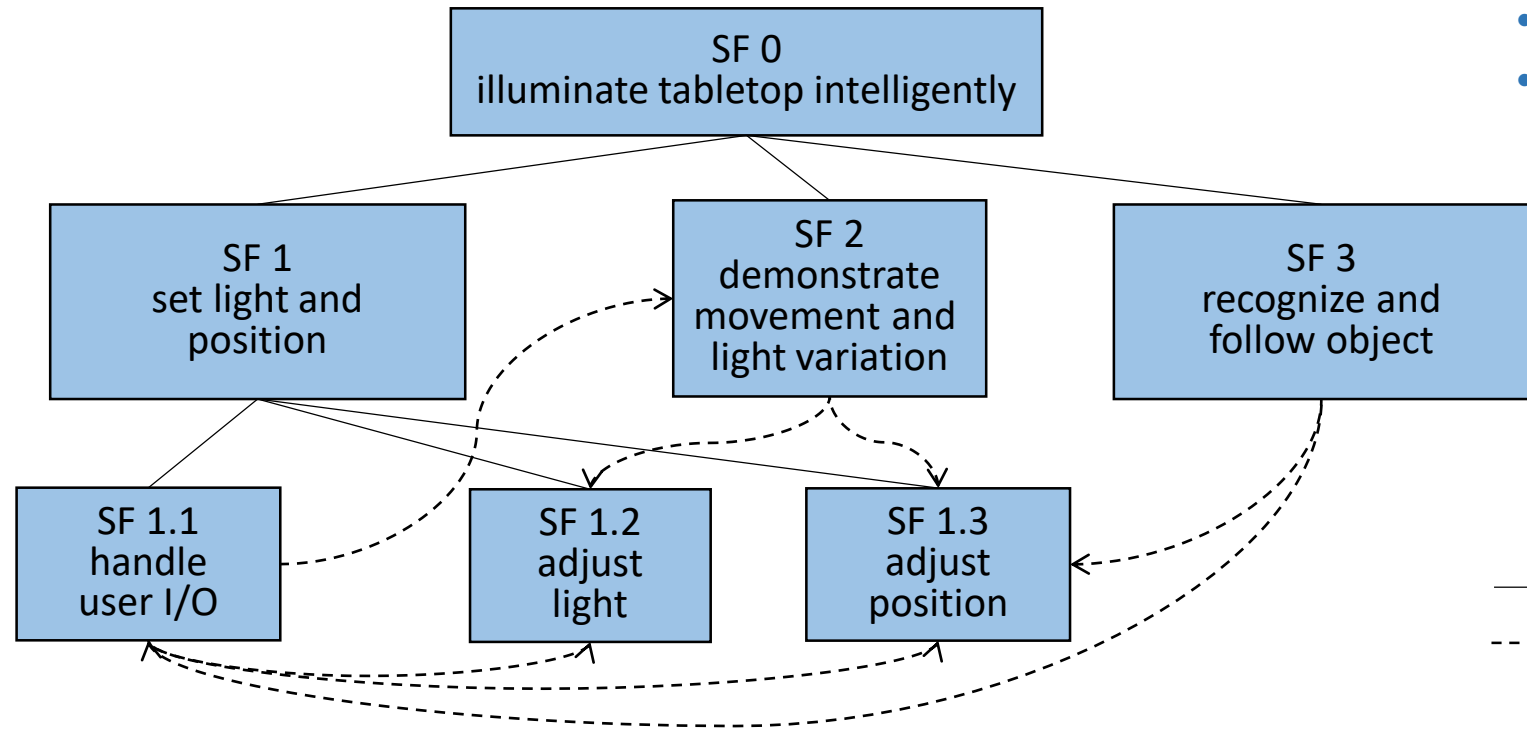
put object  
take snap shot  
analyze picture → object position (x,y) on the image  
focus lamp on object



system border

Das Sequence Diagram spezifiziert die Szenarien detailliert.  
Es beschreibt die genaue Interaktion des Kontextes mit dem SUD.  
Szenarien können auch Alternativ- oder Ausnahmefälle beschreiben.

# Funktionale System Hierarchie - "Black-Box Model"



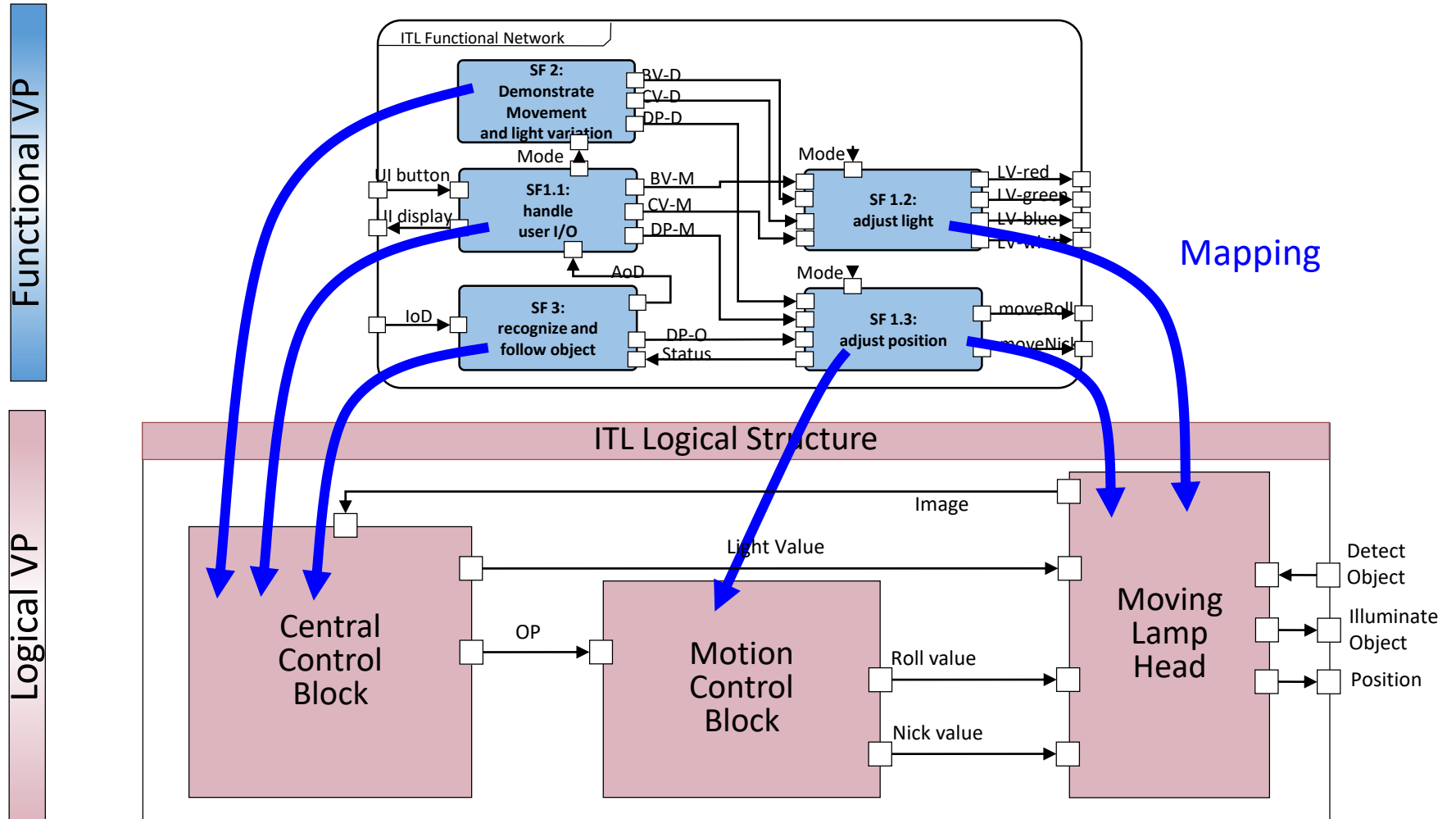
- Nur Systemfunktionen zeigen
- Funktionsname <Action-Object>

System Functions = Leaves of the tree  
—— Function is part of the parent function  
- - - - Functional dependency (Modes)

- Für das System als Black Box werden die Systemfunktionen als Hierarchie heruntergebrochen
- Abhängigkeiten zwischen Funktionen zeigen Modes an

# Erstellung einer logischen Architektur aus den System Funktionen

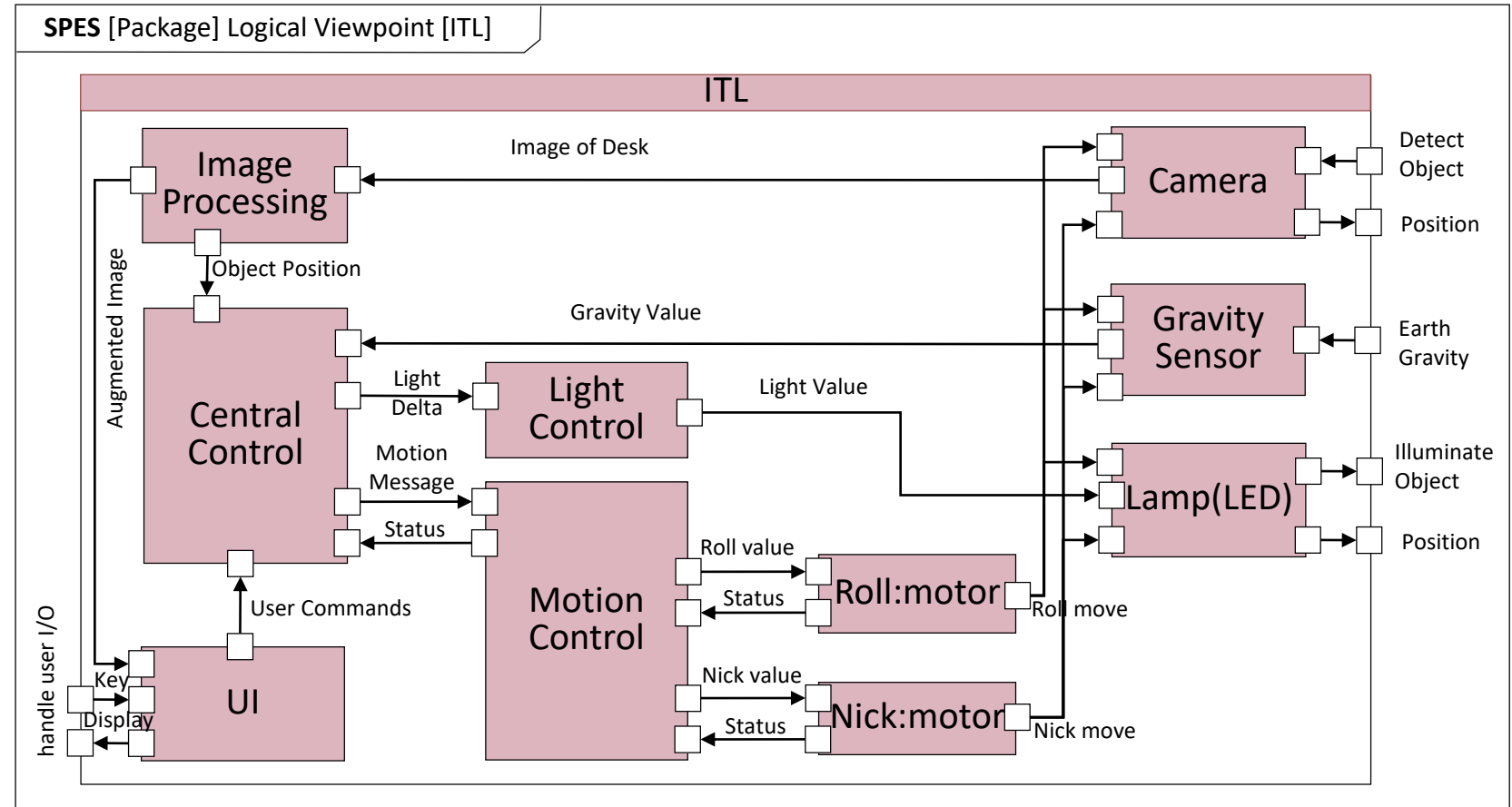
- Das Block-Diagramm für die logische Architektur des Systems wird basierend auf den Systemfunktionen erstellt
- "Mapping" ordnet Systemfunktionen logischen Komponenten zu
- Eine oder mehrere logische Komponenten können eine oder mehrere Systemfunktionen realisieren → n: m





# Logical Viewpoint – ITL Architektur

- Die ITL-Architektur wird in detailliertere logische Komponenten zerlegt
- Logische Komponenten können ggf. wiederverwendet werden
- Schnittstellen zwischen den Komponenten beschreiben den Austausch von Daten
- Das Verhalten jeder Komponente wird in einem weiteren Schritt definiert



# Design Entscheidungen ... die mit dem Kunden abzustimmen sind

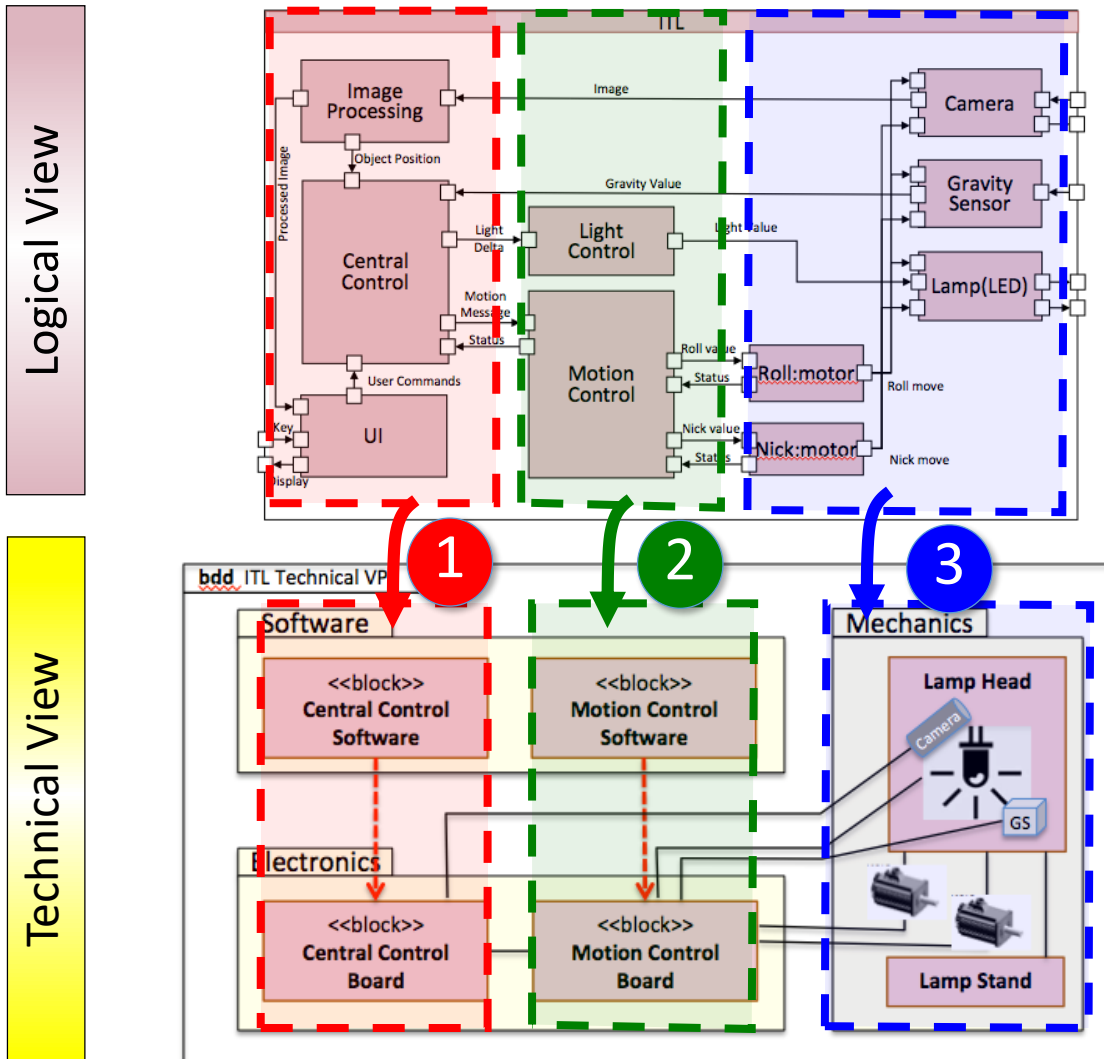


- Bevor man zum Technical VP wechselt, müssen i.a. einige Design-Entscheidungen getroffen werden
- Design-Entscheidungen mit Auswirkungen auf das Systemverhalten oder die Schnittstellen führen zu weiteren „Constraints“. Diese sollten mit dem Kunden abgestimmt werden – wie die Requirements



ID	Title	Description
R01	Object recognition	The lamp should recognize a round object on a homogeneous desk surface
R02	Beam alignment	The lamp should align the center of the light beam with the center of the round object
R03	Motion detection	If the detected object moves, the lamp should follow the movement. R02 continues to apply (the center of the light beam should align with the center of the round object)
R04	Manual settings	The user must be able to set the lamp manually via a web interface. Position, brightness and color of the light beam are adjustable.
R05	Tolerance in Positioning	Tolerance range of the deviation of the light beam from the center of the object is 5 cm
R06	Speed of positioning	The positioning should be reached within 1 sec
R07	Starting Position	The head of the lamp should be horizontal after initialization (Gravity Sensor)
R08	Costs	The cost of the lamp should be under xx €

# Mapping Logical- to Technical-View

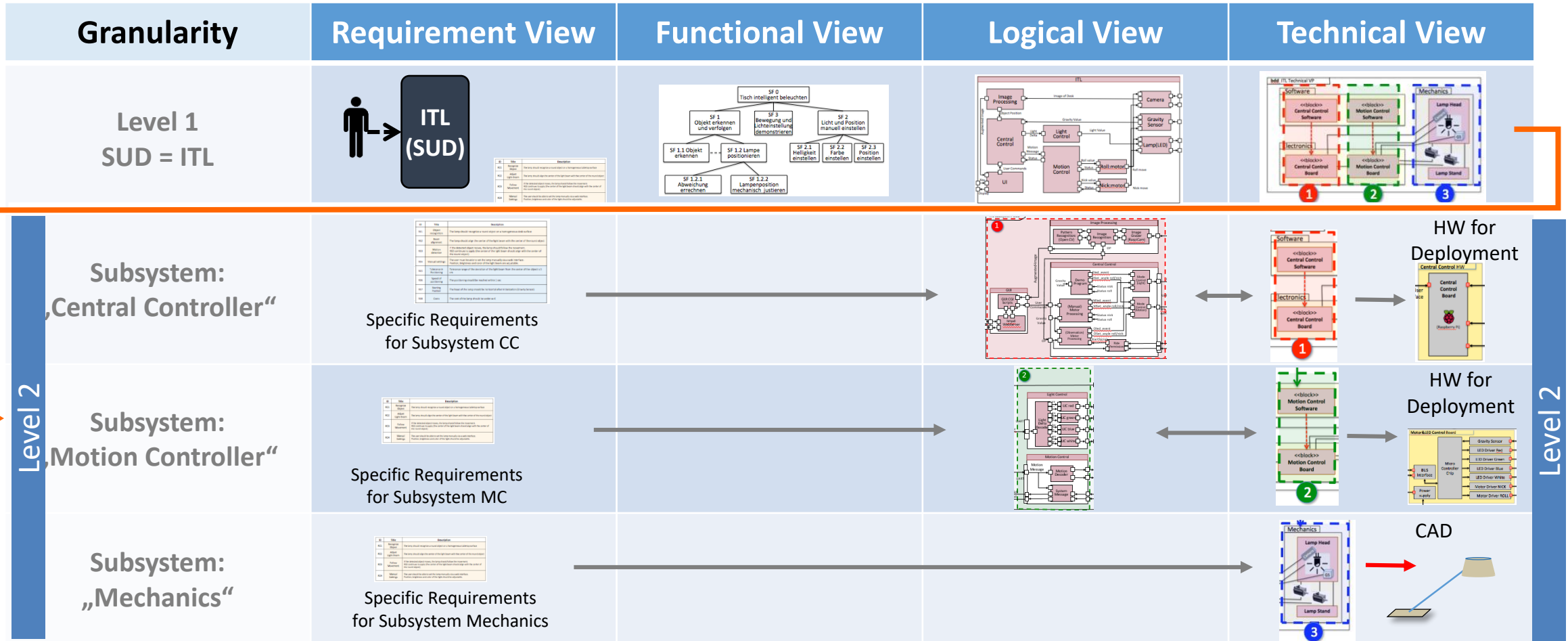


Drei Subsysteme wurden in der 1. Ebene des Technical-VP identifiziert.

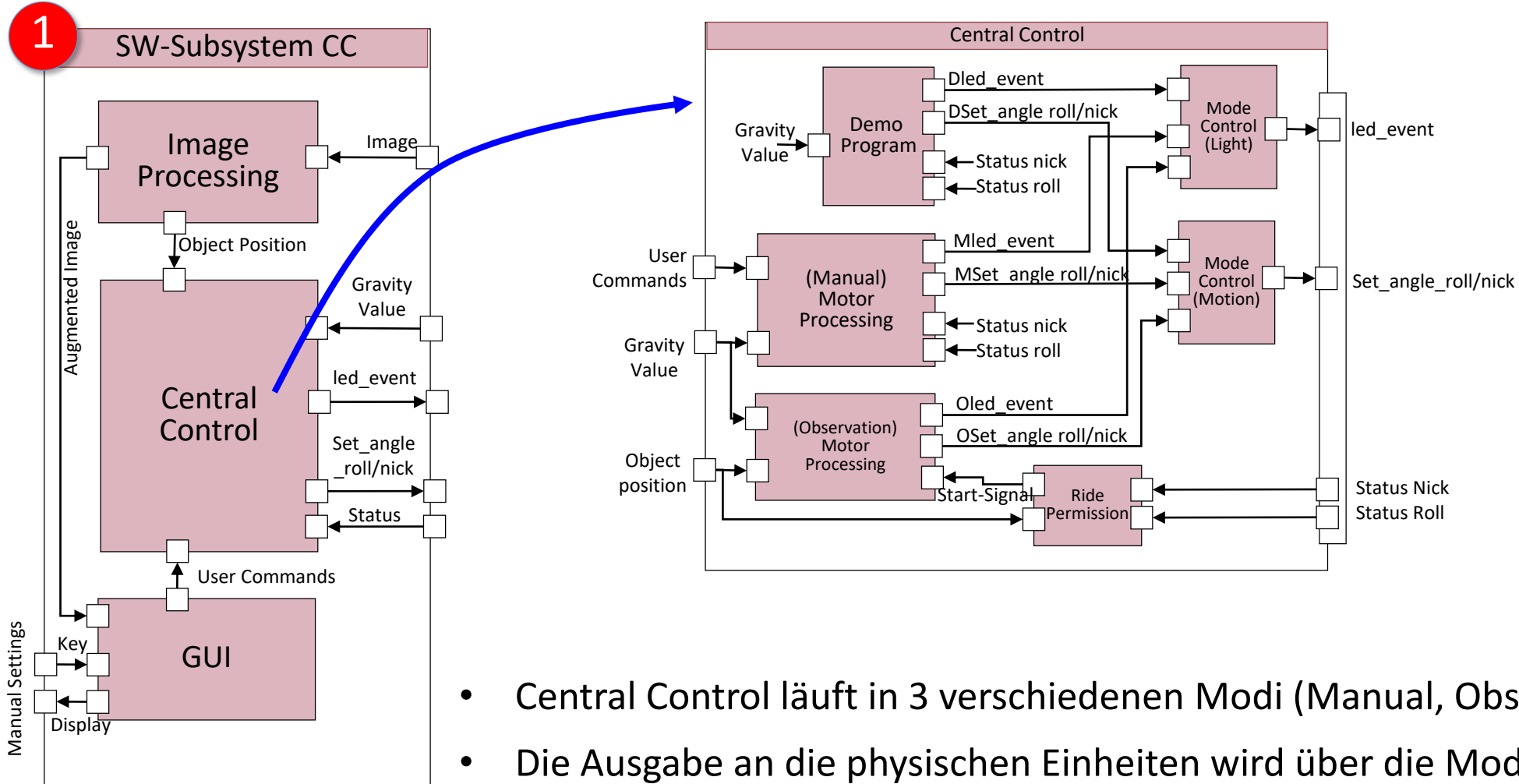
Weitere Details in der 2. Granularitätsstufe

- 1 Das **SW-Subsystem "Central Controller"** umfasst die gesamte Logik der zentralen Steuerung einschließlich der Benutzeroberfläche und der Bildverarbeitung und läuft auf dem CC-Board ab
- 2 Das **SW-Subsystem „Motion Controller“** beinhaltet die gesamte Logik der Motion Control einschließlich LED-Steuerung und läuft auf dem MC-Board ab.
- 3 Das **"Mechanics Subsystem"** beinhaltet die mechanische Montage der Lampe mit LED-Licht, Kamera, Sensor und den Motoren für die Bewegung in x- und y-Richtung

# Granularitätsebene 1 & 2

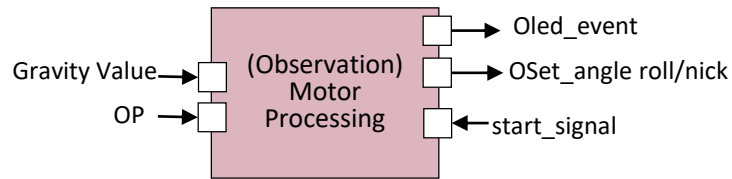


# Dekomposition des SW-Subsystems Central Controller (CC)



- Central Control läuft in 3 verschiedenen Modi (Manual, Observation, Demo)
- Die Ausgabe an die physischen Einheiten wird über die Mode Control verwaltet

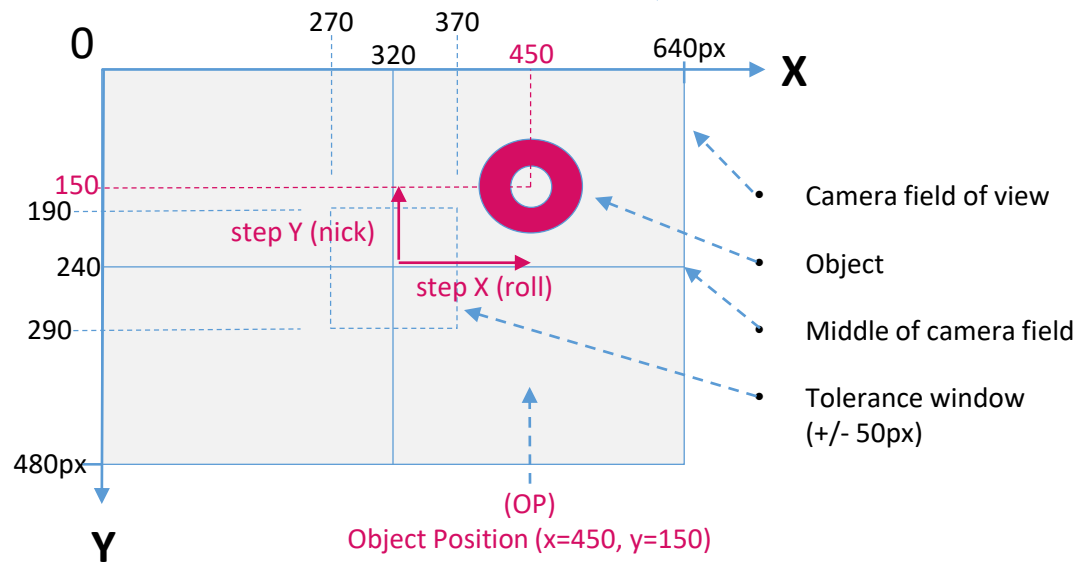
# CC im Observation Mode – Komponente “Motor Processing”



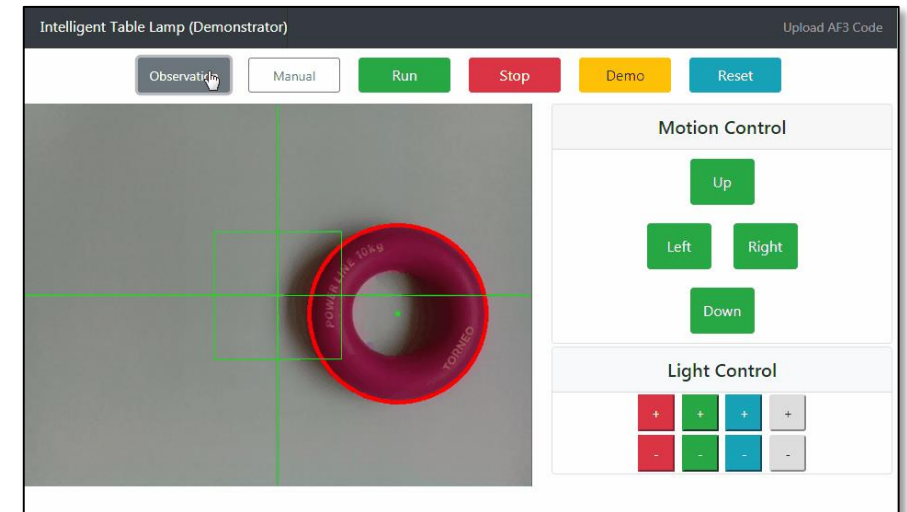
Behavior of “Motor Processing” defined by code snippet:

```

if (start_signal == true)
    {if ((openCV_cxy.x > 370 || openCV_cxy.x < 270) || (openCV_cxy.y > 290 || openCV_cxy.y < 190))
        {roll = (-1 * (openCV_cxy.x - 320)) / 16;
         nick = (openCV_cxy.y - 240) / 16;}}
else {nick = 0;
      roll = 0;}
    
```



## Augmented Image of User Interface

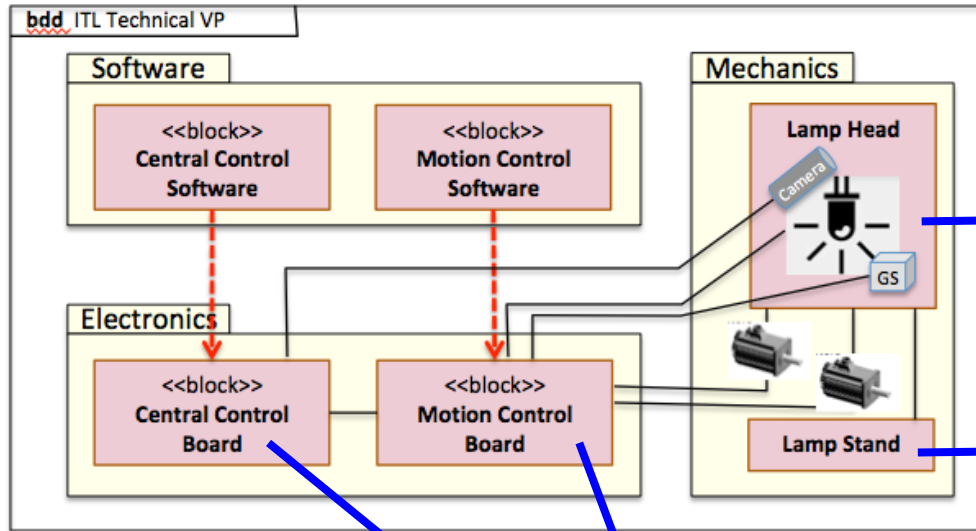


GPP Web Interface of ITL

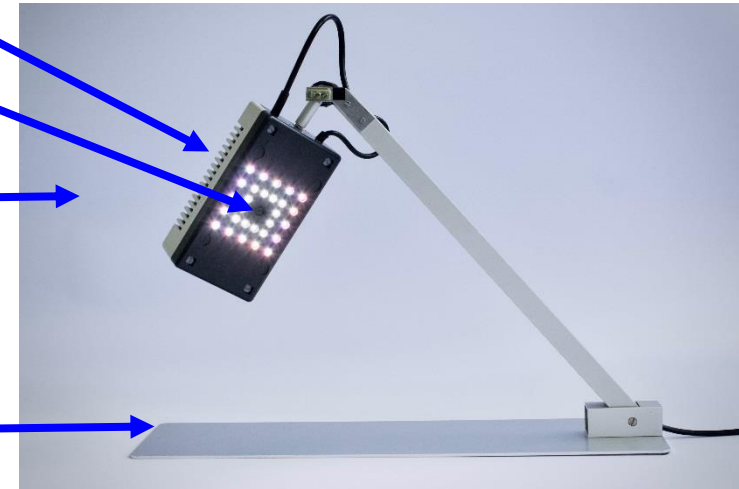


# Implementierung

Technical View

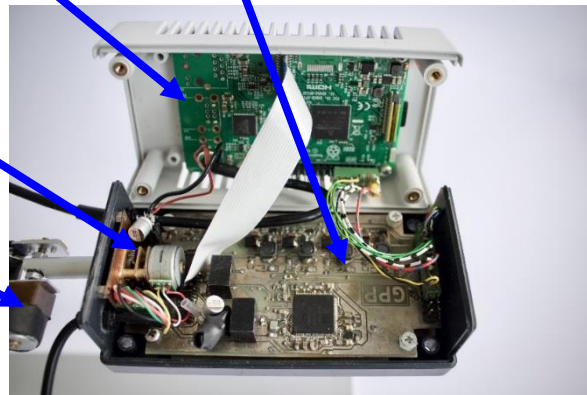


LEDs  
Camera



Motor for „Roll“

Motor for „Nick“



Mechanisches Design der Lampe

- Tischständer
- Lampenkopf mit LEDs und Kamera
- Drehung um Nick- und Rollachse

# Die praktische Anwendung - Lessons Learned



1. Es gibt vielfältige Modellierungs-Möglichkeiten  
→ der Start mit wenigen, ausgewählten Methoden hilft Anfangshürden zu überwinden
2. Entwickler wollen die Vorteile von MBSE unmittelbar verstehen  
→ z.B. einfache Dokumentation und Kommunikation selbst erfahren
3. Viele Methoden sind heute bereits in der Industrie üblich (z.B. Use Case, MSC)  
→ daran lässt sich anknüpfen  
→ zu achten ist auf die Durchgängigkeit der Methoden
4. Industrie-Trainingsbedarf kommt mit engen Zeitauflagen (z.B. 3 Tage SYSML + Methoden)  
→ Es gibt Bedarf für ein schnelles Erlernen von Sprache, Tools und Methoden



# Die “Intelligent Table Lamp”



*... modelliert, entwickelt und gebaut von:*



GPP Communication GmbH & Co. KG  
Kolpingring 18  
82041 Oberhaching

contact: [h.lagger@gppag.de](mailto:h.lagger@gppag.de)



*Wir freuen uns auf Ihren Besuch an unseren Stand !*

