



PREEvision: Beyond Requirements Engineering

REConf 2025 | Munich | May 05 – 07, 2025 | Dr.-Ing. Nico Adler, Dr. Andreas Reichenbacher

Agenda

► Introduction

Cyber-physical Systems are a Challenge

Regulations and Standards to Consider

PREEvision: Beyond Requirements Engineering

Summary

Vector: Facts and Figures

Our Mission:

Vector provides **OEMs and suppliers of automotive and related industries** with a professional and open development platform of **tools, software components and services** for creating **embedded systems**.



Employees:
> 4,500
Vectorians



Turnover:
1.01 bn €
in 2024



Associations:
participation in
16 committees

Customers:
> 9,000 companies
in 76 countries



Subsidiaries:
33 locations in
14 countries



Affiliated Companies:
CSM | EYYES | GiN

Development partner
for over 35 years



Our Portfolio: Application Areas and Product Examples

Development of Distributed Systems
PREEvision

← Focus of this Presentation

Embedded Software and Systems

MICROSAR, CANbedded, DaVinci, Customer Projects on SDV / Vehicle OS

Testing

CANoe, CANalyzer, vTESTstudio, HIL systems, CI/CT solution, VectorCAST

Diagnostics & Flashing

CANdelaStudio, Indigo, vFlash, CANoe.DiVa, vService

ECU Calibration

CANape, Calibration Data Management vCDM, VX1000

Measurement Technology

vMeasure, vSignalizer, vMDM, Logger, Analog Measurement Devices

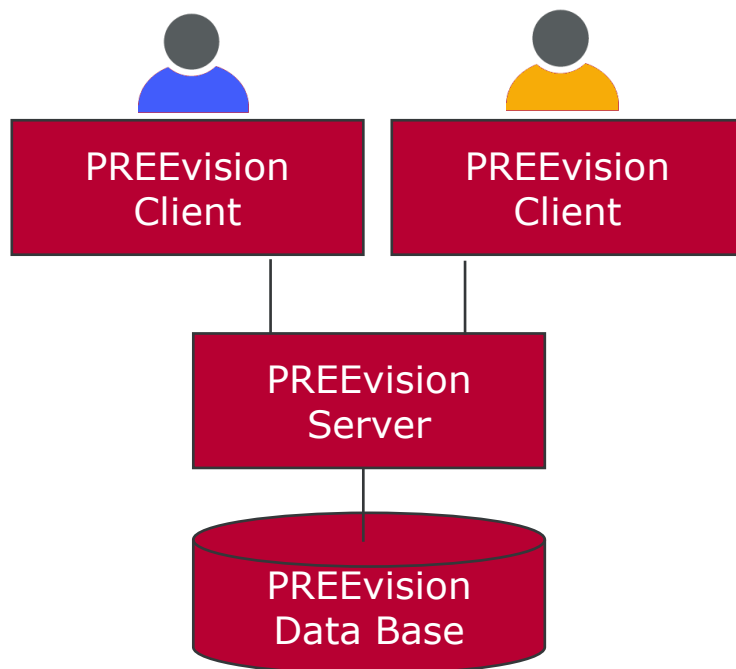
Consulting

Consulting Services, Engineering Services



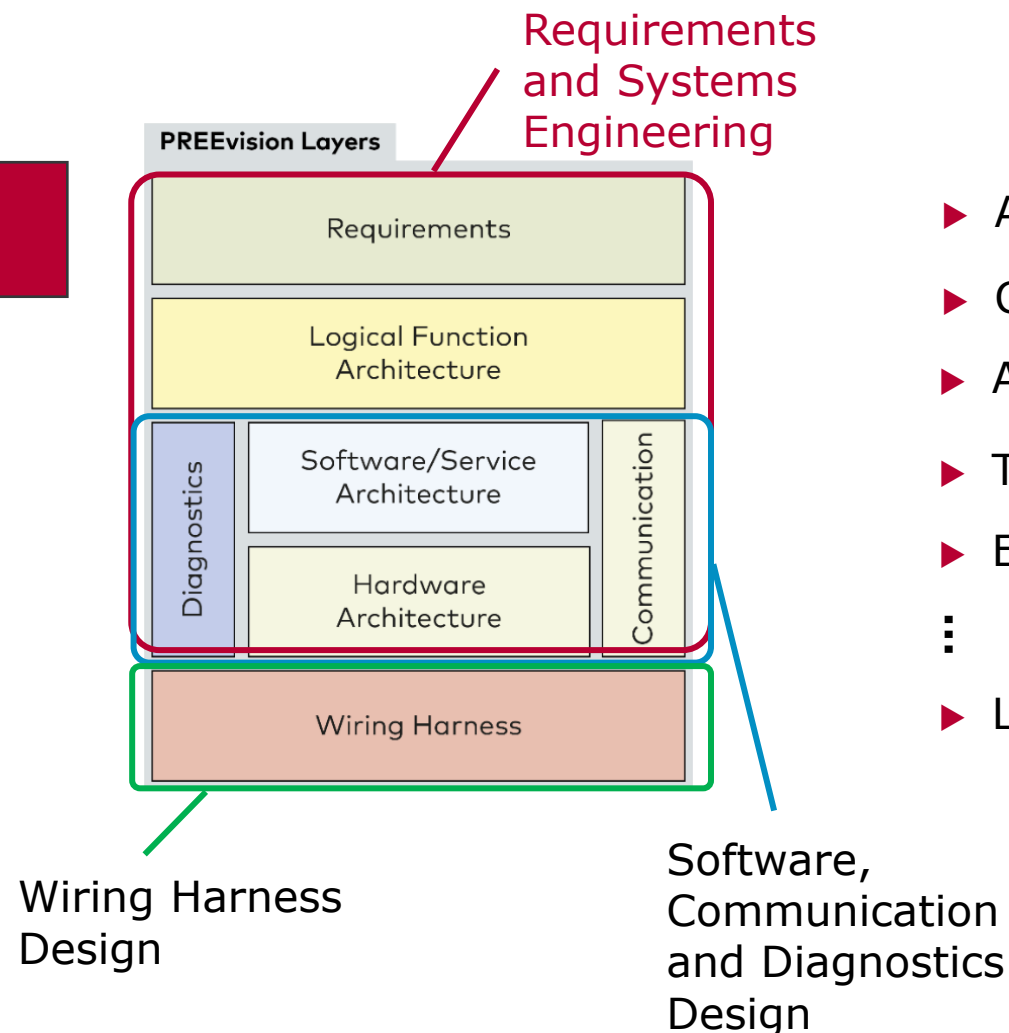
PREEvision: Use Cases and References

Product



Multi-User
Collaboration
Platform

Use Cases



References

- ▶ Automotive OEMs
- ▶ Commercial Vehicle OEMs
- ▶ Agriculture Machinery Providers
- ▶ Two Wheeler Manufacturers
- ▶ Engine Manufacturers
- ⋮
- ▶ Lithography System Providers

Agenda

Introduction

► **Cyber-physical Systems are a Challenge**

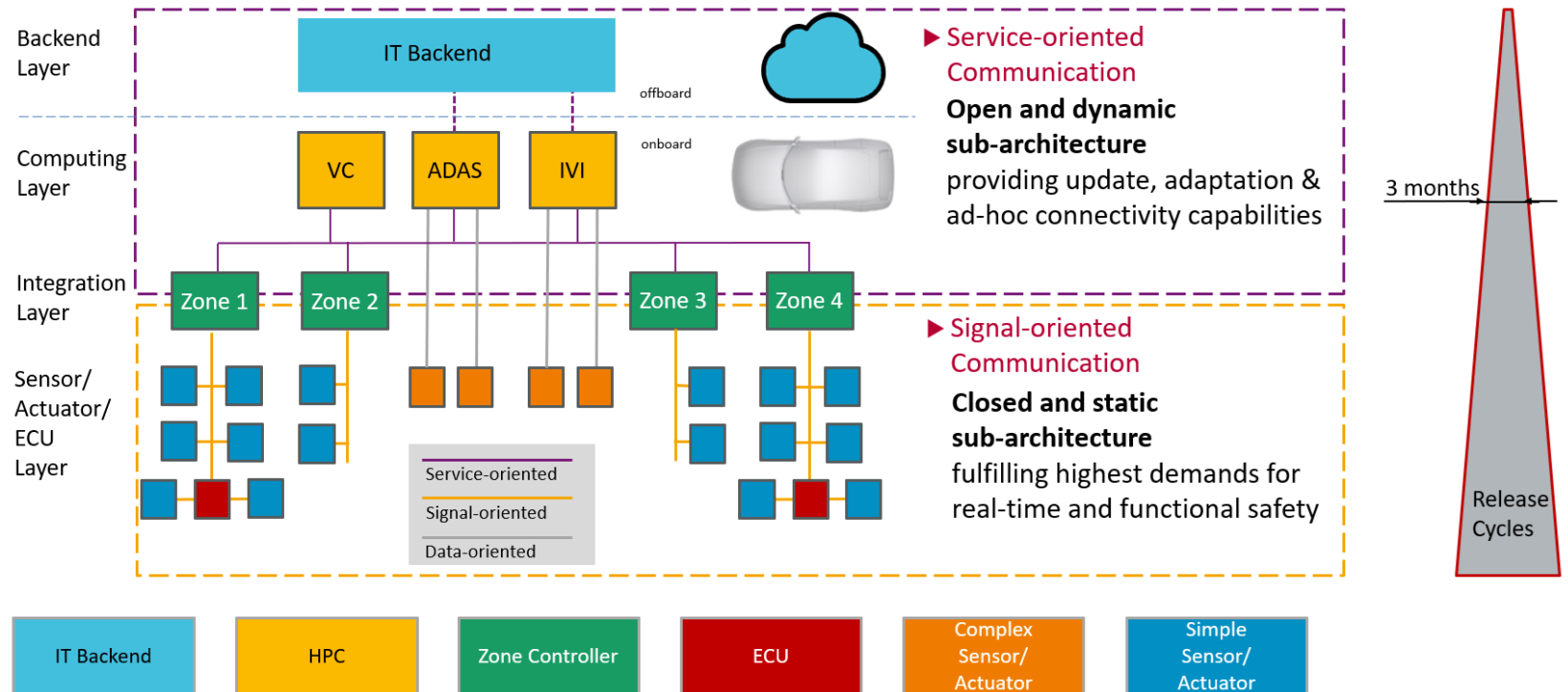
Regulations and Standards to Consider

PREEvision: Beyond Requirements Engineering

Summary

What Is a System in the Context of Automotive E/E Engineering?

E/E Architectures of Software-defined Vehicles (SDVs) are centralized and hierarchical

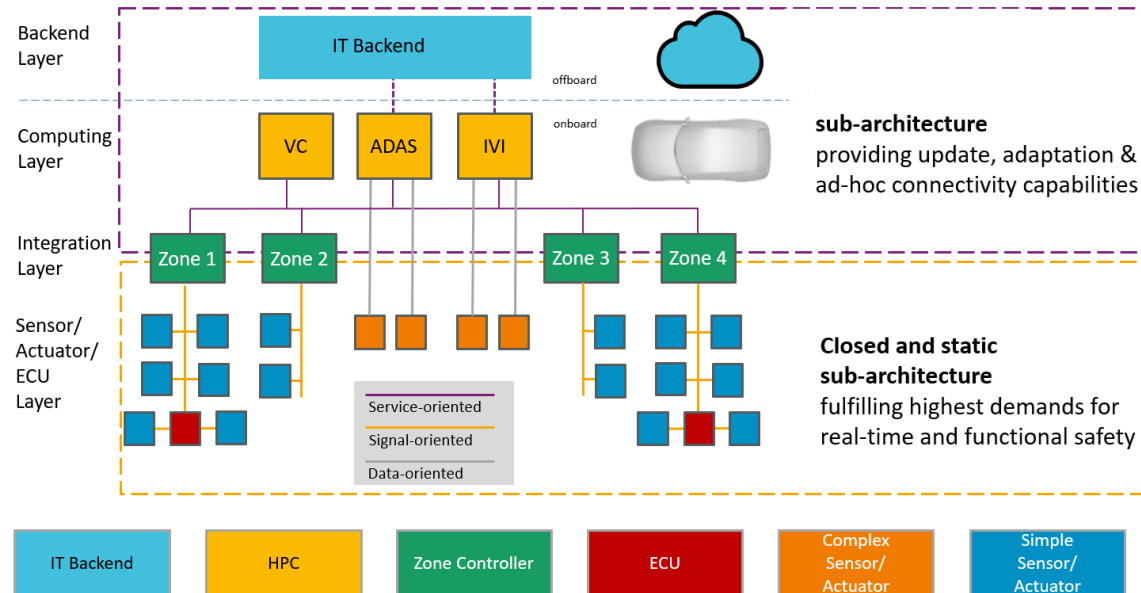


- ▶ SDVs are cyber-physical systems
- ▶ ... with 2 sub-architectures
- ▶ ... enable over the air SW updates and upgrades beyond Start of Production
- ▶ SW is never done!
- ▶ We see different SW update cycles on the different system layers.

→ A system of systems: Systems Engineering must manage the entire network

- from the IT Backend to the High-Performance Computers, zone controllers, ECUs, sensors and actuators
- from power supplies to wiring harness
- including the requirements, test specifications and traceability needed for functional safety reasons
- and the architecture describing the complete system.

Why Model-based Systems Engineering (MBSE) instead of Working With Documents?



- ▶ SDVs have a disruptive product architecture
- ▶ ... with an increasing diversity of system elements
- ▶ ... and an increasing system complexity.

! MBSE is the right approach

- ✓ to deal with „Systems of Systems“ and to oversee the complexity
- ✓ to enable a clear picture of relations, dependencies and life cycles of the system elements
- ✓ for traceability
- ✓ essential to manage the always changing stakeholder requirements and to determine their impacts
- ✓ necessary to meet functional and technical safety requirements
- ✓ A fine granular data model enables a fully transparent single point of truth, while documents are text-based black boxes

Agenda

Introduction

Cyber-physical Systems are a Challenge

► **Regulations and Standards to Consider**

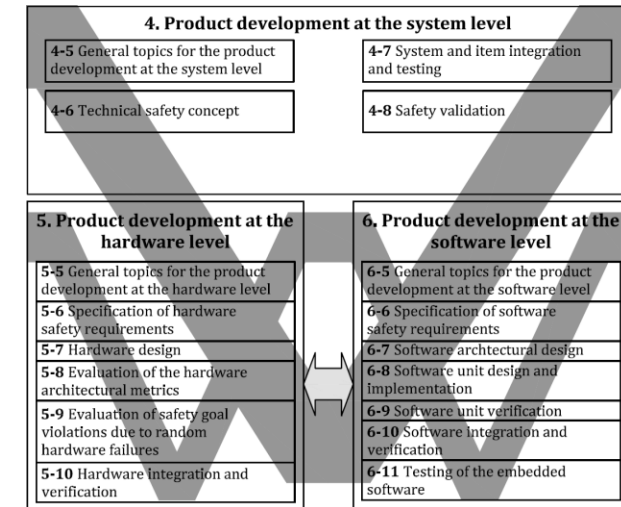
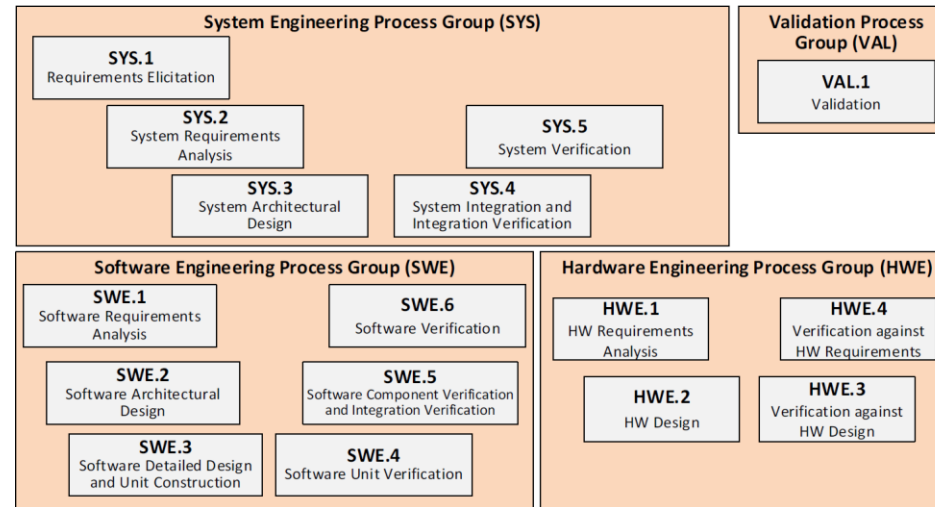
PREEvision: Beyond Requirements Engineering

Summary

Model-Based Systems Engineering Framework

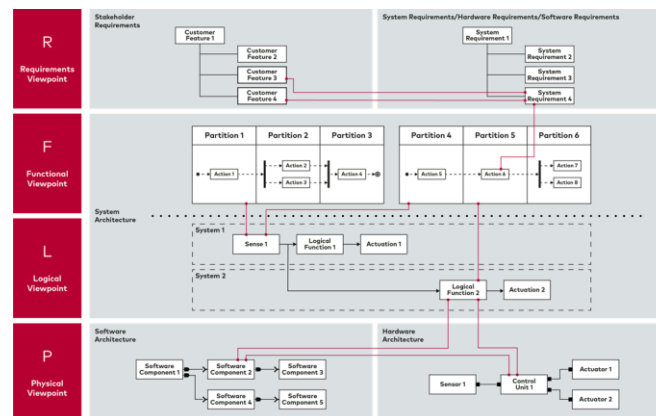
Regulations and Standards in Automotive

- ▶ **ASPICE**
- ▶ **ISO 26262**



Methodologies

- ▶ **RFLP**

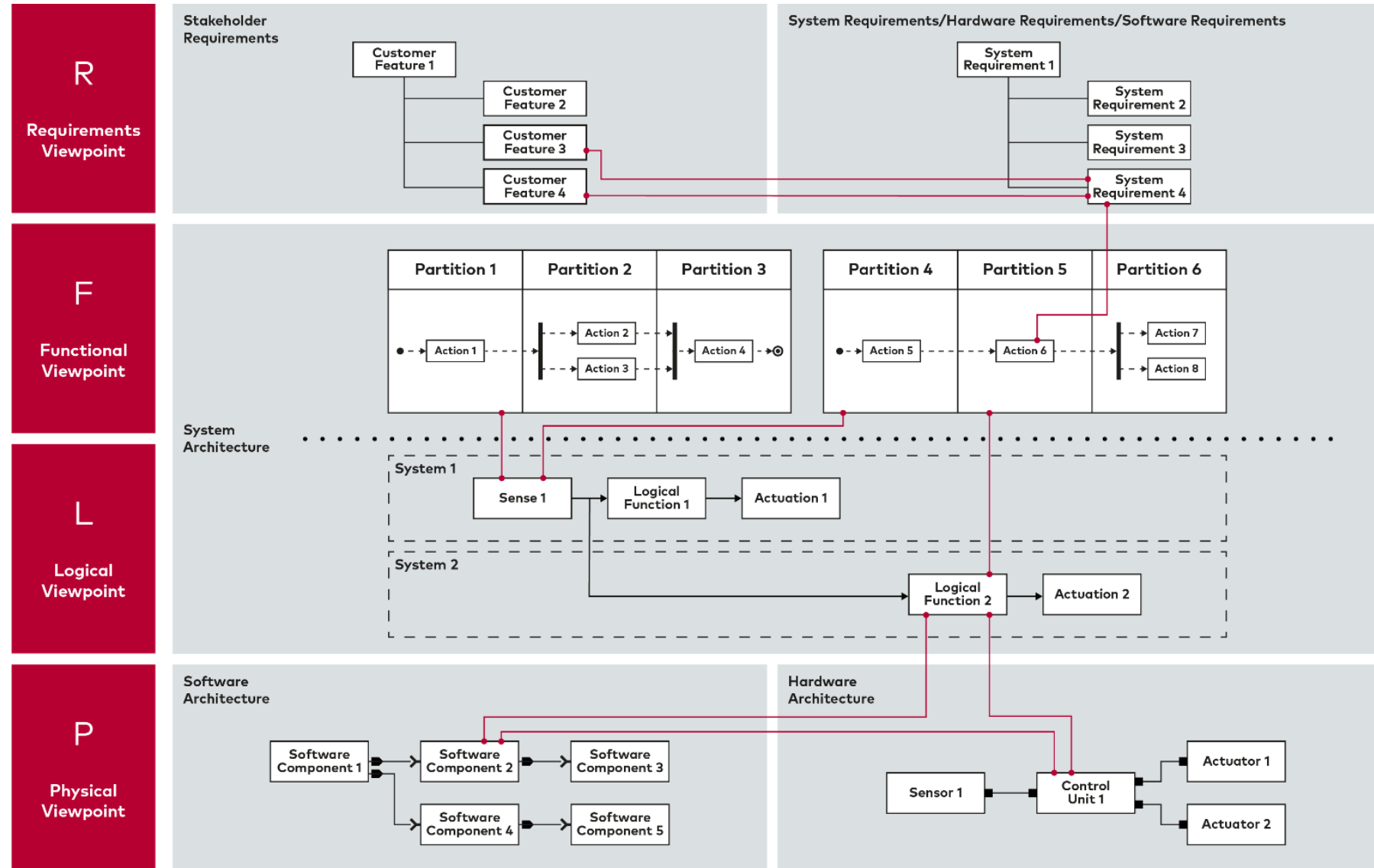


Modeling Languages

- ▶ **SysML**



The RFLP Methodology



Stakeholder and System Requirements

Isolated Perspective for every System

- ▶ from sensors to actuators
- ▶ from inputs to outputs

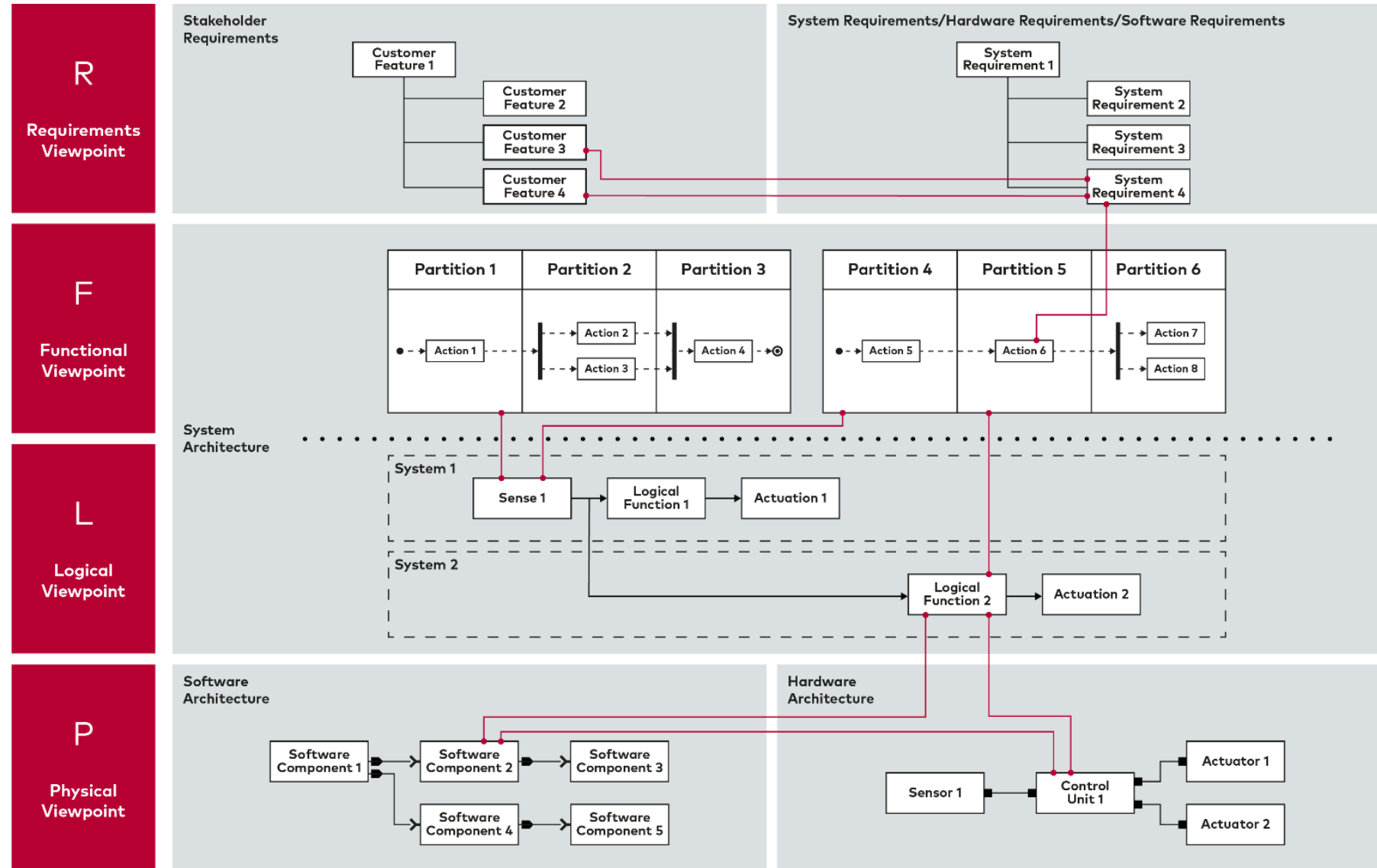
Integrated Perspective of all Systems

- ▶ shared sensors
- ▶ shared actuators
- ▶ System interfaces

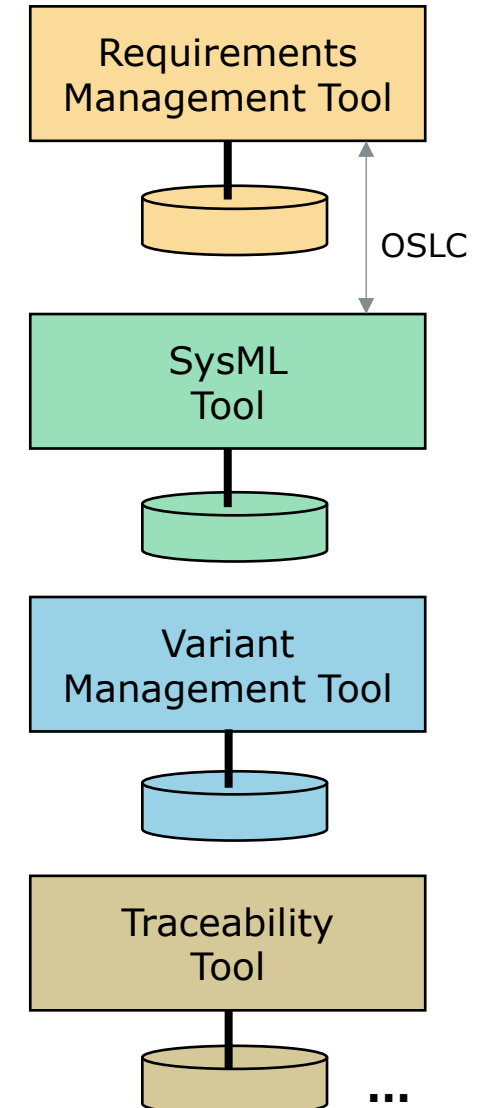
Physical Architecture

- ▶ Hardware
- ▶ Software

Typical Tool Support



Typical Tool Support



Agenda

Introduction

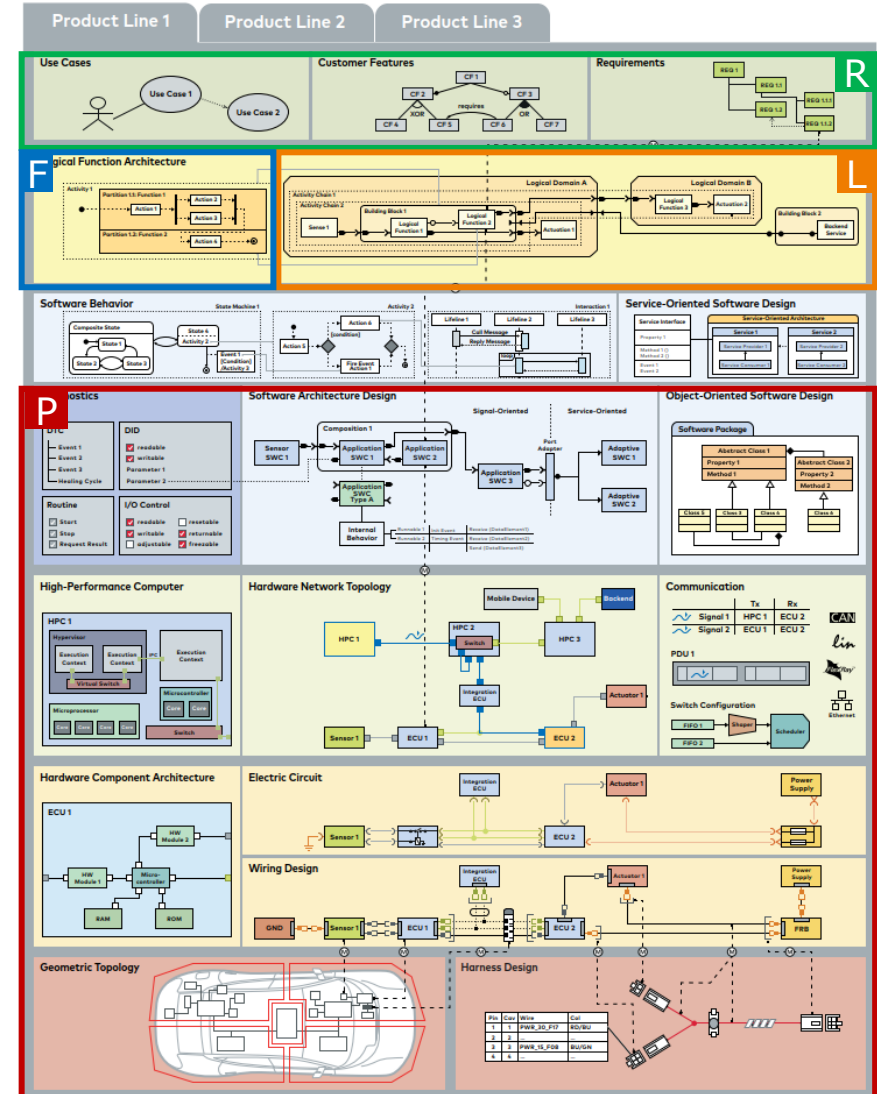
Cyber-physical Systems are a Challenge

Regulations and Standards to Consider

► **PREEvision: Beyond Requirements Engineering**

Summary

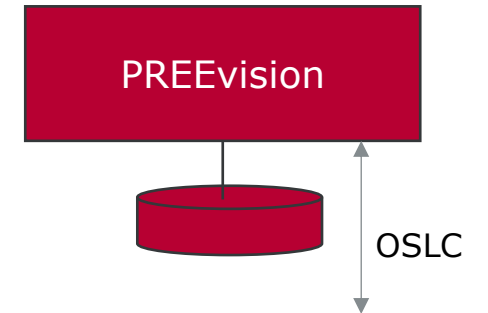
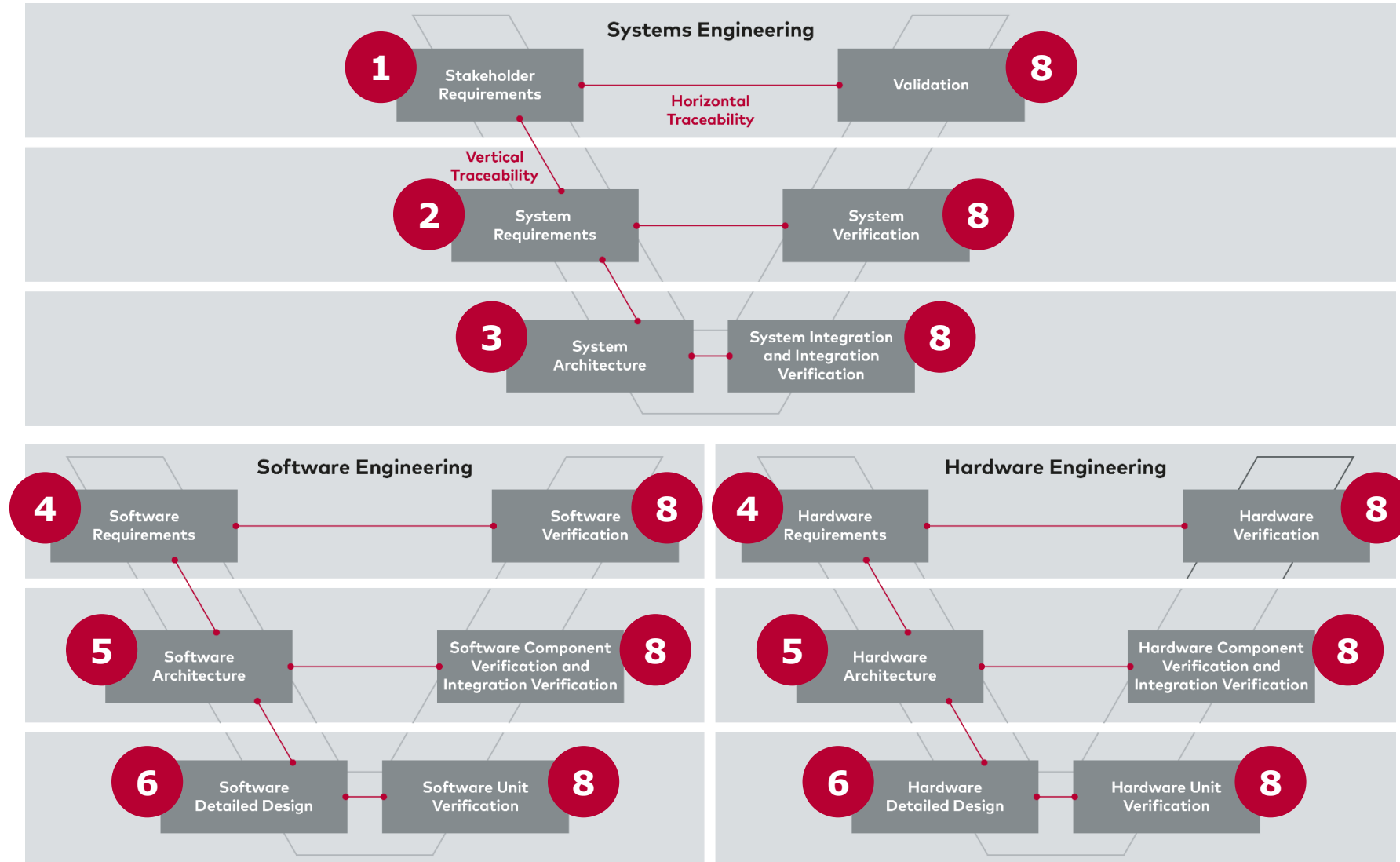
PREEvision supports RFLP completely



Main Steps to Perform MBSE

- 1 Elicitation of stakeholder requirements**
- 2 Analysis of the “problem area” and specification of system requirements**
- 3 Design of the system architecture: Derive functionality, which must be fulfilled from the system and specify the system architecture**
- 4 Specification of hardware and software requirements**
- 5 Decomposition of the system and specification of the physical hardware and software architecture**
- 6 Detailed design of software and hardware**
- 7 Implementation of software and hardware**
- 8 Verification, integration, validation ... and delivery**

Processes according to ASPICE and ISO 26262

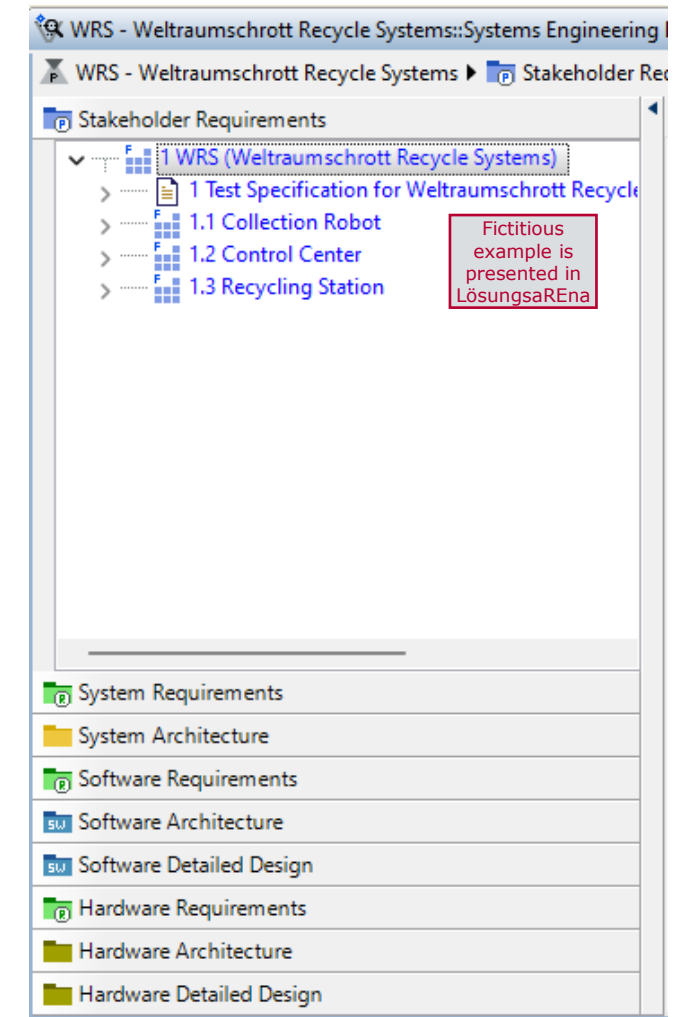
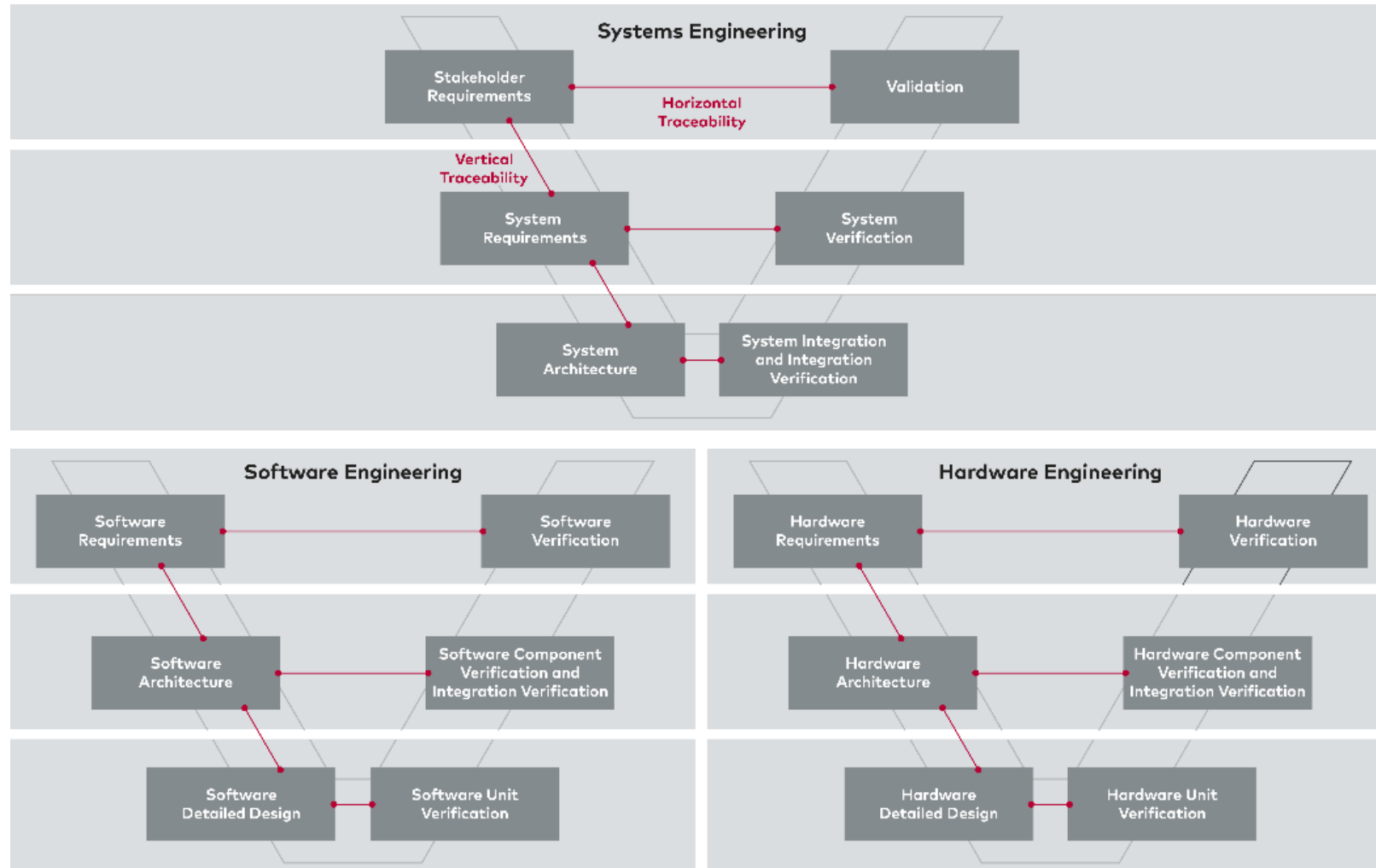


- PREEvision supports all processes except implementation steps (Step 7)

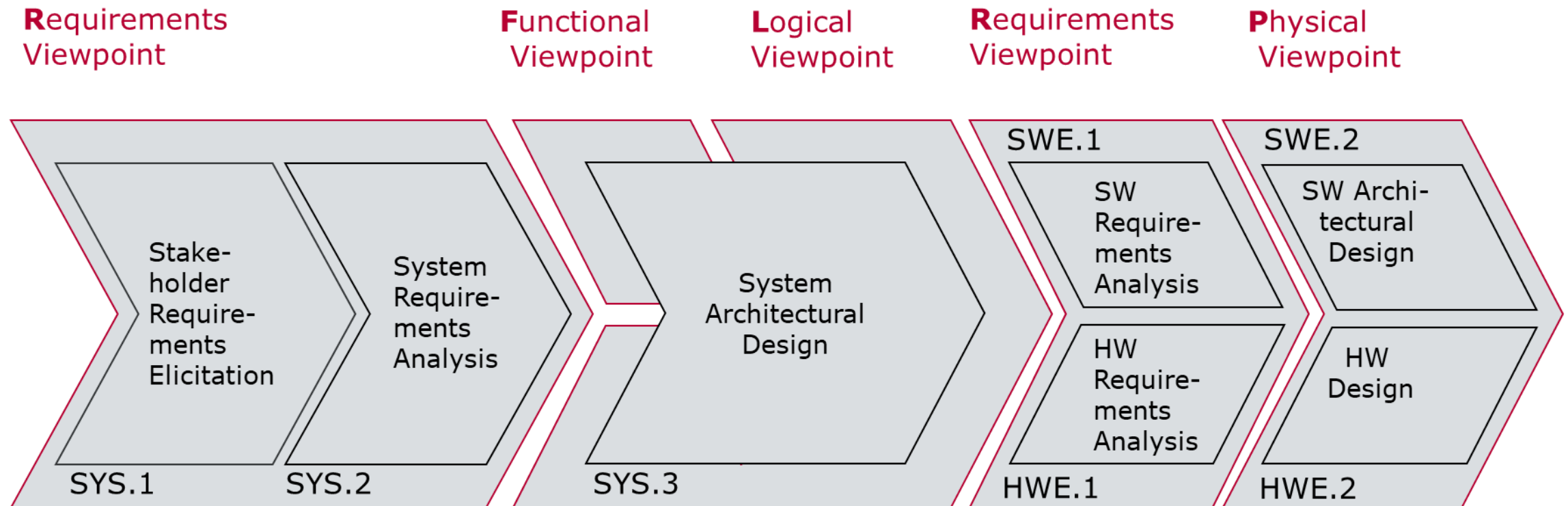
7

We Provide a Solution ready to Use

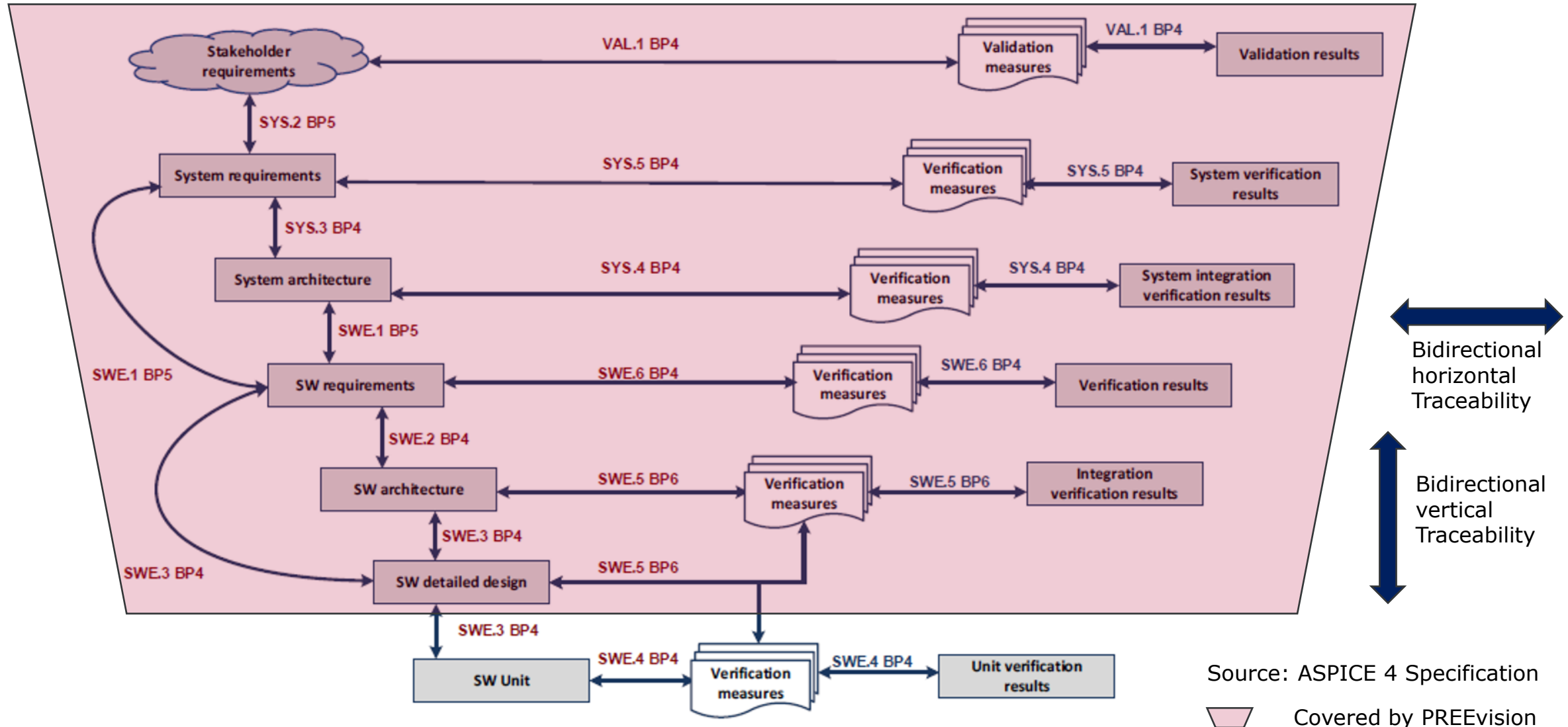
► ... with a User Interface for these Processes



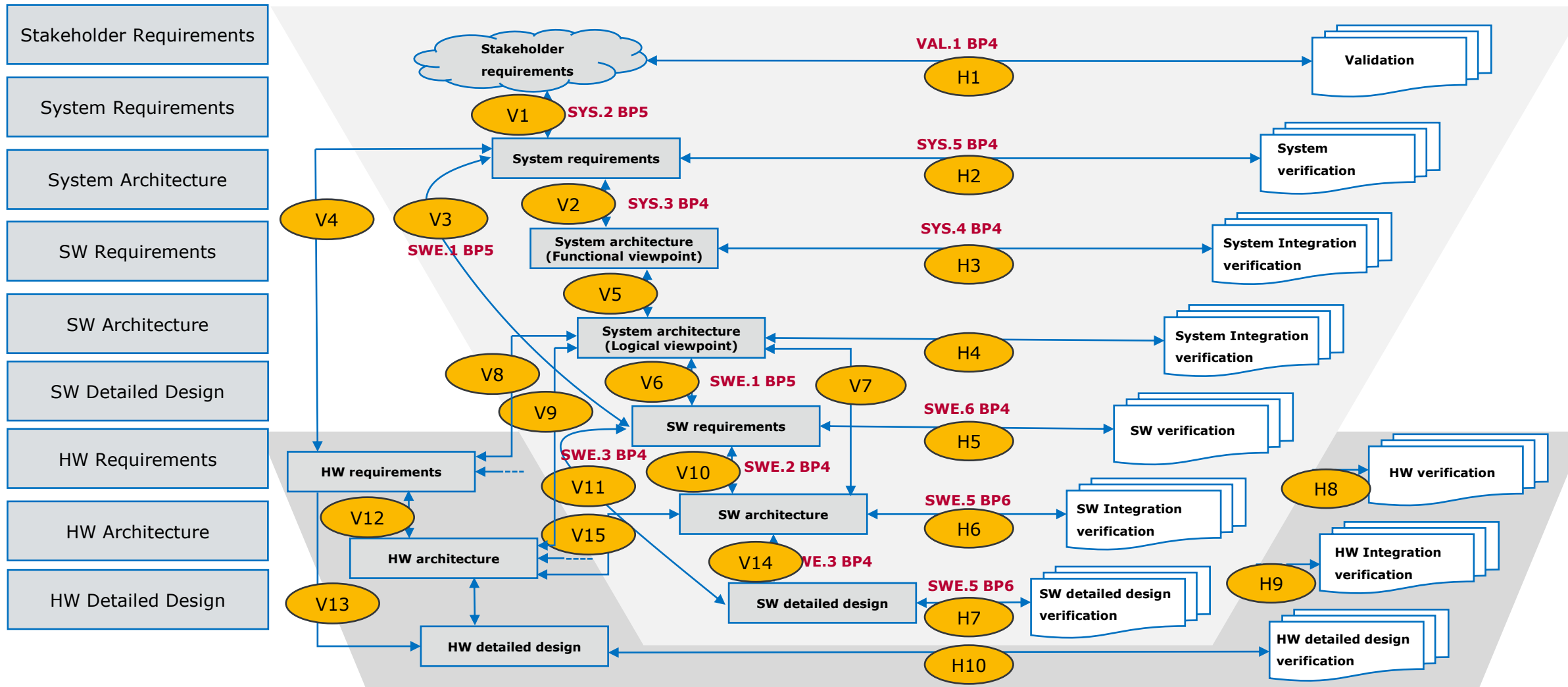
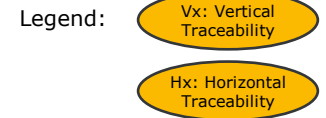
ASPICE and RFLP Are No Contradiction



Bidirectional Traceability for System and SW Processes



Traceability Overview



The Requirements Viewpoint is not limited to Text based Editors

- The Requirements Viewpoint is supported by text based and diagram-based editors

Relevant Explorer Categories:

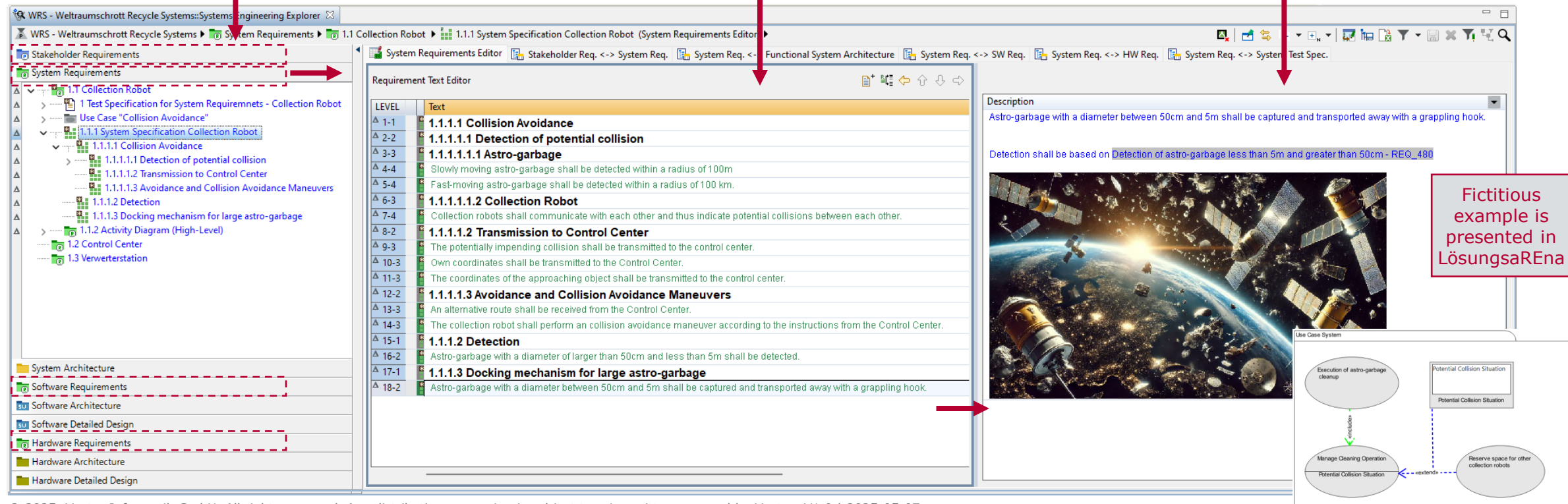
- Stakeholder Requirements
- System / Software / Hardware Requirements

Requirements Editor for ...

- System Requirements
- ... is selected and shows a compact representation of the system specification

Detailed description of the system requirement with ...

- formatted text, inserted graphic and text placeholders



The screenshot displays the WRS - Weltraumschrott Recycle Systems Systems Engineering Explorer. The left sidebar shows a tree view of requirements, with '1.1.1.1.1.1 Collision Avoidance' selected. The central pane shows the 'Requirement Text Editor' with a list of requirements and their descriptions. The right pane shows a detailed description of a requirement, including formatted text, a graphic of a satellite, and a UML diagram.

Requirement Text Editor

LEVEL	Text
1-1	1.1.1.1 Collision Avoidance
2-2	1.1.1.1.1 Detection of potential collision
3-3	1.1.1.1.1.1 Astro-garbage
4-4	Slowly moving astro-garbage shall be detected within a radius of 100m
5-4	Fast-moving astro-garbage shall be detected within a radius of 100 km.
6-3	1.1.1.1.1.2 Collection Robot
7-4	Collection robots shall communicate with each other and thus indicate potential collisions between each other.
8-2	1.1.1.1.2 Transmission to Control Center
9-3	The potentially impending collision shall be transmitted to the control center.
10-3	Own coordinates shall be transmitted to the Control Center.
11-3	The coordinates of the approaching object shall be transmitted to the control center.
12-2	1.1.1.1.3 Avoidance and Collision Avoidance Maneuvers
13-3	An alternative route shall be received from the Control Center.
14-3	The collection robot shall perform an collision avoidance maneuver according to the instructions from the Control Center.
15-1	1.1.1.2 Detection
16-2	Astro-garbage with a diameter of larger than 50cm and less than 5m shall be detected.
17-1	1.1.1.3 Docking mechanism for large astro-garbage
18-2	Astro-garbage with a diameter between 50cm and 5m shall be captured and transported away with a grappling hook.

Description

Astro-garbage with a diameter between 50cm and 5m shall be captured and transported away with a grappling hook.

Detection shall be based on [Detection of astro-garbage less than 5m and greater than 50cm - REQ_480](#)

UML Diagram

```
graph TD
    A[Execution of astro-garbage cleanup] --> B[Manage Cleaning Operation]
    B --> C[Potential Collision Situation]
    C --> D[Reserve space for other collection robots]
    D --> E[Potential Collision Situation]
    E --> C
```

Fictitious example is presented in LösungsREna

Test Specification Editor

Define Test Specification for e.g. System Requirements using ...

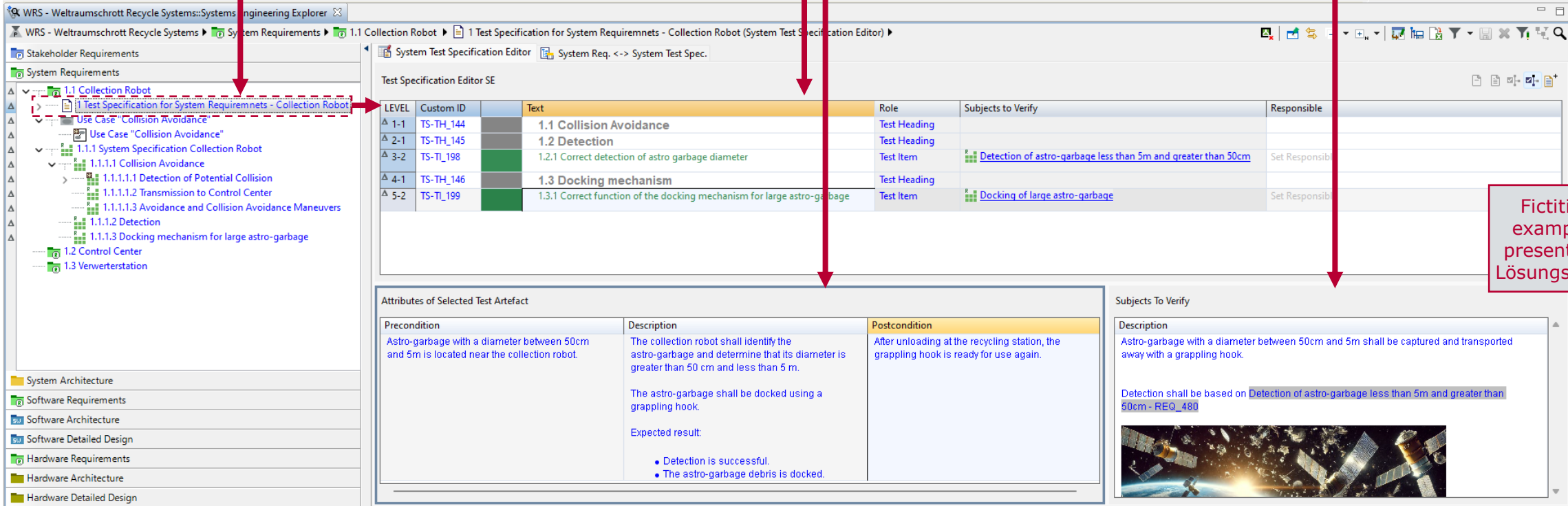
- Artifact Under Test

Test Specification:

- Specify the Test Items with Precondition / Description / Postcondition

Get a detailed overview about the Subjects to Verify. In this case for the ...

- System Requirement



The screenshot shows the 'Test Specification Editor' interface. On the left is a tree view of the system requirements. The main area displays a table of test items. Below this, there are two panels: 'Attributes of Selected Test Artifact' and 'Subjects To Verify'.

LEVEL	Custom ID	Text	Role	Subjects to Verify	Responsible
Δ 1-1	TS-TH_144	1.1 Collision Avoidance	Test Heading		
Δ 2-1	TS-TH_145	1.2 Detection	Test Heading		
Δ 3-2	TS-TI_198	1.2.1 Correct detection of astro garbage diameter	Test Item	Detection of astro-garbage less than 5m and greater than 50cm	Set Responsib
Δ 4-1	TS-TH_146	1.3 Docking mechanism	Test Heading		
Δ 5-2	TS-TI_199	1.3.1 Correct function of the docking mechanism for large astro-garbage	Test Item	Docking of large astro-garbage	Set Responsib

Precondition	Description	Postcondition
Astro-garbage with a diameter between 50cm and 5m is located near the collection robot.	The collection robot shall identify the astro-garbage and determine that its diameter is greater than 50 cm and less than 5 m. The astro-garbage shall be docked using a grappeling hook. Expected result: • Detection is successful. • The astro-garbage debris is docked.	After unloading at the recycling station, the grappeling hook is ready for use again.

Subjects To Verify
Description Astro-garbage with a diameter between 50cm and 5m shall be captured and transported away with a grappeling hook. Detection shall be based on Detection of astro-garbage less than 5m and greater than 50cm - REQ_480 

Fictitious example is presented in Lösungsarena

Vertical Traceability Editor

Explorer Categories:

- Seamless from "Stakeholder Requirements" across "System Architecture" to "Software/Hardware Detailed Design"

Traceability Editor for ...

- "Stakeholder Requirements $\leftrightarrow$ System Requirements"
- ... is selected and shows vertical traceability

The user gets direct feedback through highlighting in the "Traceability Editors", e.g. missing traceability

The screenshot displays the Vertical Traceability Editor interface. On the left, the 'Stakeholder Requirements' pane shows a tree structure with '1.1 Collection Robot' selected. The main area is divided into two panes: 'Stakeholder Requirements' and 'System Requirements'. The 'Stakeholder Requirements' pane shows a table with 'Customer Feature' and 'Linked System Requirements'. The 'System Requirements' pane shows a table with 'System Requirement' and 'Linked Stakeholder Requirement'. A color-coded legend on the right indicates the status of the links: 'No link expected' (grey), 'Link missing' (yellow), 'Ambiguous link' (blue), and 'Unexpected link present' (red). A red box highlights a specific example in the 'System Requirements' pane, labeled 'Fictitious example is presented in LösungsREna'.

Color Meaning:

- No link expected
- Link missing
- Ambiguous link
- Unexpected link present

Fictitious example is presented in LösungsREna

Horizontal Traceability Editor

Explorer Categories:

- ▶ Seamless from "Stakeholder Requirements" across "System Architecture" to "Software/Hardware Detailed Design"

Traceability Editor for ...

- ▶ "System Requirements $\leftrightarrow$ System Test Spec"
- ... is selected and shows horizontal traceability

The user gets direct feedback through highlighting in the "Traceability Editors", e.g. missing traceability

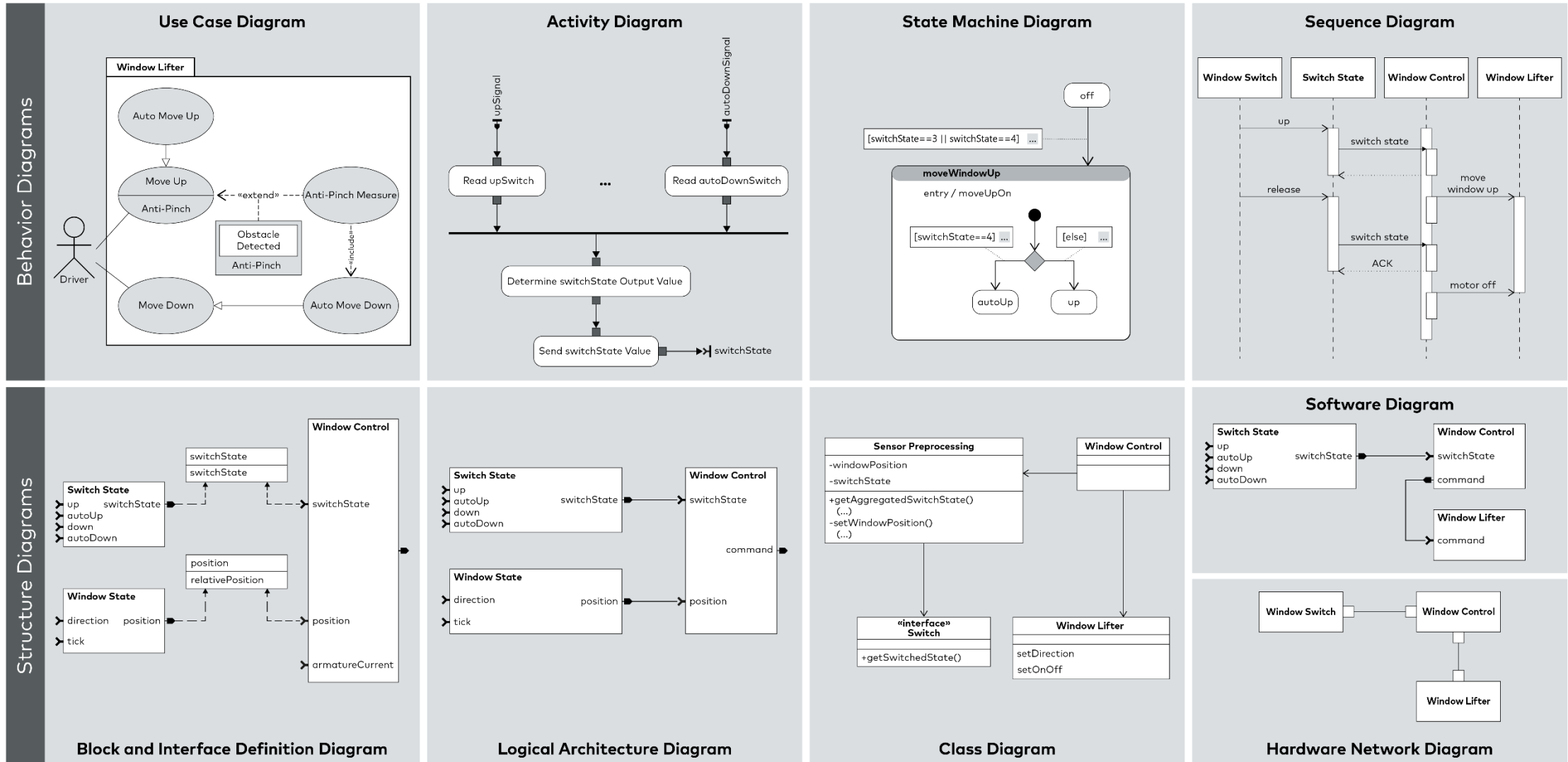
The screenshot displays the Vector PREEvision Horizontal Traceability Editor interface. The left sidebar shows the 'Explorer Categories' with a tree view of requirements. The main area is divided into two panes: 'System Requirements' and 'System Test Specification'. The 'System Requirements' pane shows a hierarchy of requirements, with '1.1 Collection Robot' selected. The 'System Test Specification' pane shows a hierarchy of test specifications, with '1.1 Collision Avoidance' selected. A 'Color Meaning' legend is visible on the right, indicating the status of links between requirements and test specifications.

Color Meaning:

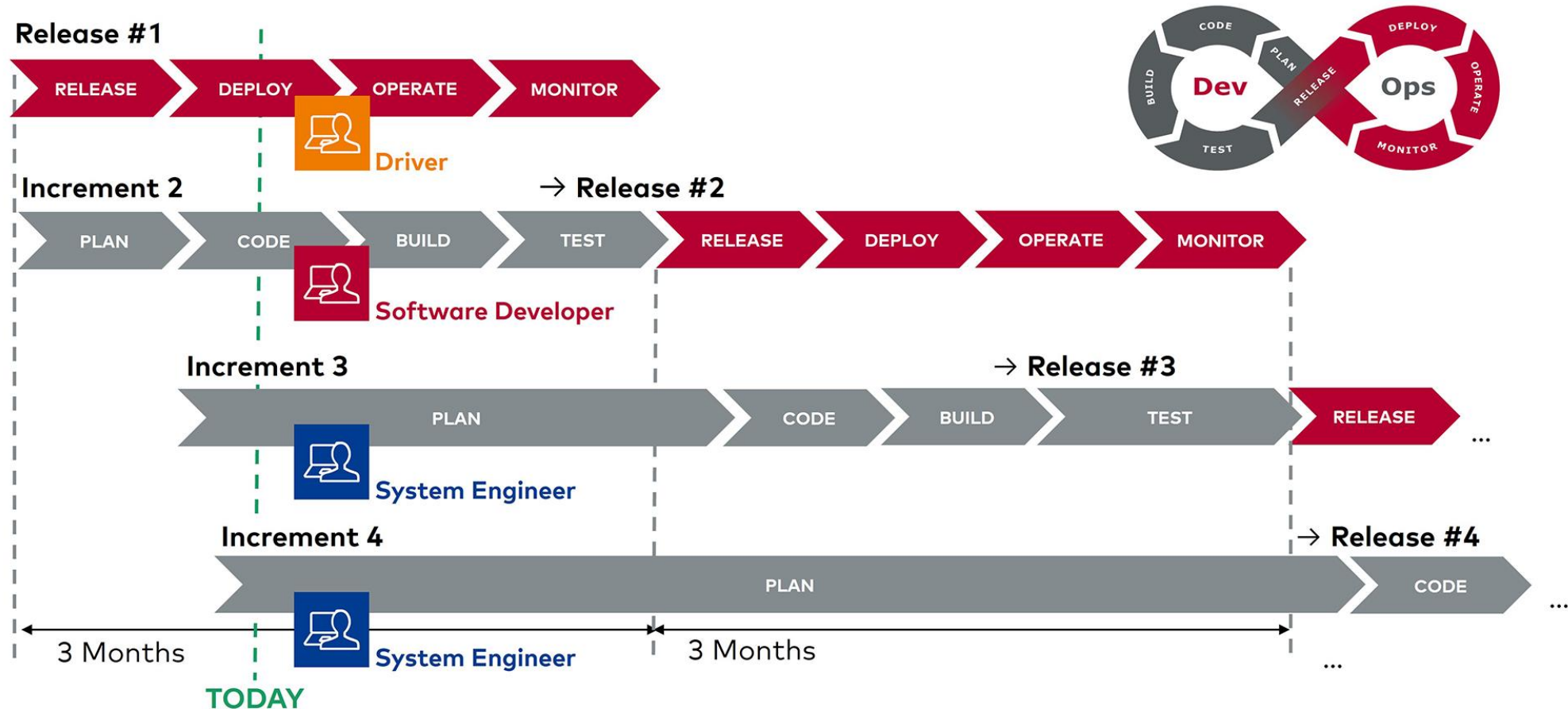
- No link expected
- Link missing
- Ambiguous link
- Unexpected link present

Fictitious example is presented in Lösungsarena

All needed SysML Concepts are available



Parallel and Agile Work on Different Releases is supported



Baselining, Release Branches and Development Pipelines are mandatory not only in Software Development, but also for Systems Engineering! PREEvision provides the needed functionality.

Agenda

Introduction

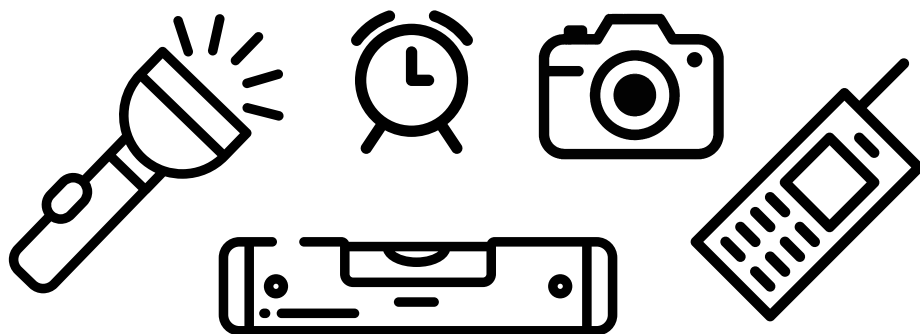
Cyber-physical Systems are a Challenge

Regulations and Standards to Consider

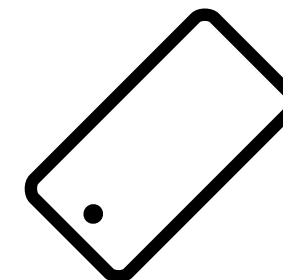
PREEvision: Beyond Requirements Engineering

► **Summary**

Tools – Integrated or Not?



- ▶ Cell Phone
- ▶ Camera
- ▶ Flashlight
- ▶ Alarm Clock
- ▶ Level
- ▶ ...



Requirements
Management Tool

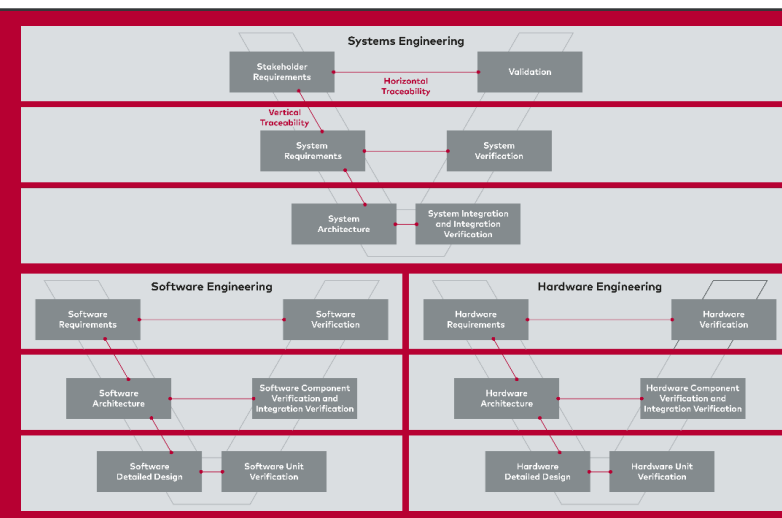
Variant
Management Tool

SysML
Tool

Traceability
Tool

“Best in Class”

Integrated Tool



“Best Convenience, Consistency and Transparency”

Your Questions are Welcome!

Author:
Dr.-Ing. Nico Adler, Dr. Andreas Reichenbacher
Vector Germany