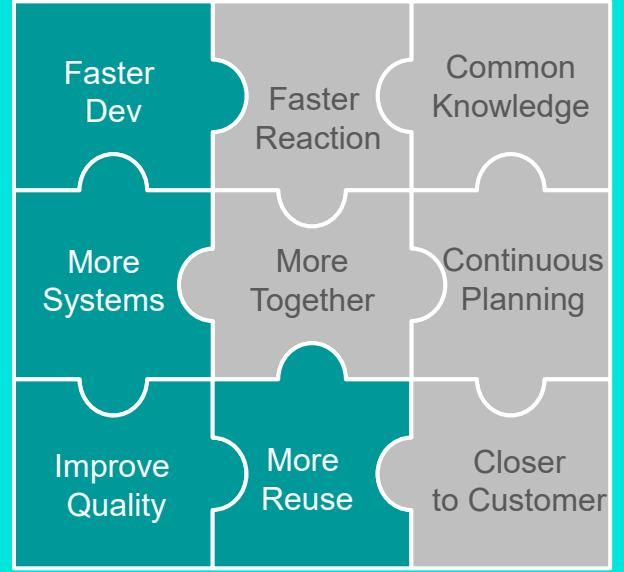


# Reuse of Requirements

Where to place potential reused requirements?



# What are we talking about?

The screenshot shows the PLM\_tool\_chain interface with the title bar "PLM\_tool\_chain" and "50 Requirements". The left sidebar includes a search bar, user profile "Steiner, Jochen (DI CS SD EH DS 3)", and navigation links: Home, RS Training RE @SI EP, Work Items, Documents & Pages, Default Space, 10 Project Overview, 20 External Requirements, 30 System Requirements (selected), 40 System Features, 50 Requirements (selected), 60 Features, and 80 Test. A mouse cursor is hovering over the "50 Requirements" link.

*When do we start?*

PM 100 ... QPM 200

*Where to place my requirement specifications?*

It depends ;-)

- 20 External Requirements
- 30 System Requirements
- 50 Requirements

# What are we talking about?

The screenshot shows the interface of the Polarion PLM tool chain. At the top, there is a header with a profile icon, a folder icon labeled 'PLM\_tool\_chain', and a text field showing '50 Requirements'. Below the header is a toolbar with various icons. The main area is a sidebar navigation menu on the left and a large workspace on the right. The sidebar includes links for 'Home', 'RS Training RE @SI EP', 'Work Items', 'Documents & Pages' (which is expanded to show 'Default Space', '10 Project Overview', '20 External Requirements', '30 System Requirements' (which is selected and highlighted in blue), '40 System Features', '50 Requirements', '60 Features', and '80 Test'). The workspace on the right is currently empty.

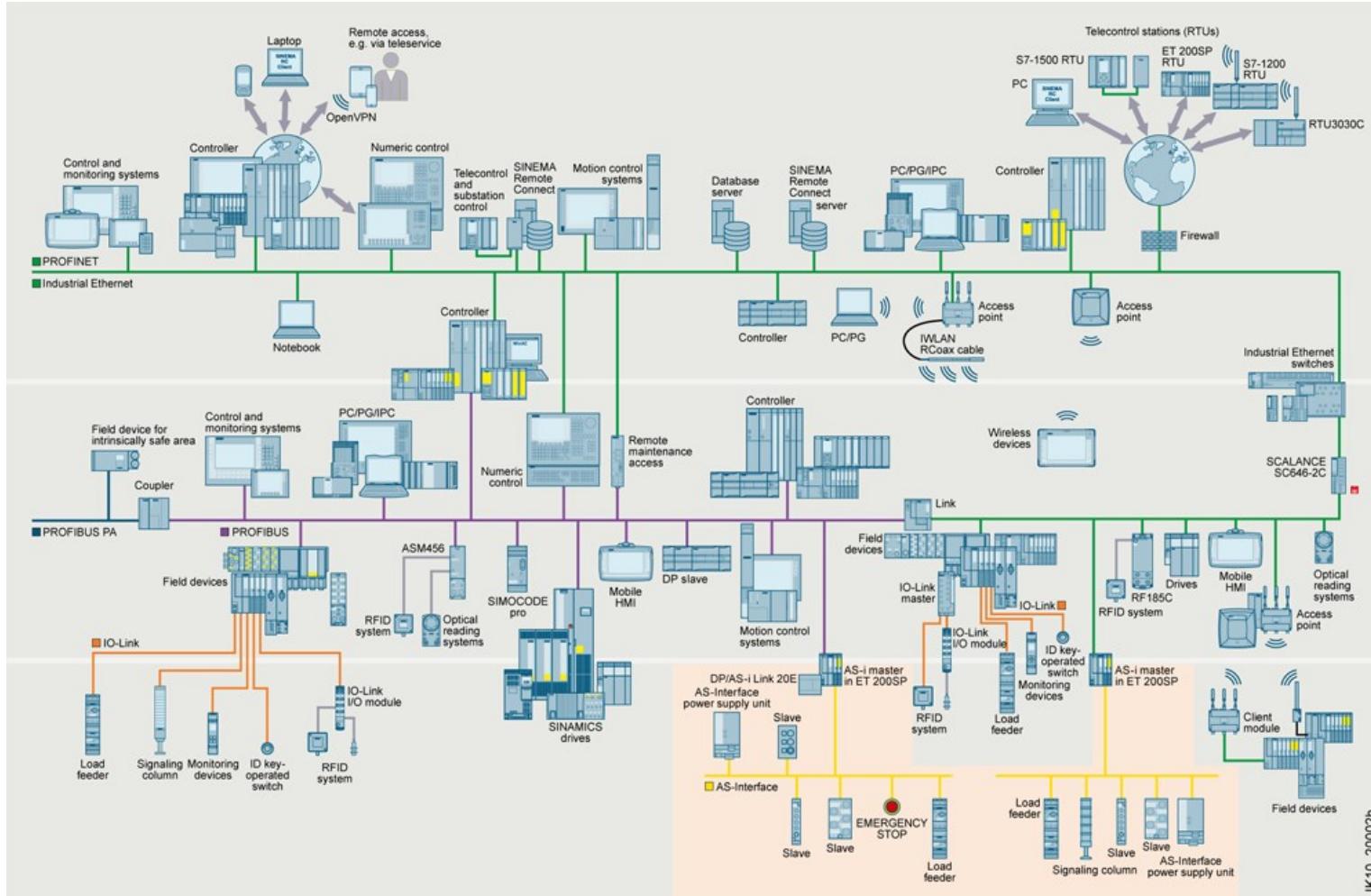
All requirements specifications that we expect below this folder:

30 System Requirements

Requirements that belong to

- a classical system, means overall behavior not component behavior

# Requirements for a classical system



A **system** refers to the complete set of components, processes, and interactions that are designed to achieve specific objectives or functions.

Normal requirement specification focuses on a single product or component.

**System Requirement Specification** looks at the system as a whole, defining the overall requirements and functionality of the entire system

# What are we talking about?

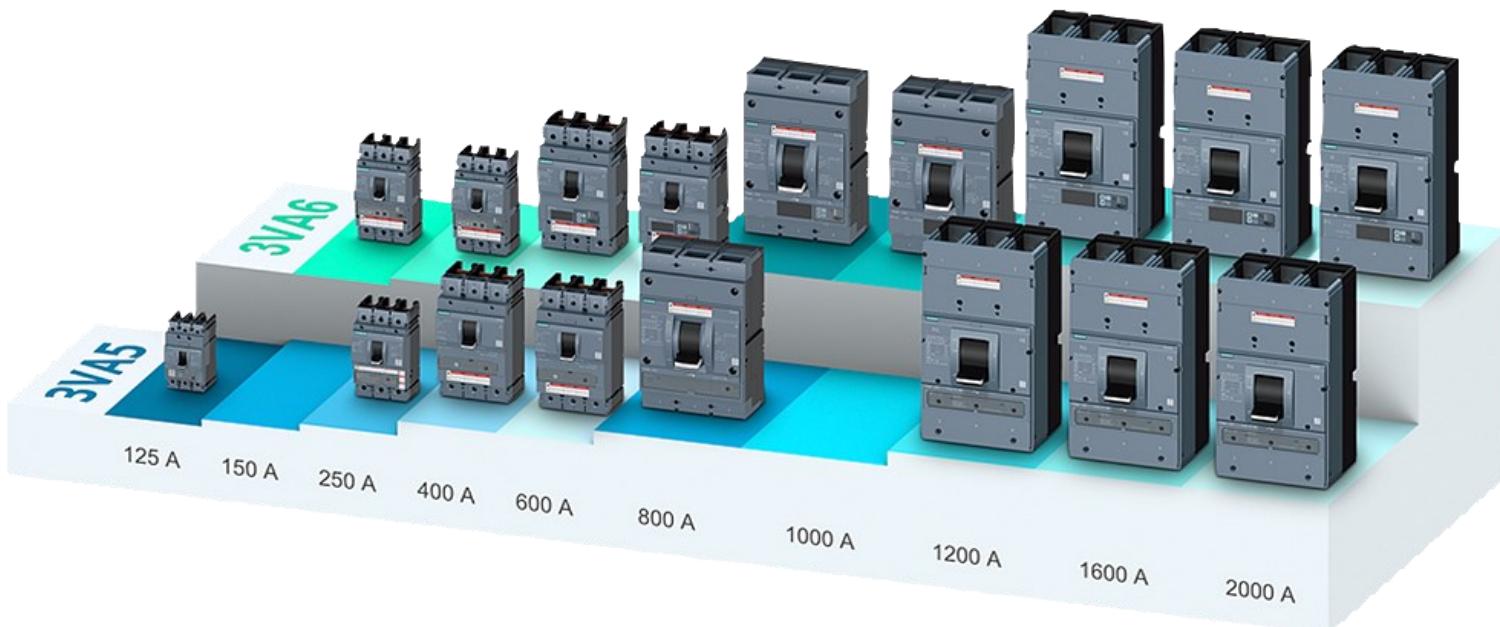
The screenshot shows the interface of the Polarion PLM tool chain. At the top, there is a header with a profile icon, a folder icon labeled 'PLM\_tool\_chain', and a text field showing '50 Requirements'. Below the header is a toolbar with various icons. The main area features a sidebar on the left with a user profile for 'Steiner, Jochen (DI CS SD EH DS 3)' and sections for 'My Polarion', 'Home', 'RS Training RE @SI EP', 'Work Items', 'Documents & Pages' (which is expanded to show 'Default Space', '10 Project Overview', '20 External Requirements', '30 System Requirements' which is selected and highlighted in blue, '40 System Features', '50 Requirements', '60 Features', and '80 Test'), and a search bar. To the right of the sidebar is a large, mostly empty workspace.

All requirements specifications that we expect below this folder:

Requirements that belong to

- a **classical system**,  
means overall behavior not component behavior
- a **product family or product line** with similar functions and behavior, so they can be reused in the product specific requirements specification

## Requirements for a product family or product line



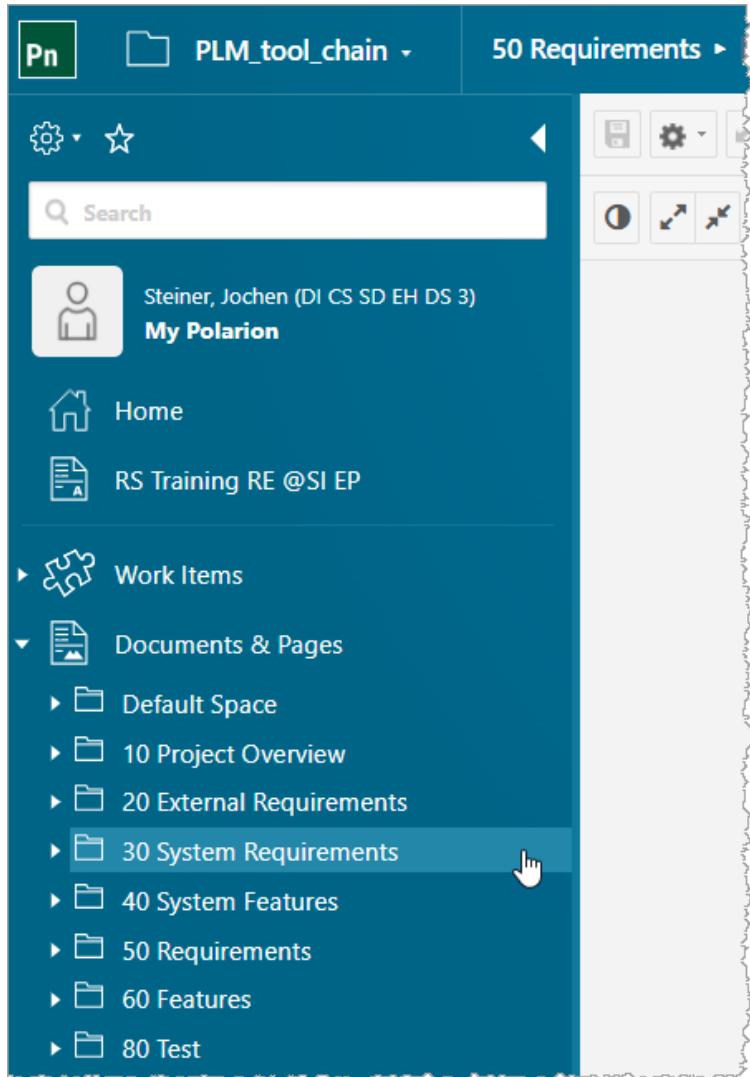
E.g.: Product line overview: 3VA UL Molded case circuit breaker

**A Product Line / Family Requirements Specification** defines the requirements for a whole portfolio of related products, rather than a single system.

It identifies the core assets shared across the product line or family, as well as the points of variation.

Goal: Product requirements can reuse existing product line requirements or derived with more detailed information.

# What are we talking about?



The screenshot shows the interface of the Polarion PLM tool chain. At the top, there is a header with a 'Pn' icon, a folder icon labeled 'PLM\_tool\_chain', and a status bar showing '50 Requirements'. Below the header is a toolbar with various icons. The main area is a sidebar navigation menu. The menu items are:

- Steiner, Jochen (DI CS SD EH DS 3)  
My Polarion
- Home
- RS Training RE @SI EP
- Work Items
- Documents & Pages
  - Default Space
  - 10 Project Overview
  - 20 External Requirements
  - 30 System Requirements
  - 40 System Features
  - 50 Requirements
  - 60 Features
  - 80 Test

A mouse cursor is hovering over the '30 System Requirements' item in the 'Documents & Pages' section.

All requirements specifications that we expect below this folder:

Requirements that belong to

- **a classical system**,  
means overall behavior not component behavior
- **a product family or product line** with similar functions and behavior, so they can be reused in the product specific requirements specification
- **a specific aspect of many products**,  
so, they can reuse these defined constraints or requirements with the main goal of a topic specific repository

# Requirements for a specific aspect of many products



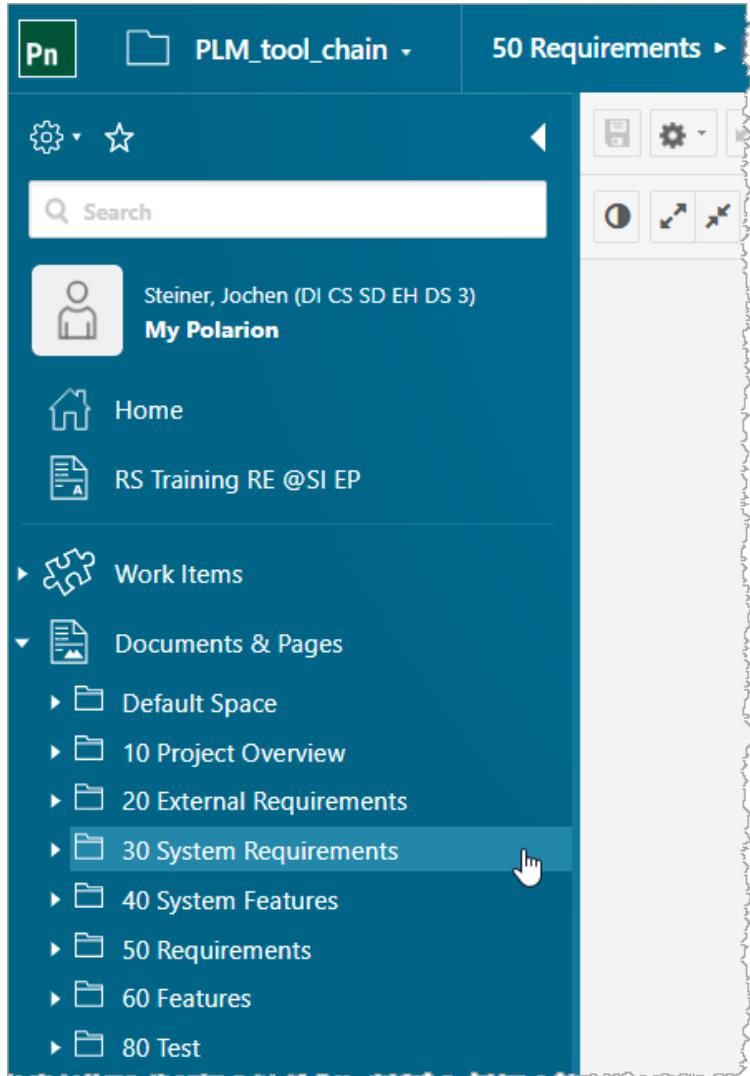
E.g. Hardware design requirements

**A Cross-Product Requirements Specification** focuses on a specific aspect or capability that spans across multiple products.

This could be things like communication protocols, serviceability, security, user experience, etc.

Goal: Product requirements can reuse existing requirements and probably the provided Solution concepts in the System Feature Specification

# What are we talking about?

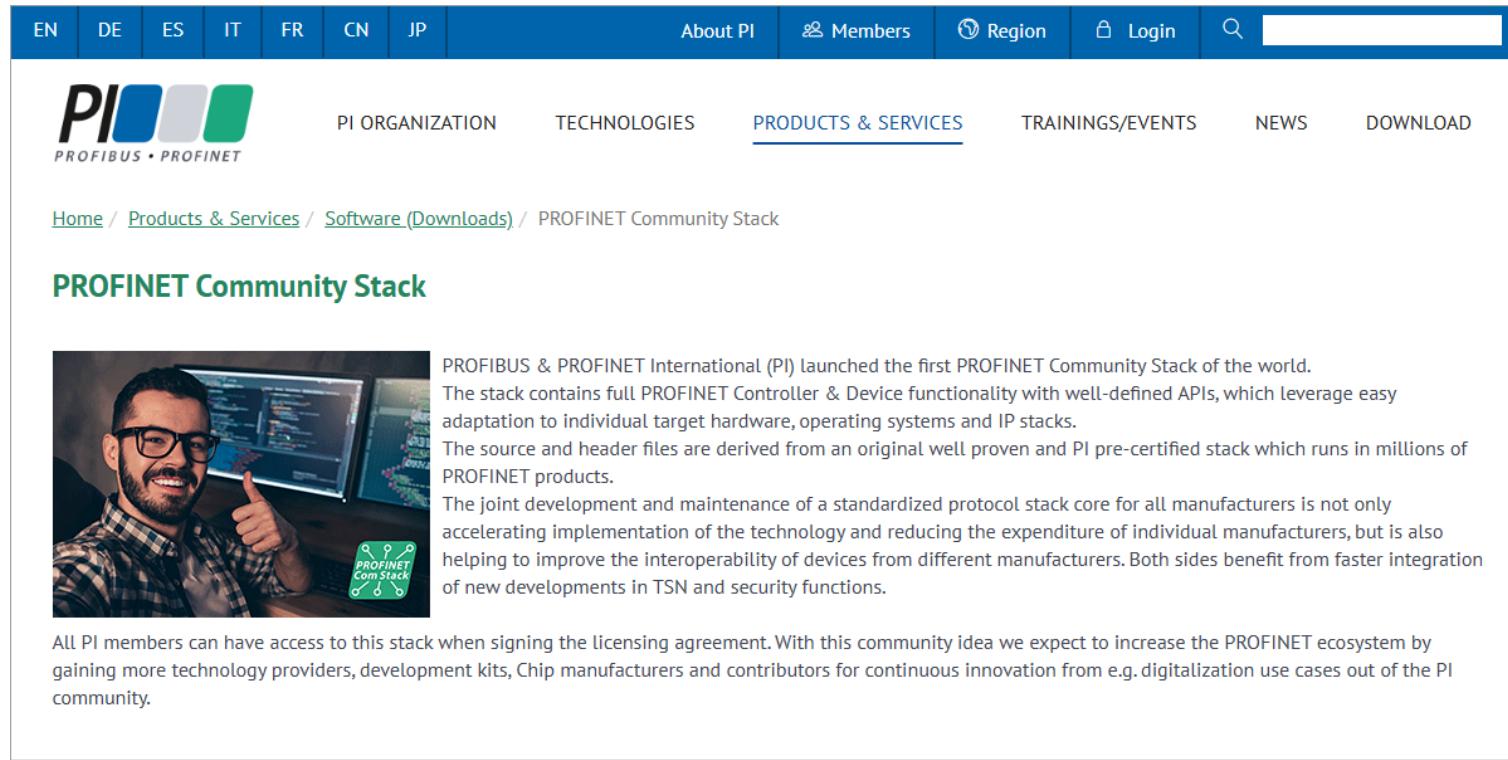


All requirements specifications that we expect below this folder:

Requirements that belong to

- **a classical system**,  
means overall behavior not component behavior
- **a product family or product line** with similar functions and behavior, so they can be reused in the product specific requirements specification
- **a specific aspect of many products**,  
so, they can reuse these defined constraints or requirements with the main goal of a topic specific repository
- **a platform**, so that the solution can directly be reused

# Requirements for a platform project



The screenshot shows the official website of PROFIBUS & PROFINET International (PI). The top navigation bar includes links for EN, DE, ES, IT, FR, CN, and JP. On the right side of the bar are links for About PI, Members, Region, Login, and a search bar. Below the navigation is the PI logo and a horizontal menu with links for PI ORGANIZATION, TECHNOLOGIES, PRODUCTS & SERVICES (which is underlined), TRAININGS/EVENTS, NEWS, and DOWNLOAD. Under the PRODUCTS & SERVICES link, there are sub-links for Home, Products & Services, Software Downloads, and PROFINET Community Stack. The main content area features a heading "PROFINET Community Stack" above a photograph of a smiling man with glasses pointing at a computer screen displaying code. To the right of the photo is a block of text explaining the stack's purpose and benefits. At the bottom of the page, there is a note about licensing and the community idea.

PROFINET Community Stack

PROFIBUS & PROFINET International (PI) launched the first PROFINET Community Stack of the world. The stack contains full PROFINET Controller & Device functionality with well-defined APIs, which leverage easy adaptation to individual target hardware, operating systems and IP stacks. The source and header files are derived from an original well proven and PI pre-certified stack which runs in millions of PROFINET products. The joint development and maintenance of a standardized protocol stack core for all manufacturers is not only accelerating implementation of the technology and reducing the expenditure of individual manufacturers, but is also helping to improve the interoperability of devices from different manufacturers. Both sides benefit from faster integration of new developments in TSN and security functions.

All PI members can have access to this stack when signing the licensing agreement. With this community idea we expect to increase the PROFINET ecosystem by gaining more technology providers, development kits, Chip manufacturers and contributors for continuous innovation from e.g. digitalization use cases out of the PI community.

E.g. PROFINET Communication Stack

**A Platform Requirements Specification** needs to strike a balance between defining the core, reusable capabilities and providing the flexibility and extensibility for diverse products to be built on top of the platform. This requires a more abstract, modular, and forward-looking approach.

Due to the high rate of derived we recommend to place it as SRS.

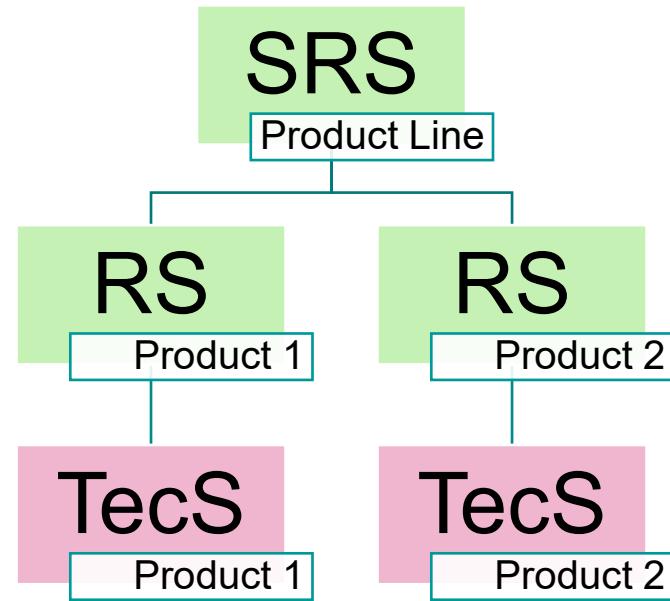
Goal: a platform is the highest level of reuse!

**SIEMENS**

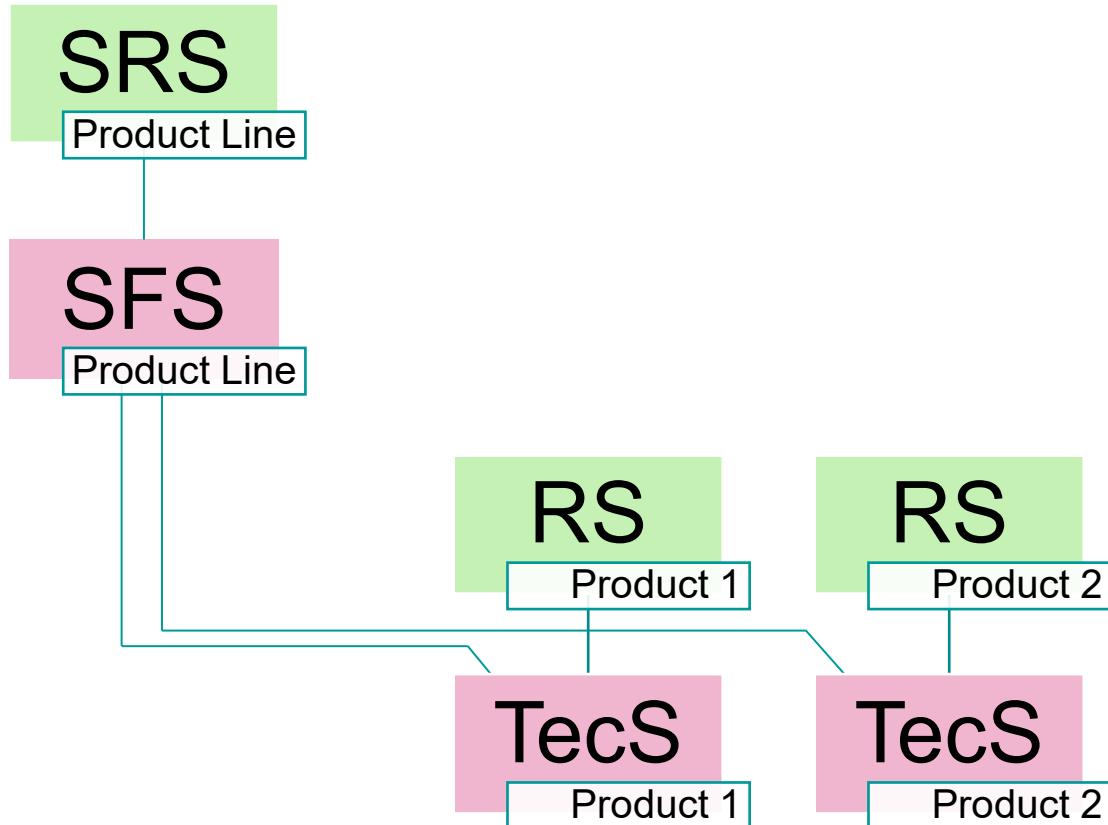
**Reuse is the goal!**

- 1. Increased Efficiency and Productivity**
- 2. Reduced Development Costs**
- 3. Improved Quality and Reliability**
- 4. Faster Time-to-Market**
- 5. Consistency**
- 6. Reduced Maintenance Overhead**
- 7. Organizational Learning and Knowledge Sharing**

# Structures for Requirements for reusing common requirements



# Structures for Requirements with System Feature Specification



## Definition Quiz



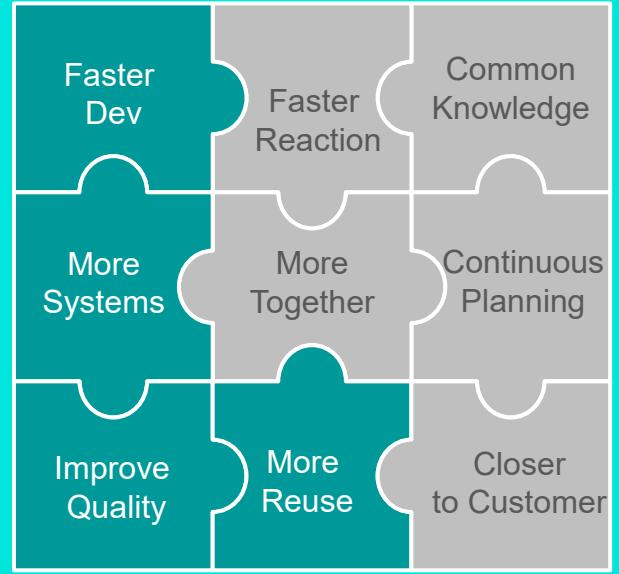
Muss ich zu jeder System Requirement Spezifikation eine System Feature Spezifikation erstellen?

- A Ja
- B Nein

Mehrere richtige Antworten möglich

# Reuse of Requirements

Where to place potential reused requirements?



Q&A