Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 136 (2025) 308–313

www.elsevier.com/locate/procedia

35th CIRP Design 2025

Designing resilient product architectures with systems engineering – Applied in the context of an automation technology product

Michael Bitzer[a], Chantal Sinnwell[a]*

Siemens Industry Software GmbH; Am Kabellager 9, 51063 Köln, Germany

* Corresponding author. *E-mail address:* michael.bitzer@siemens.com

Abstract

A PLC (=Programmable Logical Controller) is an automation component which can be build-in in multiple manufacturing machines and lines across various industries. When designing a PLC, not all possible use cases, stakeholders, or contexts, which determine the product requirements, are known upfront. To cope with this uncertainty of requirements, the PLC needs to have a resilient product architecture. Using this as an example, this paper addresses the following research question: How are resilient product architectures designed using the methodology of Systems Thinking and Engineering while confronted with the uncertainty of system contexts and requirements.

Keywords: Systems Thinking & Engineering; Product Architecture; Desgin Decisions

1. Introduction & Motivation

The industrial eco-systems, in which companies are operating their business and engineering their products, are constantly evolving. Society and technical mega-trends are impacting the industrial eco-systems. Typical mega-trends within the industrial eco-system are Sustainability, Interdependent World, Digital Transformation, Industry 4.0, Society 5.0, Smart Systems and Complexity [1]. Those megatrends and eco-system conditions force automation component manufacturers to rethink their approach to product architectures. There is a need for resilient product architectures, which can cope with a degree of uncertainty, that has not been considered so far. This paper provides an approach to leverage the concept of System Thinking and Engineering to approach that need in a structured and systematic way based on proven principles and methodologies.

Companies in discrete and process manufacturing industries are working in this industrial eco-system as well. Typical examples of such industries are machinery, plant engineering, automotive, food or agricultural industries. Those companies operate eighter in a “business to business” (B2B) or “business to customer” (B2C) context. Independent of their business context (B2B or B2C) their products go through the following Life Cycle Stages: Concept, Development, Production, Utilization, Support and Retirement [2].

In the phase of “Production”, typically, many kinds of automation technology in machines or product lines are used. From the standpoint of an automation technology company, this means that all those mentioned industries and companies are potential customers and stakeholders. Therefore, a manufacturer of automation components needs broad knowledge also in the field of manufacturing and how its products provide value to the production. From an engineering perspective, this implies that all requirements of those stakeholders need to be considered to design successful automation products.

10.1016/j.procir.2025.08.054

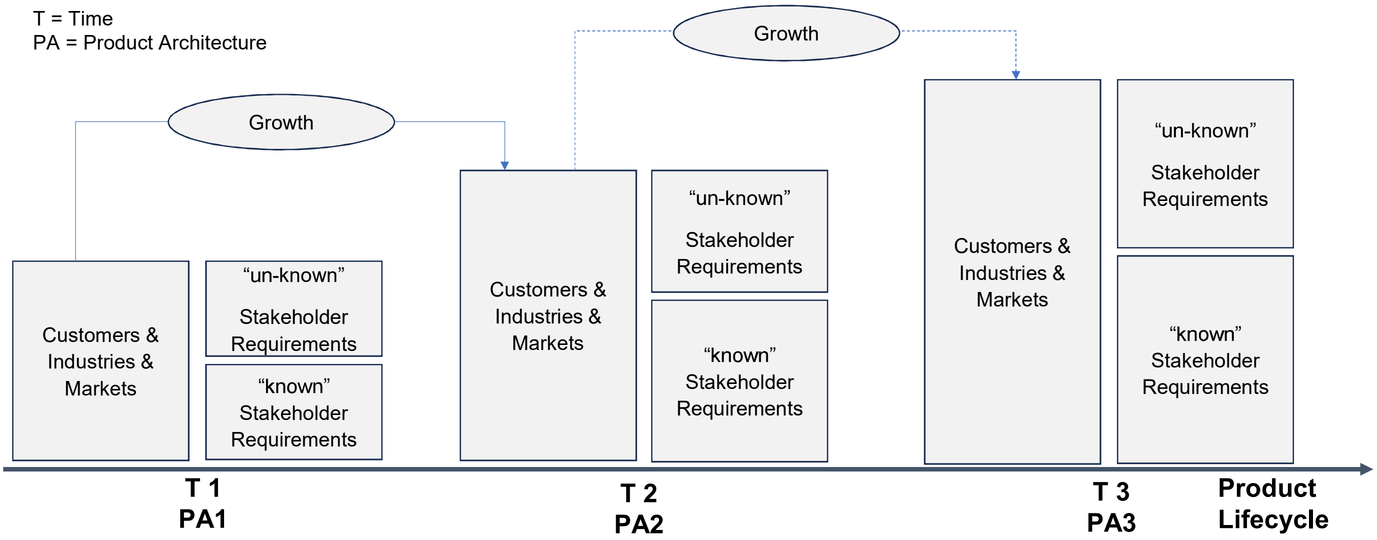

T = Time
PA = Product Architecture
Growth
Growth
Customers & Industries & Markets
"un-known" Stakeholder Requirements
"known" Stakeholder Requirements
Customers & Industries & Markets
"un-known" Stakeholder Requirements
"known" Stakeholder Requirements
Customers & Industries & Markets
"un-known" Stakeholder Requirements
"known" Stakeholder Requirements
T 1
PA1
T 2
PA2
T 3
PA3
Product Lifecycle

Fig. 1. Challenges of an automation technology manufacturer to address growing stakeholder requirements.

When starting the concept and development phase, typically, not all stakeholders are known, ("un-known Stakeholder Requirements"). One reason for this can be, that the automation technology manufacturer is operating in a B2B context, where the end-customer is not known. Or, over time there is a "Growth" in the customer base. By this, the number of stakeholders typically grows throughout the lifecycle of the automation technology.

To address both situations, "not knowing all stakeholders" and "a growing number/variety of stakeholders", automation technology needs to be able to cope with growing stakeholder requirements in the Product Architecture (PA1, PA2, PA3, …) throughout the Product Lifecycle (Time = T1, T2, T3, …). Figure 1 depictures this challenge of the product architecture of an automation technology.

From a Systems Engineering and Product Architecture perspective, there are mainly two ways to address the described challenges. Eighter, at each point in time (T1, T2, …) a new product architecture (PA1, PA2, …) is designed. Or the product architecture needs to be able to enhance to fulfill the growing set of stakeholder requirements. In this paper, this ability of the product architecture to enhance is called "resilient product architecture".

2. Product Architecture – robust & resilient

In Systems Engineering, during the phase of concept and development, the product architecture is defined. This typically is an iterative process, which helps to bring together multiple engineering disciplines and stakeholders. In this iterative process multiple perspectives/ views on the product are elaborated: requirements, functional view, logical view and physical/ technical view (called R-F-L-T). [2]

The figure 2 illustrates the "architecture definition process" in three steps. By this, both terms of "robust" and "resilient" product architectures are introduced.

Step 1: Context and Stakeholder Analysis

At the beginning of the architecture definition process, the analysis of the system context and stakeholders is essential to elaborate a holistic set of requirements.

In order to design a "robust" product architecture the stakeholders typically can be seen as "known". Stakeholders

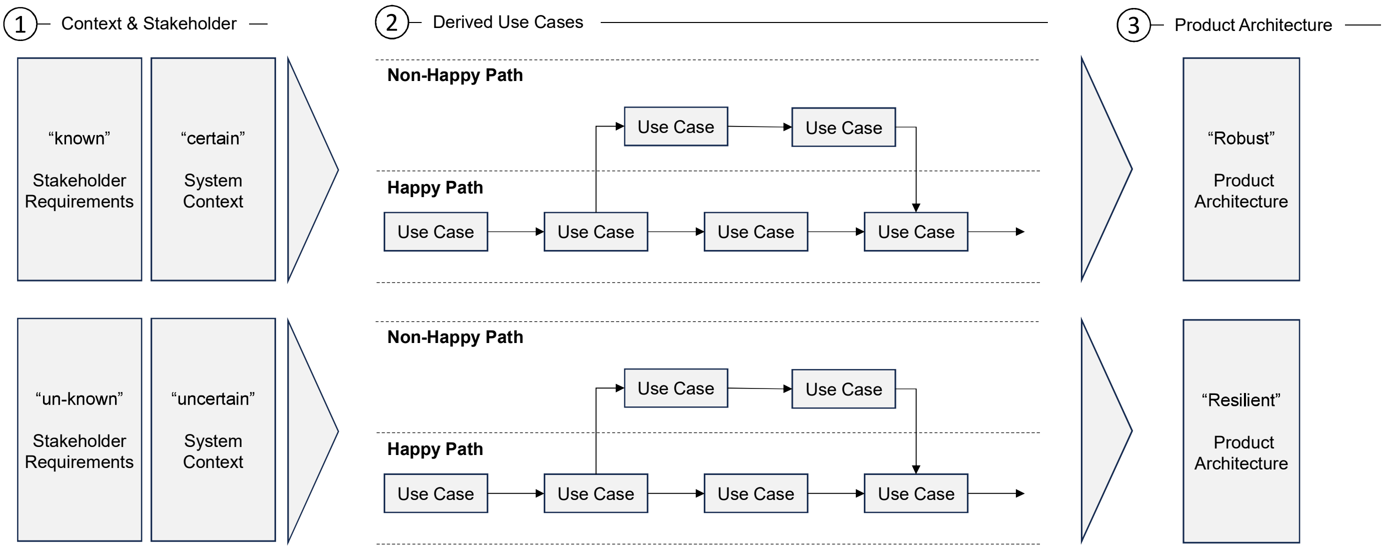

1 Context & Stakeholder
2 Derived Use Cases
3 Product Architecture
"known" Stakeholder Requirements
"certain" System Context
Non-Happy Path
Happy Path
Use Case
"Robust" Product Architecture
"un-known" Stakeholder Requirements
"uncertain" System Context
Non-Happy Path
Happy Path
Use Case
"Resilient" Product Architecture

Fig. 2. Illustrated process of architecture definition addressing both types of product architectures: robust and resilient.

from all phases of the lifecycle need to be covered: concept, development, production, usage, maintenance, retirement. The same applies for their requirements. A context typically is defined by properties (e.g. environmental conditions, governmental regulations). To design a robust product architecture, those properties can be seen as “certain”. The term “certain” is used in this paper equal to the understanding that characteristics (values) of the properties are “predictable”.

To design a “resilient” product architecture the stakeholders typically can be seen as “unknown”. The characteristics of the context are “uncertain”. The term “uncertain” is used in this paper equal to “unpredictable”.

Step 2: Derived Use Cases

As a next step in the architecture definition process, the use cases are derived based on the stakeholder requirements. Those use cases help to formally describe the requirements in a more detailed way. In operational practice of Systems Engineering typically two “paths” are elaborated:

- the “happy path” addresses a straightforward sequence of use cases in “normal operations”.
- the “non-happy path” addresses a sequence of use cases in situations as offset of “normal operations”.

This applies for both types of product architectures.

Step 3: Product Architecture

Following the architecture definition process, functional and physical views can be elaborated. In this paper the term product architecture is used in accordance with the term system architecture (following INCOSE / Systems Engineering):

Product Architecture: “The fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution.” [2]

In this paper the term “robust product architecture” is used for products which can serve stakeholder requirements also in regard to situations within “normal operations” (happy path) and outside of “normal operations” (un-happy path). [4].

Moreover, in this paper the term “resilient product architecture” is used for products which can serve stakeholder requirements also in situations of uncertainty and unknown requirements. Here the following additional definitions help to frame:

“Resilience is the ability to predict and plan for actual or potential adverse events, detect their emergence as soon as possible and prevent them from occurring, mitigate their severity, limit the damage caused, recover from them rapidly, adapt to them successfully and learn the relevant lessons.” [5]

“[…] System Resilience is the ability of an engineered system (or System of Systems) to provide required capability when facing adversity. For the purpose of resilience, an adversity is anything that might degrade the capability provided by a system. Achieving resilience requires consideration of all sources and types of adversity; e.g., from environmental sources, human sources, or system failure; from adversarial, friendly, or neutral parties; adversities that are malicious or accidental; adversities that are expected or not. Adversities may be issues, risks, or unknown-unknowns. Adversities may arise from inside or outside the system. The fundamental objectives of resilience are avoiding, withstanding, and recovering from adversity. The means of achieving these fundamental objectives include Adaptability, Agility, Anticipation, Continuity, Disaggregation, Evolution, Graceful Degradation, Integrity, Preparation, Prevention, Re-architecting, Redeploying, Robustness, Situational Awareness, Tolerance, Transformation, and other methods. Resilience focuses on providing required capability - not necessarily with maintaining the architecture or composition of the system.” [6]

3. Design Decision

In this paper the term “Design Decision” is used with the intent “to select appropriate technological or technical system elements that compose the system” [2]. Moreover, the process to come to appropriate Design Decisions includes “to analyze and estimate architectural and design characteristics of candidate architectures and system elements, […] to select the most efficient ones in terms of costs, technical risks, […] and other stakeholder concerns such as critical quality characteristics, affordability, maintenance etc.” [2].

The figure 3 continues the “architecture definition process” (abstraction of [2] and [3] within the “Technical Processes”) in three steps to enable a Design Decision.

Step 4: Risk Analysis

The approach of Systems Engineering also includes aspects of Risk Management. Risk Analysis and mitigation strategies are essential for this. A “robust” product architecture addresses the risk of the failure of a necessary product function. In case of a “resilient” product architecture, it addresses the risk of missing required product functions which needs to be mitigated. To identify such risks, possible failures must be elaborated, and potentially missing product functions must be analyzed.

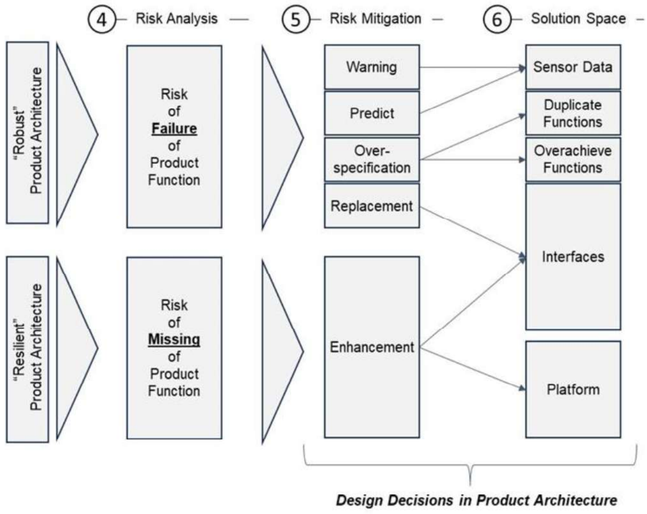
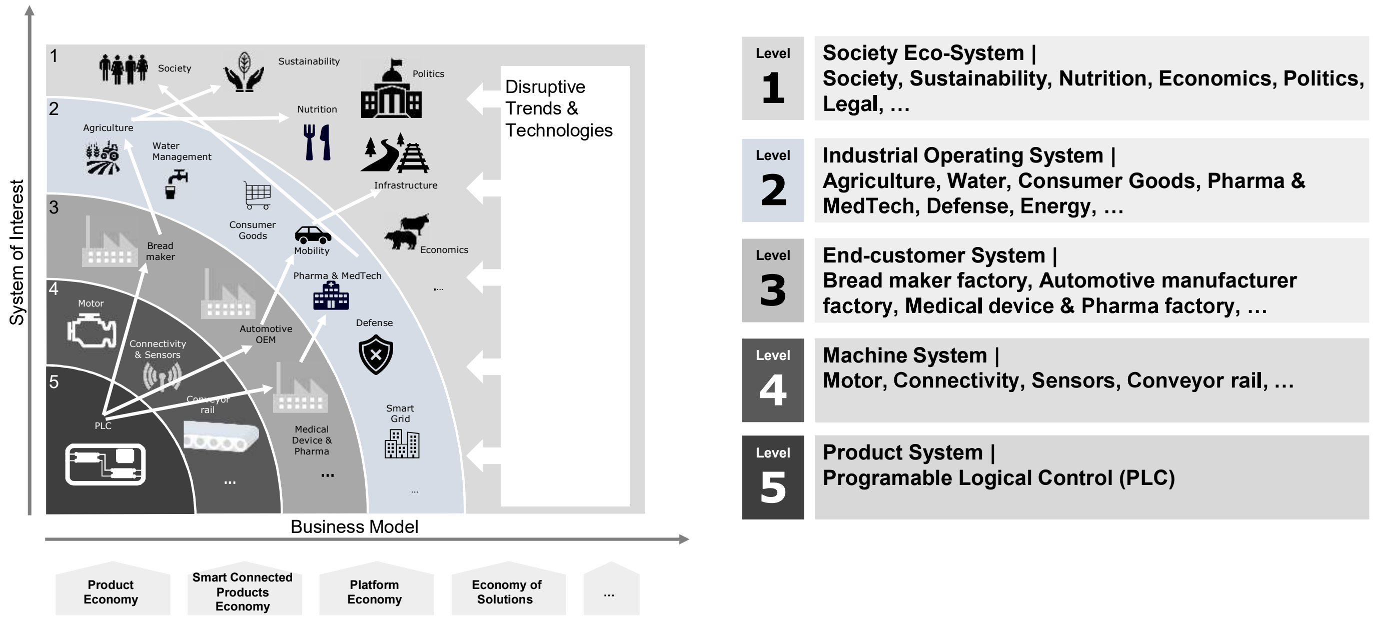

4 Risk Analysis
5 Risk Mitigation
6 Solution Space
"Robust" Product Architecture
"Resilient" Product Architecture
Risk of Failure of Product Function
Risk of Missing of Product Function
Warning
Predict
Over-specification
Replacement
Enhancement
Sensor Data
Duplicate Functions
Overachieve Functions
Interfaces
Platform
Design Decisions in Product Architecture

Fig. 3. Illustrated process of architecture definition addressing Design Decisions.



Level 1 — **Society Eco-System | Society, Sustainability, Nutrition, Economics, Politics, Legal, …**

Level 2 — **Industrial Operating System | Agriculture, Water, Consumer Goods, Pharma & MedTech, Defense, Energy, …**

Level 3 — **End-customer System | Bread maker factory, Automotive manufacturer factory, Medical device & Pharma factory, …**

Level 4 — **Machine System | Motor, Connectivity, Sensors, Conveyor rail, …**

Level 5 — **Product System | Programable Logical Control (PLC)**

Fig. 4. Industry Example: simplified and anonymized result of PLC context analysis.

Step 5: Risk Mitigation

To mitigate those risks, several levers can be used. For "robust" product architectures, those levers can be for example warnings in case a product function fails (e.g. signal light on machines in production line). This is a reactive mitigation strategy. A more proactive mitigation strategy is to predict a failure of a product function (e.g. predictive maintenance of machine oil).

For "resilient" product architectures typically the mitigation strategy is to establish the ability to enhance the product architecture and by this add the missing product function (e.g. additional software applications for machine communication protocols). For this, during the product architecture process, the design principle of "modular architectures" needs to be applied [7].

Step 6: Solution Space

The approach of Systems Engineering helps to engineer technical systems. To achieve this, the approach narrows down technical solutions step by step in the so-called solution space. Those technical solutions need to fulfill the risk mitigation. For "robust" product architectures the risks can be mitigated by multiple solutions. For example, sensor data can provide the necessary information for the risk mitigation strategy of warnings or predictions. In case of "resilient" product architectures typically two solutions are utilized: first, interfaces help to be able to enhance the product architecture and enhance product functionality. Second, platforms for software applications can be leveraged to enhance product functionality.

4. Industrial Example

The industrial example in this paper is a manufacturer of automation technologies. In this example the automation technology of a PLC is used. A PLC (=Programmable Logical Controller) is an automation component which can be build-in in multiple manufacturing machines and lines across various industries. When designing a PLC, not all possible use cases, stakeholders, or contexts, which determine the product requirements, are known upfront. To cope with these unknown and growing requirements, the PLC needs to have a resilient product architecture.

To come up with a resilient product architecture for a PLC the steps of the above-described approach were applied and described in the following paragraph.

Step 1: Context and Stakeholder Analysis

The context of the PLC was investigated by leveraging the approach described above. To illustrate the context of the PLC a framework called "Business Engineering" was utilized (based on [8] [9]).

Figure 4 is showing the simplified and anonymized result of this investigation. The result is showing the multiple dependencies of stakeholders: B2B, B2C, known and unknown, as described in the chapters above. Starting from the product PLC itself (here Level 5) the investigation was done up-to the Society Eco-System (Level 1).

Step 2: Derived Use Cases

Based on the Context and Stakeholder Analysis relevant use cases were derived. Since PLCs are used in multiple industries – such as machinery, automotive or pharma – a wide range of use cases were elaborated.

Step 3: Product Architecture

The product architecture was engineered based on the use cases and following the iterative approach "RFLP" of Systems Engineering. Leveraging the use cases as requirements (R), the necessary functions (F) were elaborated to serve those requirements. As part of the product architecture work the logical (L) and physical (P) elements, to fulfill those functions, were defined. A typical example for a function (F) of a PLC could be "control material flow on a production line".

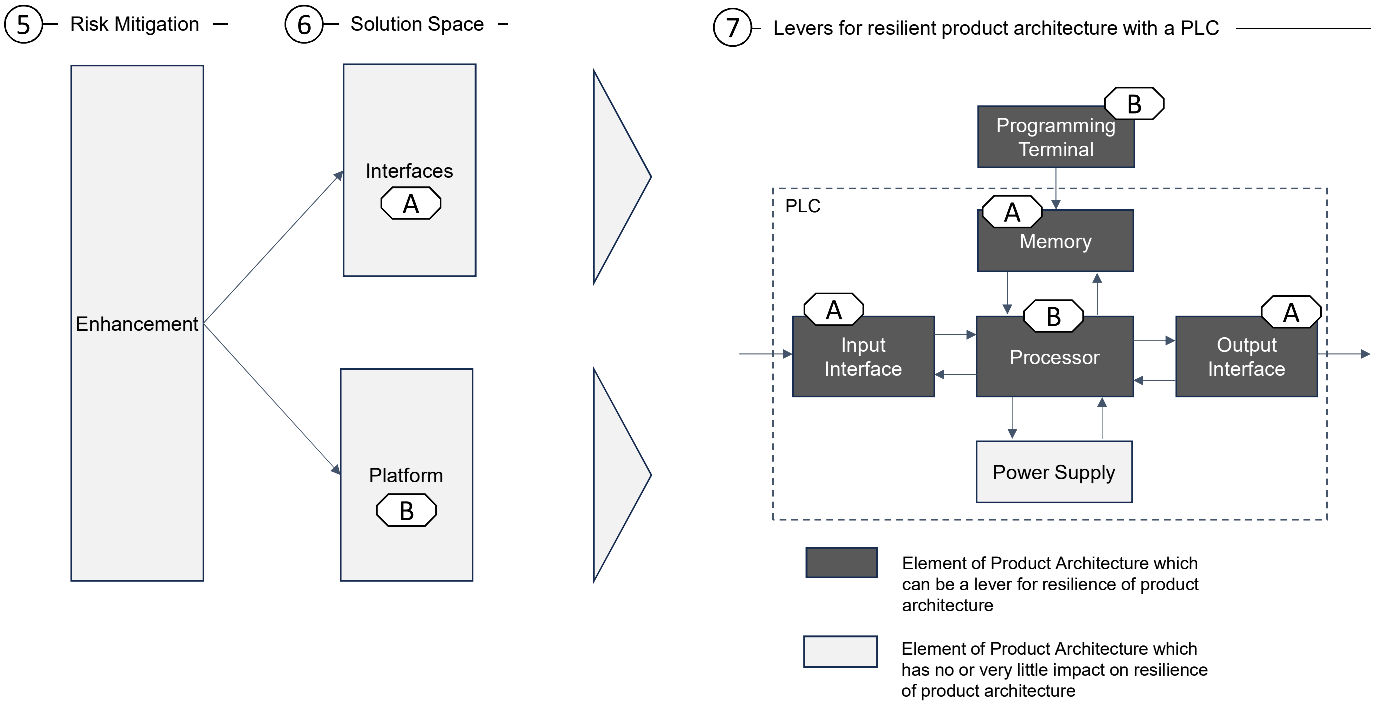

5 Risk Mitigation
6 Solution Space
7 Levers for resilient product architecture with a PLC
Enhancement
Interfaces
A
Platform
B
Programming Terminal
B
PLC
A
Memory
A
Input Interface
B
Processor
A
Output Interface
Power Supply
■ Element of Product Architecture which can be a lever for resilience of product architecture
□ Element of Product Architecture which has no or very little impact on resilience of product architecture

Fig. 5. Industrial Example of a PLC: Design of a resilient product architecture (focus on physical/ technical view)

Moreover, a typical requirement for this function could be "Rotation scalable depending on type of product from 30 to 2500 rotations per minutes".

Step 4: Risk Analysis

In industrial context a typical risk during product development are regulatory requirements which are not covered by the product design. There might be multiple reasons for this. In the example of a PLC this can happen when a new set of stakeholders with their industry-specific requirements are added over time. Example here can be pharma or medical device industry. To cope with this risk is a key element in this step.

Steps 5 to 7: Design Decision for the PLC

As part of the approach described in this paper, the Design Decision was applied for the PLC. The following figure 5 illustrated the simplified physical/ technical architecture of a PLC. In this figure, also those elements are marked, which are relevant for the resilience of the architecture. Moreover, the types of enhancement possibilities are indicated: A = Interface and B = Platform.

In the context of the PLC a typical mitigation strategy is to fulfill respective functionalities of the product architecture by software – not hardware. The intent is to be able to enhance the software in case new requirements appear over time and new functionality needs to be added. In the example of the requirement "Rotation scalable depending on type of product from 30 to 2500 rotations per minutes" the rotations per minute were identified as "parameter", which are leavers for a resilient product architecture.

During the implementation of the software the rotations per minutes were also designed and coded as parameters. By this, the software was used as a "platform" for "un-know" requirements (figure 5, label "B").

5. Discussion & Outlook

In this paper, the following research question was addressed: How are resilient product architectures designed using the methodology of Systems Thinking and Engineering while confronted with the uncertainty of system contexts and requirements.

To work on this research question, the business and product development situation of a manufacturer of automation technology was described and introduced. By this, the challenges of known and un-known stakeholder requirements were elaborated. The concepts of robust and resilient product architectures were defined. An approach was introduced step-by-step, which leverages Systems Thinking and Engineering to elaborate and develop a resilient product architecture.

Figure 6 summarized the above introduced approach: from Problem Space to Solution Space.

As a next step, the future research work will focus on the incorporation of additional engineering methodologies, with focus on product architecture alternative designs. Currently it is planned to incorporate methodologies for Design for X (e.g. design for reliability) and focus especially on aspects of Usability Engineering (e.g. Software GUI Mock-up).

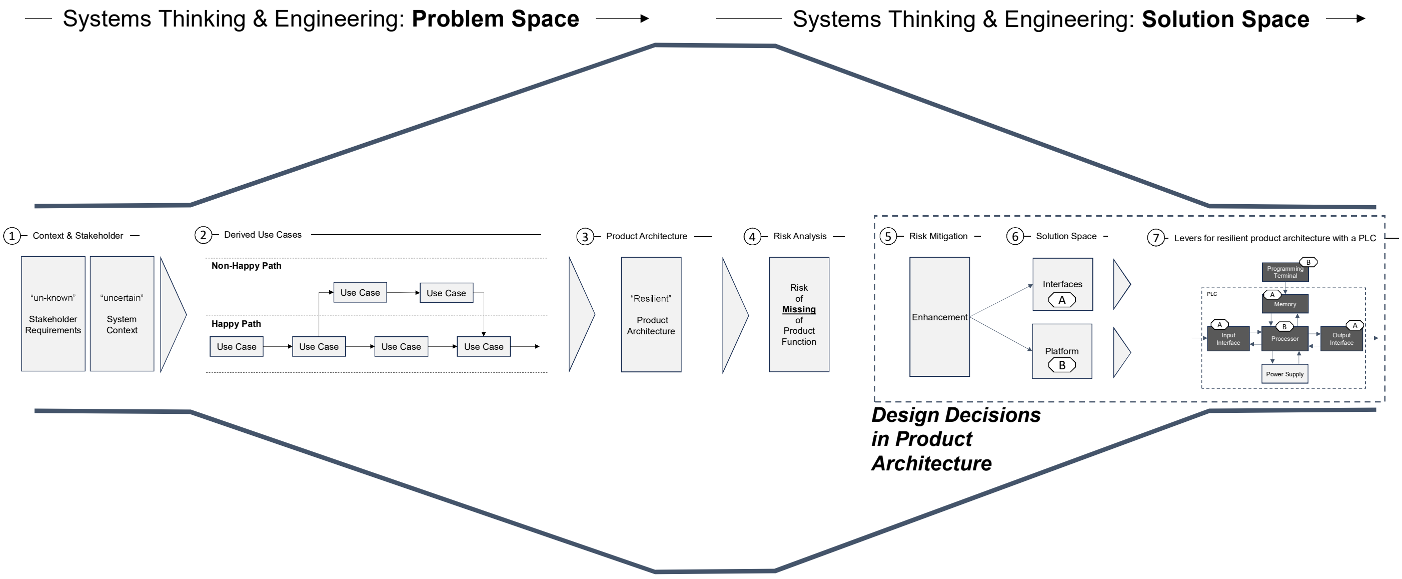

Systems Thinking & Engineering: **Problem Space**
Systems Thinking & Engineering: **Solution Space**
1 Context & Stakeholder
"un-known" Stakeholder Requirements
"uncertain" System Context
2 Derived Use Cases
Non-Happy Path
Happy Path
Use Case
3 Product Architecture
"Resilient" Product Architecture
4 Risk Analysis
Risk of Missing of Product Function
5 Risk Mitigation
Enhancement
6 Solution Space
Interfaces A
Platform B
7 Levers for resilient product architecture with a PLC
Programming Terminal
PLC
Memory
Input Interface
Processor
Output Interface
Power Supply
Design Decisions in Product Architecture

Fig. 6. Overview of approach: Leveraging Systems Engineering for resilient product architecture design

References

[1] INCOSE (Hrsg.), "Systems Engineering Vision 2035", San Diego, USA: INCOSE 2021.

[2] D. D. Walden, et al. (Eds.), Systems Engineering Handbook (5th ed.)., INCOSE International Council on Systems Engineering. San Diego, CA: John Wiley & Sons, Inc., 2023.

[3] ISO/IEC/IEEE 15288: Systems and software engineering - System life cycle processes; Second edition 2023-05

[4] Mathias, J.; Kloberdanz, H.; Engelbert, R.; Birkhofer, H.: Strategies and principles to design robust products. Design Conference 2010; Dubrovnik, May 17, 2010

[5] Wörner, J.-D./Schmidt, Chr. M. (Hrsg.): Sicherheit, Resilienz und Nachhaltigkeit (acatech IMPULS), München 2022. DOI: https://doi.org/10.48669/aca_2022-2

[6] INCOSE: Resilient Systems Working Group (RSWG) https://www.incose.org/communities/working-groups-initiatives/resilient-systems (latest access 13th Oct. 2024)

[7] Fuchs, C.; Golenhofen, F.: Mastering Disruption and Innovation in Product Management, Springer Verlag München, 2019

[8] Bitzer, M.; Michels, N.: Business Engineering: Systemische Betrachtung von Produkt- und Produktionssystemen in industriellen Eco-Systemen; ZWF Zeitschrift für Wirtschaftlichen Fabrikbetrieb; De Gruyter; Band 119, Ausgabe 1-2; 2024

[9] M. Bitzer, C. Sinnwell: Systems und Lifecycle Engineering – Strategische Souveränität durch Engineering Intelligence. In: W. Koch et al. Hg., TdSE, Bremen: GfSE 2023