Master's Thesis / Diplomarbeit

Investigating the Management of Uncertainty in Product Platform Lifecycles

by Augustin Friedel

Signature of the Author ____

Augustin Friedel
Mechanical Engineering & Management, TUM

Certified by ____

Donna H. Rhodes
Thesis Supervisor
Senior Lecturer and Principal Research Scientist, Engineering Systems Division, MIT

Accepted by ____

Eduard Igenbergs
Thesis Supervisor
Professor for Aerospace, TUM

Abstract

This research focuses on investigating the management of uncertainty in the lifecycle of product platforms. Complex systems, such as product platforms, are subject to uncertainties that may lead to suboptimal functional performance or even catastrophic failures if unmanaged over time. Identifying uncertainties in the front-end and implementing ways to mitigate the identified and possibly later upcoming uncertainties is a part of the product platform design process that can add value to the product platform as a system.

After an analysis of current systems engineering approaches for product platform development an empirical case study was conducted. The analysis of current approaches is documented in this thesis, as well as the results and the analysis of the empirical case study. The analysis of the empirical case study is presenting insights of platform projects out of different industry divisions. Insights are, for example, the consequences of ignored and unknown risks in the different parts of a platform during the lifecycle or the mapping of identified problems because of ignored or unknown uncertainties or risks to the phases of the platform lifecycle. As a result of the analysis of current approaches of systems engineering based platform development and the results of the empirical case study, a framework with 7 steps was created in this research. The framework can be used for investigating the uncertainties and related mitigation methods to develop a product platform that also delivers the expected functional performance in case of significant changes in the context. The overall goal of the framework is to identify the critical parts in a product platform design and implementing mechanisms to enable value robustness of the platform. The value is measured by the fulfillment of engineering metrics, which are performance indicators for satisfying the customer and stakeholder needs.

In the last part of the thesis the framework is applied to the development process of an illustrative product platform design. For the illustrative example the platform of iRobot® cleaning robots was amended by simplifying the design of the original robot design. As a disclaimer, the example in this research is not exhaustive and can vary significantly from the real iRobot® products. The views expressed in this thesis are those of the author and do not reflect the official strategy or position of the iRobot Corporation. It was attempted to use the best data as available in different patents and user manuals to make this example as realistic as possible.

Thesis Supervisor MIT: Dr. Donna H. Rhodes; Senior Lecturer and Principal Research Scientist, Engineering Systems

Thesis Supervisor TUM: Prof. Eduard Igenbergs; Professor for Aerospace

Acknowledgment

This thesis is the final step of my five-year journey in academics and research, which would not have been possible without the help of many people and organizations. First, I would like to begin by expressing my gratitude and thanks to my advisors Dr. Donna H. Rhodes and Professor Eduard Igenbergs for providing me with this research opportunity and for their guidance and insights that made this thesis possible. Both provided valuable feedback and guidance that shaped this research. In addition to Dr. Rhodes I would like to thank Dr. Adam M. Ross for the time with all the possibilities for successful research and personal growth in their research group Systems Engineering Advancement Research Initiative (SEAri)[1] at the Massachusetts Institute of Technology (MIT)[2].

In addition, I would like to express my thanks to Professor Ulrich Walter for the agreement of advising the research at his Institute of Aerospace (lrt)[3] at the Technische Universität München (TUM)[4].

My special thanks goes to Dr. Armin Schulz for the received trust support during my time working for his company 3D Systems Engineering [5], for the great support on my way to MIT, and for the feedback during the time there. I also would like to thank my former workmates Dr. Thilo Pfletschinger and Dr. Victor Lévárdy for providing industrial insights and valuable academic input during my research.

The MIT and especially the SEAri group was a great environment for my research and I am grateful to my colleagues and fellow graduate students at MIT for invaluable discussions, insights and suggestions related to this research. Many thanks to Kacy Gerst, Nirav Shah and all further SEAri-Members and Affiliates who gave great advice during my time. I would like to thank Professor Olivier de Weck, Dr. Qi Hommes and Mr. Pat Hale for their input and feedback on my research. My thanks goes to Professor Richard de Neufville and Professor Mort Webster for the opportunity to participate in their course on Risk and Decision Analysis, which provided useful information about methodologies I applied in this research.

I am really impressed by the atmosphere at MIT; it is a pleasant experience that an atmosphere at a school can also be so optimistic and supportive for the students. For me it is a great place to work on ideas in an absolutely professional manner and to get encouraged by the surrounding of great people to reach your goals.

[1] http://seari.mit.edu
[2] http://mit.edu
[3] http://lrt.mw.tum.de
[4] http://www.tum.de
[5] http://3dse.de

The conducted empirical case study added real value to my research and I would like to thank all participants in the, who provided valuable data about the uncertainties in product platform projects, which were giving a rise to the contributions of this thesis; I never thought that I would get answers with such a high quality of data.

I would like to thank Erich-Becker-Stiftung for the funding support during my time in the United States of America.

Thanks to my friend Julia for the feedback on the thesis and the additional support. Finally, my heartfelt gratitude goes to my family who provided never-ending encouragement for my academic endeavors over the past five years, a sympathetic ear and a place to always come home to. Danke, Mama, Oma, Thomas, Sophie, Josefa und Rosina.

Table of Contents

Table of Tables

Table of Figures

1 Introduction

This Chapter covers the research context description, and the personal and industrial motivation. It also sketches the goals, represented in two separate research questions and the outline of the masters' thesis.

1.1 Context

Platforms have been used in diverse industries over a period of time and there are numerous examples in literature of product families based on product platforms. To name some of these, product platform projects that are often quoted were developed by Black & Decker, Sony Walkman, Hewlett Packard, Volkswagen and Boeing (Simpson et al. 2006).

Sony, for example, built all of its Walkman® audio players around module based platforms by using the advantages of the modular design to produce a variety of products at low cost, allowing them to introduce more than 250 models in one market (Sanderson and Uzumeri 1997). Black & Decker developed a family of universal electric motors that were scaled along their stack length to produce a range of power output suitable for hundreds of their basic tools and appliances (Lehnerd 1997). Hewlett Packard successfully developed several of their ink jet and laser jet printers around a platform based on modular components (Feitzinger and Lee 1997). Volkswagen developed a platform, which shared common modules among a wide variety of products sold under the Volkswagen, Audi, Seat, and Skoda brands (Wilhelm 1997). Boeing developed many of its commercial airplanes by 'stretching' the aircraft to accommodate more passengers, carry more cargo, or increase flight range (Sabbagh 1996).

The primary objective of this thesis is to investigate different methods for managing uncertainty in the development of product platforms. De Weck (2006) mentioned, "One of the key challenges is to be able to predict the future or to design the platforms so that the expected and unexpected changes can be accounted for during the original design of the platform."

Gathered industrial experience has shown that it is valuable to study the uncertainties, which are occurring in the lifecycle of product platforms. The named examples have shown that a wide range of product platforms exists, and the platform approach is applied across different industries. The specifications of the different platform project are so different that it is not possible to generate a general way to manage the uncertainties. The process of managing uncertainty has to be adopted in consideration of the specifications of the product platform project. Managing in this case means identifying uncertainties and recommending ways to treat them to get a value robust product platform over the whole lifecycle.

Managing uncertainty in product platforms is a very broad field, so further research beyond this Masters Thesis can be valuable. After consideration the focus of this thesis is on managing uncertainty concerning the functional aspects of a product platform.

The thesis is also not providing a solution for a trade-off decision for changing from a single product development to a platform approach in consideration of the advantages and downsides of a product platform project. The question about trade-offs related to points like how many possibilities for customization are needed and what is the best commonality-rate are not addressed in detail in the thesis. Platform projects can be classified with different levels of complexity. In this research platform projects with different grades of complexity are examined but investigations about the connection between the grade of complexity and uncertainties are not addressed in a detailed manner.

1.2 Motivation

1.2.1 Personal Motivation

This thesis proposes a structured approach that can be used by product platform development teams to investigate the management of uncertainty regarding functional performance. I became interested in design methods for platforms after working on a platform project as a trainee in industry. I saw the huge potential of the platform approach, but I became aware that the process of platform design was generally not explicitly understood. It seems that some of the confusion and uncertainty around the platform development process can be reduced through structured methods. This would allow the teams to develop successful product platforms more efficiently, and turn them into market winners.

The thesis describes the journey of developing a method for investigating the management of uncertainty that will help platform developers with a complicated process – that of transforming an idea into a finished product.

1.2.2 Industrial Motivation

In the early part of the 20th century Henry Ford stated, “You can have any color car you want as long as it’s black” (Pine 1993). Today’s global marketplace has changed dramatically since then. In many industries the number of products per company on the market increased to satisfy the individual variety of customer needs (Martin 1999). Many companies are constantly struggling to find cost-effective solutions to satisfy the diverse demands of their customers. One possible solution is the development of a product platform, which can be used as an approach for a stream of derivative products, which can be efficiently developed and launched (Simpson et al. 2006). The stream of derivative products is often called product family, which is a group of related products that are derived from a platform.

The focus in this thesis is the product platform. This is related to numerous efforts in product family optimization design, the topic of product platform has so far received least attention and little achievement has been reported (Jiao et al. 2007). Research on product family design was, for example, done by Simpson et al. (2006).

Platform projects in the past failed or were less successful than expected because impacts on the functional performance related to risk and uncertainty were ignored. The consequences of the impact of an ignored or unknown uncertainty and risk can be immense. A current example is the recall of 5 million vehicles in U.S. because of a sticky gas pedal. Normally the platform is used for a whole product family and if there is a failure it can be harm the all derivatives of the product family

1.3 Research Questions

There are many not yet answered questions in product platform development, which were identified during the analysis of current platform development approaches. The focus of this thesis is on the following ones:

- Where do uncertainties occur in the platform lifecycle?
- What are the consequences of uncertainties and risks regarding the functional platform performance and the related products?
- How can you mitigate uncertainties by using different methods and approaches in the development process to obtain a value robust platform?

To answer the first two questions an empirical case study was conducted (see results in Chapter 3) and for the third question a conceptual framework is described and applied to an illustrative example.

The main goal in the empirical case study was the identification of the uncertainties, which are impacting the functional performance of a product platform during the lifecycle in a negative way. Another goal was the analysis and mapping of the identified uncertainties to phases of the platform lifecycle. Lifecycle in the thesis includes all phases from concept definition to retirement.

The goals of the framework were risk assessment of the identified uncertainties, identifying the critical parts in the product platform design and implementing mechanisms for the identified critical parts to build in "ilities", to deliver the context expected functional performance over epochs (fixed period of context and needs).

1.4 Thesis Outline

The thesis is split into 7 chapters; Chapter 1 starts with the description of the context, the scope and the description of the motivation and the research questions. Chapter 2 documents the results of the literature review, the analysis of current systems engineering approaches and the related terminologies. To answer the first of the research questions an empirical case study with several decision makers from different industries was conducted. The results of the empirical case study are documented in Chapter 3.

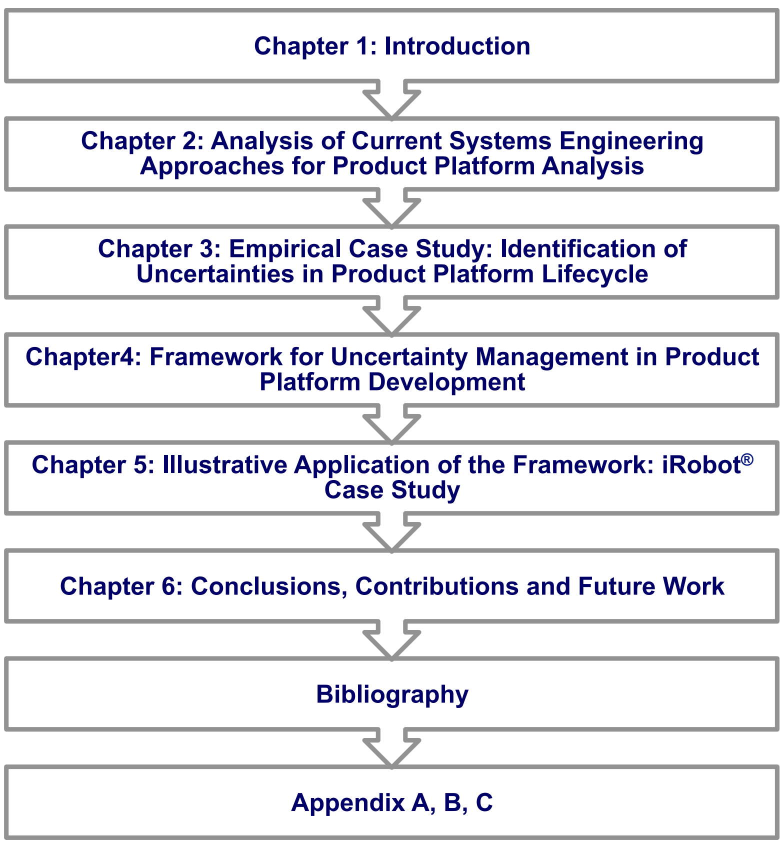


Figure 1-1: Outline of the thesis

To answer the second research question a framework for uncertainty management in product platform development was created. The description of this framework can be found in Chapter 4 of this thesis. To demonstrate the value and the logic of the framework, it was applied to an illustrative example, in this case the platform of iRobot[6] products. The application of the framework is described in Chapter 5. Chapter 6 describes the conclusions and contributions of the thesis and also some recommendations for future research. The Appendix includes the knowledge-gathering instrument for the empirical case study described in Chapter 3, Appendix B covers the flowchart of the framework described in Chapter 4 and in Appendix C the model-based systems engineering SysML diagrams of the illustrative example are documented.

[6] www.irobot.com

2 Analysis of Current Systems Engineering Approaches for Product Platform Analysis

This chapter covers the literature review about definitions of the term system and the derived definition of a platform. Furthermore, metrics for the sensitivity and functional performance of a product platform are described based on literature. This is a thesis in systems engineering and the most related terms are described in this chapter, further information about systems engineering can be found, for example, in the INCOSE Systems Engineering Handbook (INCOSE 2010).

2.1 Common Definitions of a System

There are various definitions for the term system in systems engineering literature and standards. Organizations like INCOSE, NASA, and US Department of Defense (DOD) define a system as follows:

- "A System is a combination of interacting elements organized to achieve one or more stated purposes an integrated set of elements, subsystems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements." (INCOSE 2010)
- "A system is a construct or collection of different elements that together produce results not obtainable by the elements alone. The elements, or parts, can include people, hardware, software, facilities, policies, and documents; that are all required to produce system-level results. The results include system-level qualities, properties, characteristics, functions, behavior, and functional performance. The value added by the system as a whole, beyond that contributed independently by the parts, is primarily created by the relationships among the parts; that is, how they are interconnected." (NASA 2007)
- "A System is a functionally, physically, and/or behaviorally related group of regularly interacting or interdependent elements; that group of elements forming a unified whole." (DOD 2008)
- A system defined by Igenbergs (2007) is an object with four characteristics. It consists of elements, the elements have attributes, relations describe the interaction between elements and an element can be a system.

The following two sections are describing matrices, which can be used for the description and schematic visualization of system and in this thesis for a platform. A methodology is the quality function deployment method by Hauser and Clausing (1988) and the engineering system matrix by Bartolomei (2007) is another one. The Quality Function Deployment is a part of the House of Quality (Hauser and Clausing 1988). The definition of a system can be used to describe a product platform as a system, what is documented in section 2.2.

2.1.1 Quality Function Deployment

Quality Function Deployment (QFD) is a process for systematically translating customer requirements into related engineering metrics during all stages of product development from the earliest stages of concept definition through production (INCOSE 1998a). It is a useful technique, particularly where the 'voice of the customer' is not clear. It provides an easy way to translate customer requirements into specifications and systematically flow down the requirements to lower levels of design, parts, manufacturing, and production (INCOSE 2010).

The part of the QFD, which is used in this thesis, is also known as creating the House of Quality and it is split into two phases, which are named phase I and II. The QFD Phase I depicts the relationships between customer requirements and engineering metrics, and the QFD Phase II depicts the relationships between the engineering metrics and the product components. That way, it is possible to analyze how changes in one part of the QFD (e.g., customer requirement) affect the other domains (e.g., engineering metric) (Hauser and Clausing 1988). The QFD approach is used in the framework described in Chapter 4 for mapping customer requirements to engineering metrics and then to connect engineering metrics with components of the product platform.

An engineering metric is a context-perceived metric that measures how well a context-defined objective is met. These are items, which are a translation of the subjective customer requirements into quantitatively expressible engineering specifications. The engineering metrics can be used as indicators for the satisfaction of the customer requirements. The characteristics of an engineering metric include its definition, units, and range from least to most acceptable values. The definition of the value range should be based on the customer requirements in order to ensure the delivery of the expected performance. The range reflects the fact that value is perceived for multiple engineering metrics levels. The limits can be named best and worst value; it depends on the engineering metric if the lower or the higher point is the best or worst.

2.1.2 Engineering Systems Matrix

Bartolomei (2007) conceptualized a matrix with five domains that are important when describing an engineering system. The five domains are social, technical, functional, process and environmental. A matrix based on these domains organizes the information about a system and can be used for network and graph theoretic analysis. Regarding to Bartolomei (2007), the derived analysis consists of varying classes of nodes, relations, and attributes. Nodes represent different classes of objects, relations describe interactions between two nodes, and attributes generically describe the parameters and descriptions for both nodes and relations. The row and column headings in the engineering systems matrix are identical, the diagonal of the matrix represents the components of the system and the cells off the diagonal represents the relationships between the components described by the row and column header. The engineering systems matrix is used in a modified form in the framework described in Chapter 4 to monitor the whole system and to trace the impact of the context changes to the different parts of the system, visualization can be found in Figure 4-18.

2.2 Platform Definition, Development Approaches and Market Segmentation

There are different definitions of the term platform documented in the existing systems engineering literature. Some of them are described in the following section. There are also several approaches for the development of a platform described in the literature; these are documented in section 2.2.2, as well as the description of a lifecycle.

2.2.1 Platform Definition as a System

On the highest level, the platform term is used in different combinations like product, process, brand, global or customer platform. Focus in this thesis is on the development of product platforms. Ulrich and Eppinger (2008) define platform as a collection of assets, including component designs, shared by multiple products. Ulrich (1995) proposed that the product architecture is the scheme where the physical components are associated to functional elements to form different products. Ulrich and Eppinger (2008) explain two dimensions in the architecture: the functional one, which is the group of operations and transformations that contributes to the general functionality of the product, and the physical one, which refers to the group of physical components and assemblies that enables a function. Jose and Tollenaere (2005) proposed that the architecture can be considered as a configuration between components of the product and the tasks that each component should do.

Meyer and Lehnerd (1997) defined a platform as “a set of common components, modules, or parts from which a stream of derivative products can be efficiently developed and launched”.

McGrath (1995) described a platform as “a collection of common elements, especially the underlying core technology, implemented across a range of products”.

Jiao et al. (2007) and Simpson and D’Souza (2004) defined a product family as “a group of related products that are derived from a common set of components, modules, and/or subsystems to satisfy a variety of market niches. The key to a successful product family is the product platform around which the product family is derived”.

The literature review proposed that product platforms have been defined diversely, but all have described the commonality aspect of a platform. Also, the definitions of the platform are based on the definitions of a system described in section 2.1. For the further use in this thesis a platform is defined as followed:

“A product platform is a set of architecture, common modules and interfaces from which a stream of derivative products can be efficiently developed and launched. The architecture is the configuration within the product; it is the scheme where physical components are associated to functional elements to form the platform. A module is part or a group that allocates a function to the product. Modules can be changed and replaced in a loose way and be produced independently. The interfaces are connections between the modules and architecture, the modules among each other, and between the platform and customized parts of the product.” (Jiao et al. 2007, Meyer and Lehnerd 1997, Simpson and D’Souza 2004, and Ulrich 1995)

The proposed definition is generic and can be adapted to many use cases and industries, which is based on the diversity of the product platform projects.

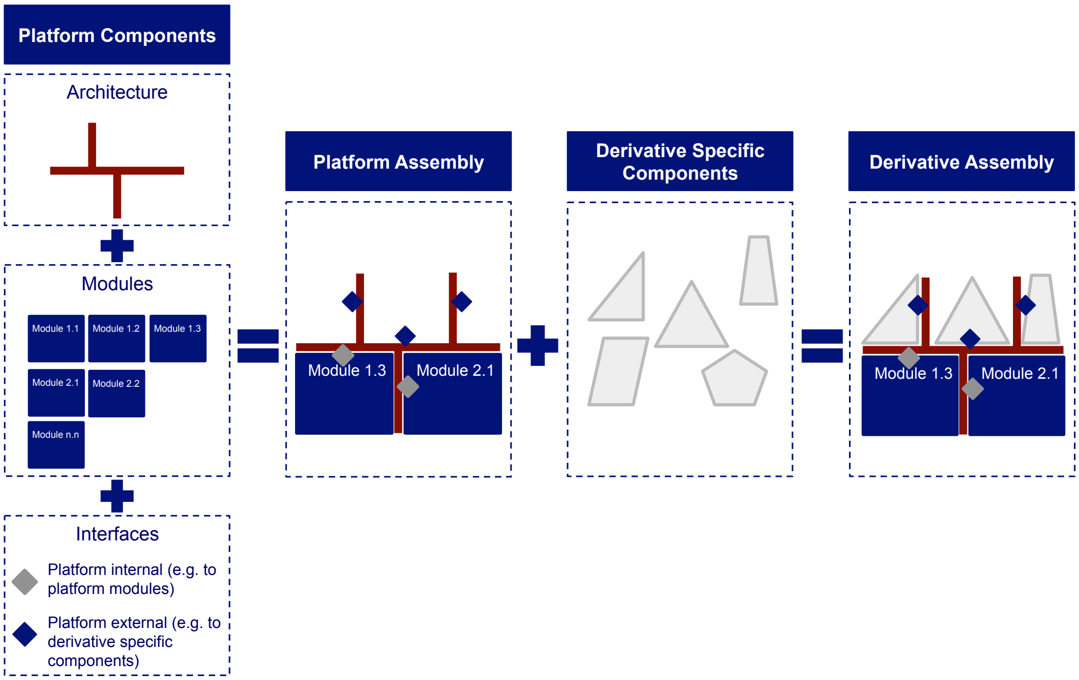


Figure 2-1: Process of platform and derivative assembling

In Figure 2-1 the definition of a platform is illustrated together with the assembling process. The whole set of derivatives per platform describes a product family, which is not displayed in this figure. The light-grey boxes in Figure 2-1 are the customized modules, which provide an opportunity to individualize a platform-based derivative. The big blue boxes are the standardized modules in the platform, which can exist in different versions based on the functionality of the derivative or on the updates of the modules along the phases of the platform lifecycle. The red lined construct displays the architecture of the platform; it is the scheme of the organization of the interfaces and the modules within the platform as mentioned in the definition above. The dark-grey diamonds in Figure 2-1 are illustrating the interfaces between the standardized modules and the architecture. The blue diamonds are the interfaces between the customized modules and the standardized architecture. The existing modules can be easily replaced by other ones, if the interfaces are well defined and designed to provide this opportunity. The sum of the customized modules is called customized part of a derivative, and the sum of the standardized modules, the architecture and the interfaces is called platform in Figure 2-1, which is coinciding with the given definition of a platform given in this section.

2.2.2 Platform Lifecycles and Development Approaches

In this research the lifecycle is the timespan between start of production and end of retirement. The definition of the lifecycle is based on the V-Model (INCOSE 2010, Walter 2009) for the development and the linear model for a systems lifecycle (Walter 2009) after start of production (SOP).

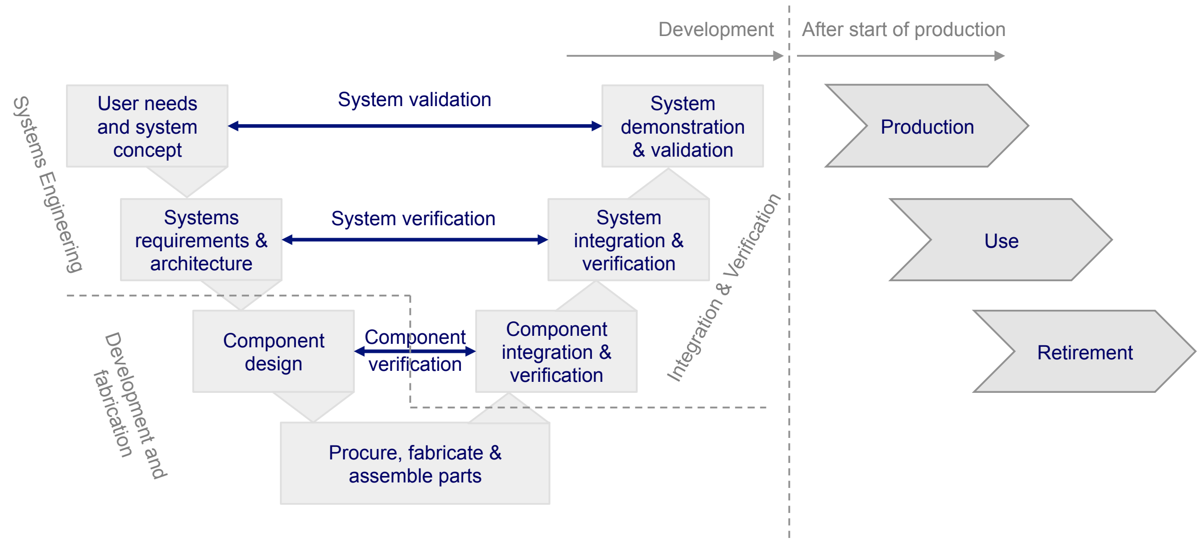


Figure 2-2: Lifecycle of a system including development phases and phases after start of production

The development is split in three phases with seven steps in total. In the systems engineering phase, the user needs are collected and a system concept is defined. The next step is deriving the systems requirements and to create system architecture based on the requirements. In the development and fabrication phase, the component are designed and it also includes the procurement, fabrication and assembling of the components. The last phase in development describes the integration and verification, split in component integration and verification, system integration and verification, and system demonstration and verification. The time after start of production is split in the phases production, use and retirement.

In the thesis, two different approaches for developing a platform are documented. The first platform development approach is displayed in Figure 2-3; in this case the platform is developed together with a lead derivative out of the product family.

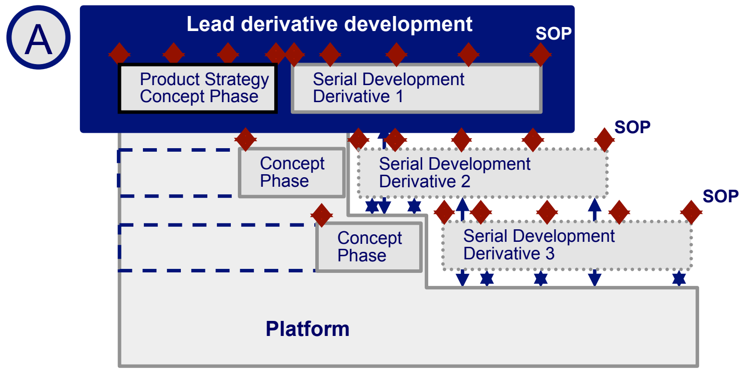


Figure 2-3: Approach A: platform development together with lead derivative

In Approach A the product strategy investigation is done together with the concept phase of the product platform, which is the same as the concept phase of the lead derivative. This means that there can be a lot of overlapping between the development of the concept of the platform and the lead derivative; that can have several advantages or downsides for the product platform, which have to be investigated separately for each product platform project. The developed 'backbone' is used for deriving the other derivatives of the product family. For each derivative a short concept phase is necessary and changes are mirrored back to the 'backbone' at defined milestones, if possible.

In Approach B the product platform is created first without a lead derivative and subsequently applied to a series of derivatives as presented in Figure 2-4. This approach is described and used by Gonzalez-Zugasti et al. (1999).

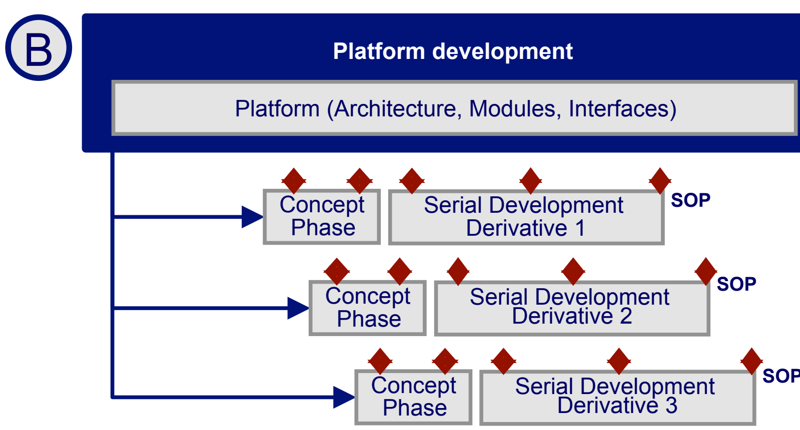


Figure 2-4: Approach B: platform development separated from derivative development

Results about the application of both approaches in industry is provided in section 3.1. In this section also approximate answers about the flexibility related to each approach are documented. In most cases, Approach B is more flexible because there are not that many functions of the lead derivative in the platform, which can be a problem in Approach A. There was also a relation indicated between the approach and the experience of a company regarding product platform development. At the beginning it can be easier to develop the platform together with a lead derivative before developing the platform first and than derive the derivatives in the second generation of the platform within a company. This approach is more similar to the single product development process used in the most companies, so it is easier for the engineers to develop a platform based on this approach.

2.2.3 Segmentation of the Market

The basic development strategy within any product family is to leverage the product platform across multiple market segments or niches. Early attempts at mapping the evolution of a product family based on extensions and upgrades to a product platform can be found in Wheelwright and Sasser (1989) and Meyer and Utterback (1993) but it was not until Meyer (1997) introduced the market segmentation grid that platform leveraging strategies were clearly articulated.

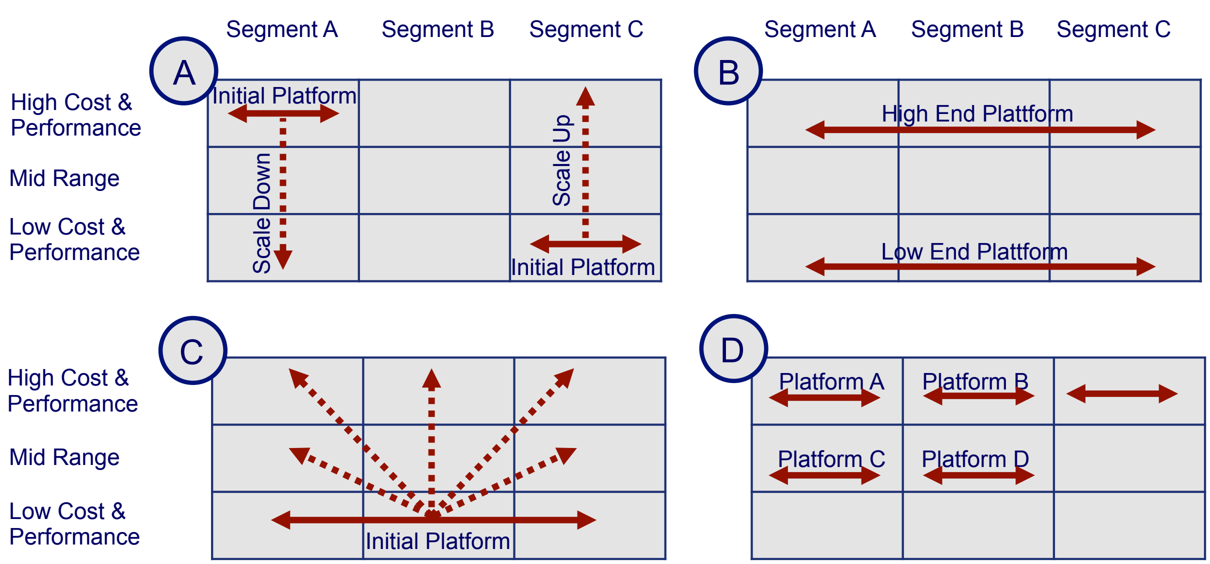


Figure 2-5: Platform market segmentation grid (Meyer and Lehnard 1997)

Market segments are plotted horizontally in the grid while cost and performance are plotted vertically; each intersection of a market segment with a cost and performance tier constitutes a market niche that is served by one or more of a company's products. Four platform leveraging strategies can be identified within the grid as shown in Figure 2-5: vertical leveraging (A), horizontal leveraging (B), the beachhead strategy (C) (which combines both), and the approach to develop a separate platform for each segment and cost and performance tier (D). Meyer and Lehnerd (1997) discuss the advantages and drawbacks of each leveraging approach, and examples of market segmentation grids can be found in Caffrey et al. (2002) for spacecraft and avionics systems and in Meyer and Lehnerd (1997) for computers, data storage systems, power tools, and office furniture. The market segmentation grid is useful for both platform development, as well as product family consolidation (Farrell and Simpson 2003).

The horizontally arrayed market segments representing each the major customer groups serviced by the products. The vertical axis of the market segmentation grid reflects different tiers of price and functional performance within a firm's markets (Meyer 2007). In case A platforms are scaled down into lower price/performance tiers; another way is to scale low-end product platforms upward. In case B product platform are leveraged from one market to the next. The beachhead strategy in case C combines horizontal leverage with vertical scaling. The company with case D has a different product platform for each market niche in which it competes, with the little sharing of subsystems and manufacturing processes between the platforms.

2.3 Uncertainty and Risk Assessment

The ultimate functional performance of a chosen platform will be uncertain at the time the development process is started but using an uncertainty and risk assessment can reduce the level of uncertainty and risk.

The process of risk assessment follows the scheme developed by McManus and Hastings (2006). The scheme says "uncertainty causes risk which is handled by mitigation ant is resulting in an outcome" (McManus and Hastings 2006).

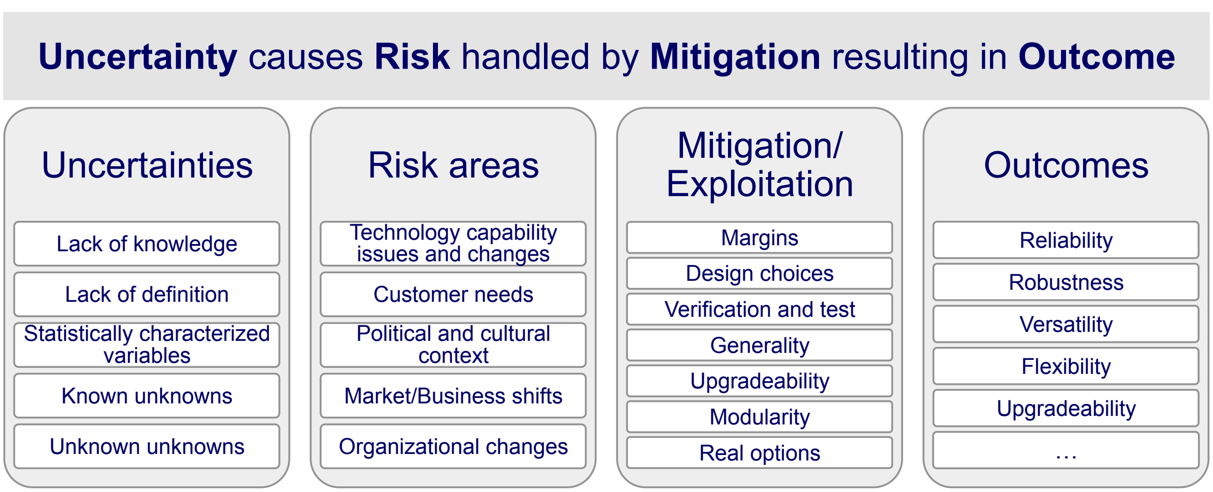


Figure 2-6: Framework for understanding uncertainty and it mitigations and exploitations (McManus and Hastings 2006)

The first step of the framework is to identify and describe types of uncertainty, followed by an analysis of different risk areas for identifying the possible hazards. The next step is to identify, investigate and implement the mechanisms for the mitigation and exploitation for the risks, which are critical for the performance. The outcomes of the framework are different "ilities", which are providing the possibility to react in case of a critical decrease of the performance because of the occurrence on an uncertainty or risk.

The mechanisms that can be used for mitigation and exploitation of the uncertainties and risks are described in section 2.3.4. The "ilities" as outcomes are described by McManus et al. (2007), Fricke and Schulz (2005), and Beesemeyer and Fulcoly (2010). The list of "ilities" is widespread and only some, which are related to the delivery of functional performance of a platform, are listed and described in this section.

- Robustness: "It is the ability of a system to maintain its level and set of specification parameters in the context of changing system external and internal forces. Robustness is determined by the sensitivity of chosen system specification parameters to context changes. A particular type of robustness, value robustness, is the ability of the system to maintain value delivery in the context of changing system external and internal forces, including stakeholder expectations. A value-robust system will be perceived to be successful to the stakeholders who continue to receive value from the system. It is important to note that value robustness can be achieved through system change or lack of system change." (McManus et al. 2007)

- Versatility: "It is the ability of a system to satisfy diverse expectations on the system without the need for changing form. It is a measure of a system's inherent, or latent, value to a possibly diverse set of expectations over time." (McManus et al. 2007)

- Flexibility: "It is the ability of a system to be changed by a system-external change agent. Flexibility is relevant because the functional performance of the platform can be increased in case of the appearance of a risk." (McManus et al. 2007)

- Adaptability: "It is the ability of a system to be changed by a system-internal change agent. The system boundary definition serves to distinguish between a flexible-type change and an adaptable-type change." (McManus et al. 2007)

- Scalability: "It is the ability of a system to change the current level of a system specification parameter." (McManus et al. 2007)

- Modifiability: "Modifiability is the ability of a system to change the current set of system specification parameters." (McManus et al. 2007)

- Changeability: "The ability of a system to change its form or function given a level of reasonableness for resources such as time, money, materials or level of effort. This "ility" is dependent on other "ilities" to describe. Flexibility and adaptability describe where the change agent is taking place to change the system's form relative to the system's boundary. Scalability and modifiability are used to describe the effect of the form change on the system." (Fricke and Schulz 2005)

- Evolvability: "The ability to change the state of the platform derivatives in a realizable subset of a possible parameter space through some regulated process of variation and selection across generations. The state is a particular set of engineering metrics that describe the platform." (Beesemeyer and Fulcoly 2010)

Table 2-1: List of techniques for the risk assessment process (ISO 2009c)

Techniques	Risk Identification	Risk analysis			Risk evaluation
		Consequences	Probability	Level of risk	
Environmental risk assessment	SA	SA	SA	SA	SA
Structure « What if? » (SWIFT)	SA	SA	SA	SA	SA
Failure mode effect analysis	SA	SA	SA	SA	SA
Reliability centered maintenance	SA	SA	SA	SA	SA
Human reliability analysis	SA	SA	SA	SA	A
Consequence/probability matrix	SA	SA	SA	SA	A
Hazard Analysis and Critical Control Points (HACCP)	SA	SA	NA	NA	SA
Cause-and-effect analysis	SA	SA	NA	NA	NA
Hazard and operability studies (HAZOP)	SA	SA	A	A	A
Scenario analysis	SA	SA	A	A	A
Brainstorming	SA	NA	NA	NA	NA
Structured or semi-structured interviews	SA	NA	NA	NA	NA
Delphi	SA	NA	NA	NA	NA
Check-lists	SA	NA	NA	NA	NA
Primary hazard analysis	SA	NA	NA	NA	NA
Root cause analysis	NA	SA	SA	SA	SA
Decision tree	NA	SA	SA	A	A
Bayesian statistics and Bayes Nets	NA	SA	NA	NA	SA
Monte Carlo simulation	NA	NA	NA	NA	SA
Bow tie analysis	NA	A	SA	SA	A
FN curves	A	SA	SA	A	SA
Risk indices	A	SA	SA	A	SA
Cause and consequence analysis	A	SA	SA	A	A
Markov analysis	A	SA	NA	NA	NA
Multi-criteria decision analysis (MeDA)	A	SA	A	SA	A
Event tree analysis	A	SA	A	A	NA
Layer protection analysis (LOPA)	A	SA	A	A	NA
Business impact analysis	A	SA	A	A	A
Cost/benefit analysis	A	SA	A	A	A
Fault tree analysis	A	NA	SA	A	A
Sneak circuit analysis	A	NA	NA	NA	NA

SA	strong applicable
A	applicable
NA	not applicable

Techniques for uncertainty and risk assessment process are listed in ISO 31010 (ISO 2009c) as shown in Table 2-1. The applicability of each technique varies from phase to phase of the risk assessment process and value of application of the techniques to the risk assessment of a special platform project has to be investigated before the usage of a technique. The different techniques are described in more detail in ISO (2009a), ISO (2009b) and ISO (2009c). The steps of uncertainty and risk identification, analysis, and evaluation are described in the following sections.

2.3.1 Uncertainty and Risk Identification

Before describing approaches of uncertainty and risk identification a definition of both terms is provided. Uncertainty defined by ISO (2009b) is the state, even partial, of deficiency of information related to, understanding or knowledge of an event, its consequence, or likelihood. Regarding to Thunnissen (2003) system engineering provides two distinct definitions/classifications for uncertainty. The first is rigorous but somewhat theoretical, the second is more relaxed but practical. The rigorous definition classifies uncertainty as either vagueness or ambiguity. Vagueness is associated with the difficulty of making sharp or precise distinctions in the world; that is, some domain of interest is vague if it cannot be delimited by sharp boundaries. Ambiguity is associated with one-to-many relations, that is, situations in which the choice between two or more alternatives is left unspecified. Ambiguity is further separated into no specificity of evidence, dissonance in evidence, and confusion in evidence (Klir and Folger 1988). The practical definition characterizes uncertainty by a distribution of outcomes with various likelihoods of both occurrence and severity. It intertwines the definition with that of risk.

Risk is defined as a measure of uncertainty of attaining a goal, objective, or requirement pertaining to technical performance, cost, and schedule. The risk level is categorized by the probability of occurrence and the consequences of occurrence and risk is classified into technical (e.g., feasibility, operability, produce ability, testability, and systems effectiveness), cost (e.g., estimates, goals), schedule (e.g., technology/material availability, technical achievements, milestones), and programmatic (e.g., resources, contractual) (INCOSE 2002). Regarding to ISO (2009b) risk is the effect of uncertainty on objectives. The effect of a risk is a deviation from the expected and it can be positive and/or negative. The affected objectives can have different aspects, such as financial, health and safety, and environmental goals. The objectives can apply at different levels, such as strategic, organization-wide, project, product and process. In most cases risk is characterized by connection to potential events and consequences, or a combination of these. Risk is often expressed in terms of a combination of the severity and the associated likelihood of occurrence.

The variety that is used in this thesis includes five types of uncertainty, which are:

- Lack of definition: Facts that are not known, or are known only imprecisely, that are needed to complete the system architecture in a rational way. This knowledge may simply need to be simply collected, or it may need to be created. It may even be unknowable, or knowable only at some time in the future. Early in development there are many of these uncertainties; they must be systematically reduced at the appropriate time. (McManus and Hastings 2006)

- Lack of knowledge: Things about the system in question that have not been decided or specified. This is not a bad thing early in a program but a current challenge is to avoid defining too much about a system too early, both in terms of defining (bad) requirement and in over-specifying the nature of the solution before any work has been done. Again, these uncertainties must be systematically reduced at the appropriate time. (McManus and Hastings 2006)

- Statistically characterized variables: Things that cannot always be known precisely, but which can be statistically characterized, or at least bounded. A strong characterization would be to know the statistical distribution of the possible values, to a known confidence level; a weaker characterization would be to know at least the bounds of the possible values. This type of uncertainty can be handled by powerful analytical techniques. Indeed, much of the science of Risk Analysis is dedicated to statistically characterizing uncertainties of various types, which may lead to risks. (McManus and Hastings 2006)

- Known unknowns: 'Known' uncertainty is based on variability in past cases. It can be characterized by probability distributions, e.g. of process task durations or the probabilities of a process (such as a computational analysis or prototype test) improving design performance. A key problem in design is the estimation of these known uncertainties in unique products and processes. Known uncertainties put limits on possibilities and describe them through probability distributions. In other cases, uncertainties may be known but their effects are unknown uncertainties in behavior. (Earl and Eckert 2005)

- Unknown unknowns: The uncertainty of surprise is an 'unknown' uncertainty in the sense that there is no particular expectation of such an event. Internal unknown uncertainties arise in the product, the process, the user or the organization itself. External unknown uncertainties come from the context in which the product or process operates, such as political events. (Earl and Eckert 2005)

In general, the types of uncertainty can be split into uncertainty of description, uncertainty of data, known uncertainty and unknown uncertainty as shown in Figure 2-7.

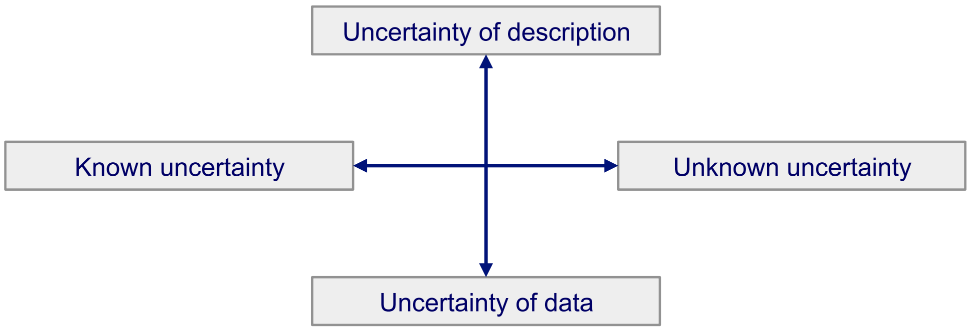


Figure 2-7: Types of uncertainty (Earl and Eckert 2005)

Also the risk is split in five risk areas, which were identified during the literature review. The identified risk areas can have a critical impact on the functional performance of a product platform.

- Technology changes: Uncertainty because of a technology change and the consequences thereof. (Thunnissen 2003)
- Technology capabilities: Uncertainty in capability of technology to provide functional performance benefits and the consequences thereof. (Thunnissen 2003)
- Customer needs/requirements: Uncertainty in change of customer needs (anticipated utility or value to the market of the chosen "design to" specifications) and in the ability of a design to meet desired quality criteria, in and the consequences thereof (Thunnissen 2003), Changing performance needs (including size, style, weight, etc.) Changing environmental conditions (temperature, humidity, vibration, etc.) New functions (due to new markets or new enabling technologies) Reliability Improvements. (Martin 1999)
- Market/Business shifts: Uncertainty in change of the market and business environment, including competition, suppliers, economic situation and the consequences thereof. (Thunnissen 2003)
- Political and cultural context (regulations, fashions, disasters): Uncertainty in political, regulatory, labor, societal (e.g. fashion), or other factors in the political environment and the consequences thereof. (de Weck and Eckert 2006)

- Organizational changes (skills, management, suppliers, etc.): Uncertainty in the organization and structure of the company (including skills of participants and roles) and the consequences thereof. (de Weck and Eckert 2006)

Based on the set of uncertainties and risks the identification of both can be performed. Regarding to ISO 31010 (ISO 2009c) an organization should identify sources of risk, areas of impacts, events (including changes in circumstances) and their causes and their potential consequences. The organization should apply risk identification tools and techniques that are suited to its objectives and capabilities, and to the risks faced. Relevant and up-to-date information is important in identifying risks and it is important to identify the risks associated with not pursuing an opportunity.

Table 2-2: List of techniques for uncertainty and risk identification, evaluated as strong applicable and applicable (ISO 2009c)

Techniques for uncertainty and risk identification	Evaluation
Environmental risk assessment	Strong applicable
Structure « What if? » (SWIFT)	
Failure mode effect analysis	
Reliability centered maintenance	
Human reliability analysis	
Consequence/probability matrix	
Hazard Analysis and Critical Control Points (HACCP)	
Cause-and-effect analysis	
Hazard and operability studies (HAZOP)	
Scenario analysis	
Brainstorming	
Structured or semi-structured interviews	
Delphi	
Check-lists	
Primary hazard analysis	
FN curves	Applicable
Risk indices	
Cause and consequence analysis	
Markov analysis	
Multi-criteria decision analysis (MeDA)	
Event tree analysis	
Layer protection analysis (LOPA)	
Business impact analysis	
Cost/benefit analysis	
Fault tree analysis	
Sneak circuit analysis	

In Table 2-2 several techniques for risk identification are listed. It depends on the platform project which ones are the best applicable ones. The description of each of these techniques is not part of this research; more information can be found, for example, in ISO (2009a), ISO (2009b) and ISO (2009c).

The goal of this step is to generate a comprehensive list of risks based on those events that might create, enhance, prevent, degrade, accelerate or delay the achievement of objectives comprehensive identification is critical, because a risk that is not identified at this stage will not be included in further analysis. Identification should include risks whether or not their source is under the control of the organization, even though the risk source or cause may not be evident. It should also consider a wide range of consequences even if the risk source or cause may not be evident. As well as identifying what might happen, it is necessary to consider possible causes and scenarios that show what consequences can occur. All significant causes and consequences should be considered.

One example of identification of uncertainties in product platforms in mentioned by Suh (2005), but it is not specified to a level that it can be applied to identify the uncertainties comprehensively. The uncertainty and risk identification in product platforms is a part of the framework that is described in Chapter 4.

2.3.2 Uncertainty and Risk Analysis

Risk analysis is an important part of developing the product platform. There are general guidelines for how it should be done, but there is no one correct way to do risk analysis (Bahill and Smith 2009).

Risk analysis involves developing an understanding of the risk and it provides an input to risk evaluation, to decisions on whether risks need to be treated, and on the most appropriate risk treatment strategies and methods. Risk analysis can also provide an input into making decisions where choices must be made and the options involve different types and levels of risk.

Risk analysis involves consideration of the causes and sources of risk, their positive and negative consequences, and the likelihood that those consequences can occur. Factors that affect consequences and likelihood should be identified. Risk is analyzed by determining consequences and their likelihood, and other attributes of the risk. An event can have multiple consequences and can affect multiple objectives. Existing controls and their effectiveness and efficiency should also be taken into account.

The way in which consequences and likelihood are expressed and the way in which these are combined to determine a level of risk should reflect the type of risk, the information available and the purpose for which the risk assessment output is to be used. These should all be consistent with the risk criteria. It is also important to consider the interdependence of different risks and their sources.

The confidence in determination of the level of risk and its sensitivity to preconditions and assumptions should be considered in the analysis, and communicated effectively to decision makers and, as appropriate, other stakeholders.

Risk analysis can be undertaken with varying degrees of detail, depending on the risk, the purpose of the analysis, and the information, data and resources available. Analysis can be qualitative, semi-quantitative or quantitative, or a combination of these, depending on the circumstances.

Consequences and their likelihood can be determined by modeling the outcomes of an event or set of events, or by extrapolation from experimental studies or from available data. Consequences can be expressed in terms of tangible and intangible impacts. In some cases, more then one numerical value or descriptor is required to specify consequences and their likelihood for different times, places, groups or situations. For making the connections between the risks clear, these can be placed on sequential trees or graphs. In these structures, the probability of a risk occurring will be dependent on predecessor risks. Also a systems engineering framework can be used, which is differentiating the risks by different framework areas (Bahill and Smith 2009).

2.3.2.1 Metrics for Uncertainty and Risk Analysis

In this research the risk equations are based on the severity of the consequences, sometimes also mentioned as impact, and the frequency of occurrence, sometimes also mentioned as likelihood. As mentioned, to analyze the uncertainty, the effect of each uncertainty to the related objectives has to be computed. Bahill and Smith (2009) mention that all of the following equations have been used in published literature to calculate risk:

$$Risk = Severity\ of\ Consequences\ x\ Frequency\ of\ Occurrence \qquad (Equation\ 2\text{-}1)$$

$$Risk = Severity\ of\ Consequences\ x\ Likelihood\ of\ Occurrence \qquad (Equation\ 2\text{-}2)$$

$$Risk = Severity\ of\ Consequences\ x\ Estimated\ Probability \qquad (Equation\ 2\text{-}3)$$

$$Risk = (Impact + Likelihood)/2 \qquad (Equation\ 2\text{-}4)$$

$$Risk = Severity + Probability\ \text{-}\ (Severity\ x\ Probability) \qquad (Equation\ 2\text{-}5)$$

$$Risk = Severity\ x\ Probability\ x\ Difficulty\ of\ Detection \qquad (Equation\ 2\text{-}6)$$

$$Risk = Severity\ x\ Severity\ x\ Probability \qquad (Equation\ 2\text{-}7)$$

$$Risk = Severity\ x\ Exposure \qquad (Equation\ 2\text{-}8)$$

For the risk analysis in the framework, Equation 2-2 is used for calculating the risk because of the uncertainties to the components and engineering metrics. As Bahill and Smith (2009) mentioned, some people have used an equation with severity plus probability of failure minus the product of severity and probability of failure. This formula does not perform satisfactorily. For example, if you set the severity to 1 (assuming a range of 0 to 1), then the probability of failure could be reduced from, say, 10^{-1} to 10^{-6} without changing the risk. Furthermore, if either the probability or the severity is zero, then the risk should be zero, but this equation does not produce this result; therefore, we do not use such an equation.

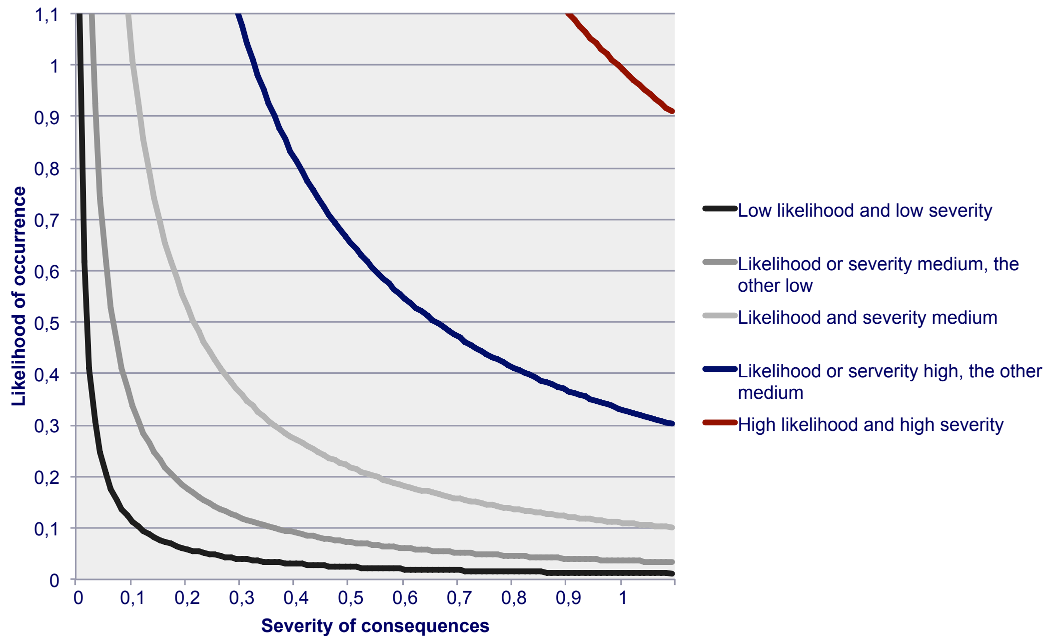


Figure 2-8: Linear plot of risk levels based on likelihood and severity

The risk plot in Figure 2-8 is showing the risk calculated with multiplying likelihood of occurrence and severity of impact. Risk plots were used in investigation of nuclear power systems (Joksimovic et al. 1977, Rasmussen 1981). Kaplan and Garrick (1981) defined them for civil engineering projects and Whitman (1984) applied them to these projects. Regarding to Bahill and Smith (2009) the risk plots are a popular method but when executing this technique it is useful to think about weak points like making failures to differentiate between levels and categories of risk or using estimated probability of the event rather than its frequency of occurrence. Also using ordinal numbers instead of cardinal numbers for severity so as the usage of different ranges for severity and frequency can be weak spots in risk analysis. Sometimes an inappropriate combining equation is used or the calculation is based on linear scales instead of logarithmic scales. Some tend to explain only intermediate risks while seeming to ignore high and low risks. Also risk interactions and severity amplifiers are sometimes ignored, and there is in some cases a confusing of risk with uncertainty.

2.3.3 Uncertainty and Risk Evaluation

The purpose of risk evaluation is to assist in making decisions, based on the outcomes of risk analysis, about which risks need treatment and the priority for treatment implementation. It is only possible to compare to similar metrics, which means it is only possible to compare the risks with a risk level set as a benchmark level.

Risk evaluation involves comparing the level of risk found during the analysis process with risk criteria established when the context was considered. Based on this comparison, the need for treatment can be considered. Decisions should take account of the wider context of the risk and include consideration of the tolerance of the risks come by parties other than the organization that benefits from the risk. Decisions should be made in accordance with legal, regulatory and other requirements. In some circumstances, the risk evaluation can lead to a decision to undertake further analysis. The risk evaluation can also lead to a decision not to treat the risk in any way other than maintaining existing controls. This decision will be influenced by the organization's risk attitude and the risk criteria that have been established.

2.3.3.1 Metrics for Evaluation of Platform Sensitivity, Monetary Aspects and Customer Satisfaction

This section covers the description of different metrics for evaluating the product platform. Some of them are related to the sensitivity and changeability of the platform others are related to the monetary value of a platform. The metrics for sensitivity used in this thesis are Commonality Index, Generational Variety and Coupling Index. Also the Change Potential Number is a metric for the calculation of the sensitivity of a platform. The goal of the sensitivity analysis is to compute value of the potential of change and variety in the platform and also to identify which parts in the platform are coupled and affected by changes in the platform. These metrics for sensitivity are a basis for the evaluation of the risk related uncertainties in the framework, which can be found in Chapter 4.

<u>Commonality Index</u>

The commonality index is a measure of how well the design utilizes standardized parts and is similar to work done by Collier (1981). A higher CI is better since it indicates that the different varieties within the product family are being achieved with fewer unique parts (Martin and Ishii 1997). Tsubone et al. (1994) pointed out the necessity to clarify two different sources of commonality, namely the component part commonality and the process commonality, for systematic studies of the product family. The idea behind component part commonality is to count the mean number of applications per component (Roque 1977). In the analysis of whether or not a product family is adequately designed, measuring component part commonality depends on more dimensions than only repetition, such as the cost or price of each component part, the volume of the final product, and the quantity per operation. The process commonality of a product family is characterized by the mean utilization of manufacturing capabilities for producing all the internally made parts and end products in the family.

$$PCI = \frac{\sum_{i=1}^{P} n_i \times f_{1i} \times f_{2i} \times f_{3i} - \sum_{i=1}^{P} \frac{1}{{n_i}^2}}{(P \times N) - \sum_{i=1}^{P} \frac{1}{{n_i}^2}} \times 100$$

Equation 2-9: Calculation of the product line commonality index (PCI) (Kota et al. 2000)

The calculation of the product line commonality index (PCI) is done with Equation 2-9 proposed by Kota et al. (2000), where P is the total number of non-differentiating components that can potentially be standardized across models, N the number of products in the product family, n_i number of products in the product family that have component i and f_{1i} the size and shape factor for component i.

Generational Variety Index

The generational variety index (GVI) is an indicator of which components are likely to change over time. The GVI is defined as “an indicator of the amount of redesign required for a component to meet the future engineering metrics” (Martin 1999).

The calculation is based on a QFD approach described in section 2.1.1. For each component/engineering metric node in the QFD Phase II matrix (see also 4.1.2), the cost for the component redesign (including design effort, tooling, and testing) required to meet the future target value for that engineering metric are estimated. These costs are expressed as a percentage of the original cost to design and represented with a scale, which is documented in Table 2-3.

Table 2-3: GVI matrix rating system (Martin 1999)

Rating	Description
9	Requires major redesign of the component (>50% of initial redesign costs)
3	Requires numerous, simple changes (<30%)
1	Requires few, minor changes (<15%)
0	No changes needed

The QFD Phase II matrix is updated after the ranking was accomplished with the input about GVI rating displayed in Table 2-3. In a next step the GVI for each component *k*, is calculated by summing the columns of the updated QFD Phase II matrix, as seen in Equation 2-10, where *j* is the indicator for the engineering metric and *n* the number of them. RGVI is out of the rating {0,1,3,9}.

$$GVI_k = \sum_{j=1}^{n} RGVI_{jk}$$

Equation 2-10: Calculating the GVI for each component (Martin 1999)

All in all the GVI is a measure to specify the amount of redesign required for a component to meet the engineering metrics and so to deliver the expected functional performance. Together with the coupling index, which is described next, it is an important part of the design for variety method developed from Martin (1999)

<u>Coupling Index</u>

As mentioned, the coupling index (CI) is important in the design for variety method. There are various external drivers for changing a design. The changes created by these external drivers may in turn require other changes within the design. Such changes do not directly enhance the value of the product, except to the extent that these support the initial changes. These changes are created by the interaction, or ‘coupling’, within the design. It is crucial to understand the coupling within a design for the development of product platforms that are robust to future changes in the context.

The CI is defined as “the indicator of the strength of coupling between the components in a product. The stronger the coupling between components, the more likely a change in one will require a change in the other” (Martin 1999).

Ulrich (1995) asserts that two components are considered coupled if a change made to one of the components can require the other component to change.

The calculation of the coupling index defined by Martin (1999) is based on an enhanced coupling matrix. The coupling matrix is enhanced with specification flows at each coupling node between two components. The next step is the estimation of the sensitivity of each component to changes. If a large change is needed in a component because of a small change in the specifications, than the sensitivity is high. If the specification can be changed widely without creating a change in the component, the sensitivity is low. One intermediate level is also used and the description of all three ratings is listed in Table 2-4. For this rating system, the assumption is that the impact caused by a chance in a specification is equivalent and linear across all components of a platform.

Table 2-4: CI rating system for sensitivity of specifications (Martin 1999)

Rating	Description
9	Small change in specification impacts the receiving component (high sensitivity)
3	Medium sensitivity
1	Large change in specification impacts the receiving component (low sensitivity)
0	No specifications affecting component

Developing of the coupling index is done by considering the specification flows among components. These specification flows are defined as the design information that must be passed between designers to design their respective components. By mapping out the specification flows early in the design process, the team explicitly describes the relationships that couple the parts. (Martin and Ishii 2002).

The coupling index is split in two types. The coupling index – receiving (CI-R) indicates how strong the specifications are that one component is receiving from another on. The coupling index – supplying (CI-S) describes how strong the specifications are that one component is supplying to another. The calculation of CI-R is done with Equation 2-11, and the calculation of CI-S with Equation 2-12, both are based on Equation 2-13.

$$(CI - R)_a = \sum_{b=1}^{m} TS_{ab}$$

Equation 2-11: Calculation of CI for receiving specification (Martin 1999)

$$(CI - S)_b = \sum_{a=1}^{m} TS_{ab}$$

Equation 2-12: Calculation of CI for sending spefification (Martin 1999)

$$TS_{ab} = \sum_{c=1}^{m} SS_{abc}$$

Equation 2-13: Calculation of total sensitivity of component a to a change in b (Martin 1999)

In the three equations, a is the component which is receiving specifications, b is the component that is supplying specification, c is the specification and m describes the number of components. TS describes the total sensitivity of component a to changes in b.

Change Potential Number

Rajan et al. (2003) developed a metric that is based on possible change scenarios. Rajan et al. (2003) propose potential change modes and estimate the readiness of the company to deal with the change as well as flexibility of the platform. In addition, they estimate how often or how likely the change is to occur (Simpson et al. 2006).

$$CPN = \frac{10}{N}\sum_{i=1}^{N}\frac{(R_i + F_i) - O_i + 8}{27}$$

Equation 2-14: Calculation of the change potential number (CPN) (Rajan et al. 2003)

Equation 2-14 documents the calculation of the change potential number (CPN), where *N* is the maximum of total change modes, total potential effects of change, or total causes of change, *R* describes the readiness on a 1-10 scale, where 10 means being completely prepared. The flexibility is described with *F*, also on a 1-10 scale, where 10 is no redesign and 1 means new part. *O* is the probability of occurrence, in this case number of times in every 10 years.

The CPN values are calculated for each product, which are then averaged to get the platform score. The minimum value that the CPN can hold is ‘0’ which means that the platform is completely inflexible for any change and ‘1’ means that the platform is completely flexible for any future change. Based on this formula a completely flexible platform is, which the redesign cost incurred is \$0 for any future change in the design.

Platform Efficiency and Effectiveness

Platform effectiveness measures the ratio of the revenue a product platform and its derivatives create to the cost required to develop them. Platform efficiency assesses how much it costs to develop derivative products relative to how much it costs to develop the product platform within the product family. This approach was developed by Meyer et al. (1997).

$$Platform\ Efficiency = \frac{R\&D\ Costs\ for\ Derivative\ Product}{R\&D\ Costs\ for\ Platfoom\ Version}$$

Equation 2-15: Calculation of the platfrom efficience (Meyer et al. 1997)

$$\bar{E}_{p,v} = \frac{\frac{1}{N_f}\sum_{f=1}^{N_f} C'_{p,v,f}}{C'_{p,v,base}}$$

Equation 2-16: Average platform efficiency for a generation of a product family (Meyer et al. 1997)

The scheme of computing the platform efficiency is displayed in Equation 2-15 and more detailed in Equation 2-16, where E is the efficiency, C are the costs attributable to a platform or derivative product within a product family, p is the platform index, f the derivative product index, N_f the number of derivatives of a platform and v the platform version index.

$$L_{p,v} = \frac{\sum_{f=base}^{N_f} S'_{p,v,f}}{\sum_{f=base}^{N_f} C'_{p,v,f}}$$

Equation 2-17: Calculation of platform effectiveness (Meyer et al. 1997)

The effectiveness of the platform L considers R&D returns as accumulated profits divided by development costs and can be calculated with Equation 2-17, where S is the sales attributable to a platform or derivative product within a product family.

A similar approach is taken in Schellhammer and Karandikar (2001), wherein a project ranking index, which combines an investment index and a revenue index, is introduced to assist in project planning.

Customer Needs

Simpson et al. (2006) defined that the customer need metric measures the fulfillment of the customer needs provided by the platform products. It is an important metric; a failure here implies the platform doesn't satisfy the customers with the result that the company would not reach their sales volume in the targeted market. Simpson et al. (2006) proposed a comparison of the derivative's ideal target on each critical requirement to what the platform can actually provide.

$$Y_{CR} = \frac{1}{M} \sum_{variants\ i} \frac{1}{K} \sum_{requirements\ j} w_{ij} R_{ij}$$

Equation 2-18: Score of customer needs (Simpson et al. 2006)

In Equation 2-18 the calculation of the score of the customer needs by Simpson et a. (2006) is documented, where w_{ij} is the revenue weighted importance requirement j for product i; R_{ij} is the score for a customer requirement j for a product i on a 0-10 scale, reflected the gap from its target; K is the number of requirements; and M is the number of variants. The customer score, R, can be computed, for example, by a comparison of the achieved level of a customer need to the range between the target and the starting level of that customer need. In Equation 2-18 the calculation for the whole family is presumed.

A comparable approach is used in the framework in Chapter 4 to calculate the reached fulfillment of the engineering metrics, which is shown in Equation 4-7.

2.3.4 Uncertainty and Risk Treatment

Having a complete uncertainty and risk assessment, risk treatment involves selecting and agreeing to one or more relevant mechanisms for changing the probability of occurrence, the effect of risks, or both, and implementing these options. According to ISO (2009a), uncertainty and risk treatment involves selecting one or more mechanisms for modifying risks, and implementing those mechanisms. If the mechanisms are once implemented, treatments provide or modify the controls. ISO (2009a) describes uncertainty and risk treatment as a cyclical process. The first step is assessing a mechanism for uncertainty and risk treatment; the second one is about deciding whether residual uncertainty and risk levels are tolerable. If they are not tolerable, a new risk treatment is to generate; and the effectiveness of that treatment is to assess.

Risk treatment options are not necessarily mutually exclusive or appropriate in all circumstances. The options can include the following:

- Avoiding the risk by deciding not to start or continue with the activity that gives rise to the risk.
- Taking or increasing the risk in order to pursue an opportunity.
- Removing the risk source.
- Changing the likelihood.
- Changing the consequences.
- Sharing the risk with another party or parties (including contracts and risk financing).
- And retaining the risk by informed decision.

According to de Neufville (2004) the typology of uncertainty and risk treatment focuses on the time scale at which engineers and managers might choose to manage uncertainty. These would range from the shortest to the longest terms. Thus, it might be reasonable to think about decisions that are operational, tactical, and strategic risk treatment methods.

De Neufville (2004) describes the different responses for uncertainty as followed:

- Reducing uncertainty: Aim to increase knowledge about the system and its environment (e.g. collect more data, performing additional analyses, partitioning of the system).

- Protecting system actively: Design the system capable to adapt itself to effectively deal with unknowns, i.e. that it is capable of adapting itself to deliver acceptable results despite changes in the results themselves.

- Protecting a system passively: Design the system capable to withstand the influence of uncertainty without the need to change its structure or basic mode of operation during operation.

Combining the possible time scales and modes of response leads to the two-way typology for managing uncertainty in engineering systems, reducing uncertainty and risk and protecting the system.

The approach in this research to treat the risks is to implement mechanisms in the platform for mitigating uncertainties and risk. Regarding to Ross (2006) a mechanism is defined as the set of actions, decisions or designs that enable a change in the platform. A further distinction can be made between active and passive mechanisms. An active mechanism is defined as a mechanism that directly enables a change. A passive mechanism is defined as a mechanism that indirectly enables a change.

The path of implementing a mechanism that is taken in this research is based on a certain level for getting a value robust platform. The goal of implementing a mechanism is to minimize the severity of an uncertainty caused risk with the result to deliver functional performance over the lifecycle. The scheme of implementing a mechanism is shown in Figure 2-9. There are uncertainties in the context cloud, which can cause in a risk anytime during the lifecycle. Based on the uncertainty and risk analysis and evaluation, mechanisms are implemented to have the possibility to change the platform that it can handle the impact of the risk and deliver the context expected performance.

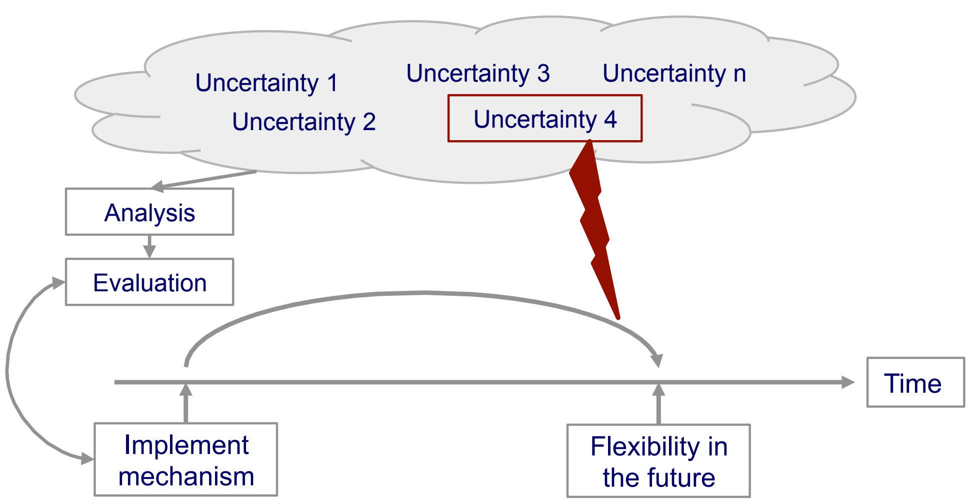


Figure 2-9: Scheme of implementing a mechanism (adapted from Mikaelian et al. 2009)

In Figure 2-9 the timeline of implementing a mechanism is described but an important part of this approach is to describe the impact of a mechanism to the state of a system and the change event.

According to Ross, Rhodes and Hastings (2008) a change event can be characterized with three elements. The first element is the agent of change, which is the instigator, or force, for the change. The agent can have an intentional or implied role, but it always requires the ability to set a change in motion. The second element is the mechanism of change; it describes the path, which is taken to succeed from state 1 to state 2. It includes any costs, time and money related to the mechanism. The last element is the effect of change. The effect of change is the actual difference between the origin and destination states.

An illustrative example is also provided by Ross, Rhodes and Hastings (2008). They used a system that is black in time period one and gray in time period two, so it has changed its color. In this case, the change agent can be Nature, for example, by imparting physical erosion due to weather effects like wind, water, or sun, or it can be a person with paint can and brush. The change mechanism can be the erosion or painting process. The erosion process is costing no money, but taking a long time and the painting is costing some amount of money, but it normally can be done faster than the erosion. In Figure 2-10 the scheme is shown by multiple system changes depicted using the agents, mechanisms and effects as explained above.

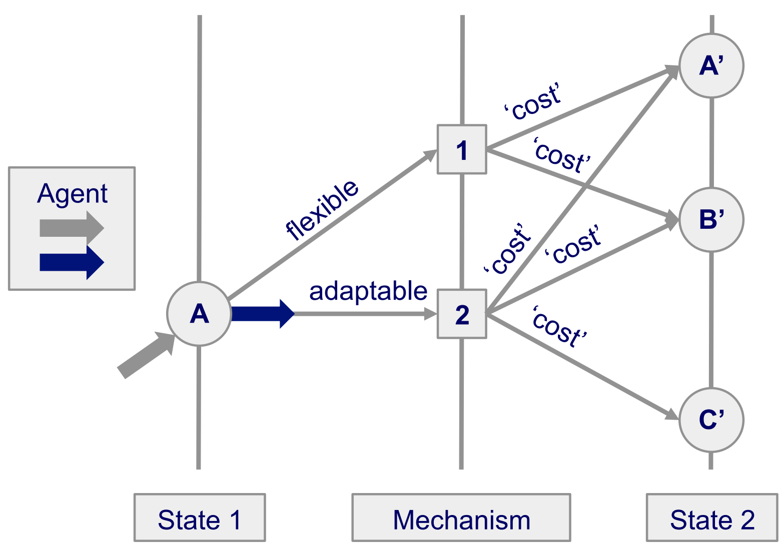


Figure 2-10: Scheme of state changes by mechanisms (Ross, Rhodes, Hastings 2008)

The types of mechanisms are described in this thesis are margins, modularity and real options. This is not the complete set of choices for mechanisms and more can be found in further literature.

There are many difficulties facing the analysis of mechanisms, some more are listed by Wang (2005):

- What is the best implementation amount or rate per mechanism?
- What is the risk level for start investigations regarding mechanism?
- How to calculate the costs of a mechanism and what is the NPV of the mechanisms?
- What is the tradeoff between implementing a mechanism for treating a risk and having no possibility to react on a decrease of functional performance?
- How to set the indicators for exercising a mechanism?

An intensive analysis and knowledge is necessary before implementing mechanisms to treat the risk, because there can be disadvantages, at least monetary, if the mechanisms are implemented without delivering value.

Margins

McManus and Hastings (2005) defining margins as a mechanism for designing a platform to be more capable, to withstand worse environments, and to last longer than "necessary". This is more like an imbedded mechanism for change. The change agent can range the state up an down between the lower and upper restriction of a margin to deliver performance until the gap is bigger than provided by the implemented margin.

Modularization

Modularity definitions and methods depend on the purpose of modularity. For example, at what point are the benefits of modularity wanted – during the design phase (design reuse etc.) or at the end of life of a product (recycling etc.) or during the other phases of the lifecycle.

Otto (2005) describes modularity as an approach to put flexibility within a system. Modularity is a concept that has proved useful in a large number of fields that deal with complex systems. Based on the given definition of a module in section 2.2.1, the idea behind this approach is that functions are grouped into modules and connected by standard interfaces in such a way that they can "plug and play".

The mechanism of modularization gives the opportunity to replace modules to reach another state of the system, which is better concerning the performance expected by the context. It can be during each phase of the lifecycle, as mentioned earlier.

Real Options

De Neufville et al. (2004) said that engineers increasingly recognize the great value of real options in addressing several uncertainties facing large-scale engineering systems and, more importantly, are learning to manage the uncertainties proactively.

Regarding to Wang (2005), "real options 'on' projects are financial options taken on technical things, treating technology itself as a 'black box'. Real options 'in' projects, which are the interesting ones in this research, are options created by changing the actual design of the technical system. These create options by design of technical system."

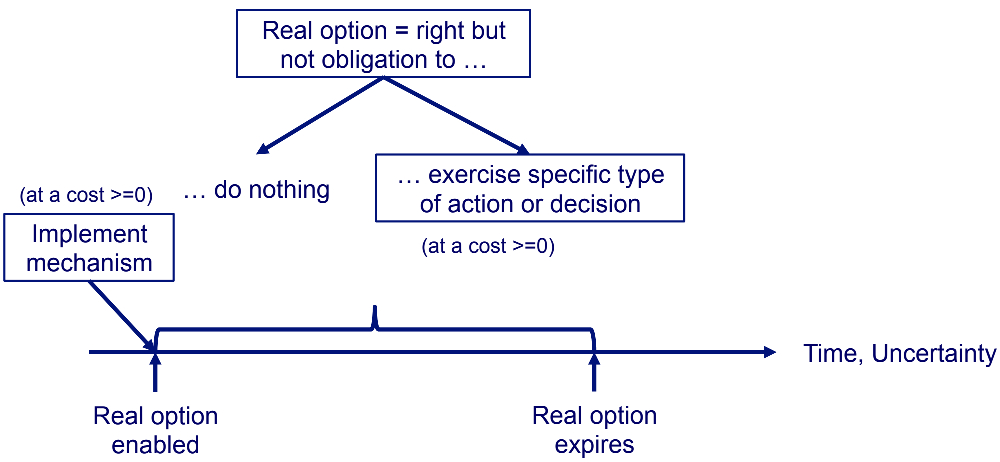


Figure 2-11: Scheme of implementing a Real Option (Mikaelian et al. 2009)

As shown in Figure 2-11, a real option is a right, but not an obligation to take some action now or in the future for a pre-determined condition. Some characteristics are that it is only to exercise, if the output of exercising it is advantageous to the platform. For example, if the functional performance of the platform can be hold stable or can be increased by exercising the real option. Another characteristic is that there is an asymmetric return if the real option is exercised. It was implemented in the development phase of the platform, so there are usually only how costs or efforts to enhance the existing platform, there can be costs for developing a component that is been adding to the platform. An example is the implementation of a standardized interface and adoptable software for multimedia devices in a multimedia platform in cars. If it is fully implemented with taking everything into account, the manufacturer has the right during the whole lifecycle to replace the implemented CD player with the newest MP3 player without any costs for putting a new cable in the architecture of the car. The only occurring costs are the costs for buying and assembling the MP3 player or updating the software with specifications that couldn't have been predicable. This example can be also considered as a multi level real option if it is split into a real option to replace the CD player and the real option to enhance or change the software for using the interface for the multimedia device. The example shows that it is necessary to draw a borderline around what is included in a real option. De Neufville (2010) separates between call and put options. Call options are the right to take advantage of an opportunity (e.g., ability to expand the data storage if the demand is high) and a put option is the right to limit losses of a bad situation (which is what an insurance policy provides).

In summary, real options embody formal concepts of flexibility and can add value to the system. Real options are not "alternatives" and the return of a Real Option is asymmetric, it is a source of value for increasing the functional performance.

2.4 Chapter Summary

The in Chapter 2 documented information is used in the following parts of this research. The definition of the platform as a system is the basis for the empirical case study described in Chapter 3. It is also used for developing the framework for investigating the management of uncertainty in Chapter 4. The descriptions of risk areas and types of uncertainties were a fundamental part of the empirical case study and these were also used for developing the framework and especially for the application of it in Chapter 5 to the illustrative iRobot case.

The review of the literature has shown that there was certain research done concerning metrics for evaluating the design of a product platform. The ones, which are most significant for this thesis are documented in this chapter, and used, some of them in a modified version, in the framework in Chapter 4.

This chapter also provides an overview of an assessment process for uncertainties and risk. First the way of identifying uncertainties is described, then a description of uncertainty and risk analysis together with some techniques for doing it are documented. Both steps, as well as uncertainty and risk evaluation and treatment methods, which are also described in Chapter 2, are an important part of steps 3 and 4 of the framework described in Chapter 4.

3 Empirical Case Study: Identification of Uncertainties in Product Platform Lifecycles

To answer the first research question "Where do uncertainties occur in the platform lifecycle and what is the impact on the functional platform performance and the related products?" a case study with several participants from various companies was performed. The definition of a platform described in section 2.2.1 and the description of uncertainties, risk areas, and mitigation methods described in section 2.3.4 were the basis for this empirical case study.

3.1 Approach Description and Introduction of Participants

Based on the methods provided by Yin (2009) and Babbie (2007), a knowledge gathering instrument for face-to-face and phone interviews was developed for the empirical case study, and is documented in Appendix A. The goal was gathering knowledge concerning critical uncertainties in lifecycles of various product platform projects, which were reducing the functional performance of the platform during the lifecycle or killed a platform project. The questions were organized in three different sections, the empirical case study asked in the first section open questions about the role of the participant within the company, about the company itself and the investigated platform project. The second section of the empirical case study was a structured part to gather information regarding the risk areas and the impact onto the architecture, modules and interfaces of the product platform. Also data about the types of uncertainties, which caused problems in the different risk areas, were collected, as well as data about the application of mitigation methods connected to the different risk areas. In the second section also a mapping of the identified problems in the different risk areas to the phases of the lifecycle of a product platform was conducted. The third section of the empirical case study gathered data to compare the overall platform performance of the participating companies (e.g. top vs. low performer). The second part was related to the functional performance of the platform projects, because the focus of this thesis is on managing uncertainties to deliver the expected functional performance over the lifecycle of the product platform. To measure the robustness of the overall platform performance, domains like production or supply chain have to be taken into account. This would have been too extensive for one empirical case study so approximated data were gathered to get a general survey over domains like cost savings in production, development and maintenance, re-use rate within the derivatives, time saving aspects or variety of products on the market because of the product platform.

In total ten participants out of automotive, electronics, agriculture machinery and defense industry were interviewed for the empirical case study, the investigated variety of platform project reach from car platforms on the level with the highest complexity to platforms for processors on the lower end of the complexity scale. The participants and companies are spread all over the world, some are in the US others are in Asia (India) or Europe (Germany, Switzerland).

All of the participants were anywhere between a lead engineer and senior project director, so every participant has several years of working experience regarding platform development. Most of the companies have one platform, which they use for deriving up to 30 derivatives per platform. The number of platforms is dependent on the level in the system, for example, some using a platform on product level and others using the platform on component level. It is also dependent to the complexity of the product. The reason, which was mentioned why they have only one platform in most cases, is the complexity of the development process. They would like to handle more different platforms, but they can't because of cost reasons. The experience of the companies regarding platforms differs between one and 25 years and most of them started with the same development approach for developing the first platform. Most of the companies used Approach A, which is described in section 2.2.2. Most of the companies started to develop the platform together with a lead derivative. During the platform development and lifecycle they realized that this approach was inflexible, because, for example, too many specifications of the lead derivative were in the platform if this approach was chosen and it was not so easy to respond on changes because of uncertainties in the different risk areas. Some of the participants switched to Approach B for the second platform project, where they developed the platform first with taking all the specifications of the derivatives into account.

One detail all participants have in common is that they are using a module-based platform approach. Module-based means that the platform is based on a variety of modules from which a stream of derivatives can be build and it is not necessary that all modules of the platform are in a derivative. Scale-based means that number, type and the function of modules in each derivative is identical but the size of the modules is different. Some of the participants mentioned, that they used scaling within the modules.

Asked about the decision-making in the platform development process the participants provided a variety of strategies. In most of the investigated cases the decisions were not taken because of technical requirements but on requirements provided by the marketing department within a company. The goal in most cases was to build a platform, which fits for all derivatives and which is satisfying the requirements of all customers and markets. In some cases, the decisions are based on layout limitations and government regulations. Some other companies are using a more systemized approach and combining top down with bottom up, like a function driven top down and a CAD model based concept-modeling bottom up approach for complexity reduction. This process is run in several loops until a satisfying solution is reached. Also economies of scale (potential commonality, carryover parts from previous generation and from other products), growth potential in markets, modularity concerning benefits in the supply chain, potential parts sharing with cooperation partners and repair and maintenance concepts are influence factors in decision making. In some companies the senior chief engineer, based on the opinion of the development engineers, makes the decisions about product platform architecture. It is a subjective multi attribute decision with input from all stakeholders.

The data collected about the leveraging and vertical market segmentation of the platforms pointed out that some companies went the path from big to small, which means that they developed the high-end derivative first and scaled down the platform later; others went the path the other way round. There was no trend detected, which approach is more successful, and the decision of which strategy was taken was more related to the complexity of the product platform and the targeted market. Both market-leveraging strategies had in common that lessons learned from the first derivatives were used and applied to the later derivatives. Another approach that was discovered in the empirical case study was that companies tried to develop the platform around a core element, for example, a company had developed a computer chip, which covers important core functions of the product portfolio. They developed the core component before they decided to switch to a platform approach; the platform was developed around this component afterwards. The result was that the company had to struggle with a bottleneck, because the capability of an interface was limited and not enough for all later on planned derivatives.

3.2 Investigation of Impact, Consequences and Mitigation of Uncertainty

As mentioned the second section of the empirical case study was a structured part to gather data regarding the risk areas and the impact onto the different parts of the product platform, the types of uncertainties, which caused problems in the different risk areas, as well as data about the application of mitigation methods connected to the different risk areas.

3.2.1 Identified Problems and Mapping to the Phases of the Lifecycle

In this section the gathered data about the identified problems in the lifecycle of the investigated product platform projects are documented. The results are not discussed in the same order as gathered in the empirical case study, for example this sections covers the analysis of the data gathered with questions 6 and 7 of the in Appendix A documented questionnaire.

First of all, the mapping of the occurred problems, which were followed by a needed change in the related product platform project, is illustrated in Figure 3-1. In this figure, the reasons are mapped to the different phases of the lifecycle of a platform.

		Concept definition	Design Phase	Development and Fabrication	Integration & Verification	Production	Use	Retirement
Technology changes and capability issues	Major technology changes	◆			◆	◆	◆	◆
	Limitations of technologies		◆		◆			
	Predicted technology performance failed	◆	◆	◆	◆			
	Technical failure reports	◆	◆	◆	◆	◆	◆	◆
	Cost pressure		◆	◆				
	Manufacturing problems (design for manufacturing; serialization problems)			◆		◆		
Changes in customer needs	Need change of critical customer		◆		◆	◆	◆	◆
	Unstable customer requirements	◆	◆					
	Feedback from marketing & sales	◆						
	New requirements from expanding markets & applications				◆			
	Lack of definition of customer requirements				◆			
	Direct customer feedback						◆	
Market and business shifts	Change of supplier	◆	◆		◆			
	Competitors launched new/different concepts	◆						
	End of lifecycle of component purchased from supplier			◆	◆	◆		
	New/additional competitor			◆				
	Inaccurate market forecast						◆	
	Obsolescence of supplier						◆	
Political and cultural changes	New/change of regulation		◆	◆	◆	◆	◆	◆
	Doctrin changes						◆	
	Lack of knowledge about cultural context				◆	◆		
Organizational changes	Change of internal personal organization	◆	◆	◆		◆	◆	◆
	Change of product strategy					◆	◆	
	Closure of internal department	◆	◆		◆			
	Lack of knowledge about internal components and competences	◆	◆					

Figure 3-1: Mapping of uncertainty related reasons to the phases of a lifecycle (results of empirical case study question no. 7)

The lifecycle is separated into a development period and the period after start of production (SOP), as described in section 2.2.2. The development period is split into concept definition, design phase, development and fabrication, and integration and verification. The period after SOP is split in production, use and retirement phases. The displayed reasons are a summarization of all identified problems, which occurred during the lifecycle of the investigated product platform projects.

For example, in some of the investigated cases, problems like reaching of the end of lifecycle of a purchased component occurred late in the development phase or in phases after the start of production.

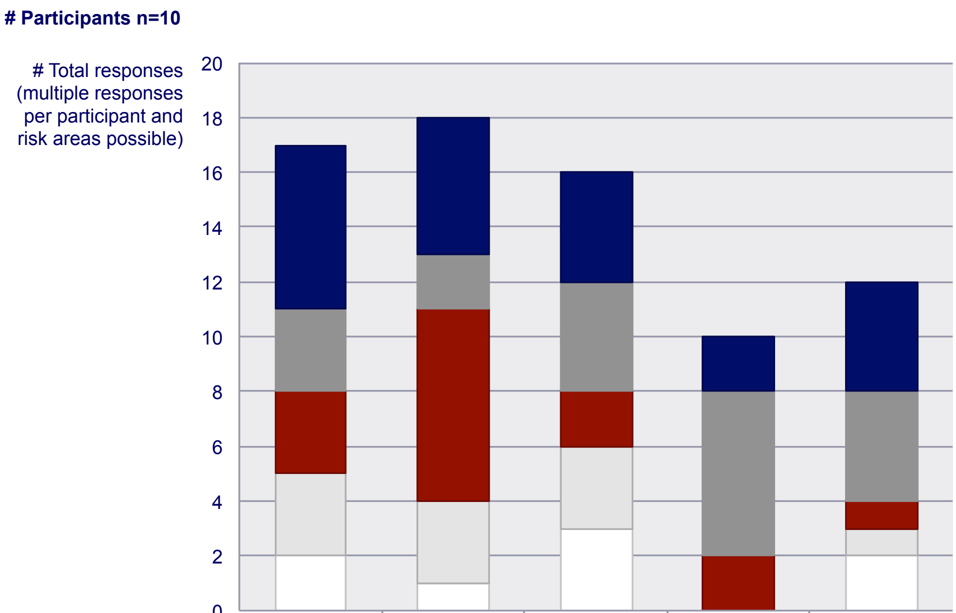
The consequences of these problems can be very critical, especially if the platform is not that flexible, or if there are market and business limitations like single sourcing. Problems like this can be mitigated by gathering more knowledge in all risk areas, which could have an impact on the performance of the product platform.

One reason why it was not possible to build a platform that can handle all the mentioned problems was that the customer requirements and behavior was different than assumed. Another reason was that the company decided to enter new markets with their products but they didn't think about it while designing the platform. The consequence was that it was not possible to meet all customer requirements without changing the platform with huge effort. Also mistakes in development process were made regarding internal decisions, for example, the organization decided to switch the operation system on the platform so it was indispensable to need a new design of nearly the entire platform. Furthermore, the necessary data about the customer needs were not available during the development of the product platform with the consequence that the platform didn't hit the requirements. In some cases not all specifications were defined in bids or the specifications changed over time because of different reasons and also the development of the lead derivative together with the product platform also caused problems in the product platform.

An additional result is that most of the problems were indicated in the integration and verification phase; that means that the uncertainties and risks were often identified late in the development process. There were not many reasons mentioned for platform changes during retirement, a reason can be that it is not efficient to update a platform in retirement phase because the platform is no longer in production. More interesting can be the cross-reference to the following platform project if it can be the possibility that this platform is facing the same problems in its retirement phase.

A fact was mentioned several times; the specifications of the platform were defined and locked too early. Furthermore, most of the considered product platforms were not designed to handle all the changes, which can occur in the risk areas, but the companies tried to adopt it to new use cases. One participant mentioned that it was like using a car chassis for designing a truck. Critical problems during development were cost and schedule issues, but these problems can be seen as a result of problems in the different risk areas.

In Figure 3-2 the number of occurred problems in the different risk areas related to the types of uncertainty are illustrated. The number of total responses in this figure is higher per risk area than the number of total participants, because more responses per risk area were possible.



	Technology Capabilities/ Changes	Customer Needs	Market/ Business shifts	Political/ Cultural Context	Organizational Changes	
Known Unknowns	6	5	4	2	4	∑ = 21
Unknown Unknowns	3	2	4	6	4	∑ = 19
Lack of Definition	3	7	2	2	1	∑ =15
Lack of Knowledge	3	3	3	0	1	∑ =10
Statistically characterized variables	2	1	3	0	2	∑ = 8
	∑ = 17	∑ = 18	∑ = 16	∑ = 10	∑ = 12	

Figure 3-2: Uncertainties that affected the risk areas

The gathered data have shown that in the considered cases known unknowns is the type of uncertainty which led to the most problems in the risk areas, followed by unknown unknowns, lack of definition, lack of knowledge and statistically characterized variables. Excepting unknown unknowns the impact of all types of uncertainties are dependent to the knowledge which is gathered and the likelihood can be reduced by gathering more knowledge at the frontend of a development project.

The most affected risk areas were customer needs, followed by technology capabilities and changes, market and business shifts, organizational changes and political and cultural changes. It seems that the pattern of technology changes and capability issues, customer needs and market and business shifts are more affected than the pattern of political and cultural changes and risks in the organizational context.

Known unknowns dominated risks in the area of technology changes and capabilities, followed by unknown unknowns, lack of definition and lack of knowledge with the identical number of responses and statistically characterized variables at the end. In the technology changes and capability risk area the problems were, for example, wrong predicted technology capabilities or too little data about the technology at the beginning of the platform development. In some cases the decision, which technology is implemented in the products, was taken too optimistically. This means that they implemented a technology that is relatively new but with a low technology readiness level but it sounded good to have the new technologies in the product. The effect was even worse when the amount of taking decisions without having detailed knowledge was increasing, because there were too many new technologies with too less readiness in the derivatives.

Lack of definition dominated the risk area customer needs; it can signify that the companies developed their products without listen enough to their customers. In some cases the customer requirements were not documented with sufficient detail. Lack of definition is followed by known unknowns, lack of knowledge, unknown unknowns and at the end statistically characterized variables.

Known unknowns dominated risks in market and business shifts and unknown unknowns identically, followed by lack of definition and statistically characterized variables with the same number of responses and lack of definition at the lower end. An unknown unknown in this area was, for example, that a supplier of a module quit because of certain reasons. Lack of definitions led to problems with components, which were outsourced and not manufactured or developed within the company. Some participants had the problem that a component they purchased was at the end of its lifecycle before the platform was at the end of the lifecycle.

Risks in political and cultural context were dominated by unknown unknowns, followed by known unknowns and lack of definition with the same number of responses. The most problems in this area were related to new regulations, which were released without having any knowledge about it. It can be that the problems regarding to political changes are dependent on the market and country. An assumption is that the likelihood of new and unknown regulations are released is higher in countries with less political transparency like China and lower in countries with more transparency in politics like the US or Europe Union. There were no problems mentioned in the empirical case study related to political or cultural changes because of lack of knowledge or statistically characterized variables.

Risks because of organizational changes were dominated by known and unknown unknowns identically, followed by statistically characterized variables and at the lower end lack of definition and lack of knowledge with the same amount of responses. Problems because of unknown unknowns in this area were related, for example, to an unknown change of the product strategy during the use phase of the platform. In some cases risks occurred because of closing or moving internal departments or development and production locations, in some cases these were known unknowns and in some others the changes were unknown unknowns.

3.2.2 Impact of the Identified Uncertainties to Components of a Platform

In this section, the analysis of the gathered data regarding the consequences of risks in the different risk areas is documented. These are the data gathered with questions 5 of the knowledge gathering instrument described in Appendix A.

In Figure 3-3 the consequences in the components of a platform because of the impact of risks in the different areas, measured on a scale from 'no consequences' to 'platform project killed', are illustrated. The consequences are measured in cost over budget per component in percentage and time delay per component in percentage and more detailed table is provided in Table 3-1.

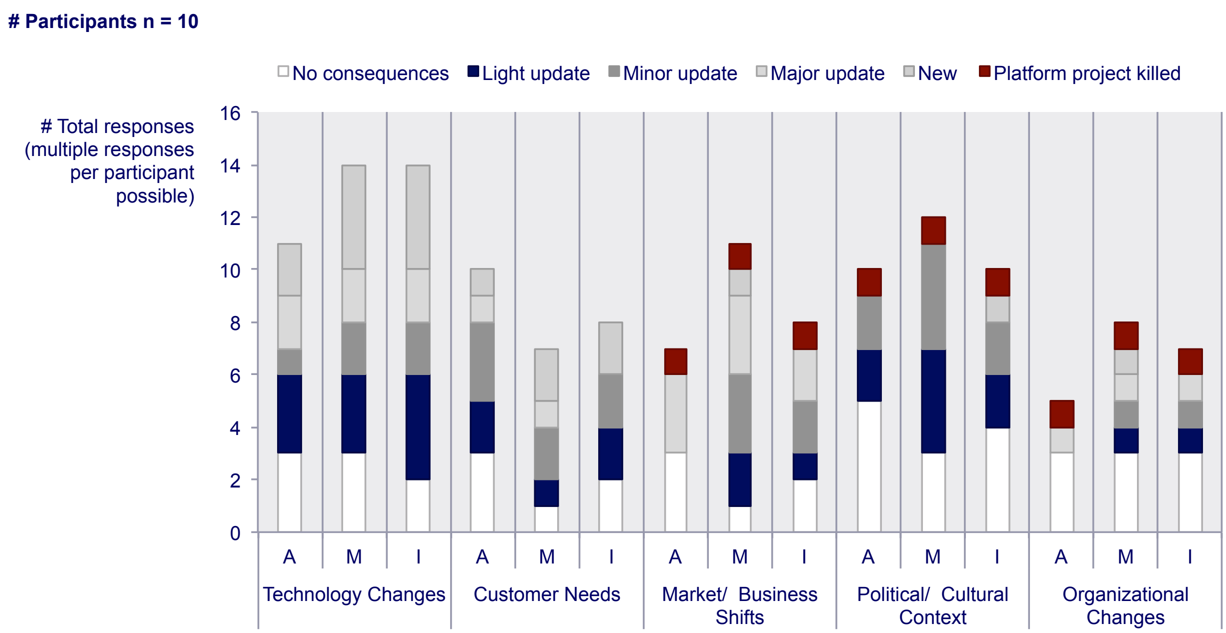


Figure 3-3: Consequences in the platform because of problems in the risk areas

In Figure 3-3 the components of the platform are separated and indicated with *A* for architecture, *M* for modules and *I* for interfaces. Most of the consequences were indicated in the risk area of technology changes and capability issues followed by consequences in the platform components because of problems related to risks in the area of political and cultural changes. A result was that none of the platform projects were killed because of technology related issues or changes in customer needs. Platform-killers were risks in market or business context (e.g. a significant supplier quit), political context (e.g. a new regulation was released) and organizational context (e.g. an internal design center was closed). New parts were often developed as a consequence of the impact of risks because of technology changes and capabilities issues, followed by impacts of risks in customer needs. A major update of the components of a platform was necessary most frequently because of risks in the market and business context, followed by risks related to technology changes or capability issues, followed by risks related to organizational changes. Minor updates of the architecture, modules or interfaces were indicated most frequently, because of risks in political and cultural context, followed by risks related to technology changes or capability issues, followed by risks related to organizational changes. The last category, light updates, were indicated most frequently because of risks in the technology context, followed by risks in political and cultural context, followed by risks because of changes in customer needs.

Table 3-1: Scale for measuring the consequnces in the platform components because of the risks

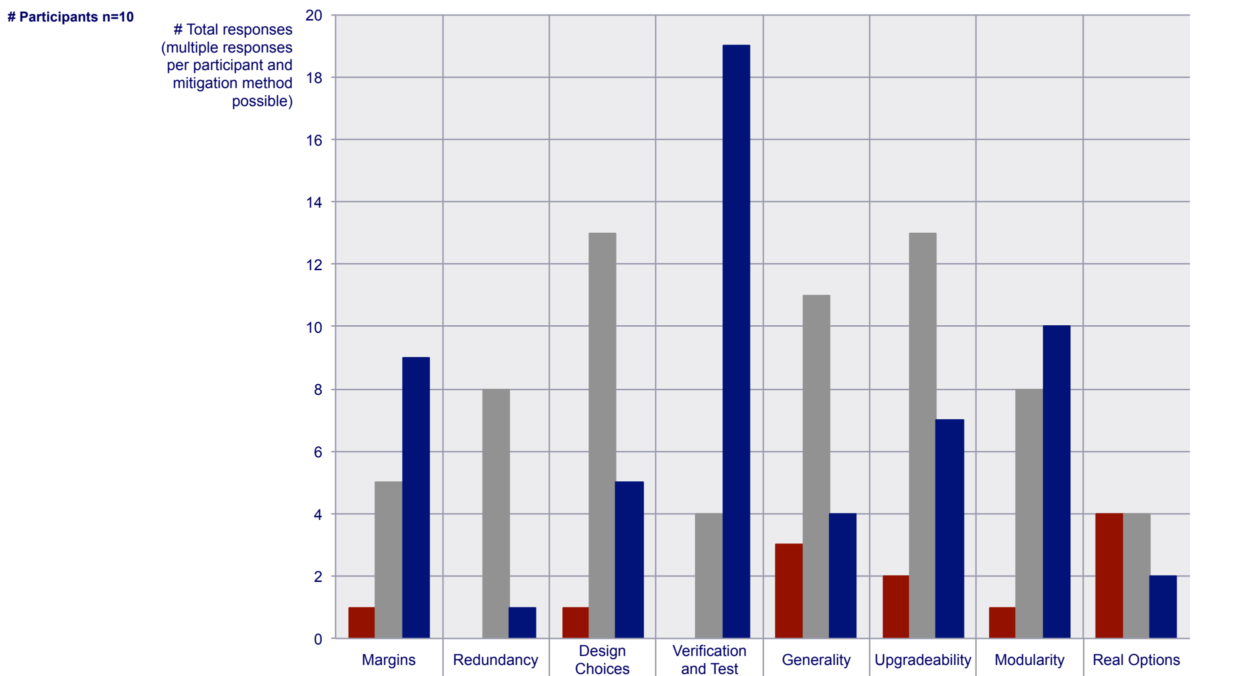
Costs over budget in % (risk caused)		Time delay in % (risk caused)				
		< 5 %	5 - 20 %	20 - 35 %	35 - 50 %	> 50 %
	> 50 %	*9* Major Update	*9* Major Update	*9* Major Update	*9* Major Update	*9* Major Update
	35 - 50 %	*3* Minor Update	*3* Minor Update	*3* Minor Update	*9* Major Update	*9* Major Update
	20 - 35 %	*1* Light Update	*1* Light Update	*3* Minor Update	*3* Minor Update	*9* Major Update
	5 - 20 %	*1* Light Update	*1* Light Update	*1* Light Update	*3* Minor Update	*9* Major Update
	< 5 %	*0* No consequences	*1* Light Update	*1* Light Update	*3* Minor Update	*9* Major Update

Most consequences were indicated in the empirical case study related to modules (32 total responses) followed by interfaces (28 total responses) and architecture (26 total responses). The architecture should be the most stable part in the platform because it is the core element in the platform, but the empirical case study has shown, that the architecture is not that stable and was updated or renewed in several cases.

3.2.3 Usage of Risk Mitigation Methods

The participants were asked in question no. 8 in the empirical case study, which methods they are using for mitigating uncertainty and risk. The mitigation methods are a part of the research questions as mentioned in section 1.3, and they can be split in mechanisms, which can be implemented into the platform and mechanisms that can be done with or adopted to the platform as a strategy to reduce the impact of the risk. Mitigations that can be implemented in the platform are mechanism like margins, real options, upgradability, redundancy and modularity. Mitigations that can be applied as a strategy are design choices, principles of generality, and verifications and testing.

The result of the empirical case study was that the method verification and test is used in 34% of the answers followed by upgradeability, which is used in 32%. Modularity (28%), design choices (28%) and generality (27%) are in the middle. Real options (15%) and redundancy (13%) are relatively seldom used for mitigating uncertainties and risks.



	Margins	Redundancy	Design Choices	Verification and Test	Generality	Upgradeability	Modularity	Real Options	
we used it, but it was not effective	1	0	1	0	3	2	1	4	∑ = 21
we used it, the effectiveness was moderate	5	8	13	4	11	13	8	4	∑ = 19
we used it, the effectiveness was superior	9	1	5	19	4	7	10	2	∑ =15
	∑ = 15	∑ = 9	∑ = 19	∑ = 23	∑ = 18	∑ = 22	∑ = 19	∑ = 10	

Figure 3-4: Usage of different mitigation methods, total responses

The usage of a method was classified as ‘used, but it was not effectiveness’, ‘used, and the effectiveness was moderate’ and ‘used, and the effectiveness was superior’, it was also possible to mark a mitigations as ‘not used for this kind of risk’. The rating of each mitigation method was a subjective estimation of each participant. The responses are only considering the use of the mitigations in the platform projects and not the effectiveness of the mitigations in general in the companies of the participants. In Figure 3-4 the effectiveness rating of each mitigation is illustrated. When verification and testing was used, it was used in 83% with superior and in 17% of the application with moderate effectiveness; it wasn’t used with no effectiveness. When upgradeability was used, it was used in 32% with superior, in 59% with moderate and in 9% of the applications with no effectiveness. The investigation of the platform projects has shown that the strategy of design choices was used in 26% with superior, in 68% with moderate and in 6% of the cases with no effectiveness, when it was used. Looking at modularity, when it was used, in 53% it was used with superior, in 42% with moderate and in 5% with no effectiveness. When generality was used, it was used in 22% of the applications with superior effectiveness, in 61% with moderate effectiveness and in 17% with no effectiveness. Looking at margins, in case of application the effectiveness was in 60%

superior; in 33% moderate and in 7% it was not effective. Considering real options, the effectiveness in the product platform projects where it was used, was in 20% superior and in 40% moderate or not effective. When redundancy was used, it was used in 11% with superior and in 89% with moderate effectiveness; like verification and testing redundancy wasn't used with no effectiveness.

As mentioned in the paragraph above, the mitigation method, which was indicated the most as used with superior effectiveness, is verification and testing. An explanation for this result can be that the companies are using this strategy for long time in other projects and have significant experiences in using this mitigation method. As mentioned before, the most problems were identified in the verification and testing phase, which can be related to this result. It can be valuable to investigate other mitigations methods get experience in using them for identifying uncertainties and risks earlier in the development process. With a big gap in number of total responses verification and testing was followed by modularity and margins, followed by upgradeability, design choices and real options and at the end redundancy.

The mitigation methods, which were indicated most frequently as used with moderate effectiveness, are the strategy of design choices and systematic implementation of upgradeability. The next in the row of numbers of responses is the approach of generality, followed by modularity and redundancy. Margins, verification and testing, and real options are at the end of this row.

The total number of mitigations marked as used not effectively was relatively low in comparison to used with moderate or superior effectiveness. The implementation of real options got the most responses in this category, followed by generality upgradeability, and modularity, margins and design choices with the same number of total responses.

As it could be seen in Figure 3-4, the ratio of used with superior to used with moderate effectiveness is higher than 1 for verification and testing, modularity, and margins. There is no mitigation method where a ratio higher than one of used with no effectiveness to used with moderated effectiveness was indicated. Looking on real options the ratio of used with no effectiveness to used with superior effectiveness is greater than 1. In this case it is 2, which means that it was used twice as much with no effectiveness than with superior effectiveness.

The number of total response per mitigation method in Figure 3-4 can be higher than the number of participants, because the figure illustrates a summary of the responses over all risk areas. As can be seen in Appendix A, question 8, the participants had the chance to indicated more than one mitigation method per risk area. The indicator n=10 provides the information that ten participants answered the question 8 of the knowledge gathering instrument. To providing more detailed results, in Figure 3-6 to Figure 3-10 the effectiveness of the mitigation approaches is illustrated for each risk area, in this figure the maximal number of total responses per risk area is equal to the number of participants (n=10). For all these figures the legend is documented in Figure 3-5.

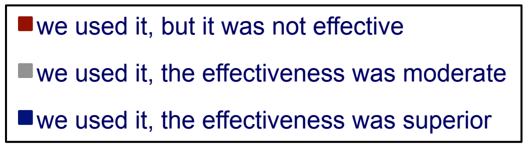


Figure 3-5: Legend for Figure 3-6, Figure 3-7, Figure 3-8, Figure 3-9 and Figure 3-10

As it can be seen in Figure 3-6, seven out of ten participants said that the effectiveness was superior when they used verification and testing for mitigating uncertainty and risks concerning technology changes or technology capability issues.

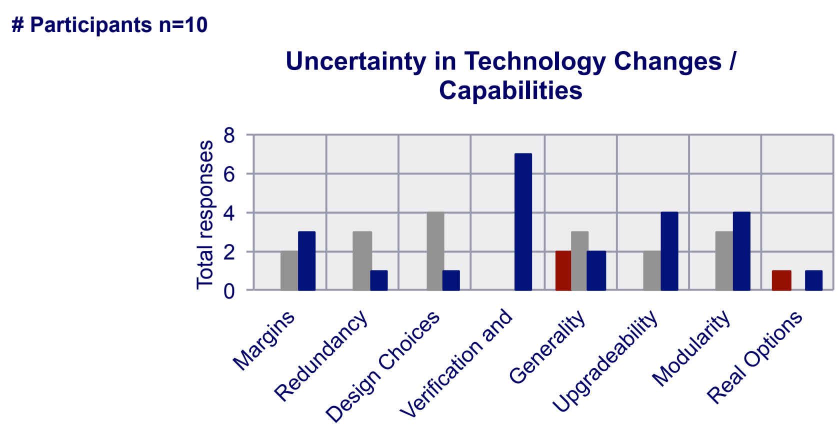


Figure 3-6: Effectiveness of different mitigation methods in risk area of technology changes and capability issues (43 total responses)

Four out of 10 said that the effectiveness of upgradeability and modularity was superior, when they used it in this risk area. Design choices were used by 4 out of ten with moderate effectiveness, when this strategy was applied to mitigate uncertainties and risks related to technology changes and capability issues. The approach of generality was used by 2 out of ten with no effectiveness for mitigating risks when it was applied the technology risk area. Real option was applied in on case with superior and in one case with no effectiveness to mitigate technology risks in the investigated platform projects.

Looking on the effectiveness of the mitigation methods regarding to risks in the area of customer needs, illustrated in Figure 3-7, it can be seen that verification and testing was also the method which was used most frequently with superior effectiveness, in this case 5 out of 10 marked it in the empirical case study.

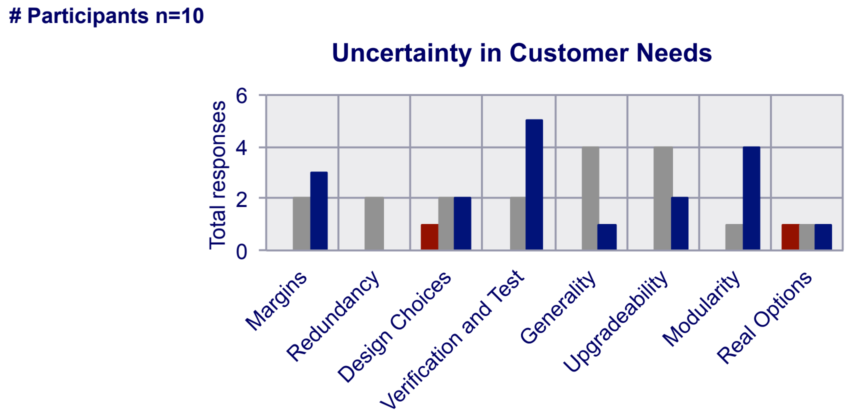


Figure 3-7: Effectiveness of different mitigation methods in risk area of customer needs (38 total responses)

The approach of modularity was indicated as used with superior effectiveness in the cases of application by 4 out of 10 participants, followed by margins (3 out of 10), design choices, upgradeability (both 2 out of 10) and generality and real options with each 1 out of 10 responses for used with superior effectiveness. Redundancy was not used in any case with superior effectiveness to mitigate risks regarding to changes in customer needs. Generality and upgradeability were marked by 4 out of 10 participants with moderate effectiveness to address risks regarding to customer needs. Both, design choices and real options, were marked by 1 out of 10 participants, that the application of this mitigation method was not effective for mitigate risks regarding customer needs.

Figure 3-8 illustrates the effectiveness of different mitigation methods regarding to risks in the area of market and business shifts.

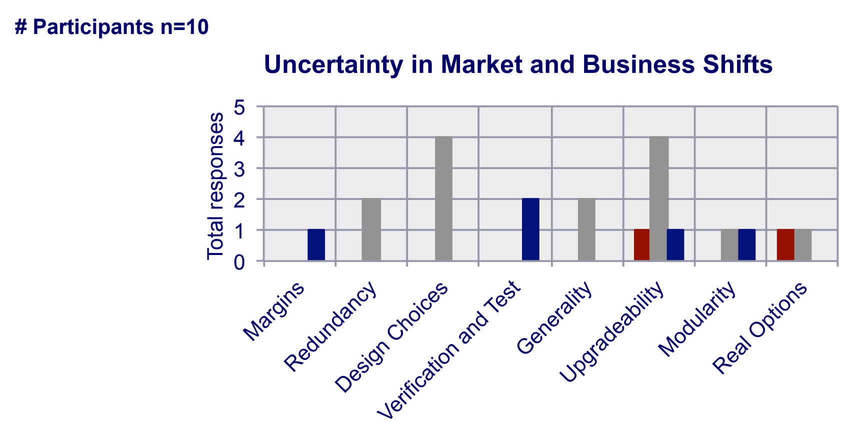


Figure 3-8: Effectiveness of different mitigation methods in risk area of market and business shifts (21 total responses)

In comparison to the effectiveness of the mitigation methods regarding to risks in technology changes and capabilities, and customer needs, there is no mitigation method indicated, which is used more often with superior effectiveness than with moderate effectiveness. In the both other areas there was at least one mitigation method that was indicated more often with superior than moderate effectiveness. That means the likelihood of moderate effectiveness when using one of the investigated mitigation methods is higher than the likelihood of applying it with superior effectiveness for addressing risk regarding market and business shifts. As it can be seen in Figure 3-8 the methods redundancy, design choices, generality and real options were only applied with moderate or no effectiveness for mitigating risks related to the market and business.

In Figure 3-9 the effectiveness of the different methods to mitigate risks related to political and cultural changes is illustrated.

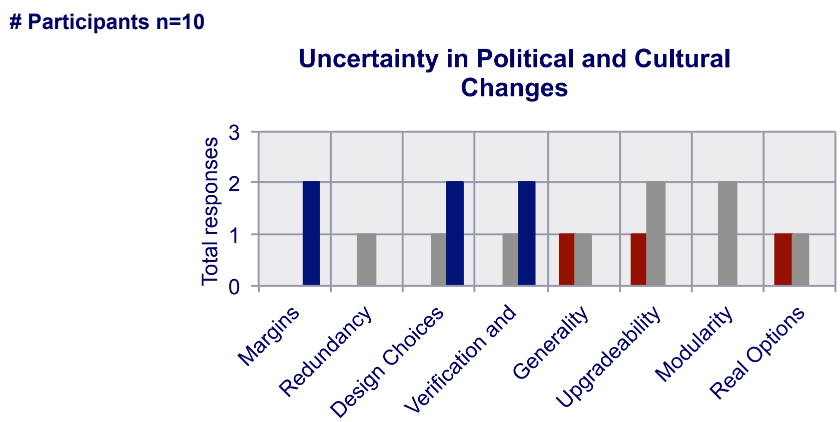


Figure 3-9: Effectiveness of different mitigation methods in risk area of political and cultural changes (18 total responses)

The number of responses in this area in the empirical case study was relatively low in comparison to other areas like risks in customer needs or technology changes. There was no peak indicated of a mitigation method, which was used with superior effectiveness. Margins, design choices and verification and testing were used by 2 out of 10 with superior effectiveness for mitigating risks because of political or cultural changes. The same number of participants used the methods upgradeability and modularity with moderate effectiveness to mitigate risks in this area. Generality, upgradeability and the real options approach was each indicated by 1 out of 10 participants that each mitigation was used for addressing risks in this area, but without any effectiveness.

In Figure 3-10 the responses of the empirical case study regarding the effectiveness of the mitigation methods by addressing risk because of uncertainties related to organizational changes are shown.

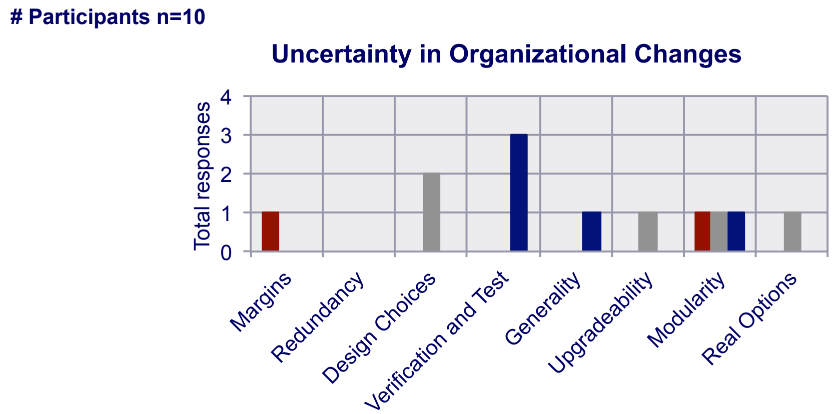


Figure 3-10: Effectiveness of different mitigation methods in risk area of organizational changes (12 total resoponses)

Three out of 10 participants used verification and testing with superior effectiveness to mitigate risks in the area of organizational changes. Design choices were used by 2 out of 10 with moderate effectiveness to mitigate risks in this area. Most of the other mitigation methods were use by 1 out of 10 with superior, moderate or no effectiveness to mitigate risks related to uncertainties in organizational changes, as it can be seen in Figure 3-10.

3.2.4 Performance of Analyzed Platform Projects

This section covers the analysis results of the data gathered with question 9 of the knowledge gathering instrument, which is documented in Appendix A. The results of the empirical case study related the overall performance of the companies in product platforms are documented in this section. The investigation of the different platform projects has indicated, that the participants reduced the time to market of all products based on one platform in average by 38%. Some mentioned that it took longer than with a single product development approach to have the first derivative on the market but all in all they saved time because they had the possibility to launch the other derivatives of a product family based on one product platform quickly after the first one.

The results of the investigation of the possibility offering better quality, reacting faster to changes are illustrated in Figure 3-11, as well as the possibility to increase the variety of products on the market because of a platform. The participants ranked the performance of their product platform subjectively on a scale from 1 to 7, where 1 was ‘totally applicable’ and 7 was ‘not at all applicable’. The average ranking of the responses regarding offering better quality because of a platform was indicated with a merit of 3.11, what is between the middle and the better end of the 1 – 7 scale. The average ranking of the participants regarding the speed to react faster after changes in the customer context was indicated with a merit of 3.17 on the 1 – 7 scale, which is also better than the arithmetic mean on the provided scale. As illustrated in Figure 3-11, the average of the ranking regarding to the increase of product variety because of the platform was next to 1, what means that nearly every participant of the case study totally agreed with this statement of having a higher variety of products on the market because of the platform, the merit of the responses in this question was 1.67.

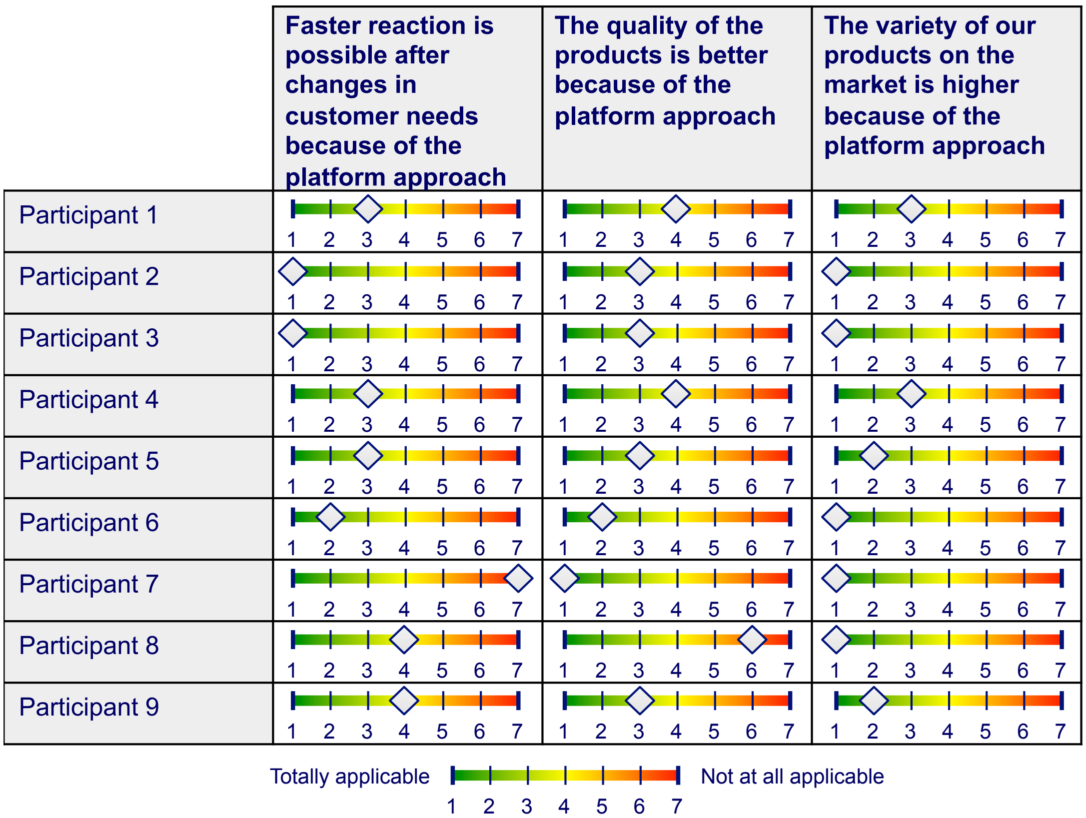


Figure 3-11: Characteristics regarding quality, reactivity and range of products.

For measuring the performance, the participants had to rank their product platform projects related to the reached cost savings on a scale with seven steps, ranging from saving costs more than 60% to less than 5%. The result of this investigation is illustrated in Figure 3-12.

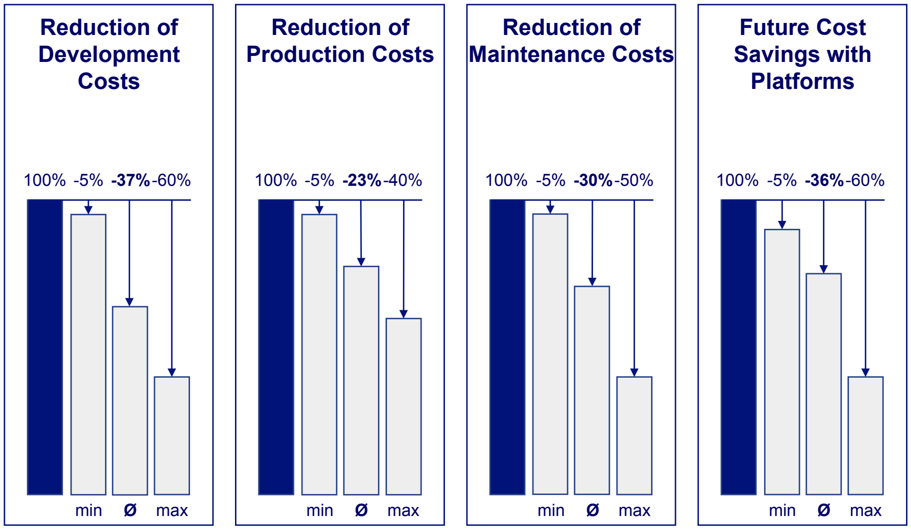


Figure 3-12: Reduction of different cost factors because of the platform

The investigation of cost savings because of the product platforms was split in development, production, and maintenance costs and future cost savings in total. The largest reduction was in development costs, followed by maintenance costs. The reduction of production costs was lower; this can be based on the fact that most for the derivatives based on the investigated product platforms are manufactured in low-cost countries in Asia. The average of all domains was between 23% and 37%; it indicates the benefit of the platform approach related to the possibility of saving costs.

The potential of future cost savings was indicated as more than 36%. This indicates that there is a huge potential of cost savings in the future and it might be useful to investigate different leverages for realizing this potential by forcing the platform approach. None of the participants indicated that there is no further cost saving potential. Providing the last result of the empirical case study, the re-use rate in the considered platform projects is 42% in average; it varies from 10% to more than 60%.

3.3 Chapter Summary

A conclusion of this empirical case study is that most of the problems occurred because of uncertainties that could have been predicted. In some cases the information about uncertainties and risks were available too late in the development process. Information about customer needs and markets were not detailed enough with the consequence that not all use cases can be addressed by the platform. Furthermore, the uncertainties are related to the complexity of the product and the specification of the platform. Uncertainties and the treatment of uncertainty are also related to the kind of products (e.g. mass consumer products with high replacement frequency vs. specialized products with long time on market). For example, if changes are needed in a mass product, companies often develop a new part or even a new platform, because it is not worth it updating the existing one. In industries with specialized products and long time in market the platforms are often updated. There was also a large decrease of time to market because of the platform project and the case study has shown that the different companies reduced their costs dramatically but there is still a huge potential for cost savings.

One fundamental result was, that the companies often started to react after they realized that the platform didn't meet the expected performance. That means that the companies often realized too late that there was a problem because of an uncertainty. It appeared that they did not have an approach for successfully managing uncertainty and building a capable enough platform to deal with all these points. As a result of all the outcomes of the empirical case study, a framework was developed to provide a tool for managing uncertainties and risks within a platform. This framework is described in the next chapter and it is applied to an illustrative example in Chapter 5.

4 Framework for Uncertainty Management in Product Platform Development

This chapter proposes a framework for managing uncertainty and risk in product platforms. The empirical case study has shown that having more knowledge at the beginning of platform development can reduce the likelihood and severity of the consequences of uncertainty and risk. The overall goal of this framework is to gather enough knowledge at the front-end to develop a platform that delivers the expected functional performance over the different epochs of a platform lifecycle. The approach in this research is to identify the critical parts in the platform first before investigating the implementation of mitigation mechanisms to build in ‘ilities’ like flexibility, changeability or evolvability. The visualization of the platform can be done with model-based systems engineering approaches and tools like SysML and the comparison of the different designs can be conducted with Multi-Attribute Tradespace Exploration.

4.1 Framework Overview and Description

The framework is based on the literature review in Chapter 2 and it is also derived from the results of the interviews of the empirical case study described in Chapter 3. As shown in the framework overview in Figure 4-1 the steps are not linear, steps 2, 3 and 4 are in a cyclic order and should be repeated until all conceivable uncertainties are identified.

Some of the steps are commonly used in systems engineering so they are not described in detail in this chapter, a description can be found in systems engineering and product development bibliography. The focus in this chapter is on the steps related to management of uncertainties; these are Step 4 and Step 5 in this framework. Step 1 describes how customer needs and requirements can be identified, Step 2 is about investigating the product platform designs and bandwidth, based on the customer requirements. After the characterization of uncertainty and risk in Step 3, epochs are described and analyzed in Step 4. Step 5 is about the performance assessment of the different platform designs in consideration of the identified uncertainties and risks. In Step 6 the platform designs are compared in a tradespace and the best suitable is selected if it fits with the stakeholder requirements. If the results are not satisfactory, the process can be done again. Step 7 is for monitoring the context and reviewing the selected design if the performance is satisfying after context changes. Step 7 is more or less parallel to all the other steps. If a performance problem is identified during the monitoring of the functional platform performance, a decision must be taken what the best step is to reenter into the framework. An activity is also to identify the implications of changes in the platform to other important areas.

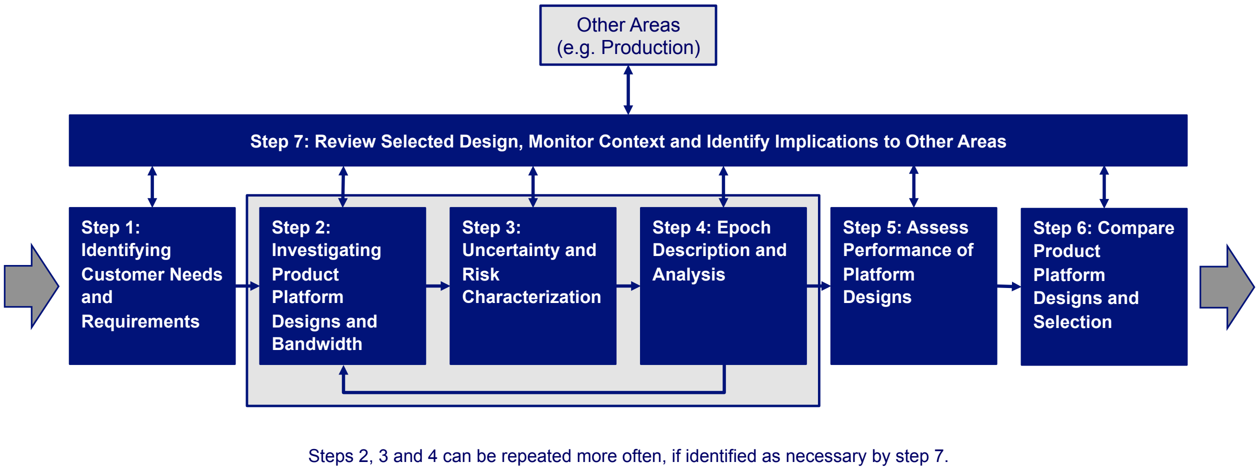


Figure 4-1: Framework overview

Each of the seven steps is split in inputs, activities and outputs, as documented in Figure 4-2. An input describes the information needed from the context or from another step, most frequently the step ahead.

Step 1

Inputs
1.I.1 Internal Capabilities
1.I.2 Market information
1.I.3 Use case information
1.I.4 Customer information

Activities
1.A.1 Determine lifetime
1.A.2 Set customer requirements
1.A.3 Identify engineering metrics
1.A.4 Specify expected platform performance
1.A.5 Create QFD Phase I

Outputs
1.O.1 Project time constraints
1.O.2 Market segments
1.O.3 Expected platform performance
1.O.3 Customer requirements
1.O.4 Engineering metrics
1.O.5 QFD Phase I

Step 2

Inputs
2.I.1 Project time constraints
2.I.2 QFD Phase I

Activities
2.A.1 Create QFD Phase II
2.A.2 Create components coupling matrix
2.A.3 Determine platform bandwidth
2.A.4 Specify flows between components

Outputs
2.O.1 QFD Phase II
2.O.2 Coupling matrix
2.O.3 Platform bandwidth
2.O.4 Platform designs
2.O.5 Flows between components

Step 3

Inputs
3.I.1 Platform bandwidth
3.I.2 Platform designs
3.I.3 Flows between components
3.I.4 Risk areas

Activities
3.A.1 Identify uncertainty and risk
3.A.2 Analyse uncertainty and risk
3.A.3 Evaluate uncertainty and risk

Outputs
3.O.1 List of uncertainties and risks
3.O.2 Uncertainty and risk analysis
3.O.3 Uncertainty and risk evaluation

Step 4

Inputs
4.I.1 Uncertainty and risk evaluation
4.I.2. Internal capabilities
4.I.3 Market segments

Activities
4.A.1 Combine uncertainties to epochs
4.A.2 Trace impact of epochs on engineering metrics
4.A.3 Trace impact of epochs on components
4.A.4 Identify critical components

Outputs
4.O.1 Epoch description
4.O.2 Impact of changes in epochs on platform
4.O.3 Critical parts in platform

Step 5

Inputs
5.I.1 Critical parts in platform

Activities
5.A.1 Identify mechanisms for critical parts
5.A.2 Describe changes of mechanisms on critical parts
5.A.3 Update QFD Phase II
5.A.4 Assess performance over time

Outputs
5.O.1 Mechanisms for critical parts
5.O.2 Updated or new engineering metrics
5.O.3 Updated or new components
5.O.4 Updated QFD Phase II

Step 6

Inputs
6.I.1 Platform designs
6.I.2 Key stakeholder attributes

Activities
6.A.1 Build tradespace of platform designs
6.A.2 Compare platform designs
6.A.2 Select platform design

Outputs
6.O.1 Tradespace of platform designs
6.O.2 Raking of Platform designs
6.O.3 Final platform design

Step 7

Inputs
7.A.1 Final platform design
7.A.2 QFD Phase I
7.A.3 QFD Phase II
7.A.4 List of uncertainties and risks
7.A.5 Components coupling matrix

Activities
7.A.1 Create Engineering System Matrix
7.A.2 Monitor Context
7.A.3 Trace impact of changes to Engineering System Matrix
7.A.4 Take mechanism decision
7.A.5 Connection to other areas.

Outputs
7.O.1 Changes in context
7.O.2 Changes in Engineering System Matrix
7.O.3 Mechanisms related decisions

Figure 4-2: Inputs, activities and outputs per step in detail

Activity describes the actions that are done within a step, and outputs are the information generated in a step and which are useful for the context or needed in another step. It is recommended to determine the activities in the displayed sequence, starting by the first one per phase. Steps 1, 2 and 3 (e.g. Martin 1999, Ulrich and Eppinger 2008) are common and well described in the existing literature. Step 6 is based on the research of Ross (2006).

4.1.1 Step 1: Identifying Customer Needs and Requirements

According to Ulrich and Eppinger (2008) identifying customer needs is an integral part of the concept development phase. The resulting customer requirements are used to guide the team in establishing platform specifications, generating platform concepts, and selecting a platform concept for further development. As it can be seen in Figure 4-3 Step 1 includes four inputs, five activities and also five outputs. It is also indicated, that there is a bi-directional connection to Step 7 and an output connection to Step 2.

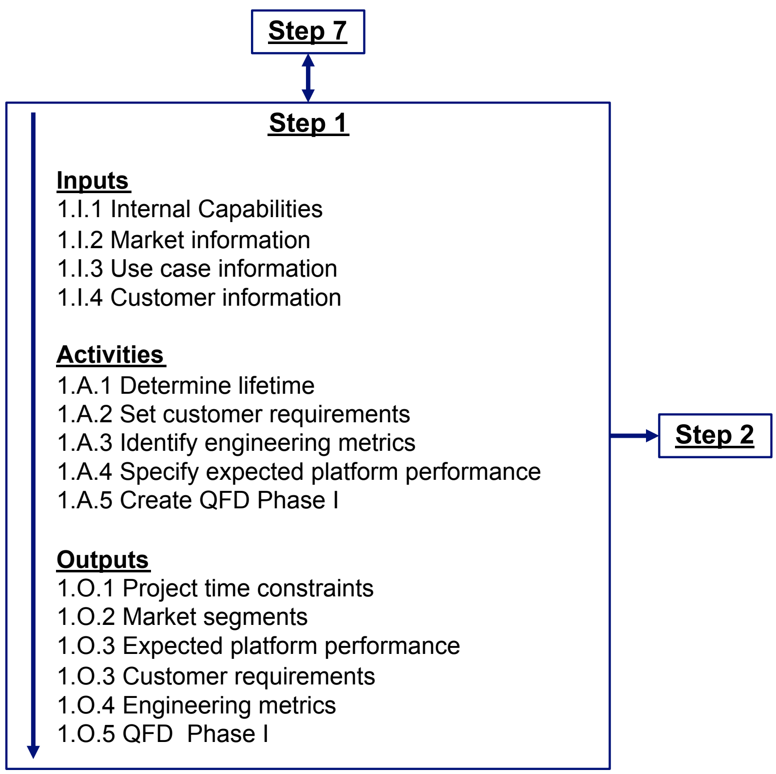


Figure 4-3: Overview Step 1 and connections to related steps of the framework

First of all, information about the internal capabilities of a company is necessary for a successful product platform development. With this information decisions about the complexity and timeline of the product platform project can be taken as well as decisions about adding additional capabilities or outsourcing parts. With this information the development process can be started with collecting customer requirements. At the beginning of a new platform development, there is a fuzziness concerning customer requirements. The customer fuzziness can be split into portfolio, preference, lifecycle and volume fuzziness (Zhang and Doll 2001). In order to develop a successful product platform, it is useful that a corporation listen accurately to identify the needs and expectations of each market segment and market level. Market levels can be, for example, low end, middle or high end products. While looking at the competitive landscape, each market niche needs to consider the segments and levels.

To identify the 'voice of the customer' a '360 degree' view need to be developed by the company to understand their needs, requirements or patterns of usage. This approach can be effective to set the product platform specifications and features (Gordon 2004).

The following questions can be asked to identify the customer requirements and market segments:

- What is the significance of each segment?
- What are the key products?
- What are their volumes, revenue, and profits?
- What is the outlook for sales volumes, revenues and profits the next 5years for this market segment and product?

An understanding of where the market is headed is critical to assess the functional performance of a platform. Also, the team must determine how long they would like the product platform to last. Methods to map the future product plans were discussed by Wheelwright and Sasser (1989) and Wheelwright and Clark (1992). Also, Schofield (2008) proposed that time constraints of the platform project should be discussed and listed including the expected time to market and expected service life of the platform related derivatives.

The process of identifying and set customer requirement attributes includes five steps (Ulrich and Eppinger 2008). The first step is to gather raw data from customers, which are interpreted into customer requirements in a second step. After the requirements are set the next step is to organize these into a hierarchy. The fourth step is to establish the relative importance of needs before the process of identifying and needs the results of the process are reflected.

The key benefits of the process are to ensure that the platform-based products are focused on customer requirements and that no critical customer requirement is forgotten, and also that a clear understanding among members of the development team of the requirements of the customer in the target market is developed. The creation of a fact-base to be used in generating concepts, selecting a platform concept, and establishing product specifications; and creating an archival record of the needs phase of the development process are also benefits of the process. The customer requirements are phrases, which customers use to describe products and product characteristics. The customer requirements can be split into three levels as shown in Table 4-1, the level of detail is growing from the left to the right columns (Hauser and Clausing 1988).

Table 4-1: Levels of customer requirements (Hauser and Clausing 1988)

Primary	Secondoy	Tertiary
Good operation and use	Easy to open and close door	Easy to close from outisde
		Stays open on a hill
		Easy to open from outside
		Doesn't kick back
		Easy to close from insie
		Easy to open fom inside
	Isolation	Doesn't leak in rain
		No road nois
		Doesn't leak in car wash
		No wind noise
		Doesn't drip water or snow whe open
		Doesn't rattle
	Arm rest	Soft, comfortable
		In right position
Good appearance	Intrior Trim	Materaial won't fade
		Attractive (nonplastic look)
	Clean	Easy to clean
		No grease from door
	Fit	Uniform gaps between matching panels

Based on the tertiary customer requirements a Quality Function Deployment (QFD) Matrix can be set up. After the customer requirements are identified, the next step is to identify and specify the related engineering metrics and to create the Quality Function Deployment Matrix Phase I.

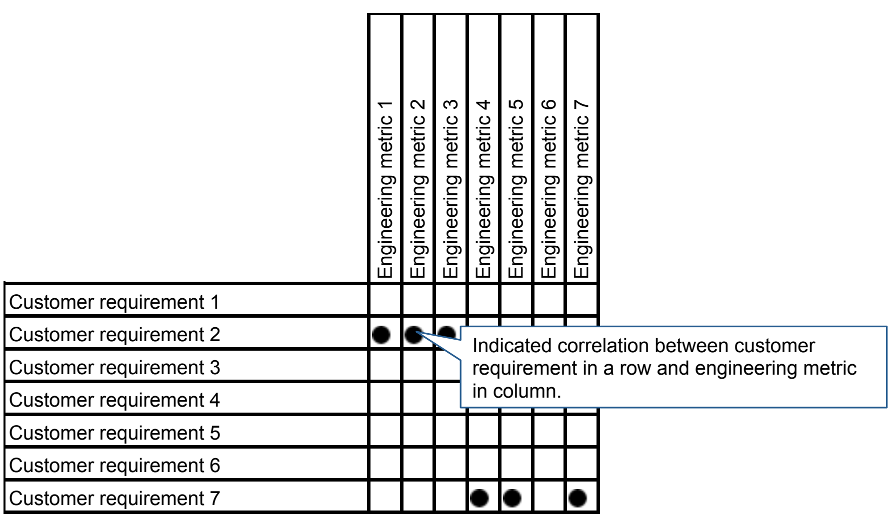


Figure 4-4: Example of a QFD Phase I

The illustrative scheme of a QFD Matrix Phase I is displayed in Figure 4-4 and it maps the engineering metrics to the identified customer requirements. The customer requirements are the headings of the rows and the engineering metrics are the headings of the columns. For each engineering metric a bandwidth of values with a minimal and a maximal value has to be defined, it depends on the type and the context of the engineering metric, if the minimal or the maximal value is the preferable one. For example, if the engineering metric ‘weight’ of a sports-car is specified in this step, the lower value is the better one but if the engineering metric ‘speed’ is characterized, normally the higher value is the better one, so it is type dependent. If the engineering metric ‘weight’ is specified for a sports-car and a hydraulic shovel, the lowest one is the best one for the sports-car because the relation of power/weight is better and the car can go faster, but for the hydraulic shovel, the higher weight is probably better, because it is more staple and can carry more weight with its shovel, so it is context dependent. The bandwidth between the both limits can be described with a utility curve or a quadratic loss function in cases where more accuracy is needed.

The identification of the bandwidth of the engineering metrics is called specifying the expected platform performance. This is an important step for the process, because based on this values the components are identified in Step 2 of this framework. The engineering metrics are also a part of the uncertainty and risk analysis of Step 4. So if the metrics are not defined well enough, there can be problems in the further steps of the framework.

4.1.2 Step 2: Investigating Product Platform Designs and Bandwidth

The goal of Step 2 is to investigate the components based on the customer requirements and engineering metrics defined in Step 1. Also the flows between the components are determined in this step and the coupling matrix of the components is created. The bandwidth of a product platform describes which range of components is in the platform, and which parts are customized for the range of derivatives. As it can be seen in Figure 4-5 Step 2 includes two inputs, four activities and also five outputs. It is also indicated, that there is a bi-directional connection to Step 7 and an input connection from Step 1 and an output connection to Step 3.

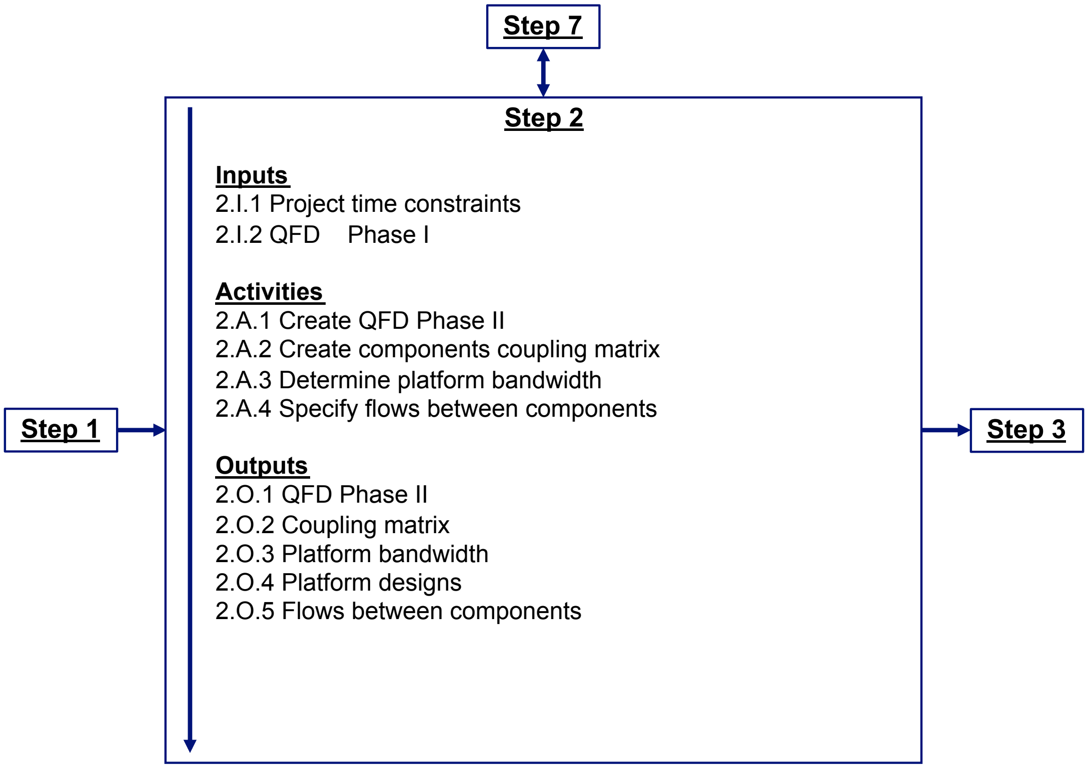


Figure 4-5: Overview Step 2 and connections to related steps of the framework

Inputs for this step are the determined time constraints and the QFD Matrix Phase I out of Step 1. For creating QFD Matrix Phase II different sets of components sets are investigated and mapped against the engineering metrics. Each node in the QFD Matrix Phase II can be rated on a 9/3/1/0 scale. This rating indicates which components have a high, medium, low or no influence to the corresponding engineering metric (Martin 1999). Another analysis is possible with this matrix when the path from the engineering metrics to the components is analyzed. With this path it can be analyzed which engineering metric is fulfilled at which level by the coupled component. An example of the QFD Matrix Phase II is displayed in Figure 4-6, where the engineering metrics are the headers of the rows and the components the headers of the columns.

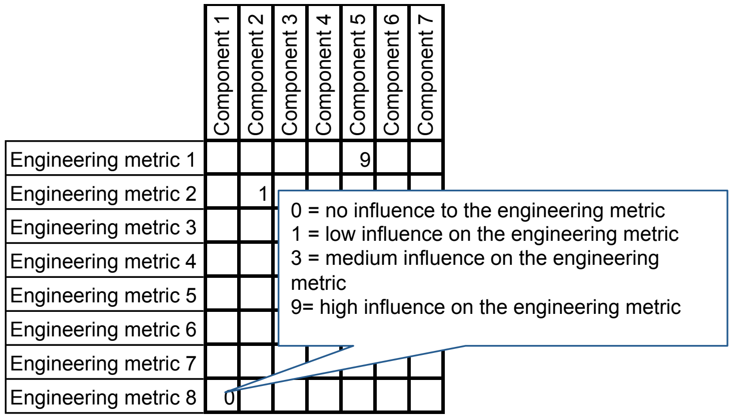


Figure 4-6: Example of a QFD Phase II

The next step is to create the component-coupling matrix based on QFD Matrix Phase II. In this matrix components are indicated in rows and columns of the matrix identically. The diagonal of the matrix represents the components and the nodes of the diagonal representing the connections and flows between the each of the components. If a connection or flow is indicated it means that there is a flow from component in the column to the component indicated in the row.

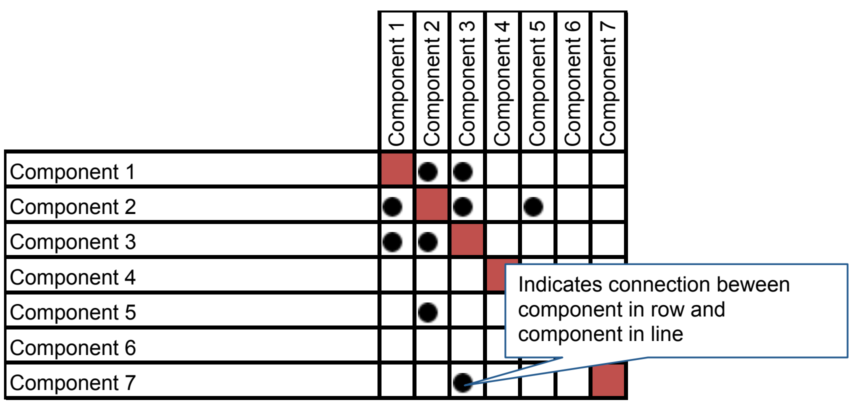


Figure 4-7: Component-coupling matrix

Optimizing the component-coupling matrix can help to find and determine the modules of the platform. Optimizing means that the matrix is transferred to a version in which the nodes are along the diagonal as much as possible. This is a part of the next step to determine the platform bandwidth. It means that the decision is taken, which components of the product are in the platform and which ones are in the architecture and in the modules. The goal is to determine the best combination of the components within the modules and the architecture to have a platform for all use cases the company wants to use it. If the ration of components in the platform is high, it means that the grade of standardization can be high but the rate of customization can be low. When developing a platform a decision concerning the trade-off between standardization and customization has to be taken depending on the context of the product platform. This analysis can be based on cost measures that are not part of this thesis. The goal is to have a high standardization rate without losing the uniqueness of each product based on a platform. Some car manufacturers had some problems differentiating the interior of their cars spread over different segments, because the standardization ratio was high and visible to the customer by having the same dashboard, for example, in a midsize and a luxury vehicle for a significant difference in the purchase prize. It is obvious that some customers would not be satisfied because of this reason. The last activity in this step is to specify the nodes indicated in the coupling matrix. These nodes are indicating the flows between the components. The flows can be described with technical specifications and values. This step is on one side for a better understanding of the product platform and on the other hand for investigating type and size of the interfaces between the standardized modules, the architecture and the customized parts of the product.

4.1.3 Step 3: Uncertainty and Risk Assessment

This step is to characterize the different risks and uncertainties. The key for engineers is to think about the changes in the context, and design the platform to deliver the expected functional performance in case of predicted and unpredicted changes.

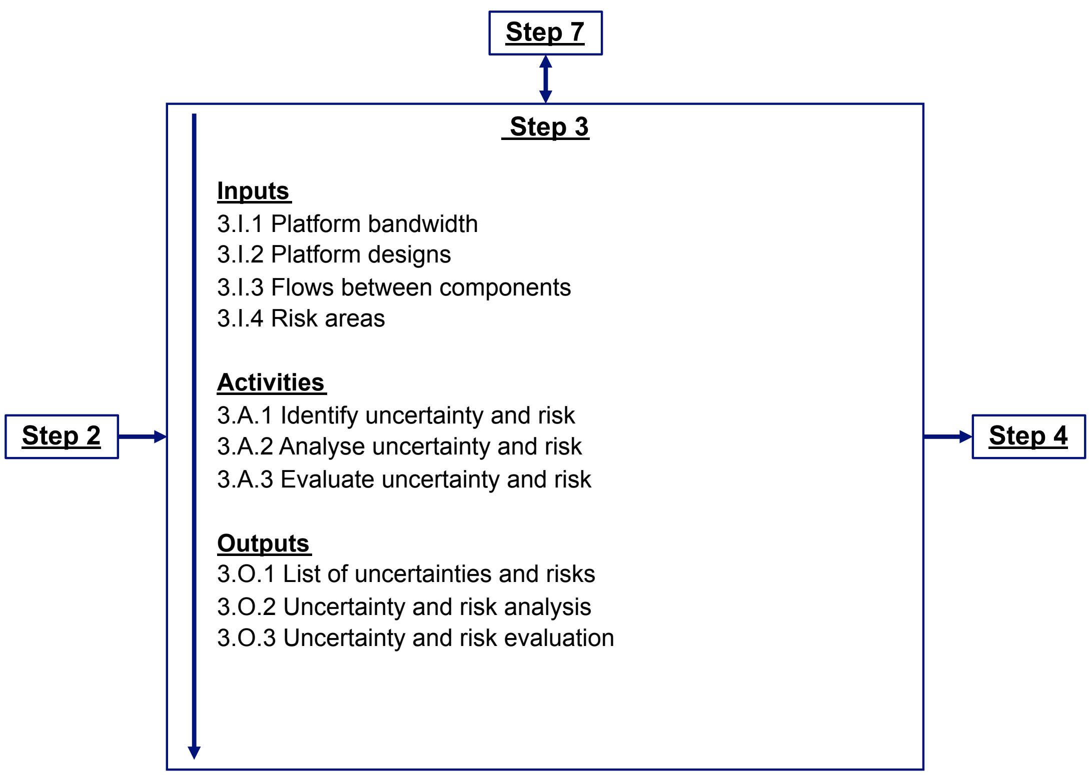


Figure 4-8: Overview Step 3 and connections to related steps of the framework

As it can be seen in Figure 4-8 Step 3 includes four inputs, three activities and also three outputs. It is also indicated, that there is a bi-directional connection to Step 7 and an input connection from Step 2 and an output connection to Step 4.

Uncertainties like lack of definition, lack of knowledge, statistically characterized variables, known unknowns and unknown unknowns can cause risks in the areas, which are described in section 2.3.1. These risk areas are technology changes, technology capabilities, changes in customer needs, market and business shifts, political and cultural changes, and organizational change. The identification of risks can be done with different methods like failure mode effect analysis, cause-and-effect analysis, scenario analysis, brainstorming, structured or semi-structured interviews or checklists. All these and further methods are described in ISO norms (ISO 2009a, 2009b, 2009c). The identified uncertainties can be listed in the uncertainty descriptor matrix like the one in Table 4-2.

Table 4-2: Uncertainty descriptor matrix

Risk Area	Uncertainty		Impact on platform		Likelihood	Phase of occurrence in Lifecycle (Assumption of recognition)
	description	type	type	qualitative or quantitative description		
Technology change						
Technology capability						
Customer requirement						
Market/ Business						
Political change						
Cultural Change						
Organizational change						

The first column is filled with the different risk areas mentioned earlier. Columns two and three are for specifying each uncertainty with a description and type. The description of uncertainty is a verbal description of each uncertainty (e.g. capabilities of technology X not certified). Type of uncertainty is within the range mentioned before, and at least one out of the list of types (lack of definition, lack of knowledge, statistically characterized variables, known unknowns, unknown unknowns) can be taken.

The next activity is to analyze the identified uncertainties and risks. This can also be done with techniques proposed by ISO (2009a, 2009b, 2009c) like root cause analysis, decision tree, risk indices or cause and consequence analysis and for describing the results of the analysis Table 4-2 can be used. The type of impact on the platform can be positive, negative or indifferent. Related to managing uncertainty for delivering the expected performance over time, the uncertainties with negative impact are the interesting ones for further investigations in this research. The positive ones are also important, for identifying opportunities. The impact of an uncertainty can be described qualitatively or quantitatively as disaster if the system causes harm, in case of related uncertainty and risk occurring. The outcome can be described as a failure, if the system does not work in this case or as degradation of capacity, if the system works, but not up to initial expectations. The impact can be described as funding, cost, or schedule deviations, if the program of developing the platform gets in one of several kinds of trouble because of the identified uncertainty. Other descriptions can be related to market or business shifts, if the system works, but there are changes in business surrounding or need shifts, if the system works, but function desired from the system has changed from that for which it was designed. The impact can be described as a coupling problem, if the system works in case of the related uncertainty but coupling problems occurring.

The next column in Table 4-2 is to assume the likelihood of each uncertainty and related risk, expressed in different stages like almost never, high, medium or low and rated with a 0/1/3/9 scale. The last column is to determine the phases of occurrence within the product platform lifecycle. Following algorithm can be used for calculating the severity of an uncertainty related risk. First, assign a likelihood of occurrence of each uncertainty related risk. The next step is to find the risk with the most severe consequences to the platform performance; this can be called S_{worst}. The following step is to compare the other identified risks to the worst one, the question here is "how many of these failures would be equally painful as the one identified as worst?"; this measure can be called N_i. In a next step the severity of each other severity can be calculated with the equation $S_i = S_{worst} / N_i$. The last step is to compute the risk using a combining equation described in section 2.3.2.1 and used for the Step 4 of the framework, which is documented in the next section.

The phases are the different ones mentioned in the description of the lifecycle described in section 2.2.2, which are concept definition, design phase, development and fabrication and integration and verification during the development of the platform. After start of production the lifecycle can be split in the phases production, use and retirement. The phases are described in detail in section 2.2.3. This analysis is used in Step 4 of the framework, where different epochs are defined for investigating the impact of uncertainties on the functional platform performance. Each phase can be divided in several epochs, this more important during the period when the derivatives of the product platform are on the market.

The last activity in this step is to evaluate the identified and analyzed risks on a high level. A more detailed evaluation is done in Step 4 of the framework, when the impact of the uncertainties and risks is traced to the engineering metrics and components.

4.1.4 Step 4: Epoch Description and Analysis

The goal of Step 4 is to trace the impact of the identified uncertainties to the engineering metrics and platform components. One valuable output is the list of identified critical parts in the platform, which are the base for identifying the mechanisms in Step 5 of the framework. As it can be seen in Figure 4-9 Step 4 includes three inputs, four activities and three outputs. It is also indicated, that there is a bi-directional connection to Step 7 and input connections from steps 1 and 3 and an output connection to Step 5.

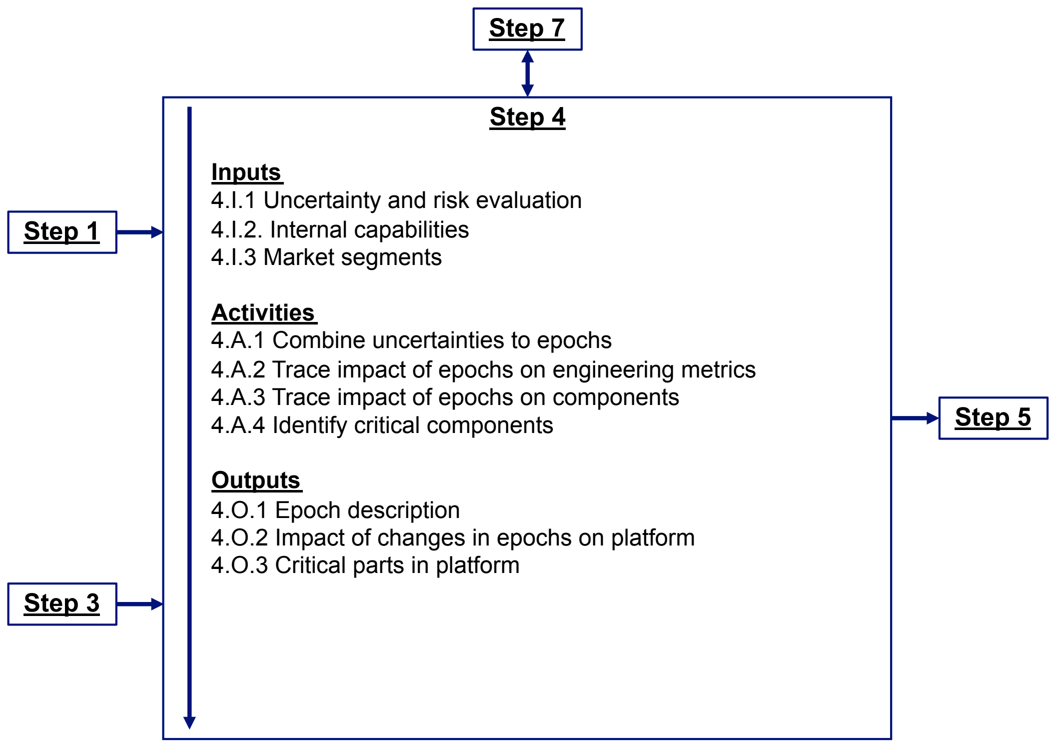


Figure 4-9: Overview Step 4 and connections to related steps of the framework

Based on the uncertainty descriptor matrix the impact on the components and the engineering metrics can be calculated. As listed under the activities and shown in Figure 4-10 there are two paths for identifying the critical parts in the product platform.

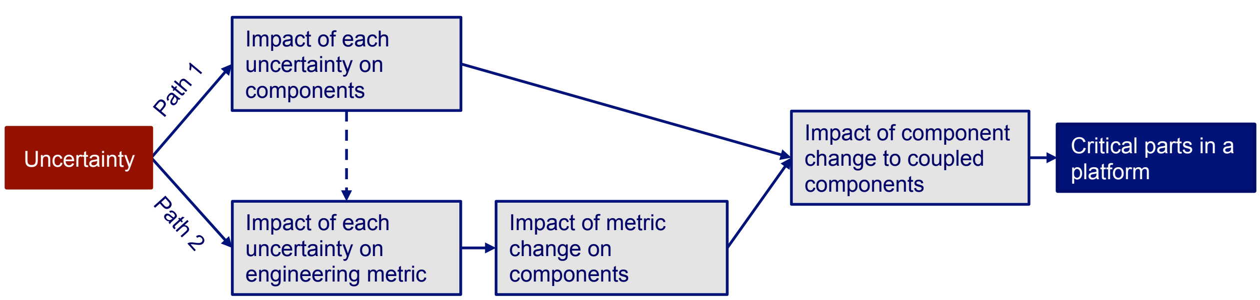


Figure 4-10: Two paths of identifying critical parts in platform

One path is from the uncertainty to the components, from the components to the coupled components and in some cases to the engineering metrics, normally the impact of the uncertainties in the engineering metrics is identified with path two. If the critical coupled components are identified the output of path one is the first part of a list with critical parts. Also the impact of the risks and uncertainties on the components to the engineering metrics has to be investigated in some cases. From this point the path is identical to the second path, which starts also at the uncertainties and the impact on the engineering metrics is identified. The next step in this path is to trace the impact from the engineering metrics to the related components and from the components to the coupled components.

In both cases the impact is measured as severity on a 9/3/1/0 scale where 9 is high severity, 3 describes medium severity, 1 describes low severity and 0 is no impact.

The impact of the uncertainties to the components and engineering metrics can be computed with following equations adapted from the ones proposed by Bahill and Smith (2009). For risk calculation in the components Equation 4-1and Equation 4-2 can be used, where Equation 4-2 is the normalized form of Equation 4-1.

$$R_{C(k)}(j) = L_{(ij)} \times S_{C(ij)}$$

Equation 4-1: Risk calculation for components

In Equation 4-1 and the following ones, $R_{C(k)}(j)$ is the risk in component k in epoch j and $R_{EM(n)}(j)$ the risk in the engineering metric n in epoch j. $S_{C(ij)}$ is the severity onto the components, $S_{EM(ij)}$ the severity onto the engineering metrics and $L_{(ij)}$ is the likelihood of uncertainty i in epoch j.

$$R_{C(k)}(j)(norm) = \frac{L_{ij} \times S_{C(ij)}}{L_{Max} \times S_{Max}}$$

Equation 4-2: Risk calculation components, normalized

For the calculation of the risk in the engineering metrics Equation 4-3 and Equation 4-3 can be used, where Equation 4-4 is the normalized form of Equation 4-3.

$$R_{EM(n)}(j) = L_{ij} \times S_{EM}(ij)$$

Equation 4-3: Risk calculation for engineering metrics

$$R_{EM(n)}(j)(norm) = \frac{L_{ij} \times S_{EM}(ij)}{L_{Max} \times S_{Max}}$$

Equation 4-4: Risk calculation for engineering metrics, normalized

The calculation of the risks in the components is illustrated in Figure 4-11, in the last row of the 'risk in component' section of this figure the maximum of the evaluated risk per component is calculated and normalized.

Risk Area	Uncertainty			Impact of uncertainty on components							Risk in Components						
	Description	Likelihood		Component 1	Component 2	Component 3	Component 4	Component 5	Component 6	Component 7	Component 1	Component 2	Component 3	Component 4	Component 5	Component 6	Component 7
Technology change	Description 1	medium	3	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0
	Description 2	medium	3	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0	0	0	0	0
Technology capability	Description 3	medium	3	0	3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	9	0	0	0	9	0
	Description 4	low	1	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Customer requirement	Description 5	medium	3	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	Description 6	medium	3	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Market/ Business	Description 7	medium	3	0	0	[illegible]	0	0	0	0	0	0	[illegible]	0	0	0	0
	Description 8	medium	3	0	9	0	0	0	0	0	0	27	0	0	0	0	0
Political change	Description 9	high	9	0	0	0	0	0	9	0	0	0	0	0	0	81	0
	Description 10	medium	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cultural Change	Description 11	medium	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Description 12	high	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Organizational change	Description 13	medium	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Description 14	low	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						MAX risk					0	27	0	0	0	81	0
						MAX risk (normalized)					0,00	0,33	0,00	0,00	0,00	1,00	0,00

Severity of uncertainty on component
0 = no impact
1 = low severity
3 = medium severity
9 = high severity

Risk = Likelihood x Impact of uncertainty on component

Figure 4-11: Visualization of risk calculating in product platform components

The calculation of the risks in the engineering metrics of the platform is illustrated in Figure 4-12, in the last row of the 'engineering metrics' section of this figure the maximum of the evaluated risk per engineering metric is calculated and normalized.

Risk area	Uncertainty			Impact on engineering metrics							Engineering metrics						
	Description	Likelihood		Engineering metric 1	Engineering metric 2	Engineering metric 3	Engineering metric 4	Engineering metric 5	Engineering metric 6	Engineering metric 7	Engineering metric 1	Engineering metric 2	Engineering metric 3	Engineering metric 4	Engineering metric 5	Engineering metric 6	Engineering metric 7
Technology change	Description 1	medium	3	0	0	0	3	0	0	0	0	0	0	9	0	0	0
	Description 2	medium	3	0	0	0	3	0	0	3	0	0	0	9	0	0	9
Technology capability	Description 3	medium	3	0	0	0	9	0	0	9	0	0	0	27	0	0	27
	Description 4	low	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Customer requirement	Description 5	medium	3	9	[illegible]	[illegible]	[illegible]	[illegible]	1	3	27	27	27	3	0	3	9
	Description 6	medium	3	3	[illegible]	[illegible]	[illegible]	[illegible]	3	9	9	9	9	0	3	9	27
Market/Business shifts	Description 7	medium	3	0	[illegible]	[illegible]	[illegible]	[illegible]	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0
	Description 8	medium	3	0	[illegible]	[illegible]	[illegible]	[illegible]	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	0
Political change	Description 9	high	9	0	[illegible]	[illegible]	[illegible]	[illegible]	0	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	3
	Description 10	medium	3	0	[illegible]	[illegible]	[illegible]	[illegible]	0	3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0	9
Cultural Change	Description 11	medium	3	0	0	0	0	0	0	0	0	[illegible]	0	0	0	0	0
	Description 12	high	9	9	3	3	0	0	0	0	27	9	9	0	0	0	0
Organizational change	Description 13	medium	3	0	0	0	3	0	0	0	0	0	0	9	0	0	0
	Description 14	low	3	0	1	9	0	9	0	0	0	3	27	0	27	0	0
							MAX risk				27	27	27	27	27	9	27
							MAX risk (normalized)				0,33	0,33	0,33	0,33	0,33	0,11	0,33

Severity of uncertainty on engineering metric
0 = no impact
1 = low severity
3 = medium severity
9 = high severity

Risk = Likelihood x Impact of uncertainty on engineering metric

Figure 4-12: Visualization of risk calculation in product platform engineering metrics

The risk calculation is to be done for each uncertainty in every epoch. Both, impact and probability can vary from epoch to epoch. L_{Max} and S_{Max} are the maximal values for likelihood and severity in each case, which can be chosen.

Based on the calculated risks in the engineering metrics the next task in path two is to investigate the impact to the coupled components. This calculation can be done with Equation 4-5,

$$R_{EM_nC_k}(j)(norm) = \frac{MAX\{R_{\ EM(n)}(j)(norm)\} \times N_{kn}(j)}{N_{Max}}$$

Equation 4-5: Impact of maximal risk in engineering metric to component

where $N_{kn}(j)$ is the coupling strength between engineering metric n and component k, and N_{Max} is the maximal coupling strength that can be set between the components.

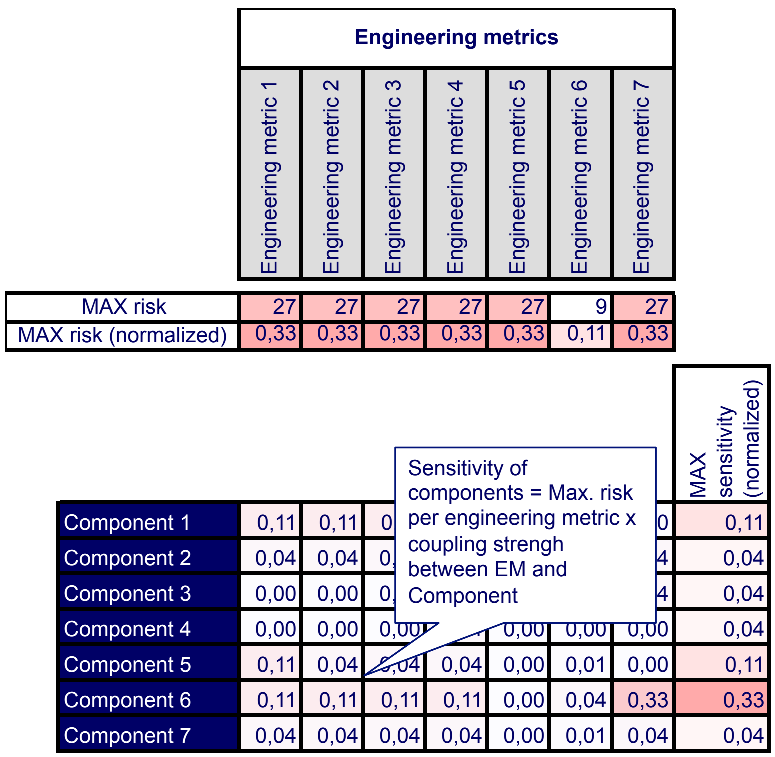


Figure 4-13: Visualization of the impact maximal risk in engineering metric to component

The calculation of the risks in the components because of the risk in the engineering metrics of the platform is illustrated in Figure 4-13, in the last column of the 'component' section of this figure the maximum of the evaluated risk per component because of the risk in the related engineering metrics is calculated and normalized.

4.1.5 Step 5: Assess Functional Performance of Platform Designs

This step is based on the results of identification of the critical parts in Step 4 and the goal is to identify and implement mechanisms for the critical parts for having the possibility to adapt the platform in case of a critical reduction of the platform performance. As it can be seen in Figure 4-14 Step 5 includes one input, four activities and also four outputs. It is also indicated, that there is a bi-directional connection to Step 7 and an input connection from steps 4 and an output connection to Step 6.

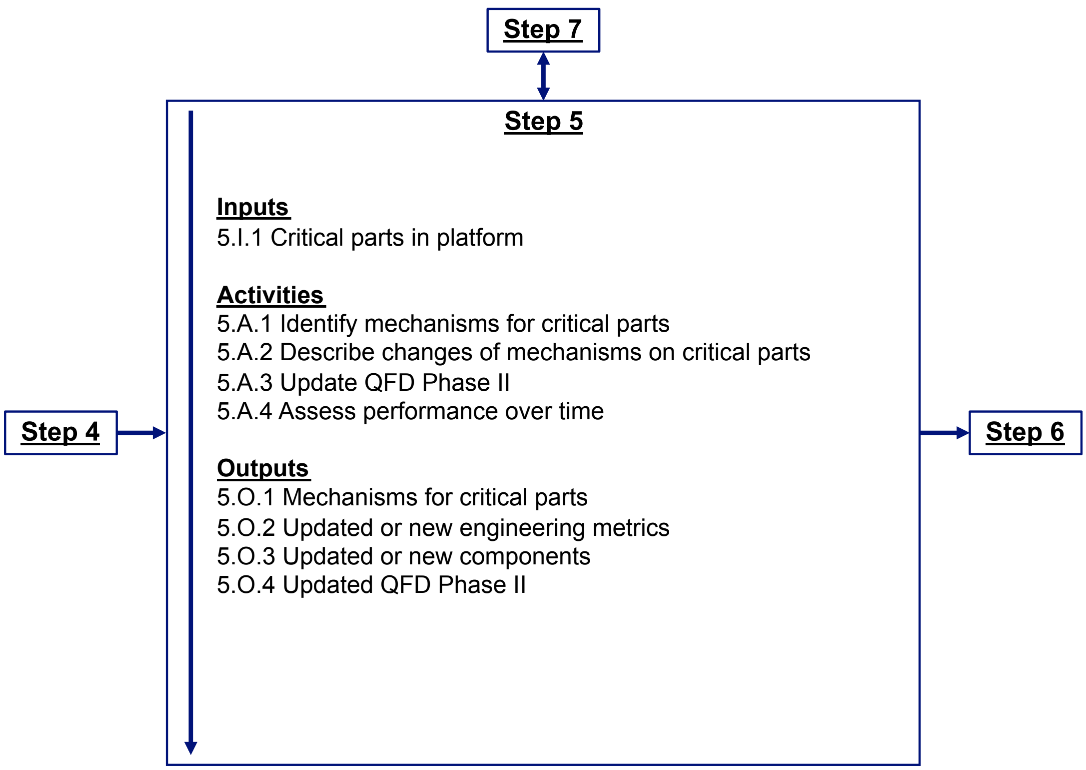


Figure 4-14: Overview Step 5 and connections to related steps of the framework

The sense of implementing mechanisms like real options or margins is, for example, avoiding the loss in functional performance by eliminating the risk source or changing the likelihood. Another result can be the change of the consequences of an uncertainty or risk, or sharing and thereby reducing the risk with another component in the platform. Also deciding not to start or continue with the activities that gives rise to the risk is an option for a mechanism.

Based on the list of critical parts, which was generated in Step 4 of the framework, mechanism for critical parts can be identified. Each critical part from Step 4 needs to be analyzed, important, for example, is which are the coupled components and is there an impact to the coupled ones, if a mechanism is implemented.

For each identified critical part in the platform the effect of a mechanism must be evaluated regarding possible value and costs. Implementing a mechanism can introduce risk itself to the platform and a significant risk can be the wrong implementation or ineffectiveness of the treatment measure. The questions are how to decide which are the right mechanisms for the critical parts and how to take the decision if it is valuable to implement the mechanisms. Some points that can support the decision of implementing a mechanism are listed in. The implementation is strongly related to the context of the product platform, for example, to factors like the length of the lifecycle, sales volume, development costs or production costs.

Table 4-3: Checklist for mechanism decision

Check list for choosing a mechanism
1.) Review the calculated risk for the critical parts
2.) Set rules for risk level treatment
3.) Investigate mechanisms for critical parts
4.) Cost and benefit analysis for each mechanism
5.) Set indicators for decision making of exercising mechanism
6.) Investigation of timeline for exercising mechanism
7.) Decision if and which mechanism to implement

There can be a timespan between taking the decision and having the output of the mechanism, which is enhancing the functional performance. For example, if the decision is taken to update or replace a module, there can be a timespan until the module is replaced, because it has to be developed first. From a cost perspective it is necessary to calculate the cost and the benefit of the mechanism and for each mechanism the impact on the functional performance of the platform is to calculate too. The goal is the reduction of risk impact. The change mechanisms influence the severity of the risk impact to the functional platform performance.

For each mechanism a set of indicators has to be created so that there are metrics for evaluating the decision of exercising the mechanism or not. It is also necessary to evaluate the length of the period between taking the decision, exercising the mechanism and have the output of the exercised mechanism that is enhancing the functional performance. Decision rules for the mechanisms can be based on different risk levels, which have to be set for each platform project individually.

All decisions related to implementing and exercising the mechanisms are based on comparing the fulfillment of the engineering metrics, which are the indicators for the functional performance. The idea of performance assessment is derived from the Epoch-Era Analysis of Ross and Rhodes (2008). They describe Epoch-Era Analysis as a systematic approach to think about the temporal system value in different environments. In this thesis it is adopted as a process for assessing the product platform performance of several platform designs to give the opportunity for comparing and selecting a design. The platform lifecycle (comprising an era) is divided into epochs; each epoch is described by an epoch vector (e.g. set of engineering metrics), which defines its key exogenous factors that are describing the platform context. An epoch is a time period with a fixed context; characterized by static constraints, design concepts, available technologies, and articulated attributes (Ross 2006). The platform lifecycle described in section 2.2.3 is the core construct that designers use to characterize the phases of a system during its lifespan, from initial concept to end of life. The scheme of the adapted approach is shown in Figure 4-15. In this figure the point of exercising a mechanism is mapped. The approach behind exercising a mechanism is earlier described in this step.

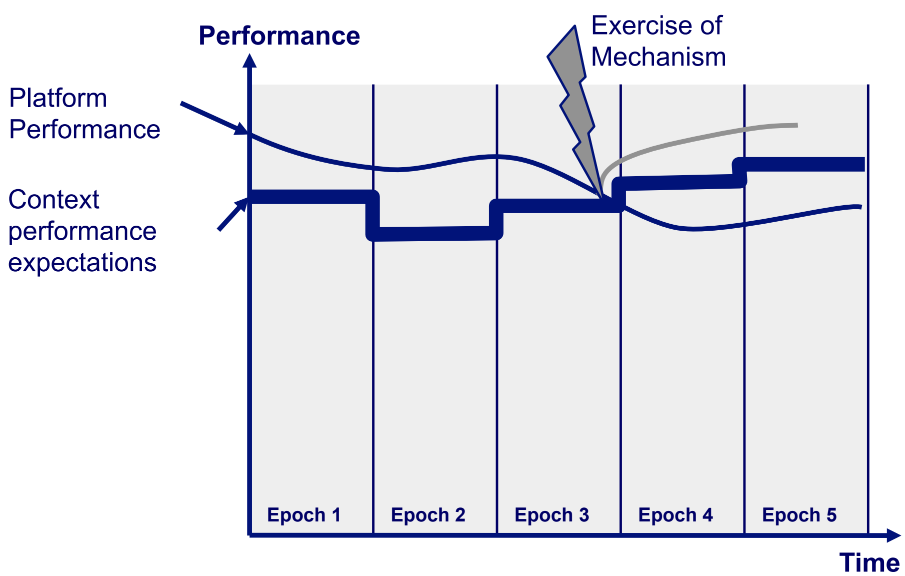


Figure 4-15: Adopted approach of epoch-era analysis (adopted from Rhodes and Ross 2010)

The functional performance is an important metric for evaluating the utility of a platform over the timespan of the lifecycle. As mentioned, in this thesis the platform's functional performance is related to the engineering metrics and the components.

The critical line is the border between the lower point of the bandwidth and the critical area. Fuzziness around this border can be used to have a buffer regarding to the functional platform performance. With a lax border, the functional performance of the platform is longer in an acceptable band.

The calculation of the functional performance can be done with following equation:

$$F_{EM_i}(t) = \begin{cases} 1, & (1 \pm k)\times EM_i(min) < EM_i(t) < \dfrac{EM_i(max)}{1 \pm k} \\ \dfrac{EM_i(t)}{EM_i(min)}, & EM_i(t) \leq (1 \pm k)\times EM_i(min) \\ \dfrac{EM_i(max)}{EM_i(t)}, & EM_i(t) \geq \dfrac{EM_i(max)}{1 \pm k} \end{cases}$$

Equation 4-6: Calculation of the fulfillment of each engineering metric

where $EM_i(t)$ is the output of the engineering metric i at time t and $F_{EM_i}(t)$ is the fulfillment of engineering metric i at time t, k describes the fuzziness around the minimal value $EM_i(min)$ and maximal value $EM_i(max)$ of engineering metric i. In some cases it makes sense to have separate fuzziness indicators around the minimum and maximum values of the engineering metrics and instead of connecting the engineering metrics to the time t they can be connected the different epochs e, because a epoch is defined by time constraints.

The term $(1 \pm k)\times EM_i(min)$ in Equation 4-6 describes the lower end of the engineering metrics bandwidth, where $(1 \pm k)$ indicates the fuzziness around this border. The term $\frac{EM_i(max)}{1 \pm k}$ describes the upper end of the engineering metrics bandwidth and $(1 \pm k)$ indicates also the fuzziness around this value. All values of the engineering metric, which are between the lower and the upper end of the bandwidth are satisfying the customer needs, so the fulfillment of the engineering metrics is 1, which is the highest possible value in this case. The fulfillment of the engineering metric is $\frac{EM_i(t)}{EM_i(min)}$ if the value of the engineering metric $EM_i(t)$ is below the lower end $(1 \pm k)\times EM_i(min)$ of the bandwidth. The fulfillment of the engineering metric is $\frac{EM_i(max)}{EM_i(t)}$ if the value of the engineering metric $EM_i(t)$ is bigger than the upper end $\frac{EM_i(max)}{1 \pm k}$ of the bandwidth. In both cases, the fulfillment is between smaller than 1, but bigger than 0.

The performance of the platform is calculated with Equation 4-7,

$$P_P(t) = \frac{1}{n}\sum_{i=1}^{n} w_i F_{EM_i}(t)$$

Equation 4-7: Calculation of the platform performance

where $P_P(t)$ is the platform performance at time t, n is the number of engineering metrics i, and w is the weighed importance of engineering metric i. The calculation of the platform performance is based on research of Simpson et al. (2006), which is described in more detail in section 2.3.3.1.

The performance of the product family can calculated with Equation 4-8, where the last part is identical with calculating the performance of the platform.

$$P_{PF}(t) = \frac{1}{r}\left\{\sum_{1}^{r}\left(\frac{1}{m}\sum_{j=1}^{m} w_j F_{EMC_j}(t) + \frac{1}{n}\sum_{i=1}^{n} w_i F_{EM_i}(t)\right)\right\}$$

Equation 4-8: Calculation of the performance of the product family

In Equation 4-8, $F_{EMC_j}(t)$ is the fulfillment of the engineering metric *j* at time *t* of the customized part in the derivative, *m* is the number of engineering metrics *j* in the related derivative, *r* is the total number of derivatives per platform.

The engineering metrics are the outputs of a platform that should satisfy the customer requirements and for every engineering metric there is an expected bandwidth of outcomes. The functional performance calculation of the platform is based on the engineering metrics and the different decision rules based on the fulfillment of the engineering metrics are documented in Table 4-4 and Equation 4-9.

Table 4-4: Evaluation of functional platform performance

Decision rules for mechanisms
If output of at least one engineering metric is below this bandwidth, then the functional platform performance is critical.
If output of every engineering metric is within the expected bandwidth, then the functional platform performance is good.
If output of at least one engineering metric is above this bandwidth and every other engineering metric is within the bandwidth, then the functional performance is more as good.

$$A = \begin{cases} Monitor\ Context, (1+k) \times EM_i(min) < EM_i(t) < \dfrac{EM_i(max)}{1+k} \\ Investigate\ exercising\ mechanism, \dfrac{EM_i(max)}{1+k} \leq\ EM_i(t) \leq\ (1+k) \times EM_i(min) \end{cases}$$

Equation 4-9: Decision rules for activities regarding the fulfillment of the engineering metrics

The resulted activity *A* in Equation 4-9 is based on the value of the engineering metrics, where all used variables are identical to the ones in Equation 4-6.

In the described bandwidth approach it is assumed that the customer satisfaction is the same for every value within the range from the minimal to the maximal value. In some cases this approach is not detailed enough. Methods, which can be more detailed and sophisticated, are the evaluation of the fulfillment of the engineering metrics with utility curves or quadratic loss functions.

After the decision is taken, which mechanisms are implemented because the implementation is adding value to the platform at anytime during the lifecycle, the changes of mechanisms on the set or specifications of the components have to be described and the QFD Matrix Phase II is needed to be updated, especially if there are new or updated components.

4.1.6 Step 6: Compare Product Platform Designs and Selection

In this step the platform designs investigated in the steps before are compared with the goal to select the final design of the platform. It is indicated in Figure 4-16 that Step 6 of the framework includes two inputs, three activities and also three outputs. It is also indicated that there is incoming connection from Step 5.

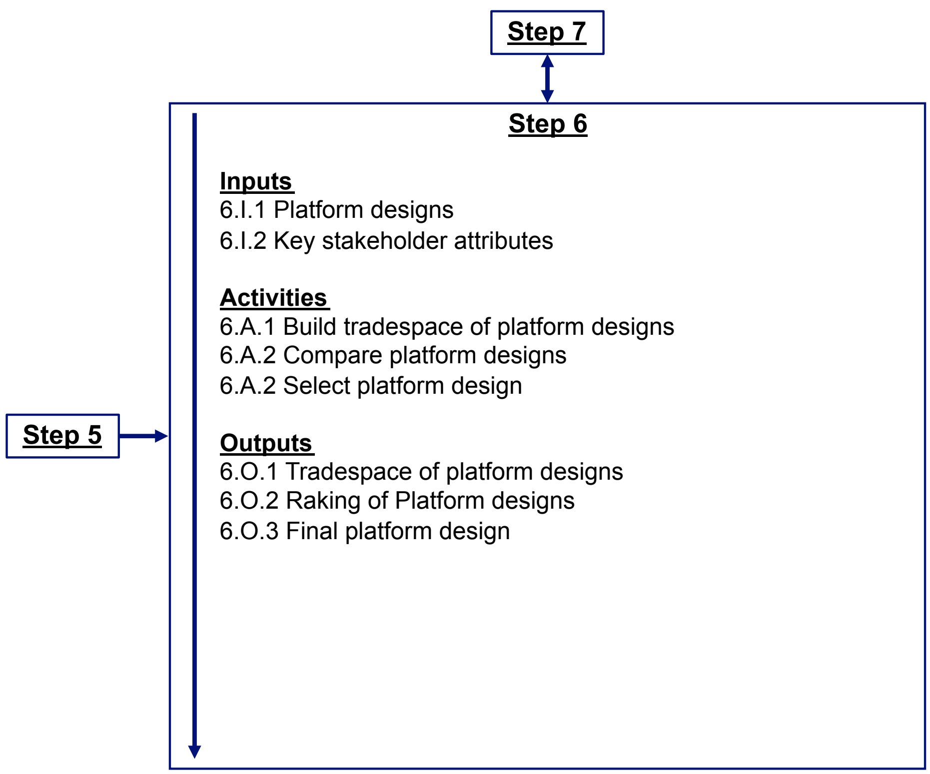


Figure 4-16: Overview Step 6 and connections to related steps of the framework

Wasson (2006) defined a tradespace as an area of evaluation or interest bounded by a prescribed set of boundary constraints that serve to scope the set of candidate alternatives, options, or choices for further trade study investigation and analysis. A typical tradespace plot used in MATE analysis proposed by Ross (2006) is the utility-cost space and the goal for design option selection is to find options at highest utility at a given cost level, which forms the pareto frontier of solutions. The pareto frontier is defined by Ross and Hastings (2005) as the curve (or multi-dimensional surface) along which an objective value must be traded for another and no mutually improving movement is possible.

Building a tradespace in this research means that based on the outcomes of the steps 1 – 5 a space with every design vector of the platform characterized by cost and performance is defined. The tradespace guides the ranking of the platform designs is based on stakeholder attributes, also the selection of the final design. The final design can than be modeled, for example, with model-based systems engineering methods like SysML for getting a better overview of the platform. The advantage is that a clear overview of the system is created, but it is a static model and the changes over time cannot be displayed in this static model. Based on the tradespace the different designs can be compared and the best suitable one can be selected.

4.1.7 Step 7: Review Selected Design and Monitor Context

The monitoring aspect in Step 7 allows reaction on uncertainty before the risk can take effect with the result that the system cause harm or the functional performance is below the context expected functional performance.

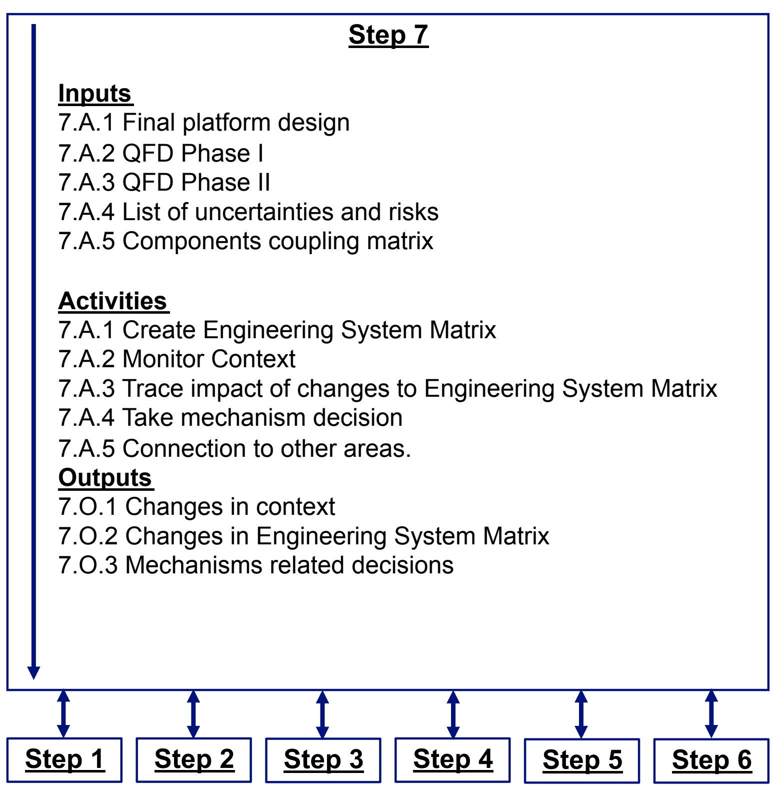


Figure 4-17: Overview Step 7 and connections to related steps of the framework

In Figure 4-17 the inputs, activities and outputs of Step 7 are illustrated. This step includes five inputs, five activities and three outputs. There are no specific incoming or outgoing connections at this step but there are all the bi-directional connections to the other seven steps.

The inputs of this step are the outputs of the earlier steps of the framework. As mentioned in section 4.1 and displayed in Figure 4-1 Step 7 is parallel to all the other steps of the framework. The goal is to monitor the product platform development project and more important, the actual and future context in which the product platform has to deliver the expected platform performance to be most successful.

For monitoring the context, documenting the changes and investigating the impact to the other elements of the platform it is proposed to use an engineering systems matrix, which is illustrated in Figure 4-18. Further information about it can be found in section 2.1.2 and the methodology behind the engineering systems matrix was described by Bartolomei (2007).

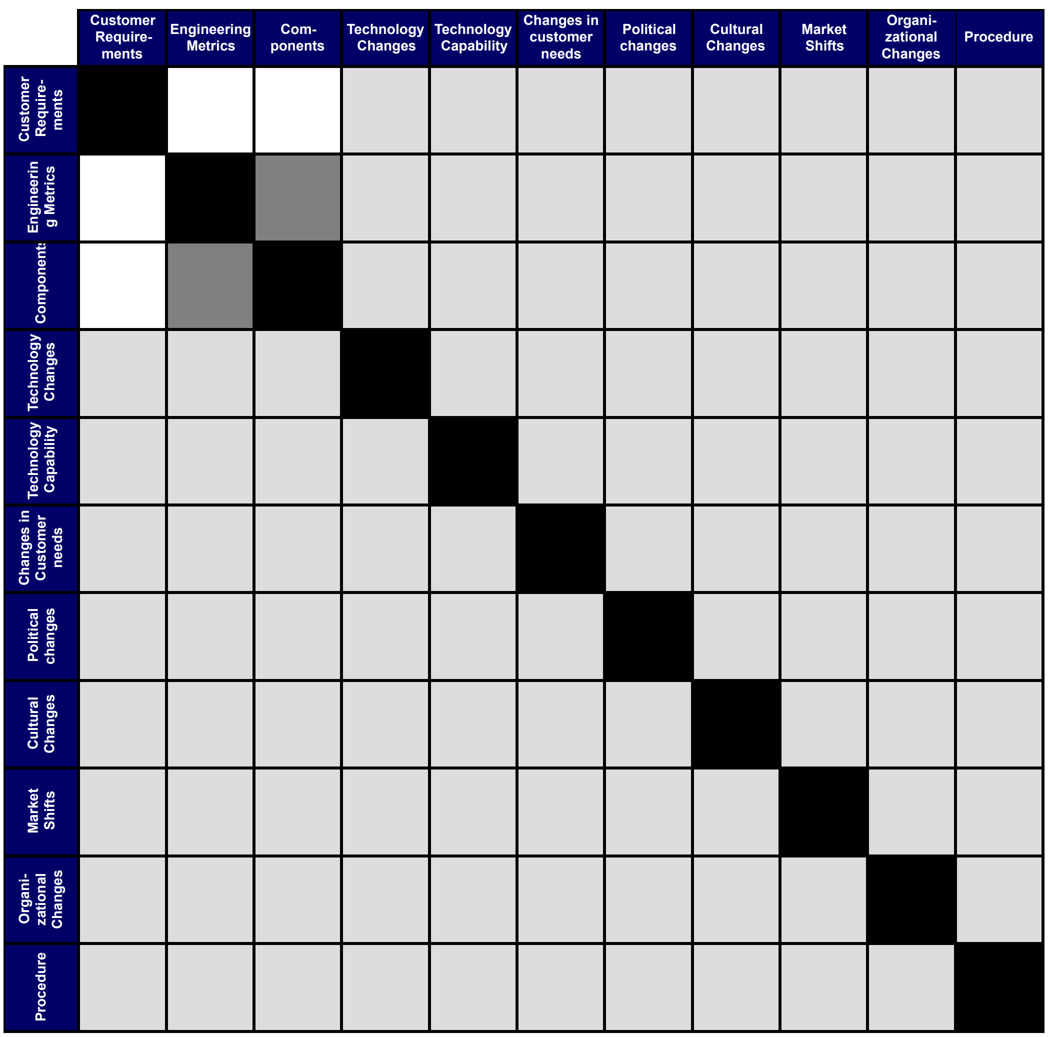


Figure 4-18: Engineering Systems Matrix (adapted from Bartolomei 2007)

As mentioned in section 2.1.2 the Engineering Systems Matrix (ESM) organizes information using a matrix structure that can be used to facilitate network and graph theoretic analysis and it is designed to show changes over time. The ESM is represented as an adjacency matrix with identical row and column headings. Thus, the diagonal represents the platform and the context elements and the off-diagonal cells represent the relationships between the elements. The black cellblocks along the diagonal represent a graph of a particular class of node; the specifications of the most nodes are described in the other steps of the framework. Customer requirements are described in Step 1 of the framework, as well as engineering metrics. The components are derived from the customer requirements and the engineering metrics and are described in Step 2. Technology changes, technology capability issues, changes in customer needs, political and cultural changes, market shifts and organizational changes are the different risk areas described in section 2.3 and investigated in Step 3 and 4 of this framework. Procedures are the actual activities, which are undertaken to influence the system, for example, reducing a specific risk by implementing a mechanism. The domains in the rows and columns can be expanded by additional entities if there are changes over time or if new ones are identified.

With the ESM it is possible to indicate all the connections between the different domains of a system. Based on this knowledge the decision about exercising mechanism or changing product platform can be investigated. The question is how to decide when to exercise the mechanism. Based on the results different indicators can be defined, which can guide the activity of exercising a mechanism. These indicators can be time or functional performance related. For example, there can a timespan between taking the decision of exercising a mechanism and adding the real value of the exercised mechanism to the platform; the length of the timespan can depend on the type of the mechanism and the type and complexity of the product platform.

The Engineering Systems Matrix can also be expanded to get the connection to other areas like production domains, which are also important for the overall value of the product platform. As mentioned in section 1.1 it is necessary to consider domains in production. These domains are related to cost, time and quality, for example, reduction of production costs because of commonality or change in cost for production, because of changes in the tools for production. A well planning and configuration for product platform development requires estimating expected financial benefits both in terms of savings due manufacturing, inventory, training, maintenance and revenues due to successful product performance in the market (Simpson and Tucker 2006).

4.2 Chapter Summary

The sections of this chapter cover the description of the framework, the application to an illustrative case is described in the next chapter. The framework provides a step-to-step process for investigating uncertainties and risks and providing a path how the identified uncertainties and risks can be mitigated. The first to steps are to creating the design of a platform, the methods used in these steps can be replaced with comparable ones to get the outputs, which are necessary for the other steps of the framework. All in all the activities within the steps are in some kind flexible that the framework can be adapted to different kinds of product platform projects. Due to the time limitations of this thesis research, the framework has been developed in a preliminary form. Future research can serve to evolve and further test this framework.

5 Illustrative Application of the Framework: iRobot® Case Study

In Chapter 4 a framework for the management of uncertainty is proposed; to show the value of the framework it is applied to an illustrative platform project. In this chapter all seven steps are applied illustratively to the iRobot cleaning product portfolio, some of the steps are described more detailed then others that is related to the constraints of the illustrative example. The iRobot product portfolio includes small robots for vacuum cleaning, floor washing and floor sweeping. The company also developed robots for pool cleaning and gutter cleaning, but these robots are not visibly based on the platform. Most of the data used in this case were taken from the iRobot company website, the iRobot product manuals (iRobot 2006a, 2006b, 2010) and the patent US 7636982 B2 (Jones et al. 2009). This example is not exhaustive and can vary significantly from the real iRobot products. The views expressed in this thesis are those of the author and do not reflect the official strategy or position of the iRobot Corporation. For a better understanding, a model was built in SysML, based on the results of the framework described in Chapter 4 and the data from the mentioned iRobot documents.

5.1 Case Study Background

As mentioned, the illustrative case study is based on the cleaning products of iRobot. iRobot has three different types of cleaning robots on the market, which are obviously built on the same platform. The first robot, which was on the market is called 'Roomba', it is illustrated in Figure 5-1. It is an autonomous robotic vacuum cleaner, and under normal operating conditions, it is able to navigate a living space and its obstacles while vacuuming the floor. The second one on the market is called 'Scooba'; it is an automated robotic floor washer. The 'Dirt Dog' was the third one on the market, it is a cleaning robot also based on the 'Roomba' platform, which replaces the Roomba's vacuum cleaner module with a series of brushes designed for cleaning up loose hardware and debris from workshop and garage floors.

Figure 5-1: iRobot Roomba vacuum cleaning robot (iRobot 2011)

In Figure 5-2 the structure of the real iRobot Roomba model is provided. It can be assumed, that the robots are based on a platform because of the separation of the robot into different modules. The exploded assembly drawing in Figure 5-2 would be the final structure a after the application of the framework proposed in Chapter 4.

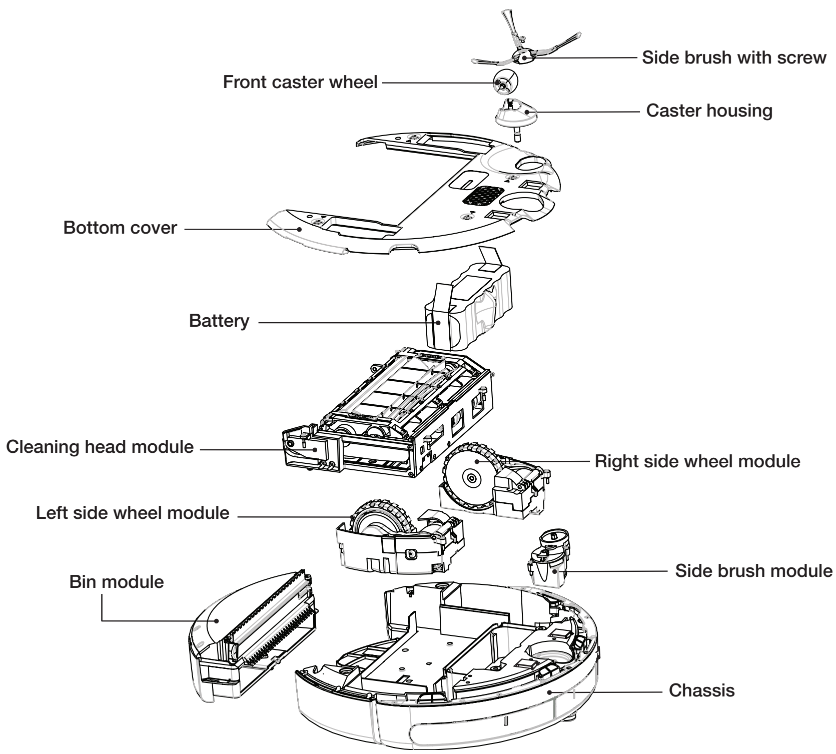


Figure 5-2: iRobot modules for vacuum cleaning robot ‘Roomba’ (iRobot 2010)

For the application in this chapter, the structure of the parts and the modules were simplified because the goal of this chapter is to demonstrate the value and the logic of the framework, it can be demonstrated with the simplified model. The ‘Roomba’ product was developed in a period of three years. During this period, several important principles guided the activities were identified. The principles demonstrate that it is reasonable to have a process like the framework for managing uncertainty described in Chapter 4. Jones (2006) mentioned some principles about the iRobot Roomba robot, like:

- The application comes first: Given the declaration about the application, customers can easily understand how the robot will benefit them, and how much this benefit might be worth. Furthermore, such a clear description of the application simplifies development-phase decision making (Jones 2006).

- Cost matters: In the marketplace, a robot must compete with every device or service provider that offers a function similar to the robot's function. Consumers reasonably compare the costs of accomplishing a task in different ways. Roomba competes with familiar items like vacuum cleaners and cleaning services, and customers know the price of a clean floor (Jones 2006).

- Only real-world testing can reveal the robot's flaws: Neither simulation nor careful thought can substitute for extensive testing. Had the development taken place only in the laboratory, Roomba would have been ill-equipped to meet the rigors of the real world (Jones 2006).

- "Usually" is unreliable: Given a tight development schedule and limited resources, there is always a temptation simply to ignore robot-challenging situations that will not "usually" occur (Jones 2006).

- Complexity kills: Complexity has a great capacity to kill budgets, schedules, and ultimately products. Robots are especially susceptible to complexity-induced difficulties. This is because the field is immature, meaning that few complexity-managing heuristics are yet in place, and also because the robot's various systems often interact in unexpected ways. The fewer systems a robot possesses the fewer surprises developers will face. Because of this, it is sometimes better to implement a needed feature by inventing a simple, new system than to add two or more familiar systems to the robot to accomplish the same purpose (Jones 2006).

5.2 Application of the Framework

This section is about the application of the framework described in Chapter 4. The application is more output based, so in most of the steps the outcomes are described in detail but less the activities, because the description of the activities can be found in Chapter 4.

5.2.1 Step 1: Identifying Customer Needs and Requirements

The first step of the framework, which is described in section 4.1.1 is applied to the illustrative example. The internal capabilities of the organization are set. In this case the available budget is more than enough, the development and production capabilities are also available to develop and produce the platform during a short time. The information about the market, which is documented in Table 5-1 was created from the marketing department of the company.

The lifetime was determined, the development period was set to 2 years and the time on the market of the platform was determined with 5 years. In time on market was combined with the market information to evaluate the importance of the different locations and surroundings.

Table 5-1: Table with illustrative market estimations for robot application locations and contexts

		Lifetime					
		Market entry	Market entry + 1 t	Market entry + 2 t	Market entry + 3 t	Market entry + 4 t	Market entry + 5 t
Location	Office	9	9	9	9	9	9
	Factory	3	3	3	9	9	9
	Warehouse	1	1	1	1	1	1
	Homes	9	9	9	9	9	9
	Airport	9	0	3	3	3	3
	Trainstation	1	3	3	3	0	0
	Gasstation	1	1	3	3	3	3
	Hospital	1	1	9	9	9	9
	Library	9	9	3	0	0	0
	Trains	9	9	3	3	1	0
	Labs	1	3	3	9	9	9
	Planes	3	1	9	3	9	3
	Garden	1	1	9	9	9	9
Context	Health	3	3	9	9	3	3
	Education	0	1	1	3	1	3
	Security	3	3	9	3	9	3
	Industry	3	3	3	3	3	3
	Private	9	9	9	9	9	9
	Military	9	9	9	9	9	9
	Sport	0	1	0	1	1	0

not important	0
low market potential	1
medium market potential	3
high marketpotential	9

As a result from the market estimation the company decided to offer solutions mainly for home, factory, and office locations, they also decided to provide it for the private and industrial context. Further research yielded to the use cases for vacuum cleaning, floor washing and floor sweeping. All these use cases should be developed on one platform. Furthermore, the market estimation was forecasting that the market for a grass-cutting robot is increasing in the third year. The company decided that it want to have the possibility to offer grass-cutting robot when the demand there. The use-cases are illustrated in Figure 5-3

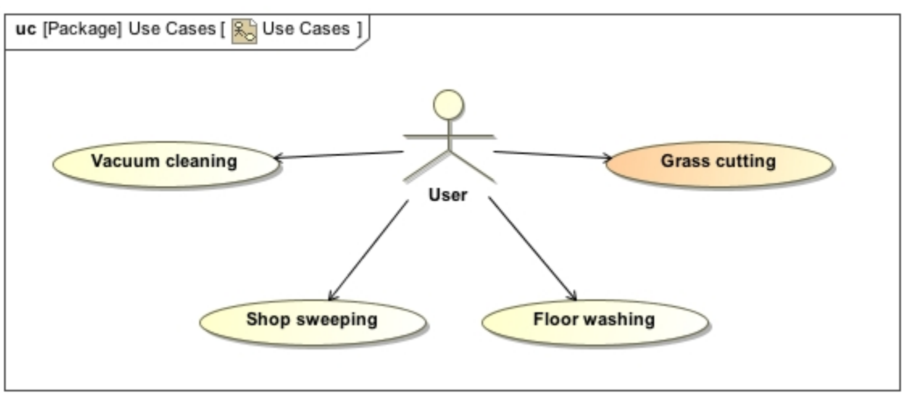


Figure 5-3: Use cases for cleaning and cutting robot

An illustrative set of customer requirements was collected. As shown in Table 5-2 the requirements were split in general requirements and requirements related to the different use cases vacuum cleaning, floor washing, pool cleaning and floor sweeping. The list of the customer requirements in Table 5-2 are already split into general requirements and requirements related to the use cases.

Table 5-2: List with a section of illustrativly identified customer requirements

Customer requirements	**General**	Cleans effectively
		Cleans under objects
		Covers entire space
		No hang ups
		Short clean Time
		Long run time
		Easy to maintain
		Inexpensive to operate
		No damage
		Deep cleaning ability
		Quiet during movement
		Movement on different floor materials
		Autonomous movement
	Vacuum cleaning	High vacuum power
		Excellent filter
	Floor washing	Dries floor effectively
	Floor sweeping	Load much debris
	Grass cutting	Collect grass
		Cut high grass

The grass cutting is colored different because the decision about this use case should be taken later but the platform should be flexible enough that it can be used for the grass cutting use case.

The next step is to develop the engineering metrics for the specification of the expected functional platform performance. For each engineering metric a bandwidth of values with a minimal and a maximal value has to be set. The step of defining the bandwidth of the engineering metrics is not a part of this illustrative example.

Related to the identified customer requirements listed in Table 5-2, engineering metrics were identified, which are shown in Table 5-3.

Table 5-3: List of illustratively identified engineering metrics

Engineering Metrics
Height Roboter (in)
Depth Roboter (in)
Width Roboter (in)
MTBF (h)
Bin volume (gal)
Weight (lb.)
Run time (h)
Speed (ft./s)
Reachable Distance (ft.)
Floor dry time (min)
Vacuum Volume (gal/min)
Airflow (cf/m)
Impuls to the surroundings (FΔt)
Noise (db.)
Vacuum lift (Pa)
Detection distance (ft.)
Dust holding capacity (lb.)
Charging time (min)
Communication distance (ft.)
Communication speed (MB/s)
Min size of dust particles to detect (in^2)
Rotational energy blade (J)

The QFD Matrix Phase I was created by combining the customer requirements (Table 5-2) and engineering metrics (Table 5-3). The result with the interconnections is documented in Figure 5-4, where the customer requirements are the headings of the rows and the engineering metrics are the headings of the columns. If there is indicated a 1 at a node it means that the engineering metric in the column is there for satisfying the customer requirement in the related row. In this figure the customer requirements and the engineering metrics for the grass cutting use case are already included.

			Engineering metrics																					
			Height Robot (in)	Depth Robot (in)	Width Robot (in)	MTBF (h)	Bin volume (gal)	Weight (lb.)	Run time (h)	Speed (ft./s)	Reachable Distance (ft.)	Floor dry time (min)	Vacuum Volume (gal/min)	Airflow (cf/m)	Momentum to the surroundings (FΔt)	Noise (db.)	Vacuum lift (Pa)	Detection distance (ft.)	Dust/Debris holding capacity (lb.)	Charging time (min)	Communication distance (ft.)	Communication speed (MB/s)	Min size of dust particles to detect (in^2)	Rotational energy blade (J)
Customer requirements	General	Cleans effectively										1	1	1			1	1	1					
		Cleans under objects	1	1	1													1						
		Covers entire space							1	1	1													
		No hang ups																1			1	1	1	
		Short clean Time								1			1											
		Long run time				1	1		1															
		Easy to maintain	1	1	1	1		1																
		Inexpensive to operate				1																		
		No damage						1		1					1								1	
		Deep cleaning ability												1			1							
		Quiet during movement														1								
		Movement on different floor materials								1														
		Autonomous movement				1															1	1		
	Vacuum cleaning	High vacuum power				1		1					1	1			1							
		Excellent filter				1													1					
	Floor washing	Dries floor effectively				1						1								1				
	Floor sweeping	Load much debris	1	1	1	1	1																	
	Grass cutting	Collect grass				1							1	1			1							
		Cut high grass				1																		1

Figure 5-4: Section of cleaning and cutting robot QFD Matrix Phase I

Creating the QFD Matrix Phase I was the last activity in Step 1 of the framework. In summarization the further outputs are project time constraints and the market segments. Also the bandwidth of the expected functional platform performance in form of the engineering metrics was defined after the customer requirements were identified. In the next step the platform designs are created based on the outputs of Step 1.

5.2.2 Step 2: Investigating Product Platform Designs and Bandwidth

As mentioned in section 4.1.2 this step is to develop different platform designs and to set up the bandwidth. Based on the QFD Matrix Phase I, which is displayed in Figure 5-4 in section 5.2.1, the QFD Matrix Phase II was developed as shown in Figure 5-5. As mentioned in the description of this step in Chapter 4, in this matrix, the engineering metrics are translated into component. The different components are written in the columns of the QFD Matrix Phase II in Figure 5-5, the rows are filled with the engineering metrics, which are already known from last step. Both, engineering metrics and components are mapped against each other, for indicating which engineering metric is an outcome of which component in the platform, or the other way round which component is in the platform for addressing which engineering metric. The QFD Matrix Phase II in Figure 5-5 is a section of the entire matrix of the cleaning and cutting robot platform, and it is only illustrated one set of components. The numbers in the nodes are indicating the strength between the engineering metrics and the components (1 = low strength between engineering metrics and components, 3 = medium strength, 9 = high strength). As mentioned in the description of this step in Chapter 4, one outcome of Step 2 is different platform designs. For a better manageability, all further steps of the application of this framework are based on this set of components and engineering metrics. It can be assumed that the framework is also working when it is applied to different sets of design vectors.

Engineering Metrics \ Components	Battery	Processor with on-board program memory and RAM	Antenna (Robot)	Communication processor	Wall sensor	Wheel drop sensor	Debris sensor	Cliff sensor	Bump sensor	Right wheel indoor	Left wheel indoor	Right wheel outdoor	Left wheel outdoor	Encoder right wheel	Encoder left wheel	Right wheel motor	Left wheel motor	Vacuum port large	Vacuum and heating port	Vacuum pump	Vacuum motor	Heating fan	Encoder heating fan	Heating coil	Side brushes indoor	Main brushes indoor	Side brushes outdoor	Main brushes outdoor	Side brushes washing	Main brushes washing	Main cutting blades	Side cutting blades	Encoder brushes/blades	Brush/blade motor large	Brush/blade motor medium	Power LED	User buttons	Indicator LED	Speaker	User button LED	Filter	Fill level sensor	Waste bin	Liquid tank
Height robot (in)	3		9		1	1	1	1	1	3	3	3	3			1	1	3	3	3	1	3		1	1	1	1	1	1	1	1	1		3	1						1		9	3
Depth robot (in)	3				1	1	1	1	1	3	3	3	3			1	1	3	3	3	1	3		1	9	9	9	9	9	9	9	9		3	1						1		9	3
Width robot (in)	3				1	1	1	1	1	3	3	3	3			1	1	3	3	1	1	3		1	9	9	9	9	9	9	9	9		3	1						1		9	3
MTBF (h)	3	3	1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bin volume (gal)																		1	1																								9	1
Weight (lb.)	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1	3	1	1	1	1	1	1	3	3	1	3	3	1	1	1	3	1	3	1	3	3
Run time (h)	9	1																		3	3	1		3										3	1									
Speed (ft./s)	1	1				1				1	1	1	1	1	1	1	1					3	1	1		3	3	3	3	3	3	3	1	3	3									
Reachable Distance (ft.)	1	1	3																																								3	3
Floor dry time (min)	1	1																	1			9	1	9																				
Vacuum Volume (gal/min)		1																3	3	9	0	0																			3		1	
Airflow (cf/m)		1																3	3	9	0	1																			3		1	
Momentum to the surroundings (F∆t)		1			9	1		1	9																1	1	1	1	1	1	1	1												
Noise (db.)										3	3	3	3			3	3	3	3	3	3	3			1	1	1	1	1	1	1	1		3	3				3		1			
Vacuum lift (Pa)		1																3	3	9	0	0																						
Detection distance (ft.)		1			3	3		9	9																																			
Dust/Debris holding capacity (lb.)		1																																							3		9	
Charging time (min)	9	1																																										
Communication distance (ft.)	1	1	9	9																																								
Communication speed (MB/s)		1	9	9																																								
Min size of dust particles to detect (in^2)		1					3																																		1			
Rotational energy blade (J)	1																								3	3	3	3	3	3	3	3		9	9									

Figure 5-5: Section of cleaning and cutting robot QFD Matrix Phase II

In this application the components are also set as the bandwidth of the platform in this case. The coupling matrix in Figure 5-6 is showing the connection between the different components of the platform. The activity of specifying the flows between the module and the architecture with values is not done in this illustrative example; only the interfaces are indicated as shown in the block diagrams of the different modules. The row and column headings in the coupling matrix are identical, the diagonal of the matrix represents the components of the platform and the cells off the diagonal represents the relationships between the components described by the row and column header. A node implies that there is a flow from the component indicated in the column header to the component indicated in the row header. The grey boxes on the diagonal are for showing the complexity of the modules and the connections within the modules.

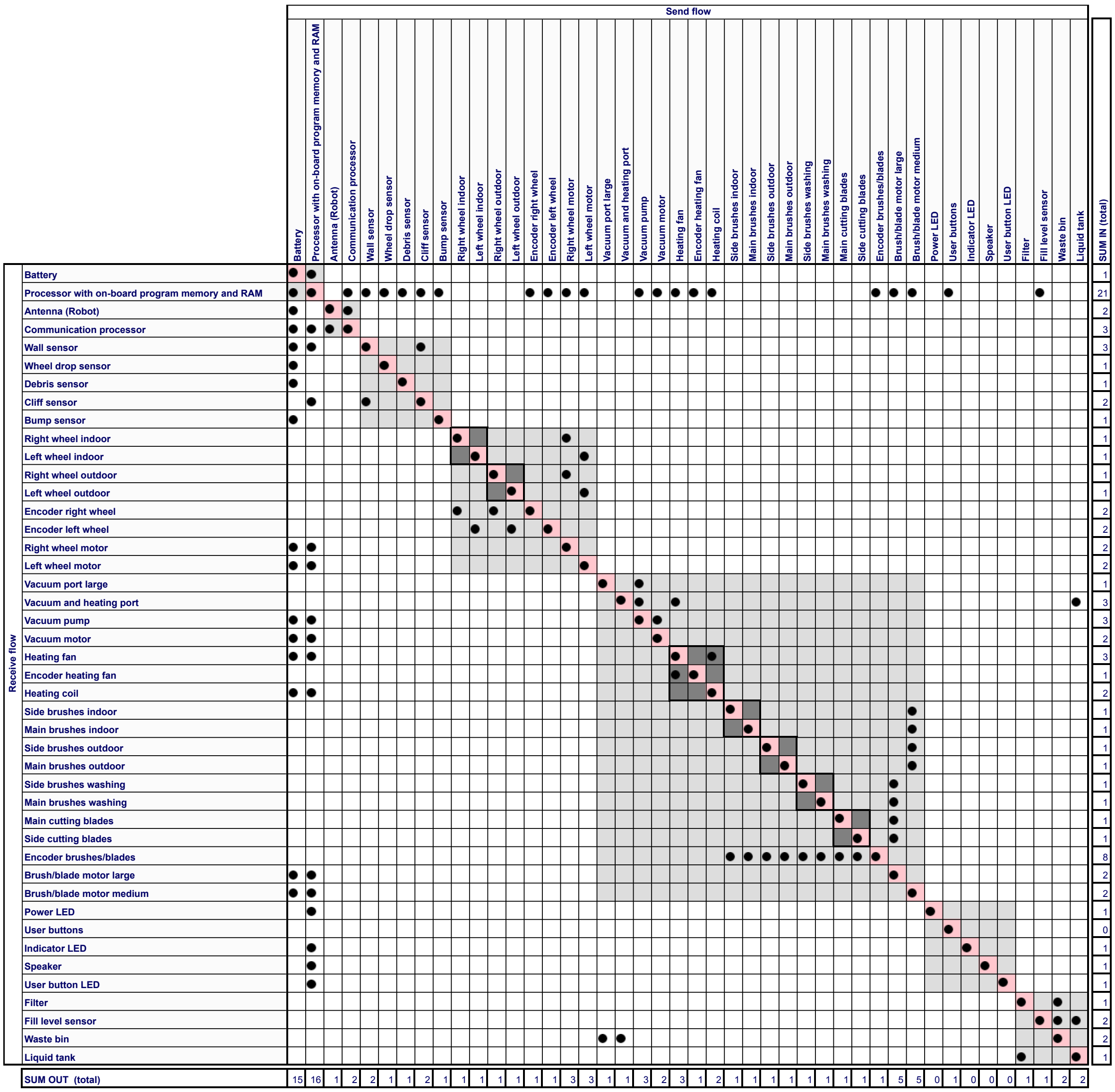


Figure 5-6: A section of the coupling matrix for the cleaning and cutting robot platform

The nodes besides or above a grey box are indicating the connection of the module to the other module or to in the architecture. In the last column the number of nodes is summarized, the sum is an indicator for the number of receiving flows per component. In the last row of the coupling matrix, the nodes of each column are summarized; the sum is an indicator for the number of sending flows per component. Both sums can be used for analyzing the sensitivity of a component in case of an uncertainty or risk. As it can be seen, the most connections are near the diagonal of the matrix and the connections to other modules are limited. The nodes need to be more specified, for example, the connections within a module and between different modules or the modules and the architecture are described more detailed in Figure 5-7 and Figure 5-8. In these two figures, the internal block diagrams of the communication module of the cleaning and cutting robot and the ones of the driving modules for indoor and outdoor use are displayed. Internal block diagrams were created for all modules of the product platform, the rest of the internal block diagrams and the rest of the SysML diagrams are attached in Appendix C. The yellow boxes in the internal block diagrams are the physical parts of a module, like ‘communication processor’ in the communication module. The green boxes with the arrows are the interfaces within the product, the direction of the arrow is indicating if it is an in- or outgoing flow. The red boxes with the arrows are the interfaces to the environment of the cleaning or cutting robot. The interfaces to the other modules or the architecture are normally arranged on the left side of the internal block diagram, the interfaces to the environment are on the bottom line of the internal block diagram. An example for an internal interface is the connection the communication processor to the main processor in the architecture of the robot. An example for an external interface is the data flow, which is send or received from the antenna or more visible the light signals sent to the user by the LEDs in the user interface module.

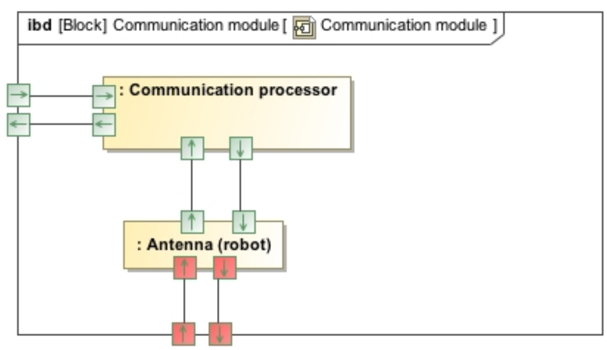


Figure 5-7: SysML block diagram of the communication module of the cleaning and cutting robot

In Figure 5-8 the internal block diagrams of the driving modules are illustrated. In this case some of the parts are the same in each modules and others are replaced to deliver the best performance in each use case replace some.

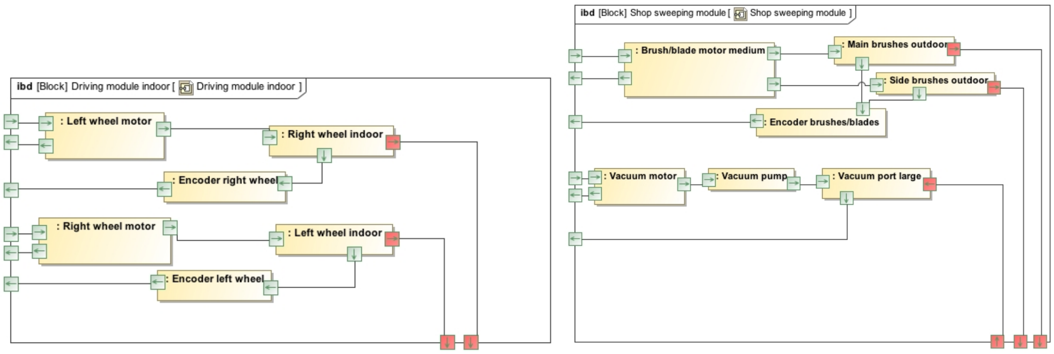


Figure 5-8: SysML block diagrams of the driving modules for indoor and outdoor use

In this example, the modules are modular itself, that means parts can be easily replaced by other parts for addressing the customer needs in each use case. Based on the outcomes of the first two steps, like customer requirements, engineering metrics, and the matrices based on these, the next step for identifying and analyzing the uncertainties and risks can be described.

5.2.3 Step 3: Uncertainty and Risk Assessment

In this section, the results of the uncertainty and risk assessment process as described in section 4.1.3 are documented. In this illustrative example, all of the uncertainties were identified in a brainstorming. In Table 5-4 the identified uncertainties are listed, where the identified uncertainties are allocated to the different risk areas. In this example 11 different uncertainties were identified in total, and each is described with a short sentence in the second row of the list in Table 5-4. The specification of the type of uncertainty is filled in in the third column of the table and the last two columns are for determining the likelihood verbal and with numbers.

Table 5-4: List of identified uncertainties over the platform lifecycle

Risk Area	Uncertainty		Likelihood	
	description	type		
Technology change	New battery technology	known unknown	medium	3
	Better technology for debris sensor	known unknowns	medium	3
Technology capability	Capability of battery technology not as expected	lack of knowledge	medium	3
	Electric wheel motor over capability in cutting use case	lack of knowledge	medium	3
Customer requirement	New use case (usage in clean environment)	lack of knowledge	medium	3
	Longer run time expected	lack of knowledge	medium	3
Market/Business	Supplier quit	unknown unknown	low	1
	End of Lifecycle of a component	lack of knowledge	high	9
Political/Cultural change	New regulatory regarding battery recycling	known unknowns	high	9
	Change in regulatory for wireless communication	known unknowns	medium	3
Organizational change	Outsourcing of manufacturing blades	unknown unknown	medium	3

All the data in Table 5-4 are also a part of the risk analysis and evaluation. In an expanded template, the severity of each uncertainty can be specified in general for the entire platform, in this example it is not done, because the severity is described as a step of the risk evaluation in each component and engineering metrics, that is described in the next section.

5.2.4 Step 4: Epoch Description and Analysis

All the indicated uncertainties of Step 3 are combined to the uncertainties of one epoch. This is the only epoch, which is part of the further investigations, because it is an illustrative example and some other outputs are also seen as static in this thesis. In a real case, it is absolutely valuable to take several epoch descriptions into account for the investigation of the functional performance of the product platform.

The output of this step of the framework is a list with the critical parts of the platform. For getting the list, the two paths for identifying the critical parts as described in section 4.1.4 are applied to the illustrative example. All values are normalized in the risk and sensitivity analysis to reduce the confusion and to get results that are comparable. The impact of each uncertainty to the components and the engineering metrics was measured on a 1/3/9 scale, were '1' is low impact (impact of risk/uncertainty negligible, no treatment but future monitoring needed), '3' is medium impact (impact in 'grey' area, treatment decision after taking into account costs and benefits and balancing opportunities against potential consequences), '9' is high impact (impact of risk/uncertainty absolute intolerable, treatment is essential whatever its cost)

Table 5-5: Section of risk calculation in components of the robot platform because of the uncertainties

Uncertainty	Risk in Components																
description	Battery	Processor with on-board program mer	Antenna (Robot)	Communication processor	Wall sensor	Wheel drop sensor	Debris sensor	Cliff sensor	Bump sensor	Right wheel indoor	Left wheel indoor	Right wheel outdoor	Left wheel outdoor	Encoder right wheel	Encoder left wheel	Right wheel motor	Left wheel motor
New battery technology	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Better technology for debris sensor	0	0	0	0	3	3	9	3	3	0	0	0	0	0	0	0	0
Capability of battery technology not as expected	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Electric wheel motor over capability in cutting use case	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	27
New use case (usage in clean environment)	0	0	0	0	0	0	0	0	0	9	9	0	0	0	0	0	0
Longer run time expected	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3
Supplier quit	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0
End of Lifecycle of a component	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
New regulatory regarding battery recycling	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Change in regulatory for wireless communication	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0
Outsourcing of manufacturing blades	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MAX risk per comonent	81	0	0	27	3	3	9	3	3	9	9	0	0	0	0	27	27
Normalized MAX risk per comonent	1,00	0,00	0,00	0,33	0,04	0,04	0,11	0,04	0,04	0,11	0,11	0,00	0,00	0,00	0,00	0,33	0,33

The first path is to identify the critical parts by tracing the impact of the uncertainty directly to the components of the platform. The risk in the components is calculated by multiplying the likelihood of each uncertainty with the impact of each uncertainty to every component. The table with the impacts of each uncertainty per component is not illustrated but the computed risk per component is illustrated in Table 5-5.

As described in Chapter 4, the goal of the second path of identifying critical components via the impact of the uncertainties to the engineering metrics.

In the next step of path two, the risk in the engineering metrics is calculated by multiplying the likelihood of each uncertainty with the impact of each uncertainty to every engineering metric. The table with the impacts of each uncertainty per engineering metric is not illustrated but the computed risk per metric is illustrated in Table 5-6. As it could be seen in this table, in this illustrative example vacuum volume and vacuum lift are the both engineering metrics with the highest risk, followed by MTBF, run time, reachable distance, minimal size of dust particles to detect, weight, speed, momentum to the surroundings, noise, charging time, and communication distance and speed.

Table 5-6: Calculation of risk in the engineering metrics

Uncertainty	Engineering metrics																					
description	Height robot (in)	Depth robot (in)	Width robot (in)	MTBF (h)	Bin volume (gal)	Weight (lb.)	Run time (h)	Speed (ft./s)	Reachable Distance (ft.)	Floor dry time (min)	Vacuum Volume (gal/min)	Airflow (cf/m)	Momentum to the surroundings (FΔt)	Noise (db.)	Vacuum lift (Pa)	Detection distance (ft.)	Dust/Debris holding capacity (lb.)	Charging time (min)	Communication distance (ft.)	Communication speed (MB/s)	Min size of dust particles to detect (in^2)	Rotational energy blade (J)
New battery technology	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Better technology for debris sensor	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Capability of battery technology not as expected	0	0	0	27	0	0	27	3	9	3	0	0	0	0	3	0	0	9	0	0	0	9
Electric wheel motor over capability in cutting use case	0	0	0	27	0	0	0	9	27	0	0	0	0	9	0	0	0	0	0	0	0	0
New use case (usage in clean environment)	0	0	0	27	0	9	0	0	0	0	0	0	9	0	0	0	0	0	0	0	27	0
Longer run time expected	3	3	3	3	3	3	27	0	3	0	0	0	0	0	0	0	0	3	0	0	0	0
Supplier quit	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	9	0	0
End of Lifecycle of a component	0	0	0	27	0	0	0	0	0	0	81	0	0	0	81	0	0	0	0	0	0	0
New regulatory regarding battery recycling	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Change in regulatory for wireless communication	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	0	0
Outsourcing of manufacturing blades	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MAX risk per engineering metric	3	3	3	27	3	9	27	9	27	3	81	0	9	9	81	0	0	9	9	9	27	9
Normalized MAX risk per engineering metric	0,04	0,04	0,04	0,33	0,04	0,11	0,33	0,11	0,33	0,04	1,00	0,00	0,11	0,11	1,00	0,00	0,00	0,11	0,11	0,11	0,33	0,11

The next step is to calculate the risk in the components of the platform because of the risk in the engineering metrics. This is done by combining the maximum risk per engineering metric with the coupling strength between the engineering metrics and the components documented in the QFD Matrix Phase II. The result of the second path is a list of the components with the biggest risks. A combination of this list and the outcome of the first path are illustrated in Table 5-7.

Table 5-7: Section of critical parts in platform because of direct and indirect impact of uncertaitnies, normalized values.

	Direct impact of uncertainty	Risk because of risk in engineering metrics
Battery	1,00	0,33
Vacuum motor	1,00	0,11
Vacuum pump	0,04	1,00
Communication processor	0,33	0,11
Filter	0,11	0,33
Right wheel motor	0,33	0,04
Left wheel motor	0,33	0,04
Main cutting blades	0,33	0,04
Side cutting blades	0,33	0,04
Vacuum port large	0,00	0,33
Vacuum and heating port	0,00	0,33
Debris sensor	0,11	0,11
Wall sensor	0,04	0,11
Bump sensor	0,04	0,11
Right wheel indoor	0,11	0,04
Left wheel indoor	0,11	0,04
Heating coil	0,04	0,11
Side brushes indoor	0,11	0,04
Main brushes indoor	0,11	0,04
Processor with on-board program memory and RAM	0,00	0,11

The list in Table 5-7 is needed for investigating the implementation of mechanism in the next step of the framework.

5.2.5 Step 5: Assess Functional Performance of Platform Designs

In this step, mechanisms for the identified critical parts are identified and investigated, as described in section 4.1.5. When a decision is taken, to implement one or more mechanisms, the QFD Matrix Phase II need to be updated, because otherwise the system description is not complete. The actual QFD Matrix Phase II is also needed for a successful analysis and evaluation in case of the occurrence of further uncertainties and risks. As seen in the list in Table 5-7, the top five parts identified as most critical are the battery, the vacuum motor, the vacuum pump, the communication processor and the filter in the robot platform. Mechanisms to reduce the risk in the critical components can be standardized interfaces for the battery to have the ability to replace it with another type, or to build in a bigger construction space for having the option to include a bigger battery or a second one. Same kinds of mechanism can be investigated for the other critical parts of the platform.

5.2.6 Step 6: Compare Product Platform Designs and Selection

This step is the application of the description in section 4.1.6; the goal is to create the tradespace with the different platform designs to compare each design related to functional performance and cost. In a real example, the changes of the utility of the designs over time can be traced. Based on this results a ranking of all designs can be created related to certain attributes like which design delivers the expected functional performance with the assumed context changes over the whole platform lifecycle.

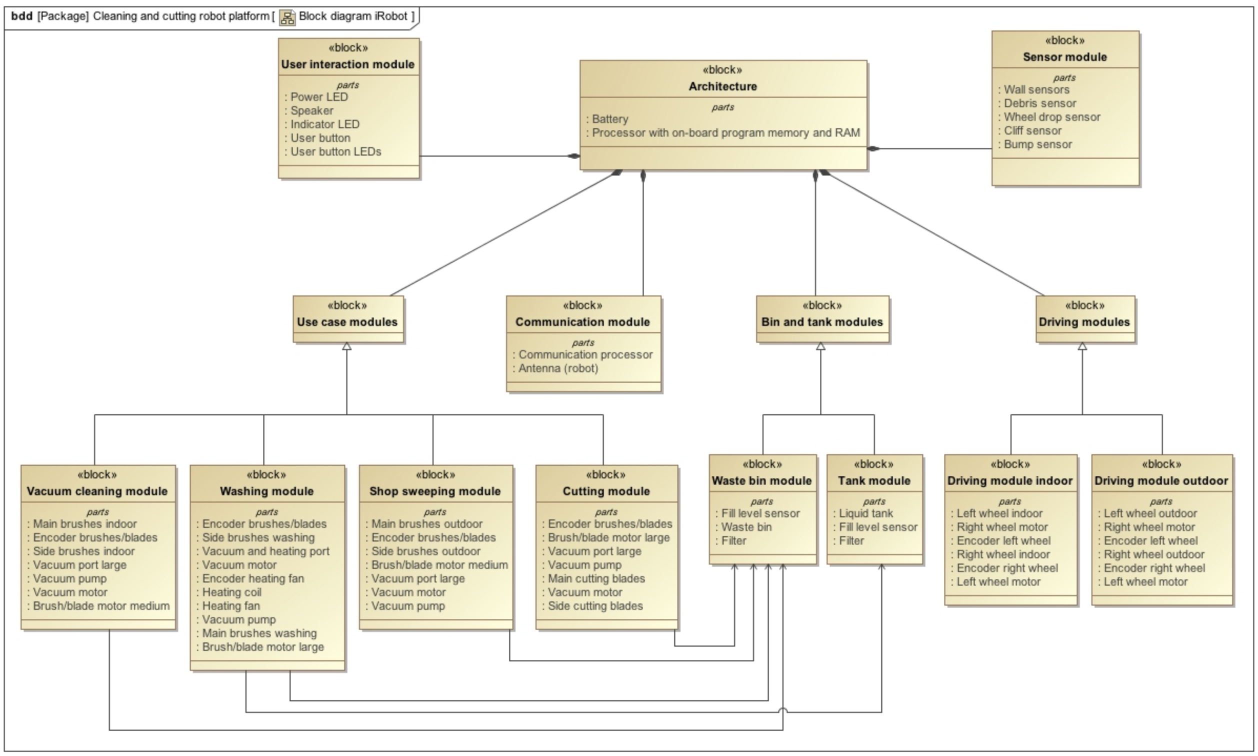


Figure 5-9: Modularization of the final platform components

The tradespace exploration is not described in this illustrative example, because only one design was part of the investigations and only one epoch was considered. For illustration, it is assumed that all the tradespace explorations and rankings of the designs were done during the process and the key stakeholder have decided to take the final design of the platform which is illustrated in Figure 5-9.

In Figure 5-10 a tradespace chart is documented, it is only an illustrative example of a tradespace chart with the utility of the design on the x-axis and the costs on the y-axis. Each dot in this chart is representing a unique design of the cleaning and cutting robot platform. Each color is representing a pattern of design vectors, which are related to the type of a component, for example. Type of a component means there are several variants of components, for example, several types of batteries, or sizes of debris bins in the portfolio of components that is investigated for use in the robot.

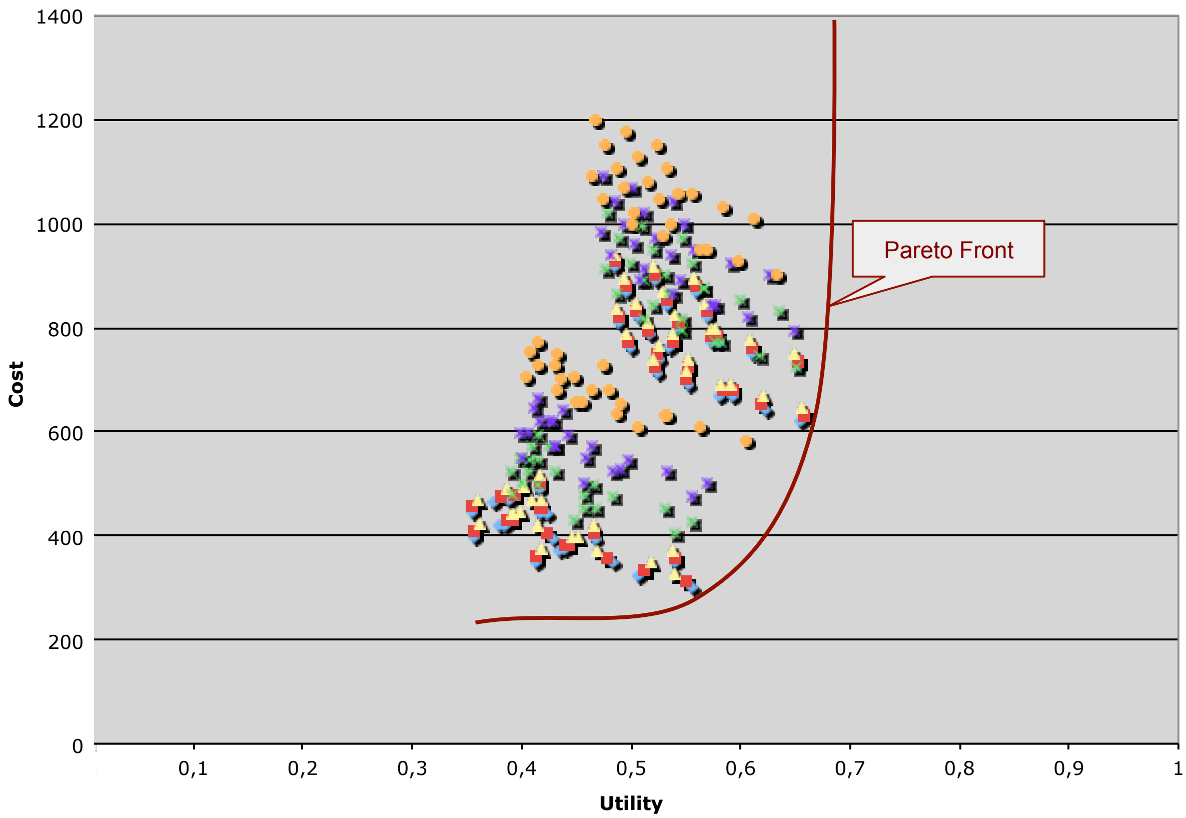


Figure 5-10: Illustrative tradespace chart

The red line in the tradespace chart in Figure 5-10 represents the pareto front, as indicated. As it can be seen when following the line of the pareto front from the left to the right side, at some point changing the designs is only increasing the costs but not adding utility to the system. It can be useful to have this information while taking a decision in the design process of a platform.

5.2.7 Step 7: Review Selected Design and Monitor Context

This step is to review the selected design but more important to monitor the context for future changes. As mentioned in section 4.1.7, the engineering systems matrix can be build during the procedure of the other steps. Outputs of these steps like the final QFD matrices phase I and II, the coupling matrix and the mappings of the impact of the uncertainties to the engineering metrics and the components are parts of the engineering systems matrix. The engineering systems matrix can be used for monitoring, for example, by adding further uncertainties and investigating the impact on the existing parts of engineering systems metrics. The investigation can be done by using the nodes of each row and column and filling them with data like coupling strength or level of impact. The engineering systems matrix is a high level perspective, which is providing a view of connections and flows in a system and it can be better to go to a lower level of the system, for conducting further analysis, what can be done, for example, better on the QFD matrices level or on the uncertainty impact matrices.

A part of this step is to monitor the context to find the right moment or timespan for exercising a mechanism. Because the mechanisms are described on a high level in Step 5 and no detailed analysis was done, no indicators are specified in this step of the illustrative case study.

The activity in this step to connect the framework to other domains, like production or marketing, not described in the illustrative example. It is essential in a real project to build the interfaces to other domains and investigate the uncertainties and the impact of the framework results in these domains. As mentioned in the description of the framework in Chapter 4, Step 7 is running parallel to all the other steps, so it is useful to analyze and evaluate the impact of the framework outcomes to the other domains early for getting a more optimized result. If is done to late, it can happen that the uncertainties and risks related to the functional performance are mitigated, but there are problems in the other domains because they were not considered while optimizing the platform for delivering the expected functional performance over time. An illustrative example is the implementation of a real option in the body in white of an automobile. In this scenario a car is designed for the European market because the market potential for the type of the designed car is very high in this area. A result of the market forecast is, that in several years the market potential in an Asian country can be also very high for a similar car. The only difference is, that the customers in Asia want to have a car, which is significantly higher than the European version, because they want to show that they can afford a bigger car. The market forecast is a kind of unsecure but there is no time for further market analysis and evaluation, because the car for the European market would be delayed if the development would be paused until the data about the market potential in Asia are clarified. The managers decided that they want to have the opportunity to sell the car also in Asia so the engineers got the instruction to build a body that is flexible for adjustments. One idea to build in the flexibility into the platform is a real option to have the ability to change the side panels of the car body. Considering only the delivery of functional performance, which is expressed in this example by the fulfillment of the customer needs concerning the height of the platform derivative, the conclusion can be that it is a good solution to implement a real option in the design of the car to change the side panels of the car body for the European market with higher side panels for the Asian market. The result concerning the value of the real option can be significantly different, if other domains like production are also considered. In this case, the tools and the machinery for molding the side panels are very expensive and the determination of the real option value can be less with this information. This example is providing, that it can be risky if the analysis and evaluation of the mechanisms only the fulfillment of the functional performance is taken into account.

5.3 Chapter Summary

This chapter has shown the application of the framework, because of the simplifications of the modules and the facts that only one designs was considered and only one epoch description was part of this illustrative example some activities could not been done in this example. But the application has shown the value of the framework that can be added to several platform projects by adjusting the framework to the individual development project.

As mentioned, the framework was only applied to one epoch of the platform lifecycle because of the thesis time limitations. For a holistic view, the framework can be applied and repeated to the other epochs of the platform lifecycle with the result of a more detailed evaluation of the platform performance. In a real case, it is absolutely valuable to take several epoch descriptions into account for the investigation of the functional performance of the product platform. A result of the application to further epochs is the evaluation of the platform performance over time; also the influence of mechanisms can be investigated over time. A huge advantage is that the product platform design with the highest value robustness in the investigated epochs can be chosen.

6 Future Work, Conclusions and Contributions

This Chapter covers the areas of future work, the research conclusions and contributions. The areas of future work are split into modeling approaches and future research topics regarding uncertainty management in platform development.

6.1 Future Work

In the framework application, a model of a platform was described with a SysML model developed after creating a model with Excel-matrices by hand. This is a good approach to create a view of the complexity of a product platform. Both built models are static, what means it is hardly possible to trace changes and the impact of this changes to the system over time. Another limitation of these models is the capability of investigating different designs at the same time. It is possible, but it is related to a lot of complexity and manual evaluation. A possibility to address the limitations of the static models can be the implementation of product platform designs with a method like Multi-Attribute Tradespace Exploration (MATE) (Ross 2006). In other research projects, systems were modeled with MATE for comparing different designs in a tradespace of utility and costs. With these models it is also possible to investigate the impact of changes in the design context to the tradespace.

The focus of the thesis is more on product related uncertainties like changes in technologies and customer requirements. For the holistic uncertainty management during the platform lifecycle also production and organizational domains can be considered with more detail in future research. For example, what are the uncertainties in an organization, especially when a firm switches from single product development to platform development. More detailed questions here can be what is the best transformation of the development organization or what is the impact of the hierarchy on the functional platform performance. Production related issues are also important, especially if the tools for production are extremely valuable. An example is provided in section 5.2.7, if a car manufacturer builds a Real Option in the platform to stretch the height of the car, it can be very valuable to address changes in customer needs. The result of the analysis of implementing the Real Option can be positive if the production is ignored. The picture can be totally different if domains like production are not ignored, because it is cost intensive to develop and build all the tooling for molding the parts of the body. So if this part is ignored the analysis of the value by implementing the mechanism can be incorrect.

Further research in areas of cost models is also valuable, especially regarding to the implementation of mechanisms. It is hard to find done research, which is proposing a way to value the mechanisms that can be implemented in a platform. Most of the industries are sensitive related to costs of their products, so it is necessary to have a fundamental value and cost analysis of each mechanism over a lifecycle of a product platform before implementing it.

Another area regarding the mechanisms can be the investigation of the timeline of exercising the mechanisms. As mentioned in this thesis in some cases it can be hard to find the indicators, which can be recognized later in the lifecycle that lead to the decision of exercising a mechanism. Besides it is also useful to investigate the timespan between taking the decision of exercising a mechanism and realizing the value of the mechanism in the product platform. Based on this it can be interesting to investigate the value related to the fulfillment of functions, cost and time aspects of the exercised mechanism and the platform over time, especially what happens when there occur unexpected changes in addition to the investigated ones.

6.2 Conclusions

The conclusion of the analysis is, that a lot of research related to product families was done but less related to the platform and uncertainty management in platforms. A conclusion of the empirical case study is, that most of the problems occurred because to less knowledge about uncertainties and risks at the front end of the development process. If more details about risks and uncertainties were known, the platform projects would have been more valuable for the customers as well as for the companies.

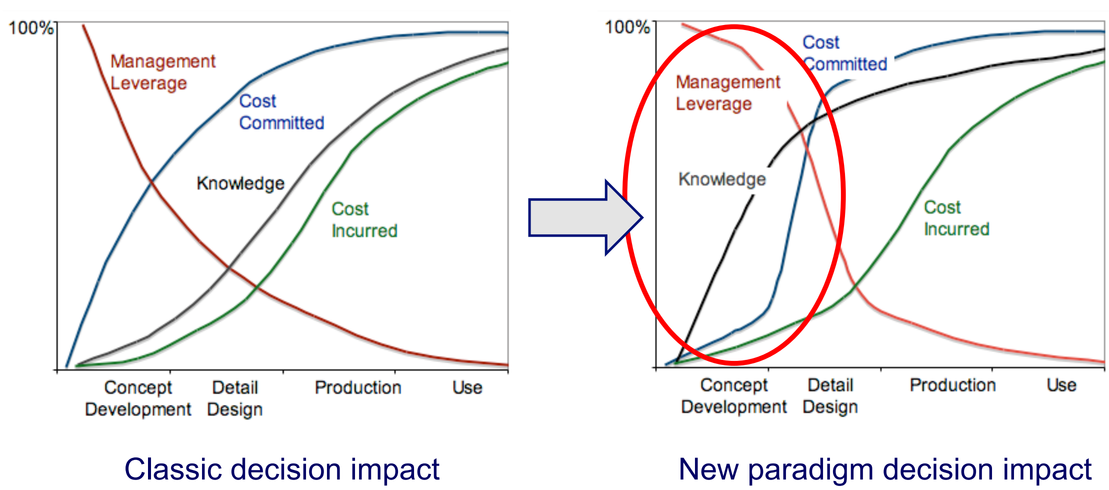


Figure 6-1: Changing the picture (Rhodes and Ross 2010)

This conclusion is coinciding with research results of the Systems Engineering Advancement Research Initiative (SEAri 2010) that increased knowledge in concept development allows better decisions and lead to a more stable platform design. As shown in Figure 6-1 it is important to gain as much as possible knowledge at the front-end of the platform development.

As it is shown in Chapter 5 the application of the framework creates value through identifying the critical parts and giving the opportunity to implement mechanisms. The framework is conceptual and applied to an illustrative example, a research topic in the future can be the application of the framework to a real platform project with adjusting the framework after gathering the real data from the application of it to a real platform project.

6.3 Contributions

In this research the knowledge about current approaches in systems engineering for platform development was summarized and later integrated in the framework. Knowledge about systems description like the engineering systems approach and the quality function deployment were analyzed for the usability in product platform development. Existing definitions of platform terminology were summarized and combined to a definition of a platform, which can be used for several levels of a system as well for differential types of platforms in various industry divisions.

With the empirical case study, evidence was added by investigating various real platform projects. Problems in different risk areas because of several types of uncertainties were identified and analyzed. The consequences because of these risks in the architecture, the modules and the interfaces of the platform were investigated and documented.

Based on the results of the empirical case study a preliminary framework was developed that can be used for the investigation and management of uncertainties. Different proven methods, which were analyzed and summarized, were combined and enhanced with further details to provide a conceptual framework that can be used for guidance of creating platform designs based on the customer requirements. Furthermore the framework provides guidance for the assessment of uncertainty and risk in platform projects.

The conclusions from the illustrative example can lead to the decision that the framework is beneficial for managing uncertainty and risk in product platforms. The application of the framework has provided a list of critical parts in the platform as a result of the process. For the critical parts different mitigation methods were recommend to ensure the delivery of the functional performance, expected by the context.

Bibliography

Babbie, E., 2007, "The Practice of Social Research", Belmont, CA, Wadsworth.

Bahill, A., Smith, E., 2009, "An Industry Standard Risk Analysis Technique", Engineering Management Journal, Vol. 21, No. 4.

Baldwin, C. Y., Clark, K. B., 2000, "Design Rules: The power of modularity", Cambridge, MA, The MIT Press.

Bartolomei, J., 2007, "Qualitative Knowledge Construction for Engineering Systems: Extending the Design Structure Matrix Methodology in Scope and Procedure", Doctor of Philosophy Dissertation, Engineering Systems Division, MIT.

Beesemeyer, J.C., Fulcoly, D., 2010, "Characterizing Evolvability for Engineering Systems", SEAri Annual Research Summit, MIT.

Caffrey, R., Simpson, T., Henderson, R., Crawley, E., 2002, "The strategic issues with implementing open avionics platforms for spacecraft", IEEE Aerospace Conf., IEEE-434-02. Big Sky, MT.

Cardin, M., Nuttall, W., de Neufville, R., Dalgren J., 2007, "Extracting Value from Uncertainty: A Methodology for Engineering Systems Design", 17th Annual International Symposium of the International Council on Systems Engineering (INCOSE), San Diego, CA.

Collier, D.A., 1981, "The measurement and operating benefits of component part commonality", Decision Sciences, Vol. 12, No. 1.

de Neufville, R., 2010, "Risk and Decision Analysis", Class Notes ESD.71, Massachusetts Institute of Technology, Fall Term 2010.

de Neufville, R., 2004, "Uncertainty Management for Engineering Systems Planning and Design", Engineering Systems Symposium, 2004, Engineering Systems Division, MIT.

de Neufville, R., 2003, "Real Options: Dealing with uncertainty in systems planning and design," Integrated Assessment, Vol. 4., No. 1.

de Weck, O., Eckert, C., 2007, "A Classification of uncertainty for early product and system design", ICED International Conference on Engineering Design, Paris, France.

DoD, 2008, "Systems Engineering Guide for Systems of Systems", Department of Defense, Version 1, August 2008.

Earl, C., Eckert, C., 2005, "Complexity", in Design process improvement – a review of current practice, London, UK, Springer.

Feitzinger, E., Lee, H., 1997, "Mass customization at Hewlett-Packard: the power of postponement", Harvard Business Review, Jan – Feb 1997.

Fricke, E., Schulz, A., 2005, “Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout Their Entire Lifecycle”, Systems Engineering, Vol. 8, No. 4.

Gonzalez-Zugasti, J., Otto, K., Baker, J., 1999, “Assessing value for product family design and selection”, 25th Design Automation Conference, ASME Design Engineering Technical Conference, Las Vegas, NA.

Gordon, P., 2004, “Tapping the full potential of product platforms: Best practices in planning, managing, and organizing for platform effectiveness”, Platform Management for Continued Growth, Atlanta, GA, IIR/PDMA.

Guo, F., Gershenson, J., 2004, “A comparison of modular product design methods on improvement and iteration”, ASME Design Engineering Technical Conferences, Salt Lake City, UT.

Hauser, J., Clausing, D., 1988, “The House of Quality”, Harvard Business Review, May – June 1988.

Igenbergs E., 2007, “Systems Engineering”, Systems Engineering lecture notes, Institute of Astronautics, Technische Universität München.

INCOSE, 2010, “Systems Engineering Handbook Guide for System Life Cycle Processes and Activities”, Version 3.2, International Council on Systems Engineering.

INCOSE (2002), “Systems Engineering Handbook Guide for System Life Cycle Processes and Activities”, Version 2, International Council on Systems Engineering.

INCOSE, 1998, “SE Terms Glossary”, Version 0, International Council on Systems Engineering.

iRobot, 2011, Companies Website: www.irobot.com

iRobot, 2010, “Roomba Vacuum Cleaning Robot, 500/600 Series Owner’s Manual”, Bedford, MA, iRobot Cooperation.

iRobot, 2006a, “Scooba Owner’s Manual”, Bedford, MA, iRobot Cooperation.

iRobot, 2006b, “Dirt Dog Owner’s Manual”, Bedford, MA, iRobot Cooperation.

ISO, 2009a, “ISO 31000:2009(E) – Risk management – Principles and guidelines”, Geneva, International Organization for Standardization.

ISO, 2009b, “ISO Guide 73:2009 – Risk management – Vocabulary”, Geneva, International Organization for Standardization.

ISO, 2009c, “ISO/IEC 31010: Risk management – Risk assessment techniques”, Geneva, International Organization for Standardization & International Electrotechnical Commission.

Jiao, J., Simpson, T., Siddique, Z., 2007, “Product family design and platform-based product development: a state-of-the-art review”, Journal of Intelligent Manufacturing, Vol. 18, No. 1.

Joksimovic, V., Houghton, W., Emon, D., 1977, “HTGR Risk Assessment Study”, in Jerry B. Fussell and G.R. Burdick (Eds.), Nuclear Systems Reliability Engineering and Risk Assessment, Society for Industrial and Applied Mathematics.

Jones, J., 2006, “Robots at the Tipping Point – The Road to the iRobot Roomba”, IEEE - Robotics & Automation Magazine, Vol.13, No.1.

Jones, J., Mack, N., Nugent, D., Sandin, P., 2009, “Autonomous floor cleaning robot”, US Patent No.: US 7,636,982 B2.

Jose, A., Tollenaere, M., 2005,“Modular and platform methods for product family design: literature analysis”, Journal Of Intelligent Manufacturing, Vol. 16, No. 3.

Kaplan, S., Garrick, J., 1981, “On the Quantitative Definition of Risk”, Risk Analysis, Vol.1, No.1.

Klir, G., Folger, T., 1988, “Fuzzy Sets, Uncertainty, and Information”, Englewood Cliffs, NJ, Prentice Hall.

Kota, S., Sethuraman, K., Miller, R., 2000, “A Metric for Evaluating Design Commonality in Product Families”, Journal of Mechanical Design, Vol. 122, No. 4.

Lehnerd, A. P., 1987, "Revitalizing the Manufacture and Design of Mature Global Products," Technology and Global Industry: Companies and Nations in the World Economy, Washington, D.C., National Academy Press.

Lévárdy, V., 2006, “Model-based Framework for the Adaptive Development of Engineering Systems”, Doctor of Science in Engineering Dissertation, Institute of Astronautics, Technische Universität München.

Martin, M., 1999, Design for Variety: A Methodology for Developing Product Platform Architectures”, Doctor of Philosophy Dissertation, Department of Mechanical Engineering, Stanford University.

Martin, M., Ishii, K., 2002, “Design for Variety: Developing standardized and modularized Product Platform Architectures”, Research in Engineering Design, Vol. 13, No. 4.

Martin, M., Ishii, K., 1997, “Design for Variety: Development of Complexity Indices and Design Charts”, Proceedings of the ASME Design Engineering Technical Conferences.

McManus, H., Richards, M.G., Ross, A.M., and Hastings, D.E., 2007, “A Framework for Incorporating “ilities” in Tradespace Studies,” AIAA Space 2007, Long Beach, CA.

McManus, H., Hastings, D., 2006, “A Framework for Understanding Uncertainty and its Mitigation and Exploitation in Complex Systems,” IEEE Engineering Management Review, Vol. 34, No. 3.

McGrath, M. E., 1995, “Product Strategy for High-Technology Companies”, New York, NY, Irwin Professional Publishing.

Meyer, M., 2007, “Revitalize your product lines through continuous platform renewal”, Research Technology Management, Vol. 40, No. 2.

Meyer, M., Lehnerd, A., 1997, “The power of product platforms: Building Value and Cost Leadership”, New York, NY, The Free Press.

Meyer, M., Utterback, J., 1993, “The product family and the dynamics of core capability”, Sloan Management Review, Vol. 34.

Meyer, M., Tertzakian, P., Utterback, J., 1997, “Metrics for managing research and development in the context of the product family”, Management Science, Vol. 43, No. 1.

Mikaelian, T., 2009, “An Integrated Real Options Framework for Model-based Identification and Valuation of Options under Uncertainty”, Doctor of Philosophy Dissertation, Aeronautics and Astronautics, MIT.

Mikaelian, T., Hastings, D.E., Rhodes, D.H., Nightingale, D.J., 2009, "Model-based Estimation of Flexibility and Optionability in an Integrated Real Options Framework", IEEE Systems Conference, Vancouver.

NASA, 2007, “NASA Systems Engineering Handbook”, National Aeronautics and Space Administration, Washington, Rev 1, 2007.

Otto, K., 2005, “Modular Product Platform Design”, Doctor of Science in Technology Dissertation, Department of Mechanical Engineering, Helsinki University of Technology.

Pine, B. J., 1993, “Mass customization: the new frontier in business competition”, Boston, MA, Harvard Business School Press.

Rajan. P., Van Wie, M., Cambell, M., Otto, K., Wood, K., 2003, “Design for flexibility – measures and guidelines”, International Conference on Engineering Design, Stockholm, Sweden.

Rasmussen, N., 1981, “Application of Probabilistic Risk Assessment Techniques”, Annual Review of Energy, No. 6.

Rhodes, D.H., Ross, A.M., 2010, “Methods and Metrics for Multi-Epoch Analysis – Epoch-Based Thinking: Anticipating System and Enterprise Strategies for Dynamic Futures, Short Program PI26s MIT Professional Education, MIT.

Roque, I., 1977, “Production-inventory systems economy using a component standardization factor”, Proceedings of the Midwest AIDS Meeting, Indianapolis, IN, American Institute for Decision Sciences.

Ross, A.M., Rhodes, D.H., 2008, “Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis”, INCOSE International Symposium 2008, Utrecht, the Netherlands.

Ross, A.M., Rhodes, D.H., Hastings, D.E., 2008, “Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining Lifecycle Value.” Systems Engineering, Vol. 11, No. 3.

Ross, A.M., 2006, "Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration", Doctor of Philosophy Dissertation, Engineering Systems Division, MIT.

Ross, A.M., Hastings, D.E., 2005, "The Tradespace Exploration Paradigm," INCOSE International Symposium 2005, Rochester, NY.

Sabbagh, K., 1996, "Twenty-First Century Jet: The Making and Marketing of the Boeing 777", New York, NY, Scribner.

Sanderson, S.W., Uzumeri, M., 1997, "Managing product families", Chicago, IL, Irwin Professional Pub.

Schellhammer, W., Karandikar, H., 2001, "Metrics for executing a product platform strategy", 13th International Conference Engineering Design, Glasgow, UK.

Schofield, D., 2010, "A Framework and Methodology for Enhancing Operational Requirements Development: United States Coast Guard Cutter Project Case Study", Masters of Science Thesis, MIT.

Simpson, T., D'Sousa, B., 2004, "Assessing Variable Levels of Platform Commonality Within a Product Family Using a Multiobjective Genetic Algorithm", Concurrent Engineering, Vol. 12, No 2.

Simpson, T., Siddique, Z., Jiao, J., 2006, "Product platform and product Family Design: Methods and Applications", New York, NY, Springer.

Simpson, T., Tucker, M., et al., 2006, "Platform-based design and development: current trends and needs in industry", ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Philadelphia, PA.

Suh, E. S., 2005, "Flexible Product Platforms", Doctor of Philosophy Dissertation, Engineering Systems Division, MIT.

Thunnissen, D., 2003, "Uncertainty Classification for the Design and Development of Complex Systems", 3rd Annual Predictive Methods Conference, Newport Beach, CA.

Tsubone, H., Matsuura, H., Satoh, S., 1994, "Component part commonality and process flexibility effects on manufacturing performance", International Journal of Production Research, Vol. 32.

Ulrich, K., Eppinger, S., 2008, "Product Design and Development", Fourth Edition, New York, NY, McGraw-Hill.

Ulrich, K., 1995, "The role of product architecture in the manufacturing firm", Research Policy, Vol. 24, No. 3.

Walter, U., 2009, "Systems Engineering", Course Reader Summer Term 2009, Technische Universität München.

Wang, T., 2005, "Real Options "in" Projects and Systems Design – Identification of Options and Solution for Path Dependency", Doctor of Philosophy Dissertation, Engineering Systems Division, MIT.

Wasson, C.S., 2006, “System Analysis, Design, and Development: concepts, principles, and practices”, Hoboken, NJ, John Wiley & Sons.

Wheelwright, S., Clark, K., 1992, “Creating project plans to focus product development”, Harvard Business Review, March – April 1992.

Wheelwright, S. C., W. E. Sasser, Jr., 1989, “The New Product Development Map”, Harvard Business Review, March – April 1989.

Whitman, R., 1984, “Evaluating Calculated Risk in Geotechnical Engineering”, Journal of Technical Engineering, Vol. 110, No. 2.

Wilhelm, B., 1997, “Platform and Modular Concepts at Volkswagen – Their Effects on the Assembly Process”, Transforming Automobile Assembly – Experience in Automation and Work Organization, New York, NY, Springer.

Yin, K., 2009, “Case Study Research – Design and Methods”, Fourth Edition, Thousand Oaks, CA, Sage Publications.

Zhang, Q., Doll, W., 2001, “The fuzzy front end and success of new product development: A causal model”, European Journal of Innovation Management, Vol. 4, No. 2.

Appendix A: Knowledge Gathering Instrument

This part of the appendix covers the knowledge gathering instrument, which was used for gathering the data analyzed and described in Chapter 3.

Identification of uncertainties in product platform lifecycle – knowledge gathering instrument

Background: This survey is in support of research identifying the uncertainties in lifecycle of product platforms. A further goal is to anticipate methods for a better treatment of uncertainties in platform development.

Your Rights: Your participation in this survey is voluntary; you may decline to answer any or all questions; you may decline further participation in this survey at any time without adverse consequences; and your confidentially and/or anonymity are assured.

Section 1: The following questions relate to the participant's expertise as well as their role in their company.

1. **Your name, company and current job title including the current functions/ responsibilities of your position and the products:**

2. **Could you please specify the platform project you are/were involved?**

Please describe the type of platform you use:	
Number of platforms and derivatives:	
Sales volume of the derivatives:	
Duration platform development:	
Duration platform lifecycle:	
Development team size:	

3. **What strategy do you use for developing a platform (e.g. <first platform →derivate> or <platform and first derivate together>)? How do you make the architecture decisions? Please describe your company concerning platform development experience and expertise.**

4. **Think about the platform project you did, what went wrong during the lifecycle of the platform because of failures made in the development process? Why wasn't it possible to build a platform, which is robust to mitigate these problems?**

Below are definitions of related terminologies. Please use this table as a reference for subsequent questions.

Terminology	Description[1]
Platform	Set of the architecture, common modules and interfaces from which a stream of derivative products can be efficiently developed and launched.
Architecture	The product architecture is the configuration of components within the product. It is the scheme where the physical components are associated to functional elements to form platform.
Module	It is a part or a (complex) group that allocates a function to the product and which could be changed and replaced in a loose way and be produced independently.
Interface	Connections between the modules and architecture of a platform, and between the platform and the customized parts of the product.
Uncertainty	Uncertainties are things that are not known, or known only imprecisely.
Risk	Risk is the effect of uncertainty on objectives.
Lack of Knowledge	Facts that are not known, or are known only imprecisely, that are needed to complete the system architecture in a rational way.
Lack of Definition	Things about the system in question that have not been decided or specified.
Statistically characterized variables	Things that cannot always be known precisely, but which can be statistically characterized, or at least bounded.
Known Unknowns	Things that it is known are not known. Future budgets, future adversaries, the performance of new technologies, and the like fall in this category.
Unknown Unknowns	By definition not known.
Margins	Designing platform to be more capable, to withstand worse environments, and to last longer than “necessary”.
Redundancy	Including multiple copies of modules or interfaces to assure that at least one works.
Design Choices	Choosing design strategies, technologies, and/or modules that are not vulnerable to a known risk.
Verification and Test	Testing after production to drive out known variation, bound known unknowns, and surface unknown unknowns.
Generality	Using Multiple-function modules and interfaces, rather than specialized ones.
Upgradeability	Modules and interfaces that can be modified to improve or change function.
Modularity	Functions grouped into modules and connected by standard interfaces in such a way that they can “plug and play”.
Real Options	Emerging technique originating in the financial world. Allows program strategy of carrying various design options forward and trimming options in a rational way as more information becomes available and/or market conditions change.

[1] Partly adopted from: McManus and Hastings (2005), Papalambros et al. (2002), Wilhelm (1997)

For answering question 5 please use following matrix as support for leveraging the impact of the sources of uncertainty:

Costs over budget in % (risk caused)	Time delay in % (risk caused): < 5 %	5 - 20 %	20 - 35 %	35 - 50 %	> 50 %
> 50 %	***9*** Major Update	***9*** Major Update	***9*** Major Update	***9*** Major Update	***9*** Major Update
35 - 50 %	***3*** Minor Update	***3*** Minor Update	***3*** Minor Update	***9*** Major Update	***9*** Major Update
20 - 35 %	***1*** Light Update	***1*** Light Update	***3*** Minor Update	***3*** Minor Update	***9*** Major Update
5 - 20 %	***1*** Light Update	***1*** Light Update	***1*** Light Update	***3*** Minor Update	***9*** Major Update
< 5 %	***0*** No consequences	***1*** Light Update	***1*** Light Update	***3*** Minor Update	***9*** Major Update

Section 2: The following questions are about what can go wrong in a platform lifecycle and because of which reasons.

5. **Different Risks Areas are listed below. Please rate the impact of these on your platform. (See matrix on page 3 for advice: 0 = no consequences, 1 = light update, 3 = minor update, 9 = major update, N = New, X = Project cancelation)**

Risk Areas	Part of the Platform	No consequ.	Light update	Minor update	Major update	New	Cancellation
Technology Capability/Changes (Uncertainty in capability and changes of technology to provide performance benefits (within cost and/or schedule expectations) and the consequences thereof)	Architecture	0	1	3	9	N	X
	Module	0	1	3	9	N	X
	Interface	0	1	3	9	N	X
Customer Needs (Uncertainty in change of customer needs (anticipated utility or value to the market of the chosen “design to” specifications) and in the ability of a design to meet desired quality criteria, in and the consequences thereof)	Architecture	0	1	3	9	N	X
	Module	0	1	3	9	N	X
	Interface	0	1	3	9	N	X
Market/Business shifts (Uncertainty in change of the market context, including competition, suppliers, economic situation and the consequences thereof)	Architecture	0	1	3	9	N	X
	Module	0	1	3	9	N	X
	Interface	0	1	3	9	N	X
Political and Cultural Context (Uncertainty in political, regulatory, labor, societal (e.g. fashion), or other factors in the political environment and the consequences thereof)	Architecture	0	1	3	9	N	X
	Module	0	1	3	9	N	X
	Interface	0	1	3	9	N	X
Organizational Changes (Uncertainty in the organization and structure of the company (including skills of participants and roles) and the consequences thereof)	Architecture	0	1	3	9	N	X
	Module	0	1	3	9	N	X
	Interface	0	1	3	9	N	X
Additional	Architecture	0	1	3	9	N	X
	Module	0	1	3	9	N	X
	Interface	0	1	3	9	N	X

6. **The knowledge about the uncertainties related to the risks enables the recommendation of methods for addressing them. What were the uncertainties that affect the different Risks listed in the matrix? Please mark an "X" in each box that you feel causes the different types of risk in the platform projects you worked on.**

		Uncertainties				
		Lack of Knowledge	Lack of Definition	Statistically characterized variables	Known Unknowns	Unknown Unknowns
Risk Areas	Technology Capability or Changes					
	Customer Needs					
	Market shifts					
	Political and Cultural context					
	Organizational Changes					
	Additional (Q5)					

7. **Imagine the platform project in your company, at what point in the lifecycle changes in the platform occurred because of the risks and what was the reason? Please provide reasons that could be written in the fields of the matrix.**

		Phases in Lifecycle						
		Product development				After SOP		
		Concept definition	Design Phase	Development and Fabrication	Integration & Verification	Production	Use	Retirement
Risk Areas	Technology Capability or Changes							
	Customer Needs							
	Market shifts							
	Political and Cultural Context							
	Organizational Changes							
	Additional (Q5)							

8. **A further part of this research is about methods, which are used or mitigate risk in platforms. What approaches/methods do you use to avoid the effects of uncertainties? Please rate your experience about the methods to avoid the risks by using following scale:**

0 = no use for this kind of risk	**1** = we used it, but it was not effective	**3** = we used it, the effectiveness was moderate	**9** = we used it, the effectiveness was superior

		Risks					
		Technology Capability or Changes	Customer Needs	Market Shifts	Political and Cultural Context	Organizational Changes	*Additional (Q5)*
Risk Mitigation	Margins						
	Redundancy						
	Design Choices						
	Verification and Test						
	Generality						
	Upgradeability						
	Modularity						
	Real Options						
	Additional:						
	Additional:						

9. **One last question is about the performance of the platform project in your company. Based on your experience or made assumptions, how would you rate your platform project?**

Time	How big was the reduction of time to market of new products because of the platform?	1	2	3	4	5	6	7
		1= >60% **2=**50% **3=**40% **4=**30% **5=**20% **6=**10% **7=** <5%						
	We could react faster after changes in customer needs because of the platform	1	2	3	4	5	6	7
		1=totally applicable; 7 = not at all applicable						
Quality	What is the re-use rate in your company?	1	2	3	4	5	6	7
		1= >60% **2=**50% **3=**40% **4=**30% **5=**20% **6=**10% **7=** <5%						
	Do you offer better quality because of the platform?	1	2	3	4	5	6	7
		1=totally applicable; 7 = not at all applicable						
	Do you have more different products in the market because of the platform?	1	2	3	4	5	6	7
		1=totally applicable; 7 = not at all applicable						
Costs	How big was the reduction of the development costs of the related products because of the platform?	1	2	3	4	5	6	7
		1= >60% **2=**50% **3=**40% **4=**30% **5=**20% **6=**10% **7=** <5%						
	How big was the reduction of the production costs of the related products because of the platform?	1	2	3	4	5	6	7
		1= >60% **2=**50% **3=**40% **4=**30% **5=**20% **6=**10% **7=** <5%						
	How big was the reduction of the maintenance costs of the related products because of the platform?	1	2	3	4	5	6	7
		1= >60% **2=**50% **3=**40% **4=**30% **5=**20% **6=**10% **7=** <5%						
	How much cost could you save with platform projects in the future? (in average)	1	2	3	4	5	6	7
		1= >60% **2=**50% **3=**40% **4=**30% **5=**20% **6=**10% **7=** <5%						

End of Survey

Thank you for your time and participation.

Appendix B: Framework Flowcharts

This part of the appendix covers the flowchart of the framework described in Chapter 4. Steps 1 to 6 are in a sequence with a possible loop in between; Step 7 is parallel to Steps 1-6.

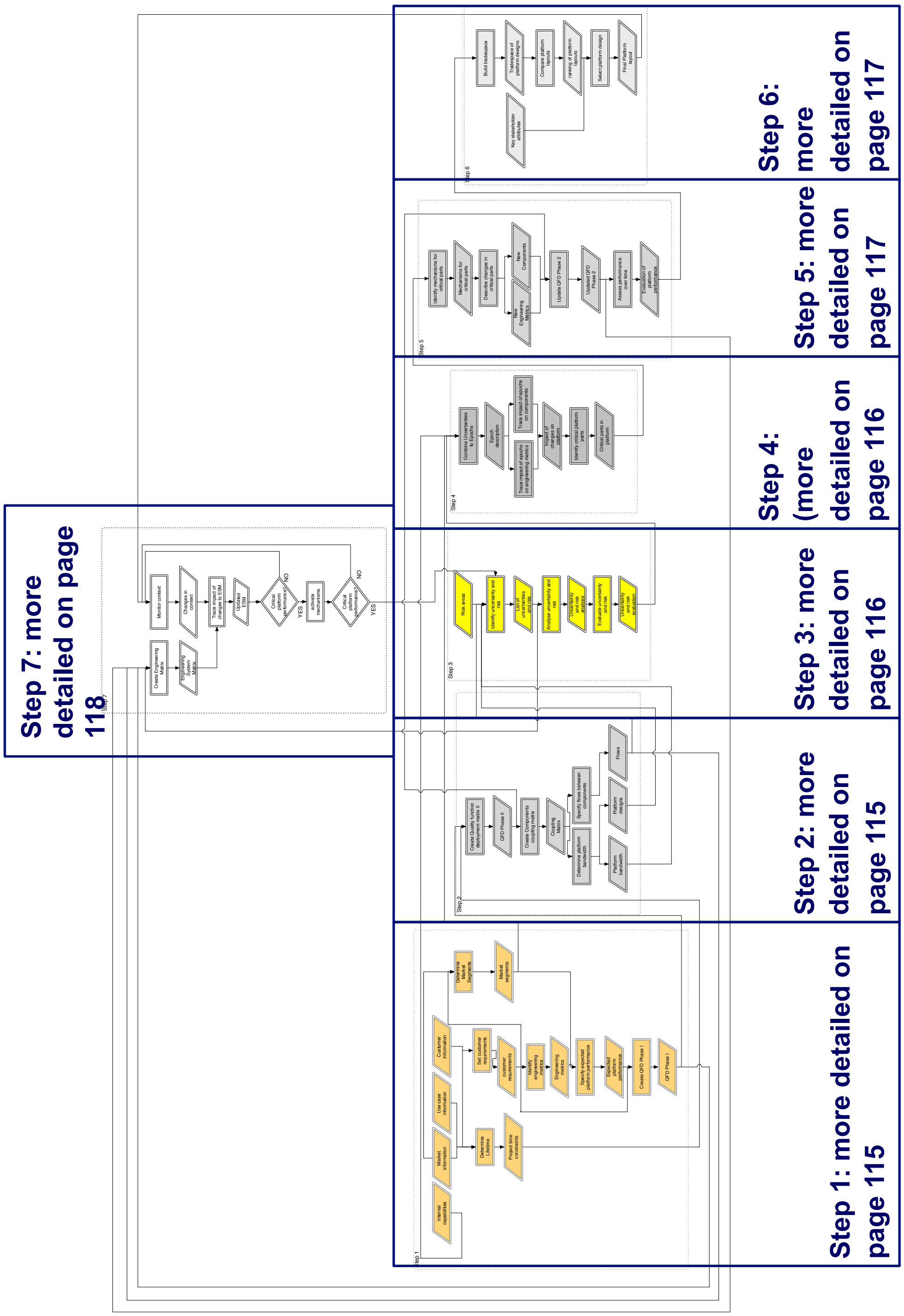


Figure B-1: Framework overview (steps more detaield on next pages)

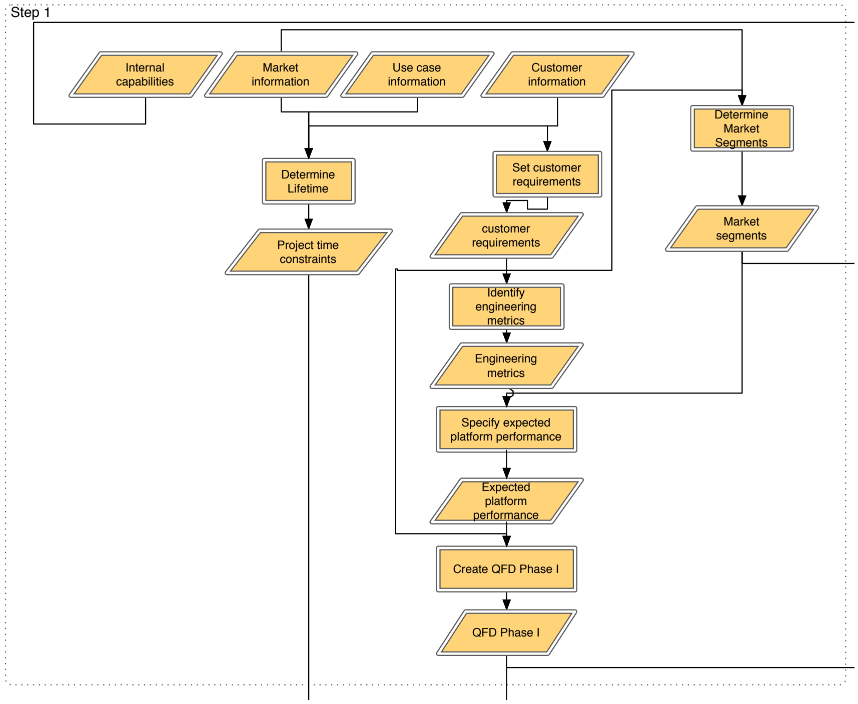


Figure B-2: Flowchart of Step 1 (Identifying Customer Needs and Requirements)

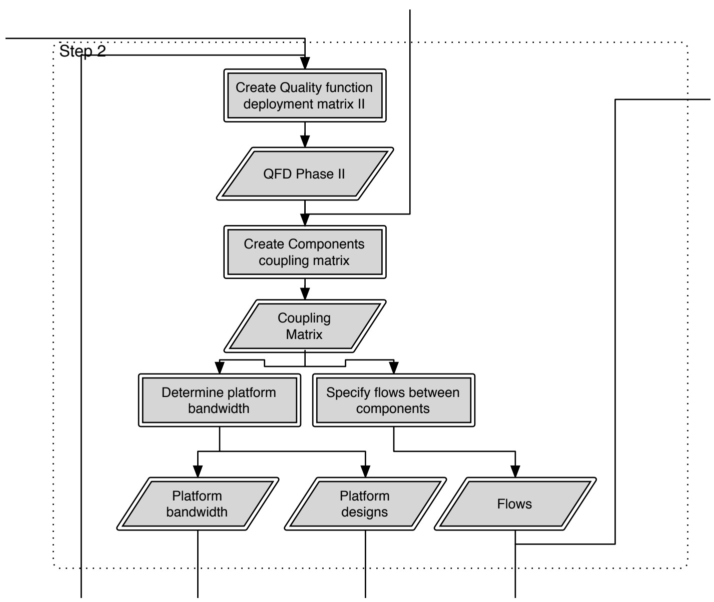


Figure B-3: : Flowchart of Step 2 (Investigating Product Platform Designs and Bandwidth)

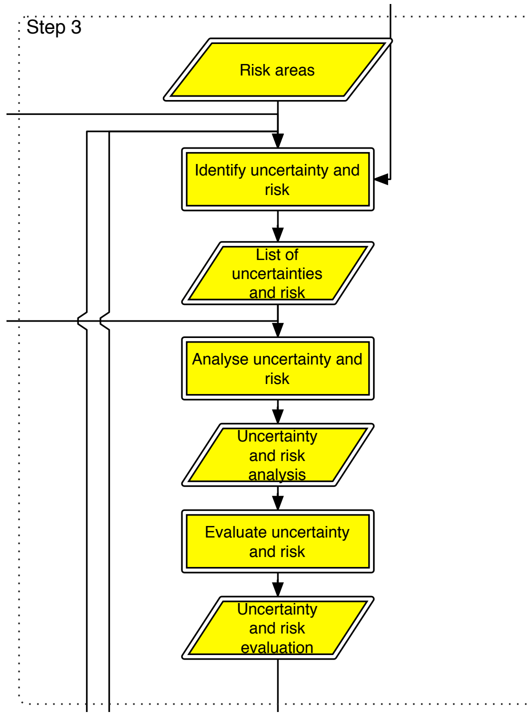


Figure B-4: Flowchart of Step 3 (Uncertainty and Risk Characterization)

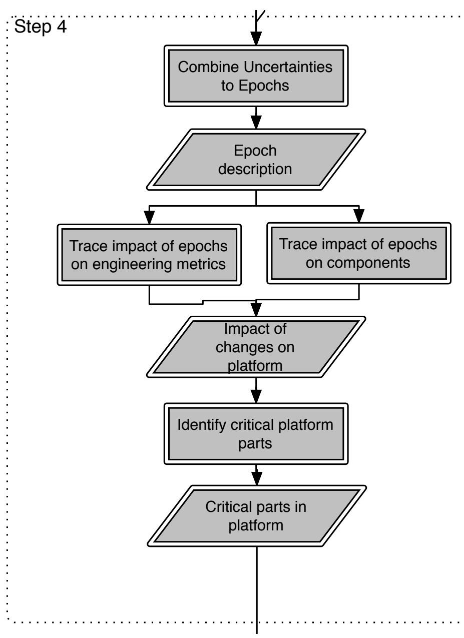


Figure B-5: Flowchart of Step 4 (Epoch Description and Analysis)

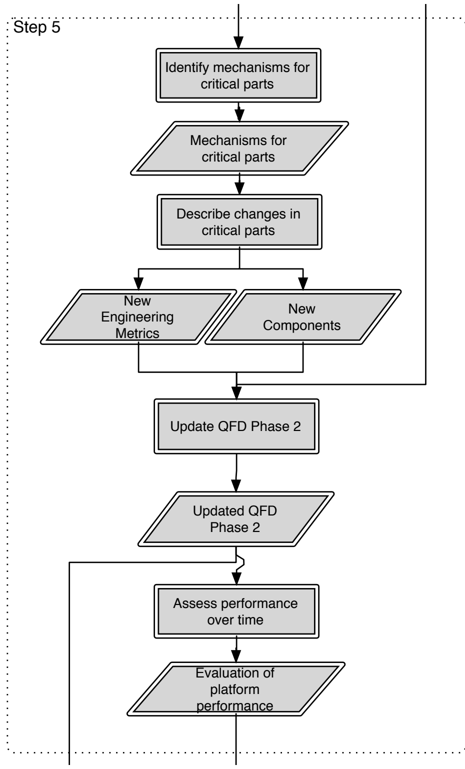


Figure B-6: Flowchart of Step 5 (Assess Performance of Platform Designs)

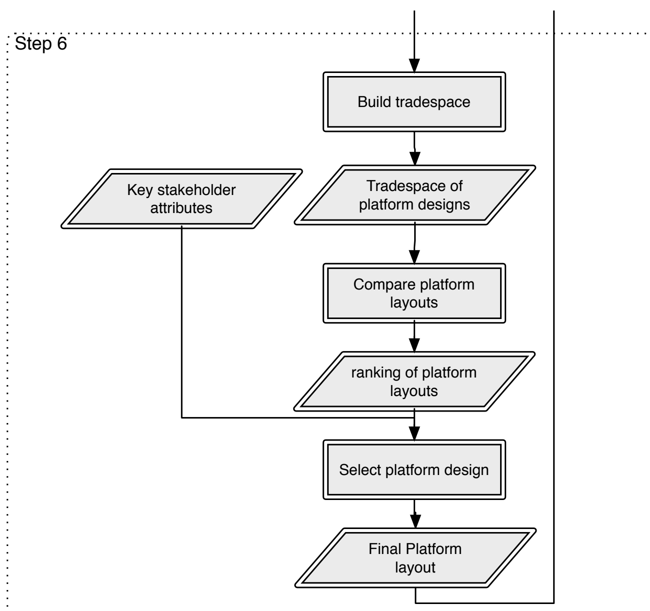


Figure B-7: Flowchart of Step 6 (Compare Product Platform Designs and Selection)

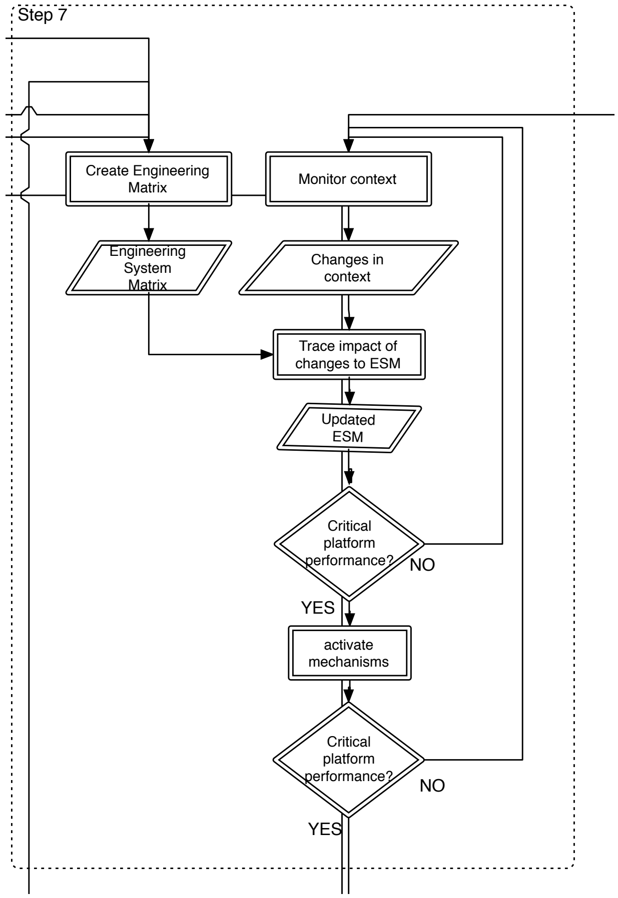


Figure B-8: Flowchart of Step 7 (Review Selected Design, Monitor Context and Identify Implications to Other Areas)

Appendix C: Cleaning and Cutting Robot SysML Diagrams

This appendix documents all diagrams developed in the SysML modeling for the illustrative case in Chapter 5.

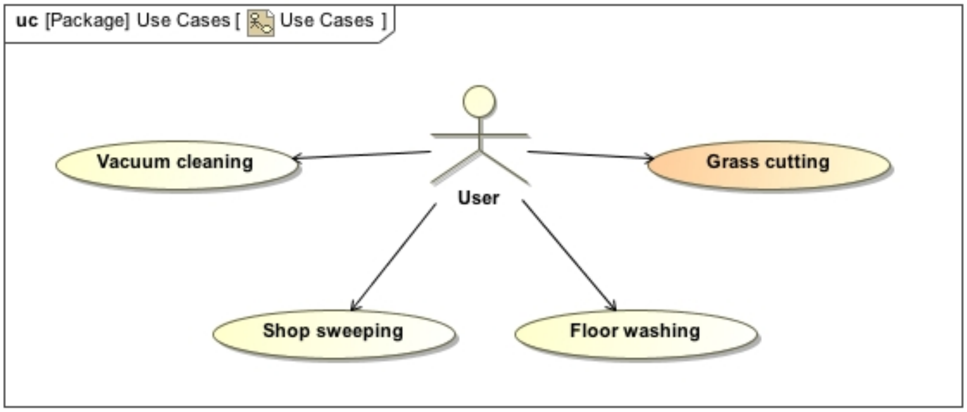


Figure C-1: Use cases for cleaning and cutting robot

#	ID	Name
1	3	Short cleaning time
2	7	No damage
3	11	High vacuum power
4	13	Dries floor effectively
5	14	Load of much debris
6	16	Cut high grass
7	8	Quiet during movement
8	15	Collect cut grass
9	2	No hang ups
10	10	Autonomous movement
11	6	Inexpensive to operate
12	12	Excellent filter capacity
13	4	Long run time
14	5	Easy to maintain
15	9	Movement on different floor materials
16	1.3	Deep cleaning ability
17	1.2	Covers entire space
18	1.1	Cleans under projects
19	1	Cleans Effectively

Figure C-2: List of functional requirements

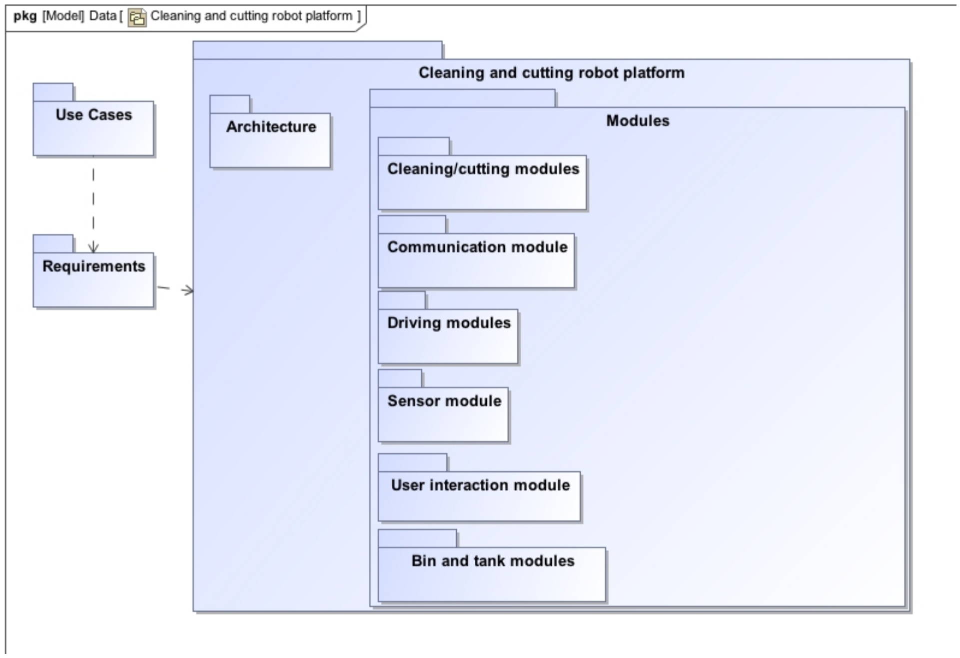


Figure C-3: Package diagramm cleaning and cutting robot

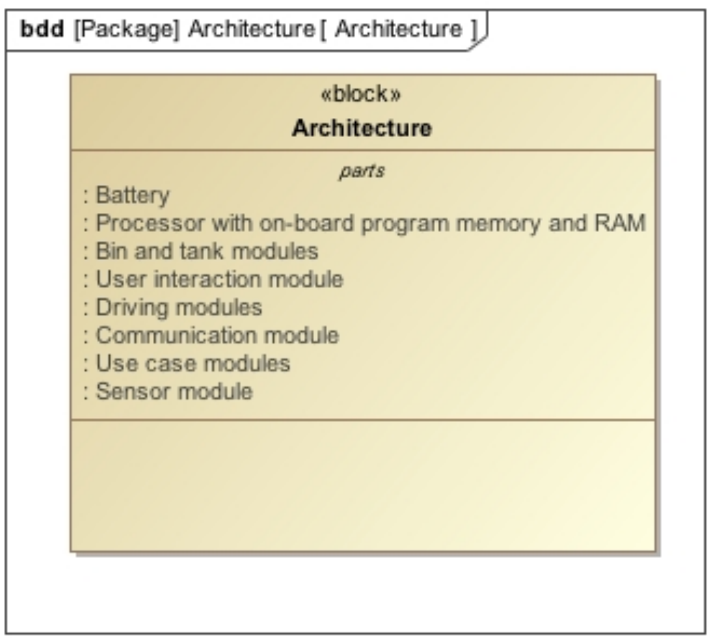


Figure C-4: Block diagramm of the cleaning and cutting robot architecture

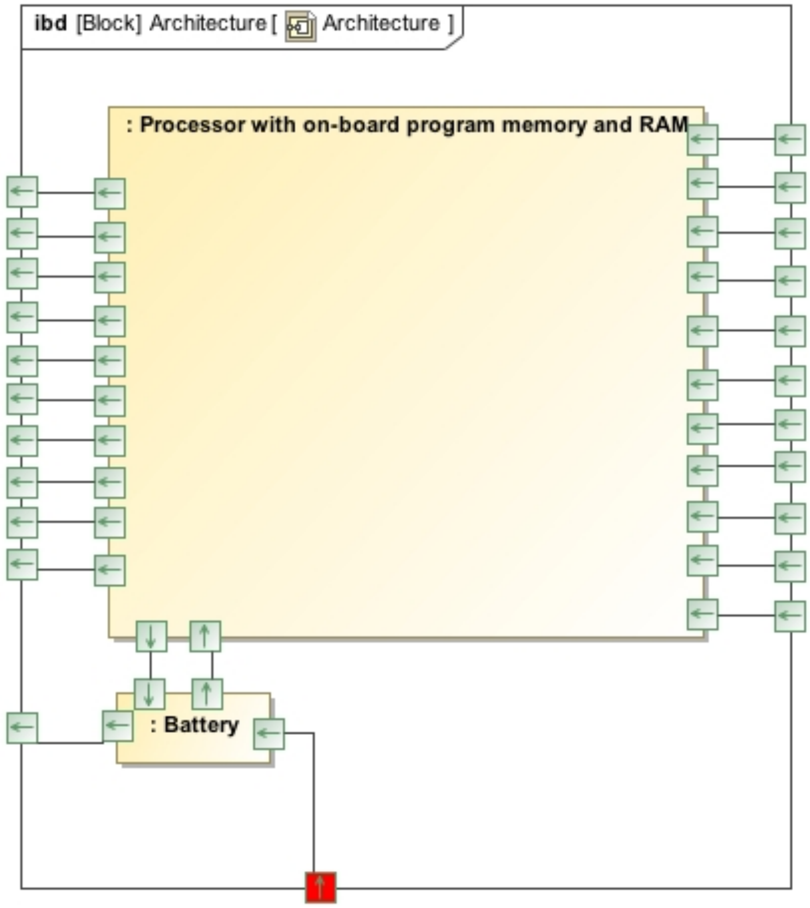


Figure C-5: Internal block diagramm of the cleaning and cutting robot architecture

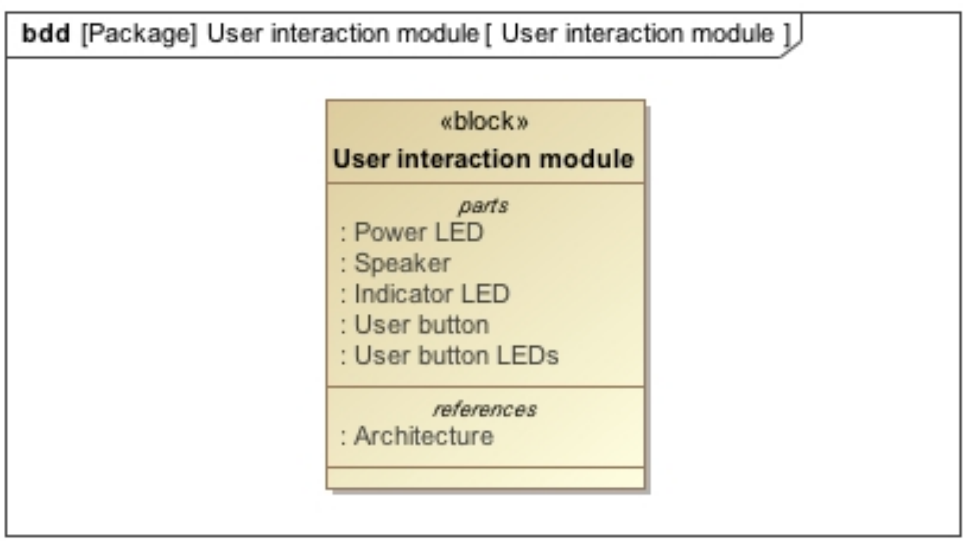


Figure C-6: Block diagramm of the cleaning and cutting robot user interaction module

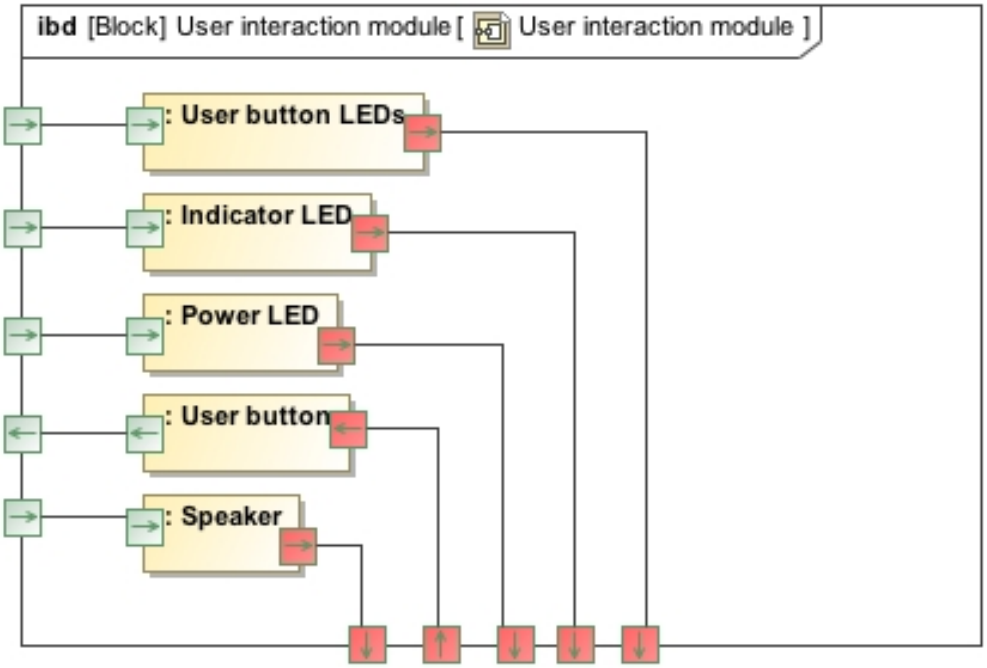


Figure C-7: Internal block diagramm of the cleaning and cutting robot user interaction module

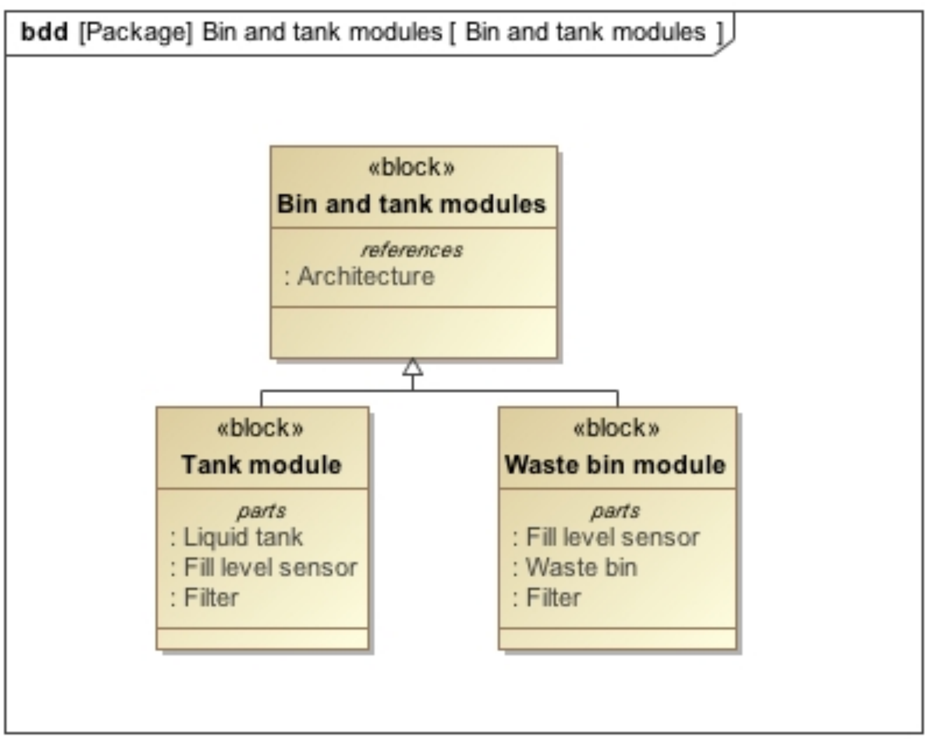


Figure C-8: Block diagramm of the cleaning and cutting robot bin and tank modules

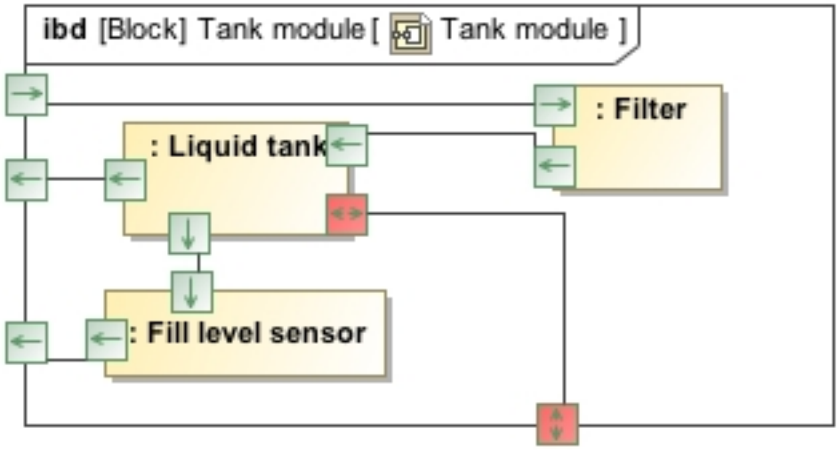


Figure C-9: Internal block diagramm of the cleaning and cutting robot tank module

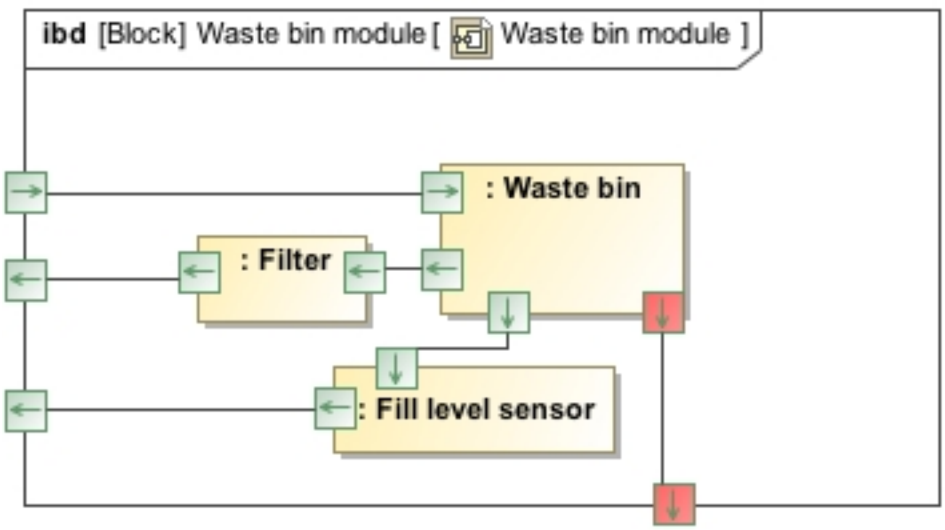


Figure C-10: Internal block diagramm of the cleaning and cutting robot waste bin module

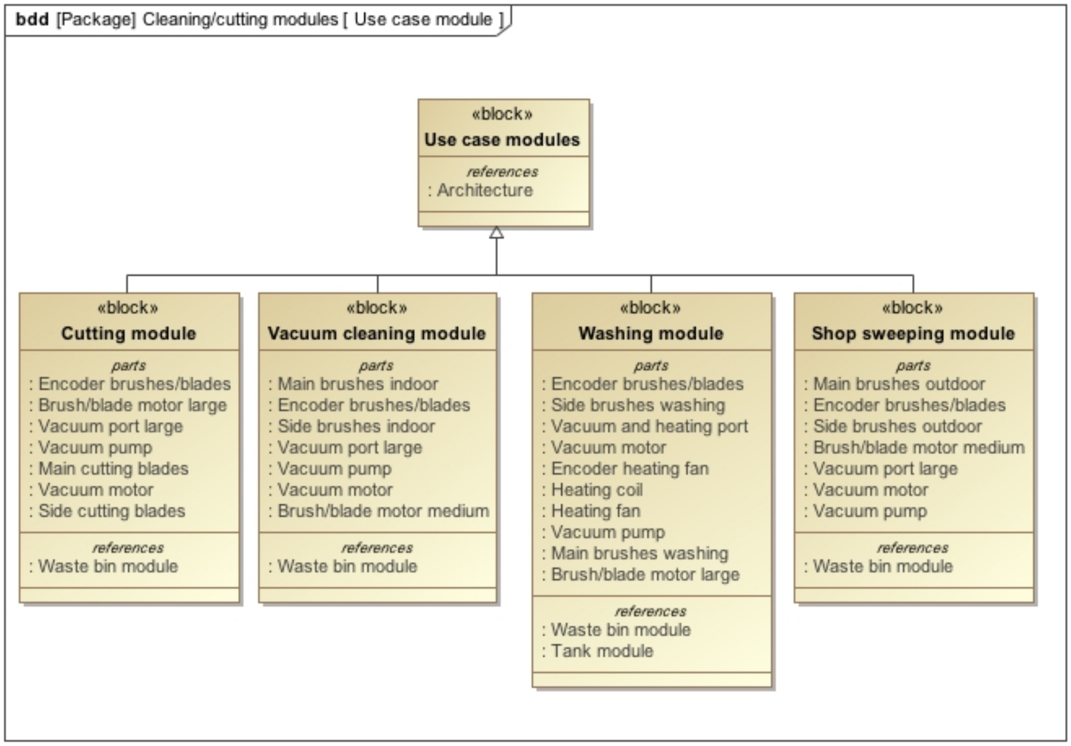


Figure C-11: Block diagramm of the cleaning and cutting robot use case modules

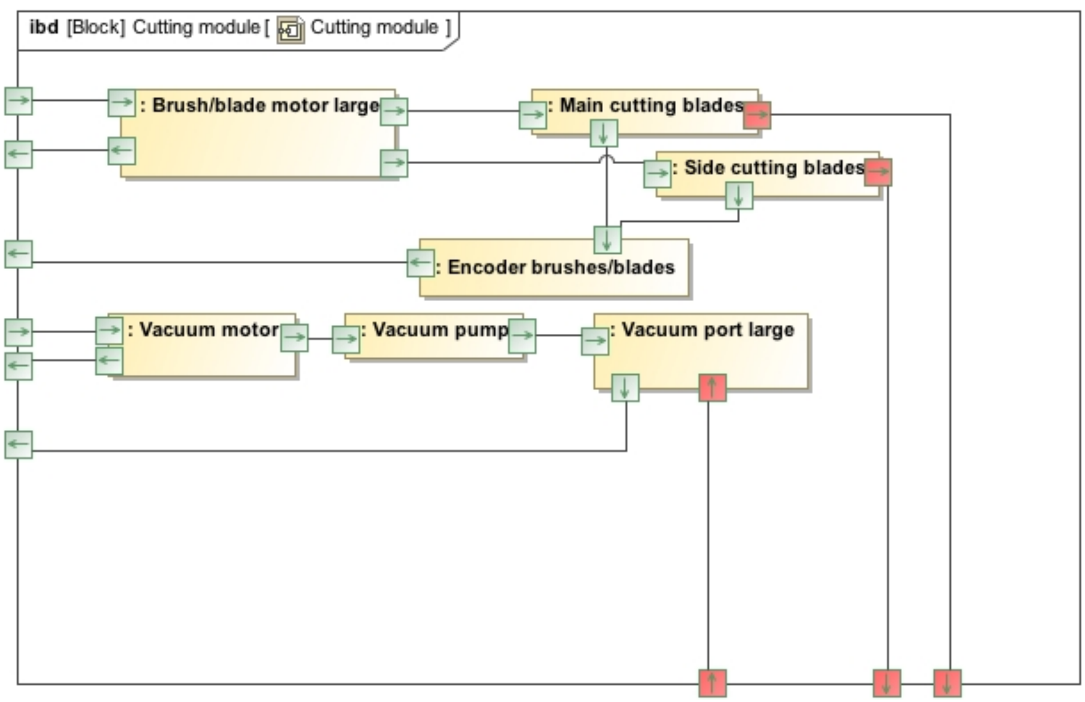


Figure C-12: Internal block diagramm of the cleaning and cutting robot cutting module

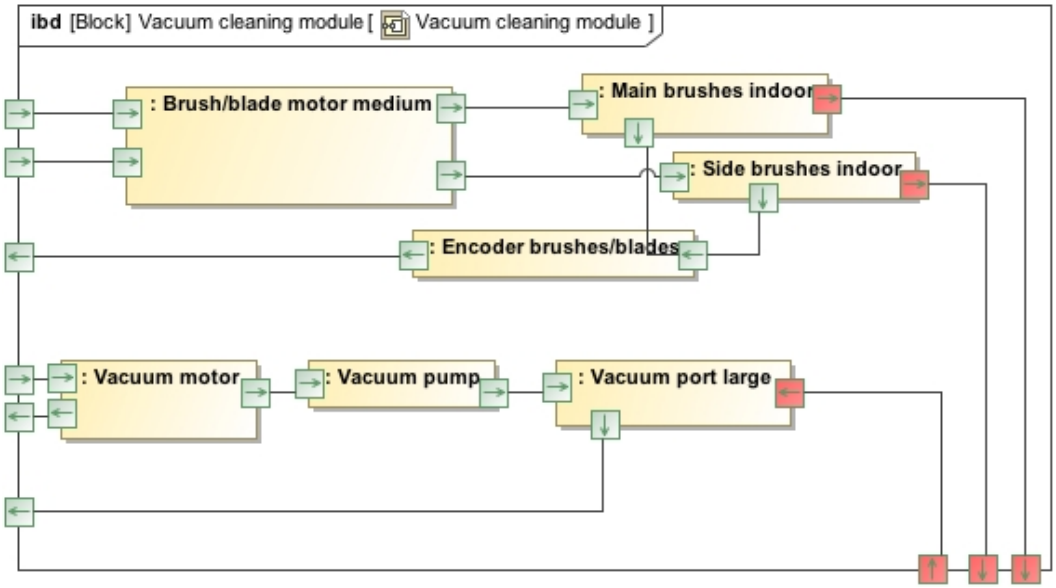


Figure C-13: Internal block diagramm of the cleaning and cutting robot vacuum cleaning module

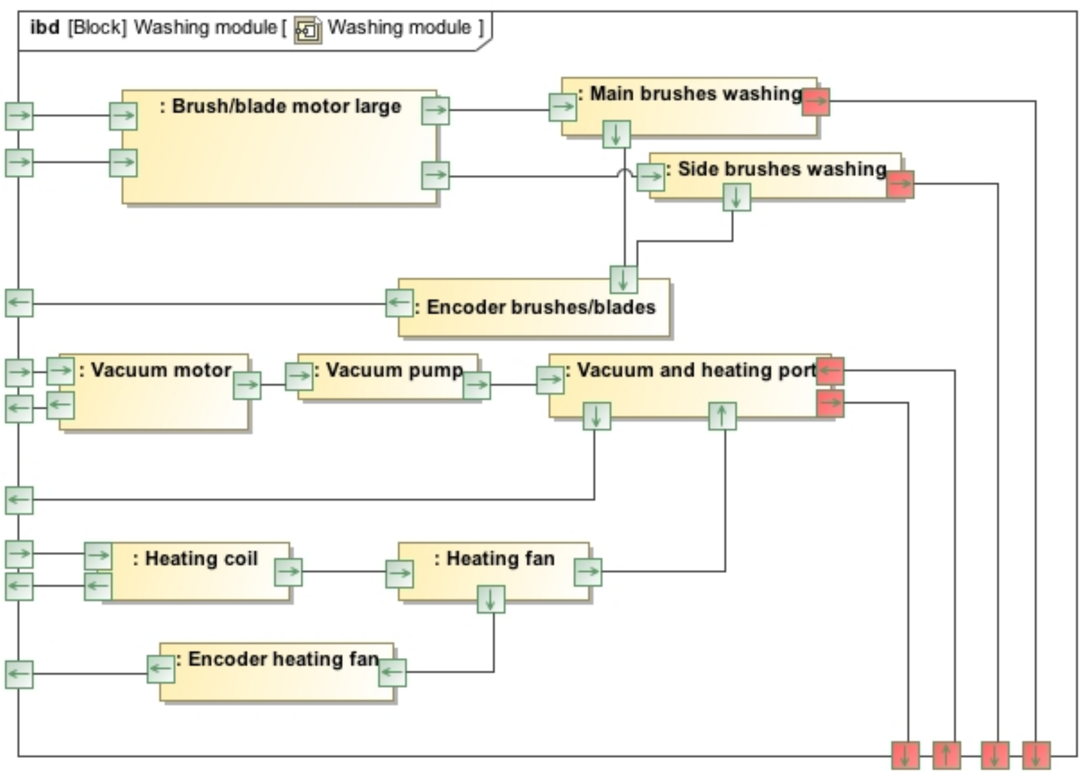


Figure C-14: Internal block diagramm of the cleaning and cutting robot washing module

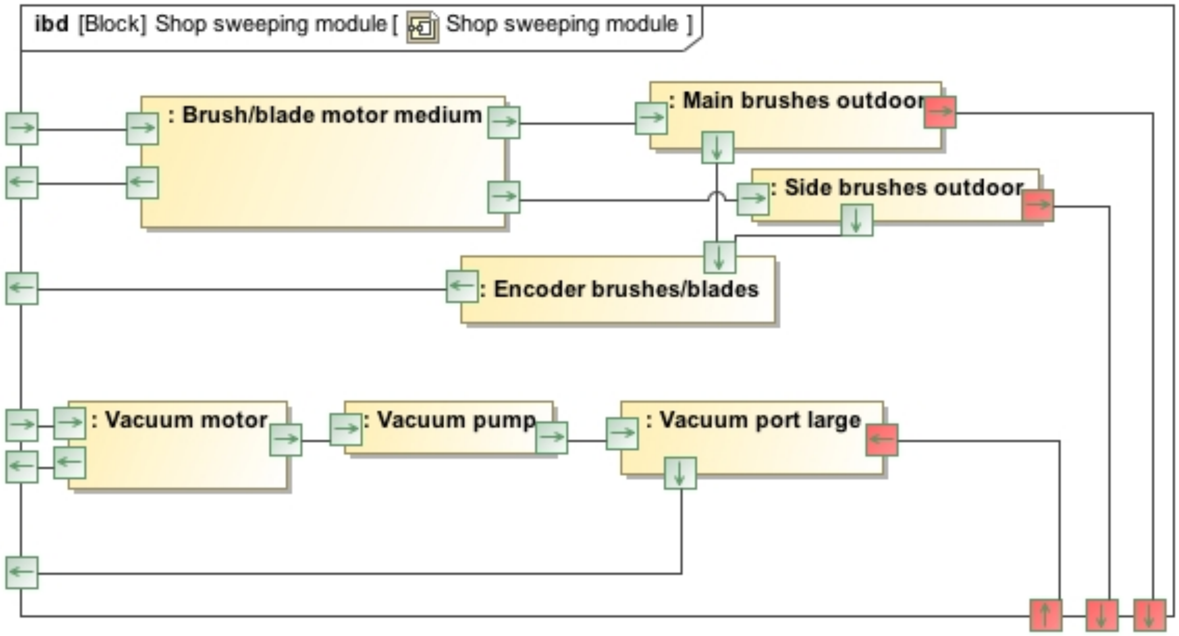


Figure C-15: Internal block diagramm of the cleaning and cutting robot shop sweeping module

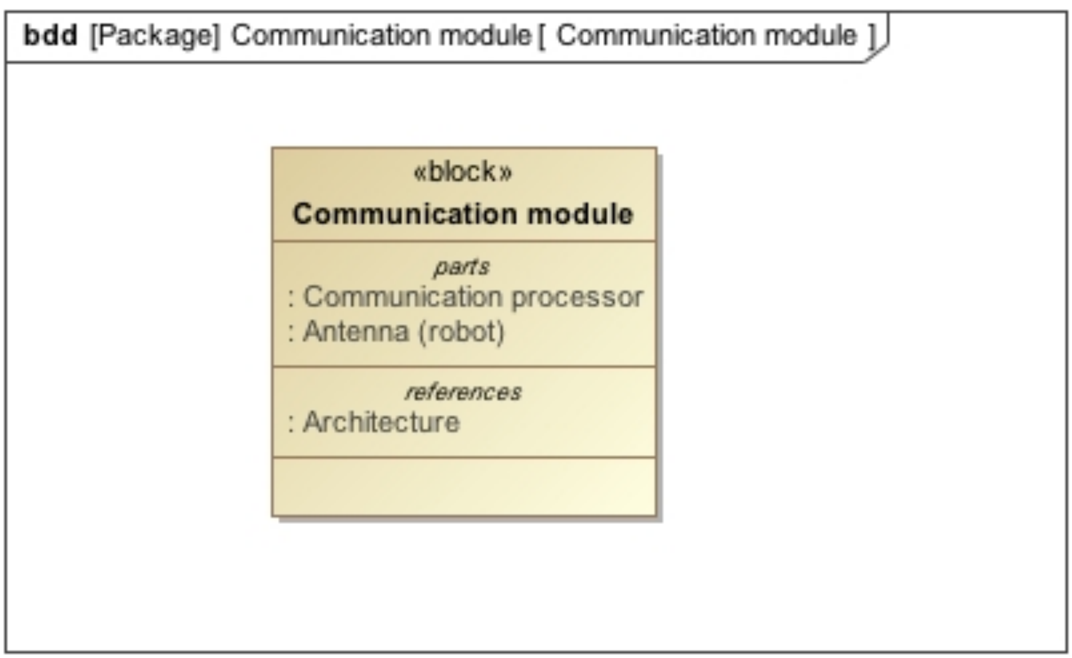


Figure C-16: : Block diagramm of the cleaning and cutting robot communiaction module

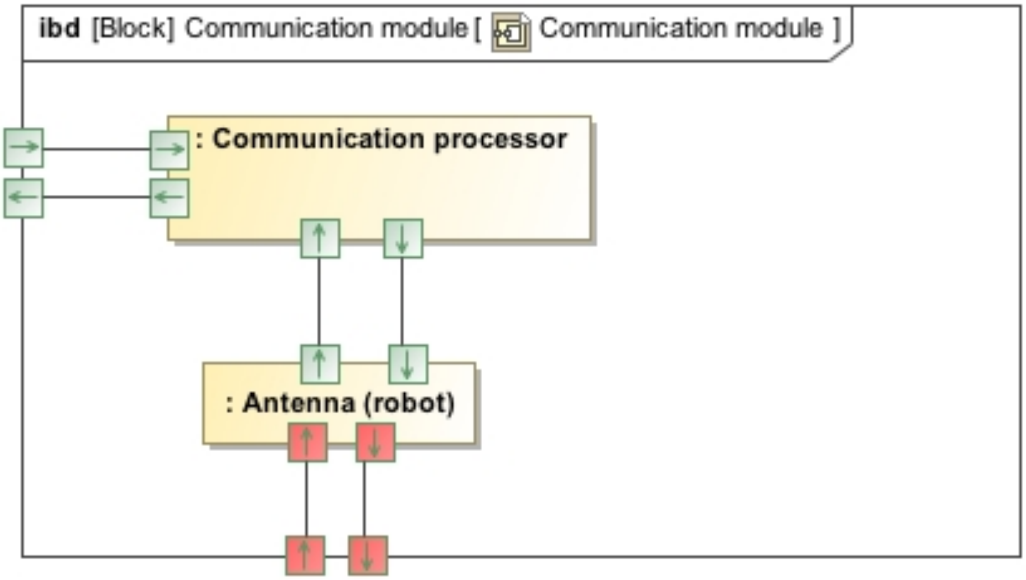


Figure C-17: Internal block diagramm of the cleaning and cutting robot communication module

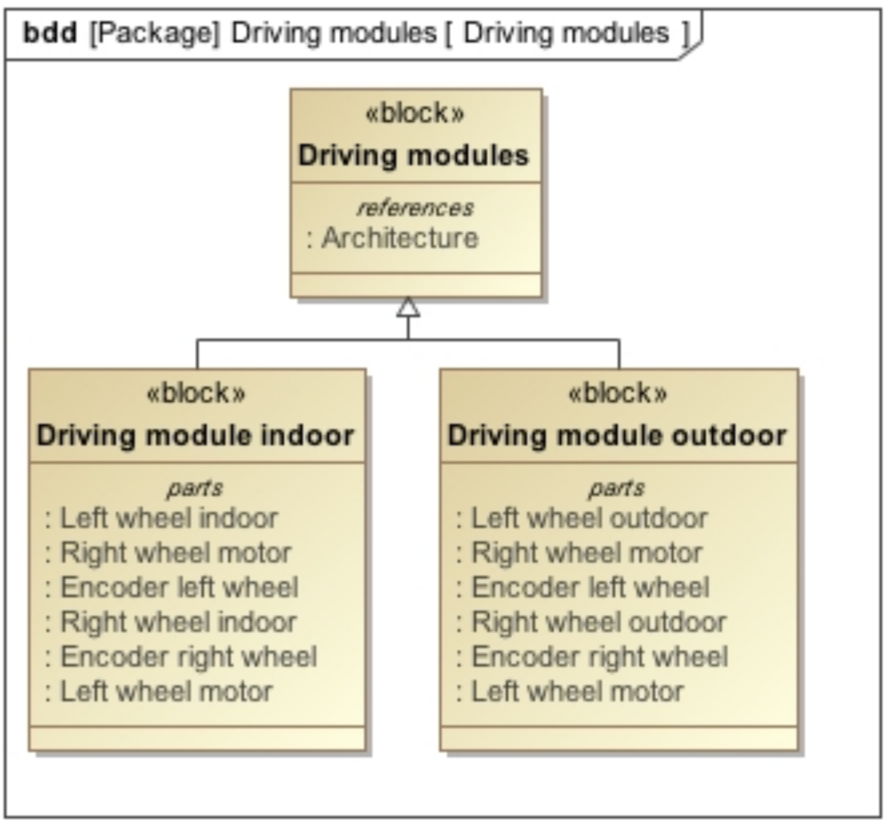


Figure C-18: Block diagramm of the cleaning and cutting robot driving modules

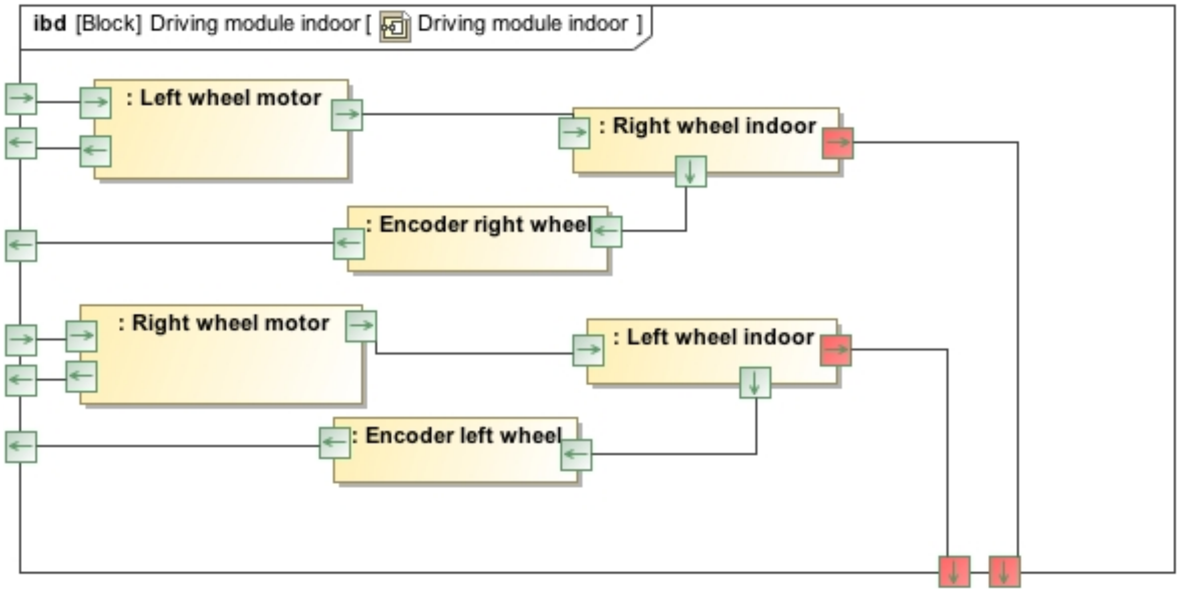


Figure C-19: Internal block diagramm of the cleaning and cutting robot driving module indoor

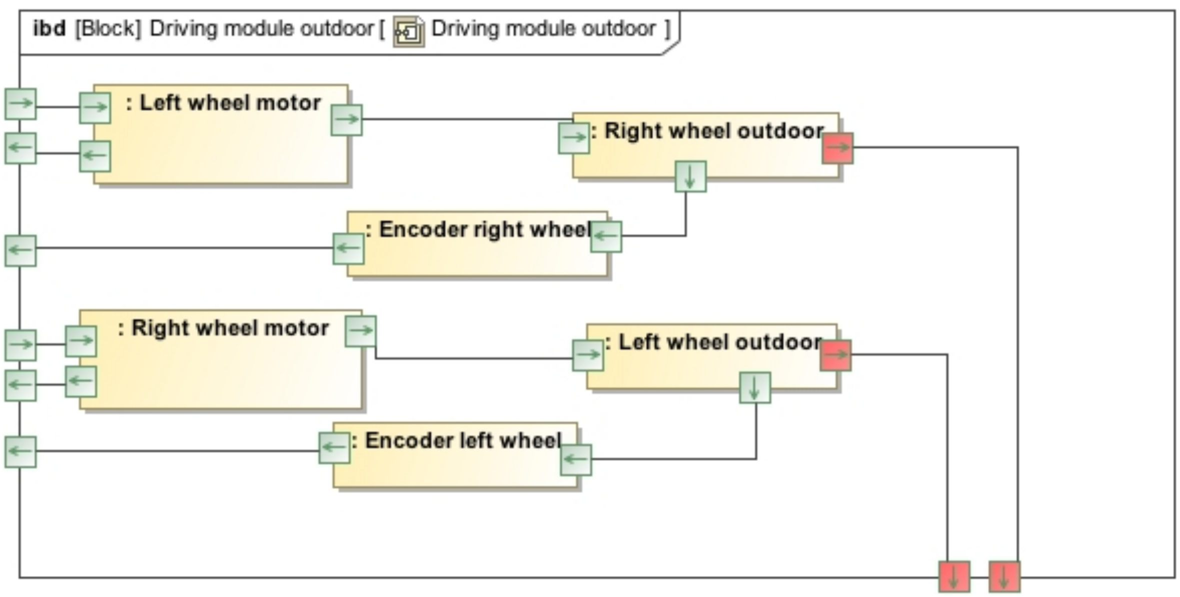


Figure C-20: Internal block diagramm of the cleaning and cutting robot driving module outdoor

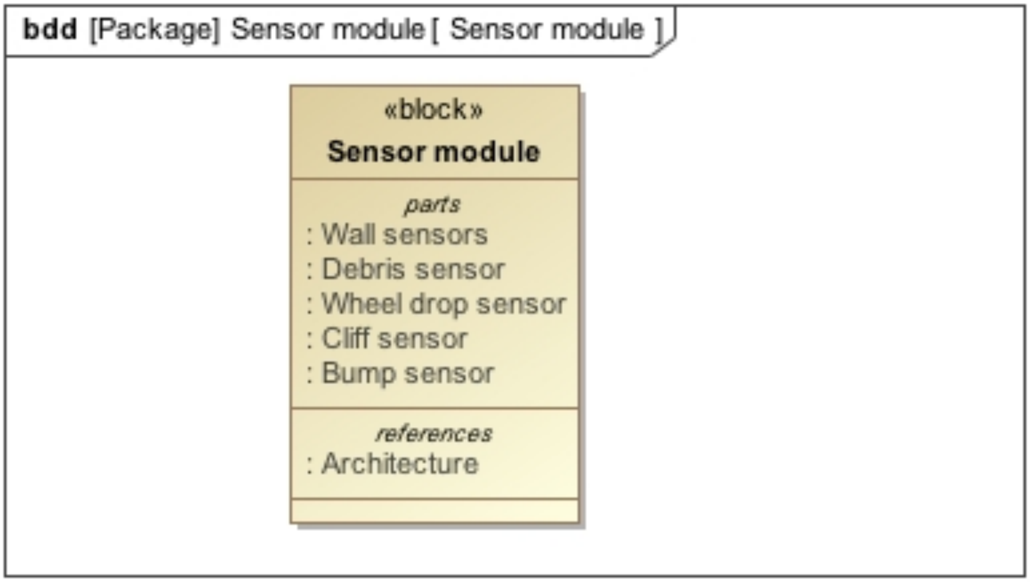


Figure C-21: Block diagramm of the cleaning and cutting robot sensor module

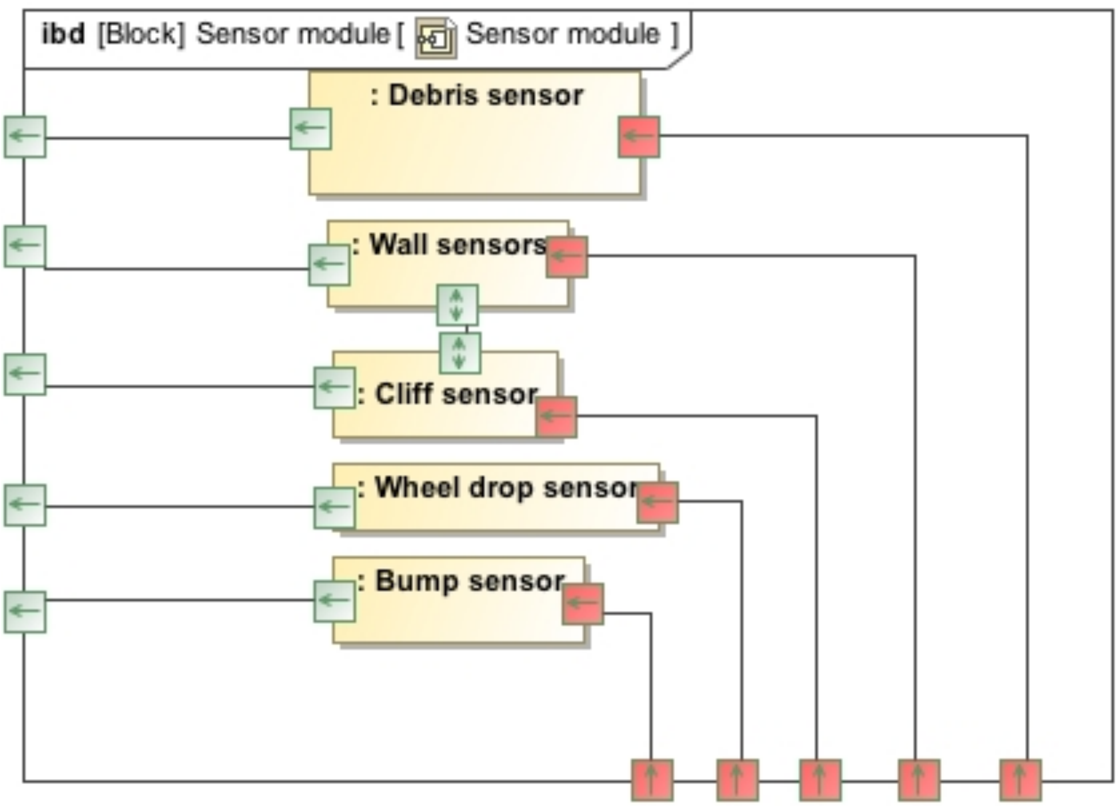


Figure C-22: Internal block diagramm of the cleaning and cutting sensor module

Affirmation

I hereby assure that the single-handed composition of this diploma thesis is only supported by declared resources and that this thesis has not been presented nor published before in the same or a similar form.

Cambridge/Munich, January 31st, 2011

Augustin Friedel