Valentin Hollweger, BSc

Design and Implementation of a Production Data Acquisition System in Discrete Manufacturing

Masterthesis

receiving the academic degree

Dipl.-Ing.

Master's Degree Programme Mechanical Engineering and Business Economics

Graz University of Technology

Adviser: Univ.-Ass. Dipl.-Ing. Dietmar Neubacher
Univ.-Ass. Dipl.-Ing. Clemens Gutschi
Auditor: Univ.-Prof. Dipl.-Ing. Dr. techn. Siegfried Vössner

Institute of Engineering and Business Informatics

Graz, 2018

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Acknowledgement

I would like to take this opportunity to express my gratitude to all my companions during my studies.
Further I would like to say a big thanks to my co-workers at the Industrial Engineering department at WP Performance Systems, who always had an open ear and supported me in the best possible way in the realization of the project. Special thanks also go to my manager, who made this project possible and who always motivated me.
I would also like to thank my two supervisors Dietmar Neubacher and Clemens Gutschi at the Institute of Engineering and Business Informatics, who have always supported me with good advice. I greatly appreciated their valuable feedback.
For their great support, not only in financial terms, I would like to thank my parents Franz and Susanne.
Finally I would like to thank my partner Kerstin for her almost unlimited patience and her constant support, that she showed especially at the end of my studies.

Kurzfassung

Die gesamte Industrie ist dabei, sich zu verändern, denn jeder spricht nur noch von Industrie 4.0 und welche Möglichkeiten sich dadurch ergeben könnten. Ein Wegbereiter für Industrie 4.0 Anwendungen ist klarerweise das Potential aller möglicher Daten die während eines Produktionsprozesses anfallen. Um diese Potentiale ausschöpfen zu können, sollte man sich mit der systematischen Erfassung dieser Daten beschäftigen.

Diesen Zweck erfüllen so genannte Manufacturing Execution Systems. Um aus der Vielzahl an potentiellen Software Anbietern eine passende Lösung auszuwählen soll zum einen die Methodik des Requirements Engineerings und zum anderen das Konzept des Systems Engineerings zur Hilfe genommen werden.

Die Erkenntnisse der Literaturrecherche konnten in einer Fallstudie in einem Industrieunternehmen praktisch angewandt werden. Dabei wurden zunächst grob die Projektziele und Erwartungen der Stakeholder abgesteckt. Anschließend konnten in einer Ist-Analyse die derzeitigen Probleme mit und Bedürfnisse für IT-unterstützte Systeme in der Produktion erhoben werden. Aus den gewonnenen Erkenntnissen konnten gemeinsam mit den Stakeholdern Use-Cases identifiziert und daraus Anforderungen an ein Betriebsdatenerfassungssystem abgeleitet werden. Anschließend wurden aus der Vielzahl am Markt angebotenen Lösungen fünf potentielle Systeme ausgewählt und anschließend mit Hilfe einer Nutzwertanalyse verglichen. Für die Software mit dem höchsten Nutzwert wurde eine Systemarchitektur und ein Konzept zur Betriebsdatenerfassung entworfen. Dieses Konzept wurde anschließend in einem Proof-of-Concept an drei unterschiedlichen Produktionsanlagen auf seine Tauglichkeit geprüft.

Aufgrund der Ergebnisse des Proof-of-Concepts konnte gezeigt werden, dass sich das gewählte System für das Unternehmen zur Betriebsdatenerfassung eignet.

Abstract

The whole industry is about to change, since everyone is talking about Industry 4.0 and what possibilities this might create. An enabler for Industry 4.0 applications is clearly the potential of any data generated during the production process. In order to be able to exploit these potentials, it is important to focus on the systematic collection of this data.

So-called Manufacturing Execution Systems fulfill this purpose. In order to select a suitable solution from the multitude of potential software vendors, the methodology of Requirements Engineering on the one hand and the concept of Systems Engineering on the other hand should be considered.

The findings of the literature research could be applied in a case study in an industry company. At first, the project goals and expectations of the stakeholders were roughly defined. Subsequently, the current problems with and needs for IT-supported systems in production could be determined in an as-is analysis. From the findings gained, use cases could be identified together with the stakeholders and requirements for an production data acquisition system could be derived. Afterwards, five potential systems were selected from the multitude of solutions offered on the market and subsequently compared with the help of a value-benefit analysis. For the software with the highest value benefit, a system architecture and a concept for production data acquisition were designed. This concept was then tested for suitability in a proof-of-concept at three different machines in production.

Based on the results of the proof-of-concept it could be shown that the selected system is suitable for the company for production data acquisition.

Contents

Figures

Tables

Abbreviations

CIM	Computer Integrated Manufacturing
CIP	Continous Improvement Process
CPS	Cyber Physical Systems
FR	Functional Requirements
HMI	Human Machine Interface
IDE	Integrated Development Environment
MES	Manufacturing Execution System
NFR	Non-Functional Requirements
OEE	Overall Equipment Effectiveness
OPC UA	Open Platform Communications Unified Architecture
PDA	Production Data Acquisition
PLC	Programmable Logic Controller
RE	Requirements Engineering / Requirements Engineer
SCADA	Supervisory Control and Data Acquisition
SE	Systems Engineering
SysML	Systems Modelling Language
TCO	Total Cost of Ownership
TR	Technical Requirements

UML Unified Modeling Language

UCTMF Use Case Technology Mapping Framework

VBA Value Benefit Analysis

1. Introduction

The whole industry is about to change. Everybody is talking about the hype topic *Industry 4.0* and this buzzword is no longer just a term for insiders from the manufacturing industry, but the future hope of an entire nation (Kollmann and Schmidt, 2016). With *Industry 4.0* the general hope is to secure the production sites in Europe in the long term (Müller et al., 2016). Politicians and companies have recognized that digitization and intercommunication are creating great growth opportunities (Bauer et al., 2015). However, this inevitably raises the question of what the term *Industry 4.0* actually hides, or what can be expected from this philosophy. But there is a lack of specific knowledge about adopting *Industry 4.0* technologies in companies (Armin Roth, 2016). This raises the question of which basic technologies are needed to create the basis for *Industry 4.0* in the company. An enabler for *Industry 4.0* applications are certainly all possible data that occur during a production process (Wee et al., 2015). In order to exploit the potential of this data, it is advisable to consider systematic collection of this data with the assistance of a Production Data Acquisition System.

1.1. Motivation

Most companies do not start their *Industry 4.0* path with new greenfield factories, but face the problem of having to adapt existing brownfield factories (Huber, 2016). In these brownfield factories, the IT systems, plants and processes have grown historically and there is also a strong heterogeneity in the origin and technology of the machines (Huber, 2016). In order to master the new challenges in this environment well, it is even more important to plan the introduction of new systems sophisticatedly. Such a system can be, for example, a *Manufacturing Execution System* for collecting operational data. However, the market offers a large number of systems in various forms and with varying degrees of quality. Finding a suitable system for production data acquisition is therefore a great challenge. On the one hand, the preconditions in the company must be taken into account and on the other hand, future developments and requirements must be considered as well in the best possible way. Furthermore, the consideration of all requirements of all stakeholders for such a system represents a great challenge. For this reason it is important to support the selection, design and implementation of such a system with the use of a good methodological process.

1.2. Scope

The aim of this thesis was derived from the initial problem description. Therefore, the goal was to find a methodology for selecting a Production Data Acquisition System. The first step was to elaborate the theoretical principles of *Manufacturing Execution Systems*. Furthermore, the fundamentals of *Requirements Engineering* and *Systems Engineering* were discussed. The results of the literature work were then applied in a case study. Therefore, the requirements of WP for such a system were established. Furthermore, an analysis and evaluation of several software systems for production data acquisition were done. To verify the evaluation result, a Proof-of-Concept were implemented.

1.3. Methodological Approach

The thesis is divided into two parts: On the one hand the theoretical background of Manufacturing Execution Systems, Requirements Engineering and Systems Engineering and on the other hand the application of the findings in a Case Study at a partner company. In the theoretical part of this study the background of Manufacturing Execution Systems, but also the theory of Systems Engineering and Requirements Engineering will be presented. The Case Study is based on the findings from the literature study in sections 2, 3 and 4. The applied methodology can be divided in five main phases (see also figure 1):

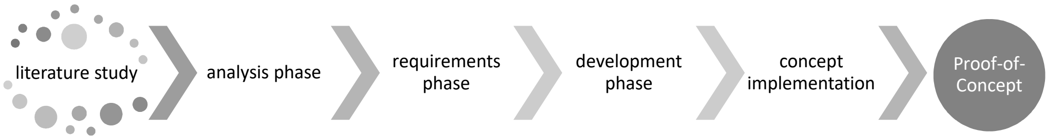

Figure 1.: Project phases of the thesis

Analysis phase

In the analysis phase, an evaluation of current production data acquisition within the company was performed. Furthermore the production systems at the company were analysed. The IT systems currently in use were also included in the analysis. From the knowledge gained, the understanding of the problems was improved.

Requirements phase

First, the relevant stakeholders for such a system were identified. Subsequently, individual use-cases for such a system were identified. Finally with the techniques of Requirements Engineering the requirements for a Production Data Acquisition system were collected.

Development phase

In the development phase, the market situation was initially examined roughly. Then, potential software systems were undergone an in-depth analysis. After the check for must-have functions, the remaining tools were evaluated in a value benefit analysis.

Implementation phase

In the implementation phase, first the system architecture and then the detailed concept was developed. Therefore in the first step the single sub-systems for production data acquisition were implemented. Later the sub-systems were put together to the developed Production Data Acquisition system.

Proof-of-Concept

The most suitable system has to be verified in a proof-of-concept. So the system was deployed on three different work stations in two different production divisions. To prove the suitability for all available kind of production facilities, the system was installed at a manual work station, a semi automated assembly line and at a welding robot cell as well.

2. Manufacturing Execution Systems - MES

This section takes a closer look at so-called Manufacturing Execution Systems (MES). First, the historical development of MES is briefly discussed. Then the features and functions a MES should perform are presented and finally some future scenarios in the industrial environment are mentioned.

2.1. Introduction and History

MES is a software solution that closes the information gap between the Enterprise Resource Planning System (ERP) and the plant control system (Shop Floor) or between the planning level and the operative production areas (Gerberich, 2011). Furthermore Gerberich (2011) explains that the transparency and real-time data gained make it possible to systematically identify weaknesses in the value adding process and so deriving corrective actions is more easily. The basic idea of IT integration in the production process is nothing new. The idea of Computer Integrated Manufacturing (CIM) was already pursued in the 1980s. CIM is a collective term for various activities that are supported by IT systems in a company. The vision behind CIM is the holistic view of a company's performance, supported by integrated IT systems. The philosophy of CIM was about full automation, from the planning stage to production, everything should be controlled by computers. (Bauernhansl et al., 2014)
In figure 2 the Y-CIM model originated by Scheer (1990) is presented.

CIM failed because the necessary data systems, sensors and data transmission technology were not available at that time or were not yet powerful enough at reasonable prices (Vogel-

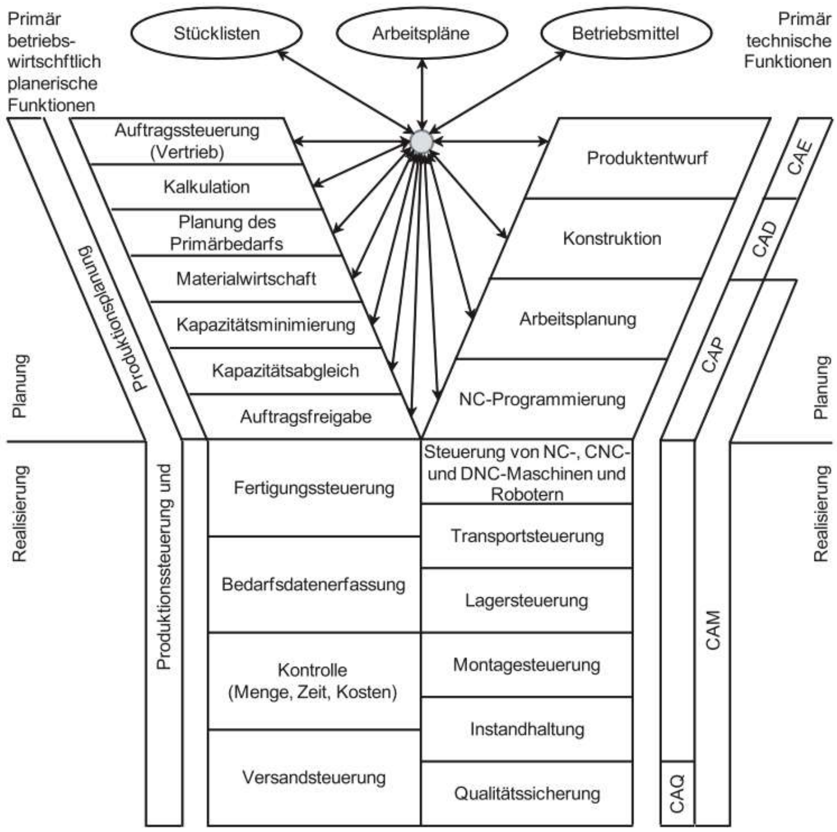

Figure 2.: The Y-CIM model by Scheer (1990)

Heuser et al., 2017).
While business production planning is mostly IT-supported within the framework of established ERP software, there were hardly any connections to physical production control. Instead, many manufacturing companies mainly use isolated information technology solutions. (Obermaier, 2016)

According to Kletti (2015) manufacturers of data collection systems came up with the idea of expanding their specialized systems and integrate related topics. Over time, three groups of data acquisition and analysis systems have been formed, some of them fulfilling multiple tasks:

- *Manufacturing*: operating data, machine data, Distributed Numerical Control (DNC), production control station, material, traceability, process data handling, tool management, energy management
- *Personnel*: personnel time recording, time management, personnel deployment plan-

ning, performance wage accounting, access control systems

- *Quality*: production inspection, complaint management, Statistical Process Control (SPC), incoming goods / outgoing goods, test equipment management, process data, escalation management, measurement data acquisition

All of the uncoordinated approaches require standardization, so a number of institutions have been working on the standardization of the MES topic. The term MES was first used in 1990 and was promoted by MESA International, among others. MESA stands for *Manufacturing Enterprise Solutions Association International*, which is a non profit organization founded in 1991. (Scholten, 2009)

In the White Paper of MESA International (1997), they have identified eleven principal functions of a MES which are shown in their Functional Model in figure 3. Those functions are: Operations/Detailed Scheduling, Resource Allocation & Status, Document Control, Performance Analysis, Process Management, Dispatching Production Units, Product Tracking & Genealogy, Labor Management, Maintenance Management, Quality Management and Data Collection Acquisition. In addition to the functions, links to other systems such as Supply Chain Management, Sales & Service Management, Enterprise Resources Planning, Product/Process Engineering and Controls are also indicated.

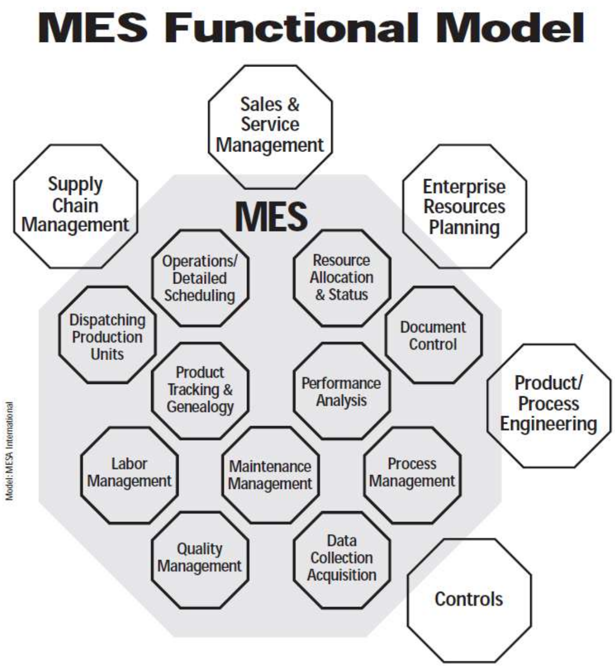

Figure 3.: The MES functions in the MESA MESA International (1997) honeycomb model

Another organization involved in the standardization process of MES is the *International Society of Automation - ISA*. ISA, founded in 1945, is also a non-profit organization whose mission is to inform the public and professionals about developments and trends in the field of automation. The definitions for a MES have been published in their standard ISA-95 with its title *Enterprise Control System Integration*. (Scholten, 2009)
One of the models presented in ISA-95 is a Functional Hierarchy model (figure 4).

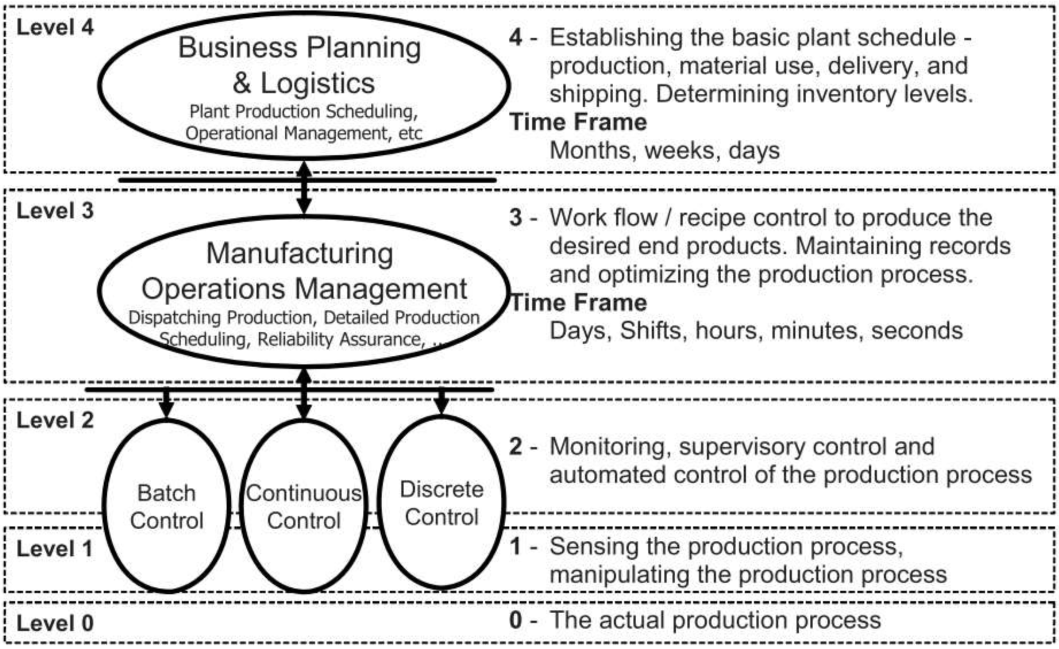

Figure 4.: The Hierarchy Model according to The International Society of Automation (2005)

This figure shows five levels of decision making processes, with different granularities of information at each level, required for them. In the highest level , the focus is on the longer term decisions (ordering materials, invoices, long term production scheduling, ...), where the time frame is on months, weeks or days. It is also called the ERP layer. Level 3 focuses on shorter term decisions like dispatching production, detailed production scheduling or machine availability. The time frame is on Days, shifts, hours, minutes and seconds. This layer is usually called the MES layer. On levels 2, 1 and 0 the production process itself takes place. Sensors, actuators, PLCs or SCADA solutions, with a timeframe of minutes, seconds or milliseconds, are used here. (Scholten, 2009)

In the German-speaking area, the *Verein Deutscher Ingenieure* (VDI) has taken the initiative and published a guideline in which the MES tasks are described in an application-oriented manner. The focus of the guideline is the presentation of the tasks and benefits of a MES system. (Kletti, 2015)

2.2. System Architecture

The classical automation pyramid consists of 6 (or 5 according to Heinrich et al. (2017)) levels where different classes of data are collected and different tasks are performed (see figure 5). The aim is to obtain an easy-to-understand model of industrial production in terms of automation. The pyramid-like structure should represent the number of components involved on these levels and the individual levels are briefly described below. (Vogel-Heuser et al., 2016)

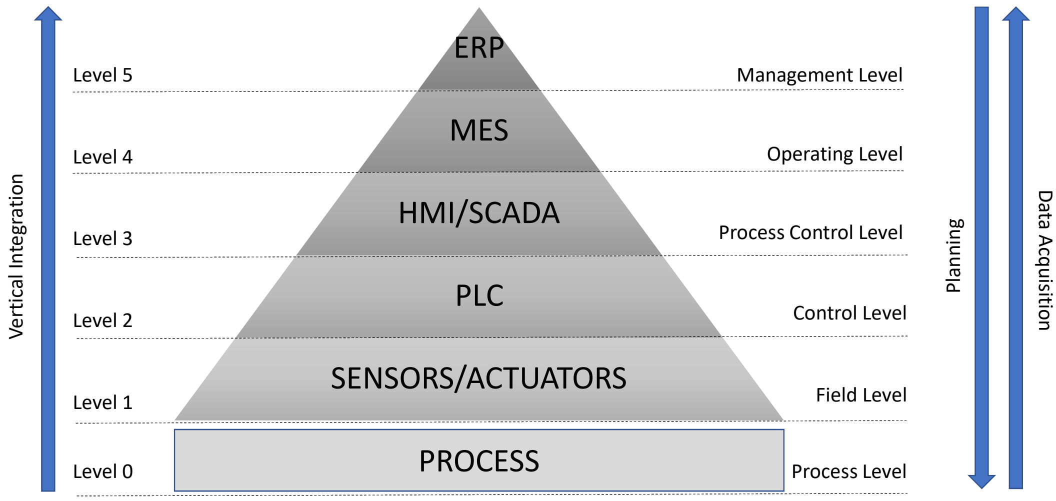

Figure 5.: Automation pyramid (authors own illustration adapted from (Armin Roth, 2016))

Level 0: On the Process Level the actual production process takes place, it forms the lowest level of the automation pyramid (Vogel-Heuser et al., 2016).

Level 1: The field level represents the technical interface to the production process, which contains sensors and actuators (Armin Roth, 2016).

Level 2: The control level evaluates input signals (sensors) via a programmable logic controller (PLC) and sends output signals (actuators) back to the field level (Armin Roth, 2016).

Level 3: The process control level visualizes production-relevant processes for the user. The control level can thus be regarded as a kind of operating and monitoring system in the form of a human-machine interface (HMI). (Armin Roth, 2016)

Level 4: At the operating level, detailed planning of the production processes takes place. This is where Key Performance Indicators are monitored and material and quality management takes place. (Armin Roth, 2016)

Level 5: The ERP system at company level is responsible for company-wide production planning and order processing in industrial production. (Armin Roth, 2016)

2.3. Functionalities of a MES

In the following section the features and functions of a MES are discussed. As Lewandowski (2011) mentioned, in the practical environment it is not necessary to have all functions implemented, this depends strongly on the production organisation and their needs. This statement is supported by Scholten (2009), who notes that most vendors have specialized in only a few of the functionalities. Therefore the chance is rather small that one finds a vendor who can offer all functionalities in one integrated system. Scholten (2009) has also limited these MES functionalities in an initial overview to the following modules:

Detailed production scheduling: This can include an inventory scheduling to get an insight to the final product inventory and the required raw material. Also the production employees can be dispatched to the respective workstations. An important point is the optimum job sequence to optimize set-up times and thus increase efficiency. It should also be possible to respond to a machine breakdown and thus redesign the detailed production plan. Furthermore, an automatic interface to level 4 systems should be provided.

Production Definition Management and Production Execution Management: This modules should cover the management of information about how a product should be produced. These can be recipes, assembly instructions, flow charts or Standard Operating Procedures (SOP). Furthermore, this can also include the automatic download of product-specific data to the PLCs. An advantage can be the central management of master recipes or instructions.

Historians: This are so called "flat" databases in contrast to relational databases. They are used to store massive quantities of process data, such as temperatures, pressures or other measurements. In order to realize the required performance, they use different storage and retrieval strategies, like compression. The data stored in this databases are the basis for other modules like the reporting modules.

Reporting, Tracking & Tracing: Reporting modules are used to display the recorded data in an aggregated form. Manufacturers often provide standard reports on OEE, quality, shifts, efficiency, capacity utilization and so on. In practice, however, it has often turned out that people want to use their own company-specific reports. This often involves additional implementation work and costs. Tracking & Tracing is used to record the production history of a product. All information from the raw material to the finished product is stored.

Dashboards: Used to provide information on the production process to the company's decision-makers at first glance. Such information can be the OEE or the order progress on a production line. So a dashboard can be a window in which the user can see information about different systems on a single screen. More detailed information can also be provided by other systems.

Workflow Management: Workflow management is required to react adequately to unscheduled events. This means that users should be informed through pre-defined information channels. In addition, the initiated measures should be supported by authorization and follow-up actions.

Interfaces with ERP: The interfaces are not only used to transmit information (bill of materials, master data, production plan) from the ERP system to the Level 3 system, but also to report back data from the production process. These information can be for example the raw material consumed, labor efficiency, production volumes, scrap rate or machine downtimes.

Plant Model: For each MES application, the plant model is the basis of the production process and its associated information. It shows the configuration of the production processes with the products to be manufactured and the materials and equipment used.

Another approach for defining the functionalities of a MES provides the guideline VDI 5600 (2016), which also distinguishes between eight different functions a MES should have:

Detailed planning: Supports the completion of the job queue and takes production restrictions into account.

Resource management: Ensures the on time availability and technical functionality of the needed resources for an order.

Material management: Ensures that production is supplied with the right material on time and that finished goods are transported away. The work in progress should also be managed. The tracking of Batch and serial numbers is also attributed to this function.

Personnel management: Provides personnel from the available personnel pool with suitable qualifications for the production process on schedule.

Data acquisition and processing: Collects event-driven, automatically, or manually data from the process. It also includes the verification and processing of this data.

Performance analysis: Includes the evaluation and analysis of processes with the aim of realizing short control loops in order to influence the process operationally based on the determined target/actual deviations. On the other hand there are long control loops to optimize the process in the long term. In addition key performance indicators are calculated and displayed.

Quality management: Supports the assurance of product and process quality through quality planning, quality control and test equipment management within the production process.

Informations management: Contains the preparation of data from the MES functions or from other company divisions and further the distribution of this information.

2.4. Advantages of a MES

After the functions of an MES have been explained, one has to ask what concrete benefits such systems can bring for a company.
In his work Gerberich (2011) points out two different directions of advantages. On the one hand, the customer benefits from increased production responsiveness and, on the other hand, the user gains significant transparency in production. Furthermore, he structures the potential benefits of MES into operational and strategic improvements, which are based on process improvements.
In her book Scholten (2009) also states that there is a connection between MES functionalities and the improvement of operational and company-wide performance.

Kletti (2015) divides the benefits into monetarily assessable and non-monetarily assessable benefits. Monetarily assessable effects are the increase in quality and productivity (plants and employees), reduction in lead time and inventories. The non-monetary benefits are: Improvement of transparency and responsiveness, increase of flexibility, increase of punctuality, increase of customer satisfaction, fulfilment of external requirements.

2.5. Selection and Implementation of a MES

After the decision to implement a MES is made, a structured approach should be considered. For this reason, various approaches mentioned in literature are reviewed in this section.

Scholten (2009) presents a four-step approach in which the scope is first to be roughly defined, then the user requirements are to be determined, a system is to be selected and finally integrated by a system integrator. To define the scope the models from ISA-95 can help well. It is important to think about which of the offered functionalities should be implemented and which should not. She also points out that it is important to know which requirements this system must meet in the future in order to avoid building up an unnecessarily complex system mix. The next step is to determine the requirements for a future MES. For this purpose, Scholten (2009) proposes a procedure based on a Requirements Engineering approach.
In the selection process a multi-stage procedure is proposed. First, a long list of possible vendors should be created. Due to the check for several criteria, the long list should be reduced to a short list. The remaining vendors on the short list should receive the requirements document and provide further requested information. A Question & Answer session should be organized to answer all possible questions at one appointment. Afterwards the vendor should present his solution, this should also include a live demo. If there is still no clarity, a reference visit can still be held or a Proof-of-Concept can be agreed. After this process, a system can be selected and implementation can be started by the system integrator.

Louis (2009) reports in his work about two different main approaches in the selection of a system that supports the function groups of the MES level. The *one-stop-shop* approach, in which the existing ERP system is extended by functions that support the production processes, on the one hand and on the other hand the *best-of-breed* approach, in which a independent

software solution for MES tasks is being implemented.
Louis (2009) suggests a four-phase process model for selecting an MES. The four phases are:

- Characterization of the production system
- Derivation of the requirements for the MES level
- Identification of the previous IT support and scenario creation
- Evaluation and selection.

The requirements for the MES level are partly determined by the different production systems. It is assumed that the requirements for a MES can be derived from the production system. Once the requirements have been derived, it is checked to which degree they are already supported by existing IT systems. It is not only a pure recording of the actual situation but also an evaluation of the support respectively. He chooses a different approach and proposes first to select possible software systems in order to then develop detailed scenarios for each product. These scenarios are then evaluated on the basis of various criteria with the aid of a value benefit analysis. A Total Cost of Ownership (TCO) model is used for the cost analysis. To select a scenario, the results of the value benefit analysis and the TCO must be combined.

Gerberich (2011) has summarized the selection of a MES system in the following way. First of all, the company has to ask itself a number of questions:

- Which tasks should be fulfilled by a MES?
- What information is required, for example machine availability, performance data or quality?
- Are single CIM modules sufficient?
- Can an extension of the existing ERP system bring the desired success?
- Have the gaps in the IT structure been identified?
- Are the objectives of the MES project and system requirements precisely defined?

In the next step, a suitable software vendor can be selected. To simplify the selection process, selected industry journals can be used.
In order to guarantee the success of the project, requirements for the MES provider on the one hand and on the other hand for the MES software itself must be fulfilled. Once a software has been selected, the required range of functions is to be defined and recorded in a specification document. Subsequently, the implementation and achievement of the goals is reviewed in a

verification process.

Meyer et al. (2009) starts the selection process with the analysis of the actual situation in the company. First of all, the existing infrastructure and processes are examined in more detail in order to define the required functions. When selecting a suitable system, the market situation should first be surveyed so that a selection of maximum three providers can then be put on the short-list. These systems will be subjected to a more detailed analysis in order to reach a decision afterwards. During the subsequent implementation, a detailed project plan is to be drawn up and a project manager should be appointed on the customer side as well. The customer's requirements must be converted into the provider's specifications. The progress of the project must be presented in regular meetings and not only to the project manager, but also to the core team.

In their paper Weissenberger et al. (2015) proposes a model-based approach that automatically generates MES functions. This method should help to reduce the implementation costs of MES. They defined three steps to generate Manufacturing Execution Systems:

- creation of a plant model
- creation of a model of the production process
- creation of a functional model of the MES

MES-ML was used as the modelling language, a modelling language for the interdisciplinary specifications of MES. The method was evaluated by modelling the production process of an industrial brew house and was also verified by MES engineers.

Brunetti (2016) took the following approach when implementing a MES in a steel plant. First, the business processes and the overall system architecture were defined. These results were then used as a blueprint to equip the plant step by step with the MES. This approach was chosen because the plant was in constant operation and therefore the previous system cannot be replaced in a single shot.

In general, it can be said that there is no single approach that is always valid. The circumstances and the environment of a company must always be taken into account, in order to select the most suitable procedure for implementing such a system.

2.6. Future of MES

Kletti (2015) claims that since the term Industry 4.0 has found its way into the manufacturing industry, there have been various approaches to revolutionizing this sector. The use of information technology and the increasing integration of objects and systems should make industrial production processes much more efficient. So this initiative has brought the production organisation back into focus. Integration and data exchange alone will not increase productivity. What is needed are applications that take advantage of these new technologies. According to Kletti (2015), many pioneers dream of the ultimate networking of all resources and systems involved in production. Every machine should know about its capabilities and every material should know which article is to be created from it. According to Kletti (2015), this theory has already been abandoned somewhat, but a more practical idea is that every resource in the production environment can be clearly identified and communication with every resource is possible through comprehensive networking. Furthermore the individualization of the single production steps will increase and the machines will become more flexible and intelligent.

Kletti (2015) also names future challenges for MES. In particular, he points out that the systems should become more flexible in order to allow short-term changes of the order to flow into the production process. Furthermore, systems must become more configurable so that customizing through programming becomes an exception. It must be possible to analyze the large amounts of data in real time, in order to exploit the full potential. According to Kletti (2015), MES are ideal for information and data rotation in a Smart Factory. The aim is to synchronize all systems involved in production.

Arica and Powell (2017) name in their paper seven Industry 4.0 key technologies that can be opportunities for MES in Industry 4.0 applications. They also propose that the MES, with its functionalities, data collection, analysis and communication, will serve as a platform to implement future Industry 4.0 technologies.

Meyer et al. (2009) suggest that the different systems, like ERP, PLM and MES, must grow closer together. They also point out that it will be important in the future to have a consistent data model in which product data and related processes are described in detail. It is proposed that the product is the central object of this data model and that all occurring data are assigned to this object.

Another future scenario would be distributed manufacturing using the features of Manu-

facturing Execution Systems. Helo et al. (2014) and also Alexakos and Kalogeras (2017) propose a cloud manufacturing solution for distributed manufacturing with the use of MES. Furthermore, the production of small series poses new challenges for many companies and their MES. Cupek et al. (2016) propose an MES architecture based on a multi-agent system. Urbina Coronado et al. (2018) have described in their work the development and implementation of a low-cost MES using smart devices and cloud computing. This solution can be of particular interest to small manufacturing companies.
The examples have shown that there is still no uniform path for the future of the MES. However, there are many ideas and approaches on how MES will continue to be a part of every modern factory in the future.

3. Requirements Engineering

Despite enormous efforts in the past decades in the field of software engineering, the planning and implementation of complex software-supported systems are still confronted with technical and economic risks, which are very often caused by an insufficient Requirements Engineering (Partsch, 2014).
This chapter should therefore give an overview about the importance of Requirements Engineering. First some definitions regarding Requirements Engineering are given, then the fundamentals of Requirements Engineering are discussed. Further a typical Requirements Engineering process is described. Finally, a new method, the Use Case Technology Mapping Framework, is presented in which both use cases and technologies are identified, in order to map them to new services.

3.1. Definitions

Before the individual steps of Requirements Engineering can be explained in more detail, some fundamental terminology, like *Requirements Engineering*, *Stakeholder*, *System* or *Functional* and *Non-Functional Requirements*, have to be worked out first.

3.1.1. What is Requirements Engineering?

According to Glinz (2009) the method of Requirements Engineering is used for the systematic identification, management and evaluation of requirements. Requirements engineering is typically used in the development of software systems or software-intensive systems. The

goal is a complete, explicit and consistent specification of all requirements. The second approach that Glinz (2009) indicates is that RE is about understanding and describing customer needs. The overall goal should be satisfied customers. The third approach assumes that RE serves to reduce the risk associated with system development. The effort required for RE should be in inverse proportion to the risk that a company is prepared to take.
Partsch (2014) mentions two meanings of the term Requirements Engineering. On the one hand, it stands for the specific activities at the beginning of a system development, but also for an entire sub-discipline at the interface between systems engineering, computer science and application sciences.
A definition of Requirements Engineering is also given by the IEEE (Institute of Electrical and Electronics Engineers, 2011) in the following way:

> "Requirements engineering is an interdisciplinary function that mediates between the domains of the acquirer and supplier to establish and maintain the requirements to be met by the system, software or service of interest. Requirements Engineering is concerned with discovering, eliciting, developing, analyzing, determining verification methods, validating, communicating, documenting, and managing requirements."

Still, and far too often, the requirements are only defined during designing phase of a system or even after its completion. Although then the chance that the requirements and the developed system match are much higher but the consideration of all stakeholder wishes is usually not given. (Partsch, 2014)

3.1.2. Stakeholder

A very important factor for Requirements Engineering are the stakeholders of a system. Stakeholders are persons or organizations, who have directly or indirectly an impact on the system. Or in other words, stakeholders are the direct or indirect deliverer of information for a system to be developed (Rupp, 2014).
According to Partsch (2014) and DIN 69905 typical examples of stakeholders are: users, operator, developers, architects, customers and managers of the system.
According to Glinz and Wieringa (2007) those are persons or organizations who:

- are users of the system or directly involved in the process the system will change
- must manage, introduce, operate or maintain the system
- are involved in developing the system
- are responsible for the business or process the system supports
- have a financial interest
- constrain the system as regulators
- are negatively affected

The IEEE standard 29148:2011 defines that a minimum set of stakeholders consists of users and acquirers, who may not be the same. But the projects requirements can necessitate including two more groups to the minimum set of stakeholders. Those are the organization developing, maintaining or operating the system and second regulatory authorities can have legal requirements.
Rupp (2014) remarks that before starting with a project it is important to know who your stakeholders are. The definition of system targets and -requirements should not be specified by only one person, but by many stakeholders. Because forgotten stakeholders lead to preventable and expensive change requests.
Rupp (2014) also suggests to document the elicited information about relevant stakeholders systematically. The simplest method is to write them down in a table. For more complex projects databases are more useful. A minimum set of data for stakeholders are: name, role, contact details, availability, relevance, knowledge, interests and goals.

3.1.3. System

The system context is the part of the real world that is relevant for the understanding of the requirements and the complete characterization of a system. The system context is important for system requirements, because it can restrict possible interpretations. There can also be direct or indirect relationships of requirements to elements of the system context. (Partsch, 2014)
As shown in figure 6 the system to be developed is surrounded by a *system boundary* that demarcates it from the environment. At the beginning of the RE process the system boundary is not defined clearly yet, so it is surrounded by a grey area. The system has sources and sinks through which the system is interacting with the environment. (Pohl, 2008)

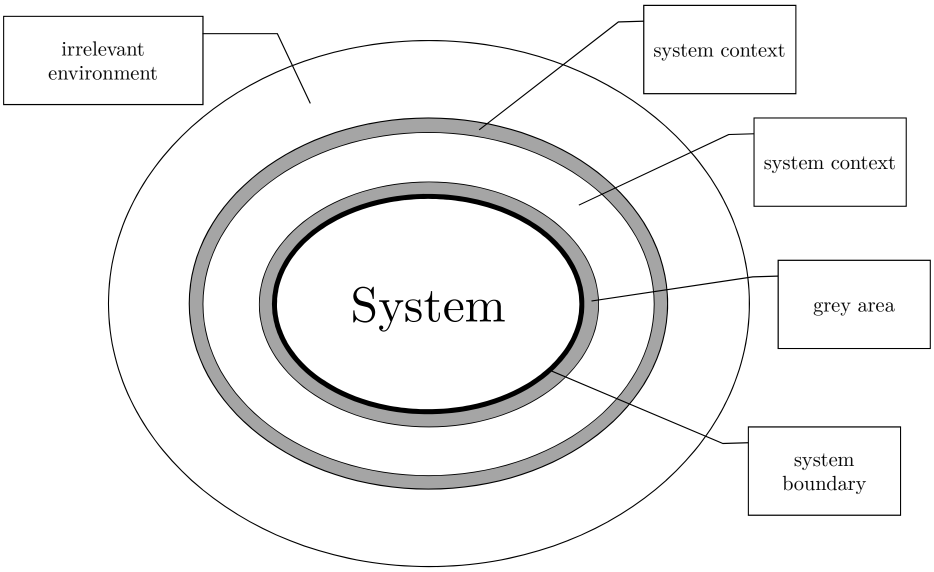

Figure 6.: The system and its context (Authors own illustration based on Pohl (2008))

In his book Balzert (2009) defines the relevant environment of a system as *system context*. Further the environment that has no impact to the system is called *irrelevant environment*. Between the relevant and irrelevant environment exists a grey area as well.
As noted by Pohl (2008) the system context has a significantly effect on the requirements. Consequently it is important to define the system and its environment precisely.

3.1.4. Requirements

There are a multitude of categories for requirements. Rupp (2014) is talking about functional requirements, technological requirements, quality requirements, requirements regarding the user interface, requirements for other delivery components or legal and contractual requirements. In contrast, Glinz (2009) makes the following classification: functional requirements, attributes, constraints, performance requirements and special qualities.
Altogether there are requirements concerning the functions of a system and those concerning everything else (Glinz, 2007). So all classifications have in common that functional requirements and non-functional requirements occur in some way.

Functional Requirements

Functional requirements describe, as the name indicates, the functions a system should be able to fulfill. According to Glinz (2007) there are two threads defining functional requirements: those that are focusing on functions and those that are emphasizing behaviour. Further Wiegers and Joy (1999) tried to synthesize the two trends in one definition. They claimed that functional requirements are:

> "...a statement of a piece of required functionality or behavior that a system will exhibit under specific conditions."

In the standard of Institute of Electrical and Electronics Engineers (2011) functional requirements are defined as follows:

> "Functional requirements describe the system or system element functions or tasks to be performed."

Functional requirements are referring to the functional aspects of a system. This requirements are the outcome of answering questions like "*What is the system doing?*" or "*What should it do because of the scope?*".
Partsch (2014) also claims that information about the structure of a system or relevant system states are functional requirements.

Non-Functional Requirements

In most system development projects the main focus is on functional requirements of a system, because these functions are those that bring users the added value and support them in their daily work (Rupp, 2014). But there are also other factors that are important for the project's success: non-functional requirements.
The understanding, of what non-functional requirements are, differ and there is no consensus about the nature of non-functional requirements and how to document them in requirements specification (Glinz, 2007).

A very general definition was made by Rupp (2014), who said that non-functional requirements are all requirements that are not functional requirements.
Balzert (2009) says that non-functional requirements describe aspects, that typically concern or cross-cut all functional requirements. In his book Partsch (2014) categorizes non-functional requirements in four groups: *quality attributes of the desired functions*, *requirements for the implemented system as a whole*, *requirements for the implementation of the system development* and *requirements for testing, introduction, supervision and operation*.
Rupp (2014) made a distinction between the following categories: *user interface*, *quality*, *technology*, *legal and contractual*, *activities to be carried out* and *other delivery components*.
Glinz (2007) claims that the problems with non-functional requirements can be divided into three categories: *definition problems*, *representation problems* and *classification problems*. So there is no common definition for non-functional requirements. Moreover, there are numerous different classifications of non-functional requirements. The representation problem describes that many non-functional requirements can be converted into functional requirements by changing the formulation. However, this does not apply to all of them.

3.2. The process of Requirements Engineering

In this section the relevant steps in a typical Requirements Engineering process are described. There are different manifestations of the Requirements Engineering process. One of them is the *SOPHIST-Regelwerk* by Rupp (2014), which is described in the following steps.
This process consists of four main phases and starts with the elicitation or more precisely with the pre-elicitation. Then the elicited requirements are specified and documented. Later the requirements are validated and managed by the Requirements Management.

3.2.1. Requirements elicitation

At the beginning of the RE process it is important to gain knowledge about the context and the limits of the system to be developed. Furthermore, the goals are to be defined and the sources of requirements are to be identified. Sources of requirements are stakeholders, but also documents or systems in operation. The system requirements are then determined on the basis of these sources. (Rupp, 2014)

Pre-elicitation

According to Rupp (2014), the pre-elicitation phase consists of three basic tasks. On the one hand, the goal that the system to be developed has to meet has to be defined, sources of requirements have to be identified and the system has to be differentiated from its environment by a system boundary. Relevant requirements sources can be:

- *Stakeholder:* The term stakeholder and what they are is described in detail in section 3.1.2.
- *Documents:* Documents often contain important information from which requirements can be derived. Requirements can be found, for example, in general documents such as norms and standards. In addition, documentation from old systems can also serve as a source of requirements.
- *Systems in operation:* Systems in operation can be predecessor systems as well as competitor systems. Based on the experience of the current system, stakeholders can demand extensions or changes.

A goal is the intentional description of a characteristic feature of the system to be developed or the associated development-project desired by stakeholders. Therefore, goals have a great influence on the success of the project. Objectives should be clearly defined, as they are the starting point for the subsequent requirements analysis by the requirements engineer. Undefined goals cannot be pursued and are therefore usually not achieved.
Furthermore the system and the environment of the system should be considered. This process is important because generally only requirements for parts, that are within the system boundaries, are created. (Rupp, 2014)
A detailed look at the system, environment and context has already been made in section 3.1.3.

Elicitation

After the system boundary has been defined, the stakeholders have been identified and the goals have been set, the actual requirements can be elicited. One of the difficulties in determining requirements is to bring the knowledge of the stakeholder from the subconscious to the surface. Different investigation techniques can support, some of them are explained in the following. (Rupp, 2014)

Observation techniques
This technique is particularly suitable if the stakeholder is not in a position to express his know-how linguistically. Also, some stakeholders do not have the time to participate in the determination of requirements. The requirements engineer observes the users of a system and derives requirements for the system to be developed from his observations. Observations alone run the risk of documenting outdated technologies or processes in need of improvement. However, the requirements engineer often has the necessary distance to identify inefficient processes and suggest improvements. All observations should be discussed with the stakeholders in order to eliminate misunderstandings in advance. Basically, a distinction is made between field observations and apprenticing. At Apprenticing, the Requirements Engineer learns the activity from the user in order to be able to form an exact picture of the work processes for himself. However, it is important to be aware that the situation of the actual system and not that of the target system will be known. (Rupp, 2014)

Interview techniques
The direct interview techniques are the classics in requirements analysis. Stakeholders are asked specifically about their wishes and needs and, if they are aware of them and can name them linguistically, they will also name them. In principle, a distinction is made between written (questionnaire) and oral (interview) interview techniques. Questionnaires have the advantage that they can be standardized and thus also evaluated automatically. Questionnaires can also be used to interview a large number of stakeholders simultaneously (online questionnaires). However, questionnaires are poorly suited for determining implicit knowledge. Some types of requirements (non-functional requirements) are difficult to identify through questionnaires because they are difficult to quantify. In an interview, on the other hand, arising questions can be answered immediately. This also enables the Requirements Engineer to recognize new requirements or uncover implicit requirements. (Rupp, 2014)

Questionaires

In a questionnaire, a series of closed and open questions are asked, with their assistance the knowledge of the stakeholders is determined. The questionnaire can be done electronically or on paper or can be sent to a large target group as an online questionnaire. Respondents should be adequately motivated to return the completed questionnaire. The most important questions should be asked at the beginning so they are not left unanswered if someone does not complete the questionnaire to the end. With the help of a questionnaire, a large number of stakeholders can be included in the analysis in a short time. Questionnaires, on the other hand, are poorly suited for determining implicit knowledge. Since the questions are asked in written form, it is difficult to ask any further questions. (Rupp, 2014)

Creativity Techniques

Creativity techniques help to break up thinking in conventional ways and help to make room for unusual ideas. They are particularly suitable for developing the first vision of a system, gaining an overview and collecting innovative ideas. Unconscious ideas can be identified. Some of the creativity techniques are *brainstorming*, *6-3-5 method* or *The Change of Perspective*. (Rupp, 2014)

Use case modelling
An use case makes it possible to document the functionality of a system abstractly without immediately getting lost in details. A use case represents a service of a system that offers added value for a user. Use cases can be written down, but can also be represented in so-called use case diagrams. The use case diagram according to the SysML notation is a feasible modeling language. (Rupp, 2014)

3.2.2. Requirements specification

In the previous section it has been presented how requirements can be determined, now these requirements have to be specified in a structured way. Documentation techniques are suitable for this purpose in order to bring the requirements to paper. There are a variety of techniques for documenting requirements. For each application it has to be considered which technique

is appropriate in the given context. The simplest way would be to use natural language to write down the requirements. One advantage of this is that none of the stakeholders need to understand any complicated notation. Instead of natural language, diagrams can also be used. Use case diagrams, activity diagrams, sequence diagrams or the system-use-case diagram are particularly suitable for this. (Rupp, 2014)
Those diagrams are presented with the Systems Modeling Language in section 4.3.

Use-Case-description
A use case diagram only provides an overview of the functionalities of a system, therefore the individual use cases must be described in more detail.

Section	Content
Designation	UC-12-37
Name	Navigate to destination
Authors	John Smith, Sandra Miller
Priority	Importance for system success: **high** Technological risk: **high**
Criticality	High
Source	C. Warner (domain expert for navigation systems)
Responsible	J. Smith
Description	The driver of the vehicle types the name of the destination. The navigation system guides the driver to the desired destination.
Trigger event	The driver wishes to navigate to his destination.
Actors	Driver, traffic information server, GPS satellite system
Pre-condition	The navigation system is activated.
Post-condition	The driver has reached his destination.
Result	Route guidance to the destination
Main scenario	1. The navigation system asks for the desired destination. 2. The driver enters the desired destination. 3. The navigation system pinpoints the destination in its maps. 4. On the basis of the current position and the desired destination, the navigation system calculates a suitable route. 5. The navigation system compiles a list of waypoints. 6. The navigation system shows a map of the current position and shows the route to the next waypoint. 7. When the last waypoint is reached, the navigation system shows "destination reached" on the screen.
Alternative scenario	4a. Calculation of the route must honor traffic information and avoid traffic congestions. 4a1. The navigation system queries the server for updated traffic information. 4a2. The navigation system calculates a route that does not contain any traffic congestions.
Exception scenarios	Trigger event: The navigation system does not receive a GPS signal from the GPS satellite system.
Qualities	→ QR.04 (reaction time upon user input) → QR.15 (operating comfort) *(QR = quality requirements)*

Figure 7.: Example for an use-case description by Pohl and Rupp (2015)

A frequently used method is the use case description, which documents the use case in natural language, but in a form style notation. This also ensures that certain information is retrieved that would otherwise be often forgotten. In figure 7 an example for a tabular use-case description is shown. If required, further information can be added to the table. (Rupp, 2014)
The structured description in natural language is accepted by most stakeholders and makes it

easier to selectively capture the information contained. Use cases usually describe only a part of the requirements, so that in most cases another technique is required. (Rupp, 2014)

Requirements template

Rupp (2014) has developed a template (see figure 8) that can be helpful in the specification of requirements. Thus, the requirements are syntactically structured in such a way that typical formulation errors can be excluded from the beginning. Furthermore, a similar structure is imposed on each requirement. Therefore, a requirement template is like a construction plan that defines the structure of an individual requirement sentence.

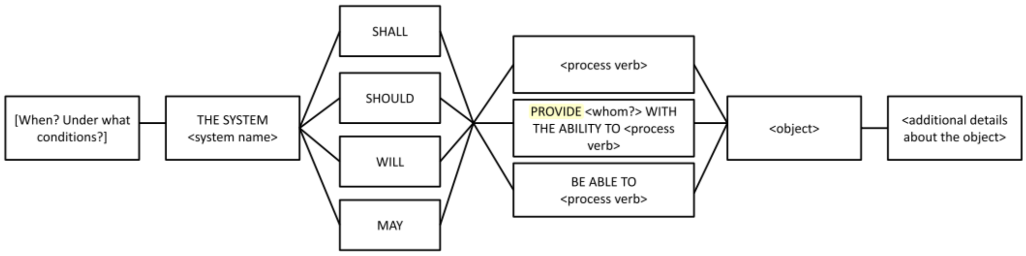

Figure 8.: Requirements template by Pohl and Rupp (2015)

Rupp (2014) shows some examples of requirement sentences built with the template:

> *If a video is returned AND if this video was lent the 30th time, the library system shall display a message that the video has to be replaced.*
>
> *If no library item is lent by a customer, the library system shall provide the librarian with the ability to remove this customer.*

The word *shall* is used to indicate that this requirement is mandatory.
The word *should* is used to define requirements that are not mandatory but increase stakeholder satisfaction.
The word *will* is used to define requirements to be integrated in the future. They help to prepare the current solution for these requirements.

3.2.3. Requirements validation and evaluation

Once the requirements have been formulated, they must be subjected to a quality management process. The aim of quality assurance is to define stakeholder requirements in such a way that they meet the required quality standards. There are a number of techniques for this, an excerpt of them is briefly presented below.

Reviews
The review is one of the most frequently used test techniques, which helps to detect anomalies in requirements. A distinction is made between comment, walkthrough and inspection. (Rupp, 2014)

- *comment:* A selection of requirements is given to a colleague to review.
- *walktrough:* The examiners are guided step by step through the test object. The development process and ideas are explained. The auditors ask questions and try to identify problems.
- *inspection:* The inspection is carried out in several successive phases: Trigger, planning, pre-meeting, individual preparation, review meeting, follow-up and evaluation, conclusion.

Prototype
The requirements are partially implemented in a prototype to test their feasibility. Prototypes are also a good way to uncover missing or unsuitable functionalities. Prototypes can be used in any phase of the analysis. Physical prototypes convey a feeling for the appearance or behaviour of the product to be developed. They also increase the imagination of complex systems. But the development is time-consuming and often a wrong impression of the maturity of the product arises. It is often assumed that the product will be finished soon. (Rupp, 2014)

Test Cases
In requirements engineering, test cases serve to qualitatively improve the requirements within the analysis. As an analytical quality assurance measure, it is extremely effective to create test cases for requirements, because hardly anyone reads requirements as intensively as a test case builder. A test case is therefore an instruction for testing a requirement that describes

the fulfillment of the requirement in the created product. It is advisable to create the test cases already during the requirements engineering process, since the requirements can still be changed easily in this phase. (Rupp, 2014)

3.2.4. Requirements management

The fourth main activity of Requirements Engineering is its management. Requirements Management includes all necessary steps to document, change and track requirements.
A main reason for requirements management is the fact that requirements may change during the lifecycle of a project.
Another reason for Requirements management is the fact that requirements are re-used. Requirements Management helps to master these challenges with as little effort as possible. Rupp (2014) names four main tasks for RM:

- exchange of information
- process control
- Management of internal contexts
- Evaluation and project management

It is important to assign a unique ID to the requirements. Only this object ID ensures that all persons involved always speak about the same information. A requirement passes through various states during its lifecycle. Possible states can be: deleted, created, approved, archived, quality tested, designed, analyzed, tested and implemented.

3.3. Use Case Technology Mapping Framework - UCTMF

The Use Case Technology Mapping Framwork (UCTMF) set up by Vorraber et al. (2018) and presented in this section, provides guidance how to identify use cases and technologies and map them to new beneficial services. It is based on two underlying concepts: a human-centered and process driven approach and a technology driven approach.

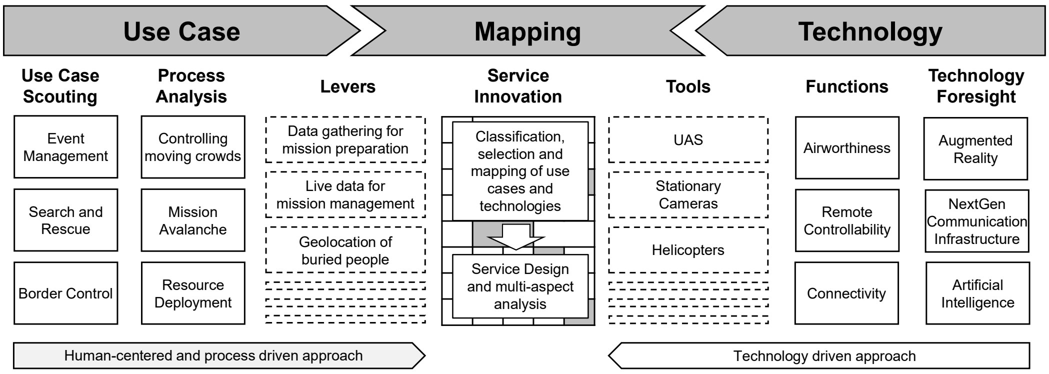

Figure 9.: The Use Case Technology Mapping Framework by Vorraber et al. (2018)

As represented in figure 9 the UCTMF consists of three concurrent main process steps, the needs of users or an organization (Use Case arrow), existing or upcoming technological possibilities (Technology arrow) which should be mapped to create innovative and beneficial services (Mapping step).

Use Case phase

In this phase a continuous process of use case scouting is being performed. In this process areas with specific relevance for the organization or the user are identified. This identified areas are the base for potential improvements, so called *improvement levers*.

Technology phase

In this phase a continuous process of technology scouting and foresight is being performed. It is followed by a categorization and assessment to ensure relevant technologies for the organization are outlined. Generally the approach can be followed by two ways:

- *Technology triggered path*: What does the technology offer and how could it be used to meet the customer needs. New technology can be a trigger for process adaption or innovation.
- *Use case triggered path*: How could this technology be used to support the implementation of in the use case phase identified improvement levers.

Both main processes, Use Case identification and Technology Scouting, should be performed continuously and iteratively to lead to the mapping process.

Mapping and Design phase

In the last step the most promising use cases are mapped with the best fitting technologies to create new service concepts. This mapping procedure is ranking the needs and simultaneously assessing how the offered functionality by the scouted technology is meeting the users needs. The resulted combinations are the starting point for the service design and engineering process to develop service concepts.

4. Systems Engineering

Systems Engineering was probably used for the first time in the Bell telephone laboratories to manage their interface problems in 1940 (Winzer, 2013). Later this method was applied in the development of NASA's Apollo program (Winzer, 2013). Over the years Systems Engineering has become a methodology that can be used to solve problems in any engineering discipline and beyond (Haberfellner et al., 2015).
For this purpose, the fundamentals of system engineering based on Haberfellner et al. (2015) are represented in this chapter. The concept of Systems Engineering is a methodology for solving any kind of problems. Systems Engineering is based on two main components, the Systems Engineering philosophy and the problem-solving process. This concept is illustrated in figure 10. The Systems Engineering philosophy should be understood as the mental framework for systems thinking. The problem-solving process is divided into a content part, the system design, and an organizational part, the project management.

4.1. Systems Engineering Philosophy

The Systems Engineering philosophy should be the mental framework for this method. This is divided into systems thinking and the application of a process model.

4.1.1. Systems thinking

The essence of systems thinking is the decomposition of complex facts into individual aspects and their interconnection (Winzer, 2013). In this way, the detail can first be solved without

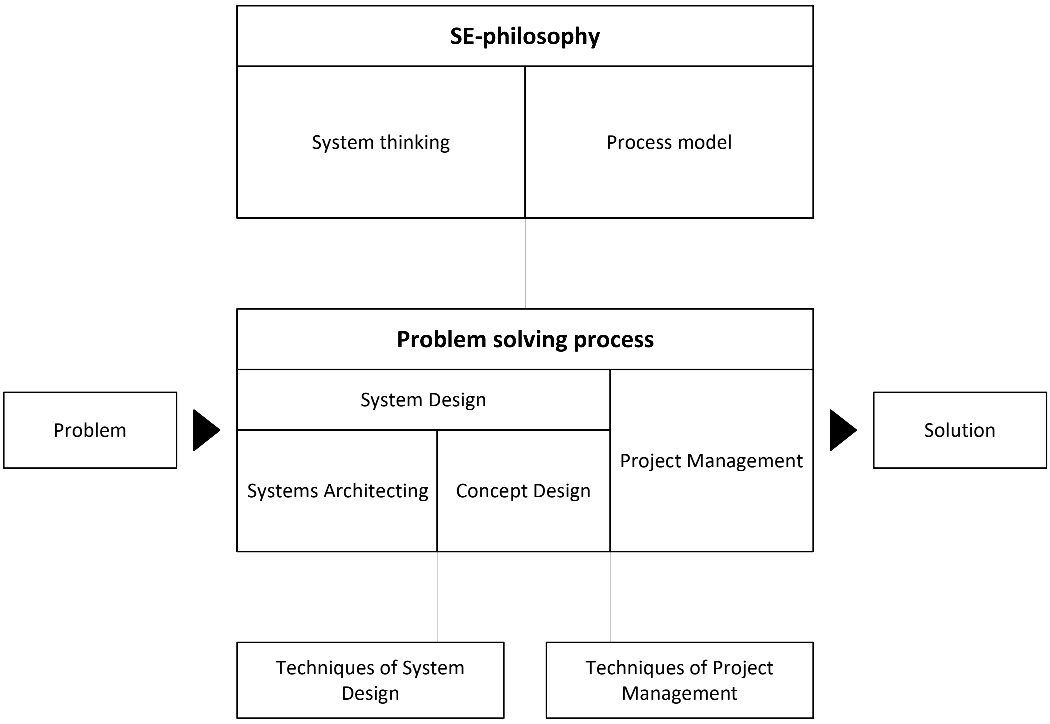

Figure 10.: The concept of Systems Engineering

losing the connection for the whole. This system thinking makes a complex task clearer and therefore easier to solve.

If the word *system* is looked up in the dictionary, a multiplicity of meanings is obtained. This shows that in normal language there are many meanings for the *system* term. There are physical, biological, chemical, technical systems. Furthermore, there are banking systems, political systems, traffic systems and last but not least IT systems. (Haberfellner et al., 2015) Beer (1963) has formulated an appropriate definition for the term system:

> "The word system stands for connectivity. We mean any collection of related parts. What we define as a system is a system because it comprises related parts and, in a sense, forms a whole."

According to Haberfellner et al. (2015) systems consist of elements and their relations to each other. These elements have properties and functions and can in turn be regarded as systems again. The relationships between systems are very general and can be for example material flow, information, location, etc.. When you work with the system term, you also have to define how a system is delimited from its environment. Systems are delimited from their

environment by a system boundary. There may be peripheral systems in the environment that can still be related to the considered system. The system boundary does not have to be physically visible, it can also be only of a mental nature.
Models can be used to illustrate the complex connections in systems. It must be noted that models can only offer a simplified view of reality. There are different perspectives for systems: environment oriented perspective, impact oriented perspective or structure oriented perspective.
When looking at the environment oriented perspective, the system is first neglected and only the connections between the system and its environment are considered. For example, the stakeholders and their interaction with the system can be examined.
The effect-oriented view can also be described as black box view. Here it is checked which input variables lead to which output variables. This can be balances of any kind (energy balances, material balances,...), but also every efficiency calculation (output to input) follows this principle.
The structure-oriented view takes a closer look at the elements of a system and their relationships to each other. Using this view, it is possible to explain how the output results from the input.

4.1.2. Process models

The second important component of the SE methodology is the process model. There are a variety of process models with different advantages and disadvantages. A distinction between plan driven and agile methods is made.

4.1.2.1. Plan driven methods

The plan driven methods bring a logical process structure into projects. But these methods are vehemently criticised, because they make the development process unnecessarily cumbersome and require long development times. Another point of critique is the difficult and long-running handling of changes in the requirements, during the development process. Nevertheless these models provide a high level of structure and combine it with the expectation of a high quality and efficient solution. (Haberfellner et al., 2015)

Waterfall model

The waterfall model is one of the most known process models in software development. It follows a classical top-down procedure. The project is subdivided in single project phases, so the outputs of one phase are the inputs of the following one.

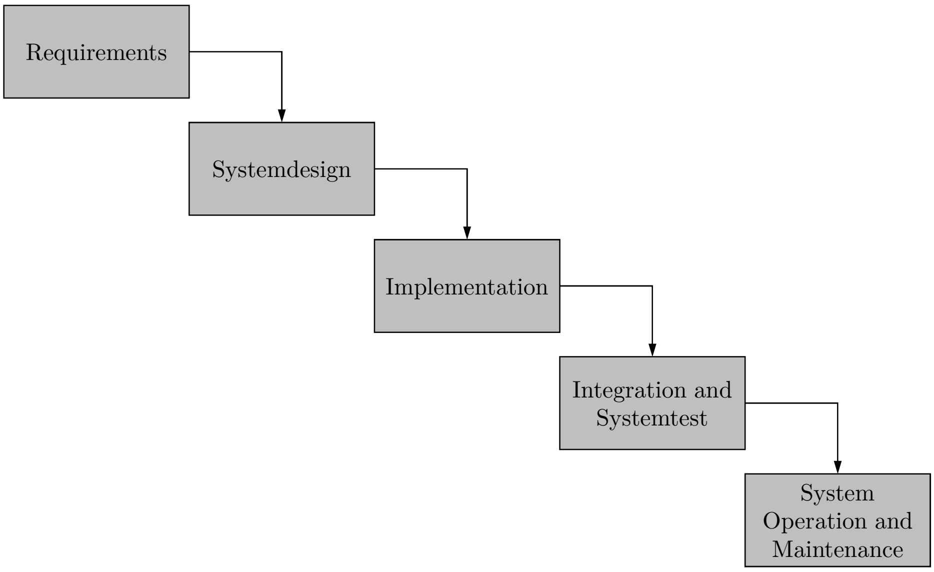

Figure 11.: A simple waterfall model based on Haberfellner et al. (2015)

The waterfall model is applied in projects, where the requirements can be specified in the planning phase very precisely. So in this case it can be a very efficient model. In practice clearly defined phases are often unrealistic and the transitions between the phases are seamless. To avoid late changes in the developed product, that often walk along with high costs, a early freeze of requirements is desired. The consequences of such a procedure are often unsatisfied customers. (Haberfellner et al., 2015)

V-Model

The V-model is a combination of a top-down and bottom-up approach. The development process is also subdivided in phases like in the Waterfall model. The name V-model is derived from the V-shaped representation of the procedural elements. The left side of the *V* represents the top down decomposition of the customer goals to requirements for the system, subsystems and components. The right side represents the integration of the components to subsystems and the overall system. At the same time a verification of the developed subsystems and systems with the requirements on the other side of the *V* takes place. This bottom up iteration secures the solution to meet the users requirements.

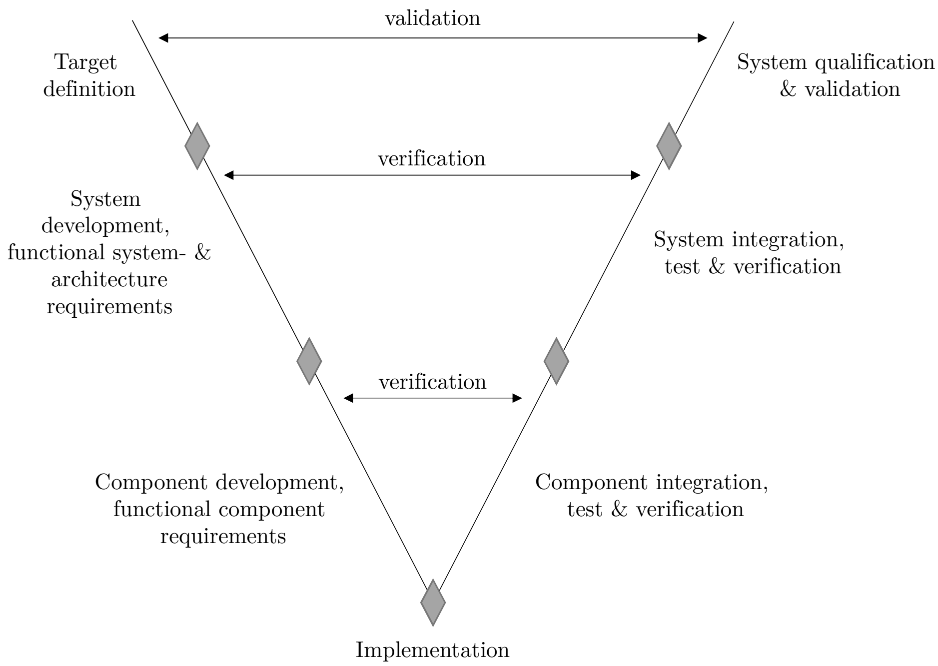

Figure 12.: A V-model and the relationships between requirements and integration based on Haberfellner et al. (2015)

The result of the integration on the right side is always been verified with the requirements on the left side. At the end the overall system is been validated if its meeting the users requirements. (Haberfellner et al., 2015)
According to Witte (2016) the terms Validation and Verification can be defined in the following way:
Verification: Checking the match between a product and its specifications. Am I building the product right?
Validation: Suitability of the product in relation to its intended use. Am I building the right product?

Spiral model

The spiral model, suggested by Boehm (1988), is an evolution of the waterfall model. The principle of the spiral model is the repeated execution of four steps:

- Define goals, solution variants, constraints and restrictions
- Development and evaluation of solution variants, identification and elimination of risks

- Development and validation of the next phase product
- Planning the next phase

In figure 13 the radial dimension shows the total expenses and the angular dimension represents the progress made in each phase or cycle of the spiral.

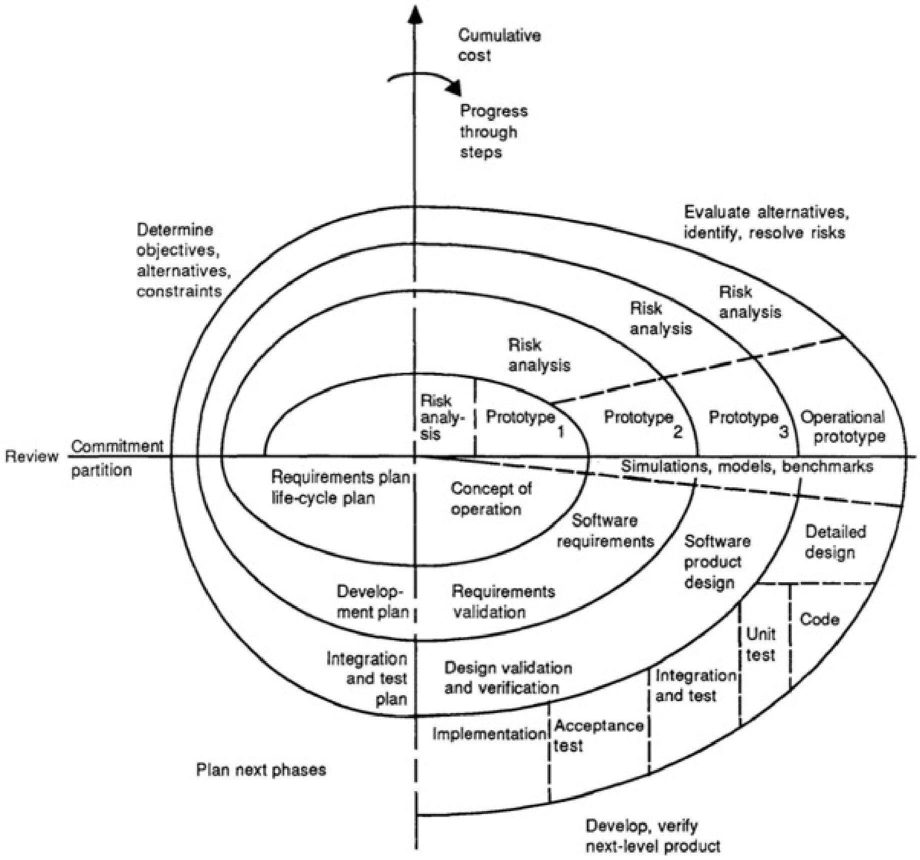

Figure 13.: The spiral model by Boehm (1988)

There is also a phase structure like in the Waterfall Model, but with the prototyping idea the phases are overlapping. The experiences gained in programming the prototypes provide a better basis for the following activities.

Because of the iterative way of proceeding, the system is built step by step. The specification is constantly tested by the developed prototypes. If the prototype is not accepted, the specification is changed or extended.

A major advantage of the spiral model is the early detection of errors. The spiral model is often used for larger projects that are associated with risks. (Haberfellner et al., 2015)

4.1.2.2. Agile methods

The plan-driven methods are accused of making the development process unnecessarily cumbersome, requiring long development times and, as a result, often not being able to satisfy the customer. This need has led to the development of agile methods. The fundamentals of agile methods were proposed in the *Manifesto for agile Software Development* by Beck et al. (2001). In the following paragraphs some agile process models are presented.

Agile manifesto

The agile manifesto is the outcome of a meeting of 17 persons in 2001 and is documented in Beck et al. (2001). The manifesto consists of twelve principles of agile software development:

- *Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*
- *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*
- *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.*
- *Business people and developers must work together daily throughout the project.*
- *Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.*
- *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.*
- *Working software is the primary measure of progress.*
- *Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.*
- *Continuous attention to technical excellence and good design enhances agility.*
- *Simplicity -the art of maximizing the amount of work not done - is essential.*
- *The best architectures, requirements and designs emerge from self-organizing teams.*
- *At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.*

The manifesto was declared, because of a general unsatisfaction with the existing inflexible procedural concepts. The hope in agile methods lies in a faster availability of results and in being able to react better to possible changes in specifications. (Haberfellner et al., 2015)

Kanban

Kanban was originally invented by Taiichi Ohno, Production Manager of Toyota, in the middle of the 20th century. His idea was to manage the material supply with cards and containers. The material in the container is used and the empty container or only a Kanban card is moved back to the source and triggers an order of new material. The supply is pulled by the customer, hence the name pull-principle. This and other techniques led to the *Toyota Production System* and its generalization is well known as *Lean Manufacturing*. (Dickmann, 2014)

At the beginning of the 21st century the ideas of lean production were transfered to the field of software development. According to Epping (2011) Kanban is characterized by four elements:

- *Pull*: Tasks are not pushed to the next phase, but are pulled from the previous one. This element avoids overloading the team and enables self-organized and autonomous work.
- *Limited quantities*: The number of tasks that may be in the value chain at the same time is limited. So the work in progress is reduced which leads to a low lead time.
- *Transparent information*: Transparency should be available for all tasks and every team member should benefit from it. Important information are: phases a task is going through in a value chain, tasks which are in a phase, persons who process a task, limitation of quantities and project key figures
- *Continous Improvement*: rigorously learning from the experience made and deriving sanctions to improve the process.

On a Kanban board with columns for every phase (like requirements analysis, design, implementation, integration, ...) the visualization principle is implemented. Every task run through this phases with the pull principle. Once a task is done in a phase it can be pulled by the following phase. The tasks in a phase are limited, so the quality of work is increased. (Rupp, 2014)

In practice Kanban can be used among others for Requirements Engineering, Development, Quality Management or Project Management (Epping, 2011).

Scrum

According to Haberfellner et al. (2015) Scrum is a collection of structures, roles, working techniques and methods for project management. The main goal is to optimize the development environment, reduce the organizational overhead and meet market requirements as close as possible. The central element of Scrum is the *sprint*, which describes the implementation of an iteration. The duration of an iteration can be 30 days. Before starting a sprint, product requirements are gathered in a *Product Backlog*.
There are three clearly defined roles for the members of a project that should follow the same goals (Pichler, 2008):

- *Product Owner*: represents the end customer needs, controls the software development and works closely with the team throughout the entire project. This role combines product and project management tasks and is at the same time firmly integrated into software development. The Product Owner prioritizes the single Product Backlog elements, from which the team makes a selection for the next sprint.
- *Team*: The team executes all the work necessary to convert the requirements into deliverable product increments. The team is self-organized and has the right to decide how many elements of the product backlog have to be implemented in the next sprint. These are known as commitments.
- *Scrum Master*: The Scrum Master acts as coach and change agent. He helps the team and the company to use Scrum correctly. The ScrumMaster should try to make itself obsolete as soon as possible. He should ensure that the team can work productively.

At the beginning of each sprint, we have a sprint planning session in which the team selects requirements and creates the sprint backlog. The team then starts to implement the requirements. Every day at the same place at the same time there is a short meeting called Daily Scrum. This allows the team to coordinate the upcoming work and identify barriers at an early stage. (Pichler, 2008)

4.2. Problem-Solving process

The Problem-Solving process is a central element of the Systems Engineering concept of Haberfellner et al. (2015). The Problem-Solving process is supported by the SE-philosophy with its two components: *System thinking* and *Process model*. The Problem-Solving process is structured into two main components:

- *System Design*: Dealing with the substantive questions of the problem and its solution
- *Project Management*: organisational measures for planning, controlling, managing and monitoring a project in terms of content, time and costs

4.2.1. System design

In system design, a fundamental distinction is made between architectural design and concept design. Systems architecting is the development of a basic system architecture. The concept design, on the other hand, provides a concrete design of the chosen architecture.

4.2.1.1. Systems architecting

The architecture of a system can be seen as a kind of solution principle that differs fundamentally from other solution principles. Haberfellner et al. (2015) define the term architecture as the *allocation of functions to elements of a structure*. On the other hand Dagli and Kilicay-Ergin (2009) says that architecting is the process of structuring the components of a system, their interrelationships, and their evolution over time and it is related to the structural properties of a system.
System architecture realizes the operational concept of a system. The system architecture orchestrate the system structure, behaviour and performance. It brings form and function of the system together to achieve the system goals and defines the fundamentals for its design. Although the design definition of a system is not the main purpose of the system architecture, the system architecture can influence and in some cases dictate the design. (Raz et al., 2018)
To increase the understanding of systems architecting the example of the Apollo mission in

the 1960ies given by Haberfellner et al. (2015) may help:

- *Direct ascent*: a high-performance rocket should bring the astronauts to the moon directly. The rocket should land there and fly back to the earth after the mission.
- *Earth-Orbit-Rendezvous*: Two modules should be sent to the earth orbit. The assembly should be done in the earth orbit, in order to fly from there to the moon.
- *Lunar-Orbit-Rendezvous*: The spaceship should be sent to the lunar orbit to send a two-part module (the Lunar Lander) from there to the moon. One half of the Lunar Lander remains on the moon, while the other half returns to the spaceship and from there to the earth.

In the systems architecting process the solution space is limited for the first time by a concrete architectural decision. The chosen architecture will be designed more precisely in the following conceptual design phase.

4.2.1.2. Concept design

At the concept design process a chosen architecture is developed more precisely. Further it is also possible to make architecture decisions on subsystem or system element level. The development of a concept should use the logic of the problem-solving cycle with its four components: *Situation analysis*, *Formulating objectives*, *Finding solutions*, *Evaluating and Preparing Decisions*

Situation analysis

Haberfellner et al. (2015) postulates that there are several reasons for a situation analysis. First, an attempt is made to structure the problem or the field of investigation. Further to know the intention of the given tasks and their initial situation. It should create an information basis for the subsequent steps of objective formulation and finding solutions. Complex situations can often not be adequately covered by one-dimensional considerations. Therefore four different approaches for the implementation of situation analysis should be considered:

- *Focus on the system*: Identifying the system and its environment. Building structure models for systems, subsystems, elements and relevant environment. By highlighting different system aspects, different considerations of the same system can be made. (e.g. material flow, information flow or energy flow)

- *Focus on the cause*: Identification of weaknesses and opportunities of the existing system. Once these are known, the root cause can be identified. Furthermore, measures can be discussed which would be able to remedy these causes.
- *Focus on the solution*: Possible solutions are already being considered, but it is not intended to work out concrete solutions already. Rather the intention is to know the state of the art.
- *Focus on time or future*: The future development of the system has to be estimated. The estimation of the future development should serve the better understanding of the problem and its urgency. Furthermore, a basis for the objective and the correct dimensioning of the future solution has to be predicted.

There are different methods to support the situation analysis, which can be divided into the following categories: *information acquisition*, *information processing* and *information presentation*.

Formulating objectives

With the objectives formulation relevant goals, but also non-goals, of a project should be defined. Goals are not obvious and must first be worked out. There are various methods to support the goal finding process.
According to Doran (1981) targets should be *S.M.A.R.T.*:

- *S*pecific: target a specific area for improvement
- *M*easureable: quantify or at least suggest an indicator of progress
- *A*ssignable: specify who will do it
- *R*ealistic: what can realistically be achieved
- *T*ime-related: specify when the results can be achieved.

Important to say is that not all objectives must meet the rules for smart targets. The closer we get to the S.M.A.R.T criteria as a guideline, the smarter our objectives will be.
Furthermore objectives should be prioritized, most of the time it is sufficient to use the MoSCoW method with the following four categories:

- *Must have*
- *Should have*
- *Could have*
- *Won't have*

According to Partsch (2014) and Balzert (2009) further prioritization methods are: *Analytical Hierarchy Process*, *Hundred Dollar Method*, *KANO Model*, *Top-Ten method* or *Simple Ranking*.

Finding solutions - Synthesis/Analysis

The solution search is a creative, constructive, but also a critical part of a project. The aim is to produce different solution variants, which can then be critically examined. Based on the situation analysis and the formulation of objectives, various solution variants have to be developed. An essential characteristic of the methodical approach is that one is not satisfied with the first solution, rather one should try to get a comprehensive overview of possible solution variants.
The synthesis of solutions is the process to create solution variants, at the corresponding phase of the project. These can be drafts, concepts or constructions. Creativity techniques are very important in this phase. There are different strategies for finding solutions, the linear search strategies are: *Routine procedure*, *Non-optimizing search strategy*, *Single-level optimizing search strategy* and *Multi-level optimizing search strategy*. On the other hand there are also *cyclic search strategies*, *methods of operations research* and *heuristics*.
While the synthesis is the constructive-creative part, the analysis should be the analytical and destructive step in a solution finding process. The purpose of the analysis is to test the developed solution variants for their suitability or to find out whether the concept has significant weaknesses. It must be determined whether a solution fulfils formal aspects, such as agreed must-have targets. Furthermore, it is important to check whether relevant topics such as safety, reliability, usability, maintainability have been sufficiently considered. The analysis provides the basis for the subsequent evaluation. The evaluation should not be confounded with the analysis. The analysis serves to check the suitability of each individual solution, while the evaluation represents a systematic comparison of the individual variants. Synthesis and analysis can often not be considered separately, since at the moment a solution idea emerges, critical analysis usually begins immediately. Modelling techniques can also

support the solution finding process, for this reason a model based approach is presented in section 4.3.

Evaluating and Preparing Decisions

Once solution variants have been worked out, they can be subjected to more detailed examination and evaluation. In addition to solution variants, appropriate evaluation criteria also have to be found. There must be the ability to classify the existing variants with regard to the developed criteria. Haberfellner et al. (2015) consider using those objectives that have been prioritized as should have or could have targets as evaluation criteria. However, this list must be completed by further criteria. These criteria result from the knowledge of the developed solution concepts. The assessment of criteria needs situational knowledge and also specialist knowledge of how the criteria is influencing the overall system. Haberfellner et al. (2015) also argue that interim decisions, if the consequences are not big, can be made intuitively. Final decisions should always be supported methodically. Methodical decisions require sufficient information about the possible alternatives. Formal procedures are used to derive a logical decision proposal, which is often supported by mathematical methods. Methodical approaches for decision problems can be evaluation methods or economic calculations. Universally applicable methods are the balance of arguments, the value benefit analysis and cost effectiveness analysis. Hereafter, the value benefit analysis should be presented more in detail.

A value benefit analysis can be used if several useful alternatives are available, but they have different advantages and disadvantages. Often very different criteria must be taken into account, when evaluating these variants. Therefore the challenge is to aggregate the degree of target fulfilment in a single key figure. (Haberfellner et al., 2015)

According to Götze (2014) a value benefit analysis consists of the following steps:

1. Determination of target criteria
2. Weighting of target criteria
3. Calculation of partial value benefits
4. Determination of the overall value benefits
5. Assessment of the advantageousness

For a better understading the criteria for a value benefit analysis are structured and combined into logical groups. Furthermore, the independence of the individual criteria must be guaranteed. Benefit independence is given if the achievement of one objective criterion is possible without the fulfilment of another criterion. In a value benefit analysis, grades are used to express how well a particular sub-target is met by a variant. So therefore every criterion has first to be formulated operational, that means that for every criterion a scale is necessary. For example the space requirement for a new machine is measured in square meters, so the more space will be used, the worse the solution variant. The achievement of the single targets can be measured with ordinal, cardinal or nominal scales. Later these scales are transformed into grades with so called scaling matrices. An example for a scaling matrix is given in table 4. In practice a simple school grades scale from 1 to 5, but also more extensive scales like a 10 points scale can be used.

grading / criterion	0	1	2	3	4	5	6	7	8	9	10	**grading scale**
	very bad		bad		moderate			good		very good		**basic nominal scale**
trust in manufacturer	very low		low		moderate			high		very high		**nominal scale**
design	yuck		well		moderate		favorably		very good		excellent	**nominal scale or**
	6		5		4		3		2		1	**ordinal scale**
space requirement	>26	26	25	24	23	22	20	19-15		14-10		**cardinal scale**

Table 4.: Example of a scaling matrix based on Haberfellner et al. (2015)

In the second step of a VBA a weighting, for capturing the different significance of the criteria, is performed. First the maximum amount of weighting-points should be defined. According to a top-down process the points should be allocated to the superior group criteria in the first place. Later, the points can be allocated to the subordinate criteria. There are several methods to support the weighting of criteria. The *direct interval scaling* takes place by assigning values from an interval scale to the target criteria in such a way that their distances reflect the preference differences of the decision maker. (Götze, 2014)
In an *indirect interval scaling*, the target criteria are ranked first. Each criterion is ranked according to its position in the ranking order. The most important criterion is assigned the value C (C = number of criteria) and the least important criterion gets the value one assigned. Then the ranked numbers are transformed into weights, which are measured with an interval scale. This substep requires an assumption regarding the preference differences between the rankings. If it can be assumed that the same preference difference exists between consecutive rankings, the ranking numbers can be used as weights. (Götze, 2014)
Another method for weighting the criteria is the *pairwise comparison*. The idea behind it is to compare each criterion with every other. For a better overview, the pairwise comparison can

also be carried out in groups according to the criteria groups. At the beginning, all criteria to be weighted are listed in a table (5).

than / more important	A	B	C	D	E	sum	relative
A		2	0	1	0	3	19%
B	0		2	2	2	6	38%
C	2	0		2	2	6	38%
D	1	0	0		0	1	6%
E	2	0	0	2		4	25%

Table 5.: Example of a table for pairwise comparison (Authors own illustration based on Kühnapfel (2014))

Now it is decided one after the other whether criterion A is more important than criterion B, criterion A is more important than C, and so on. If a criterion is more important than the other, a *2* is filled in. If they are of the same importance a *1* is used. Of course, the counter-question does not have to be asked, so the lower part of the table results from the upper part. Comparing a criterion with itself is useless and therefore these fields are not filled in. If all fields have been filled in, the total for each criterion is calculated line by line. The relative weight can subsequently be calculated from this. (Kühnapfel, 2014)

The next step in a *VBA* is the value benefit calculation of each alternative. Therefore the weights are multiplied with the criterion points. For multiplication, both the group weights and the individual criteria weights must be used.

				Variant A		Variant B		Variant C	
criteria		**weighting**		**rating**	**VB**	**rating**	**VB**	**rating**	**VB**
		group	criterion	n	w1*w2*n	n	w1*w2*n	n	w1*w2*n
Group 1		40%							
	criterion A		20%	7	0,56	10	0,80	3	0,24
	criterion B		30%	9	1,08	10	1,20	5	0,60
	criterion C		10%	5	0,20	4	0,16	1	0,04
	criterion D		40%	4	0,64	2	0,32	1	0,16
	subtotal 1				**2,48**		**2,48**		**1,04**
Group 2		60%							
	criterion E		15%	1	0,09	10	0,90	1	0,09
	criterion F		35%	10	2,10	10	2,10	10	2,10
	criterion G		10%	10	0,60	10	0,60	10	0,60
	criterion H		5%	10	0,30	7	0,21	10	0,30
	criterion I		25%	6	0,90	10	1,50	10	1,50
	criterion J		10%	10	0,60	10	0,60	1	0,06
	subtotal 2				**4,59**		**5,91**		**4,65**
	Total				**7,07**		**8,39**		**5,69**

Table 6.: Example of a Value Benefit Analysis based on Haberfellner et al. (2015)

Then the results for each group are added up and the partial value benefit for each group is obtained. Adding each partial value benefit up gives the overall benefit value for each alternative. The alternative with the highest value is to be preferred. This process is shown in table 6. (Kühnapfel, 2014) and (Haberfellner et al., 2015)

4.3. Systems Modeling Language - SysML

The Systems Modeling Language is a derivative of UML and it is used in Systems Engineering. It is a graphical modelling language and independent from methods and tools. It can be used for specification, analyis, draft, verification, validation of systems. SysML was initiated by the International Council on Systems Engineering (INCOSE). SysML is based on the four "pillars" system requirements, system structure, system behavior, parametric relationships and offers corresponding diagram types for these four aspects. (Partsch, 2014)
An overview of the diagrams and the assignment to the aspects mentioned above can be seen in figure 14.

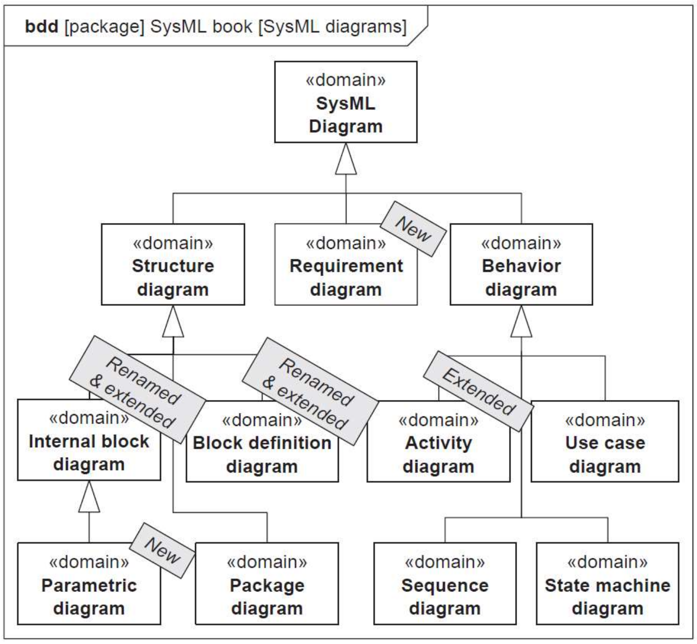

Figure 14.: SysML diagram overview (Friedenthal et al., 2015)

Sequence, state, use case and package diagrams were adopted unmodified from UML.

The activity diagram has been extended to cover system-relevant aspects. A new addition to SysML is the Parametric Diagram for displaying relationships between properties of different system components. Another new feature is the requirements diagram for displaying requirements and their interrelationships. Each SysML diagram has a rectangular frame, which is obligatory here and not optional as in UML. This frame contains a unique diagram type identifier (e.g. bdd, ibd, act), type and name of the displayed model element and possibly further information about the diagram. (Partsch, 2014)

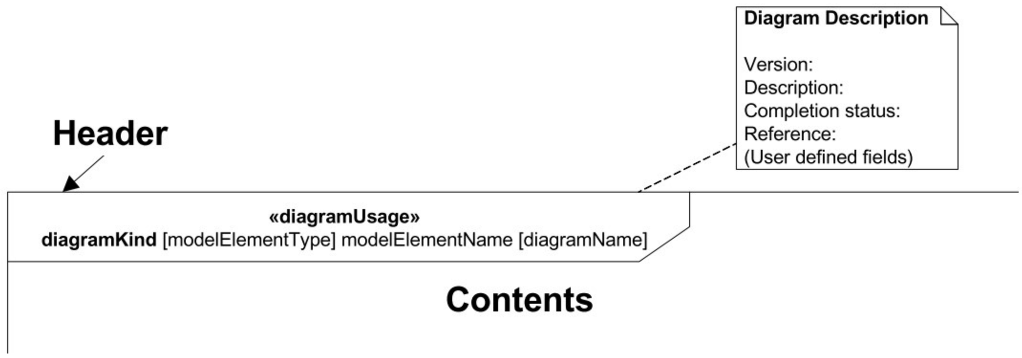

Figure 15.: SysML diagram frame (SysML, 2017)

In the header the context of a diagram is indicated by the following attributes (Friedenthal et al., 2015):

- *diagram kind*: Depending on the type of the diagram, the diagram kind can be one of the following:
  - Activity Diagram - **act**
  - Block Definition Diagram - **bdd**
  - Internal Block Diagram - **ibd**
  - Sequence Diagram - **sd**
  - Package Diagram - **pkg**
  - Requirement diagram - **req**
  - State machine diagram - **stm**
  - Use Case Diagram - **uc**
- *model element type*: Representing the type of model element in the diagram:
  - Activity Diagram - activity
  - Block definition diagram - block, constraint block, package, model, model library

  - Internal Block diagram - block
  - Package diagram - package, model, model library, profile, view
  - Parametric diagram - activity, block, constraint block
  - Requirement diagram - package, model, model library, requirement
  - Sequence diagram - interaction
  - State machine diagram - state machine
  - Use case diagram - package, model, model library
- *model element name*: This is the name of the represented model element, chosen by the modeler.
- *diagram name*: The diagram name is also user defined and is used to provide a description of the diagrams purpose.
- *diagram usage*: This is a keyword to indicate a specialized use of a diagram.

4.3.1. Diagram types

In the following section the nine different diagram types are presented and a small example for each one is shown.

Package Diagram

Many models quickly reach a number of elements that require a structured organization. For this reason, package diagrams are used to arrange the individual model elements. The criteria for grouping are not explicitly specified, this is the modeler's responsibility. (Weilkiens, 2006)
According to Friedenthal et al. (2009) the model can be organized in multiple ways:

- by system hierarchy (e.g. enterprise, system, component)
- by diagram type (e.g. requirements, use cases, behaviour)
- by the use of viewpoints

In figure 16 an example of a package diagram is shown. Remarkable is that it contains several levels of the package hierarchy.

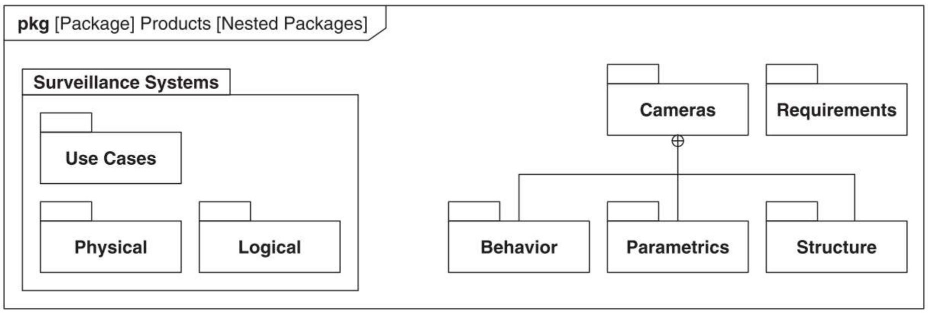

Figure 16.: Example of a Package Diagram (Friedenthal et al., 2015)

Block definition diagram:

According to the publication of the OMG SysML (2017) a block definition diagram consists of blocks and their relationships to each other. A block is a modular unit of a system description and is used to define a collection of features that describes a system. This features can be divided into structural and behavioral features. Structural features define the internal structure and properties, while behavioral features define how the block interacts with the environment (Friedenthal et al., 2015).

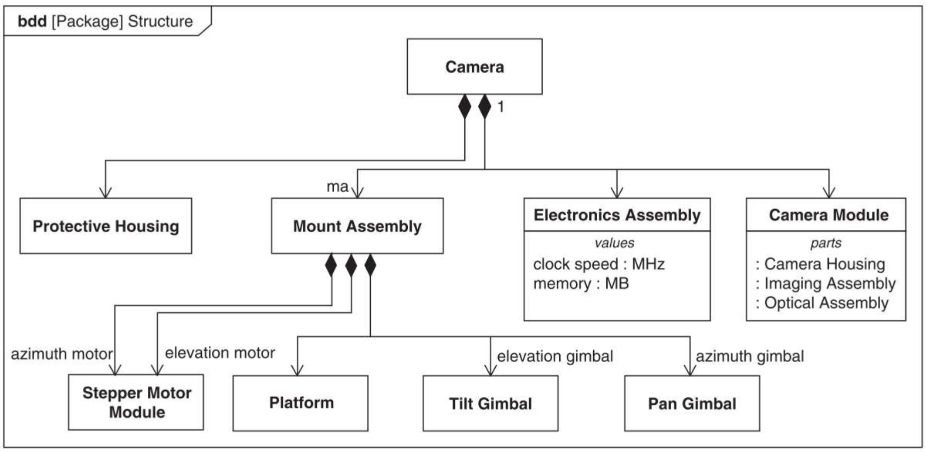

Figure 17.: Example of a Block Definition Diagram (Friedenthal et al., 2015)

Structural features can be divided in three categories: part properties, reference properties and value properties (Friedenthal et al., 2009).
On the other hand there are four behavioral feature categories: ports, operations, constraints and allocations (Partsch, 2014).

Internal Block diagram:

A internal block diagram represents interconnection and interfaces between the parts of a block. The internal block diagram, which corresponds to the component structure diagram of UML, shows the blocks of the block, their connection points (ports) and the connection lines between the connection points.

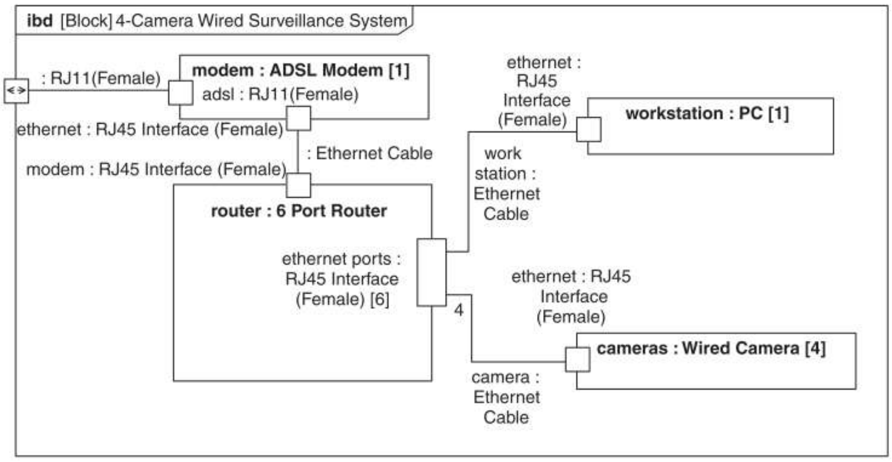

Figure 18.: Example of an Internal Block Diagram (Friedenthal et al., 2015)

There are "standard ports" and "flow ports". Standard ports are the same ports as in UML and have standardized interfaces. Flow ports are an extension of SysML for displaying physical and/or continuous flows. (Partsch, 2014)

Parametric Diagram:

The intention of a parametric diagram is to create systems of equations to analyse the system by defining constraint blocks (Finance, 2010). In figure 19 an example of a parametric diagram is shown.

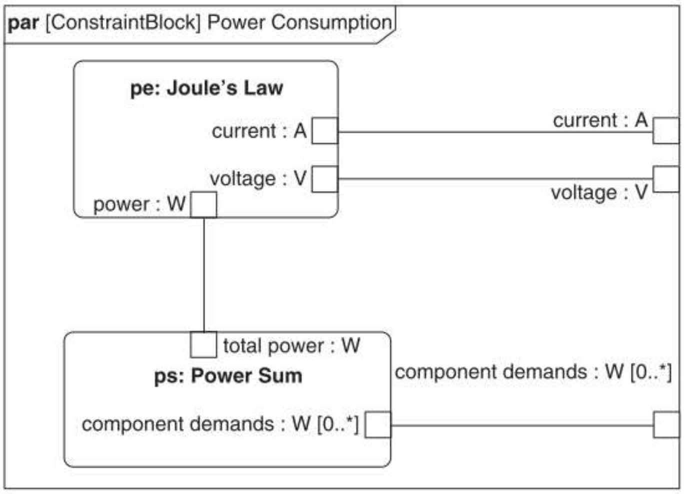

Figure 19.: Example of a Parametric Diagram (Friedenthal et al., 2015)

Weilkiens (2006) gives as an example Newton's Law (F = m * a). The overall mass results as the sum of the masses of the single blocks, while the acceleration is calculated from dynamical properties of the system. A constraint can be that the force on the overall system has to be under a certain limit. So the relation between properties of different blocks can be described and defined in a parametric diagram.

Requirement Diagram

What Requirements are, is described in detail in chapter 3. For a better organization of requirements, the requirement diagram can be used. The SysML notation provides a bridge between text based requirements and the system model (Friedenthal et al., 2015). A requirement is an element in SysML that describes the properties or behavior of a system. A graphical or tabular notation is provided for the representation of requirements (see figure 20). It is also useful to present hierarchies of specifications in a graphical way. (Partsch, 2014)
There are new types of associations in a requirements diagram (Finance, 2010):

- *Derive:* one or more requirements are derived from a requirement
- *Satisfy:* one or more model elements fulfil a requirement.
- *Verify:* one or more model elements verify that the system fulfils a requirement (e.g. a test case)

- *Refine:* one or more model elements refine a requirement further (e.g. a use case)

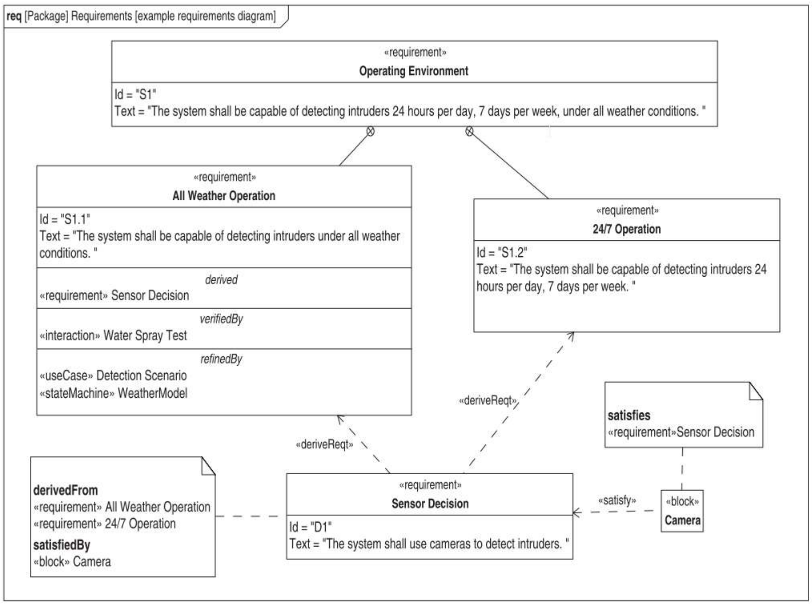

Figure 20.: Example of a Requirement Diagram (Friedenthal et al., 2015)

Activity Diagram

An activity describes behavior that specifies the transformation of inputs to outputs through a controlled sequence of actions. In an activity diagram the sequence of actions along the flow are described. (Friedenthal et al., 2015)

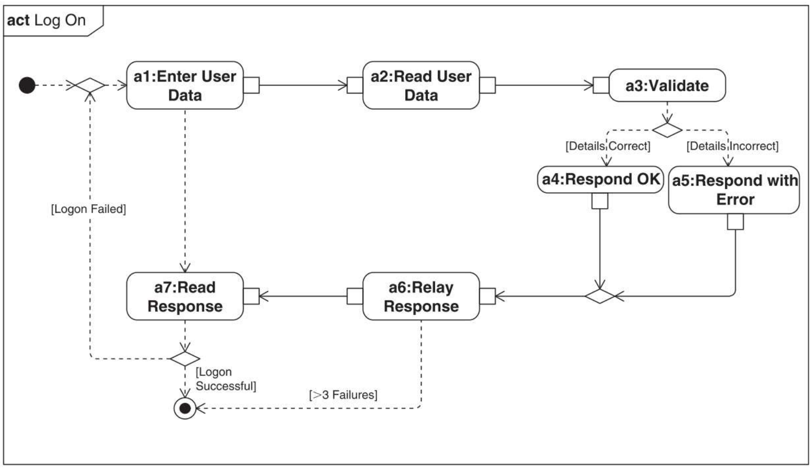

Figure 21.: Example of an Activity Diagram (Friedenthal et al., 2015)

Flows can run parallel, synchronized or split based on conditions. There are several extensions to the activity diagram in UML (Weilkiens, 2006).
In figure 21 an Activity Diagram with some of the basic Activity Diagram symbols for a simple log-on process is shown.

Sequence Diagram

Sequence diagrams are mainly used to specify scenarios. They describe the interaction between communication partners in their chronological order (Partsch, 2014). A sequence diagram represents behavior in terms of a sequence of messages exchanged between systems or between parts of systems (Friedenthal et al., 2015).
Elements in a sequence diagram are represented by a lifetime. These lifetimes can be generic instances or instances of blocks defined in the model. All sequence diagram definitions used in UML also apply to SysML. (Finance, 2010)
In figure 22 an example of a Sequence Diagram is shown. It shows the sequence of messages between an advanced operator and the surveillance system.

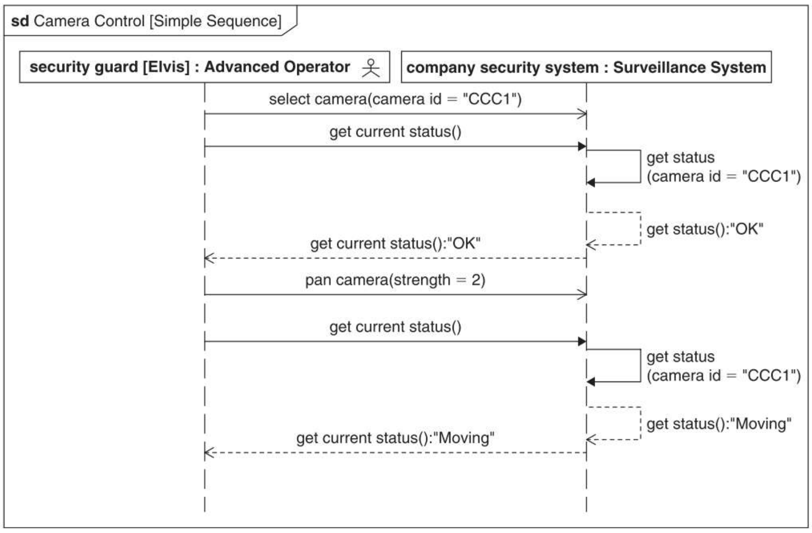

Figure 22.: Example of a Sequence Diagram (Friedenthal et al., 2015)

State Machine Diagram

A State Machine Diagram is used to represent the life cycle of a block and to represent the behavior of an entity in terms of its transitions between states triggered by events (Friedenthal et al., 2009). They are assigned to individual use cases, operations or classes and show the dynamic behavior of the respective instances (Partsch, 2014).

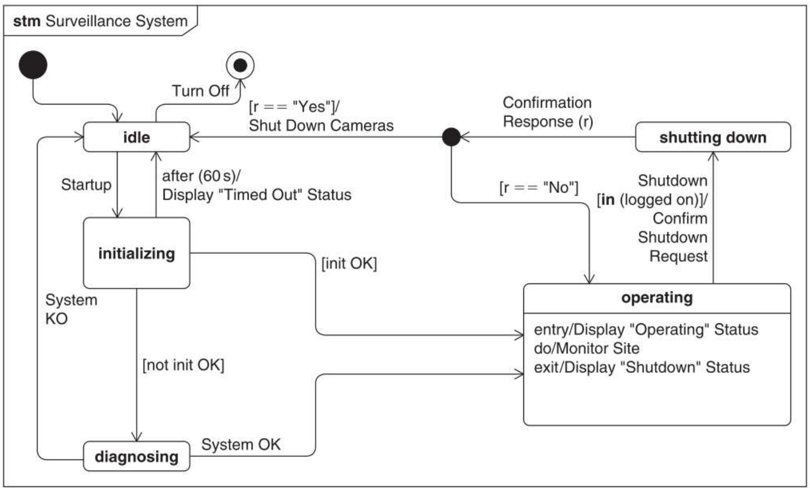

Figure 23.: Example of a State Machine Diagram (Friedenthal et al., 2015)

A state can have an entry-behavior, exit-behavior and a do-behavior (Friedenthal et al., 2009). Weilkiens (2006) gives the following definitions for the terms *state* and *transition*:
A *state* represents a set of value combinations. It has a name and can have an internal behavior that is executed on the basis of defined events.
A *transition* is the direct relationship between two states and defines a trigger and a condition that lead to the state transition.
In figure 23 an example for a simple state machine diagram is shown.

Use Case Diagram

An Use Case Diagram represents the functionality of a system in terms of how a system is used by external entities. Those external entities can be users of a system or other external systems as well. (Friedenthal et al., 2015)
Use Case diagrams provide an overview of the different use cases and the actors which they are related to. The use cases themselves can be described by text or suitable other diagrams(e.g. Sequence diagrams) in more detail. Use cases can be described in different levels of detail. In the case of requirements, details with regard to the implementation of a use case should be avoided and so-called "essential" applications should be preferred instead. (Partsch, 2014)

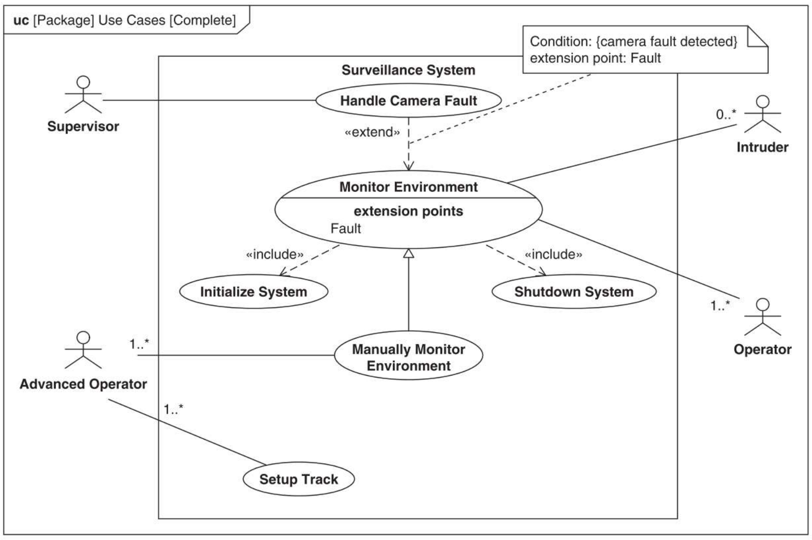

Figure 24.: Example of an Use Case Diagram (Friedenthal et al., 2015)

In figure 24 an example of an Use Case Diagram is shown. The case of extending a use case by an extension point is also shown here. An extension point is a uniquely identifiable point in a use case where action sequences of other use cases can be inserted. This is expressed in the use case diagram by a corresponding relationship with the stereotype "extend". In addition, there is "include", which represents a usage relationship between use cases. (Partsch, 2014)

5. Case Study: Production Data Acquisition in Discrete Manufacturing

In the following chapter the findings of the literature research are evaluated for their adequacy. Therefore the design and implementation of a Production Data Acquisition System at a partner company is described.
First, the partner company and its history are briefly introduced.
The case study then starts with the selection of an appropriate process model. The overall project goals and expectations were then roughly defined. The next step was to analyse the situation found at the company at the beginning of the study. From the problems found, use cases were detected and requirements for a production data acquisition system were derived. Subsequently suitable tools were searched and evaluated following a structured approach. The outcome of the evaluation then was included in the concept implementation process.
In accordance with the goal of examining the introduction of a production data acquisition system, a tailored concept was designed and verified by a Proof-of-Concept. After running the Proof-of-Concept for one month improvements and lessons learned can be given.

5.1. Partner company - WP Performance Systems

The company WP Performance Systems was originally founded as WP Suspension in 1977 in the Netherlands. The name WP is derived from the initials of its founder Wim Peters, a dutch motocross professional. Stefan Pierer, CEO of the KTM Group, always wanted to

have a manufacturer of suspension systems for his KTM motorcycles exclusively. So he acquired the company in the end of the 90s, but remained its plant in the netherlands until 2007. Emerging quality problems and centralization of the supply chain prompted him to move the company closer to the headquarter of KTM in Mattighofen, Upper Austria.
Due to the strategy focus of WP as tier-1 supplier of KTM, the production portfolio of WP was extended with water and oil coolers by the acquisition of KTM Kühler in 2010.
To bundle the production of the main motorcycle components in one place, the takeover of the frame and exhaust production took place in 2012. Today the company has two manufacturing locations: Munderfing in Austria and Dahlian in China. The production of cooling systems will be moved to China shortly.
In 2016 WP Performance Systems achieved a turnover of 166,1 million Euros with 676 people (621 in Austria) employed.

5.2. Process Modell

From the multitude of process models (cp. with section 4.1.2), a suitable one had to be selected for this project.

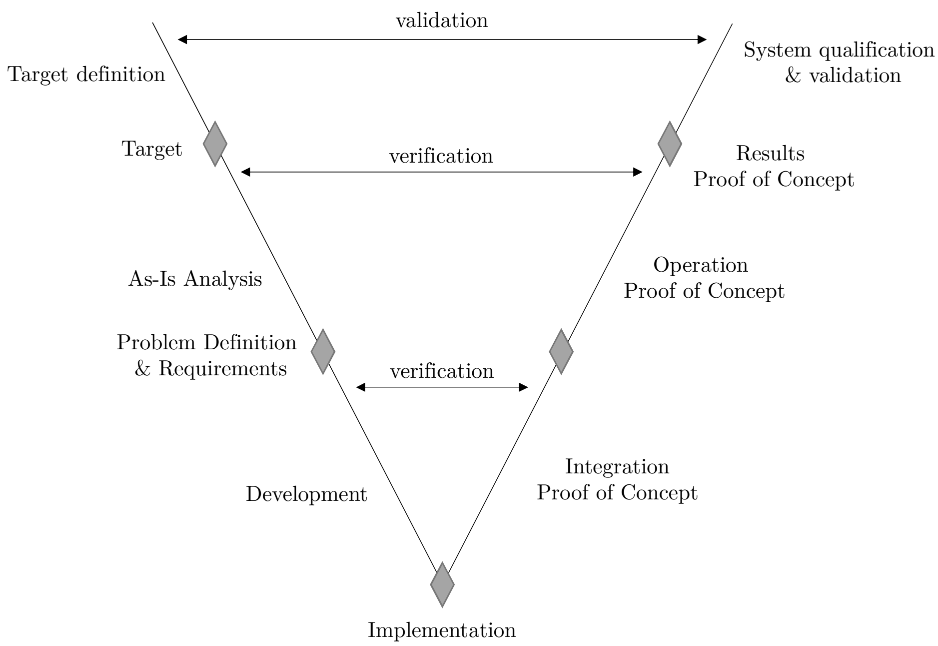

Figure 25.: The V-Model used with this project (authors own illustration)

The V-model in figure 25 was chosen, because it is a generally understandable and comprehensive model.

5.3. Project Goals

The aim of this project is to find a suitable system for production data acquisition. First, the requirements for such a system must be determined. Furthermore, an analysis and evaluation of several software systems for production data acquisition have to be done. After selecting one feasible system, the suitability of the product is to be tested with the help of a Proof-of-Concept. The general expectations before the start of the project for such a system were:

- More transparency in production
- Possibilities for efficiency analysis and optimization
- Acquisition of machine and production data
- Traceability and alarming of downtimes
- Calculation and monitoring of key performance indicators

5.4. As-Is Analysis

The following section covers the situation found at the beginning of the project at WP Performance Systems. It provides an overview of the existing manufacturing processes at WP Performance Systems. Also the pains in daily business according current data acquisition in manufacturing is shown. An analysis of the existing IT-Systems used in production is given. As a result of this analysis an understanding of WP's needs and the given constraints were established.

5.4.1. Production System

The company consists of four different production units, called divisions, which are literally and locally separated. Each division has its own warehouse for incoming as well as for outgoing goods. The four divisions are:

Suspension

In the division suspension the forks (fig. 26) and shock absorbers (fig. 27) are assembled. There is a low level of vertical integration, since all components for assembling are delivered by suppliers. Production can be divided into two main parts, pre-assembly and main-assembly. In the pre-assembly section, which is not directly linked to main assembly, parts are prepared for main-assembly. Typically manufacturing steps are press in of bushes or insert o-rings in grooves, but also the assembly of the so called "setting" is done there. The setting is part of a fork as well as of a shock absorber which defines the dampness of the suspension. Altogether pre-assembly is composed of manual workstations, assisted by simple machines like presses or manually operated screw units. Pre-Assembly is done in a workshop-like production.

Models:	AER 35-52 XPlore 48 Closed Cartridge, Cone Valve Semiactive Fork
Techn. Data:	35mm, 40mm, 43mm, 48mm, 52mm ø innertube
Application:	Motorcycles: Onroad, Offroad, Racing
Output:	140.000 forks per year

Figure 26.: Front Fork (Source: company presentation)

The final products are manufactured in four different assembly lines, two for front forks and two for shock absorbers.

Three out of four assembly lines are moved from the Netherlands to Austria in 2007 and accordingly not state of the art. The assembly line for the offroad forks has been newly developed and consists of four stations and one test bench. The benefit of the new assembly concept is, that the manual work done by the worker and the work done by the machine happens simultaneously.

Models:	Emulsion and De-Carbon type Link Xplore PDS, Trax ESA® EDS® Semiactive Shock
Techn. Data:	Piston: 36, 40, 46, 50mm ø Piston rod: 14, 18mm ø
Application:	motorcycles: onroad, offroad, racing Powersport: ATV, Snowmobiles, Sportcars
Output:	200.000 shocks per year

Figure 27.: Shock Absorber (Source: company presentation)

The assembly stations are semi-automated. So the workers function is to join the parts together, load and unload the machine, where the screw-work is done and pass the intermediate good to the next station. Between the stations is a buffer, where "good" parts and also "bad" parts are stored. Bad parts are moved to the central rework place. At the test bench every fork is tested with a static and also dynamic test for their damping qualities. Test bench data is processed automatically and signals the worker visually if the product passed the test. After test bench the product is being stored for 24h to indicate possible leaks. After this step the product is visually checked by another worker and ready for shipping to the customer. The core process of suspension production on a high level is shown in figure 28. The schematic production layout is shown in figure 29.
The conventional lines are barely automated, so the need of personnel is higher to get the same output as of the semi-automated line. A fully automated assembly line wouldn't be profitable, since there is a large variety of models and products change in different degrees at least every two years.
Every product is marked with a data matrix code, which contains relevant information of the

product, like order number, supplier id, week of production, year of production and serial number.

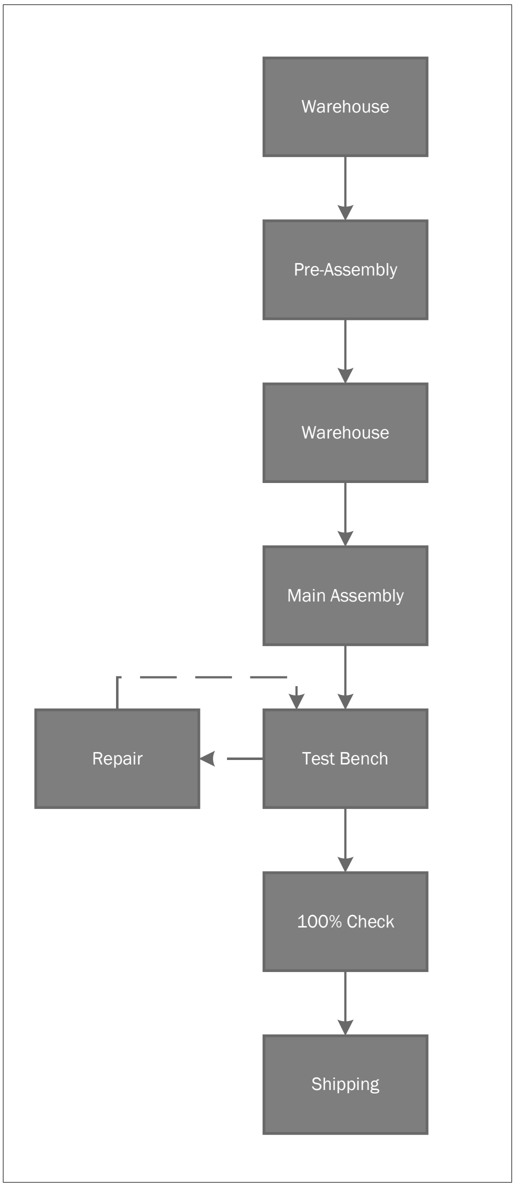

Figure 28.: High Level View of production process in Suspension (Source: authors own illustration)

Main customer of WP suspension systems is the parent company KTM AG and accounts for more than 90% of the turnover of this division. In the past WP was supplying other motorcycle manufacturer like BMW or Triumph as a Original Equipment Manufacturer. In 2016 WP Suspension accounted for more than 50% of the total turnover of the company.

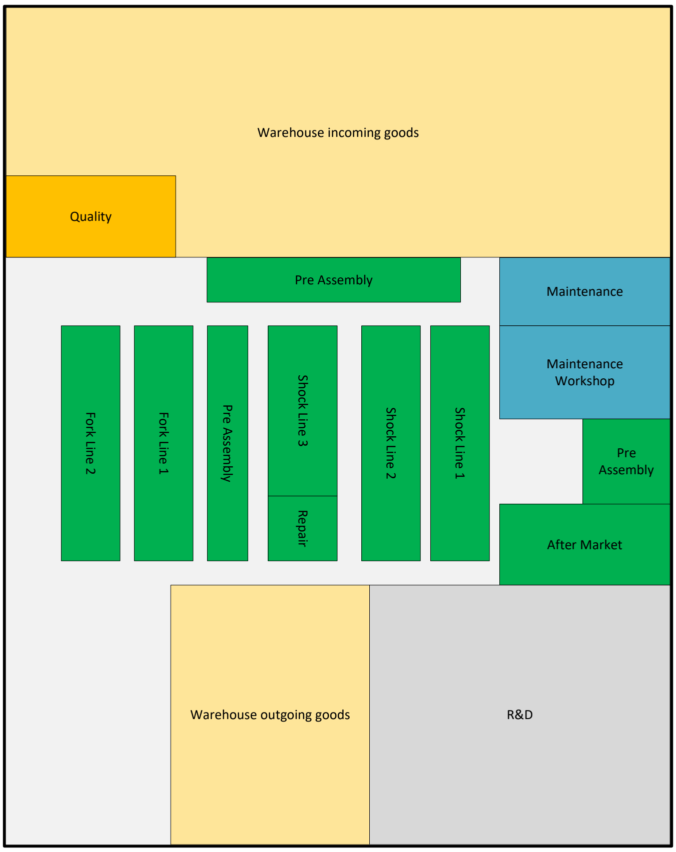

Figure 29.: Layout Suspension (Source: authors own illustration)

Radiator

In the radiator division nearly 400.000 cooling systems are made per year, which are divided in two main groups of products: water coolers (fig. 30) and oil coolers(fig. 31) . The vertical integration in this divison is very high, since the fins for the coolers are punched and bent from coils in house. Other needed parts are bought in addition from external suppliers or are made in the punching shop of the exhaust division respectively. Components are put together in a rack to avoid falling apart before joining them via soldering. Before soldering in a continuous furnace, intermediate goods are covered with a soldering flux to support the soldering process. After the oven the rack is dismounted and the cooler is moved to final assembly. Final assembly is divided in three assembly lines for A-, B- and C-coolers. Classification was done by annual quantity. At the final assembly the cooler is inspected for leakage and attachments like lugs, the honk, the water tank and cooling fans are mounted.

At the end of the assembly line all coolers are serialized with a label affixed on the cooler. Before shipment all coolers are inspected visually by a worker.

Models:	Aluminium CAB-brazed radiators
Techn. Data:	blockthickness: 23, 32mm finheight: 5, 8 and 10mm Blocklength till 900 mm
Application:	Onroad, Offroad, Racing Powersport: ATV, Kart, Sportcars, Automotive
Output	300.000 per year

Figure 30.: Water Cooler (Source: company presentation)

All production steps are done manually with assistance of machinery like presses or manually operated screw drivers.

Models:	Aluminium CAB-brazed oilcoolers tubes or shell type
Techn. Data:	blockthickness: 30 till 63 mm
Application:	Motorcycle: Onroad, Offroad, Racing Powersport: ATV Sportcars Automotive
Output:	100.000 per year

Figure 31.: Oil Cooler (Source: company presentation)

Customers of WP cooling systems are motorcycle manufacturers like KTM and BMW, but also from the automotive sector with Maserati or Bentley. In 2016 WP radiator accounts for roughly 10% of WP Performance Systems total turnover.

Frame

In the division frame the frames (fig. 32) for KTM motorcycles are produced. Production is organized in a workshop production, with slight signs of line manufacturing. In the production hall are 29 robot cells with just as much robots. Welding is done automatically by the robots. At each cell is also a place where rework is done. The worker is responsible for inserting the single parts in the welding equipment, but also for the rework to be done.

Models:	single cradle double cradle trellis frames subframes
Techn. Data:	Frame assemblies made of steel tubing
Application:	Motorcycles: Onroad, Offroad, Racing, ATV ‘s
Output	150.000 per year

Figure 32.: Frame (Source: company presentation)

In general there are two types of frames, those for the offroad motorcycles and those for the street. Here the design of the frames is fundamentally different. The frames are mostly built up from steel tubes, which are delivered with their ends cut on a 3D-Laser, so they fit together easily. Bending, where necessary, is done on two bending machines in house. Parts for welding are provided order per order at every robot-cell. After each welding process semi-finished goods are stored at a buffer in the production place. After final welding the frames are moved to a place for final check-up. If necessary frames are straightened and reworked.

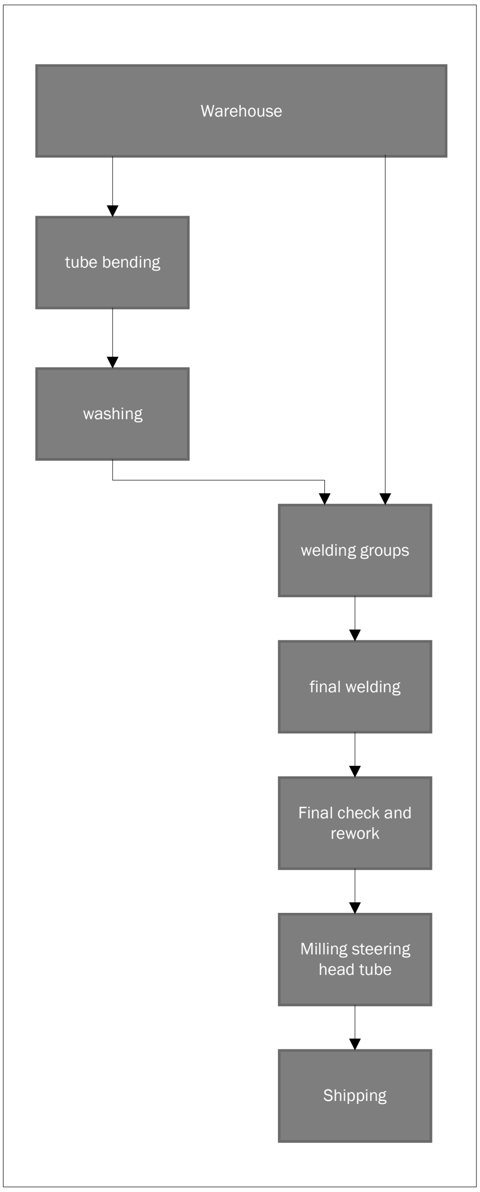

Figure 33.: High Level View of the frame production process (Source: authors own illustration)

Next step is the drill-finishing of the steering head tube on one of the two machine centers. All frames are coated at a external partner and delivered to KTM afterwards. The layout of the frame production is shown in figure 34. A high level view on manufacturing process can be seen in fig 33. The frame production accounts for one fifth of the companies total turnover.

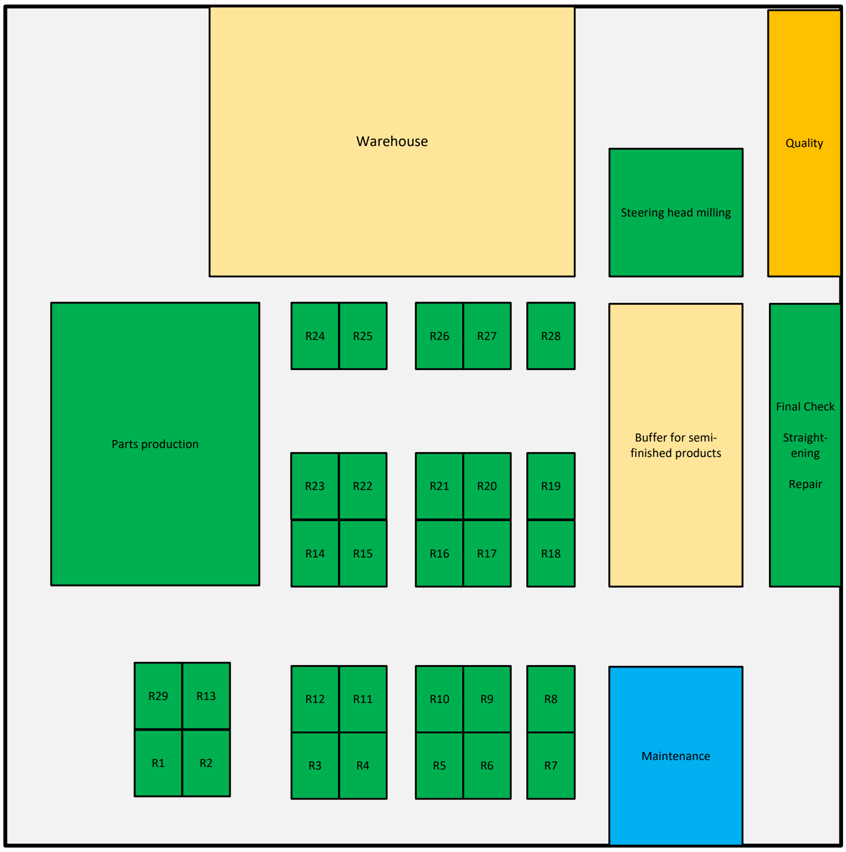

Figure 34.: Layout Frame (Source: authors own illustration)

Exhaust

In this division the exhaust systems (fig. 35) for KTM motorcycles are produced. Vertical integration is very high, because production process starts with metal sheets being cut by a flatbed laser. Also steel tubes are cut to correct length by a parting-off automatic machine. This steel tubes are bent to correct shape with bending machines and the ends are drifted to correct diameter by a machine. End caps are formed by a multistage deep drawing process. Also the half shells of the 2-stroke manifolds are made by a deep drawing process. Headers are welded manually in a workshop like production organization. But there are also robots for circular welding of the tubes and welding of the headers. The exhaust main silencers are assembled manually.The exhaust production accounts for more than 15% of the companies total turnover.

Models:	2-stroke and 4-stroke headers and silencers
Techn. Data:	stainless steel, titanium and aluminium (silencers only)
Application:	Motorcycle: Onroad, Offroad, Racing, Sportcars
Output:	> 150.000 per year

Figure 35.: Exhaust system (Source: company presentation)

5.4.2. Problem Definition

The motorcycle market is experiencing an economic boom and thus KTM and therefore also WP is expecting an increase in sales of more than 100%. In order to meet this challenge, possibilities for optimizing the existing production system are being sought. Today there is no system which automatically acquires production data from the production lines. Consequentially there is no relevant data of uptimes of the companies machinery. Furthermore there is no tracking of the production processes efficiency and as well there is no exact or rather online information of production capacity. In order to be able to further increase the efficiency of the production processes, the introduction of a Production Data Acquisition System is to be examined. The project was initiated by the department of Industrial Engineering at WP Performance Systems. The statements made in the following sections are the outcome of several meetings and interviews with the stakeholders at WP Performance Systems on the one hand, but also the observation of the production processes has been incorporated in the situation analysis on the other hand. The problems or pains found with current data acquisition in production can be classified into the following four main categories *information transparency*, *currentness*, *manual work* and *flexibility* which can be described in the following way:

Information transparency:

Today there is no system which automatically acquires production data from production lines. The yield of each production line is documented in a shift journal manually by the worker. It is obvious that there are problems with the consistency of the data in the shift journal, the data reported to the ERP system and the actual quantity produced.
On the other hand there is no automatically gathered data for the availability of the companies machinery. Machine breakdowns are documented manually by the maintenance worker in online forms on a Microsoft Sharepoint server. This data contains also information about the costs and duration to rectify the fault. There are quarterly reports from the data of the breakdown list. Data is accumulated machine by machine by the duration for fixing the problem as well as by the costs for overhauling the machine. By tracking costs and duration with this system it needs high self-responsibility of the maintenance worker to fill in the data in the list correctly. If data is not filled in, there is a high chance the specific breakdown will never be documented.
According to the companies requirement of increasing the production capacity until 2020 by 100% there is a high need of tracking the production processes efficiency to detect inefficiency and improvements can be done.
The lead time of the orders is roughly calculated from the starting point of an order in the incoming-goods warehouse and the ending point of an order when its finished and the products are booked to the outgoing-goods warehouse.
Currently there is no exact or rather online information of production capacity. So there is no exact information if the production line working to capacity or if there is idle capacity for more orders.
This would be also an important information for future investments. If an equipment is occupied at a high rate, it could be necessary to buy a duplicate. This figures are roughly calculated within the ERP-system from the lead times. As well cycle times are not recorded, so there is no validation of the planned cycle times which are collected by the time study team.

Manual Work:
Quantities are recorded either by the foremen or by the worker itself in written form for every shift. This data is passed to the production manager on a daily basis. An example for a shift journal is shown in figure 36.

Produktionsmeldung 3 FBL1

Vorarbeiter: Wieshofer Anton

Datum: 29.03.2017

Uhrzeit	MA	Modell	Bezeichnung	eff. Min	SOLL Linie bei x MA	SOLL Linie bei 9 MA	Produzierte Menge	Produktivität Linie bei x MA	Produktivität Linie bei 9 MA	Name Prüfer	Störung	Minuten	Bemerkung
05:45 - 07:00	8	12187R04	FB.SX 85	75	73	82	71	97,26%	86,59%	ZORAN			
07:00 - 08:00	8	12187R04	FB.SX 85	60	58	66	57	98,28%	86,36%	ZORAN			
08:00 - 09:00	8	12187R04	FB.SX 85	53	58	58	55	94,83%	94,83%	ZORAN			
09:00 - 10:00	8	12187R04	FB.SX 85	55	54	60	52	96,30%	86,67%	ZORAN			
10:00 - 11:00	8	12187R04	FB.SX 85	60	58	66	57	98,28%	86,36%	ZORAN			
11:00 - 12:00	8	12187R04	FB.SX 85	27	27	30	25	92,59%	83,33%	ZORAN			
12:00 - 13:00	8	12187R04	FB.SX 85	60	58	66	57	98,28%	86,36%	ZORAN			
13:00 - 14:30	8	12187R04	FB.SX 85	83	81	91	79	97,53%	86,81%	ZORAN			
			Summe ohne Überstunden	473	467	519	453	97,00%	87,28%			0	
14:30 - 15:30				0	0	0		0,00%	0,00%	ZORAN			
15:30 - 16:30				0	0	0		0,00%	0,00%	ZORAN			
			SUMME mit Überstunden	473	467	519	453	97,00%	87,28%			0	

Figure 36.: Shift Journal for one shift from the Shock Line (source: authors own representation)

Quantities are also tracked by bookings in the ERP system. Since there is no exact time recording for the actual order, the bookings are not very informative in the context of performance.

At the semi-automated machines, quality is recorded by the worker in a tally sheet. If the tightening process failed, the product has to be tightened another time. This failures are recorded in a tally sheet and these sheets are passed to the person in charge of quality control sporadically.

As described before, maintenance work is documented in forms on a Sharepoint system. The maintainer have to fill in where the problem occurred, what they did, how much time they needed and how much the repair work cost.

Currentness:

Since most of the information from the shopfloor is aggregated by manual processes, there are problems with the actuality of recorded data.

Today there is no system which provides you information which order is currently processed on which assembly line. Subsequently there is no live data of how the latest order is performing.

Furthermore there is no information about an estimated finish time of an order.

To respond to emerging problems on the shopfloor, it is important to have current data of the production processes. The current manual production data acquisition lags always behind.

Flexibility:

When a machine standstill occurs, first of all the worker itself tries to fix the problem. If he is not able to remedy the problem he has to find the foreman in charge for this production line. Once he found the foreman, he tries to eliminate the issue, if not successful the maintenance worker is finally called. As it can be seen in this little example, the time to react to a machine breakdown is high.
Material is put together order per order in the warehouse and then delivered to the production line. There is always the possibility that there is too less material for an order. So the worker has to first find the foreman and then the missing material has to be ordered from the warehouse in a complicated process.

5.4.3. IT Systems used in production

There are a number of standalone IT Systems used in production at the time of the study. One of the first steps was to unveil used IT-Systems and how they interact with the employees on the shopfloor. Figure 37 gives an overview of the existing IT Systems in suspension production at WP Performance Systems.

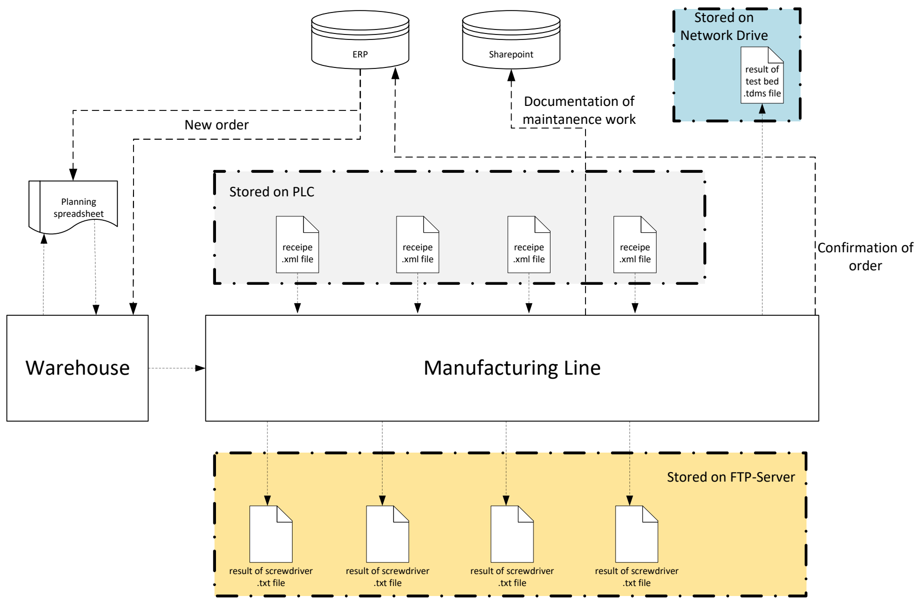

Figure 37.: Data on Shopfloor in Suspension (source: authors own representation)

First of all the orders are prepared in the central ERP system by the department of order management. The basic production dates are determined by this department as well. There is a time range of roughly two weeks in which the orders have to be proceeded. In suspension production there are meetings for detailed production planning twice a week. The detailed production plan is arranged in a spreadsheet.
In the warehouse the production orders are printed from the ERP system and needed material is picked for each workplace separately. The printed production orders are placed on the picked material and are delivered with the same to the workplace.
If a repair is necessary for a product, a repair order is created in the ERP system by the employee at the repair station. If spare parts are required, they will also be requested using this repair order. With the help of this procedure you can roughly check the first pass yield.
The configuration files for the production machines are stored on the PLC and the programs for the screwdrivers are stored on the units. There are backups from the configuration files as well as from the program files of the screwdrivers. Every product in the suspension production has to undergo a test cycle on a testing bench. For every test cycle a file with the test data is being generated. This file is stored on a network drive for further handling. The IT architecture has developed over time and therefore a overall strategy was not established.

5.5. Requirements

Once the problems with the current manual acquisition of production data have been identified, the requirements for a future system can be identified. First, the relevant stakeholders for this system have to be identified. The second step is to identify different use cases for a Production Data Acquisition System together with the stakeholders. From these use cases, the requirements can then be determined and documented in a structured form.

5.5.1. Stakeholder

Before requirements can be determined, the relevant stakeholders must first be identified. In the case of WP, the key stakeholders were identified as the following:

- Managing Director
- Production (Manager, Foreman, Worker)
- Industrial Engineering
- Maintenance
- Warehouse
- IT
- Quality
- Project Leader Strategic Projects

Besides observations in production, these stakeholders were the main sources of information for identifying the requirements for a Production Data Acquisition System.

5.5.2. Use Cases

Various use cases have been identified from interviews and discussions with the stakeholders on the one hand and from observations within the production environment on the other. After the individual use cases have been sorted and classified, the following groups have been emerged: *Worker Assistance*, *Alarm Management*, *Archiving* and *Reporting*. These use cases are represented in use case diagrams and also described in tabular form, according to the framework proposed by Rupp (2014). This method was chosen to represent the use cases and its necessary information in a very compact form.

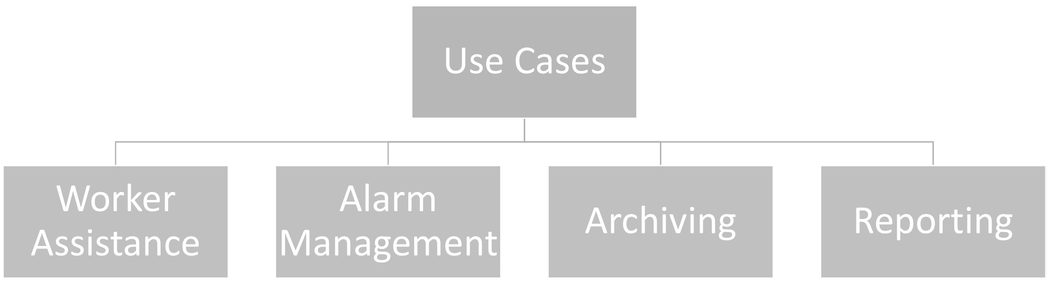

Figure 38.: Use Case Groups (source: authors own representation)

Worker Assistance

In this group, the use-cases that will assist the worker at the workplace will be collected. The use cases are shown in a use case diagram in figure 39. In the first use case, information about the current order should be made available live to the worker.

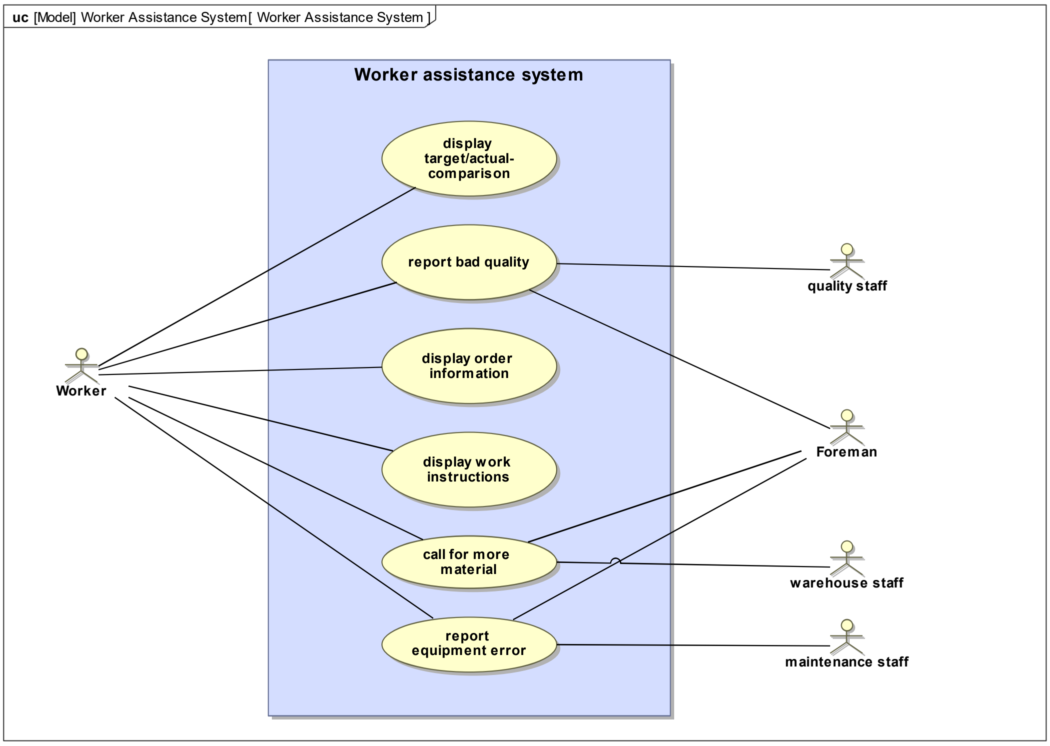

Figure 39.: Use Case Diagram Worker Assistance System (source: authors own representation)

The worker should know which order with which material number and to which quantity is currently being produced (UC2 in table 8).
Furthermore, the worker should get information about what his performance on the current job is (UC1 in table 7). Another use case is that the worker has the possibility to report problems that occur at the machine to the relevant persons by clicking on a button. First it has to be defined, what problems could occur on the production line. In several discussions the problems were restricted to three problem types: *missing material* (UC5 in table 11), *quality problems* (UC4 in table 10) and *equipment errors* (UC6 in table 12).
The intention of only three problem categories is, that the system should be as simple as possible for the worker at the production line. If necessary the problem types can be extended

easily. For each workstation there are product-specific work instructions which are currently attached to the machine in a hardcopy. In the future, it should be possible to make these available in electronic form at the workplace (UC3 in table 9). In the following the use cases are described in the tabular form as mentioned above.

Table 7.: UC1 - actual/target comparison

Use Case ID	UC1
Name	Show actual/target quantity comparison
Short description	On the HMI at the workstation, the worker should get information about his performance on the current order. So the target and actual quantity should be displayed.
Actors	Worker
Prerequisites	An order is started and the Employee-ID is on the RFID Reader
Trigger	Shown continously
Main scenario	• Worker is at the workstation and processing an order • Worker wants to know how he is performing • Worker can check his progress on the HMI, where two bars are showing the actual and target quantity

Table 8.: UC2 - order information

Use Case ID	UC2
Name	Display order relevant information
Short description	On the HMI at the workstation, the worker should get information about the current order. Relevant information should be Order Number, Material Number, Order Quantity and target pieces per hour.
Actors	Worker
Prerequisites	An order is started and the Employee-ID is on the RFID Reader
Trigger	Shown permanent
Main scenario	• Worker is at the workstation and processing an order • Worker wants to get information about the current order • Worker can check the order information on the HMI

Table 9.: UC3 - work instructions

Use Case ID	UC3
Name	Display work instructions
Short description	On the HMI at the workstation, the worker should get work instructions for the current product build at the workstation.
Actors	Worker
Prerequisites	An order is started and the Employee-ID is on the RFID Reader
Trigger	Worker needs instructions
Main scenario	• Worker is at the workstation and processing an order • Worker wants to get instructions how to assemble a product at this station • Worker presses the button "Work instructions" on the HMI • The work instructions for the current product are shown on the HMI

Table 10.: UC4 - report quality problem

Use Case ID	UC4
Name	Report Quality problem
Short description	The worker should get the possibility to report quality problems over the HMI at the workstation. Further the foreman and quality staff should be informed about the problem.
Actors	Worker, Quality staff, Foreman
Prerequisites	An order is started and the Employee-ID is on the RFID Reader.
Trigger	Quality problem occured
Main scenario	• Worker is at the workstation and processing an order • Worker wants to report a quality problem at his workstation • Worker presses the button "report quality problem" on the HMI • The foreman and quality staff get a message on their phone that a quality problem occured on the workstation.

Table 11.: UC5 - report missing material

Use Case ID	UC5
Name	Report Missing Material
Short description	The worker should get the possibility to report missing material over the HMI at the workstation. Further the foreman and warehouse staff should be informed about the problem
Actors	Worker, Warehouse staff, Foreman
Prerequisites	An order is started and the Employee-ID is on the RFID Reader.
Trigger	Material is missing
Main scenario	• Worker is at the workstation and processing an order • Worker wants to report missing material at his workstation • Worker presses the button "missing material" on the HMI • The foreman and warehouse staff get a message on their phone that material is missing at the workstation.

Table 12.: UC6 - report equipment error

Use Case ID	UC6
Name	Report Equipment Error
Short description	The worker should get the possibility to report equipment errors over the HMI at the workstation. Further the foreman and maintenance staff should be informed about the problem.
Actors	Worker, Maintenance staff, Foreman
Prerequisites	An order is started and the Employee-ID is on the RFID Reader.
Trigger	Equipment error occurs
Main scenario	• Worker is at the workstation and processing an order • Worker wants to report an equipment error at his workstation • Worker presses the button "equipment error" on the HMI • The foreman and maintenance staff get a message on their phone that an equipment error occured at the workstation.

Reporting

The acquired data should be evaluated and visualized in a meaningful way. For this reason, this group will take a closer look at use cases related to reporting issues. Several relevant types of reports have been identified and outlined in a Use Case diagram in figure 40. There has to be a report with data about historical orders (UC7 in table 13). Furthermore, a report on recorded errors has to be provided (UC8 in table 14). A report on the utilization of the individual production lines is of greater interest for future investment decisions (UC9 in table 15). To monitor the output of the individual production lines, there should be a report on the produced quantities (UC10 in table 16). An important parameter for the line balancing of a production line is the balancing factor. In order to be able to calculate it on the basis of actual data, there should be a report with the actual cycle times for each workstation (UC11 in table 17). In terms of a total productive maintenance strategy, it is of major interest which production units have the greatest potential for improvement in terms of avoiding downtimes. Therefore, there should be a report with availability key figures for each machine (UC12 in table 18).

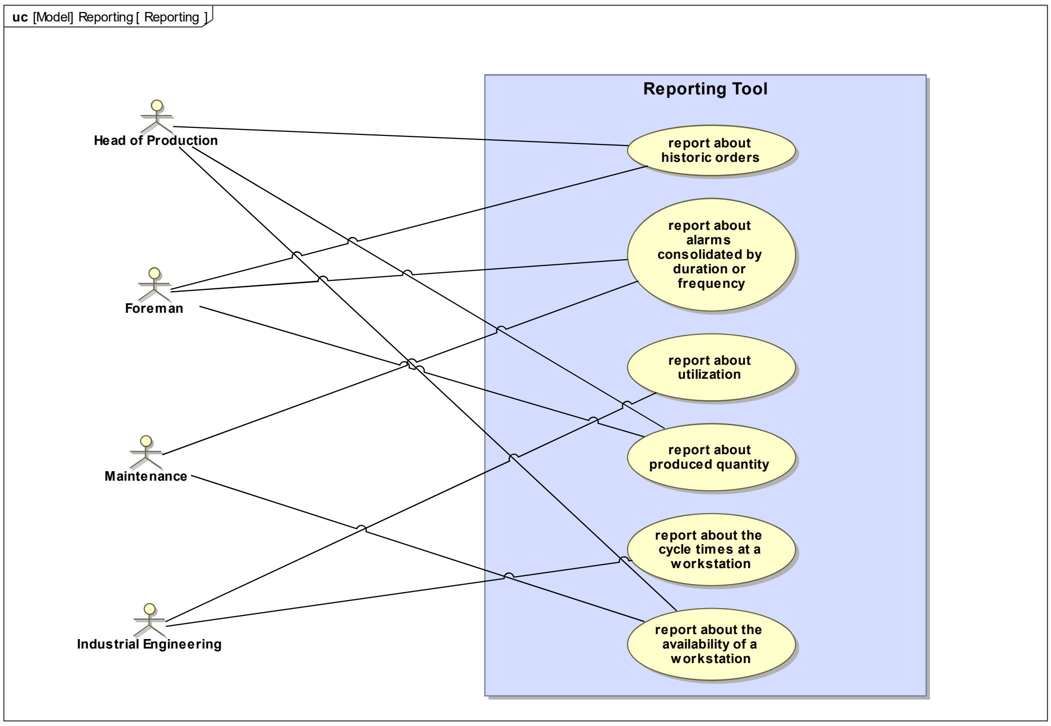

Figure 40.: Reporting use case diagram (source: authors own representation)

Table 13.: UC7 - report about historic orders

Use Case ID	UC7
Name	Report about historic orders
Short description	A report about historic order data should be provided
Actors	Head of Production, Foreman
Prerequisites	Order data is available
Trigger	Information about historic order is needed
Main scenario	• The user needs information about an historic order • The user is selecting a date range to filter the orders • The requested data is displayed in tabular form.

Table 14.: UC8 - Alarm report

Use Case ID	UC8
Name	Alarm report
Short description	A report with alarm data is shown. The data is consolidated either by duration or frequency. So the top alarms can be shown.
Actors	Foreman, Maintenance
Prerequisites	Alarm data is available
Trigger	Information about historic alarms is needed
Main scenario	• The user needs information about historic alarms. • The user is selecting a machine and a time range. • The user is selecting whether the data should be consolidated by duration or frequency. • The requested data is displayed in a table and a diagram.

Table 15.: UC9 - Utilization report

Use Case ID	UC9
Name	Utilization report
Short description	A report that shows the utilization of a machine.
Actors	Industrial Engineering
Prerequisites	Utilization data is available
Trigger	Information about utilization is needed
Main scenario	• The department of industrial engineering needs information about the utilization of a machine for example for investment decisions. • The user is selecting a machine and a time range. • The requested data is displayed in a table and a diagram.

Table 16.: UC10 - Yield report

Use Case ID	UC10
Name	Yield report
Short description	A report that shows the output of a production line.
Actors	Foreman, Head of Production
Prerequisites	Production data is available
Trigger	The actor needs information about the output of a production line
Main scenario	• The user needs information about the yield of a production line, to report it to his line manager. • The user is selecting a production line and a shift/day/week/. • The requested data is displayed in a table and a diagram.

Table 17.: UC11 - Report about cycle times

Use Case ID	UC11
Name	Report about cycle times
Short description	A report that shows the cycle times at a workstation for a period of time.
Actors	Industrial Engineering
Prerequisites	Cycle time data is available
Trigger	The user needs information about the cycle times at a workstation
Main scenario	• The user needs information about the cycle times at a workstation. One reason could be, that the buffer at this workstation is really high and this indicates that the cycle time at this workstation is significantly higher than at the other stations at this production line. • The user is selecting a workstation and an order. • The requested data is displayed in a table and a diagram.

Table 18.: UC12 - Report about availability

Use Case ID	UC12
Name	Report about the availability of a workstation
Short description	A report that shows the availability of a workstation for a period of time.
Actors	Maintenance, Head of Production, Foreman
Prerequisites	Availability data is available
Trigger	The user needs information about the availability of a workstation
Main scenario	• The user needs information about the availability of a workstation. • The user is selecting a workstation and a time range • The requested data is displayed in a table and a diagram.

Archiving

The third use case group deals with topics concerning the archiving of data that occur in various forms in production. In order to create a structured approach, the data were classified into the following categories:*product data*, *process data*, *machine data*, *alarm data* and *order data*. The use cases are shown in an use case diagram in figure 41. Product data are all those data that only affect the product. For example, the assignment of a test bench result to a serial number or the "marriage" of two products to get a complete product (UC13 in table 19). Process data are those data that are of fundamental importance for the respective production process. In the case of WP, two essential types of process data were identified: screwdriving process data and welding process data. With regard to the screwdriving process data, the angle of rotation and the tightening torque in the various screw driving phases are of particular interest. In the welding process data, the welding current, welding voltage and wire feed rate at the respective welding seams defined for the product. This use case is described in table 20.

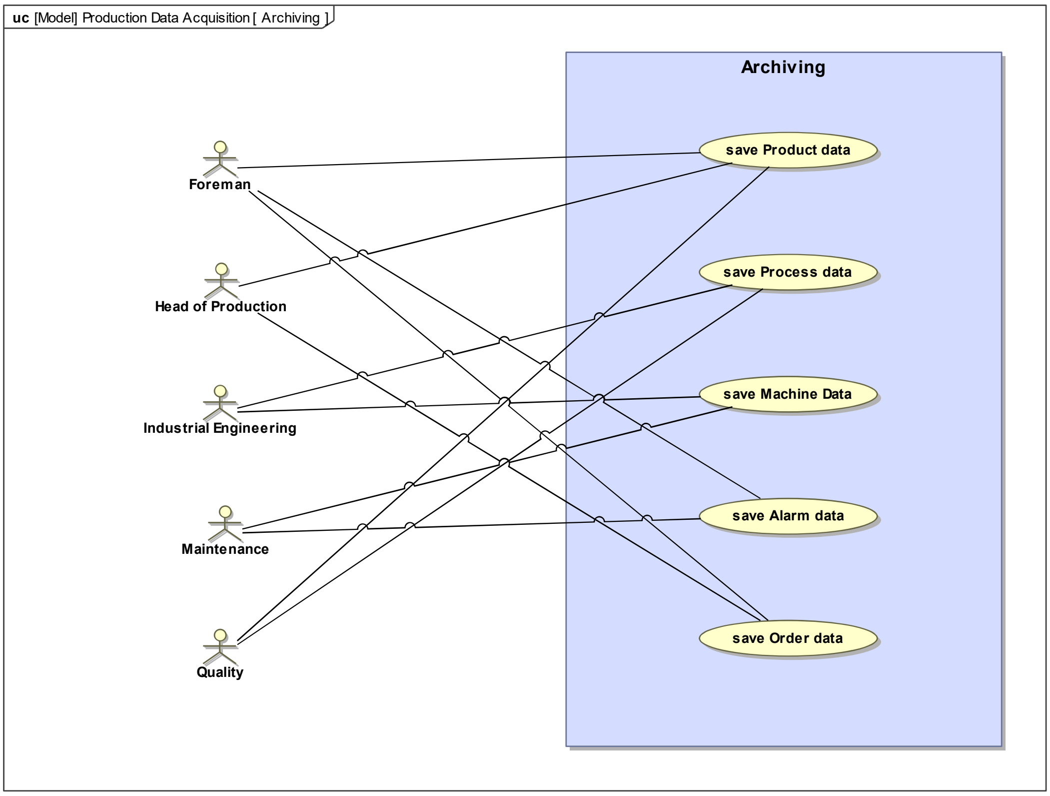

Figure 41.: Archiving use case diagram (source: authors own representation)

Machine data are all those data that are important for assessing the condition of an equipment. In the case of moving parts, for example, these can be motor currents or vibrations. Furthermore, this can also include data on the energy consumption of the respective equipment. Under alarm data all those data are bundled which are generated by errors in the equipment. In a specific case these can be the equipment number, start time, end time, error number, error text, cause and comments. This use case is described in table 21.
Order data is all the data that can be assigned to a specific production order (UC17 in table 23). This includes data such as order duration, start time, end time, employee, processing duration or various key figures for the order. Under alarm data all those data are bundled which are generated by errors in the equipment (UC16 in table 22). In a specific case these can be the system number, start time, end time, error number, error text, cause and comments. Order data is all the data that can be assigned to a specific production order. This includes data such as order duration, start time, end time, employee, processing duration or various key figures for the order.

Table 19.: UC13 - Archive Product Data

Use Case ID	UC13
Name	Archive Product Data
Short description	For each produced product, product data should be archived.
Actors	Foreman, Head of Production, Quality
Prerequisites	produced products
Trigger	A new product at a production unit to be produced.
Main scenario	• A product is produced at a production unit. • For this product relevant production data like order number, material number, serialnumber, good/bad, time stamp and employee id should be stored. • This should happen at each production unit, so the process within production is documented.

Table 20.: UC14 - Archive Process Data

Use Case ID	UC14
Name	Archive Process Data
Short description	Process Data should be archived continuously
Actors	Industrial Engineering, Quality
Prerequisites	running production unit
Trigger	production unit is operated by worker
Main scenario	• The production unit is operated by a worker. • For this production unit relevant process data should be stored continuously. Relevant process data can be data from screwdrivers, data from welding machines like current, voltage, wire feeded, but also data from test beds like graphs. • This should happen at each production unit, so the process documented and derivation can be detected.

Table 21.: UC15 - Archive Machine Data

Use Case ID	UC15
Name	Archive Machine Data
Short description	Machine Data should be archived continuously
Actors	Industrial Engineering, Maintenance
Prerequisites	running production unit
Trigger	start of the production unit
Main scenario	• The production unit is running. • For this production unit relevant machine data should be stored continuously. Relevant machine data can be motor currents, energy consumption, condition monitoring with vibration sensors, temperature, pressure ... • This should happen at each production unit. The data can be used for Predictive Analytics for example.

Table 22.: UC16 - Archive Alarm Data

Use Case ID	UC16
Name	Archive Alarm Data
Short description	Every Alarm should be archived
Actors	Foreman, Maintenance
Prerequisites	order started and production unit is running
Trigger	Alarm occurs
Main scenario	• The production unit is running and an alarm occurs • The alarm and regarding data like starting time, ending time, duration, machine number, alarm number, alarm text, alarm cause should be stored. • This should happen at each production unit. • The data can be used for reporting.

Table 23.: UC17 - Archive Order Data

Use Case ID	UC17
Name	Archive Order Data
Short description	For every order at every machine relevant data should be stored.
Actors	Foreman, Head of Production
Prerequisites	Order is finished
Trigger	Information about an order is needed
Main scenario	• The order is finished. • For every order at every involved machine relevant order data like order number, material number, order quantity, target cycle time, starting time, finishing time, lead time, KPIs, Employee ID should be stored. • The data can be used to get reports about the order.

Alarm Management

The alarm management refers to two different subsystems of the production data acquisition system. On the shop floor, this refers to the three types of errors that are triggered by an alarm button pressed by the worker. Furthermore, the automatically generated alarm texts

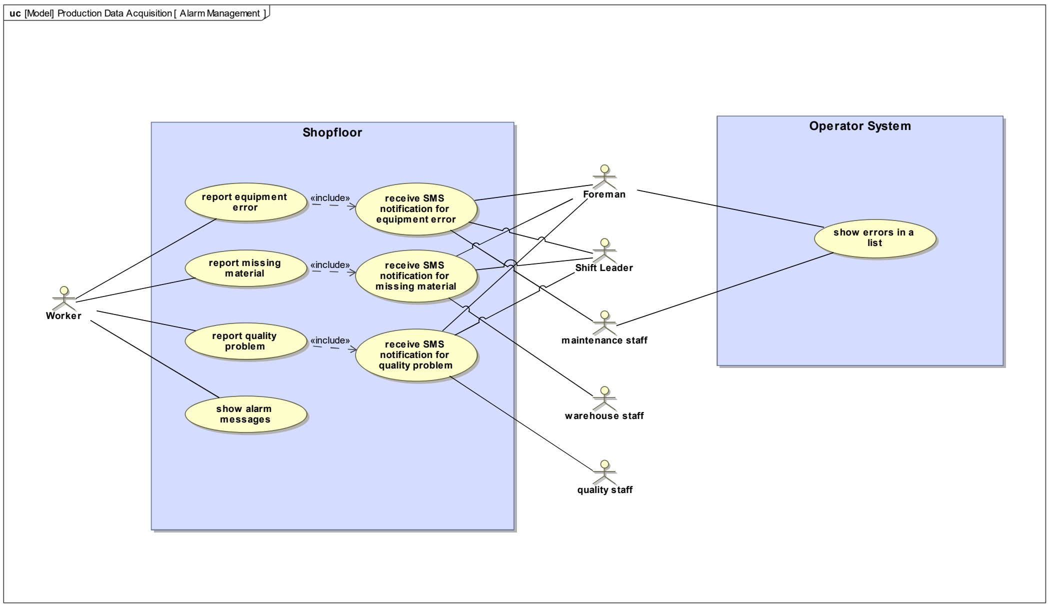

Figure 42.: Use Case Diagram Alarm Management (source: authors own representation)

should be displayed directly on the worker's HMI. In order to increase the response time to errors, the errors should be forwarded to the responsible persons by SMS (UC18 in table 24).

Table 24.: UC18 - Sending SMS Notification

Use Case ID	UC18
Name	Sending SMS Notification
Short description	For each alarm that occurs, an SMS notification is to be sent to the relevant persons.
Actors	Worker, Foreman, Quality Staff, Warehouse Staff, Maintenance
Prerequisites	running Order
Trigger	Alarm occurs
Main scenario	• An alarm occurs and is signaled either by the employee pressing a button or automatically by the system. • Depending on which category of alarm occurs, the corresponding person (Quality, Warehouse, Maintenance) is notified via SMS. • The person comes to the machine after notification and solves the problem and production can continue.

In the operator system, the alarms are to be displayed in a list, and these alarms are also to be saved (UC19 in table 25). It should be possible to display alarms broken down by type of error and frequency.

Table 25.: UC19 - Alarm list

Use Case ID	UC19
Name	Alarm list
Short description	Every Alarm should be shown in a list
Actors	Foreman, Maintenance
Prerequisites	data of alarms are available
Trigger	Information about alarms is needed
Main scenario	• The user needs information on malfunctions that have occurred at the systems. • The actor opens the Production Data Acquisition System to access the alarm list. • The user receives information on the start time, end time, duration, type of error, and machine where the error occurred.

5.5.3. Requirements specification

Once the use cases have been outlined, requirements for the system can be derived and formulated in normal language with the help of the template described in section 3.2. In order to ensure the traceability of requirements, each requirement is labeled with a unique identifier. The abbreviations stand for Functional Requirements (FR), Non-Functional Requirements (NFR) and Technical Requirements (TR). The requirements have been rated according to their importance using the MoSCoW principle (cp. section 4.2.1.2).

Functional Requirements

FR-1 **M** The system shall calculate the occupancy rate, performance, availability, quality and efficiency for every machine and show it in the detailed view of every machine.

FR-2 **M** At start up the system shall provide the user an overview of the production hall, with data about the machines in the hall.

FR-3	**M**	The system shall provide in the overview of the production hall, wether an order is active, wether the work station is occupied and wether the order is running within its limits (difference good parts to nominal parts, malfunctions).
FR-4	**M**	If the user clicks on a machine the system shall show detailed information about the machine.
FR-5	**M**	The system shall provide in the detailed view an overview about the current times (lead time, set-up time, production time, occupied time, break time, speed losses, quality losses) of an order.
FR-6	**M**	The system shall provide in the detailed view a comparison of the actual good parts produced with the actual nominal parts.
FR-8	**M**	If the user clicks on a "trend" button the system shall provide a screen on which the yield is continuously displayed in an x-t chart over time.
FR-9	**M**	If the user clicks on a "KPI Trend" button the system shall provide a chart with the course of the KPIs over time.
FR-20	**S**	The system should provide in the detailed view an overview of the calculated KPIs for this machine.
FR-21	**S**	The system should provide the user a report about the yield of every shift where the time filter can be set by the user.
FR-22	**S**	The system should provide the user a report about the historic orders, where data like the order number, material number, lead time, KPIs, starting time, ending time is shown.
FR-23	**S**	The system should provide the user a report about the utilization of a machine, where the time filter can be set by the user.
FR-24	**S**	The system should provide the user a report about malfunctions, where data like downtimes, starting time, ending time, error codes, causes and comments of the maintener are shown.
FR-25	**S**	The system should provide the user a report where the malfunctions are displayed and grouped by their cause and sorted by their frequency (Top n alarms).
FR-26	**S**	The system should provide the user a report with data about the production cycle times of a workstation.
FR-27	**S**	The system should provide the user a report about the availability of a production unit.
FR-28	**S**	The system should store product related data (order number, material number, serial number, good/bad, time stamp and employee id) in a database.

FR-29	**S**	The system should store process related data (torque and angle from screwdrivers, voltage, current and wire feeded from welding machines and test bed data) in a database.
FR-30	**S**	The system should store machine related data (motor currents, energy consumption, vibrational data, temperature, pressure) for each machine in a database.
FR-31	**S**	The system should store alarm data (starting and ending time, duration, machine number, error code, error text and cause) in a database.
FR-32	**S**	The system should store order related data (order number, material number, order quantity, target cycle time, good parts, bad parts, starting time, finishing time, lead time, KPIs and Employee ID) in a database.
FR-33	**S**	The system should provide the worker on the shopfloor information (order number, material number, quantity, starting time, target cycle time) about the current order.
FR-34	**S**	The system should provide the worker on the shopfloor information about his performance on the current order by showing him good parts produced, bad parts produced and nominal parts.
FR-35	**S**	In case a equipment error occured the system should provide the ability to alert the maintenance personal via SMS by pressing a button on the HMI.
FR-36	**S**	In case an acknowledgement-obligatory error occured the system should provide the ability to alert the foreman via SMS.
FR-37	**S**	In case of quality problems at the workstation the system should provide the ability to alert the quality personal via SMS by pressing a button on the HMI.
FR-38	**S**	In case of missing material at the workstation the system should provide the ability to alert the warehouse personal via SMS by pressing a button on the HMI.
FR-39	**S**	The system should be able to inform the maintener in case of a malfunction of a machine in automatic modus via SMS, where he gets information like machine name, starting time and error code.
FR-50	**W**	The system will provide the ability to the worker to report the number of units produced directly to the ERP from his workstation.
FR-51	**W**	The system will provide a central user authorization process according to the existing qualifications matrix.
FR-52	**W**	The system will be able to display the worker work instructions on the HMI at the workstation.

FR-53	**W**	The system will be able to link the assembled batches (batch number) with an order (order number).
FR-54	**W**	The system will be able to do the production planning.
FR-55	**W**	The system will be able to support the production logistics.
FR-56	**W**	The system will be able to monitor the energy consumption of each machine in the company.
FR-57	**W**	The system will be able to support the personal planning.
FR-58	**W**	The system will provide the ability to support paperless production.
FR-59	**W**	The system will be able to organize the recipe managment for all production units centrally.

Non-functional Requirements

NFR-1	**M**	There shall be a maintenance contract.
NFR-2	**M**	The system is to be designed/developed in the company independently.
NFR-3	**S**	The corporate design of WP Performance GmbH should be taken into account in the design of the user interface.
NFR-4	**S**	The system should be a module-based system.
NFR-5	**S**	The system should be expandable.
NFR-6	**S**	Due to possible language barriers, the worker assistance system on the shopfloor should be designed as intuitively as possible.
NFR-7	**S**	Additional machines should be able to be integrated with little effort.
NFR-11	**S**	The system should be able to connect with an existing database.

Technical Requirements

TR-1	**M**	The system must have a certified interface to SAP.
TR-2	**M**	At least OPC UA must be supported, further protocols are advantageous.
TR-3	**M**	RFID readers must support WP's employee identification tags. (Legic Prime MIM 256 system)
TR-4	**S**	The software should be based on the Windows operating system.
TR-5	**S**	The system should be accessible via a secure connection from the Internet.

TR-6	**S**	The system should be accessible on a mobile device via a native app or responsive design in the browser.
TR-7	**S**	The displays at the workplace should have at least 10" diagonal.
TR-8	**S**	The system should also be able to use data from external databases for reporting and analysis.

5.6. Development

Following the aim of providing live data from the shopfloor to the management, a Production Data Acquisition system was developed. The development combines the findings in the as-is analysis with the recommendations given by the literature study and the requirements raised at the company. The production data acquisition system treated in this thesis consists of the software tool, but also of the integration in the shopfloor. First, potential software vendors are examined in a technology screening to finally select a suitable system with the help of a value-benefit analysis.

5.6.1. Technology Screening

There are more than 200 individual software solutions competing in the market for production data acquisition. For that reason it is a challenging task finding a sufficient solution which is meeting the companies requirements.

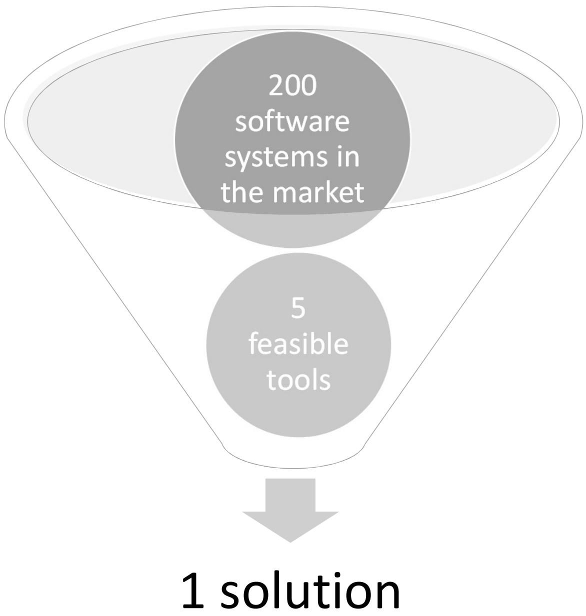

Figure 43.: Software selection procedure (source: authors own representation

Information about the tools that are currently available on the market was performed on

multiple channels. This included web search, visiting a trade fair in Nuremberg, interviews within the company and also reference visits. A well-grounded selection procedure has been applied.
First of all the amount of software solutions is reduced by the evaluation of must-have functions. The most important must-have function was the SAP-interface. It was also important that the provider offered a sales office and support in Austria.
The remaining software systems have to undergo a more in-depth analysis. In the next paragraphs the narrower range of software systems are presented. The selection procedure is visualized in figure 43.

Process Master 4.0 - Gamed GmbH

Gamed is a enterprise situated in Graz. Their portfolio consists of six different software modules for manufacturing environment. They are offering tools for Production Data Acquisition, Quality Management, HMI, Maintenance Management and CIP.

Figure 44.: Logo Gamed (source: http://www.gamed.com/img/logo-gamed.png)

Automation X - Automation X GmbH

Founded in 1989 Automation X is specialized in not only hardware but also software solutions for the automation industry. Today the company is located in Graz. Their product is a software framework for production data acquisition. This framework includes an huge amount of standard interfaces and drivers for PLC. Their framework contains also an IDE to develop a tailored solution for the company.

Figure 45.: Logo Automation X (source: http://www.automationx.at)

IPM - CSP

CSP Software is an enterprise located near Deggendorf in Bavaria. Their core competence are software products regarding data acquisition of screwdrivers. Within the KTM group a product of CSP is already in use. The software *QS Torque* is used for the investigation of process capability and machine capability. It is obvious that the other product of CSP *Integriertes Prozessmanagement - IPM* has to take part in the evaluation of feasible software tools. The software doesn't offer OPC UA support.

Figure 46.: Logo CSP (source: https://www.csp-sw.de/wp/wp-content/uploads/2016/02/logo-380px.png)

Zenon - Copa-Data GmbH

Copa-Data is a enterprise based in Salzburg. They are developing and selling software for more than 30 years. Their product consists of three layers with different modules for each. The greatest strength of zenon, the product of Copa-Data, are the more than 300 drivers for almost every kind of PLC or machine on the market. They are offering a Soft PLC, a SCADA system and an Analyzer Software as well. Regarding the companies requirements, the zenon Software needs a lot of own contributions. The software architecture is based on a server/client model with built in redundancy feature. The delivered software package can be enhanced by self developed software extensions in Visual Studio Tools for Applications.

Figure 47.: Logo Copa Data (source: https://www.copadata.com/fileadmin/images/logo.png)

Cronetworks - Industrieinformatik GmbH

The company Industrieinformatik is situated in Linz and they are selling their MES software since more than 30 years as well. Their software portfolio offers a variety of modules for manufacturing processes. In contribution to the other software tools, Cronetworks is a classical MES system and covering a lot of the functionalities described in chapter 2. A big advantage of this system is the extensive amount of functionalities, but in contrast a lot of work from the software provider is necessary to get a tailored concept for the own production.

Figure 48.: Logo Industrieinformatik (source: http://www.industrieinformatik.com)

5.6.2. Value Benefit Analysis

For an impartial selection of a software system, a value benefit analysis was done. The value benefit analysis is a method to compare alternatives in decision making processes. The method allows to consider qualitative as well as quantitativ criteria. The procedure for a value benefit analysis was explained in detail in section 4.2.1.2. The following approach for determining the value benefit was applied:

First of all criteria was derived from the requirements established in section 5.5. Secondly the found criteria was weighted with the method of pairwise comparison. To evaluate the alternatives on their subtarget fulfillment, scaling matrices are generated. The sub value benefit of each group was calculated. In the last step the overall value benefit of each tool was calculated by adding up all sub value benefits.

Criteria

The criteria are basically derived from the requirements developed in section 5.5. The resulting criteria were defined together with the project members. It is important to find the balance between information overload and too less criteria. For a better handling the criteria are categorised in groups, which are the following:

basic:

In this group, criteria associated with basic characteristics of Production Data Acquisition Systems as well as the characteristics of their vendors will be considered. Such criteria concern the modularity of the system, the extensibility or whether the engineering is done in-house or by the software manufacturer. Another important point is how dependent you will be on the software vendor. A good indication whether a software is suitable for your own company is the reference installations in other companies that may even be active in the same industry.

system:

This group includes criteria which belongs to system purposes. Criteria within this group are supported operating systems or the ease of installation the software. With the increasing interconnection of production equipment, the topic of information security in particular is becoming an increasingly important topic, and for this reason it should not be ignored in this context.

functionalities:

In this group the functionalities, that such a PDA system should have, should be considered. On the one hand, functionalities according to the definitions in chapter 2 are discussed here, but also other more narrowly defined ones such as alarm management.

interfaces & database:

This group contains criteria regarding interfaces to the ERP system as well as to the shopfloor. Also the database, storing the production data, is a criterion. It is often necessary to compare the acquired data with data from other sources than the PDA System. For this reason, the connection of data from third-party databases should also be a criterion for the selection.

support & training:

In this group criteria regarding support and training is pooled. Because the support offered by a software manufacturer is very important in the initial phase of a project, this point was included as a separate criterion. The same applies to the training offered and the training needed for the respective variant.

analysis & reporting:

All the recorded data would be worthless if there were no appropriate tools and methods for evaluating them. So in the analysis & reporting section criteria like statistic tools, report generation, kpi calculation etc. are gathered.

Weighting

In the previous section criteria for Production Data Acquisition Systems were established. Not every criterion has the same importance so a prioritization was done. The weighting of the established criteria was done by pairwise comparison. Pairwise comparison is a method to compare each criterion with each other and ensure that no criterion is overrated. To achieve a balancing effect the rating was done in collaboration with three project members. The weighting was backed by the found requirements. Pairwise comparison was done group by group to ensure comparability of the criteria.

Table 29.: Pairwise comparison of group categories

more important \ than	basic	functionality	interface & database	support & training	analysis & reporting	sum	relative
basic		0	0	1	0	1	5%
functionality	2		2	2	2	8	40%
interface & database	2	0		2	2	6	30%
support & training	1	0	0		0	1	5%
analysis & reporting	2	0	0	2		4	20%

A 0 means that the other criterion is more important, a 2 means that the criterion is more important than the other and a 1 means that both are of the same importance. So first of all the weights for the level 1 criteria were obtained,this is shown in table 29. As a result *functionalities* and *interface & database* are from high importance while *support* and *basic* criteria are from minor importance.

In the next step the subcriteria of each group were compared with each other. This has to be done for each category separately. In table 30 the pairwise comparison for basic criteria is shown. In the basic criteria group the criterion with the most importance is the *engineering (inhouse/extern)* criterion.

Table 30.: Pairwise comparison of basic criteria

more important \ than	modular	expandable	engineering (inhouse/extern)	dependency on software provider	effort to connect a new machine	references	supported operating systems	security	sum	relative
modular		0	0	1	0	0	2	2	5	10%
expandable	2		0	2	0	0	2	2	8	16%
engineering (inhouse/extern)	2	2		2	2	2	2	2	14	28%
dependency on software provider	1	0	0		1	2	2	1	7	14%
effort to connect a new machine	2	2	0	1		2	2	2	11	22%
references	2	2	0	0	0		1	0	5	10%
supported operating systems	0	0	0	0	0	1		0	1	2%
security	0	0	0	1	0	2	2		5	10%

The group with the highest quantity of criteria is the functionality category. There are five criteria which have more or less the same importance. This are production- , machine- and process data acquisition, maintenance and alarm management.

Table 31.: Pairwise comparison of functionality criteria

more important \ than	production planning	production data acquisition	machine data acquisition	process data acquisition	tracking & tracing	maintenance	alarm management	intra logistics	paperless production	receipe management	user management	sum	relative
production planning		0	0	0	1	0	0	1	1	0	0	3	3%
production data acquisition	2		2	2	2	2	2	2	2	2	2	20	18%
machine data acquisition	2	0		2	2	2	1	2	2	2	2	17	15%
process data acquisition	2	0	0		2	1	1	2	2	2	2	14	13%
tracking & tracing	1	0	0	0		0	0	1	0	1	1	4	4%
maintenance	2	0	0	1	2		1	2	2	2	2	14	13%
alarm management	2	0	1	1	2	1		2	2	2	2	15	14%
intra logistics	1	0	0	0	1	0	0		1	1	2	6	5%
paperless production	1	0	0	0	2	0	0	1		2	2	8	7%
receipe management	2	0	0	0	1	0	0	1	0		1	5	5%
user management	2	0	0	0	1	0	0	0	0	1		4	4%

In the interface & database category the criteria with the highest importance is obviously the SAP- interface and the OPC-UA protocol. But the implementation of third party databases is not negligible, as it can be seen in table 32.

Table 32.: Pairwise comparison of interface and database criteria

than / more important	database included	using third party data	SAP interface	OPC UA	number of drivers	sum	relative
database included		1	0	0	0	1	5%
using third party data	1		0	0	0	1	5%
SAP interface	2	2		1	2	7	35%
OPC-UA	2	2	1		1	6	30%
number of drivers	2	2	0	1		5	25%

In the support and training category the most important criterion is the amount of needed training, followed by the offered training by the software provider. The distance to the company is of minor importance as shown in table 33.

Table 33.: Pairwise comparison of support and training criteria

than / more important	distance from company	training offered	training needed	sum	relative
distance from company		1	0	1	17%
training offer	1		1	2	33%
training needed	2	1		3	50%

The last group consists of criteria about reporting and analysis. There are four criteria with high importance as shown in table 34. The more important criteria are KPI calculation, trend visualization, reports and dashboards.

Table 34.: Pairwise comparison of report and analysis criteria

than / more important	KPI calculation	basic statistics	trend visualization	reports	event based report generation	mobile app	dashboards	sum	relative
KPI calculation		2	1	0	2	2	1	8	19%
basic statistics	0		0	0	1	1	1	3	7%
trend visualization	1	2		2	1	1	1	8	19%
reports	2	2	0		2	2	1	9	21%
event based report generation	0	1	1	0		1	1	4	10%
mobile app	0	1	1	0	1		0	3	7%
dashboards	1	1	1	1	1	2		7	17%

Scaling Matrix

As theoretically developed in the Concept Design section (4.2.1.2), the assessments of the individual criteria must take place on a standardized scale. For this purpose, scaling matrices can be used that transform the ordinal and nominal scales to a 1 to 10 scale. In table 35 the scaled criteria is shown.

Table 35.: Scaling Matrix of Criteria

<table>
<tr><th>grading / criterion</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th></tr>
<tr><td>modular</td><td colspan="3">not modular</td><td colspan="4">limited modular</td><td colspan="3">fully modular</td></tr>
<tr><td>expandable</td><td colspan="3">not explandable</td><td colspan="4">limited expandable</td><td colspan="3">fully expandable</td></tr>
<tr><td>engineering (intern/extern)</td><td colspan="3">only extern</td><td colspan="2">mostly extern</td><td colspan="2">mostly intern</td><td colspan="3">only intern</td></tr>
<tr><td>dependency on software provider</td><td colspan="2">very high</td><td colspan="2">high</td><td colspan="2">moderate</td><td colspan="2">low</td><td colspan="2">very low</td></tr>
<tr><td>effort to connect a new machine</td><td colspan="2">very high</td><td colspan="2">high</td><td colspan="2">moderate</td><td colspan="2">low</td><td colspan="2">very low</td></tr>
<tr><td>references</td><td colspan="2">very bad</td><td colspan="2">bad</td><td colspan="2">moderate</td><td colspan="2">good</td><td colspan="2">very good</td></tr>
<tr><td>supported operating systems</td><td colspan="3">restricted to one OS</td><td colspan="4">two different OS</td><td colspan="3">more than two</td></tr>
<tr><td>security</td><td colspan="2">very bad</td><td colspan="2">bad</td><td colspan="2">moderate</td><td colspan="2">good</td><td colspan="2">very good</td></tr>
<tr><td>production planning</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>production data acquisition</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>machine data acquisition</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>process data acquisition</td><td colspan="2">not offered</td><td colspan="2">offered</td><td colspan="2">with tolerances</td><td colspan="2">action limits</td><td colspan="2">spc</td></tr>
<tr><td>tracking & tracing</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>maintenance</td><td colspan="2">not offered</td><td colspan="3">maintenance plan</td><td colspan="3">with warnings</td><td colspan="2">Pred. Analytics</td></tr>
<tr><td>alarm management</td><td colspan="3">not offered</td><td colspan="4">recording of disruptions</td><td colspan="3">recording and alarming</td></tr>
<tr><td>intra logistics</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>supporting paperless production</td><td colspan="2">not offered</td><td colspan="3">orders</td><td colspan="3">orders and instructions</td><td colspan="2">videos</td></tr>
<tr><td>receipe management</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>user management</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>database included</td><td colspan="3">is not included</td><td colspan="4">is included with limitations</td><td colspan="3">is included</td></tr>
<tr><td>using third party data</td><td colspan="3">cannot be used</td><td colspan="4">can be used with limitations</td><td colspan="3">can be used</td></tr>
<tr><td>SAP interface</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>OPC-UA</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>number of drivers</td><td>1</td><td>3</td><td>5</td><td>10</td><td>20</td><td>30</td><td>50</td><td>100</td><td>200</td><td>>300</td></tr>
<tr><td>distance from company (km)</td><td>>500</td><td>500</td><td>400</td><td>300</td><td>200</td><td>150</td><td>100</td><td>80</td><td>50</td><td>30</td></tr>
<tr><td>training offered (days)</td><td>0</td><td>2</td><td>4</td><td>6</td><td>8</td><td>10</td><td>12</td><td>14</td><td>16</td><td>>18</td></tr>
<tr><td>training needed (days)</td><td>18</td><td>16</td><td>14</td><td>12</td><td>10</td><td>8</td><td>6</td><td>4</td><td>2</td><td><2</td></tr>
<tr><td>KPI calculation</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>basic statistics</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>trend visualization</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>reports</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>event based report generation</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
<tr><td>mobile app</td><td colspan="3">not offered</td><td colspan="4">mobile HTML page</td><td colspan="3">iOS and Android</td></tr>
<tr><td>dashboards</td><td colspan="3">not offered</td><td colspan="4">offered with limitations</td><td colspan="3">offered</td></tr>
</table>

Evaluation

After all previous steps had been completed, the individual benefit values for each variant could be calculated. To do this, the benefit values of the single groups were first calculated in order to add them up and to obtain the total benefit value of the respective variant. The value benefit analysis can be seen in detail in table 36 and its result is presented in figure 5.6.2.

Table 36.: Evaluation of different Software tools

			Zenon		Cronetworks		CSP		Automation X		GAMED	
criteria	**weighting**		**rating**	**VB**	**rating**	**VB**	**rating**	**VB**	**rating**	**VB**	**rating**	**VB**
	level 1	level 2										
basic	5%											
modular		9%	7	0,03	10	0,04	3	0,01	10	0,04	10	0,04
expandable		14%	9	0,06	10	0,07	5	0,04	10	0,07	10	0,07
engineering (inhouse/extern)		25%	10	0,13	4	0,05	1	0,01	10	0,13	1	0,01
dependency on software provider		13%	9	0,06	2	0,01	1	0,01	7	0,04	1	0,01
effort to connect a new machine		20%	10	0,10	3	0,03	1	0,01	10	0,10	7	0,07
references		9%	10	0,04	8	0,04	1	0,00	5	0,02	3	0,01
supported operating systems		2%	5	0,00	7	0,01	10	0,01	10	0,01	10	0,01
security		9%	10	0,04	8	0,04	7	0,03	8	0,04	5	0,02
Partial Benefit Value				0,47		0,29		0,12		0,45		0,25
functionality	40%											
production planning		3%	1	0,01	10	0,11	1	0,01	10	0,11	10	0,11
production data acquisition		18%	10	0,73	10	0,73	10	0,73	10	0,73	10	0,73
machine data acquisition		15%	10	0,62	10	0,62	10	0,62	10	0,62	10	0,62
process data acquisition		13%	10	0,51	7	0,36	10	0,51	10	0,51	7	0,36
tracking & tracing		4%	6	0,09	10	0,15	10	0,15	10	0,15	10	0,15
maintenance		13%	10	0,51	10	0,51	1	0,05	8	0,41	10	0,51
alarm management		14%	10	0,55	5	0,27	3	0,16	10	0,55	1	0,05
intra logistics		5%	3	0,07	6	0,13	1	0,02	10	0,22	1	0,02
supporting paperless production		7%	3	0,09	10	0,29	1	0,03	1	0,03	8	0,23
receipe management		5%	10	0,18	10	0,18	10	0,18	10	0,18	8	0,15
user management		4%	10	0,15	10	0,15	5	0,07	10	0,15	10	0,15
Partial Benefit Value				3,49		3,49		2,53		3,64		3,07
interface & database	30%											
database included		5%	1	0,02	10	0,15	1	0,02	10	0,15	7	0,11
using third party data		5%	10	0,15	1	0,02	1	0,02	1	0,02	10	0,15
SAP interface		35%	10	1,05	10	1,05	10	1,05	5	0,53	10	1,05
OPC-UA		30%	10	0,90	10	0,90	1	0,09	10	0,90	10	0,90
number of drivers		25%	10	0,75	4	0,30	3	0,23	5	0,38	4	0,30
Partial Benefit Value				2,87		2,42		1,40		1,97		2,51
support & training	5%											
distance from company		17%	10	0,08	6	0,05	8	0,07	1	0,01	1	0,01
training offer		33%	10	0,17	8	0,13	1	0,02	2	0,03	2	0,03
training needed		50%	8	0,20	1	0,03	7	0,18	10	0,25	8	0,20
Partial Benefit Value				0,45		0,21		0,26		0,29		0,24
analysis & reporting	20%											
KPI calculation		19%	10	0,38	10	0,38	8	0,30	10	0,38	10	0,38
basic statistics		7%	10	0,14	3	0,04	10	0,14	10	0,14	10	0,14
trend visualization		19%	10	0,38	8	0,30	5	0,19	10	0,38	10	0,38
reports		21%	10	0,43	10	0,43	10	0,43	10	0,43	10	0,43
event based report generation		10%	10	0,19	10	0,19	10	0,19	10	0,19	10	0,19
mobile app		7%	10	0,14	5	0,07	10	0,14	10	0,14	10	0,14
dashboards		17%	10	0,33	10	0,33	10	0,33	10	0,33	10	0,33
Partial Benefit Value				2,00		1,75		1,73		2,00		2,00
Total Benefit Value				**9,27**		**8,15**		**6,04**		**8,34**		**8,06**

The highest value determines the highest benefit value of the respective solution and thus a preference for a decision. In this case, the results are ranked by its benefit value and have the following order:

1. **Zenon** 9,27
2. **Automation X** 8,34
3. **Cronetworks** 8,15
4. **Gamed** 8,06
5. **CSP** 6,04

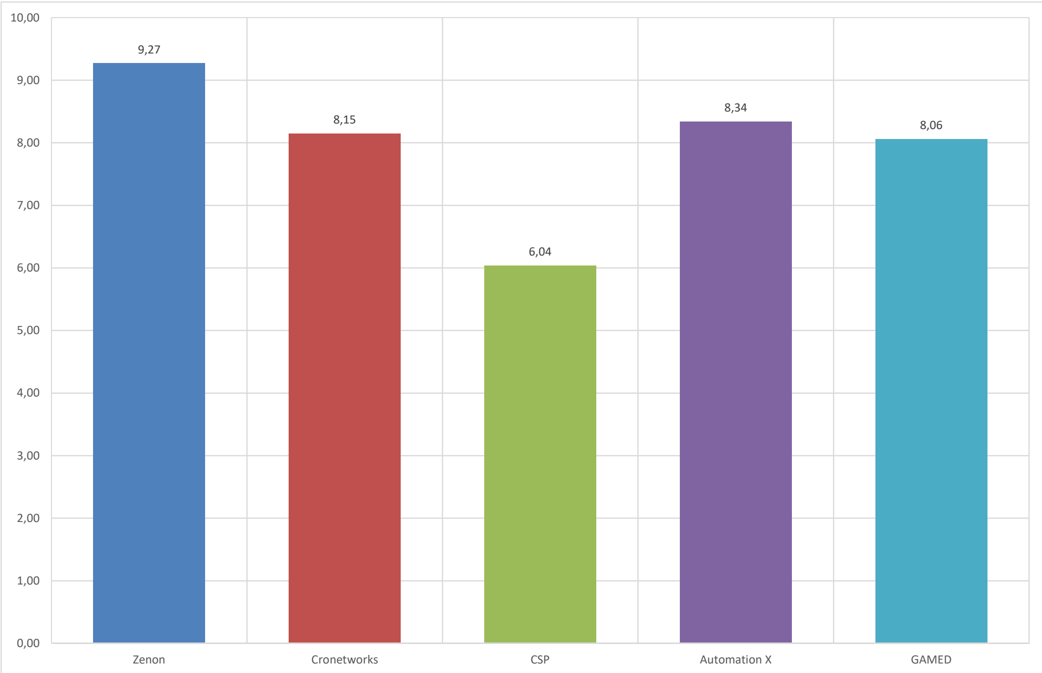

Figure 49.: Result of the Value Benefit Analysis

Recommendation

The result was then presented to the stakeholders as a decision suggestion. Because the result was very narrow and a value benefit analysis is always influenced by subjective opinions despite its objective evaluation, the best rated solution was recommended to be tested in a Proof-of-Concept for its suitability as a Production Data Acquisition System at WP. Before the installation of the Proof-of-Concept can be started, the concept for implementation must first be developed. The system architecture and its more detailed concept development is described in more detail in the next section.

5.7. Concept and Implementation

After the evaluation of the most feasible tool for Production Data Acquisition at WP Performance Systems, the concept for the implementation was designed. First of all the System Architecture was defined. Later the selected architecture was developed more precisely by developing a concept design. Based on the identified use cases and the requirements gathered subsequently, the concept of implementation has been elaborated.

5.7.1. System Architecture

In this section the system architecture will be discussed. The overall system architecture is based on the automation pyramid, which has already been described in chapter 2. In figure 5.7.1 the automation pyramid, with the products in use, respectively those who will be used in the future, is shown.

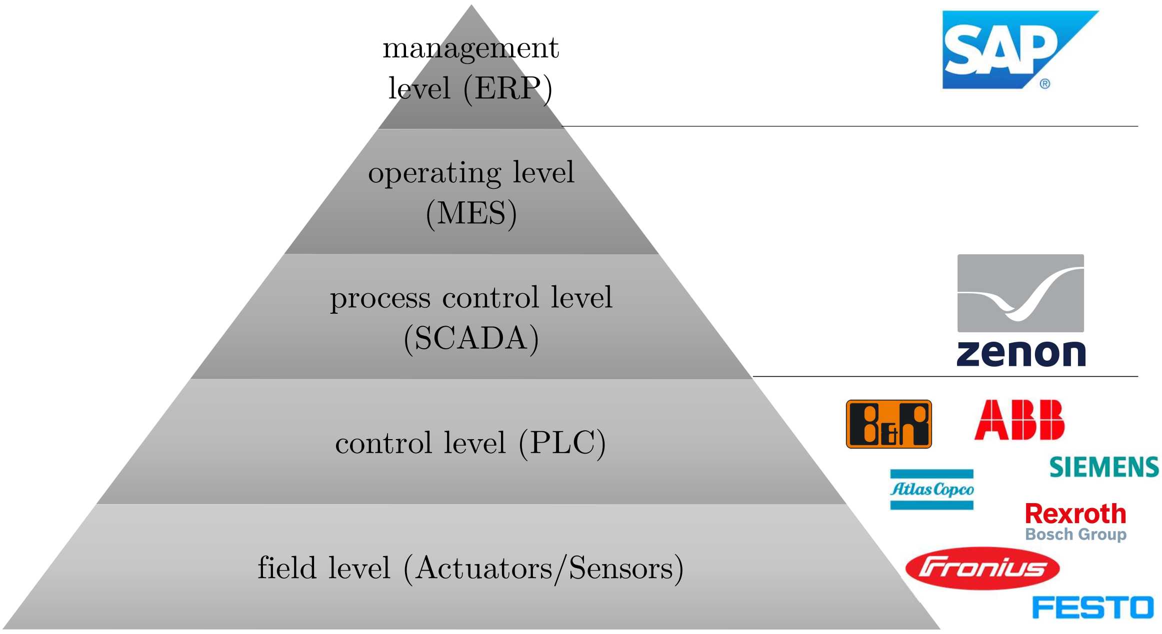

Figure 50.: System Architecture based on automation pyramid

The central ERP system of the company is thus at the top of the pyramid. WP uses the ERP system from SAP. At the lower end of the pyramid is the control and field level with its PLCs,

sensors and actuators. There are different types of sensors from different manufacturer in use. Although the products of *Bernecker & Rainer* is the companies standard for PLC systems, there are systems from other manufacturers in use in the company. There are obviously machines with different automation degrees. So a concept had to be found that integrates all machines. The operating and process control layer is to be covered by the new production data acquisition software *zenon*. Logical production units, such as a production line, are combined into a group and will have a central common line terminal installed. A more detailed description of what a line terminal is, is explained in section 5.7.1.
Every in the system integrated workstation is equipped with a so called Worker Terminal. More details on a Worker Terminal can be found in section 5.7.1 below.
The data from the integrated systems is collected by the zenon software running on a server. Users can access the software and its features via a client. The architecture can be seen in figure 5.7.1 as a schematic representation.

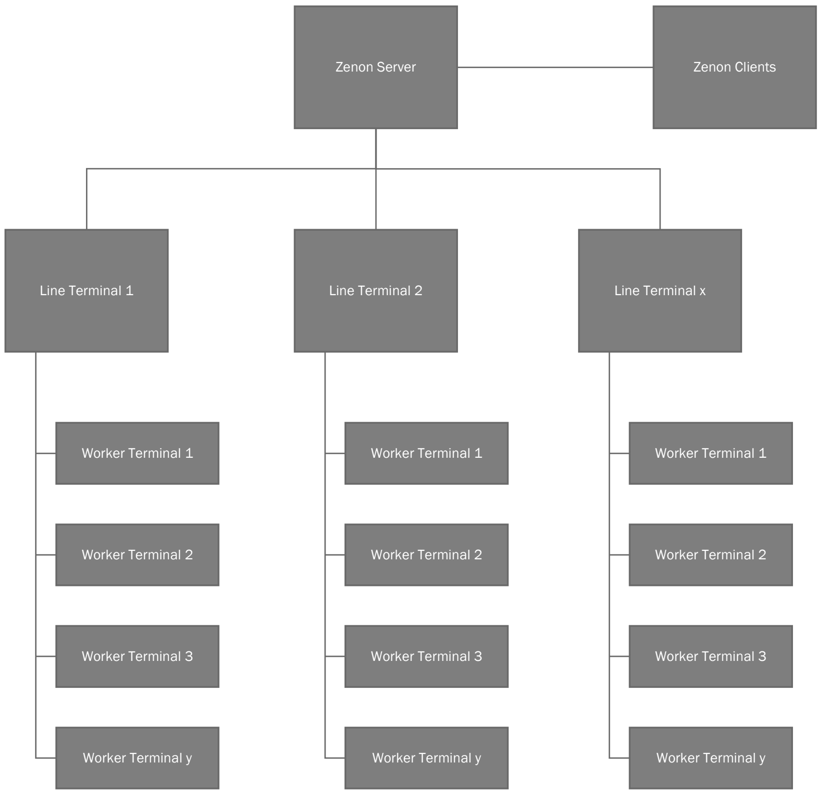

Figure 51.: System Architecture with Server, Worker Terminals and Line Terminals

Line Terminal

Logical units, like a production line, are composed in a so called line terminal. The line terminal consists of a central PLC where all the data from the workstations are collected. Furthermore, there is a touch monitor and a QR code scanner to scan the QR-code on the order sheet and therefore start the order.

Worker Terminal

Every integrated work station is equipped with a worker terminal. The worker terminal consists of a industry display and a RFID-Reader. Since most of the workstations are not occupied over the whole shift, it is necessary to track the occupation of the work station. This is done with the employees ID card on the RFID-Reader. On the touch display relevant information will be shown, but also interaction with the system is possible. The visualization on the worker terminals is realized with a B&R system.

Zenon Server

The zenon server is installed on a standard Windows Server. To gurantee a high availability a second standby server is also used. The failure detection is implemented within the zenon software itself. The application runs on the server and the connection to the individual workstations is established for this purpose. The *projects* which are running on the Zenon Server are developed in an Integrated Development Environment (IDE) called Zenon Editor. The development is happening on an engineering PC. The projects are later compiled and the executeable files are transmitted to the Server. On the server the Zenon Runtime is starting all the different executeable files from the single projects. The server also establishes the connections to the individual PLCs.

Zenon Client

The zenon client is used to grant the users access to the production data acquisition system. The Client software is accessed either on a webbased solution or a native client solution.

5.7.2. System Concept

The coarse system architecture described in the previous section has now been worked out in more detail as a concept. For this purpose, the concept design is divided into two areas. On the one hand, the development of an operator system in which certain users can obtain superordinate information about the running production processes. On the other hand, the development of a worker interaction and information system that serves the worker on the shop floor.

Operator

The operator system is designed to deliver the required information to a user's desk. Several elements have been implemented which are briefly introduced below.

Dashboard of the production hall
In order to provide the user with an overview of all production facilities, a hall overview is to be prepared. This hall overview should make it possible to see at a glance what the status of the individual production facilities is. Furthermore, the information whether a workstation is occupied should be identifiable at first sight. In figure 52 an example of a Dashboard is shown.

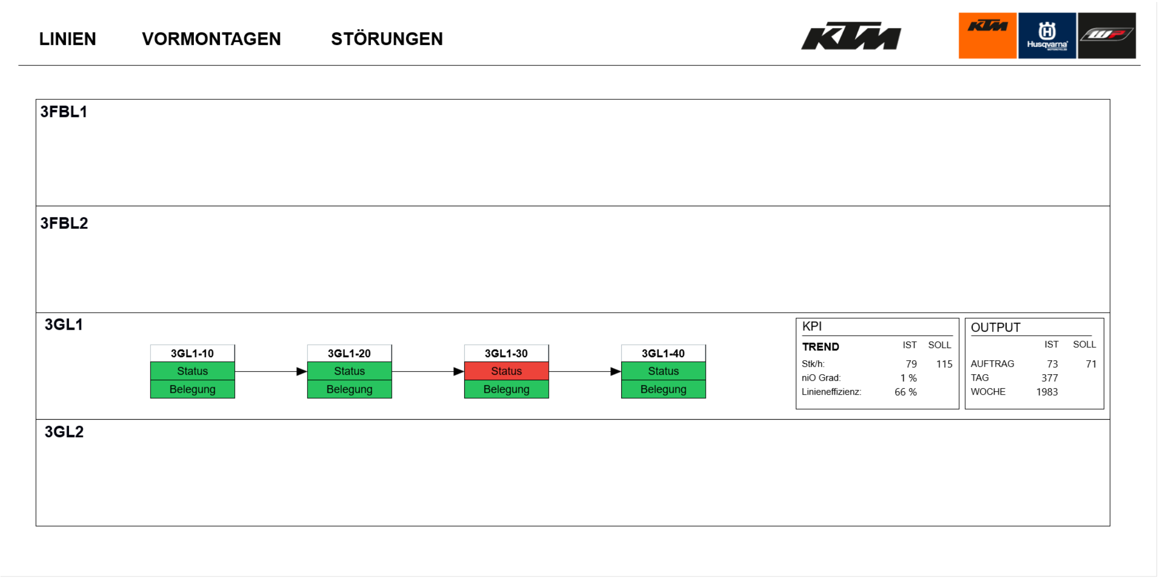

Figure 52.: The dashboard of the production hall in suspension production.

According to the concept from coarse to detail, you can switch to the detailed view by clicking on the machine name.

Detailed view of a workstation

Information on the status of an asset can be called up in the detailed view of a workstation.

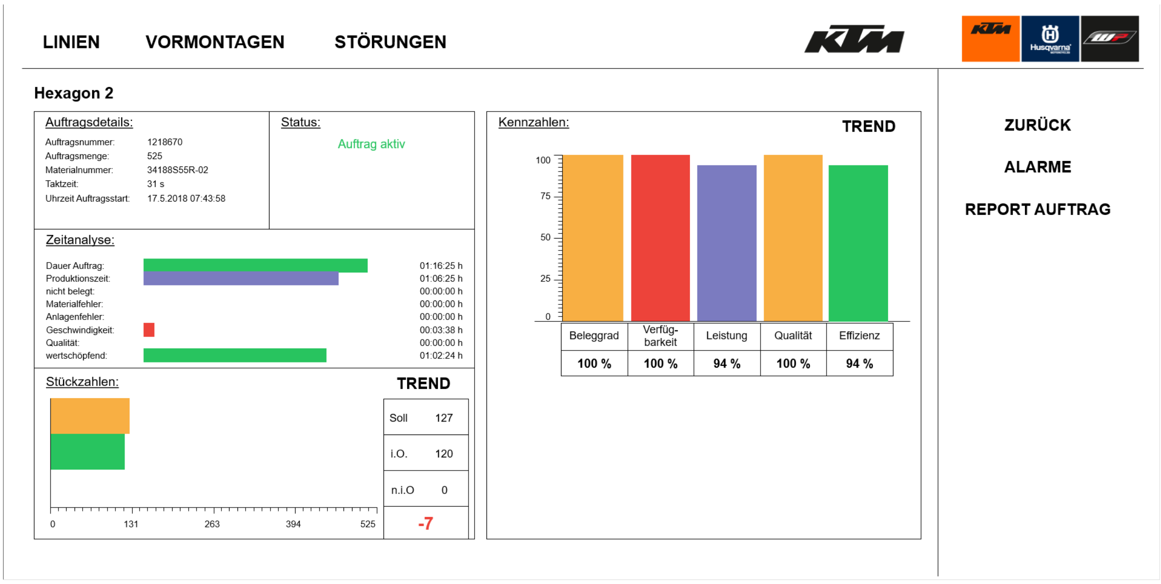

Figure 53.: The detailed view of a workstation in Suspension production

Among other things, it indicates whether an order is currently running and whether malfunctions are present. When an order is running, the master data is displayed. A time analysis is also displayed for the current order. In addition, the produced pieces (good and bad) and the nominal pieces are displayed. Key performance indicators can then be calculated from these times and quantities. How the KPIs are calculated is shown in Appendix A. Beside an individual alarm list for this machine, reports can also be displayed. Figure 53 shows an example of a detailed view.

List of alarms

To get an overview of where, when and which alarm has occurred, these are displayed in an alarm list. This alarm list consists of several columns and each line represents a new alarm. The first column shows the machine where the alarm occurred. Furthermore it is indicated when the alarm has occurred and when it has gone again. In addition, an individual error text from a previously defined error catalog is displayed. The user can filter the alarm list either by time or textual criteria. In figure 54 an example of a alarm list in zenon is displayed.

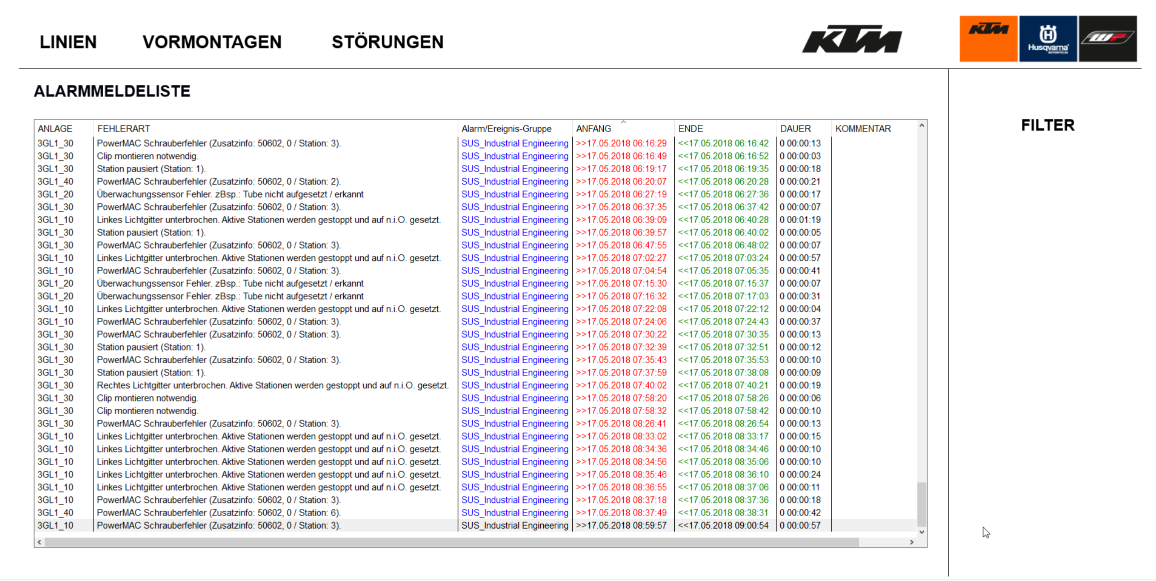

Figure 54.: The alarm list in Suspensin production

Reporting

A simple report for displaying historical orders has been configured. The user can display the orders filtered by calendar week.

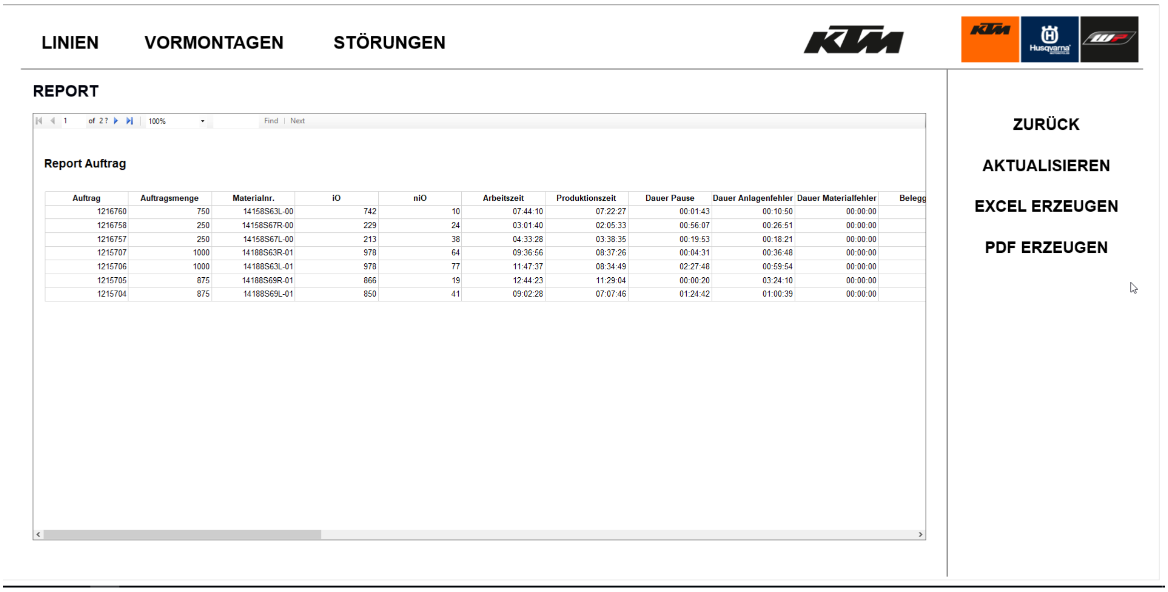

Figure 55.: A report of historic order data

Order master data such as order number, material number, target cycle time and order quantity are displayed. In addition, information about the order such as order duration, occupancy duration, good pieces, bad pieces, occupancy rate, performance level, quality level, availability or efficiency are displayed.

Worker information and interaction system

The worker information system is shown to the worker on a display directly at the workstation. As shown in figure 56, the number of pieces produced and the nominal number of pieces are displayed. Furthermore, the master data of the current order is displayed.
To speed up reaction time when a machine breakdown occurs a worker interaction system was developed. This system works like a digital andon cord. When a failure crops up, the worker has the possibility to inform his foreman in the first place and secondly the person in charge depending on the kind of breakdown.

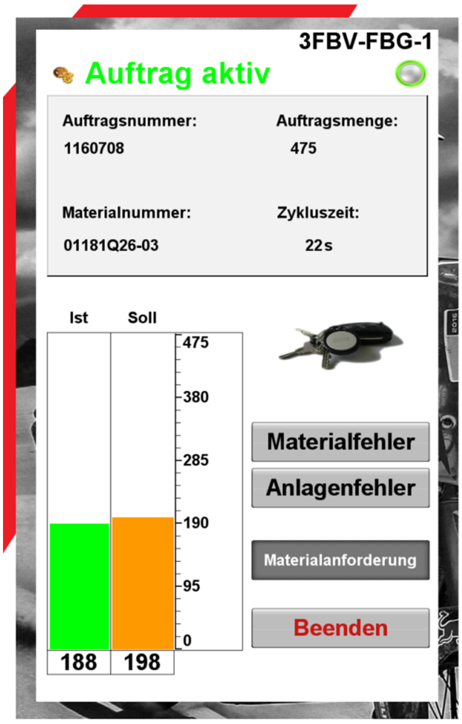

Figure 56.: The Worker Information System which is displayed at the workstations.

5.8. Proof-of-Concept

To evaluate the developed concept, Proof-of-Concepts at three different workstations in production were implemented. Therefore the system was installed at a semi-automated fork assembly line in the suspension division. There were also installations at two work places in the pre-assembly area of the shock absorber production. To show that the concept is also working beyond suspension, one robot cell in the frame division was equipped. The individual implementations are described in more detail below. The adaptation of the equipment was taken over by the in-house special machine construction department. This included design and manufacturing work for the consoles in which the display and RFID were installed. Furthermore, control cabinets for the Line Terminals had to be assembled and installed within the production lines. The proof-of-concept was implemented during a one-month phase and then tested in a one-month period. Furthermore, it is stated which insights could be gained from these proof of concepts and as a result whether this system proves to be suitable or not.

5.8.1. Semi-automated assembly line

The semi-automated assembly line consists of four interlinked workstations and a test stand at the end of the line (cp. section 5.4.1). At the semi-automated assembly line for front fork production, the retrofitting demand was minimal since there were already industrial displays at every work station installed. The only installations were RFID-Readers at the workstations and a line terminal at the beginning of the production line to scan the order sheets and therefore start the orders. The production line can be seen in figure 57. This line was chosen because the entire production machines were developed and built in-house and therefore the interfacing and adaptation of the individual controls can be handled many times easier than with bought-in machines. Another reason to try out the system on these systems is that the production staff repeatedly claim that a large number of screwing errors occur again and again on the system, depending on the product.

Figure 57.: The semi-automated fork assembly line

5.8.2. Manual work station

Furthermore, a manual workstation consisting of two electrical presses was integrated. The need for retrofitting was higher here, as neither displays nor a PLC were available. Thus, a line terminal consisting of a controller, a touchmonitor and a QR-code scanner was retrofitted on the one hand, and two worker terminals consisting of a display and a RFID reader on the other hand.The pulse for the part counter was taken from the two-hand triggers. In figure 58 the manual work place is shown. On the left side the touchmonitor of the line terminal can be seen. The picture also shows the two displays of the worker terminals. The manual workstation was chosen because the request for a duplicate of the workstation was repeatedly made. It is also often noted that the target times are set too low. By recording the data from this workstation, it is to be checked whether the complaints can be justified.

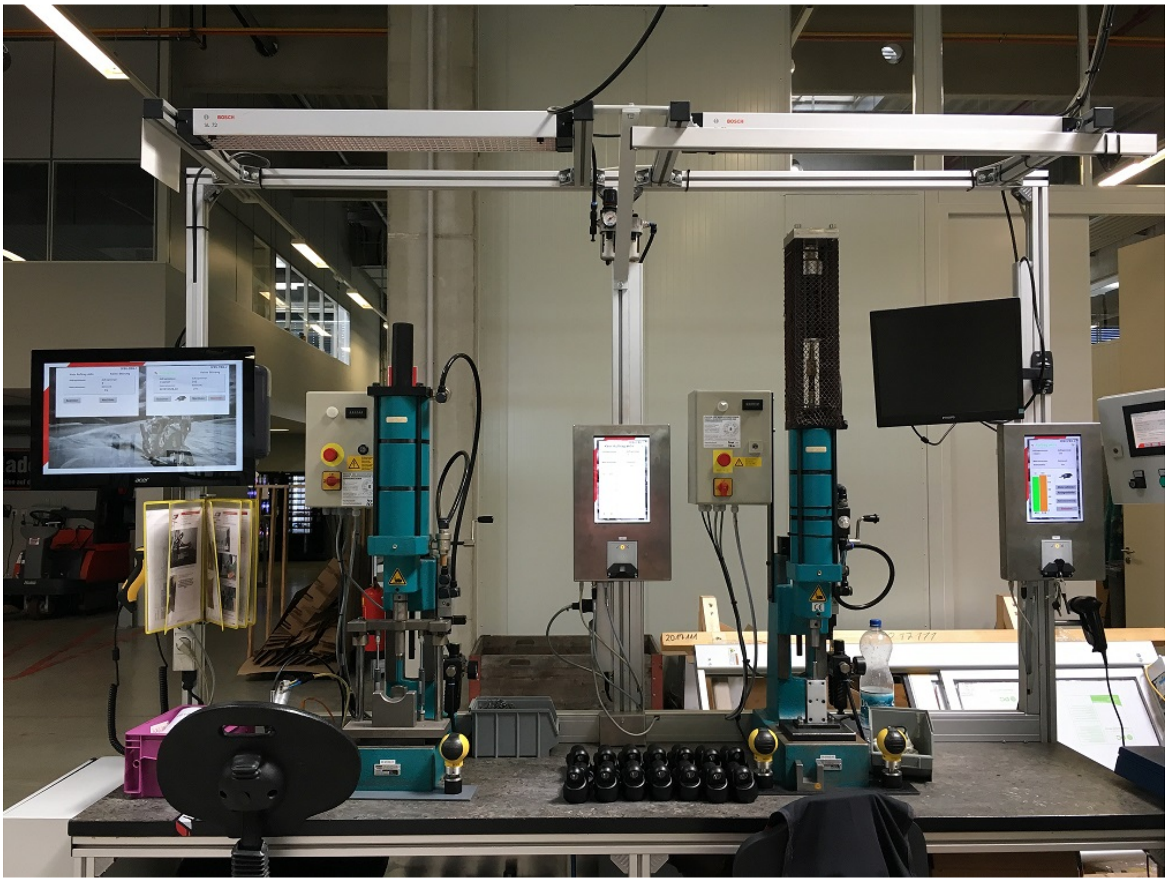

Figure 58.: The manual work place with the line terminal on the left and the two worker terminals.

5.8.3. Robot cell

In order to show that the developed system also works outside the suspension production, a welding robot was equipped in the frame division. A line terminal was installed to start the orders. This was designed in such a way that 4 more robots could be connected to it. A worker terminal was also installed. These two installations can be seen in figure 59. As there were always problems in achieving the required quantities and this was usually attributed to machine downtimes, special attention should be paid to these topics in this installation.

Figure 59.: Line Terminal (right) and Worker Terminal (left) at the welding robot.

5.8.4. Results Proof-of-Concept

This section will show the results of the proof of concept. Furthermore, a verification will be carried out with the requirements raised at the beginning of the project. It is shown which requirements could be completely fulfilled and which cannot yet, but can be fulfilled in the future. Additionally, it is shown for which requirements the system is not suitable or only suitable with considerable effort. Furthermore it is shown whether the system is suitable for a later roll-out in the company.

Semi-automated assembly line

The Proof-of-Concept for this line provided several interesting insights. Due to the automatic recording of errors on the systems, it was possible for the first time to determine how often certain errors actually occur. In order to be able to make a qualified statement, the errors were recorded for the period of the Proof-of-Concept and were then analyzed. As shown in Figure 60, the errors were then grouped according to their error number and displayed in ascending order according to the amount of errors in the diagram. Furthermore, the sum of the time the error was active was displayed in a second bar. Here it has to be said that the total duration is to be enjoyed with caution, since it can be that an error occurs at the end of the shift and this is pending until the beginning of the next shift. In the case of this production line, which is run in 2-shift operation, a falsification can take place here. From this list of errors it could be found out that screwing errors actually occur more frequently with individual products. As a result, a CIP was started to eliminate these problems. Furthermore, it was found that at a certain station the light curtain is often reached into, which results in machine downtime. By training the employee, these problems could also be eliminated.

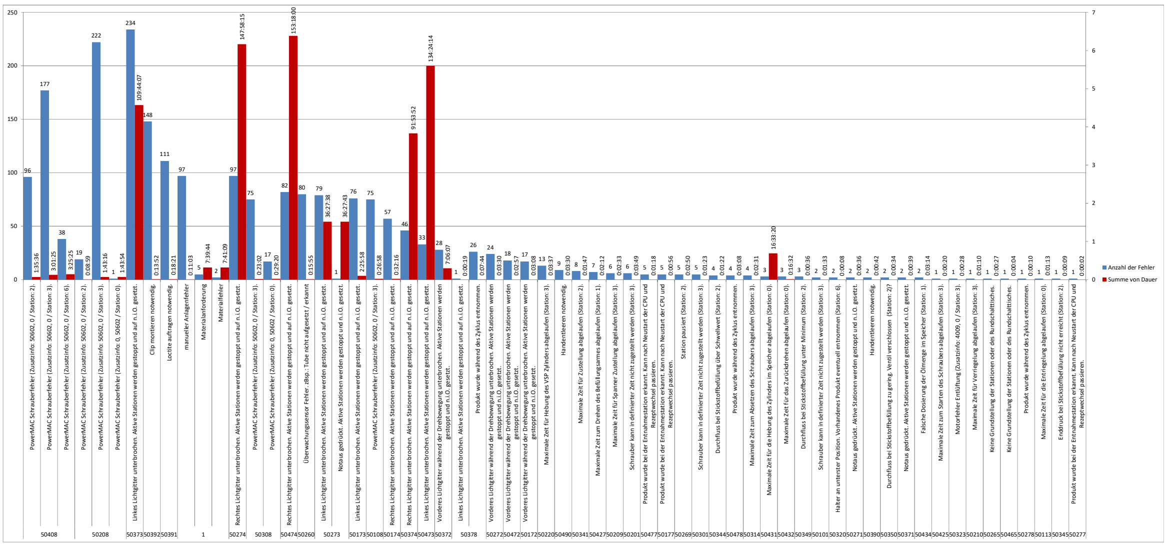

Figure 60.: Errors at the semi automated line in one month time.

With the zenon client, the foremen and management were provided with a good tool for monitoring the production process. The response time could be considerably reduced by sending SMS in case of problems at the machine. Furthermore, information concerning the current order was made available to the worker. The worker was always informed about the progress of the order and could see whether he was in front of or behind the required quantity.

Manual Work Station

Two different insights could be gained at the manual workstation. On the one hand it became clear in the detailed view due to the trend view of the produced quantities that the real work process does not follow the specifications of the work preparation. This phenomen is shown in figure 61 since there are phases where the worker is producing faster than required and other phases where he is producing nothing. This implies that he doesn't follow the instructions given with the work plan. Furthermore, due to the cycle time distribution, it could be shown that the planned times largely correspond well with the actual times.

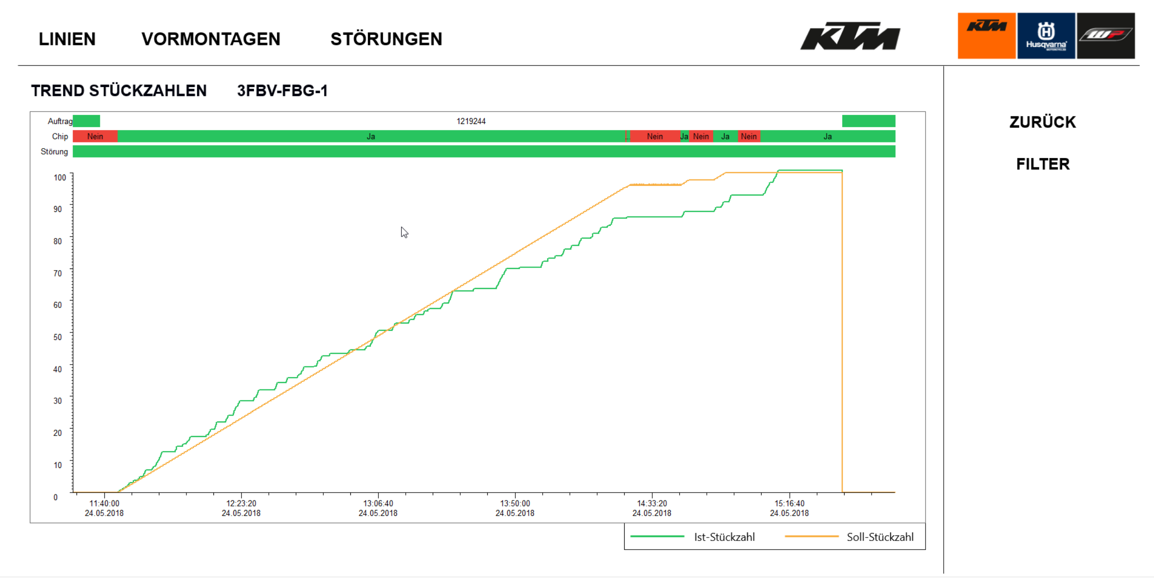

Figure 61.: Trend view of the produced quantity at the manual work station

Robot Cell

At the welding robot workstation, a good assistance tool was made available not only to the managers but also to the worker. The worker benefited from the reduced reaction times, as he only had to press the button on the touch screen in case of malfunctions. Furthermore, he was always informed about his current performance by the visualized comparison of the actual quantity to the target quantity. By accessing the Zenon client, the foreman could check how the performance was in the night shift or late shift. Furthermore, it could be checked whether the required quantities could really not be achieved due to robot malfunctions. Furthermore, by checking the cycle time report as shown in figure 62, it was possible to determine whether the worker at the workstation produces products at regular intervals.

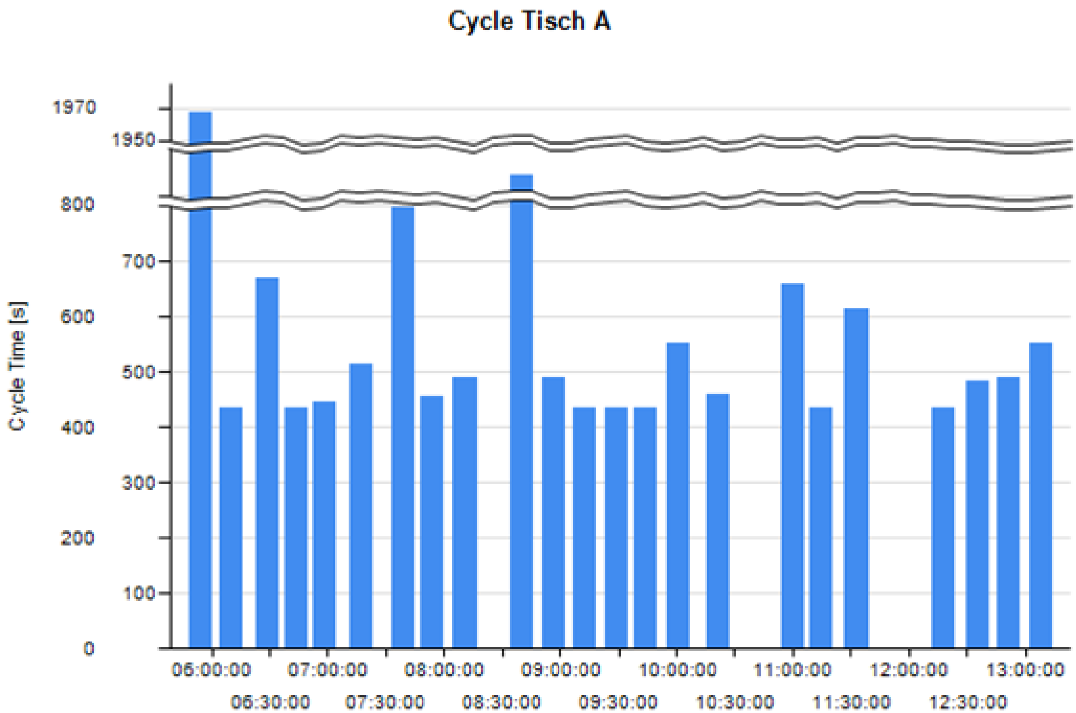

Figure 62.: Cycle Times report of the welding robot

Fulfillment of the Requirements

All must-have functional requirements could be fulfilled, which is the basic prerequisite for a further progress in this system. The functional requirements FR-21 to FR-27 contain the requirements regarding reports on various recorded information. As the preparation of the reports takes a lot of time, only reports on the historical orders (FR-22) and the recorded faults (FR-24) were created. Based on the knowledge gained in the implementation of these requirements, it can be assumed that the creation of further reports will also work. The storage of product-specific data (FR-28) could not be tested, since only systems were connected, where the handled products have no unique identification. The storage of process data (FR-29) and machine data (FR-30) has also not yet been implemented in this proof of concept. However, the archiving functions of the software promise good possibilities to record and compress such data. However, what has been saved are the alarm data (FR-31) and order specific data (FR-32). By implementing these two requirements, event-driven and cyclical archives could be tested. Requirements regarding providing information with the worker assistance system (FR-33 and FR-34) could be implemented and were appreciated by the workers.
The requirements regarding the notification of relevant persons in the event of a breakdown via text message (FR-35 to FR-39) could be implemented. However, it should be noted here that in the beginning it was not clearly defined what is a notification requiring malfunction, as a result too many texts were initially sent. Regular corrections made it possible to define the types of errors that require text message communication and the system was accepted. The possibility to report the produced quantity from the workstation into the ERP directly (FR-50) was not implemented, but will be possible with this system. Requirement FR-51 and FR-52 are hardly to implement with this system, but other solutions are considered. The link between the batch number and the order number (FR-53) can be easily implemented with this system. FR-54 (production planning) and FR-57 (personal planning) will not be possible with this system. FR-59 (central receipe management) could again be well implemented with this system.

5.9. Feedback

In accordance with the goal of finding a system to acquire production data and further use them in a loopback to control production, requirements of different stakeholders have been collected. First of all the situation within the production has been analysed. A closer look on production data recording has been done. Secondly the target state has been established and later the use cases for a production data acquisition system have been elicited. In a technology screening several software systems have been compared and the advantages and disadvantages of each system are demonstrated. To find the best fitting system for the company a Value Benefit Analysis was performed. Based on this knowledge a concept for production data acquisition has been developed. To verify the designed system, a proof of concept has been made on three different production equipments at WP Performance Systems. It was shown that the applied methodology is suitable for the selection and implementation of a production data acquisition system. In particular, it could be shown that the V-model is suitable as a process model for the design, implementation and verification of such a system. It could be demonstrated that this system is suitable for the required functionalities.

6. Conclusion and Outlook

Beginning from a general problem-description, the fundamentals of MES were first theoretically worked out. It was discussed what functionalities a MES can offer and what advantages the implementation of MES can have.
In order to carry out a structured integration of a PDA system, the techniques of Requirements Engineering were first elaborated theoretically. Basic definitions such as the system term, stakeholders and types of requirements were first made so that a typical RE process can then be outlined. Various techniques for gathering requirements were presented.
In order to provide a structured framework for system development, the theoretical principles of Systems Engineering were also covered. For this purpose, the philosophy of Systems Engineering with the two parts systems thinking and process model was fundamentally addressed. Furthermore, the problem-solving process consisting of the elements system design and project management was presented in more detail. The focus here was on system design with the two components architecture design and concept design. As a modelling language for system development, the SysML notation was briefly discussed.
The theoretical principles have been applied in a case study at a partner company. Based on a detailed problem analysis, requirements for an production data acquisition system were determined using Requirements Engineering techniques. Afterwards, five tools from the set of software solutions were examined in more detail. In order to make a decision, these were subjected to a value-benefit analysis.
Thus a potential solution was selected, but in order to test its practical suitability, the system was tested in a Proof-of-Concept on three different production units.

The result of this work for WP Performance Systems is a concept of a system for Production Data Acquisition. As we saw in chapter 5 the system is feasible for all four different production divisions. This is a very important aspect, since WP is the largest production

company in the KTM Group and it is not excluded that new products will be included in the production portfolio in the future. After the proof of concept, the statement can be made that this concept is suitable for all production areas at WP and that it can safely be rolled out at all production areas.
However, this system forms only the basis for further evolutionary steps towards *Industry 4.0*. For example, the recorded data can be used for big data analysis. The broad approach to system selection enables the later implementation of numerous use cases such as new approaches to condition monitoring or predictive analytics, to mention just two of them. The panels at the workstations are prepared for further requirements like the digital workplace for displaying work instructions and order information. It should be borne in mind that this is only an initial concept for the collection of production data, and a lot of work is still needed to achieve a well-engineered and sustainable system.
Furthermore, it could be shown that the combination of Requirements Engineering and Systems Engineering is definitely a possible way to select, develop and implement such a system. It has also turned out that the systematic identification of requirements is an advantage for the structured implementation of such a system.

References

Christos Alexakos and Athanasios Kalogeras. Exposing MES functionalities as enabler for cloud manufacturing. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, 2017. doi: 10.1109/WFCS.2017.7991966.

Emrah Arica and D.J. Powell. Status and Future of Manufacturing Execution Systems. 2017.

Armin Roth. *Einführung und Umsetzung von Industrie 4.0*. Springer Gabler, Leinfelden-Echterdingen, 2016. ISBN 978-3-662-48504-0.

Helmut Balzert. *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. Spektrum Akademischer Verlag, Heidelberg, 2009. ISBN 978-3-8274-1705-3.

Wilhelm Bauer, Sebastian Schlund, Dirk Marrenbach, and Oliver Ganschar. Industrie 4.0 - Volkswirtschaftliches Potenzial für Deutschland, 2015.

Thomas Bauernhansl, Michael ten Hompel, and Birgit Vogel-Heuser. *Industrie 4.0 in Produktion, Automatisierung und Logistik*. Springer Vieweg, Stuttgart, 2014. ISBN 978-3-658-04681-1.

Kent Beck, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for Agile Software Development. 2001.

Stafford Beer. *Kybernetik und Management*. S. Fischer, Frankfurt am Main, 1963. ISBN 978-3-1010-6801-8.

Barry Boehm. A spiral model of software development and enhancement. *IEEE Computer Society Press Los Alamitos, CA, USA*, 21(1):61–72, 1988.

Gianpiero Brunetti. Architecture and implementation of a MES system in a large scale steel plant: Severstal cherepovets success story. *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016 - Proceedings*, pages 1077–1081, 2016.

Rafal Cupek, Adam Ziebinski, Lukasz Huczala, and Huseyin Erdogan. Agent-based manufacturing execution systems for short-series production scheduling. *Computers in Industry*, 82:245–258, 2016.

Cihan H. Dagli and Nil Kilicay-Ergin. System of Systems Architecting. In *SYSTEM OF SYSTEMS ENGINEERING*. John Wiley & Sons Inc., Hoboken, NJ, 2009. ISBN 9780470195901.

Philipp Dickmann. *Schlanker Materialfluss*. Springer Berlin Heidelberg, Grafing bei München, 2014. ISBN 9783662448687.

George T Doran. There's a S.M.A.R.T. way to write managements's goals and objectives., 1981. ISSN 00251895.

Thomas Epping. *Kanban für die Softwareentwicklung*. Köln, 2011. ISBN 9783642114373.

Guillaume Finance. SysML Modelling Language explained, 2010.

S. Friedenthal, A. Moore, and R. Steiner. *A Practical Guide To SysML: The Systems Modeling Language*. Morgan Kaufamnn OMG Press, Waltham, MA, 2015. ISBN 9780123743794.

Sanford Friedenthal, Alan Moore, and Rick Steiner. OMG Systems Modeling Language (OMG SysML^{TM}) Tutorial. 2009.

Thorsten Gerberich. *Lean oder MES in der Automobilzulieferindustrie*. Gabler Research, Chemnitz, 2011.

M. Glinz and R.J. Wieringa. Stakeholders in Requirements Engineering. *Software, IEEE*, 24 (2):18–20, 2007. ISSN 0740-7459. doi: 10.1109/MS.2007.42.

Martin Glinz. On Non-Functional Requirements. *Proceedings - 15th IEEE International Requirements Engineering Conference, RE 2007*, page 361, 2007. ISSN 1090-705X. doi: 10.1109/RE.2007.45.

Martin Glinz. Requirements Engineering I, 2009.

Uwe Götze. *Investitionsrechnung - Modelle und Analysen zur Beurteilung von Investitionsvorhaben*. Springer Gabler, Chemnitz, 2014. ISBN 978-3-642-54621-1.

Reinhard Haberfellner, Ernst Fricke, Olivier de Weck, and Siegfried Vössner. *Systems Engineering - Grundlagen und Anwendung*. Zürich, 13 edition, 2015. ISBN 978-3-280-04068-3.

Berthold Heinrich, Petra Linke, and Michael Glöckler. *Grundlagen Automatisierung*. Springer Vieweg, Wiesbaden, 2017. ISBN 978-3-658-17581-8.

Petri Helo, Mikko Suorsa, Yuqiuge Hao, and Pornthep Anussornnitisarn. Toward a cloud-based manufacturing execution system for distributed manufacturing. *Computers in Industry*, 65(4):646–656, 2014.

Walter Huber. *Industrie 4.0 in der Automobilproduktion*. Springer Vieweg, Haar, 2016. ISBN 9783658127312.

Institute of Electrical and Electronics Engineers. IEEE Standard 29148:2011. Technical report, Institute of Electrical and Electronics Engineers, 2011.

Jürgen Kletti. *MES - Manufacturing Execution System*. Springer Vieweg, Mosbach, 2015. ISBN 978-3-662-46901-9.

Tobias Kollmann and Holger Schmidt. *Deutschland 4.0 - Wie die Digitale Transformation gelingt*. Springer Gabler, Köln/Frankfurt, 2016. ISBN 978-3-658-11981-2.

Jörg Kühnapfel. *Nutzwertanalysen in Marketing und Vertrieb*. Springer Gabler, Ludwigshafen am Rhein, 2014. ISBN 9783658055080.

Jakob Lewandowski. *Produktionsplanung und -steuerung in mittelständischen Unternehmen unter besonderer Berücksichtigung von Manufacturing Execution Systems*. Wien, 2011.

Philipp Louis. *Manufacturing Execution Systems -Grundlagen und Auswahl*, volume 11. Gabler Edition Wissenschaft, Marburg, 2009. ISBN 978-3-8349-1018-9.

MESA International. MES Explained: A High Level Vision. *MESA Internacional*, (6):23, 1997.

Heiko Meyer, Franz Fuchs, and Klaus Thiel. *Manufacturing execution systems : optimal design, planning, and deployment*. McGraw-Hill, München, 2009. ISBN 978-0-07-162602-6.

Felix Georg Müller, Markus Bressner, David Görzig, and Thomas Röber. Industrie 4.0: Entwicklungsfelder für den Mittelstand, 2016.

Robert Obermaier. *Industrie 4.0 als unternehmerische Gestaltungsaufgabe*. Springer Gabler, Passau, 2016. ISBN 978-3-658-08164-5.

Helmuth Partsch. *Requirements-Engineering systematisch*. Number 1. Springer, Ulm, 2014. ISBN 978-3-642-05357-3.

Roman Pichler. *Scrum - Agiles Projektmanagement erfolgreich einsetzen*. dpunkt, Heidelberg, 2008. ISBN 978-3-89864-478-5.

Klaus Pohl. *Requirements Engineering - Grundlagen, Prinzipien, Techniken*. dpunkt-Verlag, Heidelberg, 2. edition, 2008. ISBN 9783898645508.

Klaus Pohl and Chris Rupp. *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam-Foundation Level – IREB compliant,*, volume 53. Rocky Nook Inc, Essen, Nürnberg, 2015. ISBN 9788578110796.

Ali K Raz, C Robert Kenley, and Daniel A DeLaurentis. System architecting and design space characterization. 2018.

Chris Rupp. *Requirements-Engineering und -Management*. Hanser, München, 6 edition, 2014. ISBN 9783446443136.

August-Wilhelm Scheer. *Computer integrated manufacturing: CIM*. Springer, Berlin, 4 edition, 1990.

Bianca Scholten. *MES Guide for Executives: Why and How to Select, Implement, and Maintain a Manufacturing Execution System*. International Society of Automation, Rosmalen, 2009. ISBN 978-1-936007-03-5.

OMG SysML. OMG Systems Modeling Language. *INCOSE Intl. Symp*, 2017. URL `http://www.omgsysml.org/INCOSE-OMGSysML-Tutorial-Final-090901.pdf`.

The International Society of Automation. ISA-95.00.03-2005, Enterprise Control System Integration, Part 3: Activity Models of Manufacturing Operations Management, 2005.

Pedro Daniel Urbina Coronado, Roby Lynn, Wafa Louhichi, Mahmoud Parto, Ethan Wescoat, and Thomas Kurfess. Part data integration in the Shop Floor Digital Twin: Mobile and cloud technologies to enable a manufacturing execution system. *Journal of Manufacturing Systems*, 2018. ISSN 02786125. doi: 10.1016/j.jmsy.2018.02.002.

Blatt 1:2016-10 VDI 5600. Fertigungsmanagementsysteme (Manufacturing Execution Systems - MES). Technical report, Verein Deutscher Ingenieure, 2016.

Birgit Vogel-Heuser, Thomas Bauernhansel, and Michael Ten Hompel. *Handbuch Industrie 4.0 Bd.4*. Springer Vieweg, Berlin, 2016. ISBN 978-3-662-53253-9.

Birgit Vogel-Heuser, Thomas Bauernhansl, and Michael ten Hompel. *Handbuch Industrie 4.0 Bd.1*. Springer Reference Technik, 2017. ISBN 9783662452783.

Wolfgang Vorraber, Dietmar Neubacher, Birgit Moesl, Julia Brugger, Sigmar Stadlmeier, and Siegfried Voessner. UCTM - An Ambidextrous Service Innovation Framework: A Bottom-up Approach to Combine Human- and Technology-centered Service Design, 2018.

D. Wee, R. Kelly, J. Cattel, and M. Breunig. Industry 4.0 - How to navigate digitization of the manufacturing sector. *McKinsey Digital*, pages 1–62, 2015.

Tim Weilkiens. *Systems Engineering with SysML/UML - Modeling, Analysis, Design*. Number March 2000. dpunkt verlag, Heidelberg, 2006. ISBN 978-0-12-374274-2.

Benedikt Weissenberger, Stefan Flad, Xinyu Chen, Susanne Rösch, Tobias Voigt, and Birgit Vogel-Heuser. Model driven engineering of manufacturing execution systems using a formal specification. *Ieee Etfa*, 2015.

Karl E Wiegers and Beatty Joy. *Software Requirements*. Microsoft Press, Redmond, 3rd edition, 1999. ISBN 978-0-7356-7966-5.

Petra Winzer. *Generic Systems Engineering*. Springer Vieweg, Wuppertal, 2013. ISBN 9783642303647.

Frank Witte. *Testmanagement und Softwaretest*. Springer Vieweg, Landshut, 2016. ISBN 978-3-658-09963-3.

A. Key Performance Indicators - KPI

In the following section Key Performance Indicators are developed to supervise and control the production process. The developed KPIs are the outcome of a workshop with the stakeholders in the company. Since production at WP is based on batches, all KPIs are calculated for one order. The underlying time model is shown in figure 63. The order time is subdivided in work time and planned down time. Worktime is the time of the planned shifts during an order. The planned downtimes represent the times during which no shifts are planned, but also the fixed break times per shift.

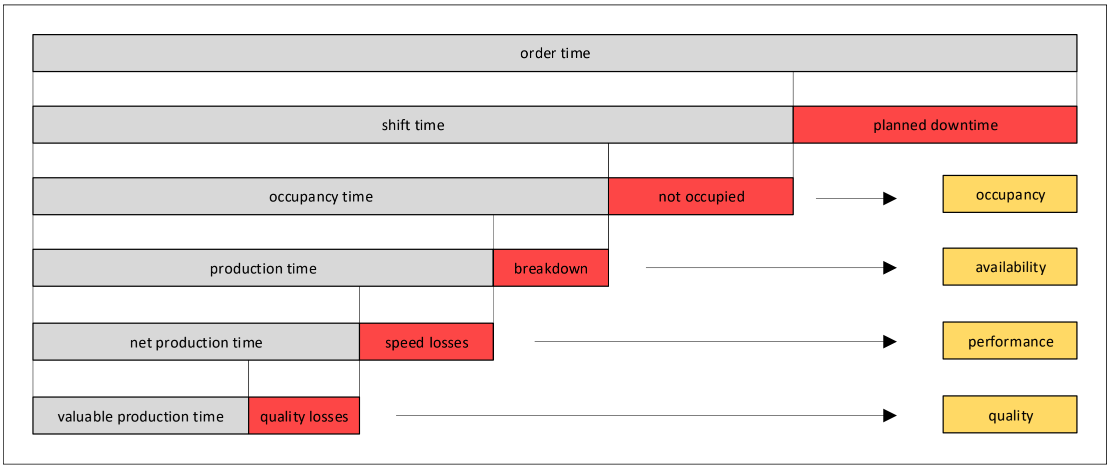

Figure 63.: time model for KPIs (authors own illustration)

Occupancy time is the time the worker is at the machine. The production time is the occupancy time minus the disruption times. Disruption times can be caused by machine errors or material errors. The net production time is calculated from the produced quantity and the given production time per unit. The difference between production time and net production time results in the speed losses. If the speed losses are negative, this means that the worker is producing faster than prescribed by the work plan. The value-adding production time is also a calculated size. This is calculated from the produced yield and the production time per unit.

Occupancy:

The degree of occupancy is used to express how well the workplace was occupied during the planned working time.

$$Occupancy[\%] = \frac{Occupancy\ Time}{Working\ Time\ (Planned\ Shifts)}$$

Reasons for not occupying a work station can be, for example, additional breaks of the worker. It may also be that the worker is removed from the work station and has to work on another order at another station.

Availability:

Availability is a measure of the disruption of a production process caused by machine errors. If no machine errors have occurred, the key figure is 100

$$Availability[\%] = \frac{production\ time}{occupancy\ time}$$

Performance:

The degree of performance indicates how well the worker complies with the given cycle time. If you work faster than specified, this figure can also be greater than 100%.

$$Performance[\%] = \frac{net\ production\ time}{production\ time} = \frac{produced\ quantity * cycle\ time}{production\ time}$$

Quality:

The degreee of quality is a measure of the amount of good parts that were produced.

$$Quality[\%] = \frac{goodquantity}{produced\ quantity} = \frac{value\ added\ production\ time}{net\ production\ time}$$

Batch Efficiency:

The Batch efficiency is inspired by the overall equipment effectiveness (OEE) in relation to a planned period of time for batch production. It is the product of the key performance indicators occupancy, availability, performance and quality.

$$Batch\ efficiency[\%] = Occupancy * Availability * Performance * Quality$$

The Batch efficiency can, with the same argumentation as before in terms of performance, become greater than 100%.

Line efficiency:

To calculate a line based efficiency rate, a few considerations have to be done. First of all it is important to know which workstations within the line are included in the specific manufacturing order. WPs production philosophy consists of flexible production entities. The production lines can be operated with a variable quantity of workers. To compute the efficiency of a manufacturing lines, it has to be known how much work force has to be put in the system production line. In return you have to know how much good parts are produced in the same period of time. For a better understanding the calculation of the line efficiency is shown in figure 64.

$$Line\ efficiency[\%] = \frac{good\ parts\ at\ end\ of\ line * given\ cycle\ time}{\sum production\ time\ of\ occupied\ work\ stations}$$

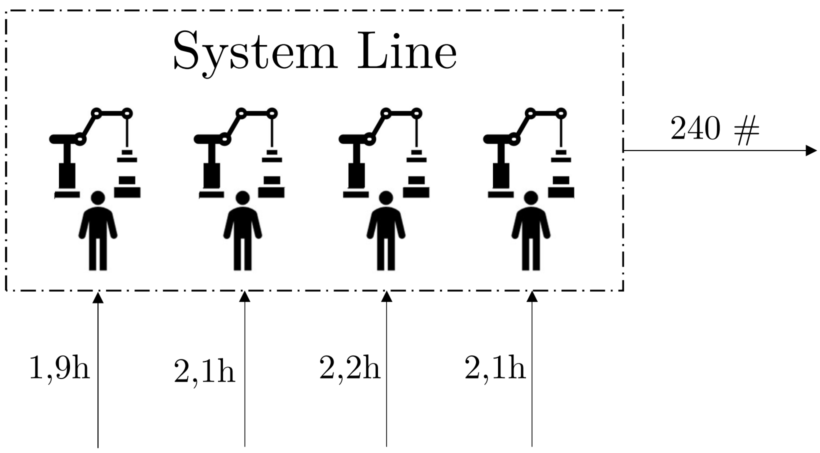

Figure 64.: Line efficiency calculation (source: authors own illustration)

Scrap rate:

In order to see at one glance if production is running smoothly, the scrap rate for each line has to be calculated. In the division suspension the scrap rate at the test beds is from most interest. So the scrap rate at the test bed counts for the scrap rate of the line.

$$Scrap\ rate[\%] = \frac{bad\ parts}{total\ parts}$$