Gerald Wagner, BSc

Management information on mobile platforms: Requirements engineering and implementation

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Software Development and Business Management

submitted to

Graz University of Technology

Supervisor

Ass.Prof. Dipl.-Ing. Dr.techn. Gerald Lichtenegger
Dipl.-Ing. Christoph Wolfsgruber

Institute of Engineering and Business Informatics

Head: Univ.-Prof. Dipl-Ing. Dr.techn. Siegfried Vössner

Graz, February 2016

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 dissertation.

Date

Signature

Abstract

Over the last decades the amount of data generated during production and manufacturing processes in industry increased markedly due to higher levels of automation and the emergence of the internet. This data represents a huge potential for optimization and progress which in many cases is not yet fully exploited. The term "Industry 4.0" describes novel technologies developed to manage and utilize this information. In this thesis the aim was to develop an information system for the Audi Hungaria Motor production plant in Györ to provide the most important production data, also called key performance indicators to the upper management. Therefore a native iOS prototype was implemented, that visualizes all relevant data of the production plant such as production output as well as potential deviations and subsequently taken measures. To avoid common problems during software engineering projects the technique of requirements engineering was applied which includes a thorough planning phase that clearly defines the desired system qualities. The resulting prototype was tested successfully on site and based on it a solution for productive use was developed externally and rolled out.

Kurzfassung

Im Laufe der letzten Jahrzehnte hat sich die Datenmenge, die während Produktions- und Herstellungsprozessen generiert wird, durch höhere Automatisierungsgrade und das Aufkommen des Internets deutlich erhöht. Diese Daten beinhalten ein großes Potential für Optimierung und Fortschritt, welches in vielen Fällen noch nicht voll ausgeschöpft wird. Der Begriff „Industrie 4.0" beschreibt neue Technologien die entwickelt wurden, um diese Informationen verwalten und anwenden zu können. Das Ziel dieser Arbeit war die Entwicklung eines Informationssystems für die Audi Hungaria Motor Produktionsstätte in Györ, welches die wichtigsten Produktionsdaten, auch Key Performance Indikators genannt, für das obere Management bereitstellt. Dafür wurde ein nativer iOS Prototyp implementiert welcher alle relevanten Daten der Produktionsstätte, wie beispielsweise Produktzahlen aber auch mögliche Abweichungen sowie die dafür getroffenen Maßnahmen visualisiert. Um häufig auftretende Probleme während der Entwicklung von Softwareprojekten zu vermeiden, wurde die Technik Requirements Engineering angewandt, welche eine fundierte Planungsphase verwendet, in der die gewünschten Systemeigenschaften klar definiert werden. Der daraus resultierende Prototyp wurde erfolgreich vor Ort getestet und darauf basierend wurde extern eine Lösung für die produktive Umgebung entwickelt und ausgeliefert.

Acknowledgments

At this point, I would like to thank all people that supported and motivated me during this thesis.

Special thanks go to Ass. Prof. Dipl.-Ing. Dr.techn. Gerald Lichtenegger and Dipl.-Ing. Christoph Wolfsgruber from the Institute of Engineering and Business Informatics as well as to Univ.-Prof. Dipl-Ing. Dr.techn. Siegfried Vössner, for the opportunity to work in this interesting project and for the helpful suggestions, organizational support and constructive criticism along the way.

I also want to thank my partner Sandra for all the unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Furthermore, I would like to thank my mother who supported all my decisions and enabled my studies at the TU Graz.

Contents

List of Figures

List of Tables

1 Introduction

Management has generally the task to coordinate, organize, plan and forecast work within an organization.[1] To fulfill those tasks and activities, managers need information in a timely manner to support them in making correct decisions. In this thesis we will take a look on management information systems to allow key staff access to the correct and up-to-date information. Recent developments in hard and software refashion the way production plants are constructed and organized (see figure 1.1).

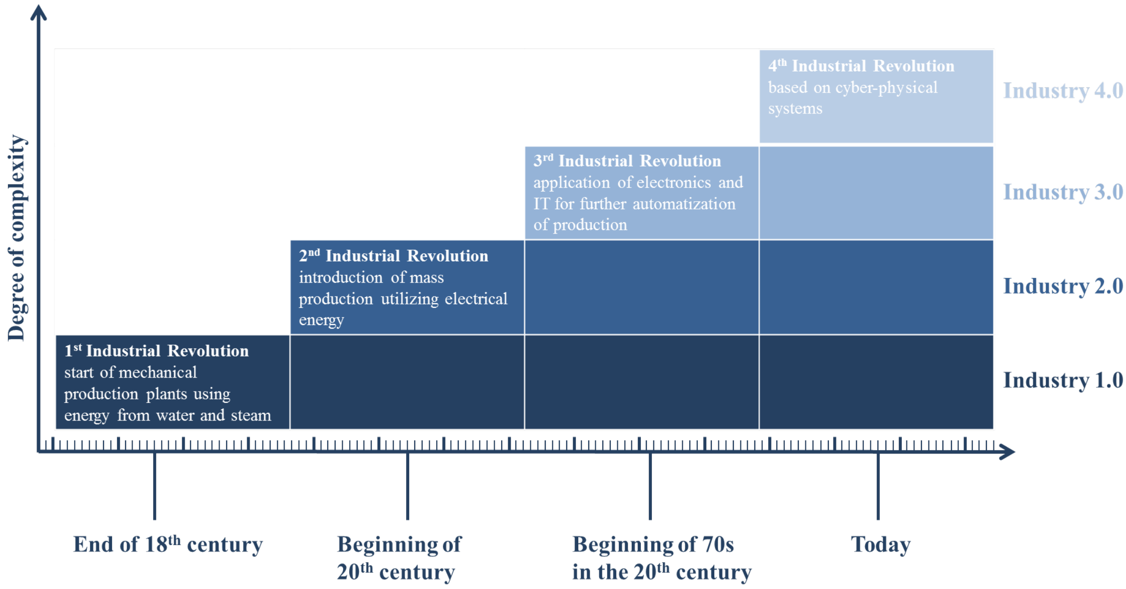


Figure 1.1: Industry 4.0 and its preceding phases based on Wernher Behrendt (2013)

Along with the availability of interlinked machines, big data and advanced processing tools, new possibilities arise for management. These developments in the production sector can be summarized under the term industry 4.0 (see figure 1.1). With modern information technology, the amount of data generated and stored is increasing enormously and enables decision makers to access a great amount of

[1] Koontz (1961) and Fayol and Coubrough (1930)

data within seconds. The challenge addressed in this masters thesis was to filter out and transform the most important pieces of data and to provide them on a mobile platform as a management information system.

1.1 Initial situation and motivation

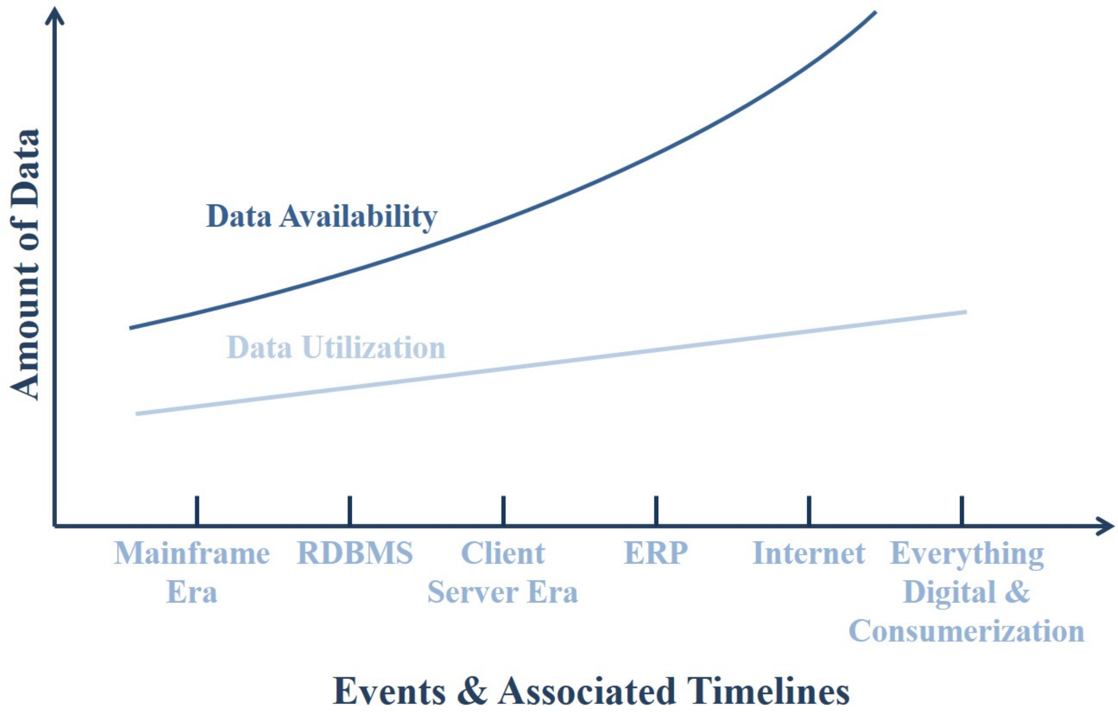


Figure 1.2: Utilization of data over time (Mohanty, Jagadeesh, and Srivatsa, 2013). *Relational database management systems (RDBMS), enterprise resource planning (ERP)*

Audi Hungaria Motor produces over 2 million engine units per year at the production plant in Györ, Hungary. One challenge the plant encounters is to cope with the data flood that is generated and aggregated during routine operation. Modern interconnected production facilities help optimizing productivity, but in the process they generate exceptional amounts of data. Today this data is stored and archived but the true potential is not yet fully exploited (figure 1.2). Utilizing stored data was a problem in the early mainframe era and still is a challenge today. (Barkin and Dickson, 1977; Mohanty, Jagadeesh, and Srivatsa, 2013)

The key task of this work was to filter and condense this information and generate relevant visual output. The goal was to select significant data, transform it into key performance indicators (KPIs), and provide them in a timely manner to decision makers. Therefore, the engineering of a management information system (MIS) for a mobile platform was proposed.

Software and system engineering ventures have a long history of project failures due to unmet deadlines and exceeded budgets. Studies indicate that 60% of all errors made in software engineering projects occur in the design and planning phase. These errors manifest themselves in later project stages and lead to extraordinary project cost escalations. The NASA published a report highlighting the respective costs of fixing errors, depending on the project phase during which they occurred (figure 1.3).

Source	Phase Requirements Issue Found			
	Requirements	Design	Code	Test
[Boehm, 1981]	1	5	10	50
[Hoffman, 2001]	1	3	5	3
[Cigital, 2003]	1	3	7	51
[Rothman, 2000]		5	33	75
[Rothman, 2000] Case B			10	40
[Rothman, 2000] Case C			10	40
[Rothman, 2002]	1	20	45	250
[Pavlina, 2003]	1	10	100	1000
[McGibbon, 2003]		5		50
Mean	**1**	**7.3**	**25.6**	**177**
Median	**1**	**5**	**10**	**50.5**

Figure 1.3: Normalized costs of fixing errors depending on the project phase during which they were identified (Stecklein et al., 2004)

To support the design and implementation as well as the validation process requirements engineering was used.

1.2 Scope and structure of the thesis

This masters thesis is embedded into a project of the Institute of Engineering and Business Informatics in cooperation with Audi Hungaria Motor to optimize the existing internal reporting system. A homogeneous solution should be implemented which is applicable from the shop floor up to the top management at the production plant in Györ. As of now, most of the processes needed for generating daily, weekly and yearly reports are automated. Nevertheless some of the steps, like feeding information from the shop floor to the quality assurance management system, still require manual contribution. The goal of this thesis was to close the missing links in the automation process and further improve the already existing reporting systems. Additionally, as part of this work, a software solution was implemented, that provides access to all information needed by the management. This master thesis focuses on the topic of mobile management information system within the project scope.

Therefor the thesis is divided into four main parts (figure 1.4): Theoretical background (IS, KPIs, RE, and data visualization), applied requirements engineering, design of the proof of concept (PoC) as well as the evaluation.

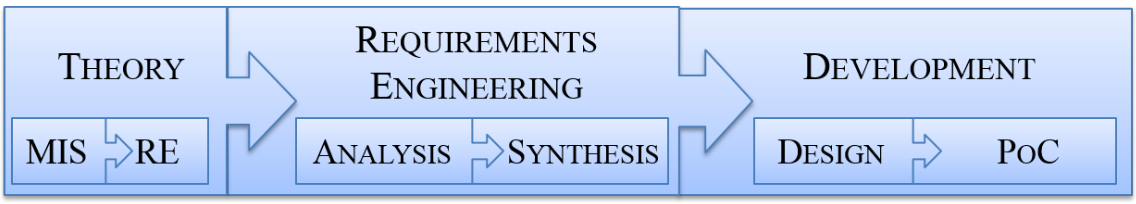


Figure 1.4: Thesis structure

In the first part, the status quo as well as the history of information systems will be elaborated. Furthermore, current architectures and implementations of IS will be presented. To conclude the theoretical background of MIS, benefits and drawbacks of IS will be discussed and current developments will be presented.

The second part of the theory will present the requirements engineering technique used for the practical part. First, the motivation behind the approach will be highlighted and the development of the field will be discussed. Subsequently, the theory behind the approach is explained and recent developments will be presented. Finally the practical implementation used in the practical part of this thesis will be stated, namely the SOPHIST approach.

The second overall part will present the applied requirements engineering. Within this chapter the current processes and systems in place will be **analysed**, followed by the **synthesis**. During the synthesis, which builds a central part of the RE process, potential improvements and new functions will be documented in the form of use cases. The use cases build the foundation for the structured documentation of the requirement, needed for the next step: The **design** of the proof of concept implementation.

Within the design phase possible solutions fitting the description of the use cases will be drafted. These solutions will be compared and evaluated for their feasibility. After considering all options, a design is chosen for the **proof of concept** implementation.

In the proof of concept section the implemented solution including all features will be presented. Along with the implementation, also a validation of all demanded requirements will be performed. The application will be tested in real world scenarios and further improvements will be noted for the requirements management.

2 Information Systems

In this chapter, the established terms and definitions related to management information systems (MIS) will be presented and further a terminology for this thesis will be defined. First, the term management information will be elaborated and the historical development as well as the current state will be discussed.

2.1 Definitions

2.1.1 Information

Every company produces data during their daily business operations. This data is the raw material for the generation of information. Repeatedly, data and information are used synonymously, although they have two different meanings. To draw a distinct line between the two terms, the following definitions will be used:

Data describes the values of attributes or events. These values are then stored according to a fixed predefined code. Facts can be determined by reading, observing, calculating or similar automated operations. (Lucey, 2005)

In contrast to data, information builds on data and adds additional aspects by means of representation leading to the definition.

Information is based on data which is transformed into a new form of representation. This information is intended to support a person in a decision making process. (Lapiedra, Alegre, and Chiva, 2011)

As hinted by the term *information* systems, such systems rely on processed data and therefore, information during operation. Providing and transforming this information is of importance, as business decisions will be based upon the information that is provided to the IS.

Characteristics of information To be of value to business, gathered information needs to meet certain quality requirements considering the following attributes. (Lapiedra, Alegre, and Chiva, 2011):

- Relevance
- Accuracy
- Completeness
- Source trustworthiness
- Exact timing
- Detail
- Comprehension

All informations feeded into the target system must be validated by the listed aspects.

Classification of information Information can be classified by different parameters, which provide important indicators by means of trustworthiness and quality for the further processing and usage: (Lucey, 2005)

- Source: The origination of the information, e.g. internal, external sources
- Nature: Information can be quantitative, qualitative, formal or informal
- Level: By means of level in the company hierarchy: for example strategic, tactical, operational
- Time: Information can be from the past, the present or predictions for the future
- Frequency: Can be continuous (real time), hourly, daily, monthly, annually
- Use: Gathered information can be used for planning, controlling, decision making or record keeping
- Form: Written, oral, visual
- Occurrence: Planned intervals, occasional, on demand
- Type: detailed, summarized, aggregated, abstracted

Throughout this thesis the classification will help to identify possible applications for information as well as potential conflicts.

2.1.2 Management

The term *management* has been defined in various forms in specialist literature. Within this thesis we will present and use a concise definition in the fits best in the context of modern information systems.

Definition (Bower and Gilbert, 2007)

> The organization and coordination of the activities of a business in order to achieve defined objectives.

Managers have the assignment to coordinate activities and tasks in order to reach the goals set by the top management of a business. The way managers make decisions plays a vital role for the development and success of a company. Therefore, we will take a look on the way decisions are made and which steps are involved. (Bower and Gilbert, 2007)

2.1.3 Decision-making process

When analyzing an abstract and simplified decision-making process, two distinct steps can be identified: decision-making process and the resulting action step.

The part of this process that is most important for this thesis is the gathering of information prior to the decision-making process. As illustrated in figure 2.1, the sole input for an ideal process is information. In order to make correct and accurate decisions, regardless at which level in an business operation, it is essential to have precise and sufficient information. Providing this information is a complex task and can involve the utilization of multiple data sources.

Figure 2.1: Simplified decision-making process according to James A O'Brien (1999)

2.1.4 Information systems

Information systems, as well as management information systems in detail, are widely described and used in manifold contexts. The focus of this work is on management information systems applicable for management decision support systems.

Preceding a detailed elaboration of the history and recent developments, two definitions are presented.

Definition 1: K. Laudon and J. Laudon (2008)

> An information system (IS) can be defined technically as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making and control in an organization. In addition to support decision making, coordination, and control, information systems may also help managers and workers analyze problems, visualize complex subjects and create new products.

K. Laudon and J. Laudon (2008) describe an information system by limiting it to the technical equipment and components required to operate it. The definition is focused on a functional perspective and highlights the solutions, providing the organization to innovate.

Definition 2: James A O'Brien (1999)

> An information system (IS) can be any organized combination of people, hardware, software, communications networks, data resources, and policies and procedures that stores, retrieves, transforms, and disseminates information in an organization.

The second definition by James A O'Brien (1999) describes an IS in a more general way. It is not limited to the technical part of the system but also emphasizes the fact, that information systems influence nearly every part of a company and vice versa.

In the context of this thesis, the second definition by James A O'Brien (1999) is the more applicable one, as it includes all the dimensions that an IS influences.

2.2 History

The utilization of information systems in business has increased over the last decades with the emergence of digital data processing. This chapter looks on the evolution of information systems in the last decades. The evolution will be presented grouped by decades and by

2.2.1 Evolution over time

The developments of IS during the last 70 years can be grouped into the following phases, as illustrated in figure 2.2. Within the next chapter all relevant improvements and distinct feature used will be highlighted. (James A. O'Brien, 2011) (Harsch, 2010)

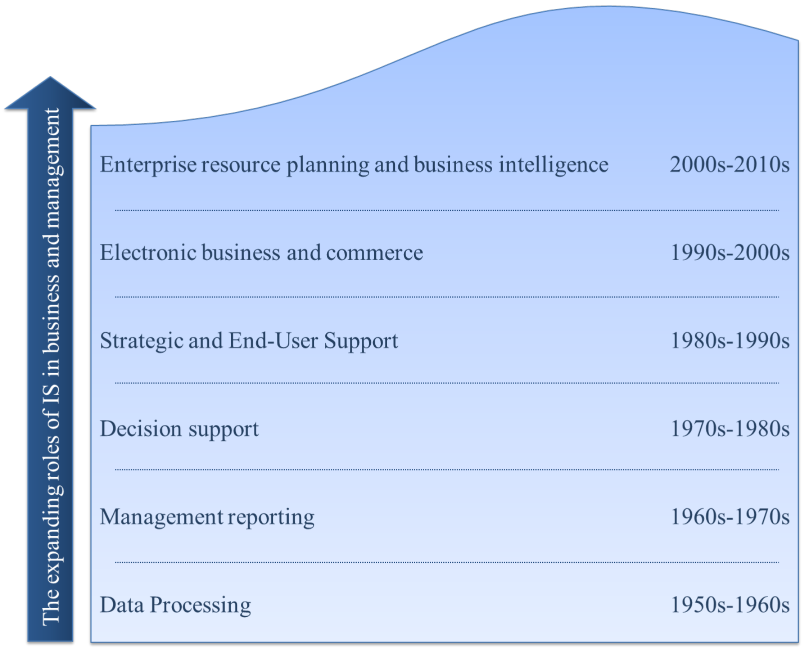


Figure 2.2: Modern information systems and their continuous development in business applications (James A. O'Brien, 2011)

1950's and 1960's Until the 1960's, the main role of IS was to keep record of business processes and provide a data basis for accounting . Later, the stored information was utilized and processed to generate reports and the concept of management information systems was introduced.

1970's By the end of the 1970's, reports providing required information were customized for various situations in business operations. Prior systems did not offer this adaptability. In order to reach this specification, decisioned support systems (DSS) were conceptioned. These systems had the advantage to provide business operations with ad hoc and custom-tailored reports. Managers were now enabled to generate reports that were suitable for short term business decision problems in real world applications.

1980's and 1990's The development rate of processing power of microcomputers increased significantly throughout the 1980's and 90's. This, and the invention of computer networks, enabled the first end user computers. The decentralized structure of end user systems created a variety of new use cases for information systems in business operations. Systems were now also designed to provide data and reports adjusted to the special needs of executives. These executive information systems were not directly used by executives but facilitated their access to critical information when necessary.

Throughout the 1980's, the term strategic information systems was defined. First appearing in 1982, these systems were described to influence a companies' products, processes and services directly in an advantageous way.

At the end of the 1990's, business experienced the development of ERP (enterprise resource planning) systems. Integrated in nearly every function of an organization, it brought a revolution to business. Processes including planning, manufacturing, sales, resource management, customer relations, inventory control as well as financial management and human resources were all supported by these systems.

2000's and 2010's With the beginning of the 21stcentury and the immense growth of inter-networked computers on the world wide web, information systems changed in terms of capabilities and usage. Now, IS could also be developed as web-enabled systems that are now at the core of global organizations and their strategies.

2.2.2 Evolution in stages

In contrast to the classification by decades, the development of information systems can be described by their evolution within a certain application (see figure 2.3). These applications can be summed up into five distinct phases. (Brobst and Rarey, 2003) Companies that want to instrument an information systems are advised to implement those beginning from stage I and subsequently moving to stage V implementations.

Stage I: Reporting Implementations of stage I build upon data sources within the organization. Data is integrated into a dedicated repository storing historical data and making it comparable across functional and product boundaries. These systems are mainly used for rendering predefined reports with low level of complexity.

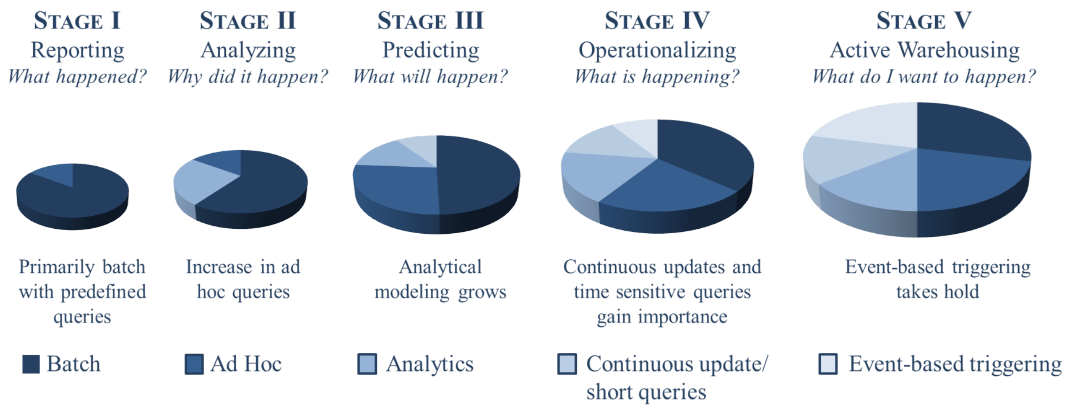


Figure 2.3: IS evolution in five stages according to Brobst and Rarey (2003)

Stage II: Analyzing In contrast to stage I, stage II implementations produce more detailed reports that can be executed on an ad hoc basis.

Stage III: Predicting Within stage III, the gathered data will be used for predictive analysis. Like in previous stages I and II, predictive systems are only used to serve reports for a small set of addressees. Additionally strategic decision-making processes are supported.

Stage IV: Operationalizing The next evolution of predictive analysis is to implement it not only on a strategic, but also on a operational level. The challenge is to provide a system with specialized queries serving up to date information and handling a large scale user base. Stage IV is characterized by the fact that employees can access critical information for decision making in the field.

Stage V: Active Warehousing Building on stage IV, active warehousing eliminates the human contribution where it is not needed or does not bring a benefit to the decision making process. Stage V systems are mainly event-driven.

2.3 Classification

Modern management information systems can be categorized by different variables like application or management hierarchy. These categories help to get a better understanding of the demands and features provided by the different kinds of implementations.

2.3.1 Application

General The general classification differentiates between operations support systems and management support systems and their respective subcategories shown in figure 2.4.

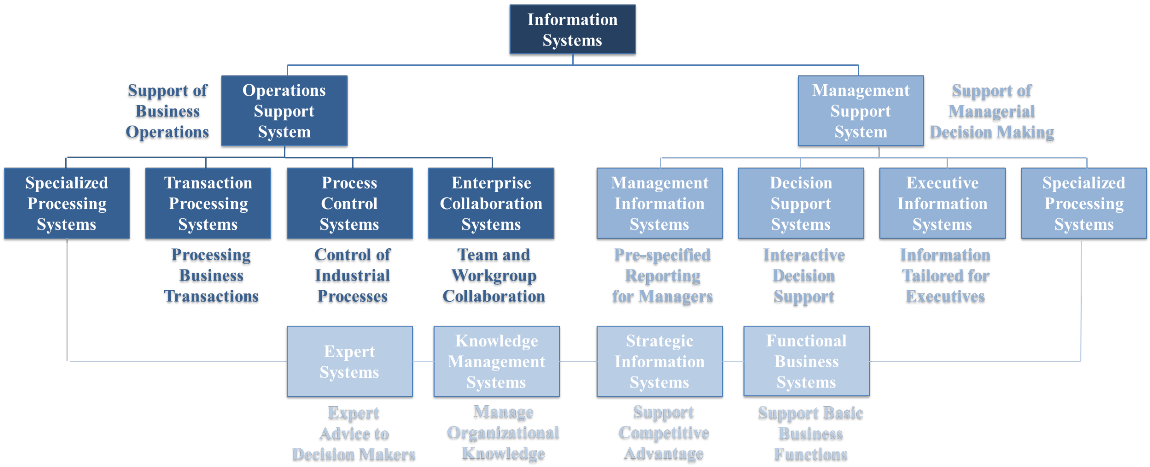


Figure 2.4: Modern information systems classification according to James A O'Brien et al. (2006)

Operations support systems Operations support systems focus on the generation of data for internal and external addressees but lack in the ability to provide specific information tailored for managers and decision makers. To provide data for the management, usually further transformation and enhancement of the data is required. (James A O'Brien, 1999)

Management support systems Information systems that provide information suitable to support the decision-making processes of managers are defined as management support systems. This branch of information systems contains usually the most complex systems, fulfilling the task of providing key information for managers in business. (James A O'Brien, 1999)

2.3.2 Organizational view

Depending on the type of implementation and usage within an organization, three levels of information systems can be identified (figure 2.5) Lower level imple-

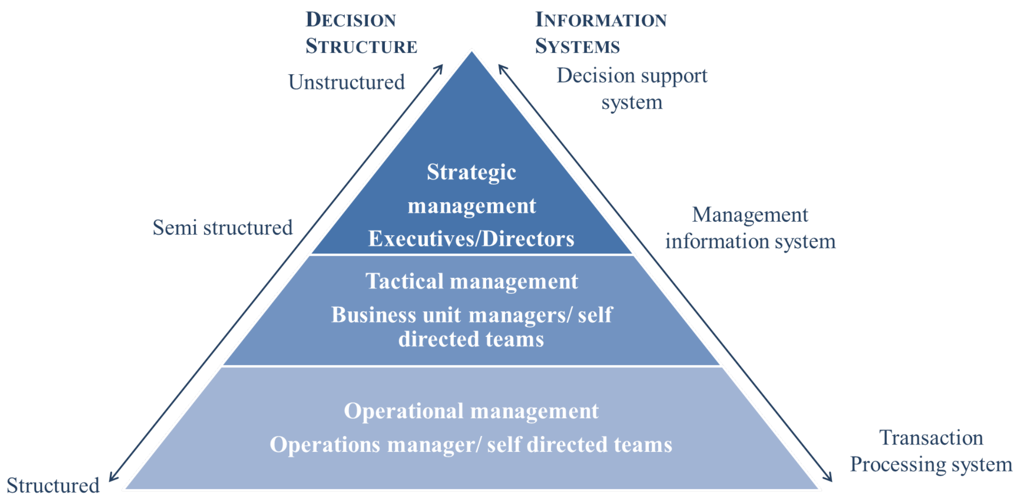


Figure 2.5: Modern management information systems (K. Laudon and J. Laudon, 2008)

mentations are easier to realize, but lack functionality, that systems for strategic management decisions provide. Depending on the level at which an information system is used, information and processing tools need to be implemented accordingly. Table 2.1 provides an overview of important parameters for the individual systems.

	Time horizon	Type of decision	Level of decision	Data actuality	Data sources
Strategical	• Long term	• Company politics	• Broad • Company	• High, historical data • Related to time periods	• Several different sources • Also unstructured • Aggregated
Tactical	• Middle term	• Control of implementation of corporate goals	• Relatively broad • Company, departments	• Rather high • Historical and related to time points	• Several sources • Structured and semi-structured • Slightly aggregated
Operational	• Short term	• pricing • contract conditions • partially automatable	• Focused • Particular processes	• Low, operative data • Related to time points	• Few sources • High level of detail • structured

Table 2.1: Comparison of decision making process parameters (Thiele, 2010)

2.4 Structure, components and key elements

Information systems can be described only by their technical implementation in an business operation, however to get an in-depth view, all dimensions that such a system offers must be taken into account. Modern information systems are in general comprised of all the technical, social and economical factors involved (see figure 2.6). (K. Laudon and J. Laudon, 2008)

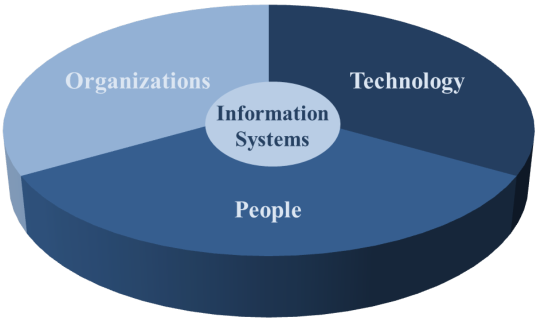


Figure 2.6: IS components (K. Laudon and J. Laudon, 2008)

Organizations An (business) organization is always in a changing state, reacting to both internal and external factors. The information system can change the organization, vice versa the organization can influence the information system itself.

People One of the most valuable assets of an operation are human resources. They are the key to information systems as they are the ones that maintain and update it.

Technology It builds the core of an IS and is comprised of computer hardware and the software.

At the center of the entire system, generally a data warehouse builds the core, providing the required data and toolset for its operation.

2.4.1 Data Warehouse

The term data warehouse was coined by Inmon (1996) and is defined as follows:

Definition

> A data warehouse is a subject-oriented, integrated, non-volatile, and time dependent collection of data in support of managements decisions.[2]

Data warehouses are used as central storage for data from different sources generated during business operation. Current as well as historical data is stored in generic formats and mainly used for reporting and analysis. Data stored in

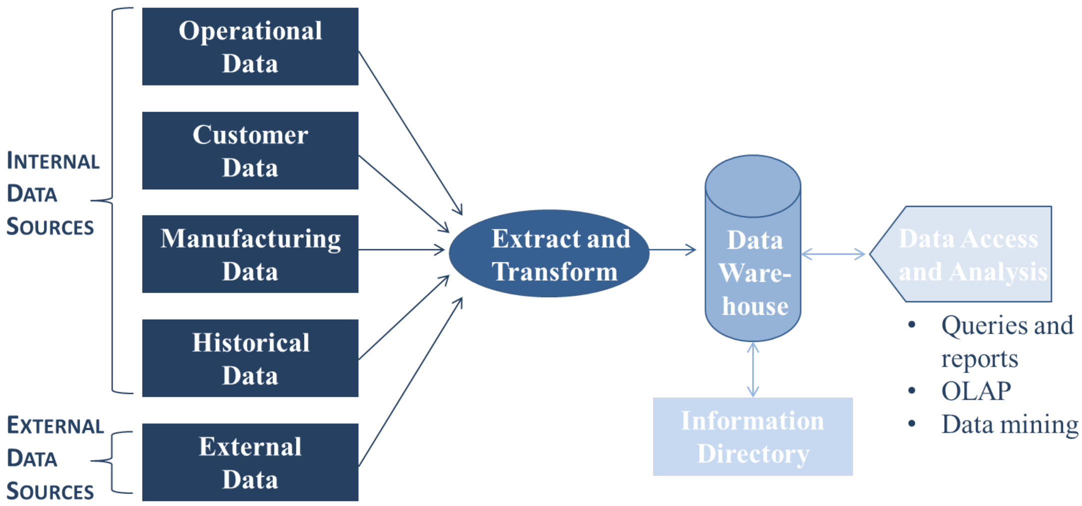


Figure 2.7: Data warehouse information source and retrieval (K. Laudon and J. Laudon, 2008)

data warehouses are *subject oriented* in terms of the limitation to a certain task, transaction or performance indicator, like key performance indicators (see section 2.7).

The data, integrated from various information sources, is further filtered and transformed into interchangeable formats in order to build one unique data base (see figure 2.7).

This process is called ETL:

- *Extract* data from various data sources

[2]translated from German

- *Transform* them into the data model of the target database schema
- *Load* it into the target database

The actuality of the data is not as detailed as in operational databases where data is a representation of the status quo.

Data has a typical resolution ranging from hours to days, weeks or months, depending on the business case. Once data is stored, it is treated as *non volatile*, so it can not be modified or deleted.

The core of a data warehouse is the connection to one, or usually several autonomous databases. Data is copied from the source databases, subsequently cleaned, integrated and finally analyzed. These operations can be carried out without compromising the function of the source databases. Data integrated into the warehouse, originating from different source systems, is organized within different data models. These models get harmonized while the data is imported and can be adopted to the requirements of further analysis.

2.4.1.1 Architecture

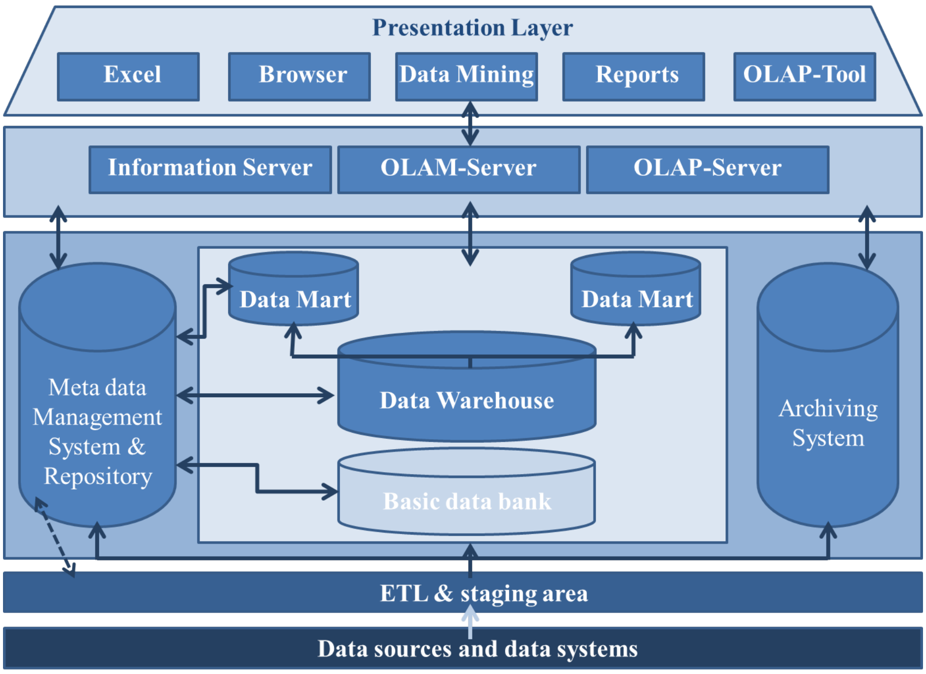


Figure 2.8: Data warehouse architecture (Farkisch, 2011)

Data warehouses need multiple components to function correctly. Figure 2.8 presents an example architecture of a data warehouse used for generating reports and analysis.

Required data is extracted on the lower level within an ETL process and stored in the data warehouse. From the information extracted, data marts can be created which build the source for the data preparation layer. The information layer features e.g. OLAM (online analytical data mining) or OLAP servers. On top of the warehouse, the presentation layer provides client and front end applications for end users. (Farkisch, 2011)

2.4.1.2 Data representation

Traditionally, data is stored in relational databases providing efficient storage and fast transaction times. However, this type of representation is not suitable for complex data modeling required for information systems. Therefor, data is modeled in multiple dimensions that allow the presentation of various aspects (e.g. time, place, type, ...) at once. Additionally this type of model, called OLAP cube (see 2.9), enables further data classification in the form of hierarchies. (Farkisch, 2011)

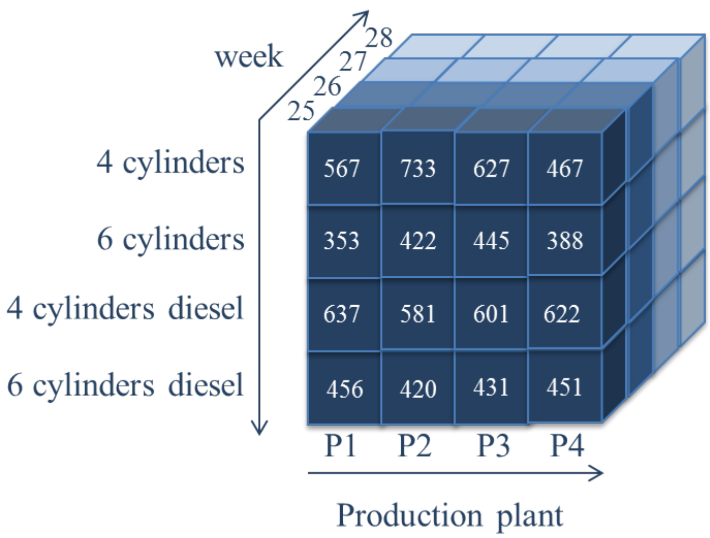


Figure 2.9: Basic OLAP cube example (Farkisch, 2011)

2.4.1.3 Online analytical processing

OLAP is a structured technique to analyze large amouts of structured data. Integrated within the technologies of a data warehouse, the center of all techniques builds the OLAP cube of which an example can be seen in figure 2.9.

Principles To describe the characteristics of OLA processing, certain rules were crafted by E. F. Codd, S. B. Codd, and Salley (1993) in the form of a 12 point description. These principles were further developed, shortened and published by (Pendse and Creeth, 1995) to form the "FASMI" (fast analysis of shared multidimensional information) concept. **FASMI:**

- **Fast** Queries should have quick response times, not more than 5 seconds in particular. Complex queries should not exceed 20 seconds.
- **Analysis** OLAP systems shall be able to fulfill all possible queries. Users with little programming knowledge should be able to create complex queries with a predefined language set.
- **Shared** The system must support multiple users, including administration capabilities.
- **Multidimensional** The system must be based on a multidimensional data structure including hierarchies.
- **Information** The entity of information stored in the system must be accessible by all users equally.

Operations The way data is represented in a data warehouse allows it to be easily transformed into different representations. These operations can be interactively performed on a given dataset, stored in the data warehouse. The basic operations will be presented with a short description:

Pivot/Rotate With the rotate function the axes can be interchanged with each other, allowing to view data from different perspectives (see figure 2.10).

Drilling When the drill down operation is performed on a multidimensional data set, data will be presented in a more detailed structure. The level of detail is increased, e.g. from years, to months or days (figure 2.11).

Rollup The rollup builds the complementary action to drilling. This operation preserves all dimensions and additionally generates new information by aggregating existing data. An example would be calculating the monthly revenue from daily data.

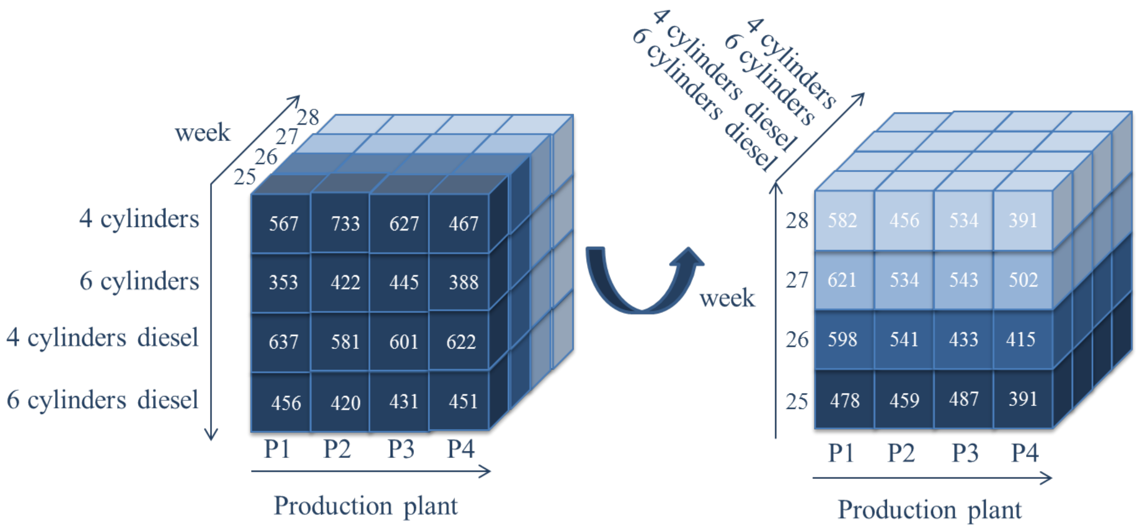


Figure 2.10: OLAP Pivoting (Farkisch, 2011)

Dicing Performing the dice operation on a data warehouse extracts a part of the data set. This preserves the dimensions of the data, but changes the hierarchy within (figure 2.12). This operation would for example extract regional sales figures from a global perspective.

Slicing The slicing operation takes a layer of data from the multidimensional representation. By aggregating data, the number of dimensions is reduced and single layers are extracted and viewed. An example would be all revenues of the past year.

Drill-across By applying the drill across action, data is presented combined with the classification from another data cube. To perform this action, queries from more than one cube are necessary.

2.5 Data source

Information systems need a constant information feed to keep the data, reports and further predictions up to date. This feed can be obtained by various information sources. In the case of manufacturing and production processes this information systems can be classified as illustrated in figure 2.13

Data gathered during the production process usually stems from manufacturing execution systems. These systems like PEL (Programmable Logic Controller) or

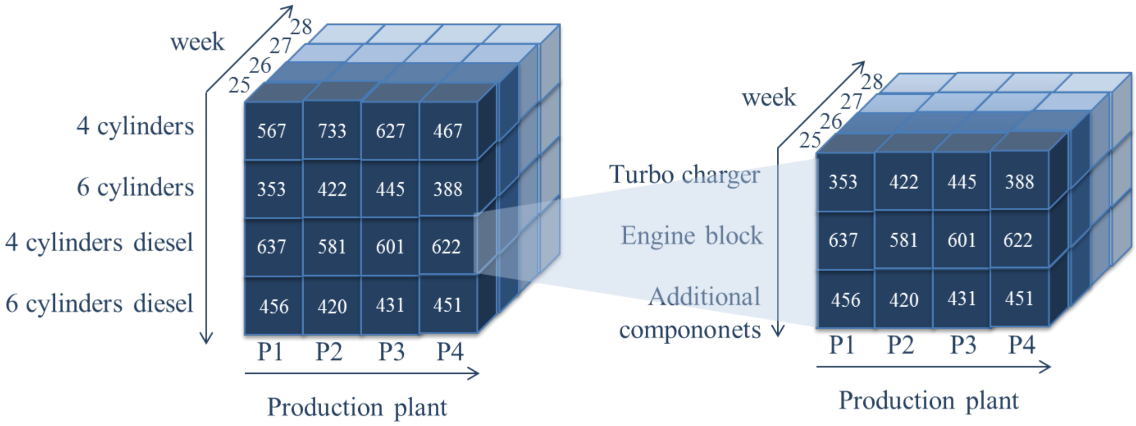


Figure 2.11: OLAP Drilling (Farkisch, 2011)

2.6 Application of information systems

Providing manifold features, information systems can be of use in various business cases. A selection of cases will be presented and the pros and cons as will be discussed.

2.6.1 Information systems embedded into business processes

Businesses implement information systems for various reasons into their organizations. Generally, these motivations can be summed up into three key arguments (James A O'Brien, 1999):

- Support for business processes and operations
- Support of decision making by employees and managers
- Support of strategies for competitive advantage

These points are also illustrated in the pyramid in figure 2.5. Structured data gathered on daily basis builds the foundation on which more sophisticated reports can be developed. Of course, businesses are additionally required to store a certain set of data to abide to the local regulations of the countries in which they operate.

Systems for tactical management add a dimension of abstraction and are addressed to managers in the operative management. Systems on this hierarchy level are characterized by the fact, that the users (managers) need to access related data content that can therefore be standardized. (Lapiedra, Alegre, and Chiva, 2011)

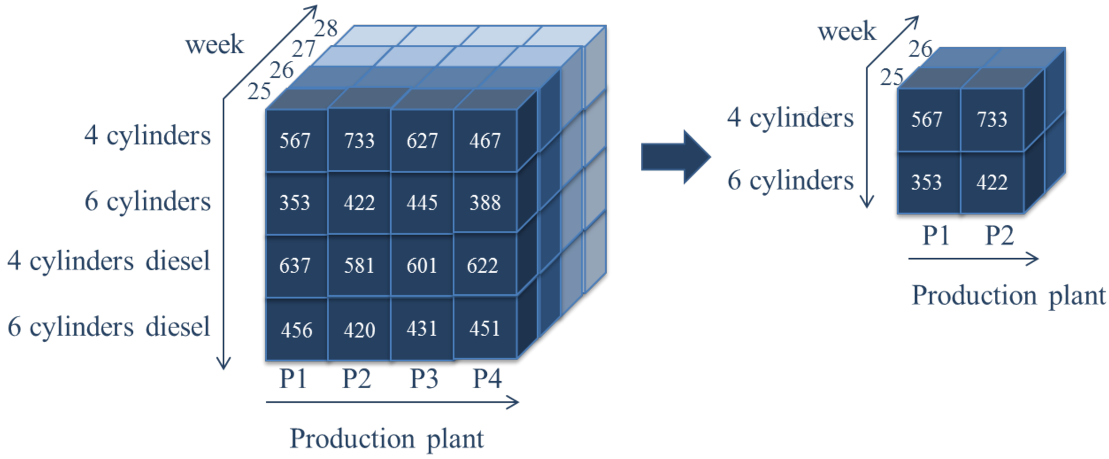


Figure 2.12: OLAP Dicing (Farkisch, 2011)

In upper level management, decisions get more complex and involve more individual freedom, therefore requiring the most complex implementations of information systems. Data provided for these users is not recurring and needs manual input. Reports generated by these systems are reliant on user interaction and contextual knowledge to provide respective information for each use case. (K. Laudon and J. Laudon, 2007)

When analyzing the motives for implementations of information systems in more detail, six key factors can be identified (K. Laudon and J. Laudon, 2008):

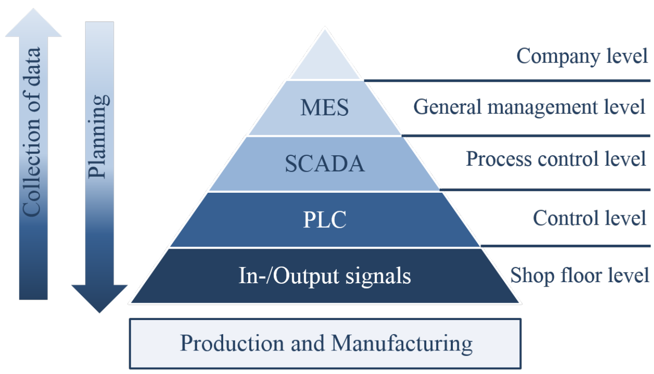


Figure 2.13: MES data source classification according to IEC 62264

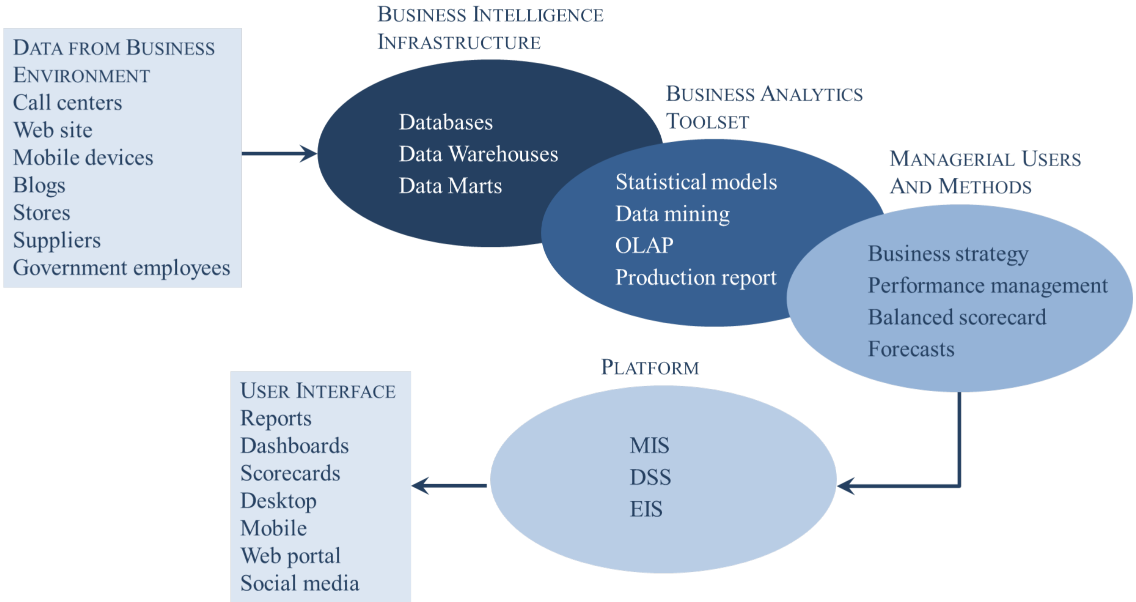


Figure 2.14: MIS embedded into business process (K. Laudon and J. Laudon, 2008)

Operational excellence Companies set measures like TQM[3] or TRIZ [4] to improve the efficiency of the business. (Stratton and Mann, 2003) By analyzing past data, problems can be identified and appropriate measures can be set. After a certain period of time, evaluation of newly gained data allows a determination of the success of these measures.

New products, services, business models Developing new services or products provides new opportunities but also bears a certain risk. Information systems help by providing data for risk assessments and market evaluations.

Customer and supplier intimacy For modern production plants with small warehouse capacity and *just in time manufacturing*, the communication with suppliers is a key to successful operation. By intensifying communication, suppliers can be more responsive and can provide better input for future improvements. Equally, customers can be valuable sources of information and their feedback can contribute to further improvements.

[3]Total quality management
[4]*Problem solving and analysis tool*

Improved decision making Many employees in companies base their decision making processes on rough estimations and predictions guesses due to the lack of qualitative information. Modern information technology makes it possible to provide that required information.

Competitive advantage Information systems help organizations to keep better records and respond to market and competitor changes faster. These results in better services, faster response times and finally also higher profits.

Survival In special cases, organizations have to implement certain information systems to stay on or enter certain markets. These can be e.g. industry standards that are already in place or rules and regulations issued by the authorities.

2.6.2 Costs and benefits of information systems

When implementing an information system into business operations, numerous advantages can be identified but also the costs of such an investment have to be considered.

Costs of implementation can include hardware, telecommunication equipment, software, services as well as specially trained personnel.

The benefits of information systems can be distinguished between *tangible* and *intangible benefits*. Tangible benefits can be directly translated into cost savings or other monetary values. Intangible benefits can not be quantified into cost savings. Additionally these benefits can not be as easily identified and measured. As examples for intangible benefits, improved decision making or better customer service could be named. (K. Laudon and J. Laudon, 2012)

Tangible benefits

- Increased productivity
- Lower operational costs
- Reduced workforce
- Lower computer expenses
- Lower outside vendor costs
- Lower clerical and professional costs
- Reduced rate of growth in expenses
- Reduced facility costs

Intangible benefits

- Improved asset utilization
- Improved resource control
- Improved organizational planning
- Increased organizational flexibility
- More timely information
- More information
- Increased organizational learning
- Legal requirements attained
- Enhanced employee goodwill
- Increased job satisfaction
- Improved decision making
- Improved operations
- Higher client satisfaction
- Better corporate image

2.6.3 Challenges of information systems

Apart from the various advantages of MIS, also problems arise when operations start to integrate these systems.

Numerous surveys of companies in the US and the UK utilizing information systems have shown, that such systems are no guarantee for instant success. Despite using advanced information systems the studies have shown *relatively little success*. The main reasons for this lack of success can be summed up into these points (Lucey, 2005):

- Absence of management involvement in the early implementation phase of IS
- The systems are used or emphasized in a wrong manner
- Too much focus on lower level data processing
- Management personal lacking computer and IS knowledge, particular in SMEs (small and medium enterprises)
- Lack of top management support

Most of the challenges listed above can be addressed by a revised planning and structuring of the design and implementation phases. Improving communication between IS experts and management leads to a better utilization of IS.

VENDOR	PRODUCT(S)
IBM SPSS	SPSS Modeler (Formerly SPSS Clementine)
SAS	Enterprise Miner
SAP Business Objects	Predictive Workbench (Based Upon SPSS Clementine)
Oracle	Data Mining option for the Oracle database or for Essbase database
MicroStrategy	Data Mining Service
Think Analytics	thinkAnalytics
Pentaho	Weka
Angoss	KnowledgeSEEKER, KnowledgeSTUDIO

Table 2.2: BI (business intelligence) market: Data mining & predictive analytics - Vendors and products (BI-Insider, 2011)

2.6.4 Implementations

Currently, various implementations of software packages offering features for information systems are available. Most software products build upon standard data warehouses and allow to deploy additional packages, thereby enabling them to be extended to modern information systems.

2.6.4.1 Market share of OLAP products

In table 2.2 a selection of current commercial implementations, featuring advanced capabilities for information systems is shown. Due a high fragmentation of the market no accurate figures regarding market shares are available.

2.6.4.2 Open source alternatives

For the sake of completeness we also want to mention alternatives to the commercial products. Druid[5] is marketed as a "fast column-oriented distributed data store" featuring a set of OLAP functions. The product sourcecode is available on Github[6] with the package still in active development.

[5] http://druid.io
[6] https://github.com/druid-io/druid/

2.7 Key Performance Indicators

Modern information systems provide the access to information generated by an particular enterprise. However decision makers require information to be available in a compressed form. This target can be achieved by reducing the available information to more simple and meaningful figures, named key performance indicators (KPIs).

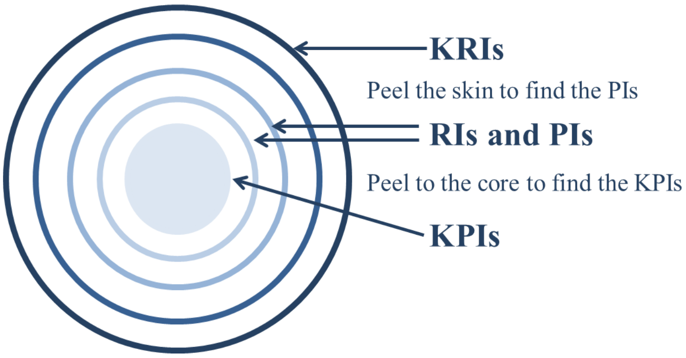


Figure 2.15: Performance measures according to Parmenter (2001)

Definition by Parmenter (2010):

> KPIs represent a set of measures focusing on those aspects of organizational performance that are the most critical for the current and future success of the organization. An overview of performance measures in general is illustrated in figure 2.15.

According to Parmenter (2010) the indicators can be described in more detail by the following seven characteristics:

- Non financial
- Measured frequently
- Acted on by the CEO and senior management team
- Clearly indicate what action is required by staff
- Measures that tie responsibility down to a team
- Have a significant impact
- They encourage appropriate action

In addition to KPIs, also other categories of indicators exist that are valuable for describing other objectives:

Key result indicators (KRIs) typically indicate the result of an action, but do not give conclusions on how to improve these results. Typical resource indicators would be customer as well as employee satisfaction, net profit before tax or return on capital employed.

Performance indicators (PIs) lie between KPIs and KRIs and are nonfinancial and help to align teams within organizations. Typical examples would be the number of customer complaints of most important customers, late deliveries to key customers.

3 Requirements Engineering

This part of the thesis will provide an overview about the theoretical background of requirements engineering. It gives an insight on the historical and recent developments of the field and an detailed look on a specialized implementation of the approach, the SOPHIST® framework. (Rupp and SOPHISTen, 2009)

This master thesis utilizes the technique of RE to provide the framework for demonstrator development process. In detail, a slightly extended and more application-oriented RE variant, namely the SOPHIST approach, was used.

3.1 Background

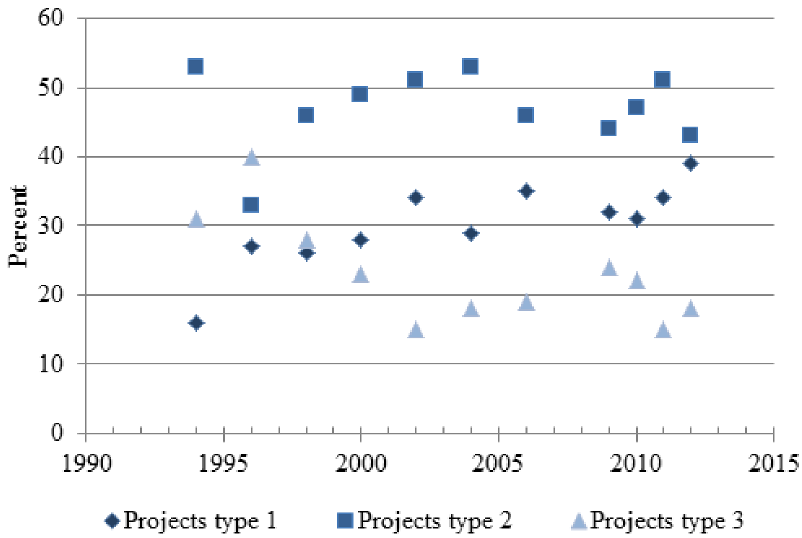


Figure 3.1: Software engineering project success by project type (Standish Group 2015)

As already mentioned in the introduction (see section 1.1) software engineering projects often face challenges during the planning phase. Since 1994, the chaos report is published annually, providing a quality measure for software engineering projects (see figure 3.1). According to the chaos report of 2015, RE is an approach that facilitates the planning process, thereby increasing the probability for project success.

The Standish Group (2015) defines the project types as follows: **Type 1** Successful projects are completed on time and within the budget limit. Additionally all initially defined functions and features are implemented. **Type 2** Challenged projects are build and functional but have exceeded their initial budget, did not complete in time and are not feature complete. **Type 3** Impaired projects are aborted at some point during the product development cycle.

3.2 Definitions

According to the author of various successful RE books, Chris Rupp, RE is defined as follows:

What is requirements engineering according to the SOPHIST book author C. Rupp:

> Requirements engineering is the approach to gather requirements from all stakeholders and to align and organize them. Stakeholders are all persons, that have any influence on a given system, ranging from the development staff to the sales team as well as the designated users. It is essential to gather all requirements, facilitated by different techniques like interviews and brainstormings, but also through system archeology (for example studying documents). The outcomes should be documented in such a manner, so that afterwards a system can be commissioned and/or built.

This description of the field gives a simplified overview about the matter. The following section will present formal definitions of the matter.

3.2.1 Stakeholder

A stakeholder involved in a system is a person or organization, which has (direct or indirect) influence on the requirements of the given system (K. Pohl 2011).

3.2.2 System context

All requirements for a new system have to be developed and are not existing beforehand. The definition of the system context is the task of the requirements engineer. It is necessary to define the system context as well as the system borders (figure 4.4). (Pohl and Rupp, 2009)

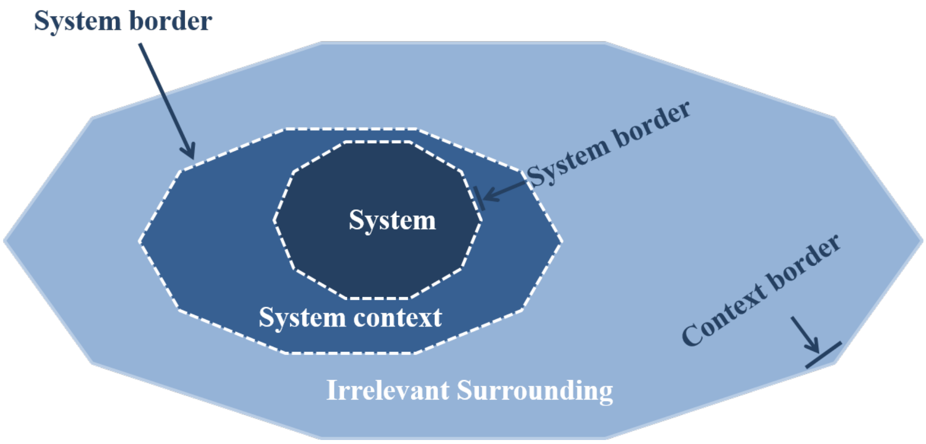


Figure 3.2: System- and contex tborder of a system according to Pohl and Rupp (2009)

Definition The system context is the part of the environment of a system, which is relevant for defining and understanding all requirements of this respective system.

System border The system border separates the planned system from its environment. It separates the part of reality that can be shaped and influenced during the development process from aspects in the environment that can not be changed by the development process.

Context border The context border separates the relevant part of the environment of a planned system from the irrelevant part; that is the part of the environment that does not influence the planned system and therefore also not the requirements of this system.

3.2.3 Requirements

In the past the term requirements in the context of requirements engineering was defined by different sources. One of the cited definitions stems from the IEEE standard:

Definition of requirements ("IEEE Standard Glossary of Software Engineering Terminology" 1990)

(1) A condition or capability needed by a user to solve a problem or achieve an objective.
(2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification or other formally imposed documents
(3) A documented representation of a condition or capability as in (1) or (2).

3.2.3.1 Requirements types

Generally two types of requirements can be distinguished, the functional and non functional requirements. Often a third type is distinguished: the One typical error made in a requirements engineering process is to neglect the non functional requirements as they are harder to quantify. (Pohl and Rupp, 2011)

Functional requirements Function requirements define the functionality that define the planned system should implement. They can be defined as functional-, behavioral- or structural requirements.

Qualitative or non functional requirements Define the qualities of a system under development. These influence the functional requirements by a great margin. Typical qualitative requirements define performance, availability, reliability, scalability and portability.

Ancillary conditions These conditions can't be influenced by people involved in the project. Ancillary conditions also can't be implemented, but limit the possible solutions for the system. The conditions can limit the system itself (e.g. the system must be implemented using web services) or the development process (usually time constraints).

3.2.3.2 Importance of qualitative requirements

In day to day business qualitative requirements are only documented insufficiently and not coordinated between stakeholders. This can influence the acceptance of the system and can lead to a great dissatisfaction of the end user when not treated correctly.

Therefore a checklist has been developed to gather all of these qualities. It is a standardized structure which can be used a checklist (compare to ISO/IEC (2001)):

- All qualities regarding functionality but especially regarding sustainability, accuracy, interoperability security and functional compliance.
- Requirements which influence reliability and a with focus on fault tolerance, maturity, recoverability and reliability compliance.
- Requirements regarding usability of a system: Most important Understandability, learnability, operability, attractiveness and usability compliance.
- Requirements regarding efficiency for a system: This is limited to time behavior, resource utilization and efficiency compliance.s
- Requirements regarding maintainability: It specifies qualities for analyzability, changeability, stability, testability and maintainability compliance.
- Requirements regarding portability: This defines the facts for adaptability, installability, co-existence, replaceability and portability compliance.

With this check list available for a requirements elicitation all non functional requirements can be obtained beforehand.

3.2.3.3 Requirements sources

An important part of the requirements engineering is the elicitation of the requirements of the target system. Generally three different requirements sources can be determined. (Pohl and Rupp, 2009)

Stakeholders are all persons that can influence the system (compare chapter 3.2.1).

Documents usually contain important information and are e.g. norms, documentations from existing systems, laws or regulations.

Operational systems can be existing implementations or a product of competitors on the market which can be used as baseline for evaluation. Stakeholders can try existing systems and give feedback for potential features or changes.

3.3 KANO Model

The KANO Model (Pohl and Rupp, 2009) illustrated in figure 3.3 provides an insight on how to categorize the expectations of a user of a system. It is regularly cited in the context of requirements engineering, as it provides a good overview of the importance requirements have on stakeholders.

Dissatisfiers/Basic needs are self-evidently assumed system characteristics (subconscious knowledge).

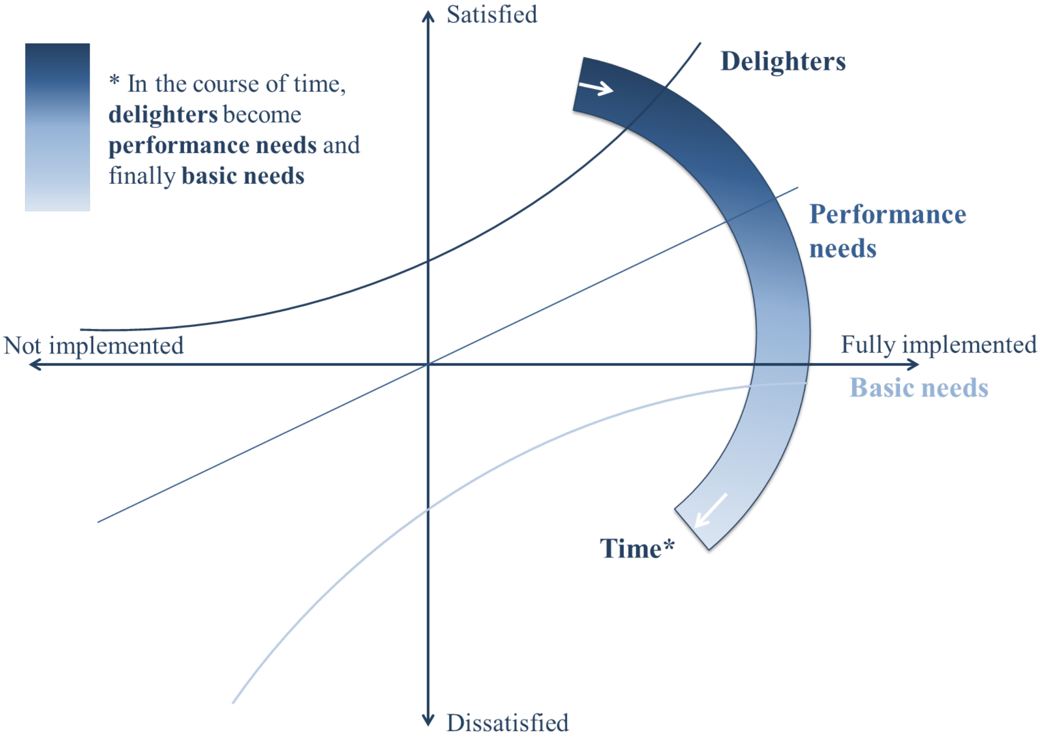


Figure 3.3: KANO model according to Pohl and Rupp (2009)

Satisfiers/Performance needs are explicitly demanded systems characteristics (conscious knowledge)

Delighters are all characteristics that are not known to stakeholders and first get noticed during usage as pleasant surprise.

Over time delighters develop into performance needs and finally to basic needs. During the identification of requirements all three categories of requirements have to be respected.

3.4 Requirements engineering phases

The typical requirements engineering approach can be split into four main phases as illustrated in figure 3.4. (Rupp and SOPHISTen, 2009)

- **Requirements elicitation and analysis**
- **Requirements specification**
- **Requirements validation**

- **Requirements management**

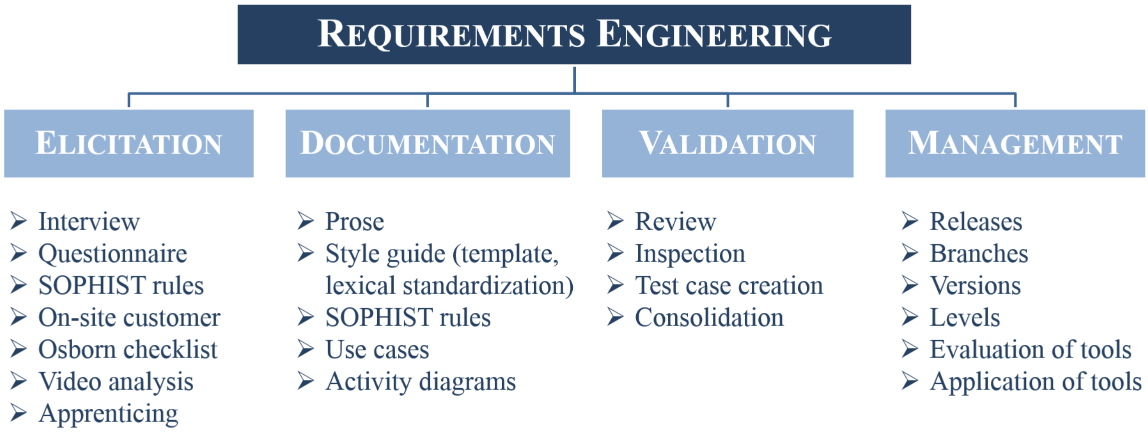


Figure 3.4: Requirements engineering steps and subtasks involved (Rupp and SOPHISTen, 2009)

In addition to the standard approach the SOPHIST REgelwerk specifies a pre-elicitation phase that is executed prior to the four main phases.

3.4.1 Pre-elicitation

In the SOPHIST approach this particular phase represents one of the most important steps in the requirements engineering process. The focus is on the analysis of the current situation (AS-IS analysis) and identification of potential improvements and objectives.

3.4.1.1 AS-IS analysis

Within the AS-IS analysis the current situation is examined and potential systems and processes already in place are evaluated. Possible sources for information are operational systems, stakeholders and documents (see section 3.2.3.3). Additionally existing problems and deficiencies are recognized which builds the basis for the next step.

3.4.1.2 Potential improvements and objectives

Based on the AS-IS analysis of the initial situation, potential improvements can be identified and a list of objectives will be defined. The determined list of goals, the system environment as well as limitations and borders have to documented precisely. Therefor, one possible format is the specification in form of use cases.

3.4.2 Elicitation

Elicitation of requirements is one of the main tasks of the RE process. Besides documents and existing systems, stakeholders represent the main information source. Therefor, a good communication between requirements engineers and stakeholders is essential. Different methods can be used for the elicitation process such as interviews, questionnaires and on-site observations.

3.4.3 Documentation

The requirements documentation has a central role in the requirements engineering technique. Generally the number of requirements rises quickly to an unmanageable amount. Therefore it is highly critical that requirements are documented in a clear and structure manner. It should be easily possible for external persons to find, understand and visualize the requirements. The documentation can use natural language as well as model based techniques such as and-or trees, use cases or UML diagrams. An established form for the natural language documentation is

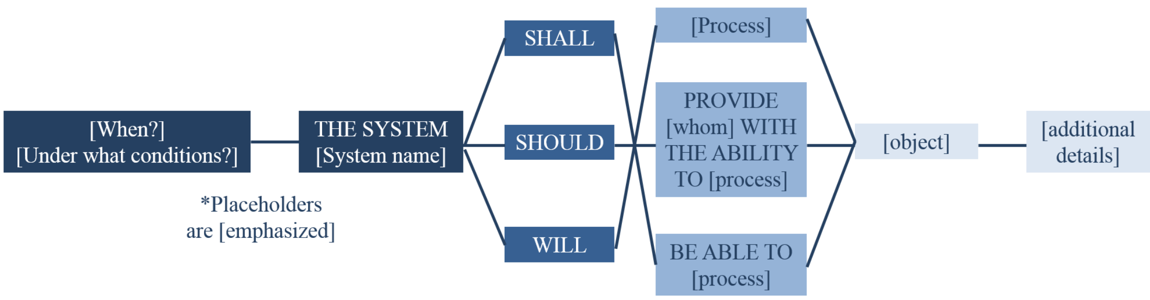


Figure 3.5: Template for formulating a requirement specification (Pohl and Rupp, 2011)

the usage of document templates (see figure 3.5) and patterns. These templates are usually project specific and can additionally contain model based approaches.

3.4.4 Validation

All elicited and documented requirements have to be validated for their quality and their compliance with the stakeholders expectations. Therefor different methods can be applied:

- Review (e.g. commenting, inspections and walkthroughs)
- Perspective aware reading
- Prototype
- Checklists

To coordinate all existing requirements for a given system, potential conflicts have to be identified, analyzed and solved.

3.4.5 Management

Aim of requirements management is ensure the availability of all documented requirements as well as all other relevant information during the whole system/product life cycle. The information has to be well structured, persistent as well as accessible. To fulfill this process several different techniques are applied:

- Attribution of requirements
- Prioritisation of requirements
- Traceability of requirements
- Versioning of requirements
- Change management of requirements

3.5 Requirements engineering versus agile development process

Size	Method	Successful	Challenged	Failed
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	1%

Figure 3.6: Software engineering project success by project size and software development process (according to the Standish report 2015)

In contrast to linear approaches like requirements engineering or the waterfall model also agile software engineering methodologies such as Scrum or extreme programming exist. Recent studies have shown that project success correlates with the applied software engineering process (see figure 3.6).

4 Case Study

The practical part of this master thesis is embedded into a project by the institute of engineering and business informatics in cooperation with Audi Hungaria Motor that aims to improve the internal reporting processes of the production plant in Györ. One part of the project is to develop a mobile application to present management information. In order to engineer the prototype, the method requirements engineering was utilized for the detailed definition of use cases and to improve the overall development process.

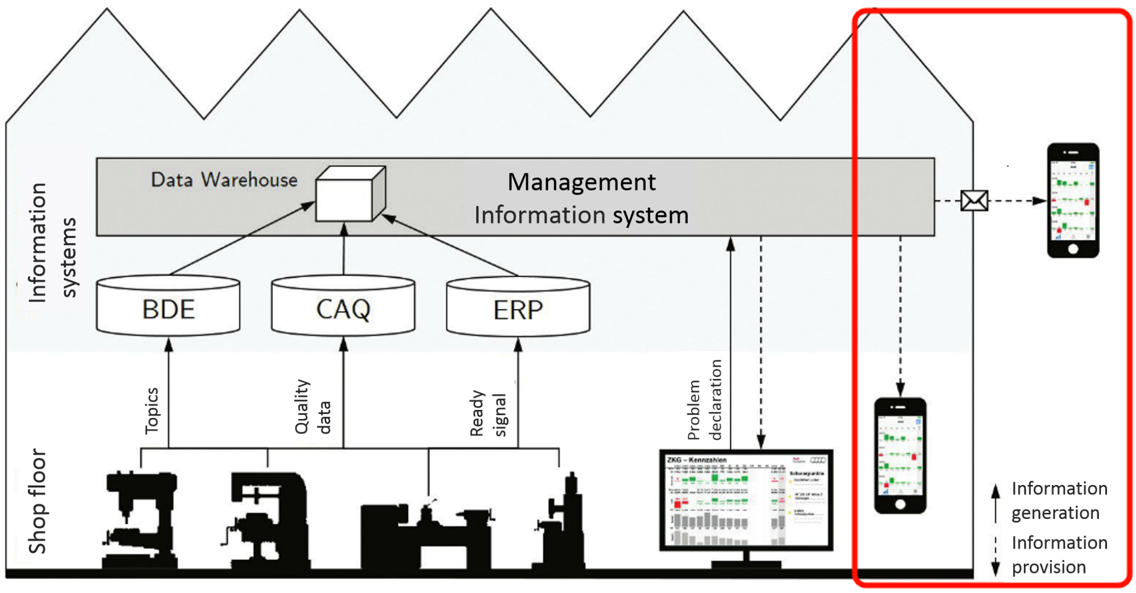


Figure 4.1: Project Overview: The red circled part highlights the scope of this thesis (Wolfsgruber and Lichtenegger, 2015)

The overview graphic (figure 4.1) from the introduction is included here again. It gives an overview of the overall project and the masters thesis embedded into it.

The overall was project was realized using a process model (see figure 4.2) suggested by Lichtenegger et al. (2016).

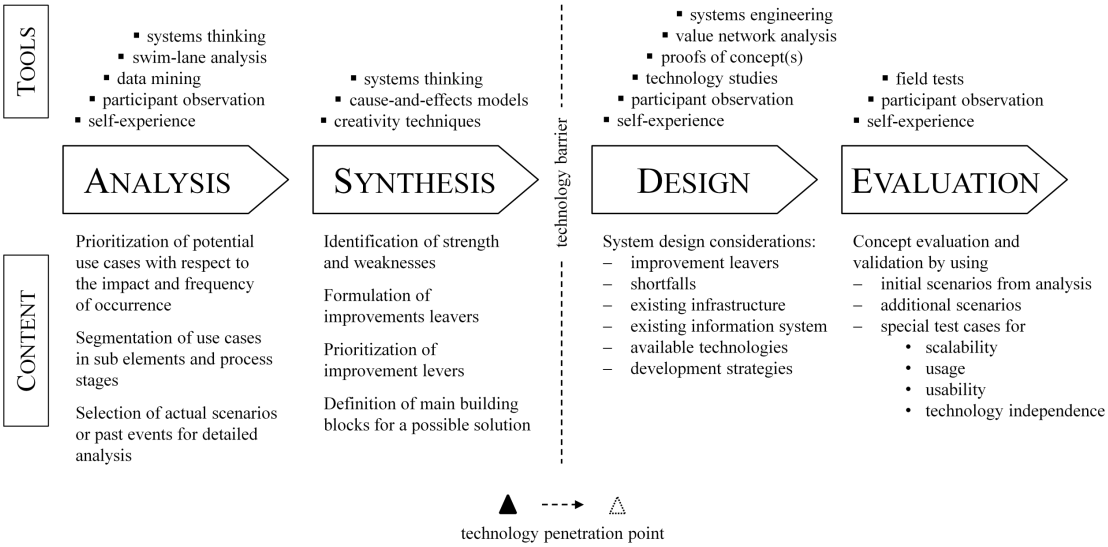


Figure 4.2: Design pattern for technology-independent information system architectures (Lichtenegger, 2015)

Within the overall project scope, the practical part of the thesis was carried out during the design phase of the suggested process model, finally resulting in the implementation of the prototype.

4.1 General project goals

As described in the theoretical part (see chapter 3.4.1.1), a part of the AS-IS analysis is to provide a list of overall project goals. These are the high level goals to achieve and can be categorized as requirements of the specification level 0.

The most important goals of the projects are also a subset of the overall project outcome of the project of G. Lichtenegger and C. Wolfsgruber:

- Improve the existing internal reporting system utilizing modern information technology
- Develop a concept for a management information system on a mobile device
- Embed the mobile MIS application in the well established internal reporting processes
- Enable the tracking of problems and topics that arise during the day-to-day production process
- Support the resolution of problems that occur during production

- Assist the TQM [7] process at the production plant
- Provide a basis for short- and midterm decision-making for middle and upper management

4.2 Use cases

In the following sections all identified use cases will be presented. The main information source used for the use case identification were the on site presence as well as informal talks and interviews with the potential users and stakeholders. All use cases can be be divided into three categories related to their usage of data.

4.2.1 Use case identification

The goals for the proof of concept (POC) implementation were defined in the early stage of the project during discussions with the key stakeholders and on-site observations and subsequently developed within the scope of requirements management. To document the use cases implemented in this thesis, the use case schema provided by the SOPHIST framework is used.

4.2.2 Use case specification

Within the work of this thesis several use cases have been implemented. The documentation was based on the suggestion of the SOPHIST framework with the corresponding use case table. (Pohl and Rupp, 2009)

The template suggested gives great detail and is scaleable from small up to large projects. Given the manageable number of use cases and the fact that during operation only a limited number of components are involved, a simplified version of the suggested scheme will be used (see table 4.1).

4.2.3 Use case 1: Key performance indicators

The main use case in this thesis is to provide an overview of the most important key performance indicators of the plants productions lines. Starting the application,

[7] Total quality management

#	Section	Content/Description
1	Identifier	Unique identifier for a use case
2	Name	Unique name for a use case
4	Priority	Importance of the use case, corresponding to the used prioritization technique
5	Criticality	Criticality of the use case concerning the extent of loss in case of failure of the use case
6	Source	Description of the source (e.g. stakeholder, document, system), from which the use case originated
8	Description	Brief description of the use case
9	Triggering Event	Specification of the event, that triggers the use case
10	Actors	List of all actors, that have a connection to the use case
11	Precondition	A list of all required preconditions which have to be fulfilled, before a use case can be started
12	Post condition	A list of all states a system can be, immediately after executing the main scenario
13	Result	Description of all outputs generated during the execution of the use case
14	Main scenario	Description of the main scenario of the use case
15	Alternative scenario	Discerption of alternative scenarios or trigger events
16	Exceptional scenarios	Description of exceptional scenarios
17	Qualities	Cross-references to qualitative requirements

Table 4.1: Adopted use case template for documenting use cases during requirements engineering suggested by Pohl and Rupp (2009). Numbering is taken from original work and too detailed sections were eliminated.

it should instantly provide an initial overview of production-relevant performance indicators.

4.2.3.1 AS-IS analysis

As of now, several different reports for the reporting of important performance indicators coexist. There is no homogeneous reporting in place and the information status varies between the production segments within the production plant. The existing systems evolved over time, provide no consistency and come in different formats (pdf, excel, generated HTML reports) and time horizons. Furthermore, the reports differ in visual representation, have unique calculation methods (e.g. complex excel computations) built in and are mainly distributed via exported PDF files. These reports are generated on the shop floor level, by manually transferring the information into summed up reports. The source for the information are three sub-systems in place that track various parameters. This systems are : The CAQ

#	Section	Content/Description
1	Identifier	UC1
2	Name	Presentation of relevant KPI of the production facility
4	Priority	High
5	Criticality	High
6	Source	Stakeholders
8	Description	The user can review important KPI's within one view
9	Triggering Event	Starting the application (see use case 7)
10	Actros	User
11	Precondition	App must be installed and .mis file must be imported
12	Post condition	App is started an UC1 – UC4 can be triggered
13	Result	Tabular view of all segments of the production plants with corresponding KPI are visible
14	Main scenario	User wants to track KPI's and lot sizes of the current time period or of past time frames • User Starts the app with or without a new file • User reviews KPI's • User performs addittional OLAP operations • User leaves APP
15	Alternative scenario	-
16	Exceptional scenarios	No data is present • Use the demo file (see use case 7)
17	Qualities	-

Table 4.2: Use case 1: Overview of important KPI's

system (computer-aided quality), MES (manufacturing execution system) and an EPR (Enterprise resource planning) system.

4.2.3.2 Potential improvements

A mobile information system can be beneficial for the process of tracking all relevant performance indicators within one application. This should limit media disruption[8] and enable monitoring of the production plant on site as well as in a remote location. Within one screen, all relevant information should be visible and display an overall picture of set production targets and their fulfillment.

4.2.3.3 Resulting use case

The resulting use case is described in table 4.2 and is tailored for the use of an application running on a mobile device. Operated by a middle or higher level manager, the application should deliver an executive summary like an overview of all key performance indicators within one glance.

4.2.4 Use case 2: OLAP functionality

With the OLAP operation drill up /down, users can maneuver up and down in the production plant hierarchy and view the corresponding KPI's. This can range from the output of a single machine up to segments or the whole production plant productivity. The effective range the operators can navigate is hugely dependent on the availability of the information, the cost of extracting it and the benefit that the information provides.

4.2.4.1 AS-IS analysis: OLAP functionality

The current reporting process only delivers static, predefined reports to decision makers in daily business. This state should be improved by facilitating OLAP functionality presented in section 2.4.1.3. Reports displayed on a mobile device should be interactive and allow the user to change the KPI hierarchy and customize the time horizon and in particular the operations drill up/down.

4.2.4.2 Potential improvements

The drill operations add a good basis for the detection of potential issues in production. With this tool it is easy to pinpoint the source of production issues within a production line.

4.2.4.3 Resulting use case

This use case is also preset to be implemented on a mobile device platform. Table 4.3 provides an detailed description of the use case.

[8] from the german "Medienbruchfrei": providing data in a consistent format throuout a process without changing the medium of communication (Fleisch and Dierkes, 2003)

#	Section	Content/Description
1	Identifier	UC2
2	Name	Drill Down
4	Priority	High
5	Criticality	High
6	Source	Stakeholders, current reporting system
8	Description	Represent the hierarchy of the production plant with corresponding KPI's
9	Triggering Event	During KPI review a deviation is detected
10	Actor	User
11	Precondition	Current dataset is present on the client
13	Result	KPI's are visible in more detail or in mor general fashion
14	Main scenario	User detects devotion during KPI review Two possibilities • Drill up to view more detailed KPI • Drill up to review aggregated KPIs
15	Alternative scenario	When the user is reviewing the overview drill up is not possible If the most detailed view is visible drill down will be not possible
16	Exceptional scenarios	Description of exceptional scenarios
17	Qualities	Comply with FASMI requirements (see section 2.3.4)

Table 4.3: Use case 2: Drill up/down

By making complex OLAP operations available on mobile devices, the barrier to use this technology is lowered significantly. This supports stakeholders in their decision-making process and finally enables faster and more accurate outcomes.

4.2.5 Use case 3: History - Change the time horizon

Use case 2 allows users to identify potential causes of KPI's deviations. The next logical step is to look at previous time frames to categorize deviations either as single occurrences or as structural problems. Depending on the frequency a deviation occurs, appropriate measures have to be taken.

4.2.5.1 AS-IS analysis

As mentioned before, currently there is no homogeneous reporting in place, resulting in different formats and scopes for diverse segments and departments.

#	Section	Content/Description
1	Identifier	UC3
2	Name	Change time horizon
4	Priority	High
5	Criticality	High
6	Source	Stakeholder, current reporting templates
8	Description	Provide different time horizons and time spans for KPI and lot size views
9	Triggering Event	Deviation is detected
11	Precondition	All data is present on the mobile platform to perform a time horizon change
12	Post condition	
13	Result	Initial view
14	Main scenario	Deviation is detected during KPI review • Data from previous periods gets reviewed • Measures get triggered (see use case 6) • After the operation the app gets closed or different use cases get triggered
15	Alternative scenario	-
16	Exceptional scenarios	Data is not or only partially present • Corresponding sections should be blank
17	Qualities	Comply with FASMI requirements (see section 2.3.4)

Table 4.4: Use case 3: Change the time horizon

Another challenge is the distribution of these reports which is currently organized document-based.

4.2.5.2 Potential improvements

When data is available in the form of an OLAP cube, it is fairly easy to perform complex transformations on the data set. Users can execute e.g. the drill across actions and compare data from different time frames and from different production segments with each other.

4.2.5.3 Resulting use case

The resulting use case is documented in table 4.4.

Browsing the data history helps to identify and categorize issues in the production process. This identification step is important in approaches like the continuous improvement process or Kaizen.

4.2.6 Use case 4: Tracking of topics

Again, use case 4 builds upon the previous findings. Following identification of topics, these have to be documented and monitored accordingly.

4.2.6.1 AS-IS analysis

According to the overall project goals, management tasks should be supported by the MIS. As of now, the tracking of topics is split over several mediums and information sources. Topics get identified and documented e.g. by shift supervisors or are reported during meetings with senior staff. There is currently a lack of a defined process for the way topics are recorded and further handled. Use case 4 will address this challenge and provide a possibility for continuous topic detection.

4.2.6.2 Potential improvements

The current situation can be improved by documenting topics in a structured way. Topics can be linked with a certain KPI and one or multiple occurrences.

4.2.6.3 Resulting use case

With the introduction of a structured tracking mechanism, uniform visual representation provides various benefits (see table 4.5). Topics occurring multiple times can be linked together and past topics can be searched for similar or equal incidents. This helps to reduce the efforts necessary for fixing problems and supports the increase of overall production plant efficiency.

4.2.7 Use case 5: Tracking of measures

Mechanical fabrication, production or assembly can be brought to a very high level of reliability but sometimes it is inevitable that topics occur. When problems arise it is important to identify them correctly (see use case 4) and set the appropriate measures.

#	Section	Content/Description
1	Identifier	UC4
2	Name	Track topics and problems during operation
4	Priority	Medium
5	Criticality	Medium
6	Source	Stakeholder
8	Description	When deviations of KPI's or lot sizes are detected measures can be viewed and tracked
9	Triggering Event	Deviation is detected
11	Precondition	An actual dataset is present on the device
12	Post condition	-
13	Result	An overview of taken measures is viewed for the current organization unit of the production plant
14	Main scenario	Current KPI's are viewed (see use cases 1,2,3) • Deviation is detected • Deviation gets reviewed (see use case 6) • If present topics and problems can be viewed accordingly
15	Alternative scenario	-
16	Exceptional scenarios	No data is available (refer to use case 7)
17	Qualities	-

Table 4.5: Use case 4: Topics

4.2.7.1 AS-IS analysis

As for topics there is also no central hub for measures taken for topics. Problems and their resolution are assigned to a responsible person which tracks the execution of all activities involved during the process.

4.2.7.2 Potential improvements

The MIS platform should support a comprehensive topic and error resolution process. Topics will be assigned to a responsible person. Additionally, it will be possible to assign priorities and deadlines to issues. These issues will be displayed using a color code helping to distinguish between critical and non critical topics.

#	Section	Content/Description
1	Identifier	UC5
2	Name	Track measures
4	Priority	Medium
5	Criticality	Low
6	Source	Stakeholder, current reporting
8	Description	within every hierarchy a set of defined measures should be displayed
9	Triggering Event	The
11	Precondition	A detected deviation in lot size or KPI
12	Post condition	-
13	Result	An overview of all
14	Main scenario	Current production plant overview in any hierachy level is visible: • Deviation of lot size or KPI is detected over longer period • Current measures for the topic or problem can be viewed • A status of the measures are visible allowing to see the progress
15	Alternative scenario	-
16	Exceptional scenarios	No measures are defined • Show a tutorial how to create scenarios
17	Qualities	-

Table 4.6: Use case 5: Track measures

4.2.7.3 Resulting use case

Within this use case a measure tracking should be designed. Topics should be linked with issues and their resolution will be documented completely (see table 4.6). Such a consistent topic solving approach acts as a documentation and knowledge source for future topic solving tasks.

4.2.8 Supporting use case 6: Data preparation

During elicitation it became evident, that within the scope of the thesis it was not feasible to implement a fully automated solution for the demonstrator. For the concept implementation it is satisfactory that some steps are still carried out manually. Supporting use case 6 relates to the generation of report files (see section 4.5).

This part of the reporting process will include all tasks to gather the information needed for the generation of an up to date report file. As described before,

#	Section	Content/Description
1	Identifier	UC6
2	Name	Data preparation
4	Priority	Medium
5	Criticality	High
6	Source	System
8	Description	Actual KPI's are available on the mobile device
9	Triggering Event	Every needed dataset for generating a valid report data set is present or a predefined time is reached.
11	Precondition	All Datasets needed for the report file are present or the time boundary is met
12	Post condition	Report file is sent
13	Result	Report file is sent successfully to all receivers and are available on the mobile device
14	Main scenario	Daily report figures are received Data gets integrated into a predefined report file format Report file is validated Report file is sent to all
15	Alternative scenario	Discerption of alternative scenarios or trigger events
16	Exceptional scenarios	No data is sent • Clients will display the data that got imported from the most recent report file
17	Qualities	Robust system design: Take measures for report file validation

Table 4.7: Use case 6: Data preparation

the information flow to the demonstrator application will have some manual steps involved. The task must be feasible within standard software available on workstations on site.

4.2.8.1 Resulting use case

The resulting use case is documented in table 4.7. It covers all tasks and activities needed to feed the application on the mobile client with the actual data.

Activities within this use case can cover data acquisition, the correction of faults and the addition of data to the report file.

4.2.9 Supporting use case 7: Data import

Closely tied to use case 6, this use case describes the data import mechanism of the client. On the first application start, only test data will be displayed on the

#	Section	Content/Description
1	Identifier	UC7
2	Name	Data import
4	Priority	High
5	Criticality	High
6	Source	System
8	Description	The report file gets loaded from an email sent from the central mailing list
9	Triggering Event	User opens an email containing an attachment holding the data required for operation
10	Actor	Mobile application user
11	Precondition	All Datasets needed for the report file are present or the time boundary is met
12	Post condition	Report file is validated Report file is copied to internal storage Content of the current report file is loaded Old report file is deleted
13	Result	Actual data is displayed on app start
14	Main scenario	Email containing an attachment with current data is sent to all application users. • All users receive the email containing the data • The users opens the attachment • Applications opens and actual data is displayed
15	Alternative scenario	First application start • Provide a demo report file for displaying test data on the first application start
16	Exceptional scenarios	Data file is corrupted or empty • Prompt meaningful and human readable error message
17	Qualities	Security constraints: • Report file must be stored encrypted on the mobile platform

Table 4.8: Use case 7: Data import

client. Actual data will only be displayed if a report file is delivered correctly to the client.

4.2.9.1 Resulting use case

Use case 7 includes all activities related to the import of the data on the mobile application (see table 4.8). The process should be easy to use and intuitive. The email sent to the client should contain a short summary for the usage describing how to install the application as well as the import process of the data.

4.3 Requirements elicitation and identification

This section will sum up all gathered requirements within the classification of hardware, functional and non functional requirements. Two main resources were used to gather information for the resulting requirements lists: The first source were interviews of potential users as well as stakeholders, which gave good insight on required functionalities of the resulting system. Another important source was the internal developer guide of Audi which mostly covered technical and security aspects.

4.3.1 Hardware requirements

During the elicitation phase several demands were raised for the hardware platform. These were mostly concerning topics such as usability, stability, reliability, cost and connectivity in no particular order.

- Device must be portable and should be easy to carry
- Platform device should feature touch input
- The device must have wireless transmission capabilities to transfer information required for operation
- If possible existing hardware should be used
- In case new hardware has to be acquired, cost for the extension of current or the investment in new hardware should be kept to a minimum
- Hardware should only contain software from verified sources to prevent data breaches
- The display should be of high resolution to fit as much information as possible on the screen
- Capability of storing data ranging back for days, weeks, or years
- Enough computational resources for fast operations

4.3.2 Functional requirements

The following list will present the findings for the functional requirements gathered during the elicitation phase.

- Data should be available offline (without a connection via data or wifi network)
- Displayed data must be accurate and updated at least daily
- It should be possible to attach documents or pictures to a deviation
- The contact details of an assignee for a given measure should be linked
- The application must be compatible with the current mobile platform of target users
- The system must work in compliance within the rules applicable for data generation and storage of AHM

- The application should provide a mechanism for restricted access for certain information
- The system should be responsive to user input via touch gestures
- Data must be available within the company wireless network as well as in public networks

4.3.3 Non-functional requirements

In this section all non-functional requirements will be listed and discussed as presented in the theoretical part in section 3.2.3.1. The requirements were defined during the open interviews with the most important stakeholders of the project.

Requirements for the client:

- If possible make use of gestures for navigation rather than keyboard or button input
- When possible, the use of existing platforms should be favored
- Sensitive data must not be disclosed to 3rd parties in case of device loss
- Corporate design of AHM must be met
- Information should be as detailed as possible
- Intuitive, uncomplicated use
- A minimum amount of training is required for new users to operate the client as well as the backend
- Application should be scalable for the use in different segments and for the future also on different sites
- Application must be free of proprietary libraries with incompatible license
- Cost of third party license should be avoided (for demonstrator purpose)
- Provide a set of predefined OLAP queries within the demo application
- Comply with the Apple ™ *iOS Human Interface Guidelines* (Apple Inc., 2015a)
- Responsive user interface

Apparently, the requirements gathered in the sections 4.3.1, 4.3.2 and 4.3.3 are quite ambiguous. This stems from the fact that they have been extracted from interviews and documentation. For a requirements documentation in better quality, this first inquiry will be transformed into a complete requirements list in the next section.

4.4 Requirements analysis and specification

In this section the requirements gathered will be further investigated for feasibility and potential conflicts. As a result the final requirements list will be provided in a formal structure suggested by the SOPHIST framework (see theoretical part section use case specification).

4.4.1 Requirements documentation

The structure for documenting the final sanitized requirements will be carried out by the suggested scheme by Sachs (2014). An example requirements index notation will be as follows: **FR-OBL-04**

The structure of the index is defined as follows:

Format: <type of requirement>-<servility of the requirement>-<index>

Type of requirement with possible values:

- **FR** for functional requirements
- **NFR** for non functional requirements
- **HR** for hardware requirements

Severity of the requirement:

- **OBL** for obligatory
- **OPT** for optional
- **FUT** for requirements planned for future versions

Index: Is a unique identifier tied to one requirement for the entire requirements engineering process.

With this indexing format all requirements can be uniquely defined and managed starting from the elicitation up to requirements management.

4.4.2 Terms and definitions

Table 4.9 gives an overview of all terms and definitions used during the formal requirements engineering documentation. It should help preventing misapprehensions and potential ambiguousness.

EXPRESSION	DESCRIPTION
Application (App)	Application installed on the mobile client
Client	Is synonymous with the application definition
System	Everything connected to the or from the application on the mobile client is part of the system including features for data preparation and processing
Decision maker	Person operating the application on the mobile device during day-to-day business
Demonstrator	The visual appearance of the mobile application
Backend	All facilities, programs and tasks used to deliver the data to the mobile application
Maintainer	Person responsible for report file generation
Report file	File with the file name extension .report containing all information for the operation of the application

Table 4.9: Terms and definitions used during RE specification

4.4.3 Functional requirements

Obligatory present and future requirements:

FR-OBL-01	When the user starts the system for the first time test data should be viewed.
FR-OBL-02	When the user starts the application for the first time using a file the actual data set should be viewed.
FR-OBL-03	At any given point in time the system must be able to start with an report file.
FR-OBL-04	When the user starts the application for the first time the system should provide a list of key performance indicators.
FR-OBL-05	When the top KPI view is visible the system shall provide the the user with the ability to perform a drill down operation.
FR-OBL-06	When the bottom KPI view is visible the system shall provide the user with the ability to perform a drill up operation.
FR-OBL-07	When any KPI view is visible the system shall provide the user with the ability to perform a drill across operation
FR-OBL-08	When the bottom KPI view is visible the system shall provide the user with the ability to perform a drill up operation.

FR-OBL-09	When a KPI view is visible expect the top or bottom ones the system shall provide the suer with the ability to perform a drill up and down operation.
FR-OBL-10	At any given point in time the system shall provide the user with the ability to view topics assigned to a given view including detailed information.
FR-OBL-11	At any given point in time the system shall provide the user with the ability to view deviations assigned to a given view including detailed information.
FR-BOL-12	At any given point in time the backend shall allow the maintainer the possibility to add and update information for existing and new report files.
FR-OBL-13	At any given point in time the backend shall provide the maintainer the possibility to send updated data to the devices running the application.
FR-OBL-14	At any given point in time the system shall provide the user the ability to import new report files.

Optional functional requirements:

FR-OPT-15	At any given point in time the system shall be able to start with test or actual data.
FR-OPT-16	At any given point in time the system shall provide the maintainer the possibility to attach arbitrary files to measures.
FR-OPT-17	At any given point in time the system will allow system to define a personal identification number.
FR-OPT-18	At any given point in time when the application is started the system will present a lock screen asking the user to enter his personal identification number (PIN).

Future functional requirements:

FR-FUT-19	At any given point in time the system should allow the user to edit the description and status of a measure.
FR-FUT-20	At any given point in time the system should allow the suer with the ability to upload documents and images to a server.

4.4.4 Non functional requirements

Obligatory present and future non-functional requirements

NFR-OBL-21	When the users starts the application the system shall be ready for user input instantly.
NFR-OBL-22	At all times the application shall respond to user interactions instantly.
NFR-OBL-23	At all times the application shall not provide data to unauthorized users.
NFR-OBL-24	At all times the system shall provide the user with a user interface in conformance to the design guidlines
NRF-BOL-25	At all times errors occur the application shall provide the user with meaningful and human readable error messages.

4.4.5 Hardware requirements

Obligatory present and future hardware requirements:

HR-OBL-26	At all times the system shall be able to connect 3G or wifi networks.
HR-OBL-27	At all times the system shall be able to provide a display with a pixel density of at least 300ppi.[9]
HR-OBL-28	The system should be able to last at least 24 hours without recharging.

4.4.6 Conflicts

All requirements listed int the sections 4.4.3, 4.4.4 and 4.4.5 were cross checked for potential conflicts.

During the review no conflicts were detected between the obligatory requirements and therefore the demonstrator design could be started.

4.5 Demonstrator architecture and design

Based on the requirements engineering conducted during this thesis a demonstrator application with a backend is implemented. The system will respect all requirements identified and will also be validated regarding its functionality, usability and performance.

[9]pixels per inch

4.5.1 Development platform

Within the constraint that the application must run on the devices handed out to the management staff at Györ the target development environment was an iPhone 5 (see figure 4.3) produced by Apple Inc. It fulfills all the software and hardware requirements particularly by means of the display, security and connectivity.

Important hardware features (Apple Inc., 2013) relevant for MIS:

- Operating system: iOS 8.1 (as of October 2014)
- Display density: 326 ppi
- Display resolution: 1136 x 640 pixels
- Storage: 16 up to 64 GB
- Multi-touch touchscreen display
- Ambient light sensor
- Display size (diagonal): 4 inch
- Display aspect ratio: 16:9 aspect ratio
- Display technology: LED backlit IPS TFT LCD
- Display contrast: 800:1 contrast ratio
- Display brightness: 500 cd/^{2} max. brightness
- Connectivity Mobile: UMTS/HSPA+/DC-HSDPA, GSM/EDGE, FDD-LTE, TD-LTE
- Connectivity WIFI: 802.11a/b/g/n WLAN

Figure 4.3: iPhone 5s front view, CC-BY-SA-3.0 (Zach 2013) featuring a retina display

The display size and resolution fit the requirements for the detailed display of visual information and in combination with the *Cocoa Touch framework* (Apple Inc., 2014) builds an ideal platform for a management information system. Furthermore, the device features all required connectivity standards, in particular WIFI and 3G.

4.5.2 Architecture variants

Within the set constraints, three potential architectures have been identified which will be described and analyzed in detail.

- Native iOS application
- Offline web application
- Online web application (needs active connection to the server)

4.5.2.1 Native iOS application

Native iOS applications include all software products developed within the tools provided by Apple™. This includes all apps developed with the programming languages Objective C or SWIFT, utilizing the XCode IDE (integrated development environment).

In general, native apps have the advantage that all hardware and software features provided by the device manufacturer are natively supported. Usually they can be used within the native IDE and have guaranteed support for a certain period of time. This is especially important concerning security features and to applications that deal with sensitive or security-relevant data.

Regarding software quality, XCode™ provides a great variety of tools to debug applications, verify their functionality and test the user interface. (Apple Inc., 2015c)

Native apps provide the best runtime speed due to the fact that compilers optimized for the platform hardware and software are used.

Another factor in favor of native apps is that the number of screen resolutions as well as densities is reasonable. Therefore the effort of supporting all screen formats is manageable.

Apps can be distributed either through the official Apple™ app store or independently with the so-called ad hoc distribution methods. When apps are distributed through the app store, they must be reviewed by Apple™ which usually lasts for one to two weeks. The distribution of native applications outside the Apple™ app store is possible within the apple developer license. With the ad hoc distribution methods no review is necessary but it is only allowed for a limited amount of devices.

4.5.2.2 Offline web application

This kind of application utilizes the fact that most modern operation systems native languages include the feature to display webviews within their native applications. The data needed to display the websites can be cached or stored locally, allowing the app to work without an active internet connection. To date, several frameworks exist for simplifying the task of cross platform apps with apache cordova being the most well-known[10]. So in theory it is possible to develop one application and allow it to run on several platforms equally.

With the use of offline web applications also some drawbacks arise. Not every feature is available on all platforms (The Apache Software Foundation, 2015).

Apart from the functional limitations, speed and performance of web applications are an issue. It is necessary to build a DOM tree for representing the app data model and user interface within web view. These structure inherently makes it not as fast and fluent as native applications. There are approaches to improve the application speed drastically e.g. the react project by Facebook (Facebook Inc., 2016).

As the user interface is browser based, it brings the challenges of testability for both user interface and functional features. Other challenges include the lack of security features and the fact that the application design is not customized for every platform supported.

For the distribution of the applications the same restrictions as for native applications apply.

4.5.2.3 Online web application

Online web applications provide a similar feature set as mobile websites with the addition that mobile operating systems feature support for functions like shortcuts on the home screen. (Apple Inc., 2015b)

One of the most important advantages is the distribution, which does not need app store submissions or reviews. In contrast to native applications and offline web apps, this type of application require an active internet connection to work

[10] Apache cordova is the free open source version of the Phonegap framework originally developed by Nitobi and later acquired by Adobe

Criterion/ Architecture	Native iOS	Web app (offline)	Web app (online)
Usability	●	●	◐
Performance	●	●	○
Compatibility	◐	◐	●
Data Security	●	◐	◐

Table 4.15: Architecture evaluation. Legend: Full circle . . . fully satisfied, Partially filled circle . . . partially satisfied, emtpy circle . . . not statisfied

correctly. Slow or unstable internet connections may also have a negative influence on the usability and performance.

By means of user interface and security features the same implications as for offline web applications apply.

4.5.3 Architecture comparison

Table 4.15 highlights the potential benefits and drawbacks of the suggested architectures and their corresponding technologies.

The criteria listed above will be briefly discussed regarding their compliance with the requirements:

- **Usability:** Native apps provide the best usability allowing all hardware and software features to be used. They are also best in terms of performance, responsibility and integration with the given operating system.
- **Performance:** Mobile websites inherently have the worst performance in comparison as all data is stored on the server and has to be transmitted to the client every page load (caching is still possible). Mobile web apps have better performance as they cache the website locally but still the DOM model is very expensive to build and display. Native apps provide the best performance.
- **Compatibility** Mobile websites are best regarding this criterion as hardly any adoption is needed for the various platforms. Another benefit is that mobile web pages can be ported to traditional websites if needed. Web applications can be built for cross platform use applying the techniques presented above. Native apps in contrast are not portable.
- **Security** By means of security features, native apps as well as offline web apps provide the best support. Mobile websites applications provide less security as there are various attack vectors possible.

4.5.4 Architecture choice

Bearing in mind the presence of the iPhone devices at the production site the decision fell to the native iOS implementation. The technological risk is low as that all gestures provided by the device can be facilitated by using the native objective C programming language API. Further the design of a native application is consistent with other applications used at Audi Györ and within the whole Audi Group.

4.5.4.1 System, context and environment

Within the requirements engineering process the differentiation between system, context and environment represents one of the key tasks. For this specific implementation the respective elements belonging to these categories were identified as follows:

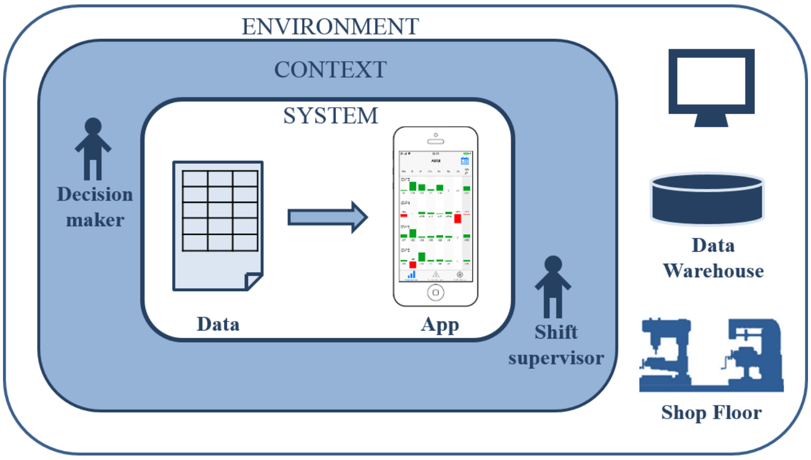


Figure 4.4: System, Context and Environment

- **System: All parts and modules comprising the system itself**
  - Application (app) displaying the data on the hand-held devices
  - Excel providing the toolset to prepare data for the application
- **Context:** Components and staff in direct contact with the system
  - Users operating the application
  - Assisting staff using Excel to prepare the .mis file
  - Email server and delivery system used to transport the data
  - IT infrastructure: Wireless network

- **Environment:** All important components influencing, but not in direct contact with the system
  - Shop floor data presentation
  - Data warehouse, ERP, CAQ
  - All systems in the production plant providing and storing data

An illustration of the described system can be seen in figure 4.4.

4.6 Proof of concept implementation

In this chapter the proof of concept implementation of the master thesis will be discussed. All gathered requirements will be respected and as a result the demonstrator design will be the outcome.

4.6.1 Software development platform

- Development platform target: iOS 8
- Programming environment: XCode 5
- Used Programming Language: Objective C
- Data delivery using a *.mis file format and the email API of iOS

4.6.2 Information gathering, processing and delivery

To enable the mobile application to access the actual datasets for displaying, a concept was needed to gather and process them in such a manner to be able to present it on the client.

4.6.2.1 Possible data delivery concepts

Initially, three possible solutions were drafted to provide the data. In all three scenarios the data need to be added to the data source manually, as the infrastructure to provide it automatically was not available at the time due to contractual and license issues. For that reason automatic feeding of the data source was discussed and postponed to the development of the productive client.

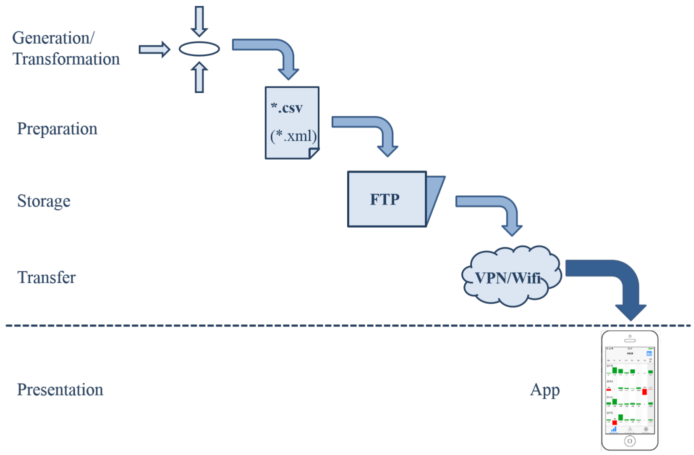


Figure 4.5: Proposed data delivery concept using FTP hosting xml or csv files containing the data

RESTful webservice The first architecture design possibility was a RESTful[11] service (Fielding, 2000). For this approach a server infrastructure is required for installing and running the service. Two frameworks were considered for the implementation, namely NODE.js[12] (Javascript) and the more complex Apache CXF[13] (Java).

FTP Another possibility is an FTP (file transfer protocol) server, that enables hosting of files (see figure 4.5). As for the RESTful concept the systems requires an server infrastructure running the service. The setup of the system is uncomplicated and various open source implementations are available (e.g FileZilla Server[14]).

Mail The third proposal is the delivery of the data via the already existing email network (see figure 4.6). Data for the client will be included as an email attachment and sent to the client platform. On the client the email can be viewed and the data can be visualized.

4.6.2.2 Evaluation of proposed data delivery variants

All three variants need manual user input to provide the data on the back end side, so this factor is not considered for the decision. In terms of usability the REST and FTP proposals are superior over mail due to the fact that these variants need no manual user input on the client side to import the data. Considering that

[11]Representational state transfer, offering an implementation- independent abstracted interface over HTTP
[12]`https://nodejs.org/en/`
[13]`https://cxf.apache.org/`
[14]`https://filezilla-project.org/download.php?type=server`

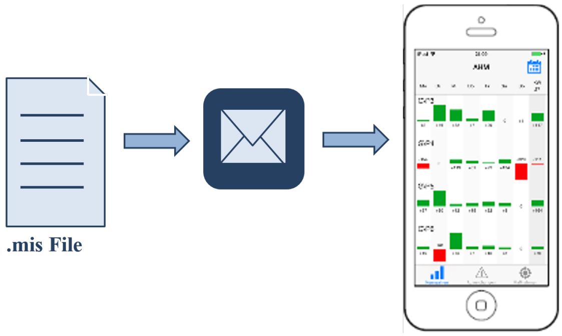


Figure 4.6: Data delivery concept via email

for future implementations it would be beneficial to provide bidirectional data flow from and to the app. Therefor the RESTful API would be the best option as it is the only variant providing a communication channel back to the server. In regards to data security all three variants provide reasonable security, as REST and FTP can communicate via SSL[15] and emails are encrypted as well. One drawback of the email solution is, that data can be sent easily to third parties outside the company network. The initial setup cost and risk of the variants vary significantly between the solutions. Email needs no setup at all, in contrast to FTP and REST which need a running server. In addition the FTP server needs to be configured and the RESTful server requires a custom implementation.

After comparing the three variants (see figure 4.16) and considering the pros and cons the decision was made to implement the third proposal, the delivery via email.

Criterion	Rest API	FTP	Mail
Usability	●	●	◐
Extensibility	●	◐	◐
Data Security	●	●	◐
Techn. risk.	○	◐	●

Table 4.16: API evaluation. Legend: Full circle . . . fully satisfied, Partially filled circle . . . partially satisfied, emtpy circle . . . not satisfied

[15]Secure socket layer

4.6.2.3 Implemented architecture

The architecture presented in figure 4.7 was implemented. Data is extracted with the help of existing reports from the data warehouse and manual input. Furthermore, information on current measures and deviation will be put together into a table calculation program (Excel). Therefore, the already in place Excel

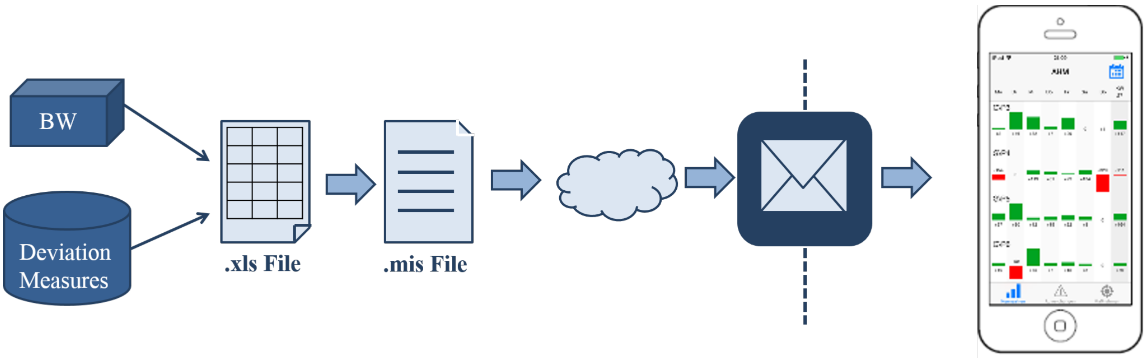


Figure 4.7: Implemented architecture for the demonstrator

files for daily and weekly reporting were extended in functionality, towards the automated generation of a transfer file.

The tables feature a predefined layout of the data with separate sheets for each department. One sheet contains the respective KPIs values data set for the desired time frame, including the actual and the target values. The design of the sheet is suitable for adding and removing departments and KPI's on demand. With the help of Excel a *.mis file can be generated, which contains a SQLITE[16] database. SQLITE is a file based and features a relational database model, implementing a subset of the SQL standard(ISO/IEC, 2011). The SQLITE format is supported by the iOS platform, additionally a plugin for Excel is available for reading and writing the databases.

As soon as all relevant data is included within one Excel file, the *.mis file creation can be triggered and it is subsequently sent to the clients via email. When the file is received and opened on the client the actual data can be displayed on the app. An example of the email and the opening process can be seen in figure 4.9. The file with the actual data is pushed in a daily basis to all clients using a mailing list.

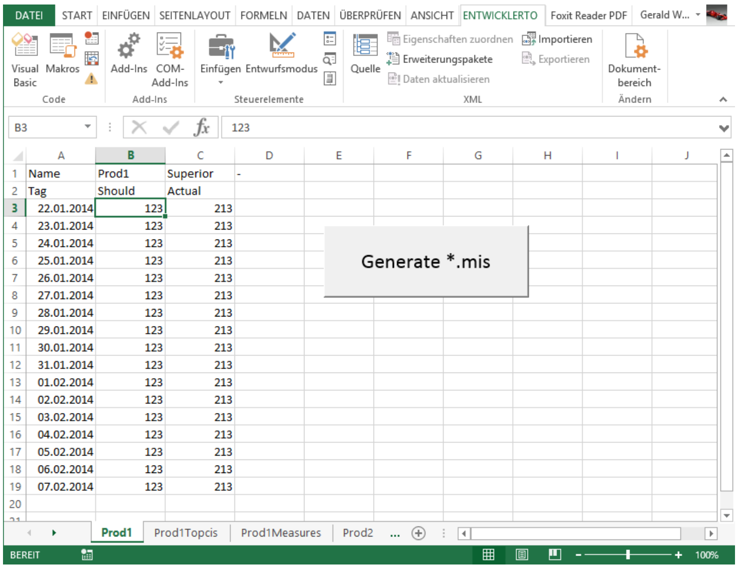
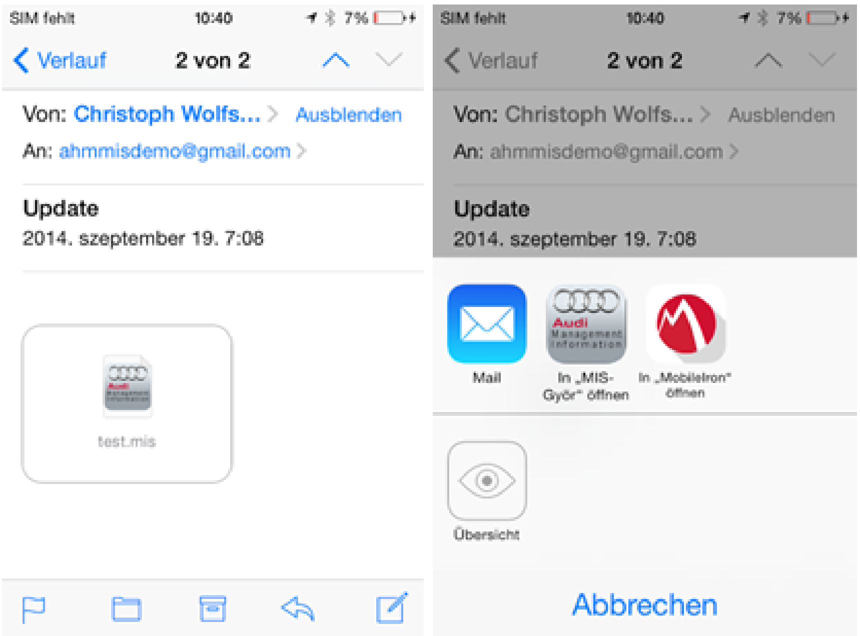


Figure 4.8: Excel file for data gathering



(a) Overview (b) Section

Figure 4.9: Data import concept: In the left picture the email delivering the report can be seen. The right picture shows the app chooser dialog

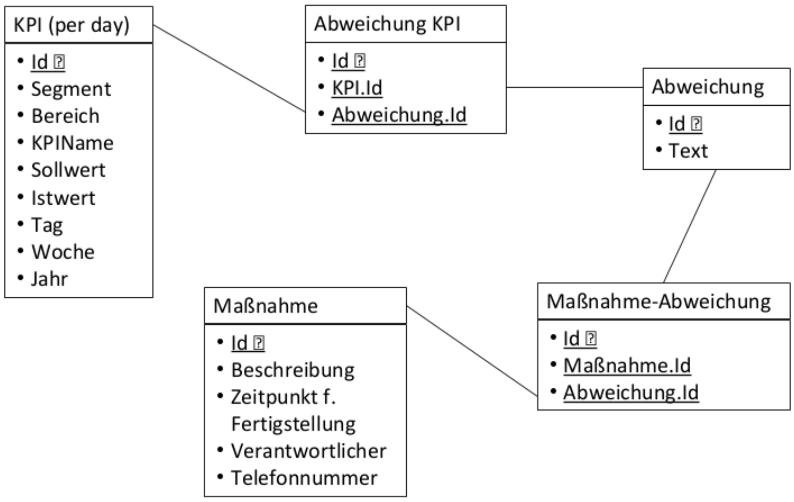


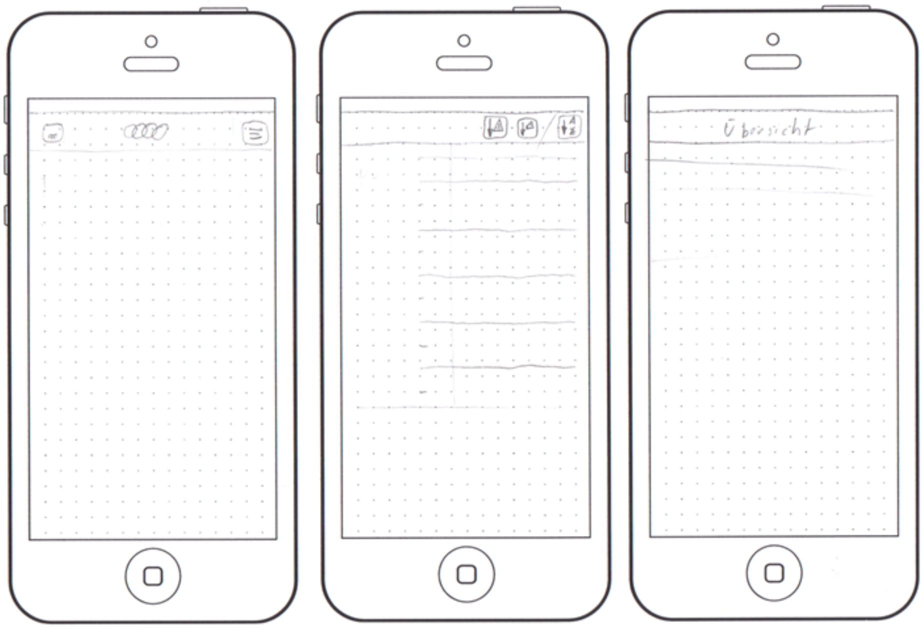
Figure 4.10: *.mis file format structure

4.6.2.4 Structure of the report data file

Data within the file contains the information of the last few weeks to display on the client. It contains KPI's target and actual values as well as tracked deviations and measures. The structure of the relational database is illustrated in figure 4.10. As mentioned before the Excel sheet can contain a varying amount of datasets. Departments and their respective KPI's as well as the data for measures and topics can be dynamically added and removed. These changes are reflected during the export process and the *.mis file is generated accordingly.

4.6.3 User interface and control

The user interface was designed by following the HIG (human interface guidelines) issued by Apple. It describes all relevant principles regarding design, control, behaviour and feedback as well as security constraints. In order to submit the application to the App Store, it has to be consistent with the guidelines and all terms of service have to be met.



(a) Overview (b) Section (c) Section

Figure 4.11: Initial pencil mockups

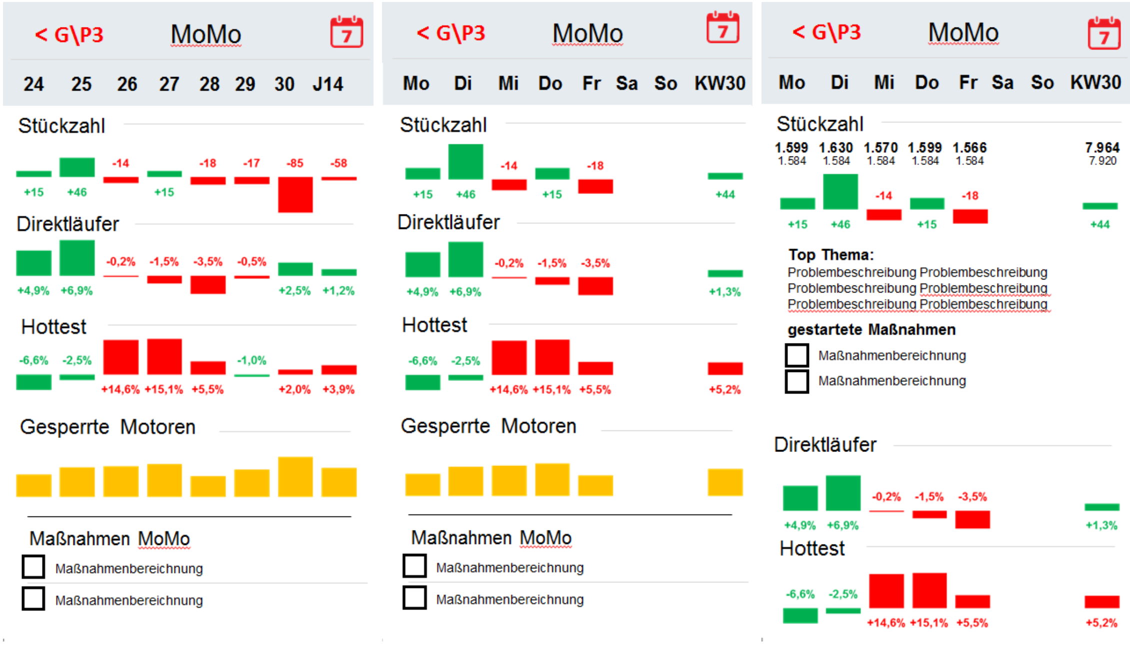
4.6.3.1 User interface

The main objective was to present relevant data, namely KPIs and topics, in a structured and informative way. The design phase was split into three phases with increasing complexity of the drafts. First, paper mockups were drafted[17], containing the main elements for presentation and navigation. This enabled testing of potential data representations suitable for the intended display size. Additionally this step was important to get an first impression of proportions of screen layouts. Examples of the resulting mockups can be seen in figure 4.11.

Next, UI models were generated using sketching software (see figure 4.12). These drafts were then used as a basis for discussions with stakeholders, in order to include their feedback already in an early phase. The models gave a good insight on several representation possibilities considered for future implementation.

Subsequently, the various proposed designs were analyzed in detail for their compatibility to be implemented within the target development platform. Furthermore,

[16] `https://www.sqlite.org/`
[17] Credit for templates: `interfacesketch.com`



(a) Overview (b) Section (c) Section

Figure 4.12: Digital drafted mockups

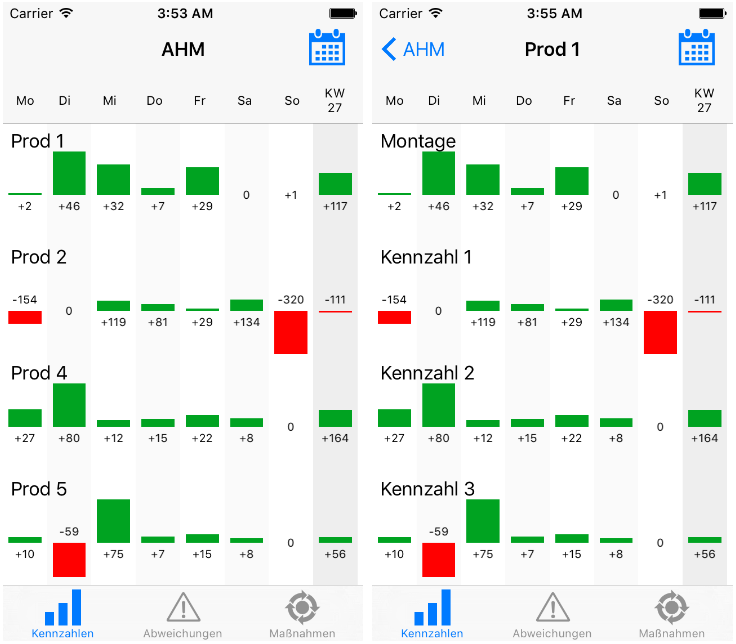
the feasibility to combine different designs in a single application was evaluated. Screenshots of the implemented system can be seen in chapter 4.6.4.

4.6.3.2 Control

In order to work correctly the demonstrator has to be compatible with the dynamic *.mis file. The format features a dynamic number of nested departments and their respective KPI's. Thus the demonstrator can view an arbitrary number of departments and KPI's. This functionality was achieved by implementing a recursively callable view controller, capable of displaying the data.

4.6.4 Features

In this section an overview of all implemented features with a special focus on the designed UI will be presented.



(a) Overview (b) Section

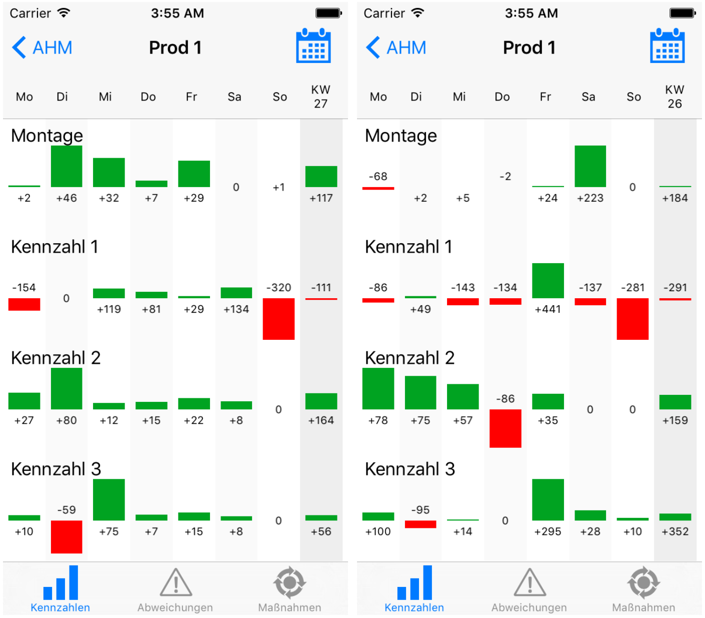
Figure 4.13: Drill down example: In the left picture the AHM overview is visible and in the right the KPI s of the subsegment can be seen.

4.6.4.1 Drill up and down

The drill up and down operations were implemented using the recursive call of the same view controller (see figure 4.13). This view controller is embedded within the navigation controller and a tabbed bar (used for the measure and deviation views).

4.6.4.2 Past KPI data

In figure 4.14 the implemented feature for displaying past KPI data can be seen. Depending on the dataset present in the currently loaded *.mis file any time point in the past can be viewed. When older data is required, past mis files can be opened and viewed. The week can be changed by swiping left and right.



(a) Week 27 (b) Week 28

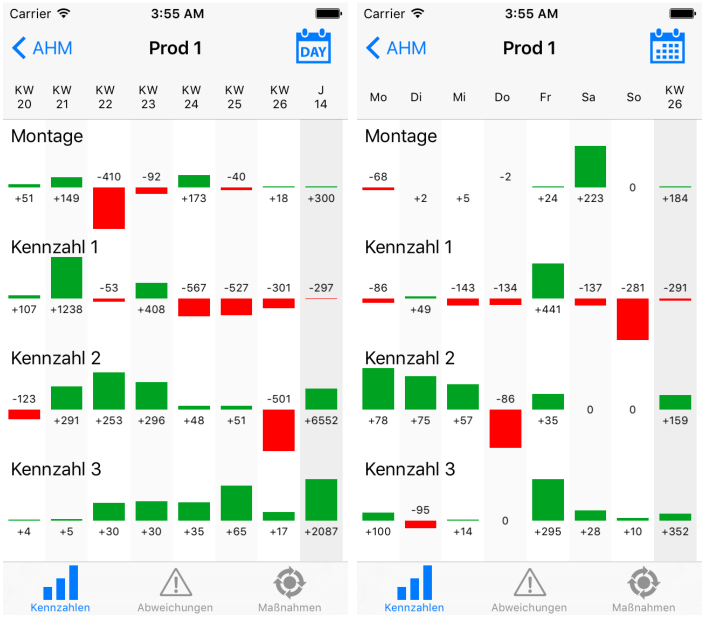
Figure 4.14: KPI history: In the left picture the week 27 can be seen whereas the right picture views figures of week 28.

4.6.4.3 Dynamic time horizon

Within every KPI view the user has the possibility to choose from two different implemented time horizons (see figure 4.15). Initially the standard view is configured to show the data on a daily basis. With the calendar icon located on the upper right position of the display, the time horizon can be changed to a weekly view. The rightmost column represents the aggregated KPI value for the whole year in the weekly view and the weekly aggregated KPI in the daily view.

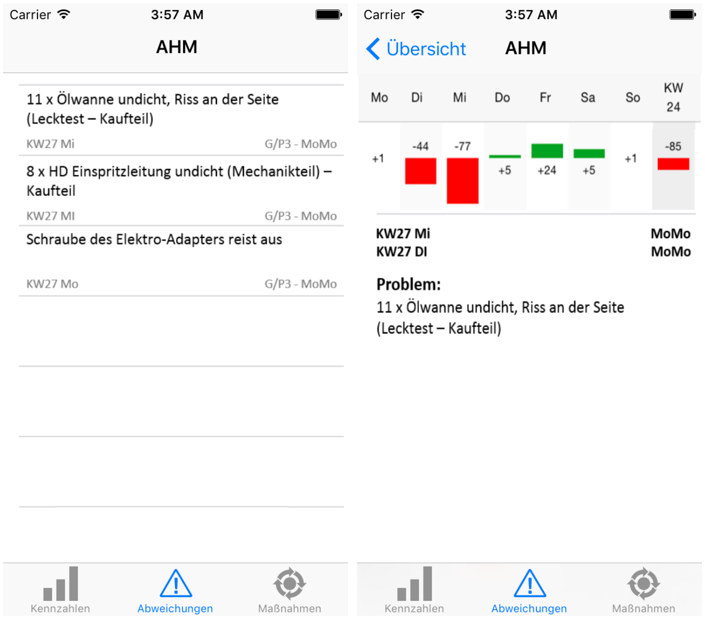
4.6.4.4 Overview section topics

On all given KPI views the user has the possibility to view topics occurring during production and manufacturing in the respective time frame (see figure 4.16). This view is accessible with the tabular controller on the bottom labelled "Themen". A list of all current topics is shown providing information on the time of occurrence and a description of the problem. By selecting one entry the user can view a more detailed description of the specific topic.



(a) Daily view (b) Weekly view

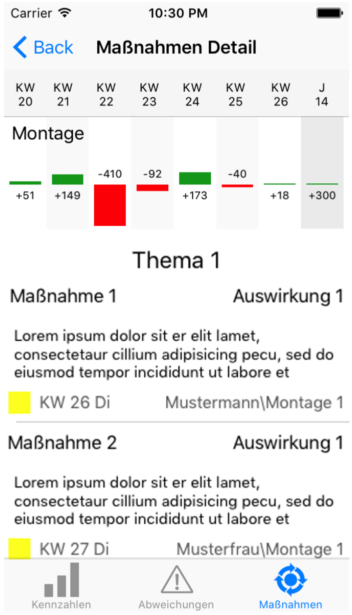
Figure 4.15: Daily and monthly view



(a) Topic (b) Detail

Figure 4.16: Topics overview and detail view

(a) Measure overview



(b) Detailed measure view

Figure 4.17: Measure overview (left) and detail view (right)

4.6.4.5 Overview section measures

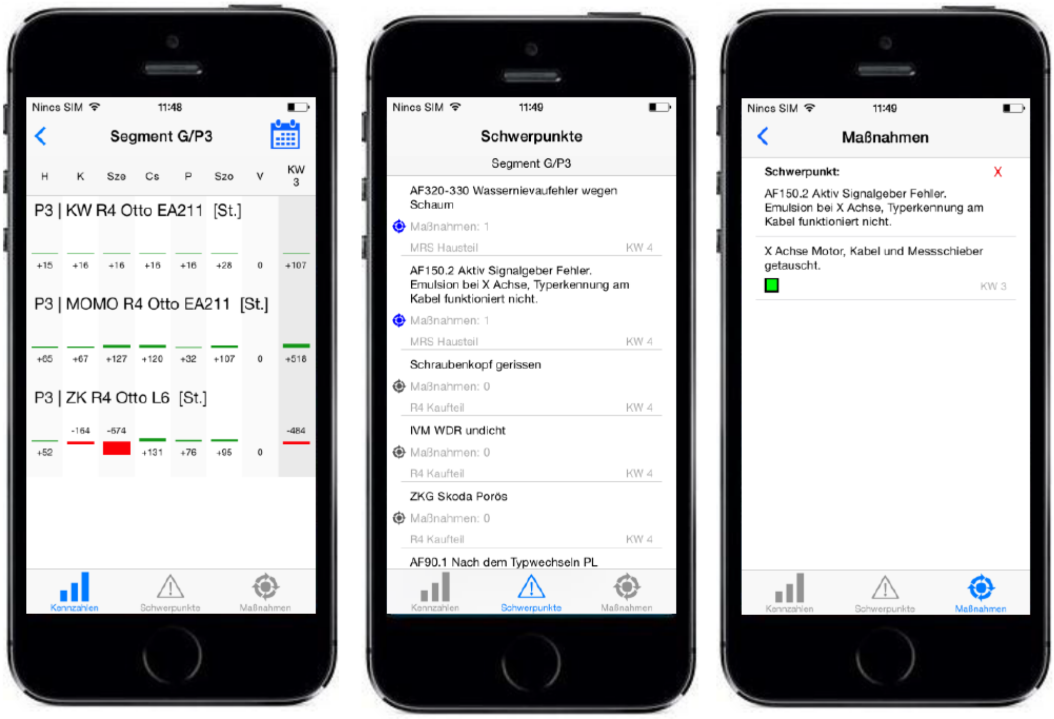
As for topics for every KPI view also a overview of measures can be accessed with the tab controller on the bottom. Here a list of all taken measures is shown (see figure 4.17). Furthermore a color code (red, yellow, green) provides information on the progress of the set measure and the responsible person is named. By selecting one measure details are given regarding the set subtasks.

5 Evaluation and conclusion

The goal of this thesis was to support decision making in an Audi Hungaria Motor production plant by implementing a management information system on a modern mobile platform. In this chapter the whole development process of the prototype will be evaluated with respect to planning and implementation, degree of fulfillment of initially defined goals and immediate applicability. In retrospect, requirements engineering represented a valuable tool to achieve the desired overall goal. By using this technique all requirements were elicited and defined unambiguously which prevented misconceptions and involved all stakeholders. In particular the AS-IS analysis with interviews with stakeholders as well as on site observations as especially beneficiary for the success of the whole process. Built on the defined requirements the iOS development platform was chosen. As there was only limited access to the data-providing systems the implemented proof of concept demonstrator was the best possible solution within the given constraints. Following the implementation itself, unit tests as well as instrumentation tests were written and performed in order to ensure the correct behavior and adequate software quality. The demonstrator was presented to the involved stakeholders and potential end users who gave a positive response to the solution. Following the presentation an on-site test was arranged during which the demonstrator was used in the production environment. After approval by the end users a sub-contractor was given the task to implement a production version of the application which is in operation since then. In figure 5.1 three screen shots can be seen showing the final application for the production. The application is available on the app store and can be used with the data providing system on site. The implemented feature set is nearly equivalent to the operations suggested in the proof of concept implementation presented in this thesis . Key features are drill up and down operations as well as the tracking of topics and measures.

5.1 Future work

Within the scope of this thesis it was not possible to implement all envisioned functionalities. The following features were considered but not feasible within the project time constraints:



(a) KPI overview (b) Topics (c) Measures

Figure 5.1: Implementation engineered by an AHM sub-contractor used in production environment featuring full access to operational data.

- The direct connection to the source systems providing live data of the production.
- Bidirectional data flow: Hosting a server storing additional data for topics and measures including web access for administration of them. Further improvements would include the ability to edit and update the status of measures and their impact.
- Enabling the users on the client to receive predictions on future production output. Further improvements could include live alerts when production output drops significantly.

Although the proof of concept did fulfill the requirements defined in the beginning, several additional analyses and improvements regarding the development process, cost effectiveness as well the degree of integration into the processes in place could be performed:

- Conducting a usability evaluation of the most frequently used features.
- Evaluate the overall project sustainability by comparing the actual costs and benefits.

- Preparation of a change management concept for the transition of the existing processes to novel ones assisted by mobile devices.
- Perform a security evaluation of the client and all servers utilized.

Bibliography

Apple Inc. (2013). *iPhone 5 - Technische Daten*. URL: https://support.apple.com/kb/SP655?locale=de_AT (visited on 05/17/2015) (cit. on p. 60).

Apple Inc. (2014). *iOS Technology Overview - Cocoa Touch Layer*. URL: https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html (cit. on p. 60).

Apple Inc. (2015a). *iOS Human Interface Guidelines*. URL: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/ (visited on 11/15/2015) (cit. on p. 55).

Apple Inc. (2015b). *Safari Web Content Guide: Configuring Web Applications*. URL: https://developer.apple.com/library/ios/documentation/AppleApp%20lications/Reference/SafariWebContent/ConfiguringWebApplications/ConfiguringWebApplications.html (visited on 01/15/2016) (cit. on p. 62).

Apple Inc. (2015c). *Testing with Xcode - About Testing with Xcode*. URL: https://developer.apple.com/library/ios/documentation/DeveloperTools/Conceptual/testing_with_xcode/chapters/01-introduction.html (visited on 10/12/2015) (cit. on p. 61).

Barkin, Stephen R. and Gary W. Dickson (1977). "An investigation of information system utilization." In: *Information & Management* 1.1, pp. 35–45. ISSN: 0378-7206. DOI: http://dx.doi.org/10.1016/0378-7206(77)90007-6. URL: http://www.sciencedirect.com/science/article/pii/0378720677900076 (cit. on p. 2).

Bower, Joseph L and Clark G Gilbert (2007). "How Managers' everyday decisions create or destroy your company's strategy." In: *Harvard Business Review* 85.2, pp. 72–9 (cit. on p. 9).

Brobst, S and J Rarey (2003). "Five Stages of Data Warehouse Decision Support Evolution." In: *DSSResources. COM* 1.06, p. 2003 (cit. on pp. 12, 13).

Codd, Edgar F, Sharon B Codd, and Clynch T Salley (1993). "Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate." In: *Codd and Date* 32 (cit. on p. 20).

Facebook Inc. (2016). *Why React?* URL: https://facebook.github.io/react/docs/why-react.html (visited on 02/02/2016) (cit. on p. 62).

Farkisch, Kiumars (2011). *Data-Warehouse-Systeme kompakt: Aufbau, Architektur, Grundfunktionen*. Springer-Verlag (cit. on pp. 18, 19, 21–23).

Fayol, Henri, John Adair Coubrough, et al. (1930). "Industrial and general administration." In: (cit. on p. 1).

Fielding, Roy Thomas (2000). "Architectural styles and the design of network-based software architectures." PhD thesis. University of California, Irvine (cit. on p. 66).

Fleisch, Elgar and Markus Dierkes (2003). "Total vernetzt: Szenarien einer informatisierten Welt." In: ed. by Friedemann Mattern. Berlin, Heidelberg: Springer Berlin Heidelberg. Chap. Betriebswirtschaftliche Anwendungen des Ubiquitous Computing — Beispiele, Auswirkungen und Visionen, pp. 143–157. ISBN: 978-3-642-55550-3. DOI: 10.1007/978-3-642-55550-3_6. URL: http://dx.doi.org/10.1007/978-3-642-55550-3_6 (cit. on p. 46).

Harsch, Stephen B. (2010). *Management Information Systems* (cit. on p. 11).

"IEEE Standard Glossary of Software Engineering Terminology" (1990). In: *IEEE Std 610.12-1990*, pp. 1–84. DOI: 10.1109/IEEESTD.1990.101064 (cit. on p. 34).

Inmon, William H (1996). "The data warehouse and data mining." In: *Communications of the ACM* 39.11, pp. 49–51 (cit. on p. 17).

BI-Insider (2011). *BI Market: Data Mining & Predictive Analytics (Vendors and Products) – 2011*. URL: http://bi-insider.com/posts/bi-market-data-mining-predictive-analytics-vendors-and-products-2011/ (visited on 09/30/2015) (cit. on p. 27).

ISO/IEC (2001). *Software Engineering - Product Quality, ISO/IEC 9126-1*. Tech. rep. International Organization for Standardization (cit. on p. 34).

ISO/IEC (2011). *ISO/IEC 9075: Information technology – Database languages – SQL – Part 2: Foundation (SQL/Foundation)*. Tech. rep. International Organization for Standardization (cit. on p. 68).

Koontz, Harold (1961). "The management theory jungle." In: *Academy of Management Journal* 4.3, pp. 174–188 (cit. on p. 1).

Lapiedra, Rafael, Joaquin Alegre, and Ricardo Chiva (2011). "The importance of management innovation and consultant services on ERP implementation success." In: *The Service Industries Journal* 31.12, pp. 1907–1919 (cit. on pp. 7, 8, 22).

Laudon, K and J Laudon (2007). *Management Information Systems: Managing the Digital Firm* (cit. on p. 23).

Laudon, K and J Laudon (2008). *Essentials of Management Information Systems*. 8th. Upper Saddle River, NJ, USA: Prentice Hall Press. ISBN: 9780136025795 (cit. on pp. 10, 15–17, 23, 24).

Laudon, K and J Laudon (2012). *Management Information System (MIS)* (cit. on p. 25).

Lichtenegger, Gerald (2015). "Engineering Communication Systems." In: *International Journal of Computer and Communication System Engineering* Vol. 2.3, pp. 518–522 (cit. on p. 42).

Lichtenegger, Gerald et al. (2016). "Industrial Engineering und Management: Beitraegge des Techno-Oekonomie-Forums der TU Austria." In: ed. by Hubert Biedermann. Wiesbaden: Springer Fachmedien Wiesbaden. Chap. Informationssystemdesign fuer Produktionssysteme der Zukunft, pp. 23–42. ISBN: 978-3-658-12097-9. DOI: 10.1007/978-3-658-12097-9_2. URL: http://dx.doi.org/10.1007/978-3-658-12097-9_2 (cit. on p. 41).

Lucey, Terence (2005). *Management information systems*. Cengage Learning EMEA (cit. on pp. 7, 8, 26).

Mohanty, Soumendra, Madhu Jagadeesh, and Harsha Srivatsa (2013). *Big Data imperatives: enterprise Big Data warehouse, BI implementations and analytics*. Apress (cit. on p. 2).

O'Brien, James A (1999). *Introduction to Information systems*. McGraw-Hill, Inc. (cit. on pp. 9, 10, 14, 22).

O'Brien, James A. (2011). *Management Information Systems*. 10th. Boston: Irwin McGraw-Hill (cit. on p. 11).

O'Brien, James A et al. (2006). "Management information systems." In: (cit. on p. 14).

Parmenter, David (2001). "Winning KPIs." In: *MANAGEMENT-AUCKLAND-* 48.11, pp. 100–101 (cit. on p. 28).

Parmenter, David (2010). *Key Performance Indicators (KPI): Developing, Implementing, and Using Winning KPIs*. Second Edition. John Wiley and Sons Inc. ISBN: 978-0470593196 (cit. on p. 28).

Pendse, Nigel and Richard Creeth (1995). "The OLAP report." In: *Business Intelligence* (cit. on p. 20).

Pohl, Klaus and Chris Rupp (2009). *Basiswissen Requirements Engineering (iSQI-Reihe): Aus-und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. 3rd ed. dpunkt. verlag (cit. on pp. 33, 35, 36, 43, 44).

Pohl, Klaus and Chris Rupp (2011). *Basiswissen Requirements Engineering, dpunkt. verlag, 3* (cit. on pp. 34, 38).

Rupp, Chris and SOPHISTen (2009). *Requirements- Engineering und -Management*. HANSER (cit. on pp. 31, 36, 37).

Sachs, Maximilian (2014). "Requirements engineering and design of human-centered information services utilizing near-eye display devices." MA thesis. TU Graz (cit. on p. 56).

Stecklein, Jonette M et al. (2004). "Error cost escalation through the project life cycle." In: (cit. on p. 3).

Stratton, R. and D. Mann (2003). "Systematic innovation and the underlying principles behind TRIZ and TOC." In: *Journal of Materials Processing Technology* 139 (cit. on p. 24).

The Apache Software Foundation (2015). *Platform Support*. URL: `https://cordova.apache.org/docs/en/latest/guide/support/index.html` (visited on 08/25/2015) (cit. on p. 62).

Thiele, Maik (2010). "Qualitätsgetriebene Datenproduktionssteuerung in Echtzeit-Data-Warehouse-Systemen." In: (cit. on p. 15).

Wernher Behrendt Georg Guentner, Siegfried Reich (2013). "Industrie 4.0 Menschen, Maschinen und Organisationen im World Wide Web." In: (cit. on p. 1).

Wolfsgruber, Christoph and Gerald Lichtenegger (2015). "Management Information 4.0." In: *WING business* (cit. on p. 41).