UNIVERSITY OF TWENTE.

Faculty of Behavioral, Management and Social Sciences (BMS)

Process Standardization in Requirements Engineering at Movares

Arjan Laout
M.Sc. Thesis

August 2017

Faculty of Behavioral, Management and Social Sciences (BMS)
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

GENERAL INFORMATION

Author:	ir. Arjan Laout
Telephone number:	+31 641142434
Email address:	a.laout@student.utwente.nl / Arjan.Laout@gmail.com
Study Program:	Industrial Engineering & Management
Specialization:	Production & Logistic Management
University:	University of Twente
Faculty:	Behavioral, Management & Social Sciences (BMS)
	Drienerlolaan 5
	7522NB Enschede
University Supervisors:	Dr. Peter C. Schuur
	Dep. Industrial Engineering and Business Information Systems
	Dr. Maria-Eugenia Iacob
	Dep. Industrial Engineering and Business Information Systems
Company:	Movares
	Daalseplein 100
	3511SX Utrecht
Company Supervisors:	ing. Liza Thole
	Consultant System Integration
	ir. Michiel Blaauboer
	Line manager System Integration

Version:	Final version Public

PREFACE

This document is the final version of my master thesis. This master thesis concludes years of study at the University of Twente in Enschede and the master program Industrial Engineering & Management, with a specialization in Production & Logistics Management.

I had a great time in Enschede, where I had all the support I needed from the university and fellow students. During the final half year of my studies I have been mentored by Peter Schuur and Maria-Eugenia Iacob. I would like to thank them for their constructive feedback that helped bringing this master thesis to a higher academical level.

I would also like to thank Movares for the warm and enthusiastic welcome in Utrecht. I enjoyed excellent guidance throughout the months I worked at Movares. In particular Liza Thole and Michiel Blaauboer provided a unique combination of freedom and support.

Finally, I am thankful for my family, friends and especially Melanie De Caluwé, who was closely involved throughout the whole thesis development.

I hope you will enjoy reading this master thesis.

ir. Arjan Laout

University of Twente, Enschede

14th of August, 2017

The most dangerous phrase in the language is, 'We've always done it this way.'
~ Rear Admiral Grace Murray Hopper

MANAGEMENT SUMMARY

MOTIVE

The department of System Integration (SI) at the engineering and consultancy organization Movares in Utrecht is particularly concerned with finding and managing interfaces between different disciplines within infrastructure projects and directing processes that lead up to delivering specific documents.

From interviews with employees at the SI department, several needs rose to the surface. One such need is that SI employees are often asked to make complete SRS documents, including the elicitation of the requirements themselves. SRSs are system requirements specifications, which are documents specifying the technical requirements imposed on the designs of a particular infrastructure project. Making these documents is part of the requirements engineering (RE) process. Eliciting the technical requirements in this process requires specialized technical knowledge, which SI employees often do not possess. Thus, the technical tasks need be delegated more to the technical disciplines, who often do not know how to correctly elicit requirements. Another issue is that work processes are often not documented and employees then usually work from own experiences, which leads to employees having to reinvent the wheel and are thus working inefficiently. Additionally, and relatable to the previously mentioned issue, employees experience the quality of some of the documents as below standard and inconsistent. The core problem that these issues have in common, is that the processes are not documented and standardized. This leads up to the main research question, which is:

How can a standardized system requirements engineering process be realized at Movares, in order to move towards a more economical way of working and allow SI employees to employ a more organizational role in projects?

METHODOLOGY OF THE RESEARCH

The report and problem approach is structured by applying two scientific articles. The research framework, which includes the appropriate steps in order to answer the research question, is formulated based on the theories of Verschuren & Doorewaard (2010). The research outline that further structures the report and includes the general phases of the research, is composed using the methods described by Heerkens & van Winden (2012).

The process that is analyzed, is the process of requirements engineering, more specifically the process of making system requirements specifications. To answer the main research question, first, the current way of producing SRSs is described and documented. A process map is created, which will already help SI employees gain insight into their current way of working. To further analyze the current process of making SRSs, performance indicators are developed, which are then operationalized using historical data. Having documented and analyzed the process, we have searched to improve it. Using literature on requirements engineering, we have tried finding the best practices that are applicable to the Movares way of working. Combining input from the current process, literature and interviews with employees, a new way of working concerning SRSs is constructed and documented in an SRS manual.

In order to obtain the maximum potential of this new way of working, data is needed to monitor the process and facilitate data driven performance measurement. This will provide more insight into the process and support continuous improvement. For this reason, a significant part of this thesis is spent on researching the

possibilities of data collection for the purpose of implementing a KPI dashboard. KPIs or key performance indicators are the most important measures that inform managers on how well an operation is achieving its organizational goals. A KPI dashboard helps in visually presenting the process with live data on KPIs.

RESULTS

The results from this thesis can be roughly split into two subjects; 1) a new way of working is established and documented and 2) KPIs are developed and a KPI dashboard is under construction. Although these subjects are intertwined, they are discussed separately.

SRS NEW WAY OF WORKING

The new way of working is described in the new SRS manual, in which the specific activities are documented and standardized. The process has significantly changed by adding RE best practices and standardizing the way these activities are carried out. The main activities of making an SRS are presented in figure 0.1. As can be observed, the RE process is an iterative process in which system requirements are derived from input such as designs and customer requirements, which are then structured, verified and sent to the designers, who use the requirements as input for new designs. This way, designs and system requirements are further specified until the desired level of specification is reached. In the manual, each main activity is divided into sub activities, which are then described in detail. The main activities remain similar in the old and new way of working. However, the sub activities are significantly changed, primarily within main activity 1 and 5. For example, in the new way of working, more effort is spent before starting to derive requirements by making a change management strategy and conflict strategy in main activity 1. Within main activities 4 and 5, newly derived system requirements are now explicitly checked on quality and usability before they are finalized. Additionally, configuration management is applied by gathering new requirements and synchronizing the different interfaces and actors that will use the requirements in the next iteration.

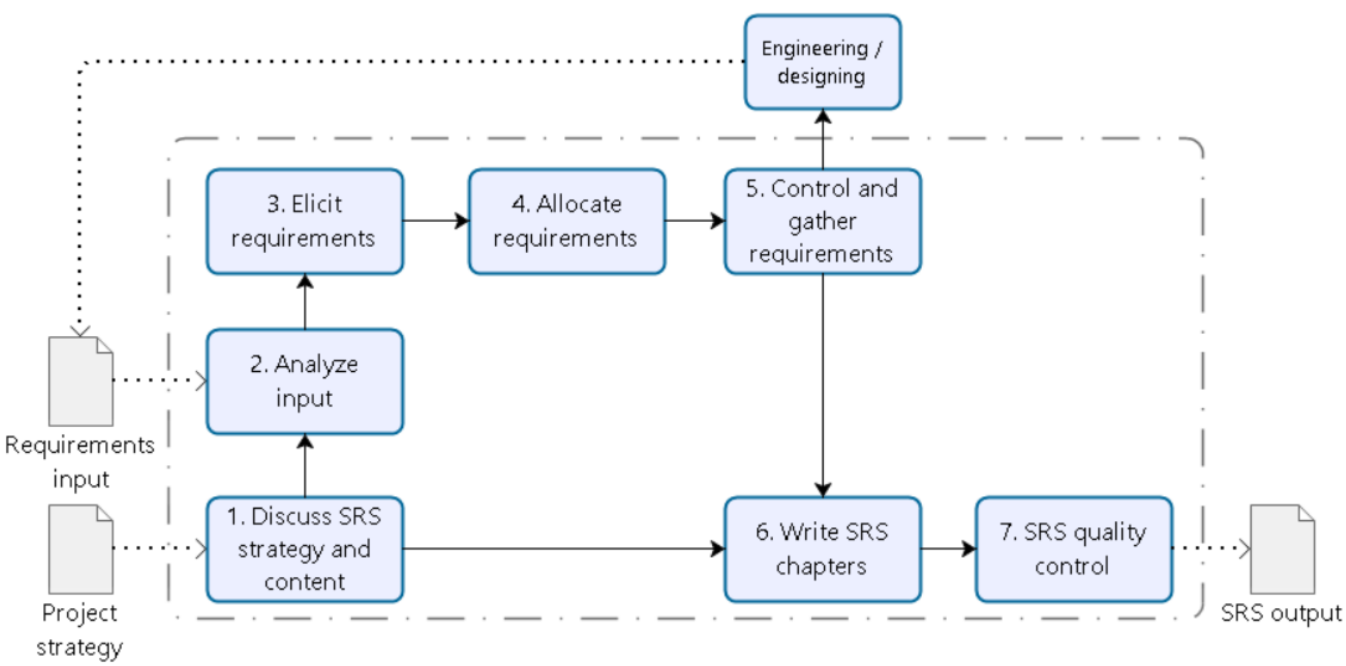


Figure 0.1 - main SRS activities

In the old way of working, sub activities were not clearly defined, only carried out sometimes and executed in different ways. In the new way of working, every sub activity is clearly described, how it should be executed and by whom. The new way of working provides the possibility to assign tasks more clearly and thus to delegate technical tasks easier. The role of a systems engineer becomes more facilitating. Additionally, both the duration of the process and the quality of output will improve. This new process is now defined, standardized and implemented. However, the process can still be further improved by managing and controlling the process based on live measurements of the process in projects.

Although SRSs are made throughout the organization, the SI department possesses the most knowledge on RE and SRS documents. This means that the SRS manual is put under control of this department. This includes continuously reviewing and improving the process every few months. Also, the manual is spread as broadly as possible among employees who are regularly involved in the process.

To help monitor the process, show trend lines and facilitate continuous improvement, a KPI dashboard is needed, which is discussed in the following subsection.

KPI DASHBOARD

The other critical subject of this thesis is developing a KPI dashboard based on live data. The necessity for change compared to the old way of working becomes clearer when KPIs are developed and operationalized using historical data from 21 SRS over the past ten years, of which the results are presented in table 0.1. Since there is no data available on KPIs 7 and 10, these are not displayed in the table. Studying these results, it turns out that making SRSs is actually not a core process of the SI department, since only a few SRSs can be found that are produced by employees from the SI team. This proves the fact that data management helps to gain insight into the processes of the department. Additionally, the results show that the process is not performing well on certain aspects such as the average cycle time, on-time delivery and the average number of days late. Next to this, there is a high variation in performance- and quality indicators. Standardizing the process should reduce this variation.

Table 0.1 – Operationalized KPIs on the process of producing SRS documents

CONFIDENTIAL

In order to apply these KPIs effectively, they need to be presented in a visualized KPI dashboard, automatically fed by live data. Managers need specific information in a single glance and easily accessible to them. For this reason, we started developing a KPI dashboard. Implementing this concept is harder than applying the SRS new way of working, but it also has more potential for improving Movares' performance in general. Implementing this change does require a cultural change. Because of this, it was chosen to start small by developing a proof of concept of the dashboard and only collect data on the requirements engineering process. In the future, data collection can be incrementally increased across more products and processes.

A proof of concept of the dashboard has been constructed. Using this proof of concept, several line- and project managers at Movares are convinced of the potential added value to the organization. Therefore, concrete agreements are already made to further develop the dashboard after my internship has ended.

Concluding, the most prominent contributions of this research are documenting and modeling the old way of working concerning SRSs, implementing a new and improved way of working and initializing the development of a KPI dashboard to better manage the process.

RECOMMENDATIONS

Throughout the thesis, multiple issues and aspects are exposed, which could still be improved or that need further research. These issues and aspects are formulated as recommendations for Movares. The recommendations are mentioned in order of importance for Movares in general:

- Professionalize the proof of concept towards a visual dashboard. The next steps in making a KPI dashboard would be to visualize the KPIs in the dashboard better and to create links from existing Movares systems, so the data is completely automatically generated and provided to the user.
- Standardize booking elements. The booking elements, which represent the activities on which employees record the hours they have worked, are different and reinvented for every project. This leads to inefficiencies and confusion among employees concerning the specific elements on which they need to record the hours they have worked. Standardized booking elements are necessary for reliable and uniform data.
- Review the SRS process and manual every six months. Now that the process is standardized, it should be continuously improved. To support improvement, data driven management could be applied in the near future.
- Investigate business cases. Once data on certain processes is available and operationalized, business cases could be made in order to support investments in standardization options. Concerning the requirements engineering process, multiple standardization options are already listed.
- Standardize more products and processes, apart from SRS documents. The contract management department has already shown interest in applying this concept to their products and processes.
- Develop the prediction model. With the development of the KPI dashboard, the development of an incorporated prediction model has been pushed towards the background, since it requires extra handlings from employees. This means it is harder to implement. However, it is certainly advised to further develop this model after a KPI dashboard is completed and data is being gathered.

LIST OF ABBREVIATIONS AND DEFINITIONS

Abbreviation or term	Stands for	Description	Explained in section
AD	Architectural Design	A schematic design in which the different functionalities and objects including their relation is described using symbols.	2.2.1
CPM	Critical Path Method	A tool for scheduling processes in projects.	3.4
Contract specification	-	A contract document, which includes the complete description of work to be carried out to complete the project.	2.4
CRS	Customer Requirements Specification	A document specifying the requirements set by different actors, including the client.	1.1
Customer requirements	-	A demand made by the client or other stakeholder regarding the engineering or realization of the future system.	1.1
Designing	-	The part of the engineering process in which a subsystem or object is designed or modeled.	2.2
Disciplines	-	Departments within Movares specialized on a certain aspect in engineering. Disciplines can either be technical or conditional.	2.1
Engineering	-	All processes that together lead up to the final designs, contracts and documents, from which the system can be realized. Also, the left side of the V-model.	2.2
FBS	Functional Breakdown Structure	A structured hierarchical overview of functions on which the future system should suffice.	2.2.3
Fish model	-	An iterative development model in which alternative options are generated, and then filtered. An option is chosen and further specified.	2.1
GR&R	Gauge Repeatability and Reproducibility	A Gauge Repeatability and Reproducibility study (GR&R) assesses the precision error in a data study	4.7.3
GWW-sector	Ground-, Water-, and Road construction sector	The sector concerned with the construction of infrastructure projects, in which Movares is active.	1.1
Hours	-	The application in which employees manage the number of hours they worked on specific activities.	3.3.1
Interface	-	Within a system, different sub systems or objects are defined, among which interfaces exist. External interfaces are interfaces between the system of scope and the environment.	2.2.2
ISO	International Organization for Standardization	ISO norms are international standards which are set by the International Organization for Standardization.	1.1

KPI	Key Performance Indicator	KPIs are the most important measures that inform managers how well an operation is achieving its organizational goals (Boddy & Paton, 2011).	3.3
N^2 matrix	N-squared matrix	A schematic representation of interfaces.	2.2.1
PERT	Program Evaluation and Review Technique	A tool for scheduling processes in projects.	3.4
Process maturity	-	A model that describes the five phases that processes go through as they move from immaturity to maturity (Harmon, 2007).	3.2.1
Process requirements	-	Specify the procedures for completing the tasks specified in the contract specification.	2.4
ProjectWise	-	The application in which the different documents of a project are stored and managed.	1.4
RE	Requirements Engineering or Requirements Engineer	The process of deriving and managing requirements. A requirements engineer is the person deriving and managing requirements.	2.1
Relatics	-	The application in which different aspects of a project such as requirements and interfaces are managed.	2.2
Rework	-	Work performed on a product after it is finished but is not deemed sufficient.	1.2
SBS	System Breakdown Structure	A structured hierarchical overview of the objects of which the future system will exist.	2.2.3
SE	Systems Engineering or Systems Engineer	An interdisciplinary approach and means to enable the realization of successful systems (INCOSE, 2006).	1.1
SI	System Integration	The department of scope at Movares.	1.1
SMART	Specific, Measurable, Acceptable, Realistic and Time-related	In literature, multiple definitions are provided. This thesis uses the definition from Leidraad (2013).	2.2.2
SRS	System Requirements Specification	A document specifying the technical requirements on the future system.	2.2
System	-	The combined structure of interacting objects within the project's scope (INCOSE, 2006).	1.1
System Requirements	-	The technical and mostly functional requirements which limit the future designs of the systems. This includes customer requirements, but is broader than just customer requirements, and further specified.	2.2
V-model	-	A model presenting the iterative steps in construction projects; top-down specifying and bottom-up realization of the system (Leidraad, 2013).	1.1
V&V	Verification & Validation	The procedures that specify how requirements should be verified.	2.2.2

TABLE OF CONTENTS

1. CHAPTER 1 - INTRODUCTION

This master thesis concludes the master program Industrial Engineering & Management at the University of Twente, with a specialization in Production & Logistics Management. The graduation assignment is executed over a period of six months at Movares in Utrecht, more specifically at the System Integration (SI) department. Movares is an engineering and consultancy organization, which is mainly active in the infrastructure and civil engineering industry. In this first chapter, the thesis is introduced by describing the issues at hand and finding an approach to deal with these problems. Before diving into possible issues, preliminary background information on the industry and company is provided in section 1.1. After that, the problem statement is defined in part 1.2, and in 1.3 the scope is delimited. Next, the research approach is constructed in 1.4, part 1.5 specifies the research questions, and finally the outline of the rest of the report is illustrated in 1.6. The process of constructing the introduction of this project is in accordance with the project design method described by Verschuren & Doorewaard (2010), which is further discussed in the research approach, in part 1.4.

1.1. INDUSTRY AND COMPANY DESCRIPTION

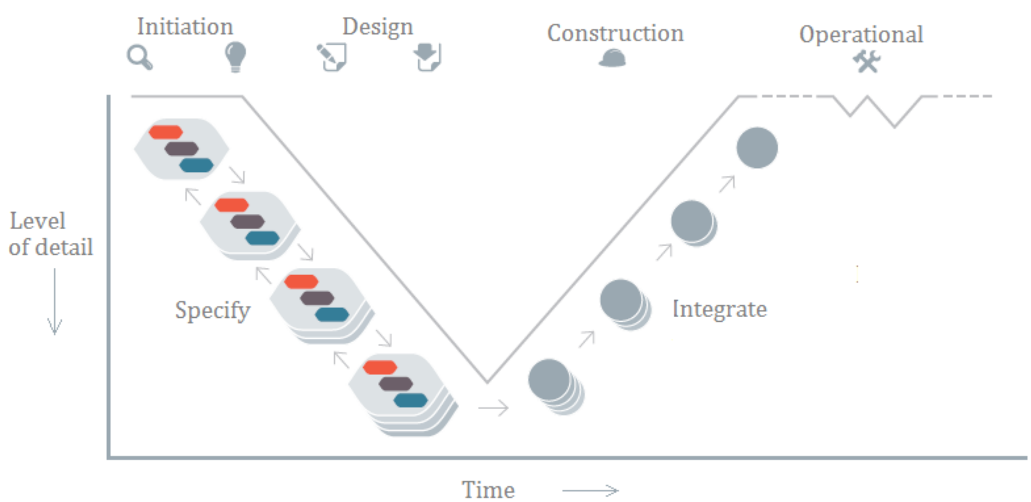


Figure 1.1 - V-model (Leidraad, 2013)

Movares is active in the GWW-sector (Ground-, Water- and Road construction). The way this sector works can be best described by using the V-model, which is presented in figure 1.1. This V-model is the result of the combined efforts of industry specialists and used throughout the industry. The details of this model are further explained during this section. A client such as ProRail or Rijkswaterstaat comes up with a problem or idea, for example a road having to cross a waterway. This problem ought to be solved by some sort of construction. This is an extensive project that undergoes different phases, during which a project team is assembled to develop construction plans. In the beginning, one of the first steps is to perform a stakeholder analysis. Each stakeholder has several demands or requirements for the future construction. These stakeholder requirements are further specified in a CRS (Customer Requirements Specification). In the example, the road must facilitate a certain number of cars per day and cannot be interrupted for longer than five minutes. Given these specifications, different alternatives are generated. For example, a tunnel could be built or a moveable bridge. Finally, an alternative solution is chosen, in this case the option of a movable bridge. Next, different requirements specific to the chosen option have to be elicited, which specify the demands that have to be satisfied. From these new requirements, more specific options can be generated. For example, if there should also be a bike lane, and in that case on which side of the bridge.

All such projects in the infrastructure industry apply systems engineering, according to ISO standards. These standards specify that projects are thought of as systems that can be separated in subsystems, which in turn can be separated in subsystems, all the way down to specific system elements or objects. This is demonstrated in figure 1.1, as the specification process is divided into multiple layers over time. The system configuration, consisting of multiple subsystems, is called the System Breakdown Structure (SBS). The way these subsystems are built up, is shown in figure 1.2.

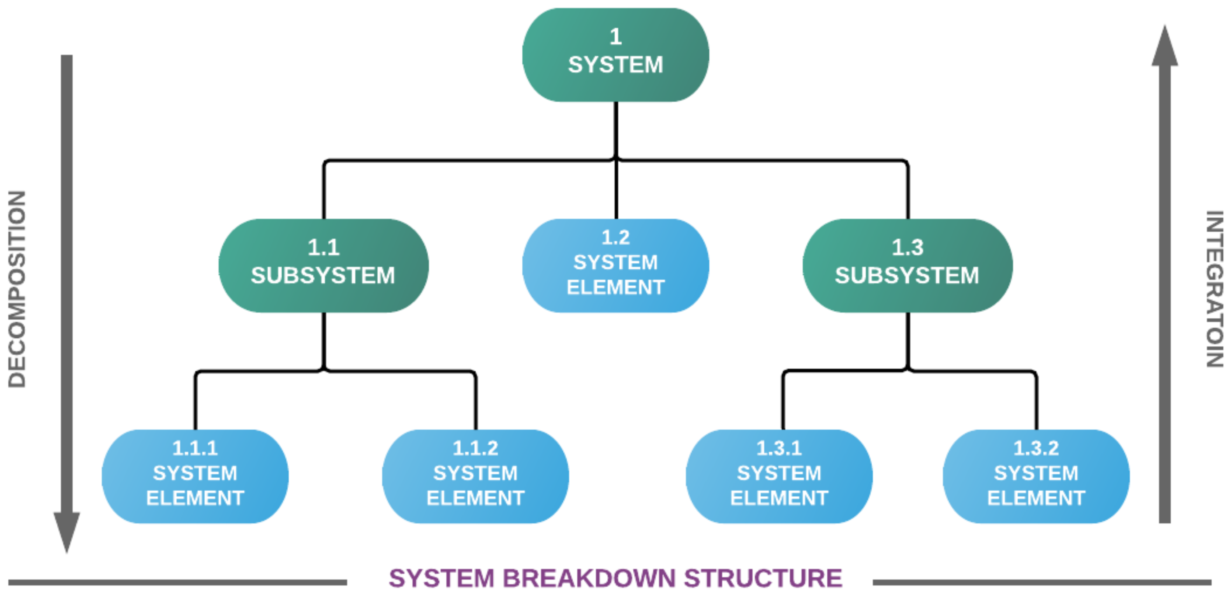


Figure 1.2 - System Breakdown Structure (SBS) (Leidraad, 2009)

After an alternative option has been selected, the project team will go down a level in the SBS. Using the same example, a movable bridge can be viewed as a number of different parts working together. Each of these separate parts have their own system requirements specification (SRS). These requirements are systematically decomposed into smaller subsystems or elements (top-down), to the point that every small part of the project that is required, is specified. This process is shown in figure 1.3, which is part the V-model. This is visually shown in the coloring, which is presented in both figure 1.1 and 1.3. The arrows returning in the opposite direction represent the verification and validation process, which is described in a V&V plan (Verification & Validation). This process checks whether a solution objectively and explicitly meets the requirements and is suitable for its intended use (Leidraad, 2013). Next, in the realization or integration phase of the V-model, all parts are realized and integrated bottom-up to the point that the whole project is finished. Later, after testing, the project enters the operational phase.

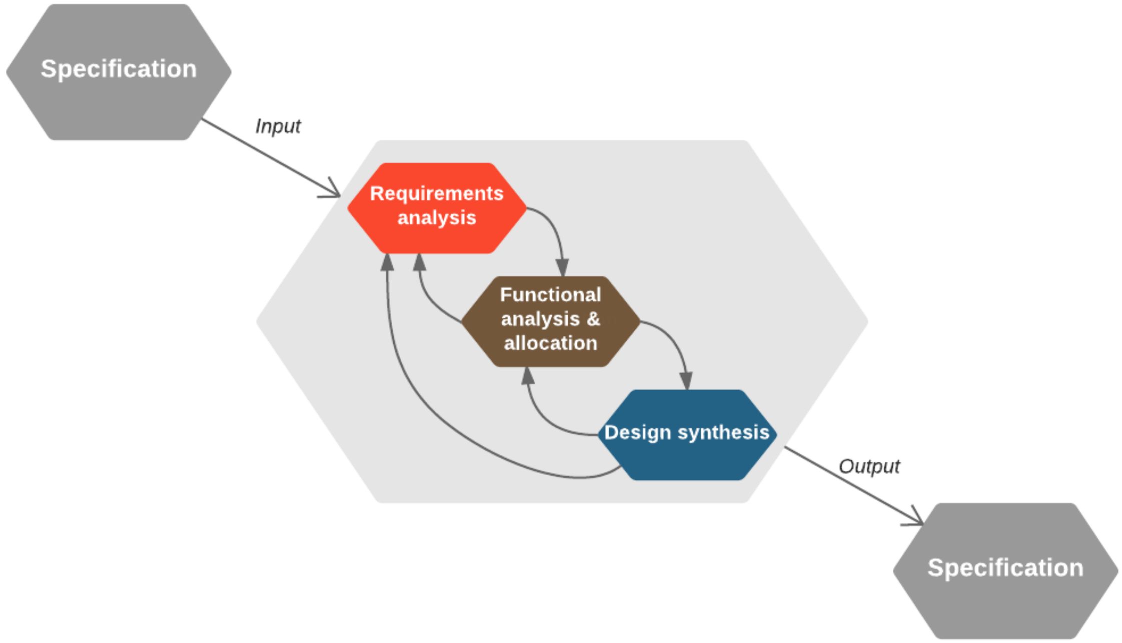


Figure 1.3 – A section of the V-model (Leidraad, 2013)

Distinctive steps of a project could be subcontracted to various parties, such as Movares. Winning a (sub)contract depends on which party delivers the best offer, considering multiple factors such as cost, quality and environment. It could be that Movares has to design the security systems of the bridge from the example, while other organizations are engineering other parts of the bridge.

For each project, the consecutive steps that have to be executed are often quite the same and are partly determined by ISO norms. During the lifetime of a project, a lot of documents have to be made and sent to the various stakeholders. These documents can be seen as products that Movares delivers to the client. The steps taken for each project are usually not unique, but the products themselves such as designs and other documents are almost always unique for a particular project or project phase. An example is an SRS, or system requirements specification. The content of an SRS is almost always different for each project. Even two bridges from the same type, such as movable bridges, can have contrasting requirements and thus separate SRSs. They can be built on diverse locations, with diverse stakeholders and thus various stakeholder demands. Furthermore, these locations have distinctive environmental factors, and thus different requirements. Other documents or products are for example a CRS, contract specification, V&V-plan and designs themselves.

For each project that is assigned to Movares, a project team is put together. A project team exists of people with diverse backgrounds from multiple departments. The role that an SI employee fulfils within a project team should be mainly organizational, to build a bridge between the different specializations (and systems) within a project. As explained before, a project's subject is seen as a system, comprising multiple subsystems. It is usually the case that various departments involved in a project only design the subsystem that is their specialty. In order to adjust these subsystems to each other, SI makes sure that each step fits into the rest of the project and the bigger picture. Regarding the example of the bridge, when making the specifications of the engine powering the moving parts of the bridge, one should take into account what parts of the bridge are moving and in which direction. The department System Integration is active during the developing and processing of system requirements specifications, but also for example during testing and system verification and validation.

The SI department is a young team, as it was formed in January 2016, and is still somewhat struggling with defining their actual internal processes and with profiling itself within the organization. The department does have a vision in what direction they want to establish and present itself. The role of an SI employee within a project is already explained, but there is a broader definition. This is the definition from one of the senior employees: “System Integration is to connect stakeholders, to direct the integral design to acquire consent of the authorized supervision, the integration and testing of (sub)systems and to prepare and regulate the commissioning of a project, succeeding the first time right.”

1.2. PROBLEM STATEMENT

The next phase of this master thesis consists of the problem statement. In this part, the goal is to find the core problem through a problem cluster, possible causes of the issue and the objective of this research.

Before finding the core problem, the handling problems and underlying issues must be identified and put in relation to one another through a problem cluster. The latter is a way of finding the core problem; it presents the causal relations between the perceived and underlying problems (Heerkens & van Winden, 2012). This way, a core problem and possible causes of this obstacle can be found. A handling problem is the perceived discrepancy between norm and reality; this is where the research starts. After a number of interviews with employees, the issues and desires within the team are found, which can be structured in a problem cluster. The problem cluster is visualized in figure 1.4, after which the different problems are explained. By numbering the issues in the cluster, it is easy to refer to the specific problem in the text.

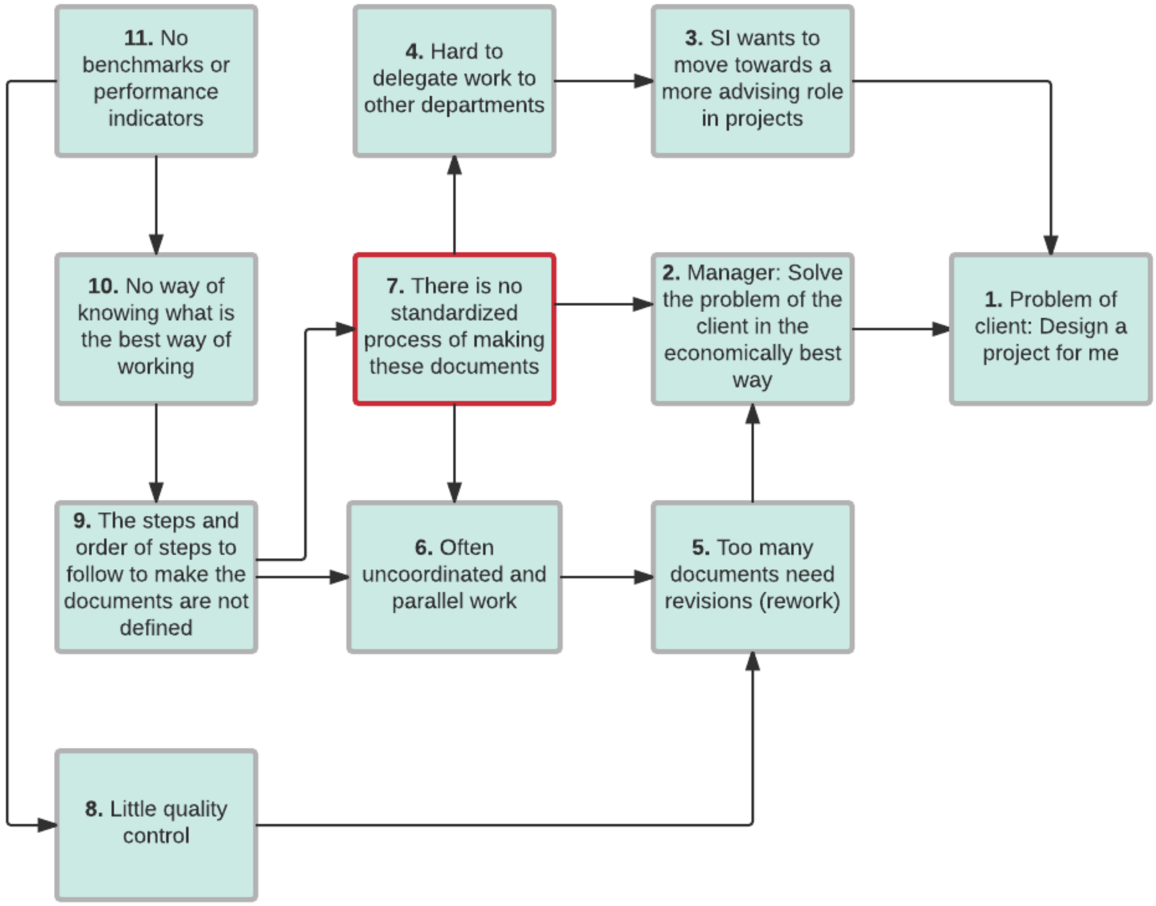


Figure 1.4 – Problem cluster

The initial assignment Movares had in mind, was to analyze the processes within SI, based on key figures and data. However, this is not the perceived problem but a means to achieve something else. After initial research, two handling problems emerged. The first handling problem that surfaced, is that SI employees wish for a more advising role in comparison to the current situation (**3**). This issue is related to the fact that SI is still a young department, trying to develop itself. One direction the department wants to develop, is towards a more advising and facilitating role within projects. At the moment, some employees indicate that they are doing too much of the 'technical' work. They have expertise in making some of the necessary documents, but would like to delegate some of that work to other departments, in order so SI can adopt a more advisory role in projects (**4**). One way of delegating work, is through standardizing processes, which makes the process and tasks more clear and easy for everyone involved. However, at the moment there are no standardized processes that define the Movares way of working in making documents (**7**). Projects or specific products are difficult to generalize, since each one is unique. Yet, the way in which a product is made can be generalized, referring to the consecutive steps that are needed in order to make a product. For example, there are a lot of projects for which a requirements document should be produced. It would be better to produce these documents in a consequent matter. However, this is not the case.

The second handling problem is perceived by the manager; to satisfy the client's demands at the economically best way possible, regarding cost and quality (**2**). The fact that projects could be done cheaper, has a lot of different causes. There are always things that could be done better or cheaper within an organization. A more concrete cause is the fact that there are no standardized processes of making the documents for the client (**7**). At the moment, SI employees are producing these documents from their own expertise, which leads to employees having to reinvent the wheel. Making a document requires input from different people and departments that are involved in a project. The fact that people within the project are doing somewhat uncoordinated parallel work (**6**), leads to some documents becoming incoherent. This comes at the expense of the quality of certain documents; documents need to be revised, leading to rework (**5**). Rework then causes projects to become costlier than they were supposed to be (**2**). Regarding an SRS, it could be that a first version SRS is reviewed, after which the product is improved and reviewed again multiple times. The percentage of rework is preferably kept at a minimum. However, this is not the case at Movares. This is partly due to the fact that quality control is only specified at a high, general level and not product specific. In other words, quality control remains somewhat vague on product level (**8**).

Solving the problem of non-standardized processes and parallel work (**6**), requires Movares to first define the steps and order of steps to follow to make the documents (**9**). Also, the delegation of work needs to be clear and coordinated. Employees making SRSs usually work from own passed experiences. However, there is no way of knowing the best way of working in complex situations (**10**) and best practices are not shared. To find the best way of working and also to determine the quality of documents, benchmarks or performance indicators of some sort should be specified and implemented (**11**). These could then be used in analyzing documents and processes.

Resulting from the problem cluster and interviews, a core problem is defined. *The core problem is that there is no standardized process of making the documents produced by the SI department.*

Now that the core problem is found, the objective of this research can be formulated. The objective is related to the core problem and refers to the desired situation.

The objective is to analyze the existing processes of SI, in order to find a way to standardize the process of making plans and documents for the successful continuation of projects at the economically best way possible, regarding cost and quality.

To be able to track the progress of this research and find out whether the objective is met or not, it should be operationalized. This also coincides with problem 11 from figure 1.4. At the moment, there are no benchmarks. A brainstorming session resulted in three metrics to be able to track the objective of this thesis:

- The average amount of hours spent on a particular type of document or product.
    - This aspect is related to cost.
- The average number of revisions of a particular type of document or product.
    - This aspect is related to quality.

These evaluation criteria will be operationalized in section 3.3.

1.3. SCOPE

In this part, the boundaries of the research are established in order for this graduation assignment to not become too narrow or too broad. The purpose of the scope is to select the problems, that when solved by the graduate student, is of most value to the organization and the student, given the amount of time. The amount of time to spend on a thesis determined by the university is 20 weeks, excluding the preliminary research, which consists of chapter 1 of this thesis. There are problems within Movares, such as the transparency and clarity of the way projects are structured in ProjectWise[1], which are left out of scope and the problem cluster. The issues to resolve or review should be in line with the study of the student, (in)direct interest of the team and can be directly influenced. Other issues, such as the one mentioned above, are informed to the manager for consideration. To resolve the core problem of standardizing document-making processes, the other underlying problems are also considered, namely problem 6, 8, 9 and 10 from figure 1.4. If these problems are improving, handling problems such as costs should also improve.

In the problem statement, 'documents' are mentioned a lot, also giving examples of these documents. Given the time, not all documents can be examined in detail. Therefore, it is chosen to analyze the most impactful documents for the SI department. After deliberate consideration, it is decided to analyze SRS documents, or System Requirements Specifications. The SRS is an important document that takes a central place in a project's development in the GWW-sector (translated from Dutch: Ground-, Water- and Road construction sector). Next to this, multiple SI employees have indicated to prefer to delegate the making of this document to other departments, which is currently not possible. The SRS document should not be seen as a document, but more as a process, namely the deriving and managing of system requirements. System requirements are usually gathered and managed in some sort of platform. In case the client asks requests it, these requirements are then inserted in an SRS document and handed over to the client, for example to review the requirements. Next, the SRS is input for the contract specification. If the client does not necessarily need an SRS, the requirements are put directly from the platform into the contract specification. If this process is to be evaluated, it is important to also take contract specifications into account. For this reason, contract specifications are analyzed as well, but to a lesser degree than an SRSs. The rationale behind this decision is that the SI department is not specialized in making contract specifications, this product is usually constructed by another department. CRSs are not taken into account, for these contain customer (stakeholder) requirements, whereas the other two products contain system requirements. Concluding, this research focuses on improving the system requirements engineering process.

1.4. METHODOLOGY AND RESEARCH APPROACH

Now that the problem and scope are explained, the research approach can be constructed. This part proceeds from the problem structure towards a solving structure, answering the question on how to solve the problem. Methodology is thinking about a structured way of working, in problem solving as well as in research. The methodology is concretized in a research approach. This is done before formulating the research questions, since research questions partly result from the research activities. The research approach includes the research activities that are described in the research framework. From the research framework, the methodology and plan of approach can be derived. A research framework is a schematic representation of the research objective and includes the appropriate steps that need to be taken in order to achieve the objective. These steps are defined using the article from Verschuren & Doorewaard (2010), which can be used as a detailed manual on how to execute a research project. After careful consideration, the research framework is defined as shown in figure 1.5. The blue, square boxes

[1] ProjectWise is the online platform in which all documents of projects at Movares are grouped, stored and managed.

represent the activities that have to be carried out. For example, 'Literature research' refers to undertaking a literature research in order to assist in the consecutive activities. The large, blue arrow represents the general order and direction of the research activities to be executed. The underlying color planes represent the more general phases of the research, which coincide with the research outline in part 1.6 and figure 1.6. The research framework is more content-oriented and directed towards the solution process, whereas the research outline is more general and also comprises the problem identification phase and formulation of the problem-solving method phase.

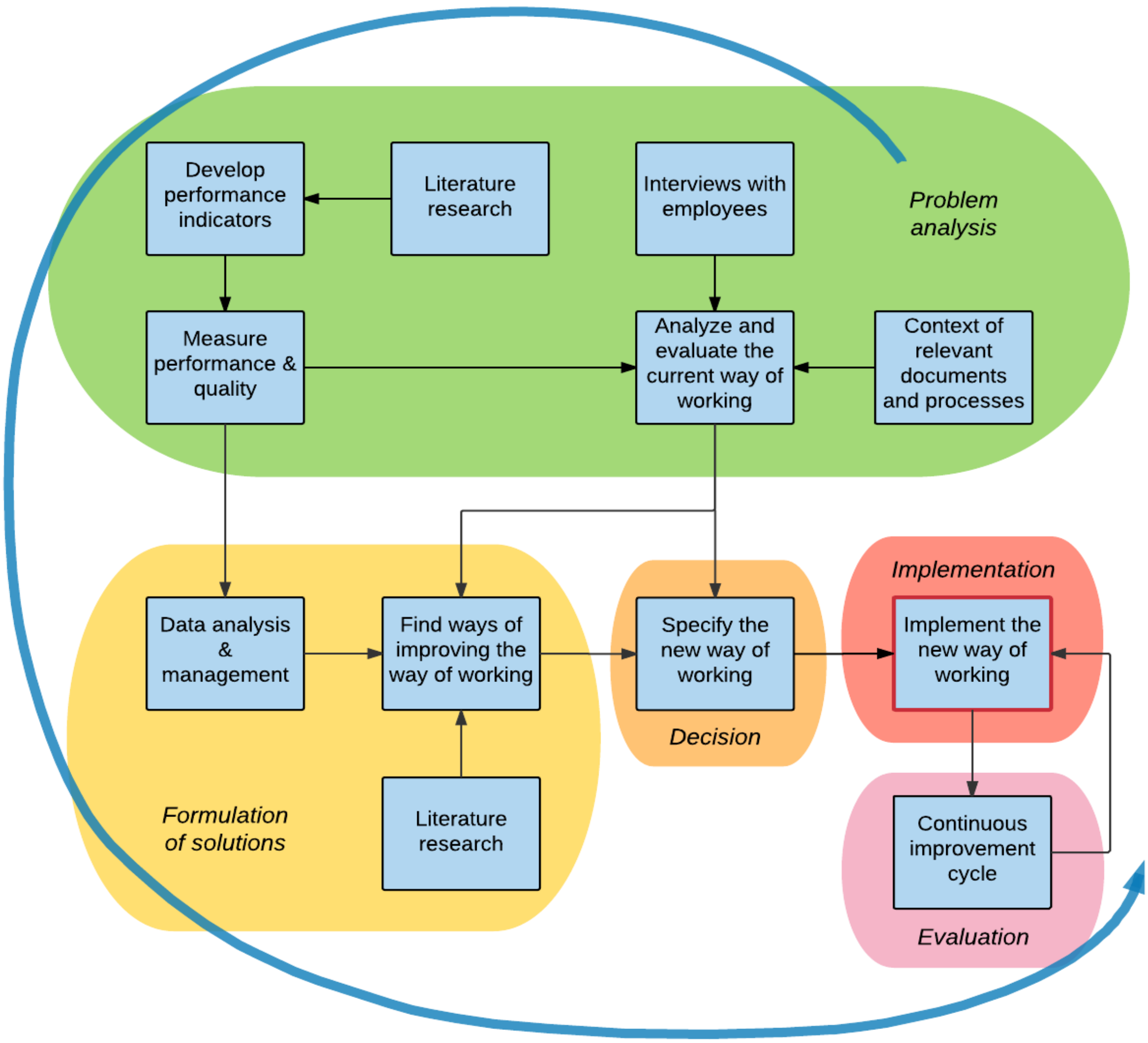


Figure 1.5 - Research Framework

The SI team is the main stakeholder involved in this project. It is therefore essential that the SI team remains involved throughout the course of the graduation assignment, ensuring that their wishes are taken into account. The team holds monthly meetings, where different matters such as problems or progress on different projects are discussed. During these team meetings, there will be time to present the findings found so far. Next to these team meetings, frequent (at least three separate employees per week) discussion sessions are held with individuals, since these sessions allow for a discussion on a deeper level. Moreover, they ensure that the conversational partner will respond more thoroughly and will offer an honest opinion.

1.5. RESEARCH QUESTIONS

To be able to reach the objective, the main research question and sub questions are formulated. These questions are set up in such a way, that when the sub questions are answered, the main research question can be derived from the results. The questions are answered in logical order throughout the thesis. One remark that has not yet explicitly been discussed, is that there was actively tried to incorporate data analytics in order to make statistically ratified statements on the processes. This is reflected in the research questions.

The main research question is related to the problem statement and refers to the core of the objective to be achieved: *How can a standardized system requirements engineering process be realized at Movares in order to move towards a more economical way of working and allow SI employees to employ a more organizational role in projects?*

The sub questions and order of questions are related to the problem statement and problem framework and are specified as follows:

1. What is the current way of managing system requirements at Movares?
1.1. What is the context of SRSs relating to other documents?
1.2. What processes can be defined in making an SRS?
1.3. Who are involved in the process?

2. How is the current way of working at Movares performing?
2.1. What performance indicators can be identified?
2.2. How can quality be measured and ensured?

3. How can the current way of working at Movares be improved?
3.1. What is the best way of working according to literature?
3.2. How can this be applied at Movares?
3.3. How can data analysis help find the best way of working?

4. How can a new and standardized way of working be realized at Movares?
4.1. What will be the steps and order of steps to follow to make an SRS?
4.2. How can the process be managed and controlled using data?

5. How can the new way of working be successfully implemented at Movares?
5.1. How can continuous improvement be ensured?

The way this report is built up, is that each research question is assigned to a chapter. Sub question 1 is assigned to chapter 2, question 2 is treated in chapter 3, etcetera. In chapter 7, the main research question is answered.

1.6. RESEARCH OUTLINE

To structure the report even further, the research outline includes the general phases of the research and the chapters where these phases are discussed. The research outline is already partly treated in parts 1.4 and 1.5, and is now further elaborated and concluded in 1.6. The managerial problem-solving method, as described by Heerkens & van Winden (2012), is used to construct the research outline and to define and connect the chapters and questions of this thesis. This method specifies the general steps that have to be undertaken in solving managerial problems. The research outline is consistent with the research framework, as specified in part 1.4 and follows the direction of the arrow from figure 1.5. The research outline is graphically presented in figure 1.6.

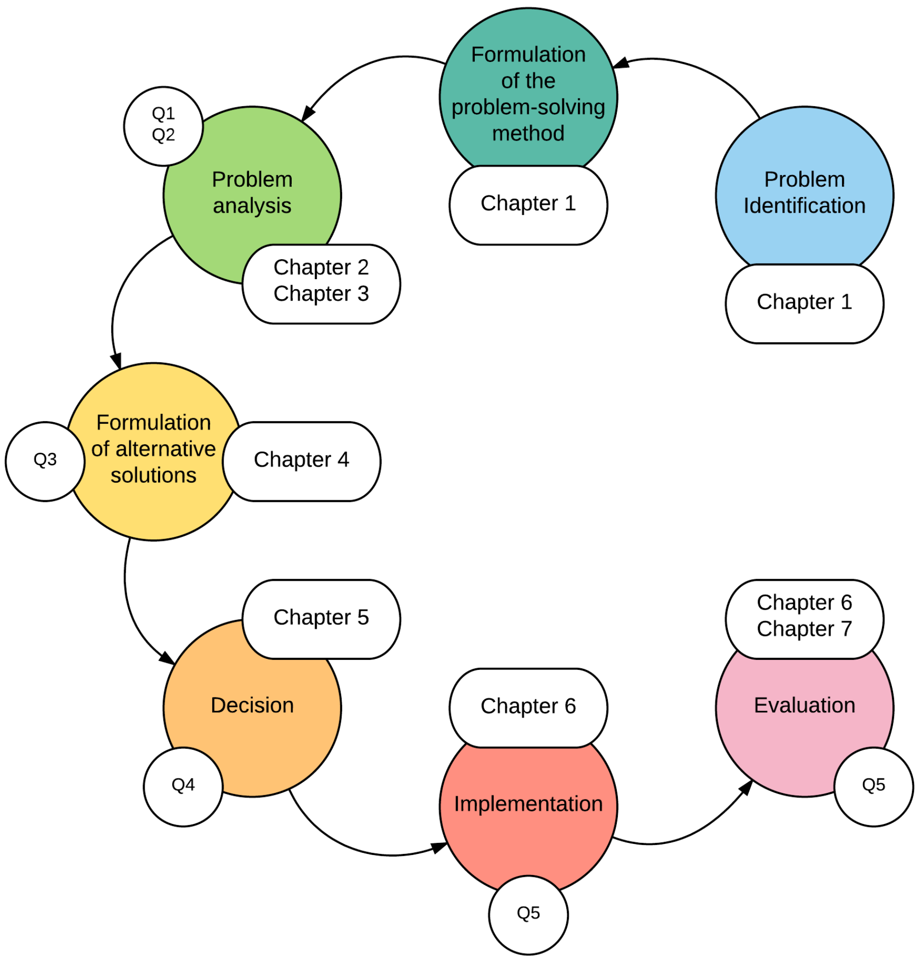


Figure 1.6 - Research Outline

In chapter 1 the problem is identified and the problem-solving method formulated. In chapter 2, the results of the research on the current way of working are described. Finding a way to measure and assess the process is reported in chapter 3. Then, in chapter 4, a literature research is performed to find ways of improving the RE process. In chapter 5, a new way of working is specified for Movares by using input from interviews, the current way of working and literature. In order to implement this new way of working, the implementation procedure is described in chapter 6. Following chapter 6, chapter 7 contains the conclusions and recommendations of the thesis.

2. CHAPTER 2 – DESCRIBING THE CURRENT WAY OF WORKING

The research object of this thesis is the requirements engineering process at the system integration department. Before diving into solutions that will improve the current RE process, it is important to describe and analyze the current way of working at Movares. This is especially important, since the process is currently not explicitly documented at Movares. The current way of working was previously only partially documented. It is now documented as complete as possible in this research by interviewing Movares employees, searching and reading existing Movares documents and by applying industry standards from Leidraad (2013).

In this chapter, it is pursued to answer research question 1, and thus to describe the current way of managing system requirements. In order to do so, first the context of system requirements documents is described in section 2.1. When this is clear, the main document of research, the SRS, is described in part 2.2. In section 2.3, the SRS is further described by mapping the activities that lead up to an SRS. After the SRS is carefully described, in section 2.4, contract specifications are specified. Following the descriptions of the main documents in the RE process, the involved actors and roles of the process are discussed in 2.5. In part 2.6, a different subject is considered, in which we deliberate the current way of estimating the amount hours spent per activity or process per project. The chapter is concluded in 2.7.

2.1. CONTEXT OF SYSTEM REQUIREMENTS DOCUMENTS

In describing the current way of working concerning system requirements engineering, first the context of this process is outlined. The relation of SRSs to other documents and the bigger process within a project are deliberated in this section. System requirements specifications and contract specifications are self-containing documents, but are still part of a bigger process within a project.

Working on infrastructure projects, all parties involved have to follow the standards specified by ISO norms such as ISO 15288. However, on a more detailed level, the concrete work processes differ per project and per person, there is no detailed documented standard yet within Movares. Describing and understanding the current way of working is necessary in order to find opportunities for improvement. It even is an improvement in itself, since the process becomes more concrete, visible and understandable. Within the organization, there are a lot of scattered brief documents and statements indicating the input, output and goal of particular work processes. By comparing these different documents with information obtained from interviews, the exact processes that lead up to an SRS document became clear. This process is too complex to capture in a single, visual overview. Next to this, it is necessary to understand the context in which a system requirements specification play a role. For this reason, multiple overviews are made from different perspectives.

The first perspective that is discussed is the perspective of the V-model, which is used industry-wide. A visual overview is made and presented in figure 2.1. The concept of the V-model is already explained in chapter 1, however in this new model in figure 2.1 it is combined with the ‘fish model’. The fish model (like the V-model) originates from the IT sector, and is also applied as a development model in infrastructure projects. The first ‘fish’ in the model from figure 2.1 is more detailed than the other two. This is done for visual reasons, but in general the same steps are carried out throughout the consecutive ‘fishes’. The main difference between the consecutive levels is the specification level. After each step of engineering, the system is further decomposed into more specific subsystems or objects as explained in section 1.1. Then, for each subsystem, new alternatives are generated, design decisions are made and from these design decisions new requirements are specified. This is performed top-down, starting at the top of the system and gradually engineering and specifying the different subsystems and objectives. The specification level may differ per object, and depends on whether or not there is a significant risk in not specifying a certain object any further, and whether or not the client wants to leave room for contractors to come up with creative solutions (Leidraad, 2013). The visualization from figure 2.1 also shows how a CRS is further specified and engineered to finally arrive at an SRS. From an SRS, a definite design is made. One of the final steps is producing a contract specification.

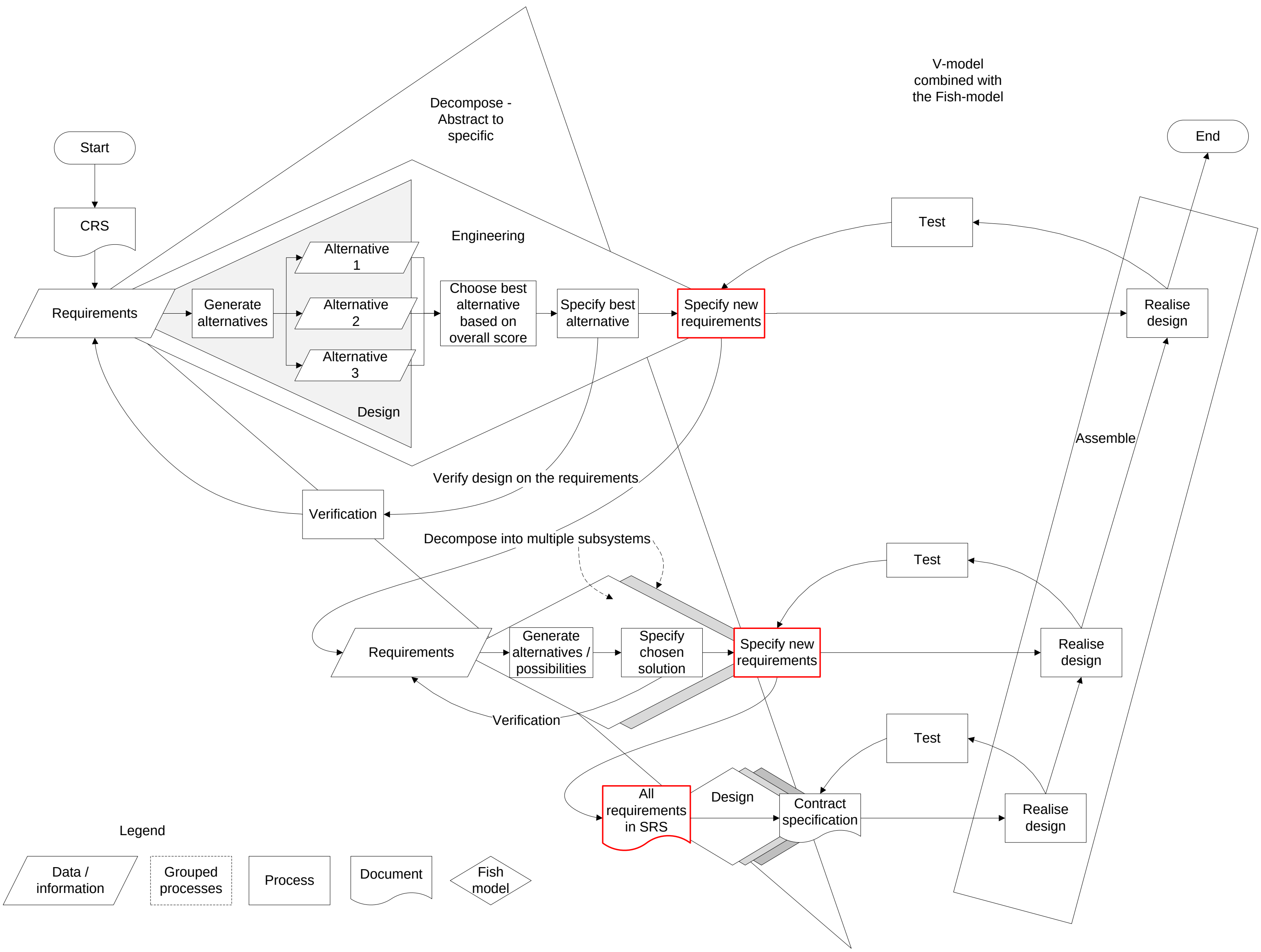


Figure 2.1 - V-Model combined with the fish model

As can be deduced from figure 2.2, after a contract specification is assigned to a contractor, the executive design usually still has to be made. It could be that Movares engineers the executive design, it could also be that a different organization or the contractor itself makes it. After the executive design is formed, the contactor can actually start realizing and building the system bottom-up. The realization process makes up the right side of the V-model.

Before engineering, the client decides to which depth the system has to be engineered, which is established per contract. This decision directly determines the point where the iteration process is cut off. The contract can be regarded as an extra dimension to the process, hovering over the other processes, creating boundaries and limiting decisions. Depending on the type of contract or strategy, the line is drawn on how detailed requirements and objects are specified.

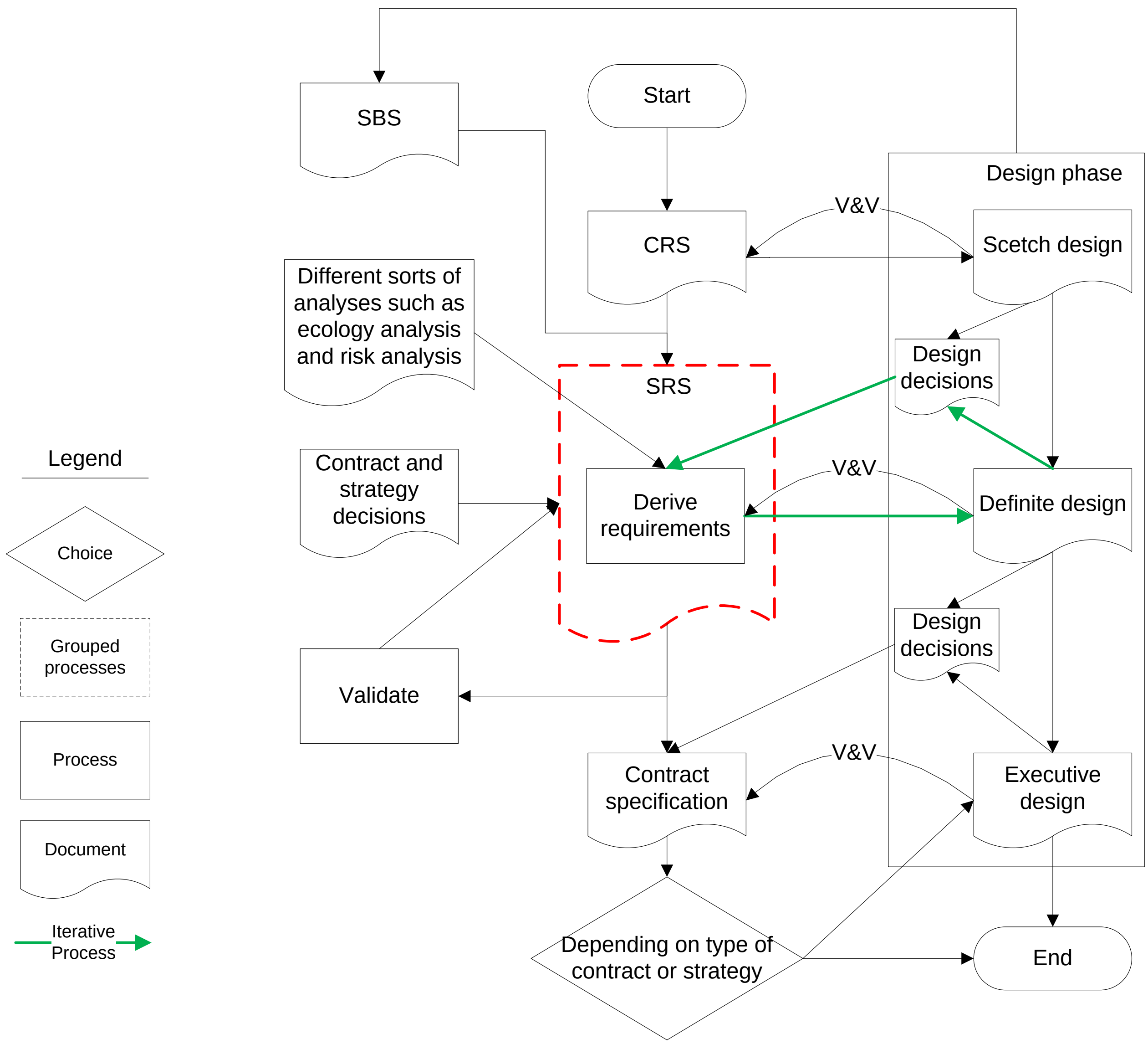


Figure 2.2 – Context of requirements documents

Figure 2.2 displays the engineering process from a different point of view in comparison to the V-model from figure 2.1, focusing mainly on the left side of the V-model. From this figure, the iterative relationship between the design phase and SRS process could be clearly observed. Requirements are mainly derived from input such as the CRS, design decisions and different sorts of analyses. Within Movares there are a lot of different departments and disciplines. Some of these departments perform analyses that are handed over to the requirements engineer in order to derive requirements from these reports. For example, ecology analyses are performed, among other reasons in order to check whether or not there are endangered species of plants and animals that have to be taken into account.

Now that the context of the system requirements engineering process is extensively described in section 2.1, the next step is to examine the main documents that are made during this process, SRSs and contract specifications.

2.2. DESCRIPTION OF AN SRS

In this thesis, the main goal is to research the possibilities of standardizing the process of making SRS documents. For this reason, it is logical to continue by explaining the function and use of an SRS in this first section.

SRS stands for System Requirements Specification, and is mainly composed of system requirements. It is a central document in infrastructure projects. The goal of an SRS is to describe a system's functionality by mapping the technical requirements that are imposed on future designs and measures. System requirements themselves are the requirements that limit the future designs of the systems. System requirements in an SRS are the starting points for designing and engineering the physical system. The process of deriving and managing requirements is called requirements engineering (RE). The goal of requirements engineering in general is the preparation of a contract specification as a basis for the future development of the system (Brosch et al., 2012). The importance of requirements engineering is stretched by the research of Fernández et al. (2012). In their research, they define three main categories of project-specific effort; effort for requirements engineering (REQ), effort for change requests or rework (CR) and effort for other tasks in the project life cycle (SWL). In their research, Fernandez et al. (2012), mainly analyze REQ in relation to CR and use these three categories for statistical testing. Interesting is that they found a negative relation between REQ and CR. This means that when more effort is put in requirements engineering, there will generally be fewer change requests or rework.

A requirements engineer is the person who deals with this process and makes the SRS. The verification of the physical concept solutions and functionalities of the realized system, results from these requirements. An SRS is not a static document, but is influenced and changed during the design process, as is visualized in figure 2.3. In the design process, design decisions are made. From these decisions, new requirements can be derived. Therefore, an SRS has an iterative character, interacting with designers in each iteration. Figure 1.3 also represents this iterative process. Only later in the RE process, requirements are put into a requirements document. Before that, a requirements document can be considered as a platform in which the requirements are managed. During the process of deriving requirements, the existing requirements already influence the design process, which is performed at the same time during which the SRS is being constructed. An SRS document is the result of this process.

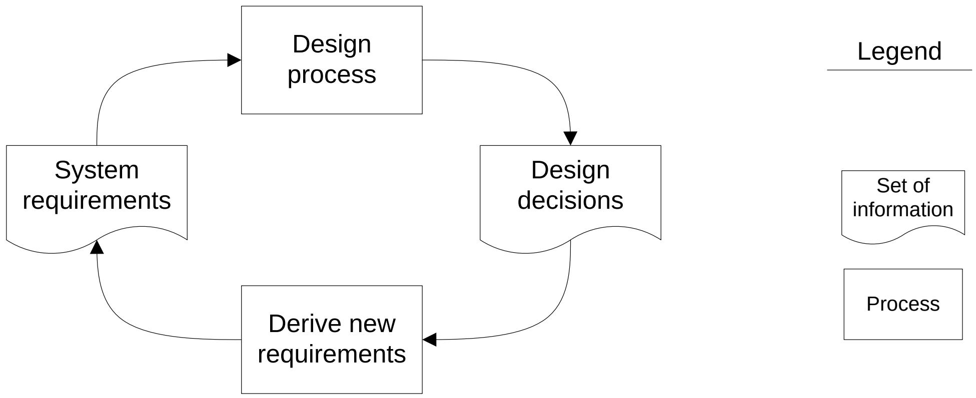


Figure 2.3 –Dynamic perspective on the SRS process at Movares

An SRS usually starts as a "translation" or derivative of the customer requirements from a verified CRS, complemented with requirements from other inquiries and reports. This way, starting requirements are generated, which is input for the engineering process. In case a CRS is not produced, the SRS starts from either customer requirements or the functional breakdown structure (FBS) that will be discussed in section 2.2.3. As mentioned before, an SRS can be viewed as a platform, but during the design- and engineering process, the engineers need input in the form of requirements. Therefore, intermediate versions of the SRS are turned over, or sometimes more informal loose collections of requirements. These intermediate versions usually have names such as SRS-A, SRS-B, SRS 1.0, or SRS 2.0.

The requirements are generally managed in a tool called Relatics, which is a cloud platform used to control information within a project. In the case of Movares, mostly requirements are managed in Relatics. Microsoft Excel is also an alternative option. The choice for either Relatics or MS Excel, or both (sometimes MS Word is used as well), depends on the complexity of the project, the format of input of the requirements and the effort involved in transferring the requirements. Larger and more complex projects are usually managed in Relatics, whereas for smaller projects it is often not worth the extra effort of transferring the requirements from one platform to the other, and is therefore managed using MS Excel. Relatics is the most ideal option as platform of choice. The reason for this is that requirements are easily linked to other aspects such as functions and objects, and be put in the correct format. These links are especially convenient if for example in a later stadium is decided to change the structure of systems or the name of an object. This is then automatically modified for all the linked requirements. In the end, the SRS document is exported to .pdf- format, signed, and handed over to the next parties.

Throughout the engineering process, the system requirements are translated to design products, such as drawings and design decisions. Design products are a concrete expression of a certain requirement. Usually, multiple alternatives are generated from the given requirements. Alternatives are assessed on specific predetermined aspects. These aspects are given a relative weight, which results in one overall grade per alternative. Then, one alternative is chosen and further examined and specified. This decision implies design decision, from which new requirements arise, which are then embedded in the SRS. This is an iterative process.

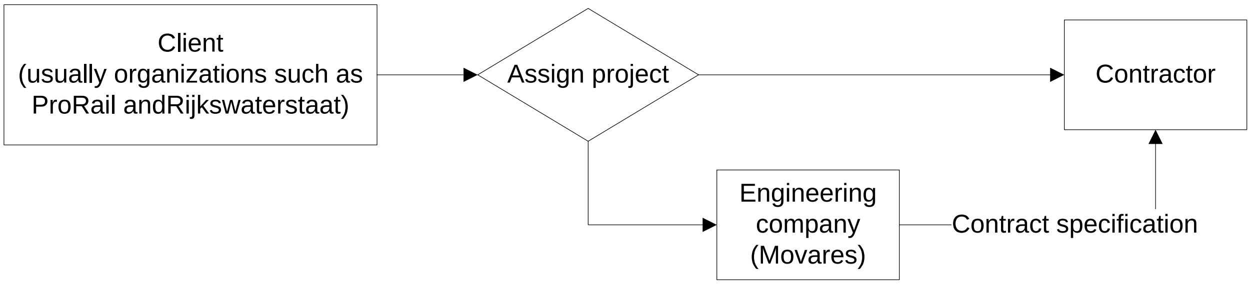


Figure 2.4 - Contract specifications in a general context of projects

In the end, an SRS document comprises of a complete list of system requirements and relating information. This is the output of the whole previous process of deriving requirements. The SRS document itself is input for other documents, such as the final designs and contract specification. This document functions as the description of the work to be done by the contractor, and thus forms the base of the contract for the contractor. The relationship between the client, engineering company and contractor is shown in figure 2.4. The contract specification consists among other things of technical requirements derived from the SRS and is further elaborated upon in section 2.4.

2.2.1. SRS CONTENT

Now that there is a clear view on the definition and function of an SRS, this subsection zooms in on the content of an SRS. An SRS is namely more elaborate than just a list of system requirements.

Next to system requirements, both contract specifications and SRSs consist of other information, which is provided throughout the different chapters of both documents. Before making such a document, there needs to be discussed what the exact content of the document will be, next to the requirements themselves. Currently, there is no standard format for an SRS within Movares, and even SRS documents for different projects of the same client are often very different. There is however some consistency throughout the documents. These common aspects of an SRS document are now discussed. One condition for an SRS, is that the document should stand on itself. It informs the reader on the most important aspects of the project, without having to read other documents. This implicates that the document consists among other aspects of background information of the project, design decisions and interfaces. The aspects of a project that are described in an SRS are:

- Introduction: The introduction usually consists of two parts, general information of the project and an introduction on the SRS:
  - The general information commonly consists of the background information of the project and system, motivation for the project, important measures taken, a ground plan of the site and satellite pictures.
  - The introduction on the SRS frequently consists of the goal and function of an SRS within the engineering process.
- System and system context: This part is an important one and consists of different aspects:
  - The goal of the project and system are described. Also, the definition and description of the system itself and main functionality are discussed. The SBS of the system is also pictured in this part.
  - The current and desired situation, which describe the current and future objects in the area to consider:

        - The current situation describes the area within the system boundaries and direct area outside of the boundaries, the current situation in relation to (third party) objects and infrastructure in the area, land owners and users, destination plans for the area and other relevant aspects around the system boundaries such as access roads.
        - The desired situation refers to the general outline of the CRS and the aspects mentioned above, in relation to the future situation.
    - A general or functional description of the system boundaries.
    - System context, which encompasses the system in relation to other systems. After all, a system is always part of a bigger system. The system context is related to the architectural design, and visually presented in a star diagram with the system of interest in the middle. An architectural design is a schematic design in which the different functionalities and objects including their relation is described using symbols.
    - The interfaces are already introduced in the system context, and further elaborated in a separate part afterwards, including a schematic representation. An internal interface overview in the form of a N^2-matrix can also be provided.
    - Stakeholders should be mentioned and schematically presented in order of importance, primary, secondary and tertiary stakeholders.
- Scope: The scope is related to the previously mentioned system context and describes the boundaries of the project, defining what exactly lies inside and what lies outside of the scope. These boundaries or aspects of the scope can be listed in a table, in which per aspect is specified what lies within and outside of the scope.
- Applicable documents: This contains a table list of binding and informative documents that contain source information on the system requirements that are listed later in the SRS. A system requirement needs to be able to be traced back to source documents, which are listed here.
- Design decisions: Throughout the engineering phase, decisions are made concerning the design of the system. It is important to register these decisions, including the source document. These are listed under design decisions.
- Requirements: The system requirements imposed on the system to be created. This is the core of an SRS, a list of the requirements including attributes which are discussed later in 2.2.2.
- List of modifications/decisions: The engineering process is an iterative process in which requirements are added, removed and changed over time. Decisions on changing these requirements are kept and tracked in a list.
- Verification & Validation: The content of this part leads from the V&V management plan, this is another document which specifies the way a verification and validation of requirements and designs should take place.
- Attachments: Under attachments, there is room for figures or tables, which are referred to in the text, but not deemed suitable in the direct location relating to the text. It is for example customary to put figures such as the SBS and/or FBS in the attachments.
- Other possible information or content: On request by the client, or if deemed necessary or suitable by the project team, other information can be put in the SRS.

2.2.2. LIST OF SYSTEM REQUIREMENTS

In the previous subsection, the content of an SRS is described. The core of an SRS, is the list or specification of the system requirements. For this reason, the system requirements are separately explained in this subsection. System requirements are already briefly introduced and defined, but in this subsection respectively the different types, specifications and properties of requirements are discussed.

As is mentioned, the most important aspect of any requirements document is the list of requirements itself. In the list of requirements, requirements are usually structured according to the requirement type. There are five different types of requirements. These types are 1) functional-, 2) aspect-, 3) external interface-, 4) internal interface- and 5) realization requirements:

1) Functional requirements describe the functional properties of the system after it is delivered.
2) Aspect requirements are requirements on the performance of certain aspects which do not add functionality to the system. These aspects are divided into five different sub-types of requirements that impose conditions on the performance of the system:
    A. Reliability requirements are requirements that specify a period in which a system should perform without failures.
    B. Availability requirements refers to the fraction of time a system should fulfill the required function.
    C. Conservation requirements specify the requirements to the necessary conservation facilities and conservation needs.
    D. Safety requirements ensure safety during and after realization of the project.
    E. Appearance requirements define the appearance of a system.
3) External interface requirements describe interfaces with other and/or future activities and systems/objects outside of the current system (system of interest). Measures need to comply with these requirements in order to not disturb the design and realization performed by third parties and existing systems and objects of the client and third parties.
4) Internal interface requirements are requirements related to interfaces within the system of interest. Within a system, different sub systems or objects are defined, among which interfaces exist.
5) Realization requirements refer to the conditions of the existing objects, systems and necessary temporary systems in the course of the realization phase of the system of interest.

An example of the way system requirements is currently presented in an SRS, can be found in appendix A. As mentioned before and as can be observed in the appendix, individual system requirements are specified using a number of attributes. Again, there is no standard format here, so the specific attributes may differ per SRS. Although there is no standard format, there is some consistency in which attributes are used, these attributes are:

- ID: The unique identification given to a requirement.
- Title: The title of a requirement.
- Description: Short description of the requirement. The description should be SMART.
- Source of requirement: A reference to the specific documents that refer to the necessity of the specific requirement.
- Overlying requirement: A requirement is always related to a system and the function of the system, which is part of an overlying system and function. The same holds for requirements, which can have overlying relations. This relation should be specified.
- Requirement initiator: The actor from whom the requirement originates.
- V&V: Verification & Validation, the way the requirements should be verified. Beforehand, a V&V management plan is made, in which a plan is described on how certain aspects of the project are verified and by whom. In this plan, it is also specified how requirements are verified. Then, per requirement, a verification procedure has to be recorded. This implicates a way of checking that a requirement is actually taken into account in the designs and realization. This procedure is an attribute that is often used in SRSs. These procedures are later embedded in the verification

matrix, which is a separate document. This document is a summary of how requirements are verified. Finally, in the verification report is described what the verification procedures resulted in, and whether the requirements are indeed taken into account in the designs and realization of the system. A V&V management plan can be viewed as input of the SRS, whereas the verification matrix and verification report are output of an SRS.

System requirements have different origins. They are for example deducted from the CRS, the design products, environment analysis, interface analysis and geo-analysis. However, the different origins or types of requirements, all system requirements have certain key common properties. These properties are also called the 'requirements of the requirements', since system requirements have to fulfill these properties in order to be useful. These requirements of requirements originate from an internal Movares systems engineering workshop:

- Identifiable: A requirement holds a unique characteristic or ID.
- Complete: A requirement is complete when it has sufficient level of detail for its purpose.
- Correct:
    - A requirement needs to avoid subjective terms such as "good", "safe" and "and as soon as possible".
    - A requirement is linguistically and rhetorically correct.
    - A requirement should have a pressing syntax, using words as "should" and "must". A requirement cannot be optional.
    - A requirement should avoid the grammatical passive form.
    - A requirement should avoid negatives such as "not" or "no".
- Feasible: A requirement should lead to at least one solution and implementation.
- Necessary: A requirement should be necessary to come to a functioning and complete system.
- Prioritized: A stakeholder should have a priority list in which the stakeholder indicates which requirements need to be fulfilled above others.
- Unambiguous: A requirement can only be interpreted and explained in a single way.
- Verifiable: A requirement should be able to be verified.
- Consistent: A requirement cannot contradict other requirements.
- Traceable: A requirement should be traceable to a source and to a higher level in the requirements-tree.
- Solution-free: A requirement should be free of prescribed solutions, components, suppliers, etc.
- Singular: A requirement can only hold one criterion on which a requirement should satisfy.

Next to these requirements from the systems engineering workshop, a requirement should be SMART. There are different definitions for the term SMART, the abbreviation that Movares uses stands for specific, measurable, acceptable, realistic and time-related (Leidraad, 2013). Although system requirements should in theory fulfill these properties, does not implicate that in practice all system requirements actually meet the requirements. For example, system requirements are currently not prioritized.

2.2.3. FBS & SBS

Two other important concepts that are often presented, but are sometimes overlooked within system requirements specifications, are the FBS and SBS. In most projects, a system breakdown structure (SBS), functional breakdown structure (FBS) or both are formulated. The concept of an SBS has already been explained through figure 1.2. An FBS is a structured hierarchical overview of functions on which the future system should suffice. For example, when considering a new railway station, the main function is to transfer people in and out of trains. This function can be further specified to 'support train traffic' and 'support people traffic'. Supporting people traffic can be further specified to 'transport people', 'facilitate waiting' and 'light the station'. These functions can be linked to one ore multiple systems or objects, as is visualized in figure 2.5. For example, the function 'light the station' can be linked to the system 'lighting system'. An FBS and/or SBS are usually attached to a requirements document in the form of an appendix.

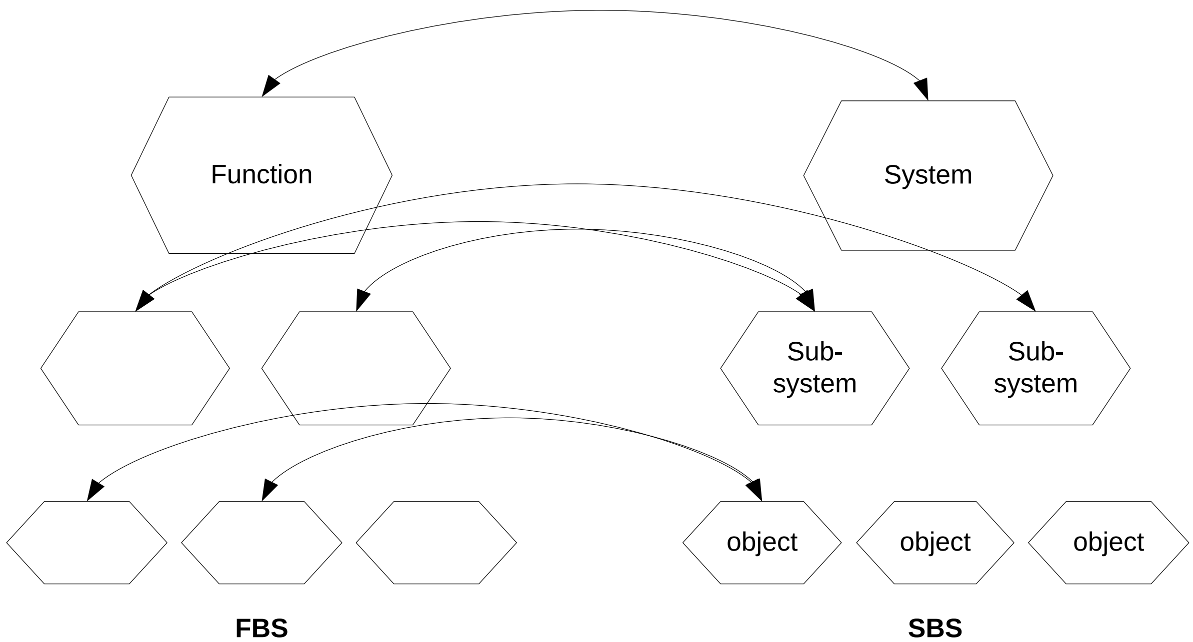


Figure 2.5 - FBS & SBS (Leidraad, 2013)

As functions and objects are linked to one another, requirements are linked to these as well. Requirements are especially often linked to functions, since the process of requirements engineering is often referred to as functional specifying, every requirement should serve a function or purpose. Each stakeholder has certain needs or wishes. A need can be viewed as a function. Using the same example from the previous chapter, one particular need from ProRail could be that a station is lit at all times, also at night. This means there will be a function that a station should be lit at all times. A suitable requirement related to this function would be that the station should always be lit whenever it is in use. Requirements should always be functional and should be linked to a function. This relation is visualized in figure 2.5.

An FBS is especially important since it helps in structuring the requirements and requirements often originate from an FBS. Additionally, functional requirements, which are the most important and form the bulk of a requirements document, should always be linked to a function in the FBS. It is a tool for a requirements engineer to check the completeness of the system requirements and is often used as a starting point in eliciting functional system requirements. Therefore, an FBS helps to improve the quality of an SRS, and should not be overlooked.

2.3. MAPPING THE ACTIVITIES THAT LEAD UP TO AN SRS

The processes and activities that are analyzed in detail, are those directly involved in making the content of an SRS. These processes are similar to making the requirements document for the contract specification. Sections 2.1 and 2.2 already give a notion of the context, function and content of an SRS. However, the exact and specific activities that lead up to an SRS document remain somewhat vague. For this reason, section 2.3 describes the detailed processes and current way of working in direct relation to the SRS.

Before diving into the activities within an SRS in detail, it is best to describe the main activities. These main activities are pictured in figure 2.6. The SRS process starts with discussing the SRS strategy and content. This means it is discussed to which detail the system is specified, and what the SRS will look like, for example with regard to the chapters and format. After the SRS strategy is discussed, activities 2 and 6 can be started. The other main initial activity is analyzing (and waiting for) the different sorts of input, such as analyses from the different engineering disciplines. The requirements engineer receives input from all kinds of sources such as the CRS, different sorts of analyses, design decisions, interfaces, etcetera. From these inputs, requirements are derived. These requirements are then allocated to an overlying requirement and to a location on the requirements hierarchy, the FBS and/or SBS. This makes for a structured and better readable SRS. Next, the requirements are verified and checked for quality and usability in the design process. Once a requirement is set, it is sent to the engineers of the design team, to further specify their designs to the predetermined level of specification. Parallel to this iterative process of deriving requirements and designing, the other chapters that make up an SRS are written. Next, the document is put in the right format and validated for quality. Finally, the document can be signed.

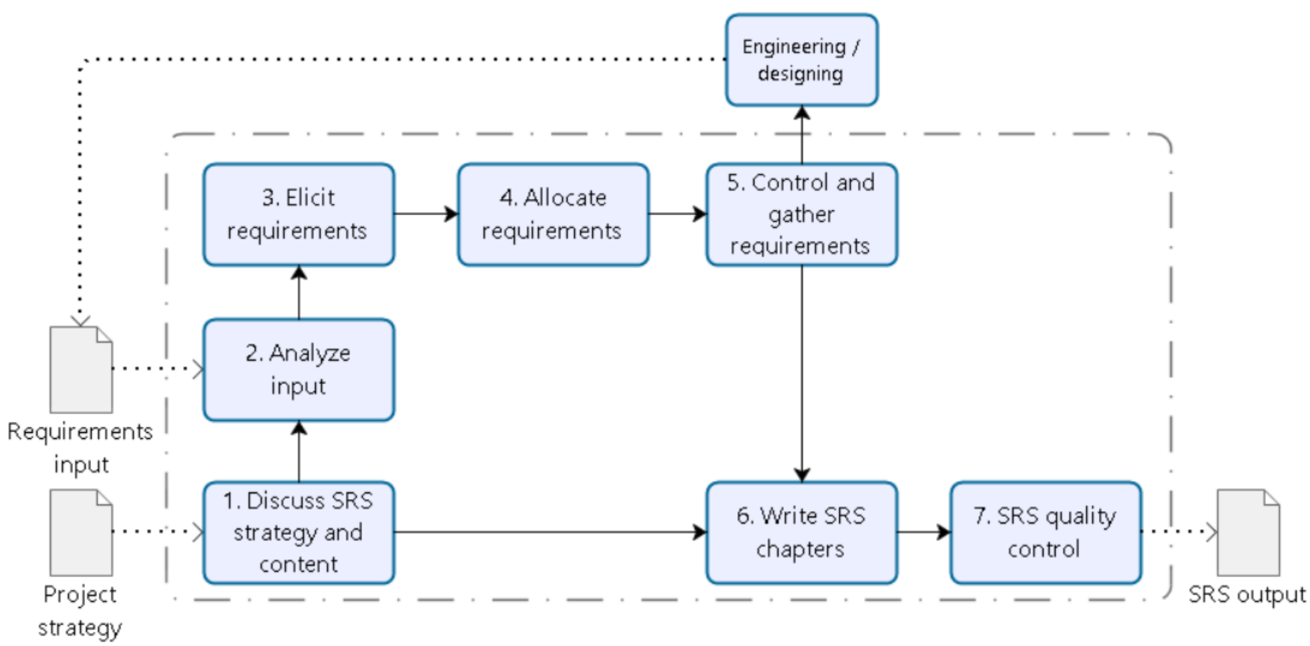


Figure 2.6 – Main SRS activities

The activities from figure 2.6 are the main activities for an SRS. All of the sub activities and detailed relations to each other are described in a schematic overview, which is provided in appendix B. These processes are described using process mapping, in which the process is mapped as a flowchart. A flowchart shows how the process really works, instead of how the process should work (Brook, 2014). There are a few loops with external processes, in order to show the iterative character of deriving requirements and making an SRS document. This process is not standardized or documented. This implies that there are currently different ways of working. However, from different documents and interviews with requirements engineers, it is tried to find common ground on the specific activities that lead up to an SRS.

In appendix B, each activity is assigned a specific color. These colors relate to the model from Sommerville & Ransom (2005), who developed a model of best practices for requirements documents. We adapted this model to fit RE activities in general. Each activity is evaluated and assigned 'not used', 'sometimes used', 'semi-standardized' or 'standardized' and the corresponding color:

- Not used: The requirements engineering activity is (almost) never used. In this case it is not visualized in the model.
- Sometimes used (orange): The requirements engineering activity is sometimes used, or it is used without using a standardized prescribed way of working.
- Semi-standardized (green): It is recorded what the activity is, when and by whom the RE activity should be executed. However, the way this is guideline is executed, is not prescribed on paper.
- Standardized (blue): The concerning requirements engineering guideline is always applied in the same, standardized way. The way this is guideline, is executed is prescribed on paper.

2.4. CONTRACT SPECIFICATIONS

In the previous sections, SRSs and the context of SRSs within Movares are explained in detail. The other main document that is part of the system requirements engineering process, but is researched to a lesser degree, is the contract specification.

The contract specification includes the complete description of work to be carried out to complete the project and is part of the contract documents. It usually consists of multiple documents such as system requirements, process requirements, and annexes such as analyses designs and design decisions. The contract specification ends up going to the market for contractors to bid on. It forms the basis of the contract between the client and contractor. The purpose of a contract specification is to define features of the technical products, which the contractor is to provide to the client (Fisher, 2004).

Figure 2.7 shows the configuration of a contract specification. The entirety of the contract comprises of multiple documents and components. The general section usually identifies the general agreements, regulatory requirements and classification rules that are to be satisfied by incorporation of certain design and construction features into the system (Fisher, 2004). The contract specification requirements part, or requirements section of the contract specification is the document that holds the system requirements, and is quite similar to an SRS. The process part of the contract specification usually defines the part of the procedure that is to be followed in achieving the other part of specification, either in the design or construction process (Fisher, 2004), this often includes the way requirements management should be performed. Annexes comprise of a broader scalar of documents, including the designs and different analyses, for example from conditional disciplines. An important remark is that in practice, clients often make the general part and process part of a contract specification themselves.

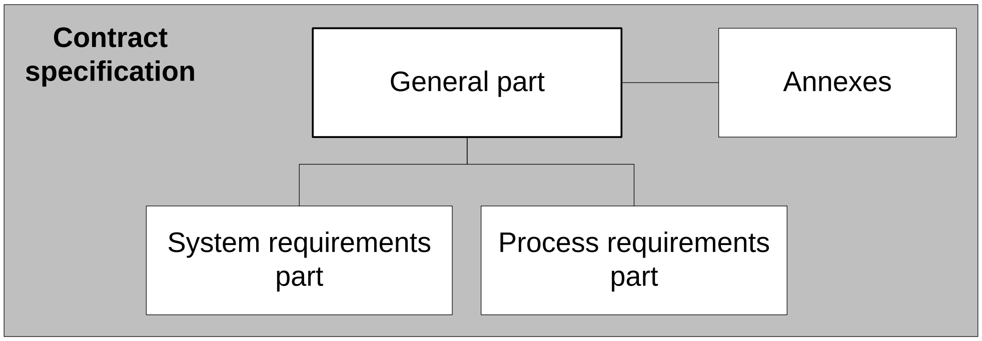


Figure 2.7 - Contract specification according to Rijkswaterstaat

As briefly mentioned in section 1.3, there are two options for making a contract specification; either making an SRS and using that as input for the contract specification, or not making an SRS and directly producing a contract specification. In principle, the same process is completed in order to manufacture a contract specification. This is visually demonstrated in figure 2.8. Within a contract specification, the list of requirements is shown in the same way as is within as SRS, which is specified in part 2.2.2. In the case of option 1, a good SRS is usually direct input for the contract specification. An exception is, as figure 2.8 shows, when new designs and design decisions are made from the finished SRS. Then, new requirements can be derived before making the contract specification. In option 2, the SRS is skipped completely, and system requirements are put directly into the contract specification.

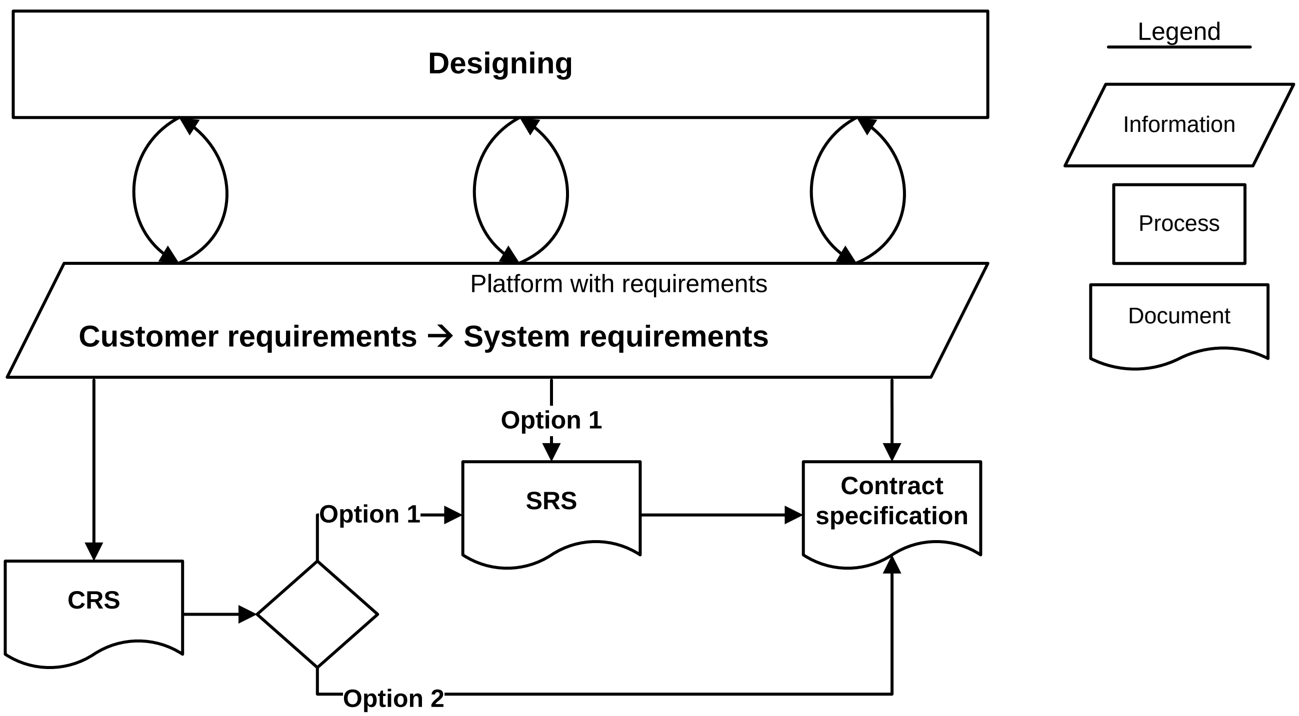


Figure 2.8 - Two paths towards the contract specification

There are a few differences between an SRS and the requirements part of a contract specification. For example, it could be that a system specification is split into multiple contract specifications. If this is the case, it is important to mind and care for the interfaces. Another difference is the goal of the documents. The goal of an SRS is to describe a system's functionality for future designs. The goal of a contract specification is to specify and describe the work to be done by the contractor (Leidraad, 2013). This difference in goal is mainly expressed in the traceability of requirements; whereas for an SRS, traceability is one of the most important aspects, the requirements in a contract specification are not given a traceability, since this would lead to legal issues.

This thesis mainly focuses on the process of deriving and managing system requirements. For this reason, the contract specification is researched to a lesser extent than the SRS. This is mainly because a contract specification is broader than only system requirements, whereas an SRS is mainly focused on mapping system requirements. The added value of incorporating contract specifications into this research, is the possibility of analyzing the requirements engineering process. Both SRSs and contract specifications are involved in this process. When analyzing this process, it would be a shortcoming to only analyze and monitor SRS data, since the whole picture of the process would not be captured. Especially since option 2 is the most commonly used alternative.

2.5. ACTORS INVOLVED IN MANAGING SYSTEM REQUIREMENTS

Analyzing any research aspect, it is always important to consider the stakeholders, and involve these actors in the research. In this section, first the general actors within a project are discussed, followed by the actors involved with the requirements engineering process in particular.

The people involved in managing system requirements are generally part of the project team. In this section, the people and organizations involved in managing system requirements are mapped, in order to consider their wishes and troubles when making a system requirements document. It is particularly important to consider key stakeholders in specific decisions.

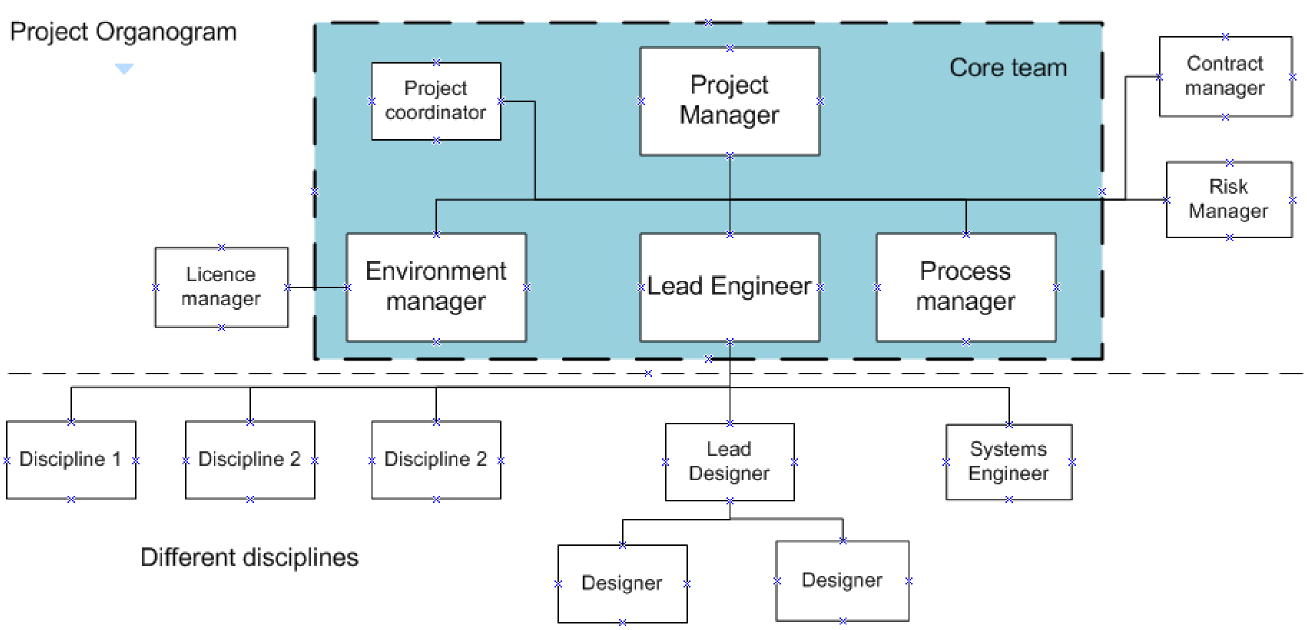


Figure 2.9 – General project organogram

Figure 2.9 represents a general organogram for projects within Movares. This is made at the start of a project. An organogram differs per project, but can be generalized by figure 2.9. An important remark is that one person can take on multiple roles, and can have relationships that differ per project. The exact roles and activities of each actor are predetermined at the start of a project. For example, the system integrator (or systems engineer) is currently linked to the lead engineer, but could also be part of the core team, answering directly to the project manager. A project manager may not even be an explicitly defined role. Its activities are then adopted by another role. Generally, the core team and number of roles increase with the size of a project.

As mentioned before, SI wants to move towards a more advisory and facilitating role in projects, designing processes instead of being part of the process. Deriving system requirements often requires technical expertise. For example, analyses from geological research come in per mail. From these texts and documents, requirements should be derived. After derivation to technical and concrete requirements, the requirements should be usable for engineers and designers, who are also technical minded and oriented. The requirements engineer in most projects is the lead engineer or someone from the SI department. A lead engineer functions as the link between the designers, the core team and the other disciplines, and is capable of understanding both technical and organizational aspects of the project. For this reason, the lead engineer is usually capable of deriving requirements or checking them for quality control. To give a better idea of the roles influencing the content and processes of managing system requirements, an overview of actor interaction is provided in figure 2.10.

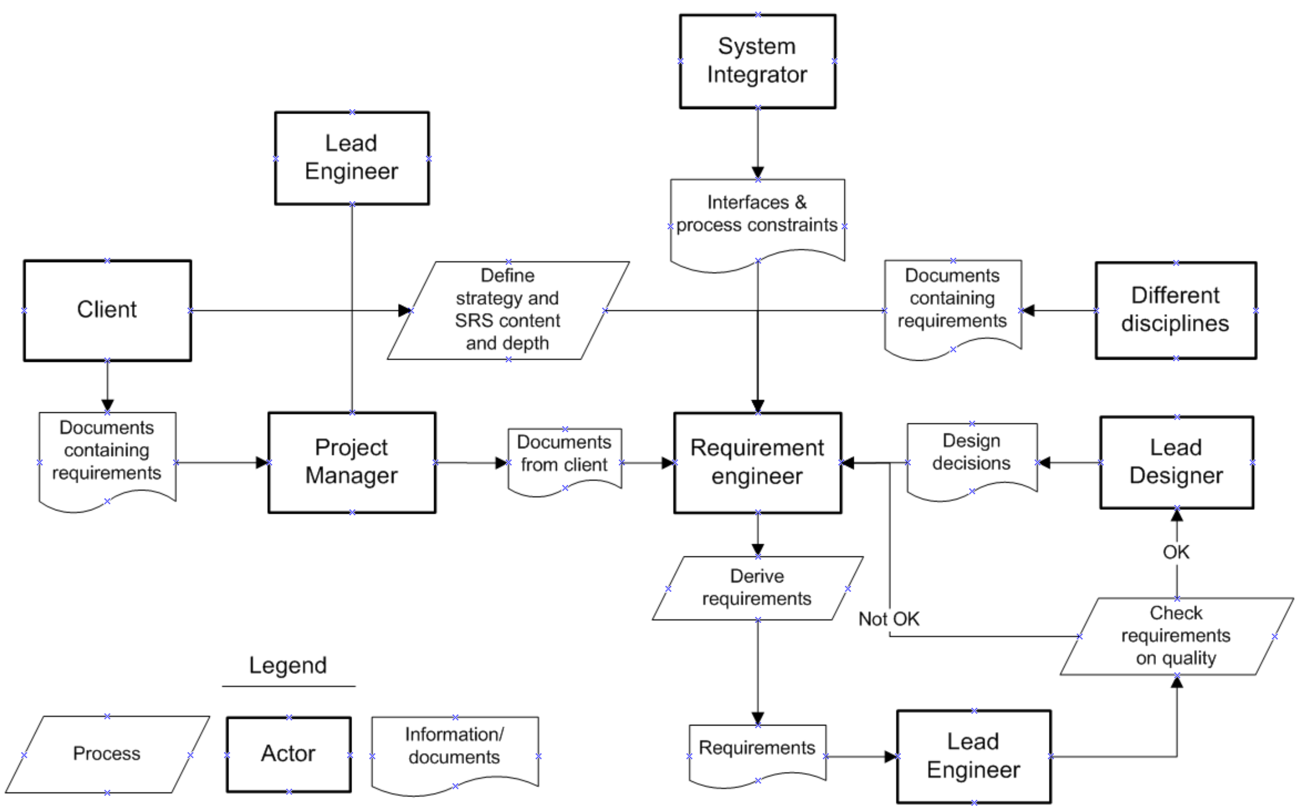


Figure 2.10 - Actor interaction

Again, the iterative process between the requirements engineer and designers is visible in the right lower corner of figure 2.10. Another interesting remark is that the activities and roles on the left side are more involved in preparing and shaping the outlines of a system requirements document, whereas the right side of the figure is more involved with the actual content of such a document.

2.6. PREDICTION MODEL

In the preceding sections of this chapter, the requirements engineering process is thoroughly described. For this final part of the chapter, a different subject is considered. This section examines the current way of working regarding the estimation of the number of hours needed for the requirements documents.

CONFIDENTIAL

Table 2.1 - Number of hours spent on making a requirements document

CONFIDENTIAL

The problem with this model, is that it is based on experience and only used by one single employee. In section 4.7, the possibility of making an improved, data-based prediction model is researched.

2.7. CONCLUSION

In this chapter, sub research question 1 is treated. This chapter therefore describes the current process of requirements engineering, the context of the system requirements documents, and the actors involved in this process. In the current way of working, requirements engineering takes a central place in the general engineering process within Movares' projects. However, most of the times, the process of making SRSs is skipped and the RE process is direct input for a contract specification. Another interesting fact is that employees mainly work based on their own past experiences and the amount of experience that is shared across the company is limited. One of the reasons for this, is the fact that few processes are actually documented and prescribed in order to facilitate a uniform way of working. For this reason, it is already useful for Movares to have the current process of making an SRS documented on paper. This documented process is provided in appendix B. From this point, it is possible to research how this current way of working is performing. This is done in the next chapter.

3. CHAPTER 3 – CURRENT PERFORMANCE AND PERFORMANCE INDICATORS

The previous chapter outlined the current steps and activities that are performed to complete an SRS. Now that the process is described, the next step would be to evaluate the process, answering sub research question 2. There are different methods for evaluating the process, which are discussed in this chapter. These methods are then actually applied in order to evaluate the process on different aspects.

In the next couple of sections, the application of performance indicators is discussed. Operationalized performance indicators are important, since they provide concrete data on the performance of a particular aspect within an organization. These support managerial decisions. There are different types of performance indicators. One dimension in which they can differ, is the organizational level or decision-making level. These dimensions are discussed in separate sections. Per dimension, we search for applicable performance indicators using literature on KPIs and RE, before operationalizing these indicators for the RE process and discussing the results. In section 3.1, Movares performance indicators on a strategic level are elaborated. In the following section, 3.2, performance indicators on a more tactical level are treated, zooming in on the RE process in particular. Section 3.3 covers indicators on a more operational level. After the performance indicators are discussed, we looked to apply these parameters in a quantitative process mapping method in section 3.4. Finally, part of evaluating the current performance of a process is examining the quality evaluation techniques of the end-product in section 3.5. Afterwards, the chapter is concluded in part 3.6.

3.1. STRATEGIC MOVARES PERFORMANCE INDICATORS

As explained above, we first start by analyzing strategic Movares performance indicators that are applicable to this research. These performance indicators are already provided by Movares, so no additional theory is applied in this section. However, the strategic performance indicators are analyzed and related to this research.

There are already some relevant performance indicators that Movares uses. An important indicator and one that is applicable to this research is the performance measurement from ProRail. ProRail is Movares' biggest client and provides a quality report on engineering consultancies once per quarter. The engineering consultancies are judged on every project a particular consultancy has worked on. This is an important quality measurement that also allows Movares to compare itself to competitors. The report is quite elaborate and consists of different measurements, these are shown in figure 3.1.

CONFIDENTIAL

Figure 3.1 - ProRail's performance indicator (ProRail, 2016)

CONFIDENTIAL

CONFIDENTIAL

Figure 3.2 - Comparison of quarter performance to competitors (ProRail, 2016)

CONFIDENTIAL

CONFIDENTIAL

One way of working more methodologically, is by standardizing processes. In the next section, this becomes more concrete as we descent to the more tactical level of the organization. The level of standardization can be viewed as a performance indicator on this level.

3.2. QUANTIFYING STANDARDIZATION ON A TACTICAL LEVEL

The objective of this research is to analyze and find a way to standardize the process of making SRSs. A logical next step in analyzing the process would therefore be to analyze the level of standardization. This way it is possible to track the improvement over time and the impact of applied changes.

In assessing the level of standardization and general improvement of the process, two general aspects of the process are analyzed. The first part is process maturity in subsection 3.2.1. Accordingly, in subsection 3.2.2, the process is evaluated based on the number of standardized elements of the process. In both cases, first the theory is discussed, followed by the appliance at Movares and evaluation of the RE process.

3.2.1. PROCESS MATURITY

The reasonable first step in evaluating a process, is checking the process maturity, which tells a lot about a process. The 'CMM model' is a generally applicable model that describes the five phases that processes go through as they move from immaturity to maturity. This model can also be applied to organizations in general, but was initially designed to describe process maturity (Harmon, 2007). These five levels are shown in figure 3.3.

Figure 3.3 - Five phases of process maturity

The five levels of process maturity are defined as follows (Macintosh, 1993) (Harmon, 2007) (Sommerville & Ransom, 2005):

1. *Level 1: Initial.* Setting up the process. The process is not defined and success depends on individual effort, experience en expertise. Requirements are generally not properly managed and inconsistent.
2. *Level 2: Repeatable.* The process is repeated. At this level costs can be tracked, the process can be scheduled and the functionality is defined. The organization has introduced repeatable procedures for requirements management.
3. *Level 3: Defined.* The process and its activities are documented and standardized.
4. *Level 4: Managed.* The process is measured and controlled. Detailed measures of the process and product quality are collected, and controlled accordingly.
5. *Level 5: Optimizing.* Continuous process improvement, which is enabled by quantitative feedback from the process and innovating ideas and technologies.

In this definition of process maturity, it is desired for an organization to reach the final phase of maturity, continuously optimizing processes (Harmon, 2007). This theory can be applied to the RE process of Movares. When analyzing this process, Movares is currently operating at level 2. The process is performed many times, it is scheduled, costs are budgeted and billed accordingly, and the functionality is defined. However, the exact activities and content are not documented or standardized. In chapter 2, an incitement is provided to improve the maturity to level 3, by describing and documenting the current way of working at Movares. As stated in chapter 1, the goal of this research is to implement a continuous improvement plan, reaching maturity level 5. Levels 4-5 require the process to be monitored and controlled (Macintosh, 1993).

To improve process maturity, a critical success factor is to increase the use of requirements engineering standards (Sommerville & Ransom, 2005). Documenting and standardizing the process, would increase the maturity level to level 3. These standards are further explained in section 3.2.2.

3.2.2. EVALUATING THE LEVEL OF STANDARDIZATION

In order to evaluate the level of standardization more directly, other models are needed. To make this more concrete, the process is evaluated based on the number of requirements engineering best practices and the amount of RE best practices being standardized. When more of the applied RE best practices are standardized, the RE process in general is more standardized as well.

Using more requirements engineering standards may directly lead to more time spent on some areas and thus higher costs, since the SRS document will be more elaborate and there will be more time spent on aspects such as prioritizing requirements. However, the costs of rework will be lower, since there will be less rework needed when more RE standards are used. Next to this, when processes are standardized, costs tend to be lower (Beimborn et al., 2009). This implies a trade-off between quality and cost. Apart from the costs or the trade-off, the level of quality depends on the extent to which each RE standard is used. Next to this, the lead time may indirectly decrease because the process will be more controlled and repeated activities take less time to perform then reinvented activities.

Sommerville & Ransom (2005) have developed a model of best practices for requirements documents. When more of these practices are standardized, the quality of the requirements documents will improve. This model is expanded with other best practices from literature on RE and adapted to apply to the Movares SRS process (Arayici et al., 2006). This is presented in appendix F. Sommerville & Ransom (2005) use their model to evaluate the level of standardization of the RE process. Using and standardizing more requirements engineering standards will lead to an improvement of quality. Each standard or guideline is given a score and value of 'not used' (0 points), 'normal' (2 points) or 'standardized' (3 points):

- Not used (0): The requirements engineering guideline is (almost) never used.
- Normal (2): The requirements engineering guideline is sometimes used, or it is used without using a standardized prescribed way of working.
- Standardized (3): The concerning requirements engineering guideline is always applied in the same, standardized way. The way this is guideline is executed is prescribed on paper.

Sommerville & Ransom (2005) also provided a formula which allows an organization to track the improvement of this score over time when applying RE best practices. This is done by using the following formula:

$$Improvement = \frac{current\ assessment\ score - previous\ assessment\ score * 100}{previous\ assessment\ score}$$

This theory can be applied to evaluating the Movares way of working concerning requirements engineering. Accordingly, all 42 RE standards are evaluated and a score is calculated. The score from this evaluation is 87, which is further specified in table 3.1. From this we can conclude that Movares is already applying a lot of best practice guidelines, but Movares has a lot to win with standardizing its current processes. After a new way of working is defined, the improvement is calculated. This improvement is calculated in subsection 5.3.

Table 3.1 – Assessment score concerning the level of standardizing RE best practices

Total points in current way of working	87
Total amount of guidelines not used:	5
Total amount of guidelines "normal":	22
Total amount of guidelines standardized:	13

The maturity level and total points contributed to the process are performance indicators of the process. In general, the RE process performs better when a higher process maturity level is reached and more RE best practices are standardized. In section 3.3, KPIs are discussed that help in decision making on a more operational level.

3.3. OPERATIONAL KEY PERFORMANCE INDICATORS

In the previous sections, the requirements engineering process is analyzed on a higher organizational level. In order to analyze the process on a more operational level, other performance indicators are needed. So far, we only mentioned performance indicators. In this section however, the focus lies on key performance indicators. Key performance indicators or KPIs are the most important measures that inform managers how well an operation is achieving its organizational goals (Boddy & Paton, 2011). KPIs must be prioritized and focused so that only strategic terms of the performance indicators are measured (Wu, 2012).

In the next two sections, in 3.3.1 and 3.3.2, KPIs are discussed for requirements documents and other documents respectively. At the moment, performance indicators are scarcely used during the RE process. Nonetheless, to reach the fourth level of process maturity, it is necessary to measure the process on an operational level. For this reason, it is tried to find performance indicators that help measure the performance of the SRS- and contract specification process by using literature on KPIs and RE. Following the new KPIs from literature, the KPIs are operationalized and analyzed in section 3.3.3.

3.3.1. KPIS IN THE REQUIREMENTS ENGINEERING PROCESS

In order to find KPIs for the RE process, a literature research is performed. It is decided to only use KPIs from literature that are applicable and usable to Movares. Additionally, as explained before, performance measures must be prioritized and focused so that only strategic terms of the performance indicators are measured (Wu, 2012). Strategic terms for the manager are costs and quality, so the KPIs are focused on these two aspects. The costs during the requirements engineering process consist of labor costs, so costs which are related to the number of hours spent on an system requirements documents. Quality is primarily linked to rework. The indicators are also filtered on data which is available without much extra effort, since the costs of performance management cannot yet become too high. Another reason for filtering out indicators which require too much effort, is that it is easier to imply incremental changes, since people will be less resisted to smaller changes (Pardo del Val & Fuentes, 2003). The final KPIs from literature on the RE process according to Sommerville & Ransom (2005) and Wu (2012):

- The elapsed time between system conception and deployment of the SRS.
- The time overrun on making an SRS.
- The percentage of on-time delivery.
- The effort devoted to rework.
- Efficiency, which can be expressed in multiple ways.

Rework is defined as the time spent on changing the document in general or the requirements in specific, after feedback or requests for changes are received, including the time spent on revised versions of the document. In section 1.2, there were already two KPIs defined. Together with the requirements derived from literature on RE, a list of eleven KPIs is made, in which all factors are operationalized and put in a single format, table 3.2. In the eventual dashboard for managing KPIs, KPIs must provide information of the SRS process in general, so therefore averages are calculated.

A lot more KPIs can be thought of. However, KPIs should fit the managerial goals of the organization. Having too many KPIs counteracts providing information on the process' aspects that are most important to a manager. Projects should avoid the collection of metrics that are not used in decision-making and detract from doing real work (INCOSE, 2006). In order to give an even faster and clearer overview of the process, the KPIs are arbitrarily prioritized (Shahin & Mahbod, 2007). This prioritization is based on the amount of information the KPI holds on the process specified to the managerial goals. There are three grades in priority:

1. Most important KPIs: These KPIs provide direct information on the process' performance, regardless of extra context, for example from other KPIs. They individually present information on projects that are most important to the concerning manager.
2. Important KPIs: These type of KPIs are valuable, but do not give as much information as the higher priority performance indicators, mostly since they require extra context in order to compare projects or provide information on the process in general.
3. Supporting KPIs: These performance measures are mainly used to support or calculate other measures, and do not by themselves provide much information on the performance of the process. Although they may provide interesting information on a more operational level.

Table 3.2 - List of KPIs for the requirements engineering process

ID	*Description*	*Formula*	*Unit*	*Priority*
KPI.SRS.1	The elapsed time in days between starting the process (the first day hours are booked) and deployment of the requirements document (final hours are booked)	-	Days	2
KPI.SRS.2	The number of days that the document is overdue on the original deadline	-	Days	1
KPI.SRS.3	The total amount of hours spent on the document, including rework	-	Hours	2
KPI.SRS.4	Number of hours spent on the process per week	KPI.SRS.3 / (KPI.SRS.1 / 7)	Hours	1
KPI.SRS.5	The total number of requirements that end up being used in the requirements document	-	Requirements	3
KPI.SRS.6	The amount of time spent on average per requirement within a requirements document	(KPI.SRS.3 / KPI.SRS.5) * 60	Hours	1
KPI.SRS.7	Number of revised versions (including first version)	-	Versions	1
KPI.SRS.8	The number of hours spent on rework	-	Hours	3
KPI.SRS.9	The percentage of time spent on rework	KPI.SRS.8 / KPI.SRS.3	%	1
KPI.SRS.10	The percentage of actual hours/budgeted hours	-	%	1
KPI.SRS.11	The percentage of requirements documents delivered on time	-	%	1

KPI.SRS.11 is the only KPI that cannot be filled in from individual SRSs, but instead follows from multiple SRSs. For each KPI, an average can be made. After the data is gathered and averages and patterns are calculated, Movares could set targets for future documents.

An important remark for assessing the requirements engineering process, is that the effectiveness of personnel needs to be measured (INCOSE, 2006). For this reason, both the project manager and the main document author are registered when for each data entry. Then, the employees can be addressed on a project´s performance.

The data on the number of hours is retrieved from an application called 'Hours'. The way this works, is that per project there is a certain budget. Within the project, the budget is divided over different work packages, for example the CRS and SRS are two different packages. Within Hours, employees 'book' the hours they have worked on specific booking elements. These booking elements are decided on for each project separately and are usually directly related to the previously decided work packages or activities, but not always. Having this data, it is possible to check when the first hours were booked on an SRS, how many hours were booked on an SRS in total and over what period of time hours were booked. This provides the data necessary for operationalizing the KPIs. In case one project holds multiple SRSs, the data from the SRSs is grouped. This is because it is hard to arbitrarily divide commonly spent time and costs over multiple SRSs, and because the hours spent on multiple SRSs are booked under the same element in the application Hours.

3.3.2. GENERAL KPIS

This research is mainly focused on the process of engineering system requirements, but the key performance indicators can easily be expanded to other products and processes. A lot of deliverable products within Movares have some sort iterative character, delivering multiple versions for control. This number of iterations should be as low as possible according to the first time right principle and to cut costs. Examples of such documents are FIS documents (Functional Integral System designs) and RVTO documents (Rail Traffic Technical Designs) and the contract specification in general. According to Black et al. (2000), contracts in the construction sector should be evaluated in terms of quality, timeliness and costs. These aspects are consistent with the strategic terms cost and quality, which are prioritized by a Movares manager, as discussed in subsection 3.3.1. The general KPIs related to these aspects are presented in table 3.3.

Table 3.3 - General KPIs for Movares documents

ID	*Description*	*Formula*	*Priority*
KPI.1	The elapsed time in days between starting the process (the first day hours are booked) and deployment of the document (final hours are booked)	-	2
KPI.2	The number of days that the document is overdue on the original deadline	-	1
KPI.3	The total amount of hours spent on the document, including rework	-	2
KPI.4	Number of hours spent on the process per week	KPI.SRS.3 / (KPI.SRS.1 / 7)	1
KPI.5	Number of revised versions (including first version)	-	1
KPI.6	The number of hours spent on rework	-	3
KPI.7	The percentage of time spent on rework	KPI.SRS.6 / KPI.SRS.3	1
KPI.8	The percentage of actual hours/budgeted hours	-	1
KPI.9	The percentage of this type of documents delivered on time	-	1

The aforementioned KPIs are all on product and process level. On project level, there are different indicators applicable. Here, teamwork is key, this can be either intra- or inter organizational. This is however hard to quantify. For example, communication is in theory quantifiable, but is in practice not doable due to the amount of effort to measure this aspect. Examples of such measures are the number of meetings held or emails sent, etc. What can be measured are the number of changes or revisions to contract (Cheung et al., 2003) and the difference between original and final contract sum and project duration (Chan, 2004). In the course of this research, these are not further elaborated since they are out of scope, but they are definitely interesting for further research.

3.3.3. PRELIMINARY RESULTS ON KPIS

Now that the KPIs are defined, data is needed to be able to analyze the process of making SRSs at Movares. Historic data is gathered from 21 projects in which SRSs were made. The results are presented in table 3.4.

Table 3.4 – Operationalized KPI values for 21 SRSs over 10 years

CONFIDENTIAL

CONFIDENTIAL

3.4. QUANTITATIVE PROCESS MAP ANALYSIS

After the general KPIs and thus the main process parameters are determined, ideally these parameters could be applied in a quantitative analysis such as process map. This way, the subsequent sub activities could be analyzed and for example bottlenecks in the process could be identified. In this section, different methods for quantitative process mapping are briefly discussed, before evaluating the possibility of applying such methods to the RE process.

Analyzing and determining the current performance of the process is possible in different ways. One direction is building on the process map from subsection 2.3 and appendix B. Techniques that are applicable in construction engineering are Gantt Charts, project evaluation and review technique (PERT) and Critical Path Method (CPM) (Tan & Lu, 1995) (Hayes et al., 2005). Value Stream Mapping (VSM) is a more advanced technique of process mapping, which uses the principles of Lean and from the perspective of value (Brook, 2014).

When looking for the application of quantitative process map analysis at Movares, PERT, CPM and Gant Charts are useful techniques for quantifying process parameters and finding bottlenecks during the process. However, these techniques are only partially applicable. The SRS process is part of a bigger process or project, for which usually Gantt charts are made. The SRS process then is an activity within such a chart. Since the SRS is a process on itself, it should generally be able to apply such techniques to the SRS process. However, this is not possible for two main reasons. The first reason being that the activities within an SRS process are too specified. The techniques mentioned before are quantitative analysis tools, which require specific data on each of the activities, but at the moment there is no such data. It would require too much effort to gather data from sub activities of an SRS on a small scale. The second reason is the iterative character of the process. The SRS process is special in the way that there is a lot of interaction with the design team and that new inputs for an SRS come in at different times during the process. This may be possible to model using VSM, but iterations over multiple activities are difficult to model using PERT, CPM and Gantt Charts in a clarifying way, which makes these techniques inapplicable to the SRS process.

Concluding, in this research we do not perform a quantitative analysis on the specific sub activities within an SRS. Consequently, quantitative scheduling techniques such as PERT an CPM are not applied. However, it is possible to quantitatively analyze the performance of the SRS process as a whole, which is done in the previous sections of this chapter.

3.5. QUALITY MANAGEMENT

In the previous sections of this chapter, the process is mainly quantitatively analyzed using performance indicators. In this final section, a different subject is analyzed, since part of analyzing the current way of working, is analyzing the quality of the products of the process. Measuring the performance of the current way of working is one thing, but measuring the quality of the end-product is another. Of course, both aspects are intertwined and influence (in causal relation) the quality of the product, the amount of rework, the total cost and customer satisfaction.

In subsection 3.5.1, the current way of working concerning quality control is elaborated. The way in which quality control concerning the RE process at Movares can be improved by applying methods from literature, is discussed in part 3.5.2.

3.5.1. QUALITY CONTROL AT MOVARES

Quality control during requirements engineering is important, since requirement errors are the most expensive to fix when found during production, but the cheapest to fix early in development (Katasonov & Sakkinen, 2006). As defined in the objective, Movares wants to find the most economical way of working, regarding cost and quality. At the moment, there is some sort of standardized quality control, but it is only defined at a high, general level, for all documents. For example, for each project a risk matrix is made. From this risk matrix, a risk profile for the project is derived, which states the level of quality measurement:

- Level 1: Random sampling to check whether all requirements are met, if the risks are controlled and if the document is linguistically correct.
- Level 2: Complete check whether all requirements are met, if the risks are controlled and if the document is linguistically correct.
- Level 3: Complete independent check whether all requirements are met, if the risks are controlled and if the document is linguistically correct.

According to this quality measurement level, the subjected document is assessed by using a checklist. Next to this checklist, a system requirements document is reviewed on its content and the system requirements themselves. The current checklist for a requirements document is shown in appendix D. This is unfortunately too general and insufficient, given the dissatisfaction within the SI-team.

Within the team there is already put some thought in what to do with quality measurement. Therefore, a quality measurement system is made by two employees. However, this system ends up not being used. The main reason for this, is that employees are occupied with the more value-adding and obligatory activities, and tracking or minding the performance or quality indicators does not currently have priority. The quality measurement system that was made, is called the dashboard performance indicator. This is a form that can be used for most documents. The concept is that before making a document, the quality of the input is judged and afterwards the quality of the output is judged. After several of these forms, one could make an estimate on the quality of the output, given the quality of input, based on for example the amount of time put in a document. So, this model has two functions; measuring quality and making a prediction model. These indicators are drafted by senior advisors with years of experience and can be useful when applied more. For this reason, the concept of this estimation or prediction model is improved in and further elaborated in section 4.7.

3.5.2. NEW QUALITY CONTROL CHECKLISTS

In this subsection, we try to enhance the current way of controlling quality by using methods from literature on requirements quality control. After all, according to the Movares website, quality of the final product comes first.

Quality control regarding requirements engineering at Movares is mainly done by verification and validation. Verification and validation are two conceptually different activities, but in practice almost inseparable (Katasonov & Sakkinen, 2006), as is within Movares. The relationship between the two activities is displayed in figure 3.4. From this figure, different quality standards on requirements can be deducted.

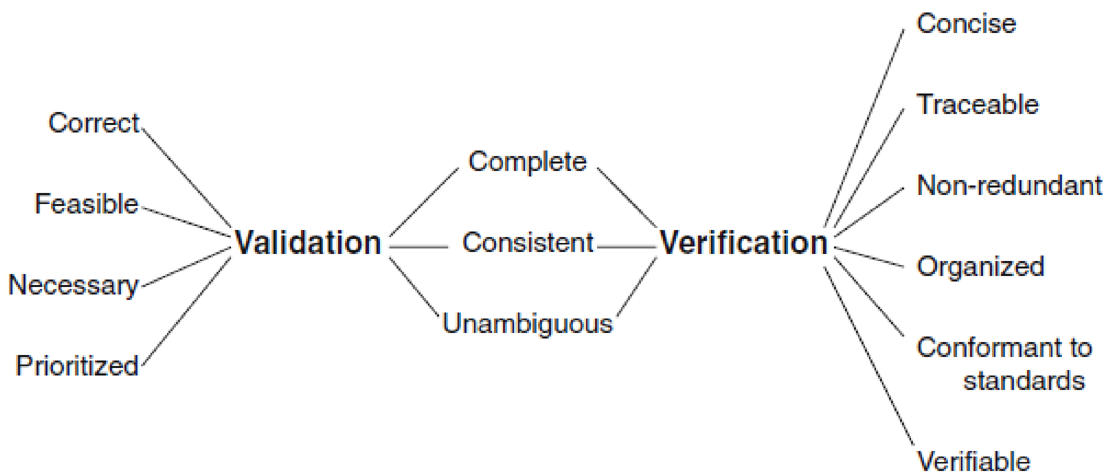


Figure 3.4 - Requirements validation and verification (Katasonov & Sakkinen, 2006)

Interesting is that a lot of the quality criteria are already included in the requirements-specific requirements, which are part of the systems engineering workshop as discussed in part 2.2.2. The problem with this workshop is that it is known among systems engineers, but it is not defined as a standard way of working. From the combined requirements from the workshop, figure 3.4 and Leidraad (2013), one standard for requirements is made, the requirement-specific requirements. This list of 21 requirements can be found in appendix E. This list can function as a quality checklist for requirements.

Additionally, a requirements document is much broader than just the requirements, for this reason a checklist is made for the SRS document in general. This is again done by combining literature standards with current Movares standards from appendix D. Eventually, one single document is made from which SRS content quality can be checked and measured. This new quality control model uses a combination of the dashboard performance indicator, general quality guidelines within Movares, the current document checklist from appendix D and the best practice guidelines from appendix F. This model is set up in such a way that each best practice can easily be answered by either a yes or no answer to whether or not it is applied. The new quality checklist is shown in appendix G, and can be used to help reviewers in reviewing SRS documents.

To make sure that both the requirements checklist from appendix E and the general SRS checklist from appendix G are actually used, the form should be implemented in a standard way of working. This is further elaborated in section 4.4.

3.6. CONCLUSION

The goal of chapter 3 is to answer sub research question 2, thus to find quality and performance indicators of the product and process and to use these indicators to measure the current performance. From the general performance indicators from ProRail, it can be concluded that there is room for improvement at Movares, especially for working methodologically. A tool which helps to track improvement on a tactical level for processes, is the process maturity model. At the moment, the process maturity is at level 2, at which point the process is repeated, but not documented or standardized. The goal is to get the process maturity up to level 5, at which the process is standardized, managed according to control measures and continuously improved. Next to the maturity model, another tool is applied for measuring the progress concerning the level of standardization in the RE process. More standardization will lead to a higher quality product. Using this method, the current way of working is evaluated at 87 points, which will be again evaluated for the new way of working in chapter 5.

Additionally, KPIs are determined on a more operational level, both for the SRS process as for other processes within Movares in general. In part 3.3, these KPIs for SRSs are operationalized and analyzed using historical data. From these KPIs can be concluded that it turns out that making SRSs is in fact not a core process of the SI department. From these preliminary results, it is also clear that the process is not performing well on certain aspects such as cycle time and days late. This proves the fact that data driven performance management helps gaining better insight into processes. Also, there is a high variation in performance and quality indicators. Standardizing the process will reduce this variation.

To conclude, it is safe to say that the process of deriving system requirements and producing requirements documents could improve by standardizing the process and driving on performance measurement based on data. In the next chapter, we continue by searching for ways of improving the current process.

4. CHAPTER 4 – IMPROVING THE RE PROCESS BY USING LITERATURE

The aim of this chapter is to answer research question 3, to find ways in which the current RE process can improve. Using the current process as described in chapter 2 as point of departure, we find methods from literature on RE that are applicable to Movares. Some aspects of the SRS process have already been partially discussed, such as the quality control process. This chapter attends more aspects and in more detail, applying theoretical models from literature to each aspect. Per aspect of the RE process that is discussed throughout the different sections of this chapter, we have first analyzed the current situation at Movares, followed by theory on the subject and the appliance of that theory to Movares towards an improved way of working.

The most important aspects of requirements engineering are discussed in this chapter. But before diving into the individual aspects of the RE process, the requirements engineering process in general is first dissected in section 4.1. Part 4.2 addresses the requirements elicitation process in specific, which is the most important facet of requirements engineering. A part of the elicitation process that has not been mentioned yet, is requirements prioritization, which is discussed in 4.3. An important aspect for Movares is quality, in section 4.4 the quality control process is examined. In 4.5 the concept of change management is introduced. In part 4.6, data management in order to facilitate KPIs is deliberated. Strongly related to data management is developing a prediction model based on data. This is discussed in 4.7. Finally, Movares wants to apply more lean methods. One such key principle is visual management (Brook, 2014), which is elaborated in section 4.8. The chapter is concluded in 4.9.

4.1. IMPROVING THE RE PROCESS IN GENERAL

4.1.1. INTRODUCTION AND CURRENT SITUATION AT MOVARES

Earlier, in section 3.2.2, we researched which RE best practices are already standardized at Movares, and which activities are not standardized, or not even used. The current detailed process is presented in appendix B and discussed in section 2.3. In this first section, we strive to find the differences between theory and practice and look in which ways the gaps in between can be filled.

4.1.2. THEORETICAL MODEL

According to Pandey et al. (2010), there are seven key, overarching requirements engineering practices that need to be explicitly defined, which is a form of standardization. These activities are project creation, elicitation, interpreting & structuring, negotiation, verification & validation, change management and requirements tracing. From these activities, it is striking that there is a lot of focus on preparation and facilitating activities, in comparison to the elicitation of requirements itself. The activities interpreting & structuring and change management are currently explicitly defined at Movares, yet are also characterized as SRS best practices by Sommerville & Ransom (2005) and Arayici et al. (2006). Therefore, Movares should focus on investing more time in these facilitating activities. Since change management is still a somewhat abstract term, this is examined in more detail in section 4.5.

Pandey et al. (2010) have also visualized what the requirements engineering process for software development should look like, which is similar to infrastructure development projects. The process and its activities are visualized in figure 4.1. This process on high level is in practice similar to the current way of working of Movares. However, this visualization only shows the main activities, and not the facilitating practices such as change management. Interesting is that the iterative characteristic of the RE process at Movares, is also explicitly modeled in the model by Pandey et al. (2010).

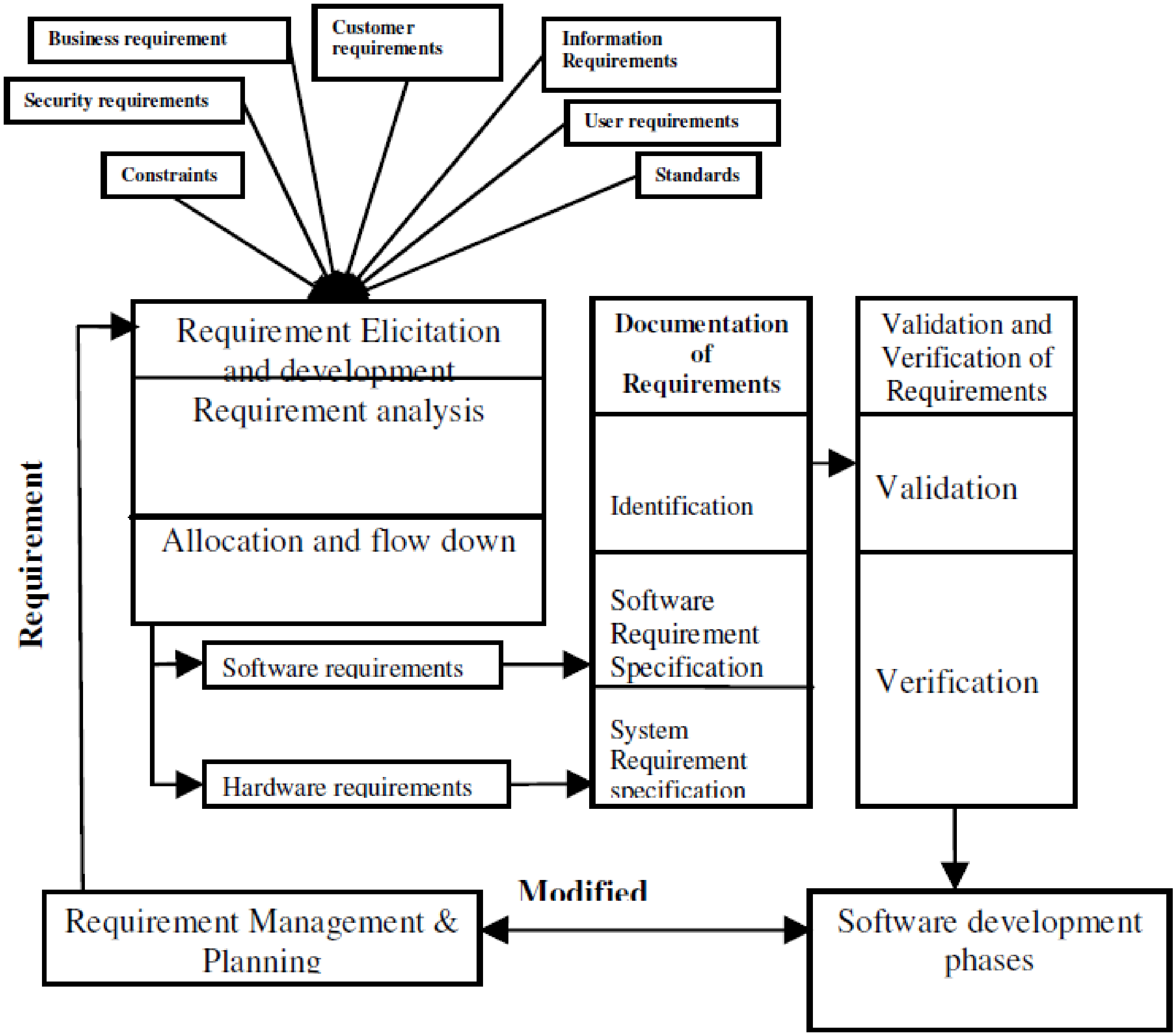


Figure 4.1 - Requirements Engineering Process Model (Pandey et al., 2010)

There are other models that describe the requirements engineering process. Part of another model is presented in figure 4.2.

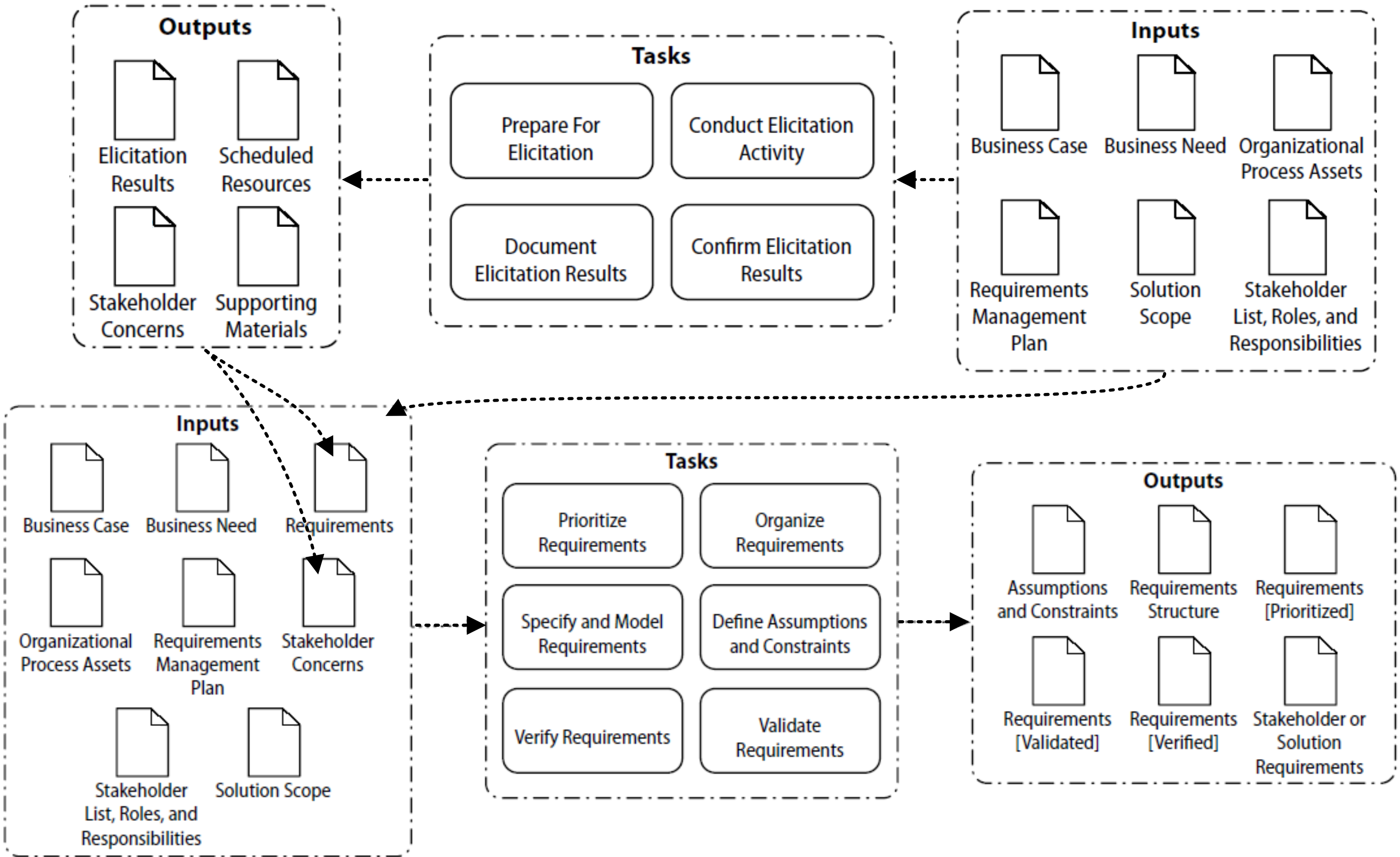


Figure 4.2 – Requirements engineering model (IIBIA, 2009)

In this model, it is shown that the same input influences multiple tasks or activities in the process. These inputs are the business case, business need, organizational process assets, requirements management plan, solution scope and the stakeholder list, roles and responsibilities (IIBIA, 2009). These are overarching inputs, influencing multiple aspects of the RE process.

4.1.3. APPLICATION OF THEORY TO MOVARES

Applying these concepts to Movares, per SRS, a strategy should be made including a requirements management plan and stakeholder management plan. For the SRS process at Movares, inputs for requirements engineering come from multiple different kinds of sources. At the moment, requirements engineers often have to wait for too long before stakeholders deliver input. Stakeholders can be both external stakeholders, or internal stakeholders such as certain involved disciplines. Every discipline and stakeholder providing input such as documents for the SRS process, should therefore be identified and managed. The solution scope is already a standardized part of the current way of working. The other plan that is mentioned in the model, is the requirements management plan, which is pointed out in both models from figure 4.1 and 4.2. Completing this plan, is one of the first activities during the SRS process. In this plan, a lot of different aspects of the SRS should be discussed and set. These aspects are for example a change management plan, the use of traceability, the requirements repository or platform, requirements attributes and the prioritization process (IIBIA, 2009). In the following sections, we continue to discuss the application of important individual activities from the RE process.

4.2. REQUIREMENTS ELICITATION

4.2.1. INTRODUCTION AND CURRENT SITUATION AT MOVARES

In the following few sections of the thesis, individual main activities within the requirements process are discussed. The requirements elicitation process is the most important sub process or activity within the SRS process. The way elicitation should be performed is neither at Movares or in much literature described in detail. This is partly due to the fact it is a complex process with a lot of different types of input and different types of requirements. By combining elements from different articles and interviews with experienced requirements engineers, we have defined a Movares way of eliciting system requirements.

4.2.2. THEORETICAL MODEL

According to Anumba & Evbuomwan (1997), the elicitation process can be divided into three major steps, which also shows from figure 4.3:

- Requirements identification: This is the first step in processing the requirements. The article from Anumba & Evbuomwan (1997) focuses mainly on client's requirements. However, as mentioned in section 4.1, input for requirements engineering comes from multiple different kinds of sources. In the requirements identification, a requirement is written down in informal words. In this step there should already be a weight attached to the requirement for prioritizing in the next activity.
- Requirements analysis and prioritization: The requirements are then analyzed and prioritized. The analysis results in structuring the requirements and linking requirements to identified needs. This is comparable to the FBS for the SRS. Next to this structuring, requirements should be prioritized, based on the weights from the previous step.
- Requirements translation: In this step, requirements are translated to their final, formal form. This implicates that after this step, requirements are structured, prioritized and they comply to all other requirements of requirements, which are specified in appendix E.

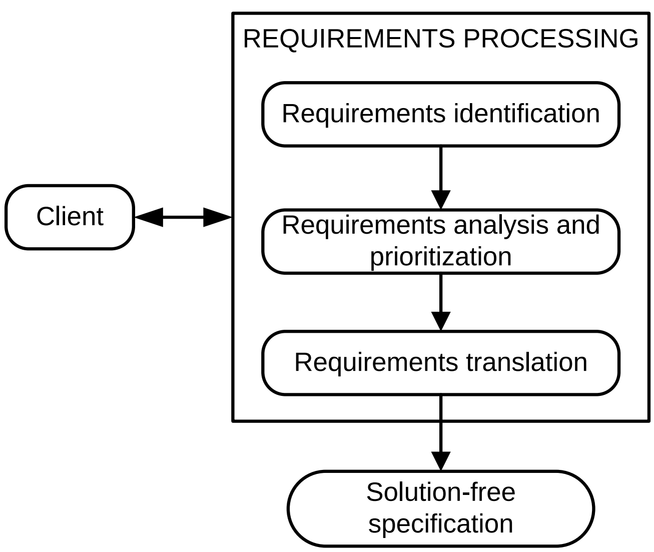


Figure 4.3- Requirements processing (Anumba & Evbuomwan, 1997)

An important side note here is that Anumba & Evbuomwan (1997) only mention functional requirements from the need of a stakeholder, as is the case in most literature on RE. However, there are multiple different types of requirements at Movares. A requirement can also be linked to an aspect, interface or to the realization part of construction. This way all requirements should be constructed (Leidraad, 2013) and is further explained in subsection 2.2.2.

4.2.3. APPLICATION OF THEORY TO MOVARES

Analyzing figure 4.3, requirements identification is the most important sub activity from the requirements elicitation process. It is the identifying of requirements from a source, such as from stakeholders, analyses from disciplines, and interfaces. Generally, these requirements are delivered via some sort of document, such as email, CRS, FBS, or via the designated platform such as Relatics. In the ideal situation, before starting the process of deriving system requirements, among other things a scope, interface and FBS are identified, which form the basis on which requirements are derived (ProRail, 2015). For example, there are requirements identified which describe the in- and output for certain interfaces (IIBIA, 2009). For functional requirements, each requirement satisfies a need from a stakeholder translated to a certain function. The function and requirement are linked to one another. Then, when a function is fulfilled by one or more requirements, the requirements for the identified underlying functions are derived from the overlying requirements, which are linked to the overlying function. The requirements of the underlying functions or sub functions to certain overlying function are chosen in such a way that together, the requirements fulfill the overlying requirement (ProRail, 2015).

Requirements can be separated into two sets; customer requirements and derived requirements, which is a useful distinction for the requirements engineer (Grady, 2006). Both requirements take up a different place in the requirements engineering process. Customer requirements are requirements which are specified from the requirements imposed by stakeholders. Derived requirements are requirements which do not directly originate from stakeholders, but are identified from driving models, for example by specifying a certain function from the function breakdown structure in more detail. The former type of requirements requires a requirements engineer in order to identify the requirements. The latter type of requirements, derived requirements, can also be identified by engineers during the design process. One of the problems at Movares is that within a project, elicitation is still too often performed by the requirements engineer. This role of eliciting requirements should shift more towards the designers and engineers from other departments who provide the input for eliciting requirements, such as the specialized engineering disciplines. The engineers and designers from the engineering disciplines would over time become specialized in eliciting the requirements from their own specialization. They would become faster at it than requirements engineers. Additionally, the requirements would already be directly elicited, instead of the current situation in which the technical specialists would first send the input to the requirements engineer before elicitation of the requirements would take place. Overall, the elicitation process would speed up over time.

A commonly made mistake when deriving system requirements within Movares that is recognized by experienced requirements engineers, is that often the derived requirements describe the system instead of describing the functionality of the system. The description of the system or object should be described by the designs. Describing the objects in requirements leads to an unnecessary large amount of system requirements, which have to be reviewed and verified, which in turn causes unnecessary costs, time spent and inefficiency in general. The final activity from figure 4.3 that is not yet considered, is requirement prioritization, which is separately discussed in section 4.3.

4.3. PRIORITIZING REQUIREMENTS

4.3.1. INTRODUCTION AND CURRENT SITUATION AT MOVARES

From figure 4.3, one activity within requirements processing is requirement analysis and prioritization. Prioritizing requirements is also one of the requirements of requirements and requirements engineering best practices from appendices E and F. It is a sub activity that is often described in literature on RE, which is the reason that this subject is treated separately in this section. Prioritizing requirements is not part of the current way of working at Movares, it is not done at all. In the following two subsections, it is discussed how this can be changed.

4.3.2. THEORETICAL MODEL

Multiple literature studies mention the necessity of prioritizing requirements (Arayici et al., 2006) (Anumba & Evbuomwan, 1997) (Sommerville & Ransom, 2005). It is also one of the requirements engineering best practices as shown in appendix F. Designers or engineers need to easily know how one requirement relates to other requirements regarding importance, and focus primarily on the most critical requirements. There are different ways of prioritizing requirements (ProRail, 2015) (Hatton, 2008). Regarding time and ease of time, the "MoSCoW" method seems to be the best option for requirements engineering at Movares. The MoSCoW method is easy to understand and quick to perform. It is suitable for both small and large numbers of requirements. Also important is that adding or removing requirements does not affect the rest of the prioritization. Requirements are then prioritized in the following way (Hatton, 2008) (IIBIA, 2009):

1) **'M**ust have': Requirements are nonnegotiable. Failure to deliver these requirements would lead to failure of the entire system.
2) **'S**hould have': These requirements have a high priority and should be included if possible, but it would not make the project or system fail in case they are not met. It includes often critical requirements, but which can be satisfied in other ways if strictly necessary.
3) '**C**ould have': Requirements that are nice to have, but less advantageous than "Should have" requirements. They are used in case resources and time allow for it.
4) **'W**on't have', also known as "Would have" or "Wish list": These are requirements that are not unimportant and nice to have, but are currently not implemented in the project. They may be considered for the future.

4.3.3. APPLICATION OF THEORY TO MOVARES

Applying this concept at Movares is not straightforward. Within an SRS at Movares, only actual requirements that must be satisfied are taken into account in an SRS. This inevitably means that 'could have' and 'won't have' requirements cannot be registered in an SRS. However, they can be collected and stored separately, and after an SRS given over to the people responsible for tender management, since these 'nice to have' requirements could play a role in contract specifications according to an employee from the contracts department. Additionally, Movares should apply a differentiation between 'must have' and 'should have' requirements.

4.4. QUALITY CONTROL

4.4.1. INTRODUCTION AND CURRENT SITUATION AT MOVARES

Another important part of requirements engineering is quality control, which purpose is to make sure that the quality of the requirements and the requirements document are of good quality. In subsection 3.5.2, we already created checklists for reviewing the quality of requirements and requirements documents. In this section, these checklists are applied to design a new quality control process. This is done by applying models from literature on RE to improve the current quality control process at Movares. The new quality control process is then elaborated in the following part of this section.

As explained in the previous chapters, there is currently no detailed, consequent or standardized way of checking the quality of an SRS document. The current quality control process is described in subsection 3.5.1. According to employees, this leads to a significant variance in the quality of SRS documents. Therefore, in the subsection 3.5.2, quality checklists are proposed. These checklists can be found in appendices E & G. However, quality control is more elaborate than these two checklists, as will be explained in the following parts of this thesis.

4.4.2. THEORETICAL MODEL

Katasonov & Sakkinen (2006) have developed a review framework for requirements quality control as part of systems engineering (SE). This framework is presented in figure 4.4.

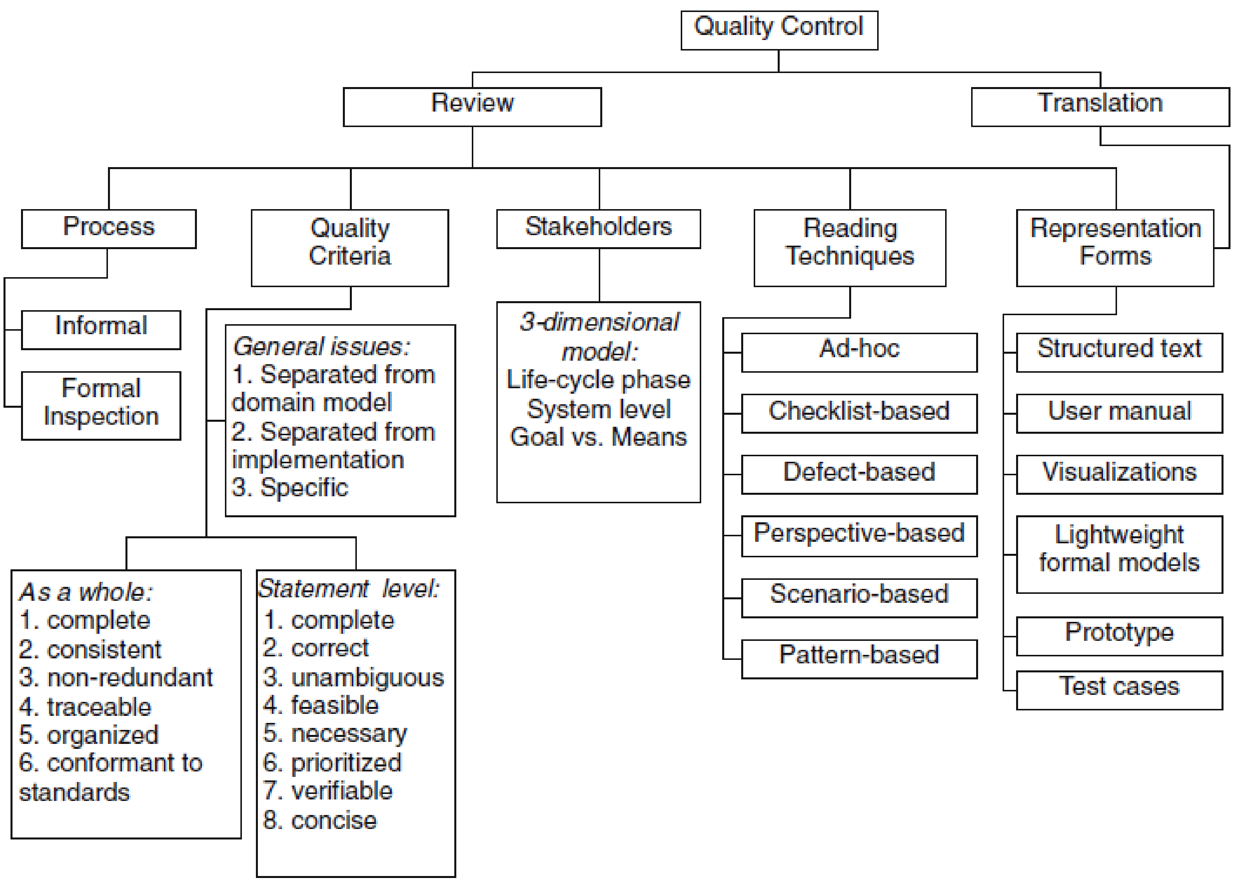


Figure 4.4 - Review framework by Katasonov & Sakkinen (2006)

This model has already been partly implemented in the new quality checklists, applying the 'quality criteria' branch. These quality criteria from the 'quality criteria' branch are consistent with other literature on RE, as discussed in section 3.5.

In the following subsections, a new quality control process is documented by applying the remaining parts of the framework from figure 4.4.

4.4.3. APPLICATION OF THEORY AT MOVARES

Continuing to look to apply the review framework from Ktasonov & Sakkinen (2006), next, the review process is discussed. Throughout the currently defined SRS process from figure 2.6, there are multiple moments of quality control. These moments may be formal or informal, depending on the circumstances. An example of an informal approach is when a designer is checking whether or not he understands the requirements provided to him and is able to apply them. A remark here is that the system requirements should be checked by someone with technical expertise (Katasonov & Sakkinen, 2006). Another example of applying informal quality control is when checking the input quality of an SRS. However, this subsection mainly focuses on the formal quality check at the end of the SRS, since this check is the most important when striving for a consistent lower bound in output quality and is most valued by the client. An example of a formal approach is the so-called 'Fagan inspection'. This research does not apply the Fagan inspection directly, since it is not suitable for a Movares way of working. It is in fact too strict, extensive and better applicable to less error-forgiving industries such as the aerospace industry (Bush, 1990). In order to create a formal review process which is better applicable to Movares, the best practice guidelines from appendix F are exercised. The four best practices that apply to quality control are as follows:

- Both internal and external reviewers are involved.
- Requirements are reviewed by people with multidisciplinary backgrounds (or by multiple people with different disciplinary backgrounds).
- Quality of the SRS is checked.
- Formal requirements inspection is organized.

Using these best practices, a new quality control process is designed, which is visualized in figure 4.5. The different sub activities that make up this figure are explained throughout the following paragraphs. The review process is ideally started during the forming of the SRS strategy. When during this strategy process the content of the SRS is determined, the quality review form can already be made, since it is known what the SRS should look like in the end. This quality form has a standard format and content, but not every SRS is intended to have the same output quality. This is why some elements of the quality form are different per SRS. So before making an SRS, the exact content of the SRS is determined. During the quality control, it is examined to which extent these content criteria are met.

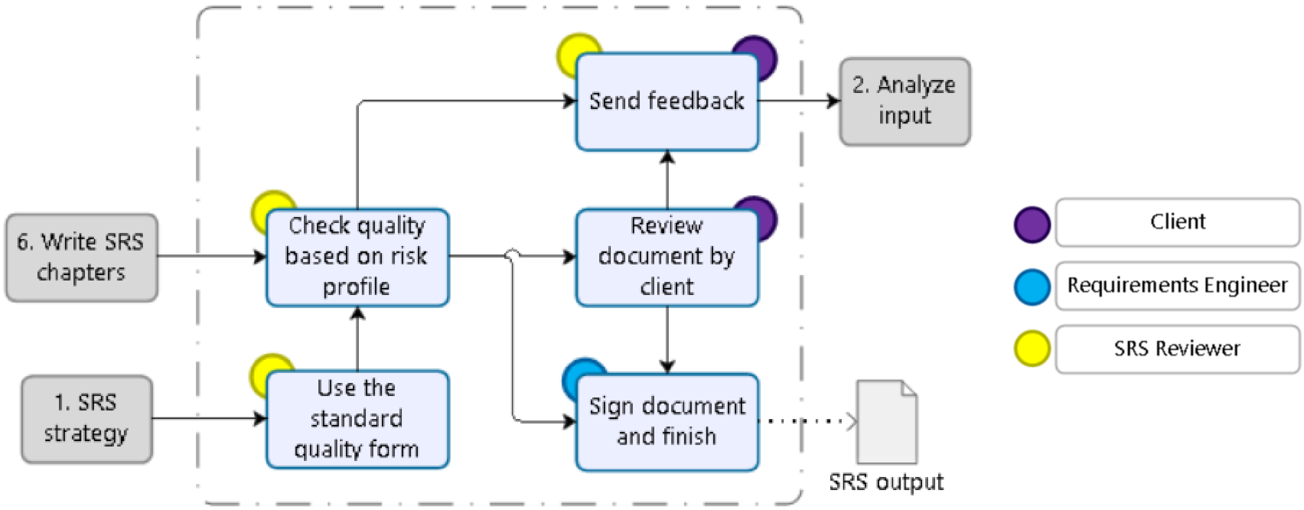


Figure 4.5 – New quality control process at Movares

When the quality control process starts, quality inspectors or reviewers are selected to check the quality of the document before it is sent to the client. In the best scenario, all stakeholders review the requirements (Katasonov & Sakkinen, 2006). However, in practice this is not possible due to high costs and level of coordination required. The number of reviewers depends on the importance, complexity and size of the SRS and is arbitrarily decided. A complex and important SRS requires reviews from multiple different people within Movares. The number of reviews also depends on the earlier mentioned risk profile which is made for each project. This is explained in subsection 3.5.2. For example, a high-risk profile project requires an independent actor from outside of the project to review the document. Examples of qualified people for reviewing an SRS are in general employees from the SI department, lead engineers and project managers.

Currently, the reading technique for reviewing SRS documents is ad-hoc, where little guidance is provided. This is changed in order to standardize the procedure and guarantee a lower limit for quality. The reading technique while reviewing the requirements depends on the readers' knowledge of RE. Less experienced reviewers require assistance in the form of checklists, procedures, scenarios, etc. However, using just a checklist, limits the creativity of a reviewer, and thus the number of defects found (Katasonov & Sakkinen, 2006). Given the different scenarios on what must be checked for quality as described in 4.4.2, a scenario-based checklist is provided for reviewers, with procedures on how to check the different attributes, also specifying elements of a defect-based approach. Concluding, a quality form is made based on the SRS quality control checklists from appendices E & G, using the aforementioned approaches.

After the document reviewers are chosen and the document is complete, the quality can be checked and feedback is documented. In case there are changes necessary, feedback is sent to the requirements engineer for rework. After the quality is deemed sufficient, there may be another review by the client is that is required by the client. When the document is finished, the document is signed and sent to the client.

4.5. CHANGE MANAGEMENT

4.5.1. INTRODUCTION AND CURRENT SITUATION AT MOVARES

Recall from section 4.1 that change management is one of the best practices in requirements engineering according to Pandey et al. (2010). In this following section, we first explain the differences and overlap between configuration- and change management, the current way of working at Movares, followed by a theoretical model on the subject and the appliance to Movares requirements engineering process.

Change management covers how changes will be handled during the process. Requirements specifications are corrected or updated when necessary, by means of either a change or a revision (Grady, 2006). There are different sorts of changes. A revision may be whole parts of the specification being changed, or it could just be a single requirement being changed. Change management shares some aspects with configuration management, but they are two different subjects. Configuration management is concerned with the interfaces among different subsystems, whereas change management is concerned with changes within a system in general.

Currently, Movares regularly applies configuration management, but change management is often not explicitly defined in projects. Within systems engineering at Movares, changes are managed through baselines, which is part of configuration management. Every system has a certain configuration. A lot of projects within the GWW-sector are in fact changes to the current configuration of a certain system. The configuration is the set of objects of which a system consists, including relevant documents such as design decisions and requirements specifications. As mentioned, part of configuration management is managing baselines. A baseline is the outline of the configuration at a specific moment, so the complete set of documentation of a system on that moment. The goal of using baselines is to have certain moments in time to align the involved disciplines and documents and make grouped decisions based on all of the information available at that specific moment (Leidraad, 2013). The reason for this is that within projects in general, different disciplines are working on their own project within a project. For example, when engineering a railroad track, one discipline may be engineering the overhead lines, whereas another discipline is engineering the railroad switches. However, decisions made or changes to the railroad switches, may influence the designs of the overhead wires. This example comprises just two disciplines. It is not uncommon to have ten or more engineering disciplines involved in a project. Therefore, a requirements engineer cannot impose new requirements, decisions and constraints to engineering disciplines at any given moment. The requirements engineer and systems engineer must first find a suitable baseline and find interfaces between the different disciplines in order to align all disciplines and make sure there are no conflicts between both internal and external interfaces. Configuration management can be complex and is the expertise of a systems engineer. For this reason, it is often performed by a systems engineer.

4.5.2. THEORETICAL MODEL

The control of baselines is a principal activity of configuration management. Once certain engineering designs are finished, standard configuration management procedures are effective in controlling the design baseline. All changes must pass through the same process of configuration management. To be able to track decisions and changes made earlier, all decisions and changes should be traceable to certain baselines (Grady, 2006).

Configuration management and change management are best visualized through the model of Kramer & Magee (1990), which is presented in figure 4.6. First, change specifications are gathered such as new requirements or decisions. During configuration management, interfaces are sought between the different subsystems and a baseline is set. Then, there is a change transaction between the different subsystems or engineering disciplines that engineer those different subsystems.

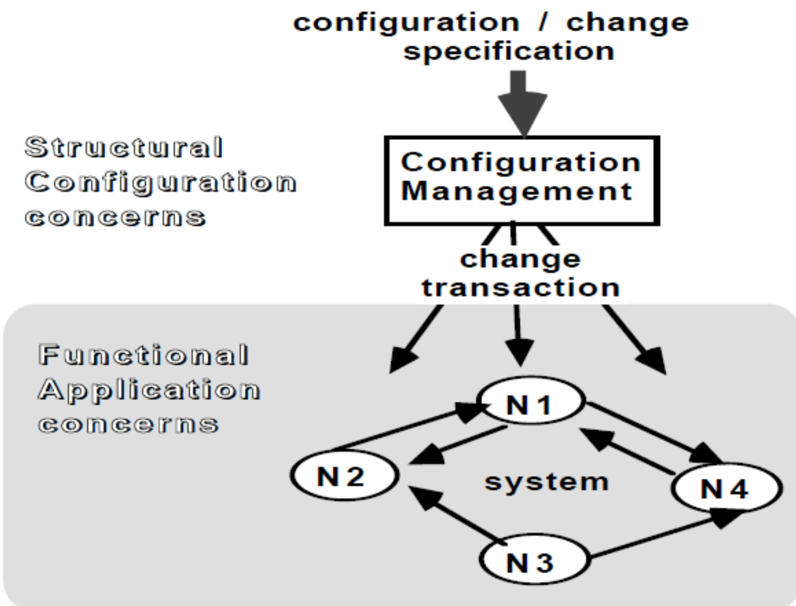


Figure 4.6 - Configuration management (Kramer & Magee, 1990)

This process is repeated multiple times throughout a project and therefore has an iterative character. In each iteration, the configuration changes incrementally, as visualized in figure 4.7. It could for example be that during the project, the scope or system boundaries are changed. This is possible by using this model.

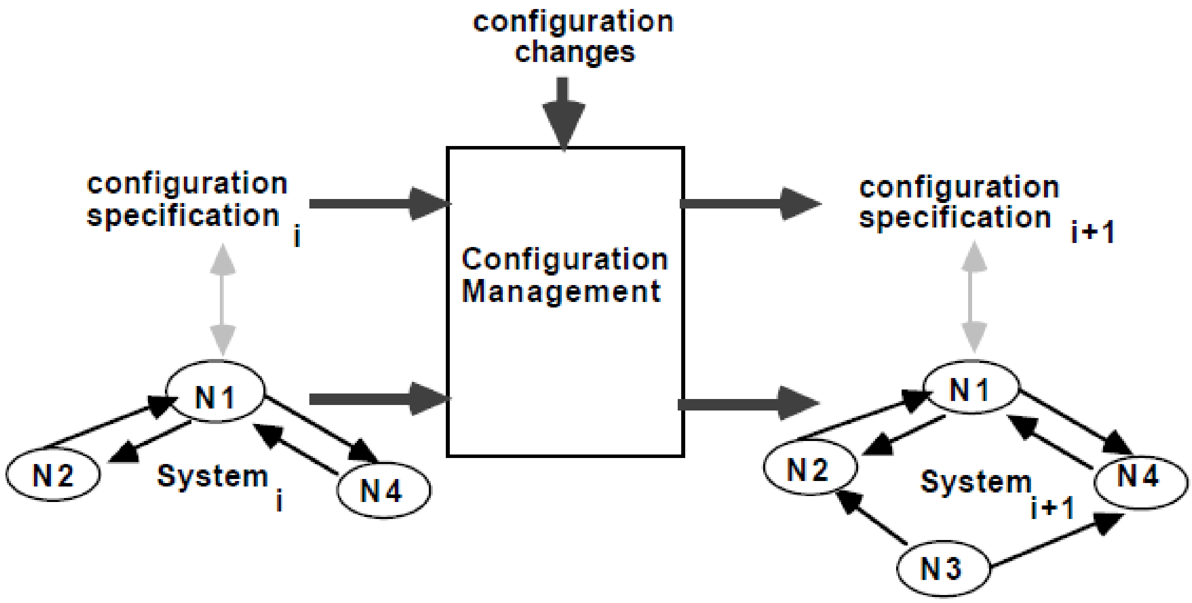


Figure 4.7 - Incremental baselines (Kramer & Magee, 1990)

Changes should be specified in terms of the system structure, in order to find affected interfaces easily. This means that changes should refer to the concerning modular subsystems. Identify which subsystems are affected by the concerning change. A minimal number of subsystems should be modified because of a change. Change specifications must also be declarative. By this, it is meant that within configuration management, the specific ordering of change operations is determined that applies to the system. One should keep in mind that during configuration management, the system is left in a consistent state, with minimal disruption to the functioning of the system (Kramer & Magee, 1990).

Inherent to requirements changes, some requirements end up being rejected. As it is one of the previously defined SRS best practices from appendix F, it is important to record rejected requirements in a repository of discarded requirements, since new requirements can be those that were previously discarded (Sommerville & Ransom, 2005). This will save time and money (Cox et al., 2009).

4.5.3. APPLICATION OF THEORY TO MOVARES

The approach of how to deal with changes in requirements and system configurations should be documented at the start of a project in a change management plan. This allows for faster and adequate response to changes. A systems engineer should always be involved in making a change management plan. This concept is applicable to the Movares way of working. Additionally, new requirements should be gathered until the next baseline. Up to the baseline, the requirements should be analyzed for finding interfaces and synchronizing the different actors and disciplines involved.

4.6. DATA MANAGEMENT

In the previous parts of this chapter, we have discussed aspects of the RE process that are defined in literature as best practices. Data management is not explicitly defined as an RE best practice. However, we find that in the modern-day era, data management can be applied to almost any process and facilitates a lot of possibilities such as data driven performance measurement. In the previous chapter, in subsection 3.3, key performance indicators are determined for measuring a process' performance. In order to measure and operationalize these indicators, data needs to be gathered and managed. Currently, there is no data mined, stored or managed related to the RE process at Movares. In the following parts of this thesis is elaborated how Movares should pursue data management. In 4.6.1, the scope of data management is discussed and in subsection 4.6.2, data quality is reviewed.

4.6.1. DATA MANAGEMENT SCOPE

The specific data to be managed which is discussed here, is directly related to the scope of this research, meaning data related to the process of making SRSs and contract specifications. This is necessary in order to analyze the requirements engineering process in general, and the system requirements engineering process in specific. Figure 4.8 is used to visualize the data needed in order analyze the system requirements process. At the moment, there is no data. It is therefore unknown how many documents are made from any of the documents mentioned in figure 4.8. In this research, we have started with system requirements documents, implicating SRSs and the system requirements part of contract specifications. This is extended to the other documents of contract specifications. After implementation of a database model, a logical next step would be to incorporate data from CRSs, since a CRS is input for either an SRS or contract specification, and also contains requirements. This way the process of engineering requirements in general is analyzed, and not just system requirements.

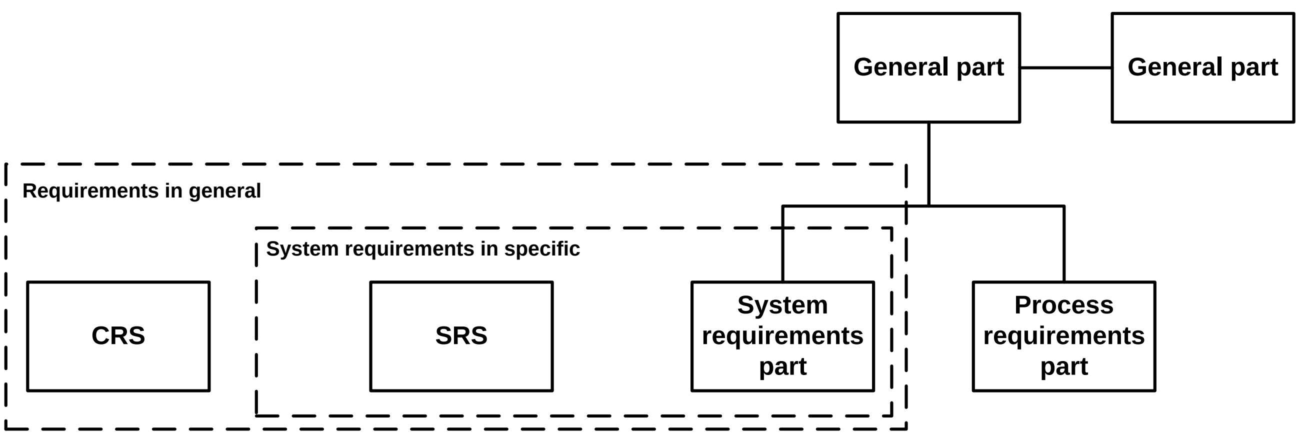


Figure 4.8 - Data management scope in this research

There is a reasoning behind the fact that data management for contract specifications is split into multiple parts, as the combined annexes are also a separate part. As explained before, contract specifications are split into multiple documents. These different documents are often made by different parties, but not always. For this reason, one contract specification cannot be easily compared to another specification. Next to this, contract documents other than the system requirements document, have nothing to do with system requirements. If one wants to know detailed information explicitly about system requirements and the process of making a system requirements document, it is best to split the data analysis of contract specifications into the different documents that make up a contract specification. In the future, this model from figure 4.8 could be further expanded to other products and processes.

4.6.2. DATA QUALITY

The parameters and KPIs are quantified, but there still is an important remark to be made here. The data is namely not 'hard' data. Different people interpret the parameters differently, and there is still room for some subjective interpretation. An example of this are the hours booked on certain activities in Hours, these numbers are rounded to whole hour and not 100% accurate. Another issue is the fact that there are no standardized booking elements throughout projects. The concept of booking elements is explained in subsection 3.3.1, but the issue is not. At the moment, booking elements are different for different projects, and are not decided on from a centralized, specified standard. This leads to confusion among employees on which booking elements they need to use each time. This will cost a lot of time and pollutes the data. Part of the variation in results between projects could be explained by this form of data pollution. It is therefore necessary that this is changed in the near future, but the issue is out of scope for this thesis to research much further. This should be changed by applying standardized booking elements.

4.7. PREDICTION MODEL

Related to data management, is prediction modeling, since a proper prediction model is primarily based on data instead of experience. Having a good prediction model can help a lot with finding influential factors on for example the duration of an SRS and expected rework. For this reason, the possibilities of developing a prediction model are researched in this section of the thesis. The goal of this prediction model is to 1) find the most influential factors on quality and cost, 2) being able to accurately predict the objective values and 3) being able to identify risks beforehand. In 4.7.1, we reviewed possible input variables, in subsection 4.7.2, the minimal sample size on the amount of data is examined, followed by a discussion on data quality in 4.7.3.

4.7.1. PREDICTION MODEL VARIABLES

A prediction model comprises of input and output variables. Input variables are independent, the output variables are dependent on the input variables. The dependent output variables that we need to predict are the KPIs which are already defined in section 3.3. In the following subsection, we have searched for the independent input variables of prediction model.

In requirements engineering, the most influential factor is complexity. Complexity can be evaluated based on costs, based on implicitly and based on entropy and information theory (Brosch, 2012). Cost-based evaluation methods evaluate the effect of complexity on costs. Implicit evaluation methods evaluate the complexity and diversity based on the number of elements. For Movares this can be interpreted as for example the number of interfaces and number of stakeholders and disciplines involved. Entropy-based and in information theory used evaluation methods evaluate the density of information and uncertainty of the system. This is interpreted as complexity influencing uncertainty. The expertise from table 2.1 is combined with literature on factors influencing the RE process and the earlier determined KPIs, to come up with a list of possible influential parameters. An important remark is that these parameters are already operationalized and kept as simple as possible. These parameters are:

- Complexity, which is defined by a combination of the following attributes:
    - Number of interfaces.
    - Number of disciplines involved.
    - Number of stakeholders involved.
    - Duration of the project.
    - Political complexity: This parameter is a bit more complex than the previously mentioned parameters. For example, if the project is dealing with a large and complex project team, or the project team has a client or stakeholder with whom every decision has to pass 3 different people, the political complexity is high. Political complexity also depends on whether or not there are stakeholders with conflicting interests. Political complexity can be divided into 1) simple – 2) average – 3) complex/special.
    - Internal complexity: The level of clear communication and teamwork within the project team. Divided into 1) bad – 2) average – 3) good.
- Quality of input: A requirements engineer receives different emails, reports and analyses of different actors within the project. From these documents, requirements have to be derived. When the quality of these input-documents is high, it is easier to derive requirements, than it is when the quality is low. Given a lower quality, a requirements engineer will need to more time deriving requirements. For example, the requirements engineer may have to send feedback to the source of the document and ask and wait for better quality documents. Quality of input is divided into:
    1) Bad: Relatively much time is spent on waiting for input documents, a lot of reminders are necessary to acquire input. In relation to the reminders, a lot of coordination necessary. Additionally, the input itself could be bad as requirements are hidden over multiple documents and contextual subtleties.
    2) Average: The input is a for example a requirements document without any structure and badly formulated requirements.
    3) Good: A structured requirement document with well-formulated requirements.
- Intended quality of output: This is determined in the strategy meeting before making an SRS. The planned level of quality of the SRS document itself depends on the requests or demands made by the client and Movares' intentions on the quality that is actually delivering. If there is for example little time, it could be that this will be at the expense of the provability.
    1) Average: Basic SRS with only the essential elements.
    2) Good: High quality and detailed SRS. Full traceability and provability.
- Level of specification: The detail to which a system is specified, which also depends on the client's demand. This is usually done during a strategy meeting and by using a reference document. This level input is important, but hard to quantify. That is why it is decided to maintain three levels:
    1) Open specification: Specify requirements to a general level which leaves a lot of design choices still open.
    2) Average specification: Specify requirements to an average level.
    3) Specified in detail: Specify the requirements in detail.
- Client: The amount of time spent may differ per client or field of activity. Clients are specified to:
    1) ProRail.
    2) Rijkswaterstaat.
    3) Municipalities.
    4) Polder boards (Waterschappen).
    5) Carrier organizations.
    6) Provinces.
    7) Remaining clients.

- Total budget of the project.
- Experience requirements engineer. This divided into three levels:
  1) Junior employee.
  2) Medior employee.
  3) Senior employee.

From these parameters or input variables, a correlation model can be made, in order to filter out the best predicting parameters. Additionally, weights can be applied to the best predicting parameter in relation to lesser input variables. The weights of the input variables may differ per output variable and is hard to predict based on experience. In order to build a prediction model, data is needed. The minimal sample size of such data is calculated in the next subsection.

4.7.2. MINIMUM SAMPLE SIZE

In order to perform a statistical analysis, a minimal sample size is needed. This minimal sample size is needed to be able to make reliable predictions on the output variables. Now that both the KPIs and the parameters are determined, data can be gathered for the prediction model. An important question is how much data should be gathered.

According to Sedlmeier & Gigerenzer (1997), there is never enough data for a prediction model. The data exists of independent and identically distributed random variables. More data means the prediction model will be more accurate. This is due to the law of large numbers. It is therefore tried to generate as much data as possible for the prediction model and KPI dashboard. There is however a minimum sample size. This minimal sample size (MSS) depends on the required precision and the standard deviation. Since the data is continuous and the sample size is too small to calculate normally, the MSS is calculated as follows (Brook, 2014):

$$MSS = \left(\frac{(2 \times Standard\ Deviation)}{Precision}\right)^2$$

The standard deviation is estimated by looking at the historical range (the difference between the highest and lowest values that are observed) and to divide it by five (Brook, 2014).

This calculation is performed for each of the KPIs, after the data is gathered of ten SRSs, except for KPIs 7 and 10, since there is no historical data available for these KPIs. For the other KPIs, table 4.1 presents the results of the calculations. After time, when data gathering is embedded in the new standard process, more data is generated and the prediction model will become more precise. For now, it is preferred to generate at least 46 records.

Table 4.1 - Calculation of Minimum Sample Size (MSS) for SRSs

	KPI 1	KPI 2	KPI 3	KPI 4	KPI 5	KPI 6	KPI 8	KPI 9
Precision	30	30	45	4	30	10	6	3
Standard deviation (estimate)	100.00	86.80	152.00	7.60	100.00	27.00	11.6	6.00
MSS	46	34	46	16	46	30	16	16

Current amount of records	17	12	21	17	18	18	8	8
Current precision	48.51	50.11	66.34	3.69	47.14	12.73	8.20	4.24
Precision met?	no	no	no	yes	no	no	no	no
Amount of records still needed	29	22	25	0	28	12	8	8

From the minimal sample size, it can be concluded that there is not yet enough data gathered in order to make a reliable prediction model. When enough data is gathered, a prediction model should be made. A suitable application for making a prediction model is Weka. It is free and easy to use, which lowers the threshold for creating and using a prediction model. Weka could be applied in the future when the minimal sample size is reached.

4.7.3. DATA QUALITY

A remark concerning data quality of the parameters, is that different employees may perceive certain aspects of a project differently. For example, one person may perceive the internal communication as average, whereas someone else may perceive it as bad. This implicates that there could be precision errors in the data, creating variation in the data (Brook, 2014). However, as explained the previous subsection, there is not enough historical data to perform a 'Gauge Repeatability and Reproducibility' study (GR&R) to assess the precision error. A GR&R study should be performed when there is enough data gathered on the prediction parameters.

To conclude this section on a prediction model, it cannot yet be developed. In order to so, first enough data needs to be gathered on both input and output variables. Both types of variables are already defined. When enough data is developed, a GR&R study can be performed, and if the quality of the data is deemed sufficient, a prediction model can be built using the free application Weka.

4.8. VISUAL MANAGEMENT

The final aspect which is explicitly analyzed in this chapter, is visualization. Visualization is an important tool for any process and refers to the use of graphical methods to display and communicate how a workplace or process is managed (Brook, 2014), it is one of the key principles of lean management. Visualization is a way to help people gain insight from large and complex data sets (Katasonov & Sakkinen, 2006).

In the following subsections, we elaborate on what visualization techniques can be applied to Movares. The SRS process can be visualized as an activity model, which is discussed in subsection 4.8.1. The core of an SRS, the requirements, are also discussed with regards to how they should be presented and visualized in 4.8.2. Lastly, in 4.8.3, this subsection examines the use of visualizations in the KPI dashboard.

4.8.1. SRS PROCESS

One of the goals of this research is to be able to let the people from whom input is provided, derive their own requirements, so that requirements documents can be made faster, and so that the requirements engineer spends less time on deriving requirements. This relates to problem numbers 3 & 4 which are identified in subsection 1.2. SI would like to move towards a more advising role, and thus doing less 'technical' or engineering work. In the case of SRSs, technical work refers to deriving technical system requirements, which should be done by the engineers providing input, who have more technical knowledge. At the moment, these people providing input often do not know the purpose of an SRS, how to make an SRS or how to derive requirements.

In order to explain how the SRS process works, visualization can help. The current way of working is visually presented appendix B. However, this large and complex visualization can be confusing to someone who is not familiar with SRSs. The process should be explained as simple and accessible as possible. Therefore, it is best to explain the main processes of an SRS, as is shown in figure 2.6, and cut the process into the main sub processes. These sub processes are then presented in the same way on a lower level, as is shown in figure 4.5. This way, the whole process could be modeled in a visualized manual. This makes it easier to explain the SRS process in detail and in an accessible way.

Movares is a complex and diverse working environment. Employees usually work on multiple different ongoing projects at the same time and each project requires a unique approach. In business process management notation literature, processes are often visualized in so-called 'swim lanes', in which each swim lane is represented by a different actor or group of actors. Due to the complexity of processes at Movares, in which processes often differ among projects and activities among processes are undertaken by varying compositions of actors, we have chosen to show processes using the Movares way. This Movares way of presenting processes is more activity oriented instead of actor oriented. Additionally, partitioning processes into lanes is not required in business process management notation (Wohed et al., 2006). It is however wise to visually indicate the actor that is responsible for each activity. This applied in visualizing Movares' processes in appendices B and C.

4.8.2. REQUIREMENTS

Requirements should be presented in a form that can be better understood by the people it is intended for (Katasonov & Sakkinen, 2006). Within an SRS, requirements are visualized in a table structure as in appendix A. In order to provide a logical structure, requirements are primarily presented following the defined order from subsection 2.2.2:

- Functional requirements
- Aspect requirements
- Realization requirements
- Interface requirements

Functional requirements are then presented according to the earlier specified functional breakdown structure (FBS). Functional requirements should always be linked to the FBS. Aspect requirements should be presented according to the relation to either reliability (R), availability (A), maintenance (M) or security (S). Interface requirements should be presented according to the different identified interfaces, both internal and external.

An important remark is that requirements can be visualized using a figure, as can be observed in appendix A. However, one should be careful to not provide solutions in a figure, since requirements should be solution-free.

An additional remark for the future is concerned with Building Information Modeling (BIM). BIM is software in which information on all aspects of a project including the surroundings are modeled in a single application. This way, interfaces are inspected easily. A BIM can be visualized in 3D, so that the user can visually select, analyze and edit that part of the model that he or she is interested in. For example, when a certain adjustment for a requirement is suggested, the feasibility and relation to other aspects of the model can be analyzed immediately. However, at the department of SI, there is still not much being done with this new technique, and the client still needs SRSs in the form of a paper document. Additionally, other people in the organization are already busy developing this concept, so the possibilities of BIM are not further investigated in this research.

4.8.3. KPI DASHBOARD

A KPI dashboard is another opportunity for applying visual management. Just collecting information on business processes from section 3.3 is not enough. One of the objectives of a dashboard is to visualize the data and provide insight on the process. Managers need specific information in a single glance and easily accessible. For this reason, the most important KPIs should be highlighted more prominently in relation to other KPIs. Next to this, it should be possible to visualize KPIs per period of time, in order to visualize trends. Lastly, the dashboard needs to be easily accessible and transparent for the managers who end up using it (Few, 2006).

The best choice of a specific visualization option for a specific KPI or aspect of the dashboard differs per KPI. For example, for KPI 10, the percentage of actual hours compared to the budgeted hours, it would be best if this is shown as a color-coded bar or meter, so that a manager could see in a single glance where the process stands on the amount of slack in spendable hours. Other KPIs, such as the number of requirements that end up being used in the requirements document, are more difficult to put in relation to other projects using visualizations, but are useful and provide information to the manager nonetheless.

There are different types of managers to whom a database on the requirements engineering process is useful. So far, both line managers and project managers have indicated they would use a KPI dashboard, if it is available. Line managers are managers of a specific department or discipline, such as System Integration. Per type of manager or user, a different dashboard is required since different types of users value different aspects of the process or project.

4.9. CONCLUSION

In chapter 4, a research is performed to find ways in which the current way of working at Movares can be improved, answering sub research question 3. This is done by finding the best way of working of the most important aspects of the RE process according to literature and applying these concepts to Movares.

There are several ways in which Movares can improve its requirements engineering process. Currently, there is little preparation before starting the elicitation process. Beforehand, usually the content and depth of the specification is defined, but Movares could improve by incorporating a more elaborate requirements management plan, including a stakeholder management plan. The requirements management plan should also include a change management plan, which includes the approach of how to deal with changes in requirements and system configurations. The process of eliciting requirements itself should also be standardized and would be more efficient when structured clearly. This implies differentiating into different types of system requirements and prioritizing them. Additionally, Movares should apply a differentiation between 'must have' and 'should have' requirements. Furthermore, the quality of requirements documents could improve a lot by standardizing the quality control process on a more detailed, product-specific level. This way, a higher, more uniform product quality is achieved.

Concerning data management, data should be gathered to facilitate KPIs, initially on SRSs and contract specifications. This data can be gathered automatically, but to improve data quality, Movares should try to standardize the booking elements. Expanding data management to facilitate in developing a prediction model, both input and output variables are defined. However, Movares should first gather data on the variables and analyze them before the prediction model can be further developed. Finally, we have searched to apply visual management to the RE process, which is a key principle of lean management. In the following chapter, we continue by applying the results from chapter 4 to specify a new way of working.

5. CHAPTER 5 – APPLIANCE TO A NEW WAY OF WORKING FOR MOVARES

A new phase of the managerial problem-solving method is entered, the 'decision' phase (Heerkens & van Winden, 2012). We now reach the heart of this research. In the previous chapter we found ways of improving the current way of working. In this chapter, the best practices from the current way of working, from literature on RE and from interviews are combined in a new way of working regarding the requirements engineering process. This chapter therefore provides answer to research question 4, on how a new and standardized way of working can be realized at Movares.

In the previous chapter, we found several ways in which the current way of working and making SRSs can improve. In this chapter, these important aspects are standardized and documented in a new way of making SRSs. First, standardization is elaborated in 5.1. In the following part, the new way of working is specified. The new way of working is documented in the SRS manual. The manual is further completed with additional improvements. The manual itself can be found in appendix I. This manual and new way of working is decomposed in seven main activities. The most relevant aspects of the new way of working are discussed throughout the different subsections of part 5.2. Section 5.3 continuous by evaluating the new way of working based on the level of standardization of best practices. In 5.4, the different ways in which the RE process can further improve in the future are pointed out. Then, the application of a data model is discussed in 5.5. Finally, the chapter is concluded in part 5.6.

5.1. STANDARDIZATION

In this first section, we examine the possibilities of increasing the use and standardization of best practices in the RE process at Movares. As explained Movares is trying to incorporate more lean strategies. This is an interesting development and certainly applicable in this research. A few lean strategies have already been discussed. Sacks et al. (2010) made a list of relevant lean principles for the construction industry, and thus are applicable to Movares. These lean construction principles are: reduce cycle times, reduce variability, standardize, apply continuous improvement, visual management and verify & validate. These principles are not mutually exclusive, but actually supplement each other. For example, standardization leads to reduced cycle times and variability. The necessity for reduced cycle times and variability also followed from the data analysis in section 3.3. In this section, standardization is discussed. Visual management is already deliberated in section 4.8, verification and validation is discussed in section 2.2 and continuous improvement in 6.4.

Process standardization is defined as the degree to which work rules, policies and operating procedures are formalized and followed (Beimborn et al., 2009). Standardization is an important goal of this report, and for good reasons. It positively influences process control, efficiency, time and quality. In chapter 3, the best practice guidelines from appendix F are already elaborated and RE activities from these guidelines are valued on the extent to which they standardized. Standardization in project management leads to shorter cycle times, improves quality and reduces variation such as variation in costs (Inman & Milosevic, 1999). Standardizing the process is also a key action towards reaching process maturity (Harmon, 2007).

Earlier in section 3.2, we evaluated the old way of working for the RE process on the standardization of RE best practices. Over the course of chapter 4, the possibilities of applying several important RE standards are examined. These best practices need to be standardized by documenting the different activities and order of activities. Documenting is the most important part of standardization. This is done by documenting all RE best practices in a manual for making SRSs. The new way of working and manual are further elaborated in the following section, 5.2.

5.2. THE NEW SRS PROCESS

In this section of the thesis, we discuss the new way of working and the implementation of this new way of working in a manual. The manual is one of the most substantial results of this research and can be found in appendix I. In this section, this manual is summarized by first discussing the process in general, after which the separate main activities are treated in the following subsections.

The manual is constructed in order to document the new process for making SRSs. Additionally, it is an opportunity to document and apply the new RE standards from literature. The manual will also allow requirements engineers such as those from SI to employ a more facilitating and delegating role during the RE process. A structured way of working by following the manual will lead to a higher quality product, less rework and in general a shorter cycle time.

As described earlier in chapter 4, it is not realistic to apply certain quantitative planning techniques such as CPM and PERT for the specific activities within an SRS. However, one planning technique that is applied, is process mapping (Brook, 2014). There are a lot of different activities that lead up to an SRS. To keep the process clear and easy to explain, the SRS is primarily presented as in figure 5.1, which shows the relations of the main activities of the SRS. Next, the separate main activities are broken down into multiple sub activities. Appendix C provides the complete overview of the newly defined process including all sub activities.

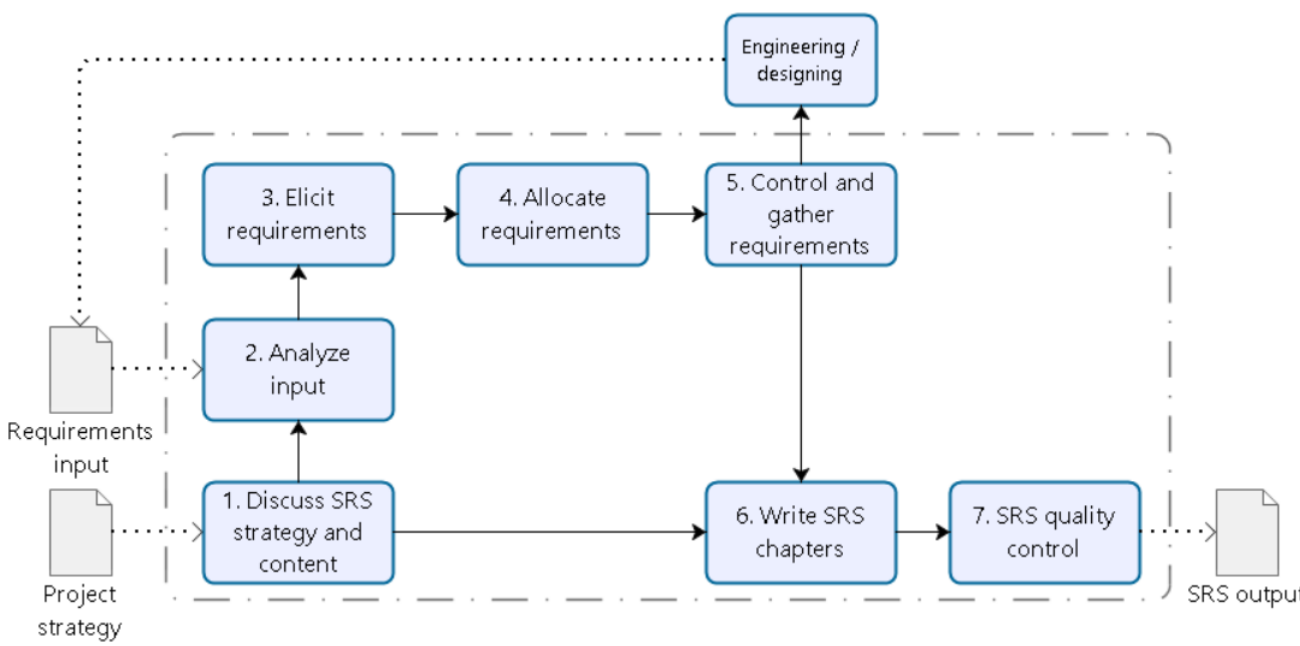


Figure 5.1 – Main activities in the new SRS process

Figure 5.1 presents the main activities of the SRS process. In the following subsections, these main activities are treated individually. In the manual, these main activities are also treated separately and are even further decomposed into sub activities. For each sub activity in the manual, the 'who-what-why-how' answers are provided. Since it then becomes quite a long document, one can easily navigate through the manual by using the hyperlinks in the .pdf file.

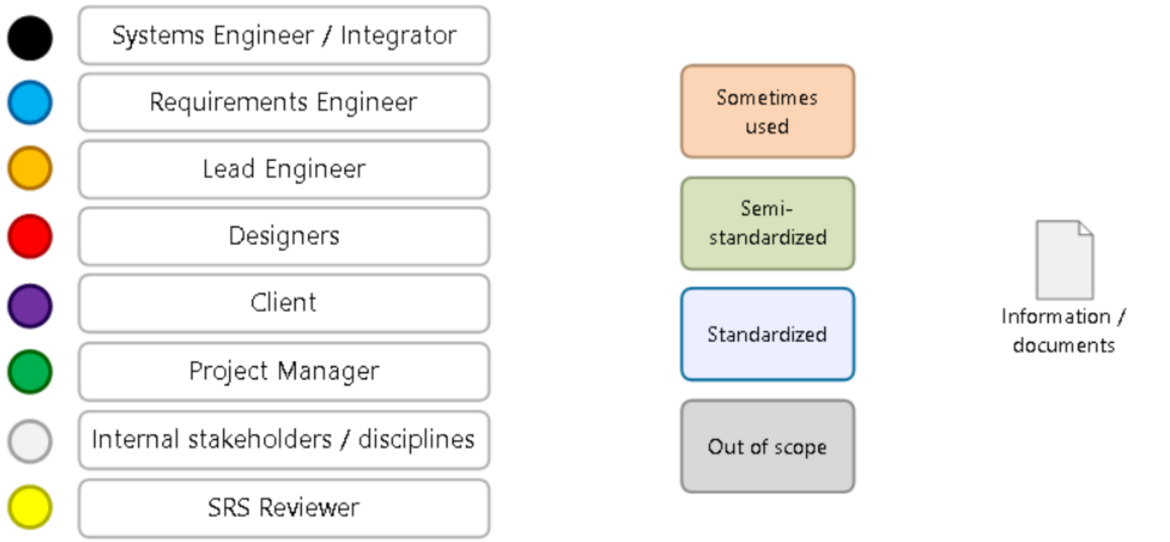


Figure 5.2 - Legend for the new process description for the figures in the following subsections

For the decomposition of the main activities, the legend is provided in figure 5.2. For each individual main activity, the decomposition is presented visually. Each activity is colorized, which is already explained in more detail in section 2.3. To summarize, the meaning of each color that is assigned of an activity is deducted from Sommerville & Ransom (2005) and corresponds with:

- Not used: The requirements engineering activity is (almost) never used. In this case it is not visualized in the model.
- Sometimes used (orange): The requirements engineering activity is sometimes used, or it is used without using a standardized prescribed way of working.
- Semi-standardized (green): It is recorded what the activity is, when and by whom the RE activity should be executed. However, the way this is guideline is executed, is not prescribed on paper.
- Standardized (blue): The concerning requirements engineering guideline is always applied in the same, standardized way. The way this is guideline, is executed is prescribed on paper.

To help compare the old and new way of working, appendices B and C can be observed for differences.

5.2.1. DISCUSS SRS STRATEGY AND CONTENT

Discussing the strategy and content of the future SRS is the first thing to do when starting to make an SRS. This activity presents the largest change in the new process. It was already defined in the old way of working, as is shown in figure 2.6. However, concerning the content, a lot has changed. Before, the content and strategy were relatively briefly discussed and not documented. In the new way of working, as can be deducted from figure 5.3, most activities are now standardized. There are more activities than before, so there is more time and effort spent on strategy in the new way of working. In general, this will lead to higher quality documents and less time spent on other activities. Spending more effort on strategy, will make a requirements engineer more prepared for uncertainties.

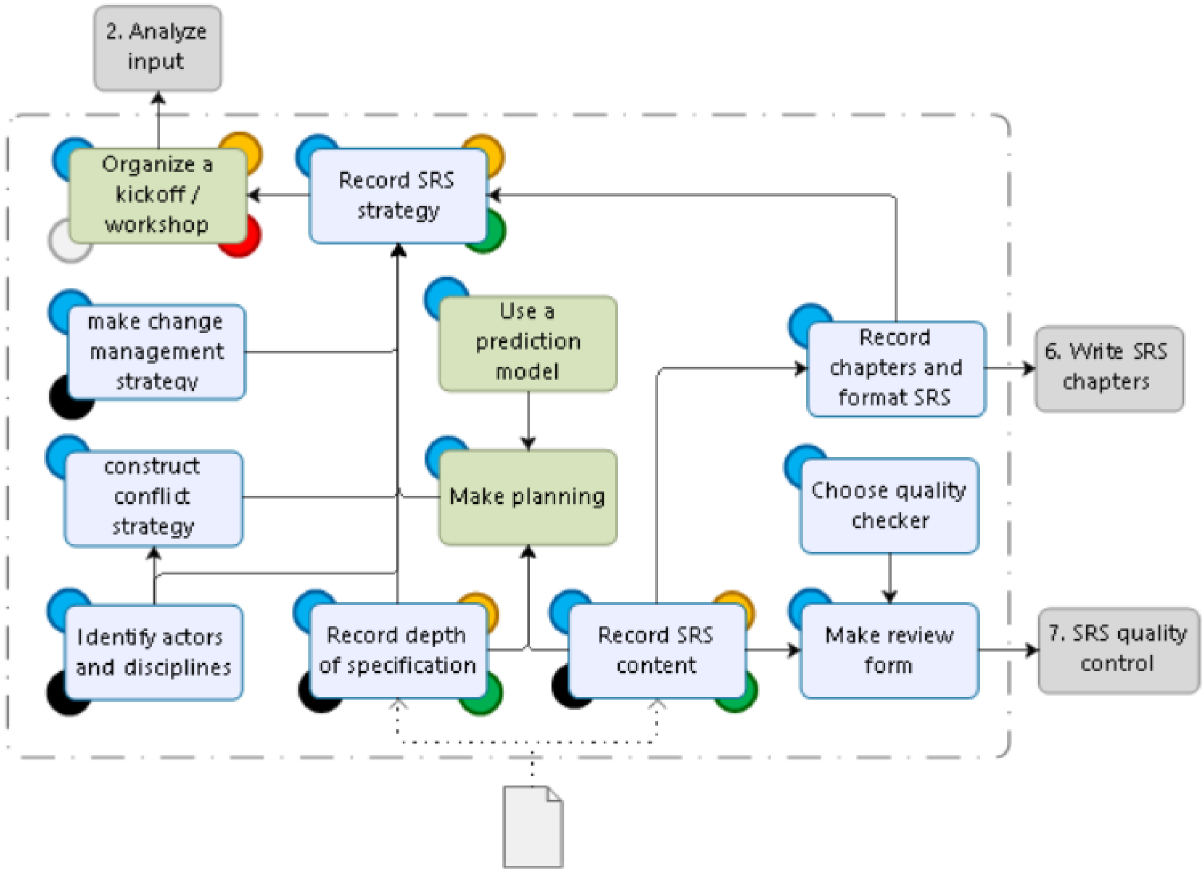


Figure 5.3 – The sub activities of main activity "Discuss SRS strategy and content"

The strategy and content of the SRS now need to be explicitly documented in a separate file. In this separate document, the following aspects are discussed:

- SRS content
    - SRS chapters
    - Requirements attributes
    - Activities to execute
- Depth of the specification
- Actors and disciplines involved
- Conflict strategy
- Change management strategy
- Planning

There are a several activities that were previously not executed. Making a stakeholder analysis, conflict strategy and change management strategy are all RE best practices that are now standardized for the new way of working. For most sub activities, it is specified how it should be accomplished. The amount of effort spent on making the strategy depends on the complexity of the project. The strategy should be more elaborate for more complex projects. For example, for large and complex projects, there should be a kick-off with all stakeholders of the RE process, including a RE workshop for everyone involved, in order to have all stakeholders and actors aligned.

5.2.2. ANALYZE INPUT

Analyzing input is the first thing to do after a strategy is constructed. Comparing the old and new way of working, there are no additional sub activities for analyzing input. The main difference is that in the new way of working, this activity is documented and it is described how input should be received and analyzed. The process is visualized in figure 5.4.

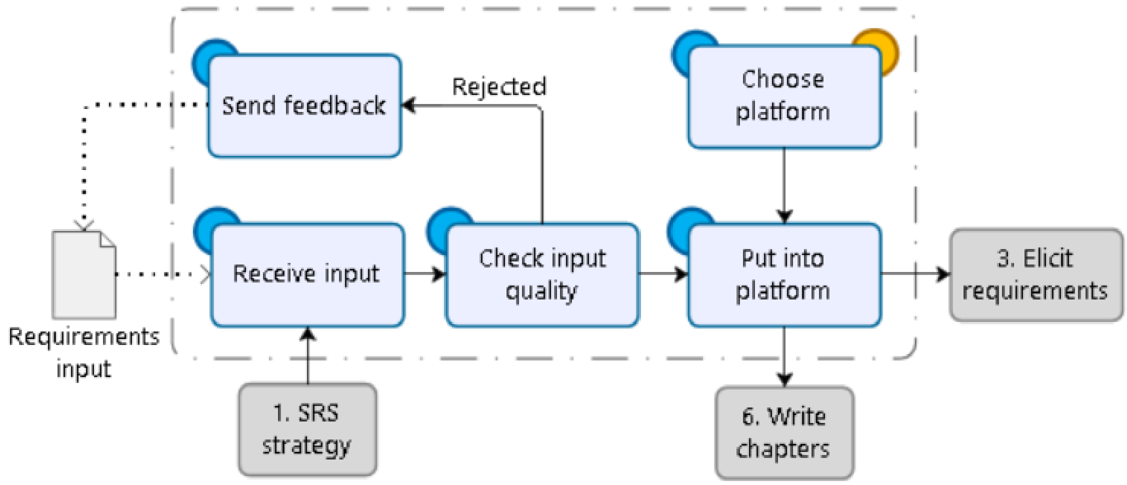


Figure 5.4 - The sub activities of main activity "Analyze input"

Input for SRSs can be different things such as requirements, visualizations of the project, descriptions, etcetera. The largest part of input comprises of requirements or input from which requirements have to be derived. As explained in section 4.2, system requirements have different origins. In an effort to construct a complete SRS in which all possible sorts of input are considered, all potential sources of input are listed in the manual for this activity. The requirements engineer then receives input from different actors that are also specified in the strategy. After input is received, the input is evaluated on the effort required to elicit system requirements from the input. Input is therefore evaluated on clarity, simplicity, traceability, function and priority. When deemed insufficient, feedback is sent to the actor providing the input. The input is stored in the appropriate platform such as Relatics, in which the requirements are managed.

5.2.3. ELICIT REQUIREMENTS

The elicitation of requirements is the activity of deriving or eliciting system requirements from input and recording the different attributes. This process is always done when making an SRS, also in the old way of working. The main difference however, is that in the new way it is described how requirements should be derived and additionally, requirements are prioritized. The new process of eliciting system requirements is presented in figure 5.5.

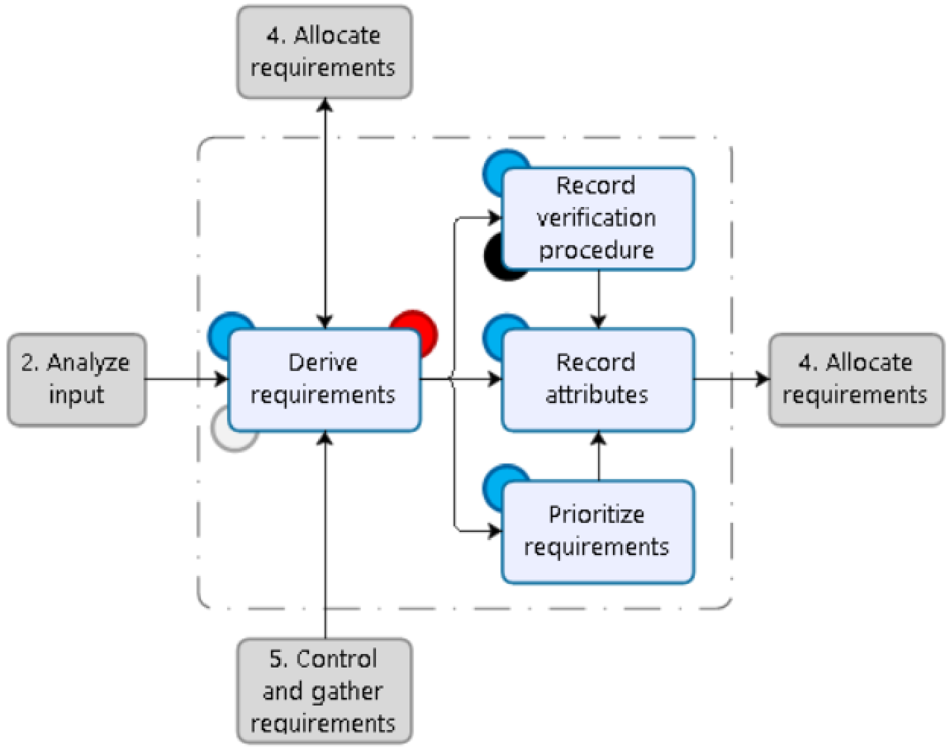


Figure 5.5 - The sub activities of main activity "Elicit requirements"

In the SRS strategy, the attributes for system requirements are already specified. These attributes are now filled out for each requirement. Two important attributes are the verification procedure and the prioritization. The verification procedure was already always important, but is now explicitly documented. Prioritizing the requirements is unique for the new way of working and based on subsection 4.3.

An important development compared to the old way of working, is for the requirements engineer to delegate more tasks within the requirements elicitation process. The actor that delivers input for the RE process, should already provide requirements he or she thinks are appropriate and necessary, including attributes such as the source. The main reason for letting the actors that provide input to elicit the requirements, is that these actors possess the most knowledge with regards to the content of their specialization. The role of the requirements engineer becomes more facilitating, structuring the process.

5.2.4. ALLOCATE REQUIREMENTS

During this activity, requirements are checked for quality and structured in different ways. The main difference of the new way of working in comparison to the old way of working is the quality check. The sub activities are presented in figure 5.6.

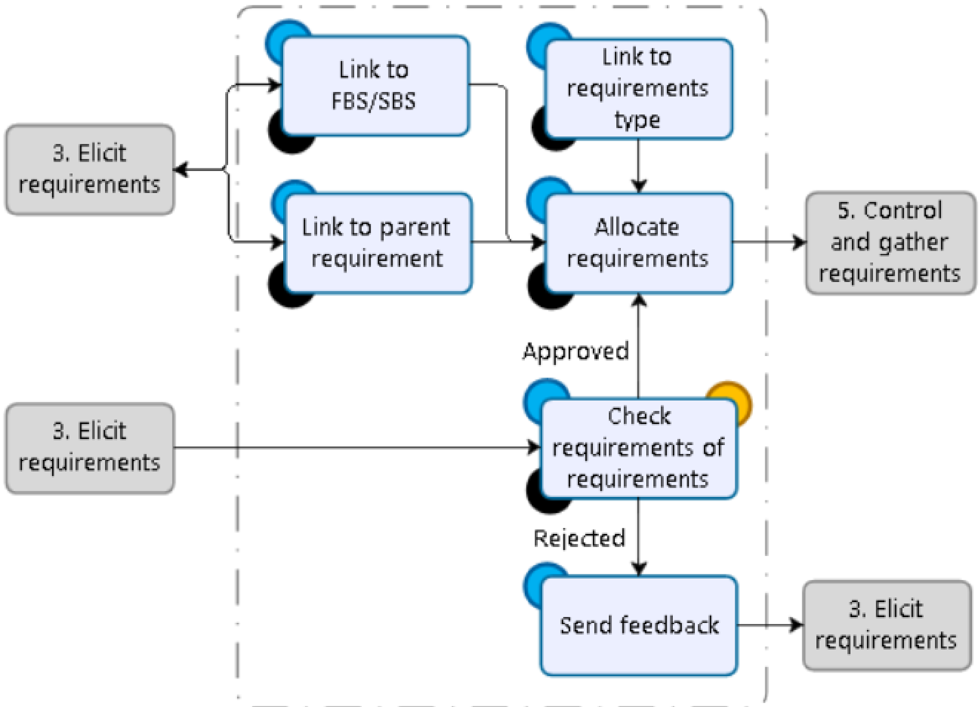


Figure 5.6 - The sub activities of main activity "Allocate requirements"

Allocating or structuring the requirements is done by the requirements engineer, systems engineer or both. First, the new requirements are checked by using the requirement-specific requirements from appendix E. Then, the requirements are structured by allocating it to a requirement type and linking each requirement to a 'parent' requirement and function in the FBS. Note that there are reciprocal arrows between "link to FBS/SBS", "Link to parent requirement" and "3. Elicit requirements". The arrow from left to right indicates that a newly made requirements should be structured according the FBS/SBS and a parent requirement. Following the arrow from right to left, parent requirements can be used as input for new requirements by further specifying them and new requirements can also be elicited by specifying a function from the FBS.

5.2.5. CONTROL AND GATHER REQUIREMENTS

Controlling and gathering requirements is a complicated activity consisting of multiple different sub activities. It comprises of checking requirements for usability and depth, gathering requirements and applying configuration management before the requirements are sent to the designers. This activity is changed significantly compared to the old way of working; an additional usability check is built in, feedback is explicitly requested from the client and configuration management is more prominently applied. The new way of working concerning controlling and gathering requirements is presented in figure 5.7.

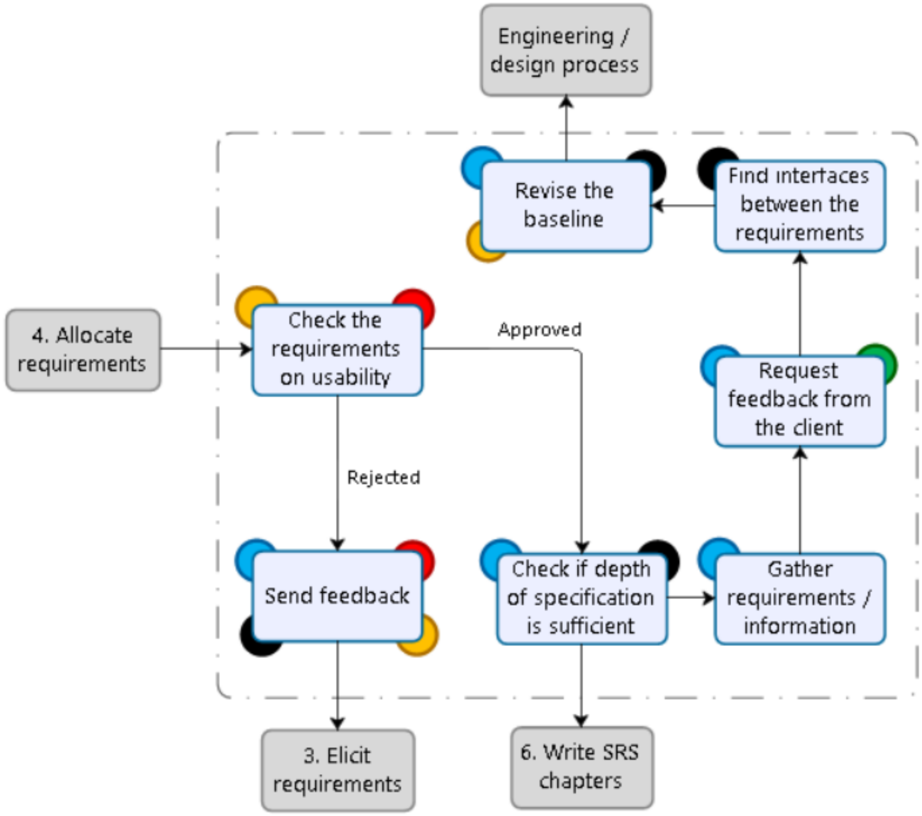


Figure 5.7 - The sub activities of main activity "Control and gather requirements"

Requirements cannot be immediately passed on to the engineers and designers who need the requirements as input for making the designs of the future system. For this, baselines are needed, as explained in section 4.5. But before the baselines are revisited, new requirements are checked for usability to check if the designers and engineers can work with the new requirements and whether they are practical and realistic. The effort of applying a new requirement may not weigh up to the benefits. Then, it is checked whether the requirements are specified sufficiently according to the level defined in the SRS strategy. If this is the case, the iteration process can be broken and no further system requirements need to be derived. The new requirements are however still used to make new designs, so the design process continues. The new requirements are gathered and feedback is explicitly requested from the client. This was not done before in the old way of working, but it is important to keep the client involved and involving the client is also defined as a RE best practice. Before the baseline is finally reached, interfaces need to be identified and actors and disciplines need to be aligned accordingly, as clarified in section 4.5. Managing interfaces and baselines are necessary to align the different disciplines and to make sure they do not work parallel to one another, creating later problems.

5.2.6. WRITE SRS CHAPTERS

Writing the SRS chapters can start after the strategy is constructed, but only ends after all of the requirements are elicited. This activity comprises of writing the content of the SRS document. Compared to the old way of working, the main difference is that this process is now documented and we created a standard Movares format for SRSs. The composition of sub activities of writing the SRS chapters is presented in figure 5.8.

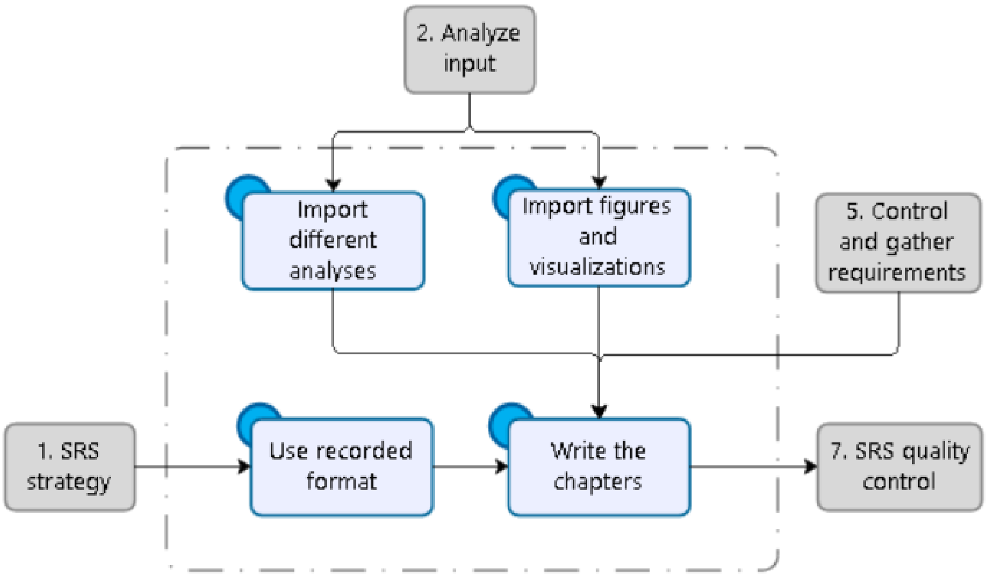


Figure 5.8 - The sub activities of main activity "Write SRS chapters"

In the strategy, the depth of the specification is set. When this depth is reached, the requirements do not need to be specified any further. The requirements then need to be put in the SRS itself. To write the document, we made a standard Movares SRS template. Before all system requirements are derived, the introductory chapters prior to the list of requirements can already be written. This can be done in parallel to the requirements elicitation process. Finally, when all chapters including the chapter containing the system requirements are written, the SRS is ready for reviewing.

5.2.7. SRS QUALITY CONTROL

In this thesis, a lot has been written about quality control. In this final main activity of the SRS process, the quality of the document is checked and signed off. Not much is changed compared to the old way of working, except that now a standard quality form is used. The process is visualized in figure 5.9.

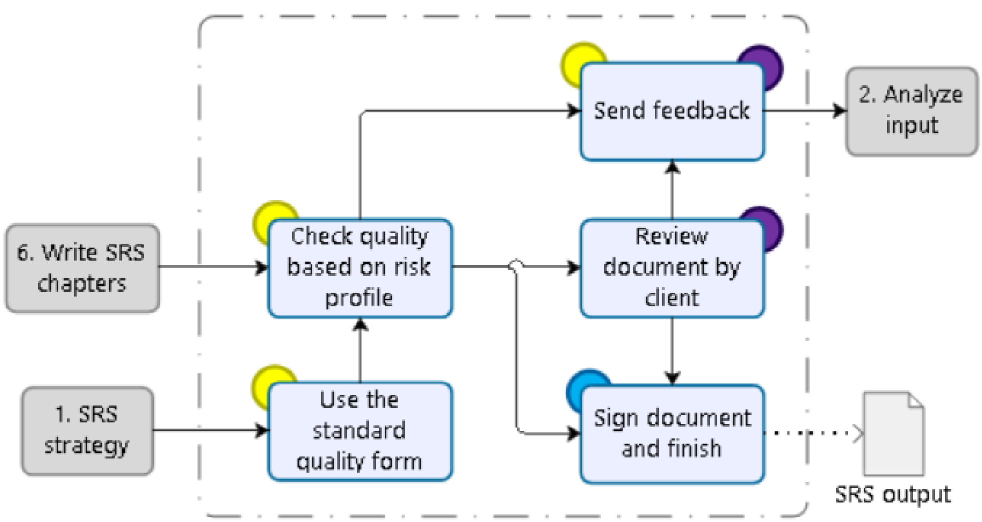


Figure 5.9 - The sub activities of main activity "SRS quality control"

After the SRS is finished, it is checked internally by using the checklist from appendix G, as is described in section 4.4. If the client wants to, a representative of the client reviews the document as well. Finally, when the document is deemed sufficient, it is signed and handed over to the client and used as input for the contract specification.

5.3. IMPROVEMENT OF THE NEW WAY OF WORKING

The new way of working for making SRSs is described in the previous section. Recall that the old way of working was evaluated based on the number of RE standards applied in subsection 3.2.2. When the new way of working regarding SRS documents is applied, the process significantly changes compared to the old way of working. Therefore, in this section, the new way of working is evaluated in the same way and the results are compared.

This change in making SRSs is quantifiable and is introduced in section 3.2.2. Recall that Sommerville & Ransom (2005) have developed a model of best practices used for requirements documents. The appliance of the best practices from appendix F are valued on a scale of 0, 2 or 3, based on the level of standardization of this best practice in the RE process. This adds for the old way of working to a total value of 87 points. The new way of working is valued at 118 points. The results are shown in table 5.1.

Table 5.1 - Improvement in usage of RE standards

Total points in current way of working	87	Total points in new way of working	118
Total amount of guidelines not used:	5	Total amount of guidelines not used:	2
Total amount of guidelines "normal":	22	Total amount of guidelines "normal":	2
Total amount of guidelines standardized:	13	Total amount of guidelines standardized:	35

Using the following formula, the improvement can be calculated:

$$Improvement = \frac{current\ assessment\ score - previous\ assessment\ score * 100}{previous\ assessment\ score}$$

This implicates an improvement of 35.6%. Especially the number of guidelines which are now standardized, has improved significantly from 13 to 35 guidelines. There are two best practice guidelines which are not yet used. The first guideline is that business concerns should be specified in requirements. After proposing this option, it became clear that only hard requirements should be mentioned in a system requirements specification. Soft requirements, or business concerns may be captured and put in the contract specification, for the contractor to take into account. However, prescribing contract specifications is out of scope for this research. The second guideline is to identify global requirements, meaning to use a database of recurring system requirements. This best practice is harder to implement than most other guidelines, since it will require a significant investment. This option is further discussed in the following section, section 5.4.

5.4. STANDARDIZATION OPTIONS

In this section of the thesis, we elaborate on the possible ways in which the RE process can further improve. For the new way of working, the SRS manual is set up in such a way, that as many elements as possible are standardized. For example, when choosing the attributes to use in an SRS, one can choose from a list of predefined attributes. By describing the process and putting the different steps and activities on paper, new possibilities for improvement and standardization arise to the surface. However, some of these possibilities of improving require a significant investment, that are not that easily implemented. These possible improvements are discussed in this section of the thesis. Such possibilities comprise of developing:

- A database of predefined possible recurring interfaces.
  - Different systems often have similar, recurring interfaces such as certain crossing structures, cables & pipes, hydraulic works, flora & fauna, etc. These could be standardized in a database in which for certain systems or objects interfaces are shown that have a high change of being applicable.
- A database of predefined possible recurring risks.
  - For a lot of projects, risks are identified. These risks are usually confined by further specifying a certain object or requirement in more detail. Recurring risks can be standardized, which saves time and helps identifying risks which would otherwise not have been taken into account.
- A database of predefined, recurring requirements.
  - From multiple interviews, it became clear that there is a need among employees for using standard system requirements. A lot of requirements are similar throughout projects, yet they need to be reinvented for each project. It is possible to make a database with recurring requirements, which would save a lot of time. These system requirements could also be linked to functions and objects.
- A database of probable recurring issues.
  - Enabling improved learning from mistakes, arising issues could be stored in a central database. For certain objects, interfaces or combinations of such aspects, recurring probable future issues could be generated, which could then be taken into account.
- A predefined standard FBS containing a database of possible functions.
  - Systems can be categorized according to certain recurring functions. For example, a bridge fulfills the function of getting people or cars across a waterway. These functions are further specified and structured using an FBS. A lot of time could be saved when these functions are partly predefined and requirements engineers have to spend less time formulating functions.
- Standard templates for specific documents.
  - Previously, For SRS documents, there was no standard template. Each SRS has recurring elements, but one SRS may look different from another SRS. Part of the new way of working is providing a standard template for SRSs. This concept could be extended to other types of documents for which no standard template exists.
- Standardized booking elements.
  - At the moment, the booking elements of a particular project are established before the start of that project. These booking elements are different for each project, which leads to ambiguity among employees on which elements to book activities on, and inaccuracy in data collection. These booking elements could be standardized so that for certain processes and products, the same booking elements are used throughout projects. This will save time and improve data quality.

The thing that these options have in common, is certain recurring elements, whether those are recurring requirements, interfaces, or other aspects. Each SRS has unique content, but most aspects of SRSs and the process of requirements engineering have recurring aspects which are not unique. Instead of having to rethink and reinvent these elements for each project again and again, one could make a database from which a requirements engineer can pick the elements that are applicable to that specific project. Of course, there will always be unique elements as well, but if for example 50% of the system requirements can be picked from a database, the database will save time for the requirements engineer. There are more advantages to storing recurring elements. Given the example of standard recurring system requirements, the output quality of SRSs will more consistent, since system requirements will be more the same throughout different documents. By using mainly system requirements from which is known and proven that the quality is good, the quality of the document and system requirements in general will increase. Next to this, it will be easier for designers to work with the same recurring requirements for different projects, which are formulated in the same way. This leads to less rework, variability and cycle time. This is also the case for the other standardization options.

These standardization options would make the RE process a lot easier, and requirements elicitation in specific as well. At the moment, some employees complain about no one wanting to elicit requirements. When standardization improvements are implemented, employees will save time, and could therefore shift their focus on other activities they prefer.

In order to prevent employees having to work with numerous different applications, most that are mentioned earlier in this section can be grouped within a single application linked to multiple databases. This would create the option to link different databases, which would lead to different opportunities. For example, specific interfaces can be linked to certain system requirements, so that the requirements engineer is immediately provided with the necessary system requirements when a certain standard interface is identified. A logical segregation within such an application, would be the type of system. For example, a building has completely different requirements as opposed to a railroad track. Two tunnels on the other hand, would probably have more requirements and interfaces in common.

The further development and execution of these options is for future research and not within the scope of this master thesis, but it is certainly recommended to further investigate these opportunities for improvement. To help investigate these opportunities, business cases are necessary, which is further explained in 5.4.1.

5.4.1. BUSINESS CASES

In this subsection, the appliance of business cases for the investment opportunities that are mentioned before, is examined.

The standardization improvements mentioned before will require investments, whereas the total cost savings remain unclear. To help in estimating the total benefits of standardizing certain aspects of an SRS, specific data on SRSs and contract specifications is necessary. Previously, there was no data collection of this sort within Movares. However, in the near future, data on products and processes will be collected, something that was not yet incorporated in the old way of working. A big advantage which has not been mentioned yet, is that data can support business cases. After data is gathered on the RE process, it will be known how much time and money is spent on the RE process. This becomes useful when making a business case for considering an investment in standardizing certain aspects of the processes or products. If Movares could make an educated estimate on the percentage of time saved per product, data will provide the amount of money and time saved per year by standardizing a specific aspect. This kind of information is valuable when considering large investments. The development of data model is further discussed in section 5.5.

5.5. DATA MANAGEMENT

In this part, it is discussed how data management should be applied in practice at Movares and thus builds on section 4.6. The goal of a data management model is to develop a KPI dashboard which is a tool for managers in data driven performance management, to gain insight into the process and support business cases. In order to develop a KPI dashboard, a development team is necessary and more importantly a budget. This budget however is not given to anyone with an idea. For this reason, a concept has to be built incrementally and needs to be proven in order to acquire a budget for further development. Accordingly, a proof of concept is made that is improved and professionalized in each iteration. In the following two subsections, the two main elements of the data model are further discussed; the interface and database.

5.5.1. USER INTERFACE

It is found that all KPIs proposed in this research could be automatically generated from the existing central systems of Movares. The only problem is that the necessary raw data is stored in different underlying systems, which are not easily accessible. In a KPI dashboard, the raw data is visualized to averages and trends that managers can use. In order to make these connections from existing systems, an investment is necessary which is not yet possible for the currently delivered proof of concept. Nonetheless, the data from table 3.4 has shown the possibilities and necessity of further developing the concept. It is necessary to remove barriers of extra work such as manual labor collection, since otherwise employees are likely to not want to gather data or forget about it, which will lower the effectiveness of the eventual dashboard. Measuring the performance of the RE process should require little to no extra effort after an initial investment. The only values that need manual input in the web application, are some of the parameters for the future prediction model. However, the KPIs are independent from the prediction input variables, meaning there is no manual input necessary for calculating the KPIs.

The next opportunity for taking away barriers, is when choosing a platform for the dashboard. It is chosen to build a web application. Another option would for example be to make a form, which could then be reached from the Movares internal website Mint or the internal application ProjectWise. The reason for choosing a web application is that this option is the most accessible, customizable and the easiest to use. A web application removes the barrier of accessibility. A web application does not require additional software, any employee could visit a website. Next to this, a web application is professional, versatile, and easily extendable. The web application is designed for collecting data for SRS documents and the requirements part of contract specifications, but is be easily extendable to other products and processes for future expansion of the model. Additionally, the functionalities could be extended.

The construction of the web application is further discussed in section 6.3, in which the implementation of the different aspects of this research are discussed.

5.5.2. DATABASE

Storing the data shown in the user interface, there needs to be some sort of database. The more data from different sorts of products and processes is stored, the larger and more complex the database becomes. For data management, it is necessary to construct a database of some sort. The database is very straightforward. The function of the database is to facilitate the web interface, which is discussed in the previous subsection and to store the data for the KPI dashboard. There are multiple types of databases, but since there is no need to make things more complicated than they could be, it is chosen to develop a relational database, which is one of the simplest and most often used type of database.

The reason for storing the specific data gathered from the web application without links from central systems is clear. However, for a live dashboard with links from existing systems and databases, it may seem a waste of effort to build an extra database. However, a database could still be useful in the future to support the KPI dashboard. A reason would be to access data faster when it is readily available in a directly connected database, then when the alternative is to calculate KPIs each time again from different underlying existing systems. Next to this, the database would be a backup in case older projects are deleted or spoiled.

For the database, it is also made sure it is suitable for future expansion. For this reason, project-specific information and information on the specific project documents are separated in different tables. A third table consists of employee data.

5.6. CONCLUSION

This chapter is the core of the thesis, specifying a new and standardized way of working and thereby answering sub research question 4. The lessons learned from RE best practices from literature, the old way of working and interviews with experienced employees have led to the construction of a new way of working. This new way of working is documented in a manual for the SRS and can be found in appendix I. In section 5.2, the most important elements of the new way of working are summarized. Key improvements are focusing more on the preparation phase, standardizing to a uniform way of working and allowing requirements engineers to employ a more facilitating role in projects and delegating the technical work to specialized actors.

Concerning the application of RE best practices, the new way of working is improved 35.6% in relation to the old way of making SRSs. Part of this new way of working is gathering data on both SRSs and contract specifications, in order to generate information on the whole requirements engineering process. This information is used in a KPI dashboard, for applying data driven performance management and supporting business cases. To facilitate this, both a web application and database are built to manage the data.

In the future, a KPI dashboard must be made from the lessons learned from the proof of concept. Additionally, the gathered data should be used to support business cases of the standardization options as defined in section 5.5. These options are to standardize recurring elements of the RE process such as requirements by developing a large database, to use standard templates, and standardize booking elements.

In the following chapter of this thesis, it is described how the newly developed way of working is implemented at Movares.

6. CHAPTER 6 – IMPLEMENTATION AT MOVARES

In this chapter, the final sub research question is answered; how can the new way of working be successfully implemented and how can we ensure continuous improvement? This implies that chapter 6 encompasses the final two stages of this research; implementation an evaluation. We described how it ensured that the new way of working is actually applied and accepted.

In order to implement a new way of working, one should always consider the organizational culture in finding a suitable implementation approach, which is discussed in section 6.1. In 6.2, the actual implementation of the new way of working is discussed. In part 6.3, the process of implementing data driven performance measurement is specified, which is more complicated since more different actors are involved. Data driven performance measurement can also drive continuous improvement, which is explained in section 6.4. In part 6.5, the results of and thesis are evaluated based on perceived usefulness. Finally, the chapter is concluded in section 6.5.

6.1. ORGANIZATIONAL CULTURE AT MOVARES

In this first section, we first elaborate on the organizational culture and resistance to change at Movares. Inherently to change, there is resistance to change. Then we discuss how to deal with this when implementing change.

The level of resistance is mostly related to the deep-rooted values in an organization's culture (Pardo del Val & Fuentes, 2003). For this reason, the organizational culture of Movares is examined. Movares has an exceptionally flat and informal hierarchy for a large organization. Employees are given a lot of freedom in doing things their own way and developing good ideas if there is one. An example of this is the web application and database that are built for this research. However, the downside of this freedom is that it is hard to make certain processes mandatory if people do not see the direct advantages for themselves. At the moment, few things like employees or processes are monitored on performance. When there is no monitoring, performance improvement is limited. Employees will often not do anything if there is no top-down enforcement to start doing it. Unfortunately, top-down enforcement is remains difficult at Movares.

Most people in the organization have a technical background, also in the higher managing positions. This implicates that managers in general have a lot of attention for technical improvements, but less attention for process improvements. Due to this relative lack of focus on processes, efficiency and the level of methodical working within Movares can improve significantly.

Given this culture, there are still ways for improving the current processes. The improvements from this research should be managed and further developed by the problem holders, those who perceive the problem the most (Lee & Dale, 1998). Additionally, innovations should be implemented incrementally, not drastically. When one small innovation proves its value, more people will want to get in on it. Since top-down enforcement is difficult, horizontal expansion of the concept is the better option. Therefore, a new way of working is established for the process of making SRSs, which can horizontally be expanded to other processes.

Movares employees in general are resistant to extra activities or extra work to be carried out in comparison to their old way of working. Consequently, one should try to incorporate existing systems as much as possible to lower the needed effort and convince people of the advantages. Interpreting this concept to this thesis, employees such as SI consultants, requirements engineers and managers are interviewed and their needs are considered throughout the whole period of developing the new way of working. Considering the use of existing systems, a lot of aspects of the old way of working are documented in the new way of working.

An important aspect influencing the organizational culture, is the implementation of data driven performance. Data driven performance measurement will enable managers to evaluate their employees in a different way. At the moment, it is often the case that for example an employee has a budget of 8 hours to make a certain document. If that employee is finished in 6 hours, the employee can still write down 8 hours, and satisfy its manager and customer. Movares then loses 25% of an employees' potential which could be used otherwise. When facilitating the possibility of comparing employees' performance based on data, it will become visible that certain employees perform the same task better of faster than other employees. This of course provides a lot of advantages, but also disadvantages. The much-valued freedom of employees would be restricted more in comparison then before. But when striving towards a competitive edge in comparison to competitors, data driven performance will be necessary.

6.2. SRS NEW WAY OF WORKING

In this section, we discuss the implementation of the new way of working concerning the making of SRS documents and the SRS manual. The goal and challenge of this manual is that everyone making an SRS, will apply the instructions from the manual. This way, everyone making an SRS will work in the same standardized way, leading to an average higher quality product in less average cycle time.

To get everyone using the manual, the document is spread as much as possible throughout the organization via people who are known to often be involved with SRSs and the process of deriving system requirements. Additionally, a lunch lecture is held to convince employees of the necessity and advantages of standardization and using the manual. This lunch lecture is necessary, since an SRS is still an abstract concept for a lot of people within Movares. This is especially true for designers, who often view SRSs as a nuisance, since the direct advantages of using it are not visible or known to a lot of the employees working with system requirements. The main reason for making SRSs at the moment, is because the client specifically asks for it. The manual and lunch lecture should start changing this negative perspective on SRS documents. It is important to explain the added value of applying SRS documents in the right way, this is also explained in the manual itself.

The core process owner of making SRSs is the SI department, since this department holds the most knowledge for this document and has historically been the department most involved in this process. It regularly occurs that employees from outside the SI department need to make an SRS and approach people from the SI department for support and SRS examples. In the future, the SI department could just send the SRS manual and be done with it. For this reason, the management of the manual is put under control of the System Integration department.

Now that the SRS process is standardized, the concept of standardizing could be transferred to other documents or products. Before standardizing SRSs, a lot of employees said it would be difficult or even impossible to standardize documents, since the content of every document is unique. This thesis research has proven that standardizing the process, and even the template of products and documents within Movares can be standardized. This implicates that this is also possible for a lot of other processes and products. The department which handles most of the contract specifications has shown big interest and is seriously considering taking steps in order to standardize the making of contract specifications.

Because of the overlap with SRS documents, the SRS manual could be used as input for standardizing contract specifications and also customer requirements specifications.

6.3. DATA DRIVEN PERFORMANCE MEASUREMENT

This section of the thesis is concerned with the implementation of the data management model, thus continuing on section 5.5. Implementing the data management model is more complicated than the SRS manual, since it requires a larger investment and cultural change and involves more employees and departments from the organization. However, this model also holds a larger potential for improving and changing the Movares way of working. The goal of the data model is to create a KPI dashboard which can be used for data based performance measurement, to gather data to support business cases and to develop a platform for a prediction model. In the following two subsections, the development of a KPI dashboard and prediction model are discussed.

6.3.1. KPI DASHBOARD

In this subsection, we elaborate on the development of a KPI dashboard and how it will be implemented at Movares. Initially, a data model is built in MS Excel. This model has two interfaces; one interface consists of a form in which the data on specific projects can be filled out, the other interface then visualizes the data gathered from multiple projects. From this historical data, table 3.4 is made. This however, is labor intensive and it is impossible to find data on all projects within Movares. Data collection would in this case require extra actions from Movares employees, which decreases the chances of people actually filling out the form properly for every SRS. In the next development phase, it is researched how the data model can be professionalized and become more accessible. This next phase, as introduced in section 5.5, is building a web application.

Previously it is mentioned that SRS documents and requirements engineering are often applied in the development of IT applications, from which a large part of the literature in this thesis originates. Developing the web application is therefore quite similar to this process. Beforehand, the basic functionality and requirements of the future application are produced. These are input for making the designs of the future application. While designing, specific elements are further specified and described. The visualization of this design is shown in figure 6.1. This design, together with the specification, is then input for the developers at Movares, who realize the designs. Parallel to the web application, the database is constructed. The web application is in Dutch since this is the first language of the people who will use it. The result of the proof of concept of the web application can be found via the following website: http://movareskpi.azurewebsites.net/. The current proof of concept is attached to a live database and fully functioning, meaning that data can already be gathered.

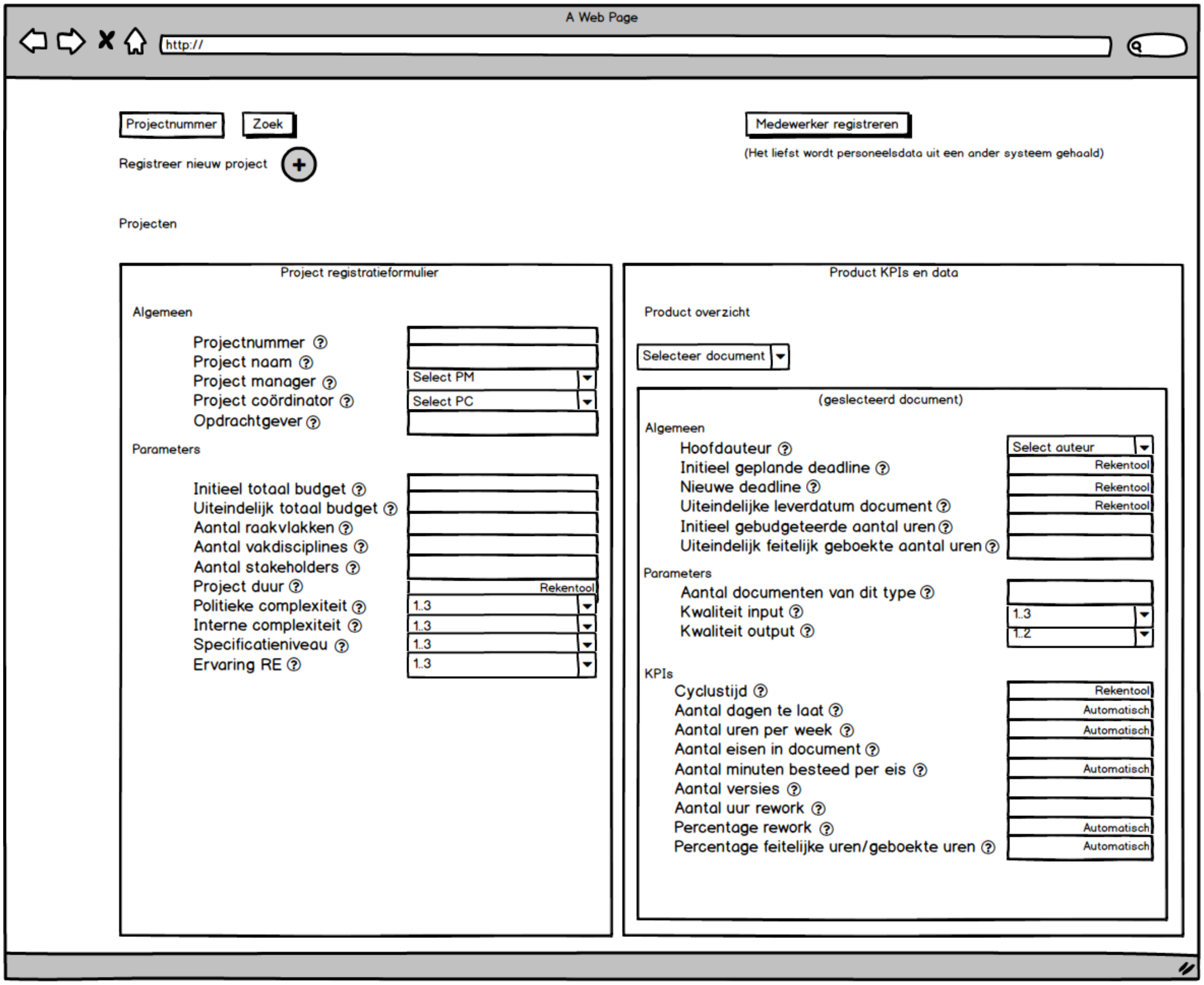


Figure 6.1 - Visualization of the web application

Brook (2014) has determined best practices for data collection, which are applied in the development of the of the dashboard. Per best practice we explain how it is applied:

- Designing the check sheet with people who are going to use it.
    - The KPIs and design are verified by the SI line manager and different project managers.
- Keeping it clear, easy and obvious to use. Trial first to check if it is.
    - Project parameters are presented on the left and SRS KPIs on the right side of the web application. A proof of concept is first created and tested before further development.
- Make sure to communicate. If people do not know why they are being asked to use the checklist, they will not fill it in.
    - The design and use of the web application is discussed with multiple SI employees and both line and project managers.
- Make the records traceable. Do not forget to record names, dates and serial numbers.
    - Every project is linked to its project number. Both names of the project manager and project coordinator as the name of the SRS main author are recorded.

The people to whom the KPI dashboard is most useful are project managers and line managers. For project managers, the KPIs will show the performance of a specific project and projects can be compared to other projects. For line managers such as the managers of the SI team and contract management, trend lines are more relevant and information on the performance of the process in general and employee specific. Employees can then learn from one another and the process can be improved.

So far, only a proof of concept of the KPI dashboard is made. The proof of concept is already used to convince all stakeholders of its use and that further development is necessary. As explained in section 5.5, the final web application should be able to automatically gather KPIs from existing central systems. Employees should then be able to include extra data that is not automatically generated. Also, it should be run on internal servers, since it contains sensitive information. This would then be the next step of the development of the KPI dashboard. Since the master thesis at Movares has ended before the KPI dashboard is completely developed, other people have agreed to take over the development of the Dashboard. At the moment of finishing this thesis, Movares has made more hours available for the further development of the dashboard, on which a small team of consultants and programmers are working actively.

6.3.2. PREDICTION MODEL

This subsection continuous on part 4.7, providing information on how the prediction model should further be developed and implemented. A prediction model is necessary in order to better predict the duration and budgeted hours necessary for making an SRS, based on specific variables. Also, risks such as rework can be identified beforehand. As explained before, the prediction model can be seen separately from the KPI dashboard. However, when the dashboard is built, it should use the same platform and database as the dashboard, since it partly uses the same data.

The output variables of the model are the KPIs from table 3.4. The input variables or independent variables of the prediction model are:

- Complexity, which is defined by a combination of the following attributes:
  - Number of interfaces
  - Number of disciplines involved
  - Number of stakeholders involved
  - Duration of the project
  - Political complexity
  - Internal complexity
- Quality of input
- Intended quality of output
- Level of specification
- Type of client
- Total budget of the project
- Experience of the requirements engineer

The difficulty of implementing the prediction model, is that several parameters require manual entry and enough data is needed to analyze. Employees who qualify for the task of filling out the specific data in the web application can be either project managers, the authors of the document or the project coordinators. These are the direct roles which are always involved in a project in which an SRS document is made. The most suitable actor for filling out and managing the application is perceived to be the project coordinator. The current role of a project coordinator in projects best matches the role of filling out the web application. The coordinator already performs administrable tasks such as starting a project, maintaining the ProjectWise folder, and wrapping up a project administratively. As explained previously in section 6.1, project coordinators are only willing to put extra effort in an activity, if there is an advantage for them as well. The main advantage for project coordinators is the potential of the model to improve planning and prediction modeling and that it would be done much faster. For the prediction model, data does not need to be collected from every single project, as long as there is enough data being gathered for the production model to be accurate. This means that not everyone needs to be convinced of the concept, just as long as a number of employees are convinced of the advantages and fill in the form for several projects.

6.4. CONTINUOUS IMPROVEMENT

If the aforementioned options in this chapter are all implemented, the RE process would reach the fourth level of process maturity. In order to reach the fifth level of process maturity, a continuous improvement model needs to be applied (Harmon, 2007) (Macintosh, 1993). Continuous improvement is the last of the lean construction principles that are analyzed in this thesis, as defined by Sacks et al. (2010). The implementation of this concept is examined in this section of the thesis. A data management model is already introduced. This model is initially used for monitoring the process. Additionally, the data model is eminently suitable for a continuous improvement model. By measuring the data, trends and progress are measured as well. This will allow process improvement, created through benchmarking, continuous improvement and quality information (Lee & Dale, 1998).

The core process owner and team should continually monitor performance, assess results and look for improvement opportunities, effectively a plan-do-check-act (PDCA) cycle (Lee & Dale, 1998). This continuous improvement cycle is depicted in figure 6.2. The core process owner for SRSs is the System Integration department, they possess the most knowledge and control of the process of making SRSs. For this reason, monitoring and improving the making of SRS documents is landed at this department. The review period is in first instance once every six months. After six months, benchmarks can be made and possible improvements can be initiated. The review period is also subjected to change when the process is reviewed. A first logical improvement would be to make a prediction model when enough data is gathered. Although data helps benchmarking for continuous improvement, the team can already improve without using data. In section 5.4, a number of improvement options have been addressed. These can be reviewed every six months. Next to this, the manual is subject to change. In the coming months, when people are actually using the manual, employees could run into issues with the manual. These issues should be reviewed at the end of the six months. In order to make sure this is done properly, a reminder should already be made six months in advance. To help implement the new way of working, the department of System Integration could incrementally implement the manual by focusing on a specific main activity of the manual each few months. These results could then be reviewed at the end of each period.

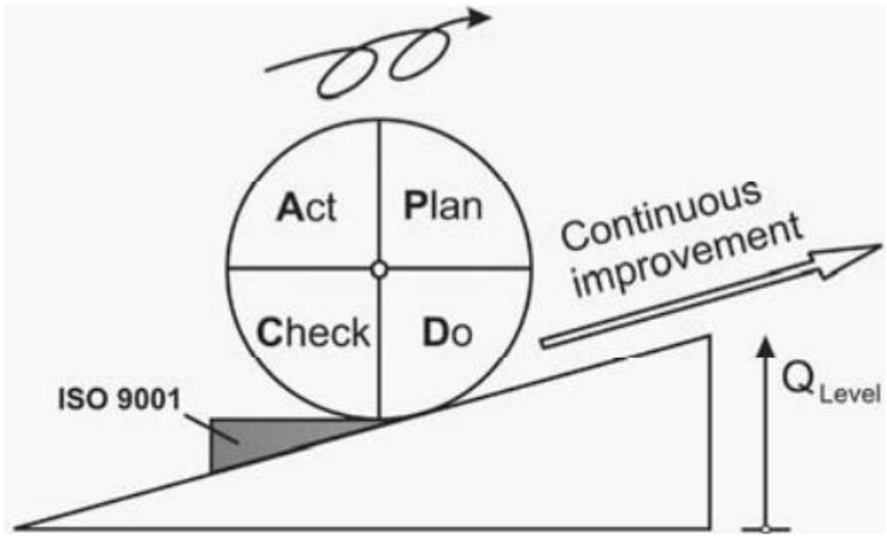


Figure 6.2 - PDCA cycle (Sokovic et al., 2010)

For the process of making the requirements part of the contract specification, the department of contract management is the process owner. There are far more contract specifications being made in comparison to SRS documents, so the review period should be shorter. For this reason, it is recommended to start with a review period of three months.

An important remark here is that concerning the process of system requirements elicitation in general, there is knowledge at both the SI- and contract management departments. In order to improve the process in general, it is recommended that both departments should work together when implementing changes in this process.

6.5. EVALUATION

An important part of the research, is the evaluation. The new way of working has been implemented. The goal of this part is to check to what degree and in what way the chosen solution has actually solved the problem (Heerkens & van Winden, 2012). The core problem is that there is no standardized way of working in making documents at Movares. There is a standardized way of making SRS documents, which can and should be applied to other documents as well. Given the solved problem, the research proves to be successful. However, the extent to which the new way of working will actually be applied and how this influences the performance indicators, remains unclear. Therefore, to help track improvement of the objective of the research objective, two metrics were specified in section 1.2;

- The average amount of hours spent on a particular type of document or product.
    - This aspect is related to cost.
- The average number of revisions of a particular type of document or product.
    - This aspect is related to quality.

These two metrics are also implemented in the KPIs. These metrics should help track the improvement or trend of a process, but unfortunately, there is no automatically generated data available yet. Since the KPI dashboard is under construction and the dashboard can relatively easily be extended to other processes when the dashboard is operational, the concept of tracking the consequences of standardization is better applicable to future standardization projects.

In order to evaluate the new RE manual that has already been implemented, we must resort to other measures instead of quantitative tracking. The continuous improvement model already helps in reviewing the process once every six months, which is also indicated in the research framework from figure 1.5. But, to evaluate the new SRS manual that is made beforehand, we developed an evaluation questionnaire by using the unified theory of acceptance and use of technology (Venkatesh et al., 2003). This will measure the perceived usefulness. From the questions that are suggested by Venkatesh et al. (2003), questions are filtered and adjusted based on suitability for the new manual at Movares. The questionnaire is then sent to Movares. Since the manual is newly made, it is not used yet and only completely read by a few people at Movares. The questionnaire is filled out by the employee working at the System Integration department that has the most knowledge concerning the way people work at Movares. The result of the questionnaire is presented in appendix H.

From the results of the questionnaire can be concluded that the perceived usefulness of the new manual at Movares is high, answering positively to most statements. The most important remark for improvement is that the level of detail of the description on how several specific tasks need to be performed could be better. This is true, but the consideration here is that if the tasks would be specified further, the manual could become too long. The manual is now still manageable at less than 40 pages, but could become incomprehensible if it would be longer in length, increasing the resistance to using the manual.

6.6. CONCLUSION

In this chapter, the actual implementation of the new and improved way of working is discussed, answering the final sub research question 5. Part of the new way of working is the SRS manual. Since most of the knowledge concerning SRSs is held at the System Integration department and they felt the initial need for change, the SRS manual is put under control of this department, but it is also widespread across the company. Additionally, a lunch lecture is held to increase employees' awareness of the possibilities and necessity of standardization in general and the SRS process in specific. Measuring the perceived usefulness, the new manual is evaluated with a positive result. Now that making SRSs is standardized, and the possibility of standardizing processes and products at Movares is proven, it is advised this concept is applied to other processes and products as well.

Implementing data driven performance measurement requires a different, more elaborate approach. This requires a more significant cultural change and a larger commitment and investment from Movares. On the other hand, the potential for working both more efficient and increasing quality, is also considerably larger. A proof of concept is made to show the possibilities of performance measurement and how it could be implemented for multiple products and processes. This concept is further developed in the future with the goal of creating a KPI dashboard, of which the data is automatically generated. Additionally, the data can support business cases and it forms a platform for creating a prediction model. The concept of the prediction model is already specified, the next step in development should be to gather data. When data is measured on the processes, trend lines facilitating benchmarking and continuous improvement will also be made possible. The PDCA cycle is chosen as an easily applicable and simple concept.

In order to make sure everyone is on the same page, and stakeholders of the new way of working and data model are involved, communication is key. For this reason, a lot of meetings are held with all stakeholders and to conclude a lunch lecture is held for anyone interested within the organization.

7. CHAPTER 7 – CONCLUSIONS & RECOMMENDATIONS

This final chapter concludes this master thesis and provides recommendations for future research, it provides an overview of the most important results of this research.

The objective of this thesis is to analyze the existing processes of SI, in order to find a way to standardize the process of making plans and documents for the successful continuation of projects at the economically best way possible, regarding cost and quality. This objective is described more tangible in the main research question, which is; *How can a standardized system requirements engineering process be realized at Movares, in order to move towards a more economical way of working and allow SI employees to employ a more organizational role in projects?*

To answer this question, five sub research questions are devised and answered separately throughout chapters 2 to 6. The sub research questions are designed in such a way, that when these questions are answered, an answer can be constructed to the main research question. In this chapter, the main research question is answered. Section 7.1 concludes and summarizes the main results from this thesis. There are some limitations to this research, which are discussed in 7.2 Taking these limitations and looking beyond this thesis, Movares can further improve and standardize its processes in the future, for both the RE process as other processes. This future necessary research in the form of recommendations is presented in part 7.3.

7.1. CONCLUSIONS

This subsection provides an overview of the most important results of this master thesis. In general, the conclusions of this thesis can be split into two parts; the new, standardized way of working when making SRS documents and data driven performance measurement, although these two parts are strongly intertwined. The two parts are discussed in subsection 7.1.1 and 7.1.2 respectively.

7.1.1. NEW WAY OF WORKING

From interviews with employees from the SI department, several needs rose to the surface. One such need is that SI employees are often asked to make the complete SRS documents, including the elicitation of the requirements themselves. However, this requires specialized technical expertise, which SI employees often do not possess. SI employees are namely specialized in finding interfaces between different subjects, assessing risks and directing and structuring processes. Thus, the technical tasks need to become more delegated to the technical disciplines, who often do not know how to correctly elicit requirements. Another issue is that work processes are often not documented and employees then usually work from own experiences, which leads to employees having to reinvent the wheel and working inefficiently. Additionally, and relatable to the previously mentioned issue, employees experience the quality of some of the documents as too low and too inconsistent. In this thesis, we have chosen to analyze the RE process and SRS documents in particular.

To solve the aforementioned issues, the process is documented in detail and standardized by using both the best practices from the current way of working from multiple employees and the best practices from literature on RE. This results in the SRS manual, which can be found in appendix I. This manual is set up in such a way that it is readable for anyone. Experienced requirements engineers should easily navigate through the document to find the specific aspect on how they need to perform a task. For employees who are new to requirements engineering, the process and purpose are specified as well. Especially explaining the purpose will help employees change their general view on the SRS from it being a nuisance, towards a tool that helps working in a more structured way in complex environments.

Compared to the old way of working, the new way of working has changed significantly, as can be deduced from appendices B and C. The new way of working provides the possibility to assign tasks more clearly and delegate technical tasks easier. The role of an RE becomes more facilitating. Next to this, both the duration of the process and quality of the output will improve. This is due to the fact that when starting the process, more time is spent on thinking about future steps and possible risks, so that more time will be saved later on. Furthermore, everyone will work in the same way, providing higher and more consistent output quality. Finally, the document's quality standards are stricter and better specified to SRS documents than before. Although SRSs are made throughout the organization, the SI department possesses the most knowledge on RE and SRS documents, this means that the SRS manual is put under control of this department. To make sure more requirements engineers will apply the manual, the manual is spread as broadly as possible among employees who are regularly involved in the process.

Quantifying the improvement of the new way of working, the level of standardizing RE best practices is improved by 35.6%. Additionally, the process maturity level is now raised from level 2 to level 3. If the process would be managed based on data, the level would increase to level 4. The final level 5 would be reached if continuous improvement is applied. Movares' level of working methodologically will increase, and if expanded to more processes, Movares' competitive edge will improve as well.

7.1.2. DATA DRIVEN PERFORMANCE

In order to obtain the maximum potential of this new way of working, data is needed to monitor the process. This will provide insight into the process and facilitate continuous improvement. For this reason, a significant part of this thesis is spent on researching the possibilities of data collection for the purpose of making a KPI dashboard. Among other advantages, this will enable the possibility of data driven performance measurement. Implementing this concept is harder than implementing the SRS new way of working, but it also has more potential for improving Movares' performance in general. Next to the aforementioned advantages, data driven performance measurement will facilitate benchmarking, detecting inefficiencies, monitoring cost and quality. Other possibilities are analyzing trends, comparing different projects´ performance, but also measuring individual employees´ performance. Currently, employees are not monitored based on such data, and employees are given a lot of freedom and responsibilities. Implementing this change, will require a significant cultural change. Since it is hard to implement this concept, it was chosen to start small and only collect data on the requirements engineering process. First, a basic web application is built to show the possibilities of such a data model. Next, a dashboard with more functionalities and linkages from existing Movares systems is built incrementally.

In order to showcase the possibilities of a KPI dashboard, historical data is collected from 21 SRSs. Table 3.4 provides the operationalized KPIs based on this data. From these KPIs, it turns out that making SRSs is actually not a core process of the SI department. This proves the fact that data management helps gaining better insight in the processes. Additionally, the results indicate that the process is not performing well on certain aspects such as the average cycle time and the average number of days late. Next to this, there is a high variation in performance- and quality indicators. Standardizing the process will reduce this variation.

7.2. RESEARCH LIMITATIONS

In this part of the thesis, we discuss the limitations of the research. There are multiple limitations that need to be considered. The limitations are mostly related to the scope, time and data:

- Scope. The scope of this research is limited to the requirements engineering process. It is proven that standardizing the RE process can improve the process significantly. Further research is necessary to find out to what extent standardization improves other processes, whether the process owners perceive the lack of standardization as an issue and to what extent it can be standardized without making too large of an investment. The effectiveness of the approach used in this research may differ per process.
- Time. There are a few limitations related to time. This research has been performed over the course of half a year. The new way of working and manuals have been implemented at the end of this period. Because of this, employees' experiences of actually applying the new way of working and manual cannot be evaluated. Instead, the perceived usefulness is measured beforehand and an internal evaluation will take place several months after the research. This is part of the continuous improvement model. Additionally, the data model including the KPI dashboard and prediction model has not been finished yet. The concept is initiated and the dashboard is being further developed at Movares, but not before the end of this graduation assignment.
- Data. At the beginning of the graduation internship at Movares, SI employees perceived the process of making SRSs to be a key process of the SI department, of which a lot of data would be available. It later turned out that it is hard to gather data, since it is hidden in different underlying systems and could only be gathered manually, unless a programmer develops an application to gather the data. For this reason, it is important to have such an application in the future, gathering key measures automatically from different processes.

The research limitations are considered for future work and recommendations, which are elaborated upon in the next section, 7.3.

7.3. FUTURE WORK AND RECOMMENDATIONS

Throughout the thesis, multiple issues and aspects are exposed, which could still be improved or that need further research in the future. These issues and aspects are formulated as recommendations for Movares. Recommendations are only made on those aspects that have been extensively discussed in this thesis. The recommendations are shown below in order of importance for Movares in general:

- Professionalize the web application. The web application is built as a proof of concept, with basic functionalities. The next steps in making a KPI dashboard would be to professionalize the proof of concept to a visual dashboard and to create links from existing Movares systems, so the data is completely automatically generated and provided. For expansion of the model to other processes, general KPIs for Movares' processes have already been designed in this thesis. A subsequent product to apply a dashboard to, would be CRS documents. Then, the entire requirements engineering process would be grasped, including customer requirements. This should be done by the team already working on it.
- Standardize booking elements. The booking elements, which represent the activities on which employees record the hours they have worked, are different and reinvented for every project. This leads to inefficiencies and confusion among employees concerning the specific elements on which

they need to record the hours they have worked. Standardized booking elements are necessary for reliable and uniform data. It is highly recommended to do this in the near future, also since every employee interviewed recognizes the added value of standardizing this aspect. This is even endorsed by the interviewed project coordinators, who devise the booking elements. Standardizing booking elements should be initialized by the managers of the project coordinators. Until then, other employees should lobby for it.

- Review the SRS process and manual every six months. Now that the process is standardized, it should be continuously improved. To support improvement, the value of data driven management is recognized, and applicable in the near future. The KPIs and review process have already been developed, now it is only a matter of actually applying data driven management as soon as the dashboard is finished. Reviewing the process should be done by the SI department.
- Research business cases. Related to continuous improvement above, improvements should be considered as business cases. Once data on certain processes is available and visualized, business cases can be made in order to support investments in standardization options. Concerning the requirements engineering process, such options have been discussed in section 5.2. These standardization options consist of developing databases of recurring interfaces, risks, functions, requirements and issues. Applying these options in practice will save time and make the process a lot easier. In researching business cases for the RE process, the departments of SI and contract business should work together, since they both possess a lot of knowledge on the RE process. Researching business cases should be done by the department of ReDesign in cooperation with the SI department.
- Standardize more products and processes, apart from SRS documents. Within Movares, it is generally conceived that standardization is not or hardly possible. However, this thesis proves otherwise. The specific content of most documents and products is usually unique, but the processes to make such products can be standardized. The department at which contracts are made have already shown interest in applying this concept to their products and processes. Standardizing processes should be researched at all departments.
    - Within the department of System Integration, other products that could be standardized are documents such as the CRS, SBS, FBS and V&V management plan. Most of these documents are intertwined; one document’s output is another document’s input, which makes it logical to continue standardizing these documents first.
- Develop the prediction model. With the development of the KPI dashboard, the development of an incorporated prediction model has been pushed towards the background, since it requires extra handlings from employees. This means it is harder to implement. However, it certainly is advised to further develop this model after a KPI dashboard is built and data is being gathered. This dashboard then provides a platform in which a prediction model can be developed. The application Weka could be used to produce a prediction model. When enough data is gathered, a GR&R study should be performed in order to assess the precision error in data. Developing the prediction model should be done by the same team developing the KPI dashboard.

REFERENCES

Anumba, C. J., & Evbuomwan, N. F. (1997). Concurrent engineering in design build projects. *Construction Management and Economics*, 271-281.

Arayici, Y., Ahmed, V., & Aouad, G. (2006). A requirements engineering framework for integrated systems development for the construction industry. *Journal of Information Technology in Construction*, 35-55.

Beimborn, D., Gleisner, F., Joachim, N., & Hackethal, A. (2009). The Role of Process Standardization in Achieving IT Business Value. *System Sciences*.

Black, C., Akintoye, A., & Fitzgerald, E. (2000). An analysis of succes factors and benefits of partnering in construction. *International Journal of Project Management*, 423-434.

Boddy, D., & Paton, S. (2011). *Management: An Introduction.* Pearson Education Limited.

Brook, Q. (2014). *Lean Six Sigma and Minitab (4th Edition).* OPEX Resources Ltd.

Brosch, M., Beckmann, G., & Krause, D. (2012). Towards an integration of supply chain requirements into the product development process. *International design conference*, 23-32.

Bush, M. (1990). Improving software quality: the use of formal inspections at the jet propulsion laboratory. *12th International Conference on Software Engineering.*

Cheung, S.-o., Ng, T. S., Wong, S.-P., & Suen, H. C. (2003). Behavioral aspects in construction partnering. *International Journal of Project Management*, 333-343.

Cox, K., Niazi, M., & Verner, J. M. (2009). Empirical study of Sommerville and Sawyer's requirements engineering practices. *IET Software*, 339-355.

Fernández, D. M., Wagner, S., Lochmann, K., Baumann, A., & de Carne, H. (2012). Field Study on Requirements Engineering: Investigation of Artefacts, Project Parameters, and Execution Strategies. *Information and Software Technology*, 162-178.

Few, S. (2006). *Information Dashboard Design: The Effective Visual Communication of Data.* O'Reilly Media.

Fisher, K. W. (2004). Shipbuilding specifications: Best practice guidelines. *International Journal of Maritime Engineering*.

Grady, J. O. (2006). *System Requirements Analysis.* Academic Press.

Harmon, P. (2007). *Business Process Change: A Guide for Business Managers and BPM and Six Sigma Professionals.* Burlington, MA: Morgan Kaufmann.

Hatton, S. (2008). Choosing the "Right" Prioritisation Method. *19th Australian Conference on Software Engineering*, pp. 517-526.

Hayes, R., Pisano, G., Upton, D., & Wheelright, S. (2005). *Operations, Strategy and Technology: Persuing the Competitive Edge.* Hoboken, NJ: John Wiley & Sons, Inc.

Heerkens, J. M., & van Winden, A. (2012). *Geen probleem, een aanpak voor alle bedrijfskundige vragen en mysteries.* Business School Nederland.

IIBIA. (2009). *A Guide to the Business Analysis Body of Knowledge (BABOK Guide)* (2nd edition ed.). International Institute of Business Analysis.

INCOSE. (2006). *Systems Engineering Handbook.* Seattle, WA.

Inman, L., & Milosevic, D. (1999). Project Management Standardization and its Impacts on Project Effectiveness. *Portland International Conference on Management of Engineering and Technology.*

Katasonov, A., & Sakkinen, M. (2006). Requirements quality control: a unifying framework. *Requirements Engineering*, 42-57.

Kramer, J., & Magee, J. (1990). The Evolving Philosophers Problem: Dynamic Change Mangement. *IEEE Transactions on Software Engineering*, 1293-1306.

Lee, R. G., & Dale, B. G. (1998). business process management: a review and evaluation. *Business Process Management Journal*, 357-378.

Leidraad. (2009). Leidraad voor Systems Engineering binnen de GWW-sector versie 2.

Leidraad. (2013, November 19). Leidraad voor Systems Engineering binnen de GWW-sector versie 3.

Macintosh, A. L. (1993). The need for enriched knowledge representations for enterprise modelling. *Artificial Intelligence in Enterprise Modelling*.

Pandey, D., Suman, U., & Ramani, A. K. (2010). Performance measurement of different requirements engineering process models: A case study. *International Journal of Computer Engineering*, 1-15.

Pardo del Val, M., & Fuentes, C. M. (2003). Resistance to change: A literature review and emperical study. *Management decision*, 148-155.

ProRail. (2015, April). Handboek Systems Engineering (SE).

ProRail. (2016). *Resultaten Prestatiemeten Movares Nederland BV.* ProRail.

Sacks, R., Koskela, L., Dave, B. A., & Owen, R. (2010). The Interaction of Lean and Building Information Modeling in Construction. *Journal of Construction Engineering and Management*, 968-980.

Sedlmeier, P., & Gigerenzer, G. (1997). Intuitions About Sample Size: The Empirical Law of Numbers. *Journal of Behavioral Decision Making*, 33-51.

Shahin, A., & Mahbod, M. A. (2007). Prioritization of key performance indicators: An integration of analytical hierarchy process and goal setting. *International Journal of Productivity and Performance Management*, 226-240.

Sokovic, M., Pavletic, D., & Kern Pipan, K. (2010). Quality Improvement Methodologies - PDCA Cycle, RADAR Matrix, DMAIC and DFSS. *Journal of Achievements in Materials and Manufacturing Engineering*, 476-483.

Sommerville, I., & Ransom, J. (2005). An Emperical Study of Industrial Requirements Engineering Process Assessment and Improvement. *ACM Transactions on Software Engineering and Methodology*, 85-117.

Tan, R. R., & Lu, Y.-G. (1995). On the quality of construction engineering design projects: criteria and impacting factors. *International Journal of Quality & Reliability Management*, 18-37.

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003, September). User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 425-478.

Verschuren, P., & Doorewaard, H. (2010). *Designing a Research Project.* Den Haag: Eleven International Publishing.

Wohed, P., van der Aalst, W., Dumas, M., ter Hofstede, A., & Russel, N. (2006). On the Suitability of BPMN for Business Process Modelling. *4th International Conference, BPM 2006* (pp. 161-176). Vienna: Springer.

Wu, H.-Y. (2012). Constructing a strategy map for banking institutions with key performance indicators of the balanced scorecard. *Eavaluation and Program Planning*, 303-320.

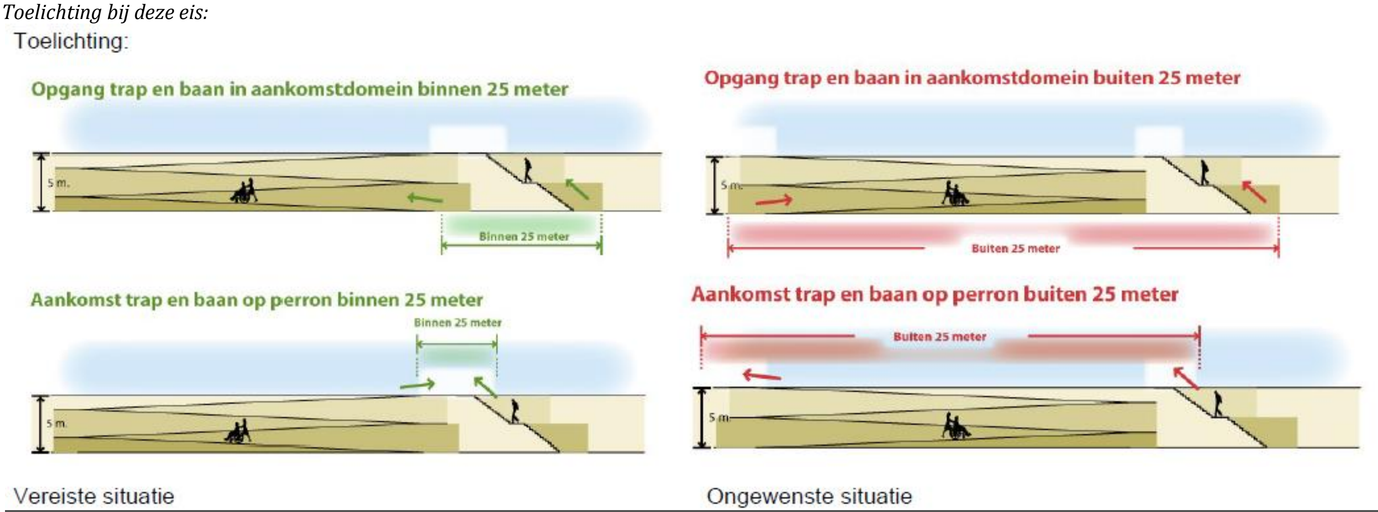
APPENDIXES

APPENDIX A – REQUIREMENTS EXAMPLES

This appendix contains an example of two requirements for a project to make a tunnel underneath a train station in Gorinchem.

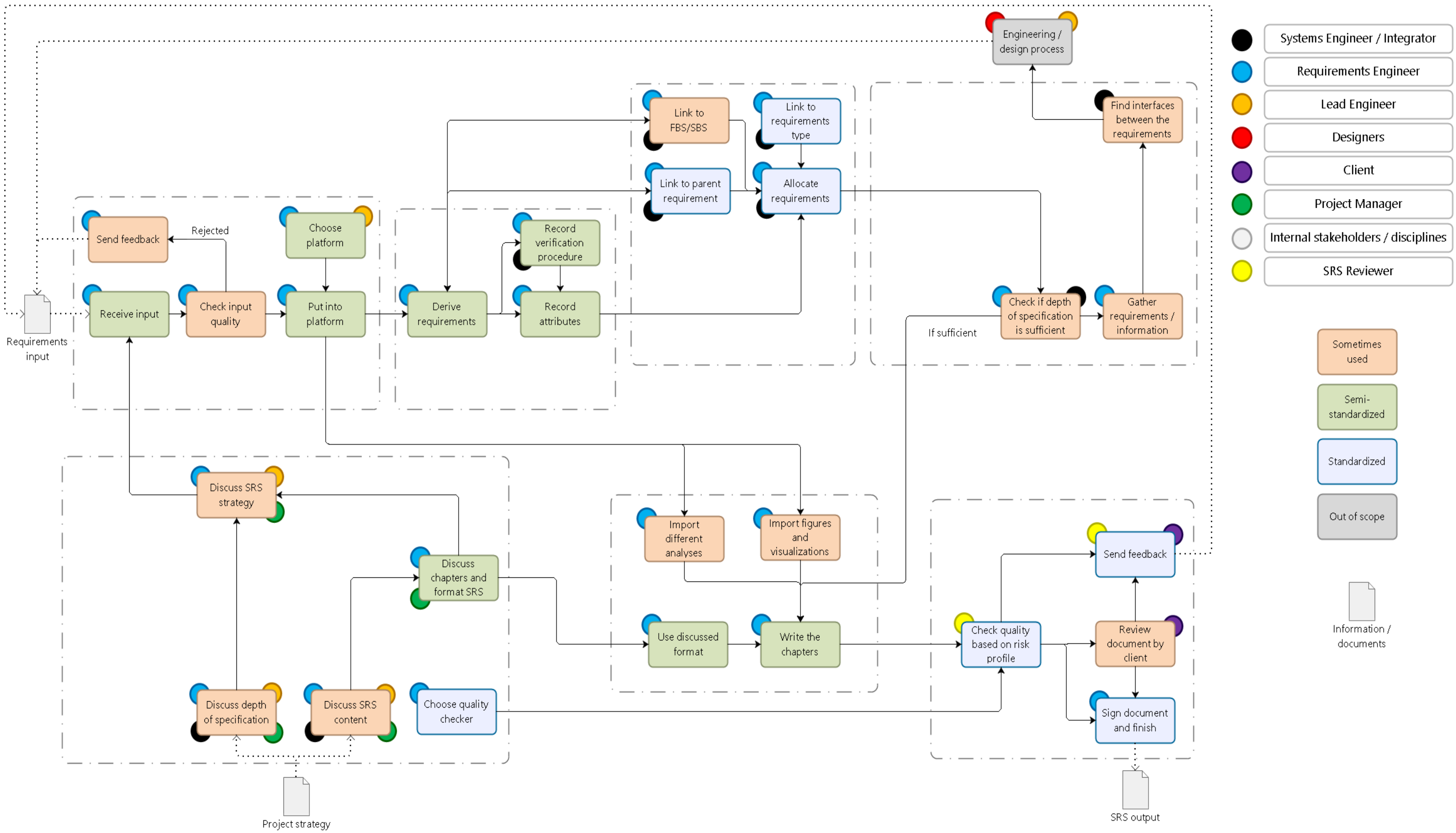
ID	Titel van de eis	Omschrijving van de eis	Bron document	Bron eis	Boven liggende eis	Eis initiator	V&V
1.2.4.10	Logische routing	NRSG dient de Routing logisch, direct, met beperkte loopafstanden en niet discriminerend te laten zijn. Concreet houdt dit in: start en eind hellingbaan/lift binnen een afstand van 25m van de vaste trap in de hoofdroute.	CRS	CRS-708	1.2.4	PPSOB	Uitvoerings ontwerp. Tekeningen

Toelichting bij deze eis:

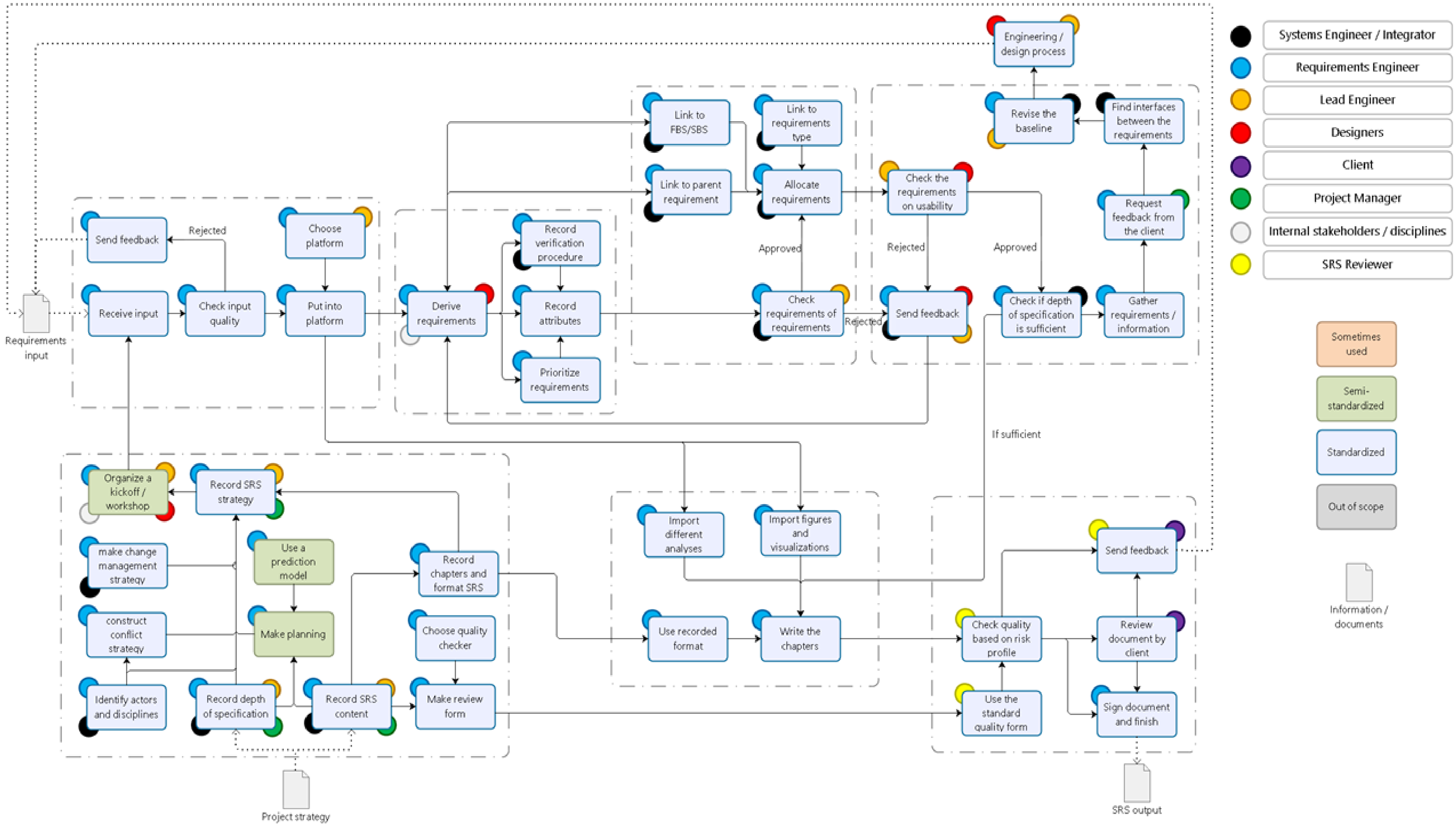


ID	Titel van de eis	Omschrijving van de eis	Bron document	Bron eis	Boven liggende eis	Eis initiator	V&V
1.2.4.11	Indeling Stations	NRSG dient te voorzien in logische routing voor fietsers, hulpdiensten.	CRS	CRS-1201	1.2.4	NSS	Uitvoerings ontwerp. Tekeningen

APPENDIX B - THE CURRENT SRS PROCESS



APPENDIX C – COMPLETE OVERVIEW OF THE NEW PROCESS



UNIVERSITY OF TWENTE.

Movares
adviseurs & ingenieurs

APPENDIX D – GENERAL DOCUMENT CHECKLIST

	Waarop wordt gecontroleerd?	Bevindingen	Bevindingen verwerkt? Indien nee, waarom niet	Aanpassingen correct verwerkt?
Layout				
☒	**Is er sprake van een consequent gebruik van opmaak (huisstijl, stijlen, lettertype, lettergrootte, uitlijning, opsommingen)?**			
☐	Is de juiste huisstijl (Movares danwel projectspecifiek) gehanteerd?			
☒	**Is de inhoudsopgave bijgewerkt?**			
☒	**Kloppen bladnummers en voettekst?**			
☐	Is sprake van een logische bladindeling (witregels/nieuwe pagina)?			
☐	Is voldoende gebruik gemaakt van verhelderde tabellen, afbeeldingen etc. om het uiterlijk aantrekkelijker te maken?			
Spelling / woordgebruik				
☒	**Is het rapport gecontroleerd op spel- en typefouten?**			
☒	**Is sprake van een consistent gebruik van terminologie?**			
Kwaliteit				
☐	Bevindt document zich in de juiste fase van werkprocedure? Versiebeheer correct?			
Volledigheid / uniformiteit				
☐	**Aanwezigheid bijlagen?**			
☐	PW velden gevuld (invoegtoepassing) – invullen lijst welke velden gevuld moeten zijn			
☐	Verwijzingen naar tabellen/figuren/bijlagen aanwezig en juist?			
☒	**Bij conclusies, voldoende herleidbaar waarop de conclusies gebaseerd zijn?**			
☐	Eenheden genoemd waar nodig?			
Inhoudelijke kwaliteit				
☐	Is de gehanteerde structuur passend?			
☐	Bij herhalingen/samenvattingen, informatie nog steeds gelijk?			
☐	Klopt inhoud technisch?			
Doel				
☒	**Blijkt het doel/belang van het rapport duidelijk uit de inleiding?**			
☐	Is sprake van een duidelijke hoofdvraag en eventuele subvragen?			
☐	Is sprake van een duidelijk antwoord op de hoofdvraag?			
☐	Is de structuur van het rapport voor de lezer te volgen?			
☒	**Is het mogelijk om aannamen, berekeningen, tussenresultaten en conclusies te herleiden?**			
☐	Kunnen de aanbevelingen en conclusies gelinked worden met de wensen van de opdrachtgever?			
☐	Zijn de argumenten onder de voorstellen aan de opdrachtgever voldoende te doorgronden?			
☐	Is de aangeboden rapportage voldoende voor de opdrachtgever om de juiste beslissingen te nemen?			
Eisen				
☒	**Wordt voldaan aan de eisen die gesteld zijn aan dit product?**			
☐				

APPENDIX E – REQUIREMENTS-SPECIFICIC REQUIREMENTS

Requirements-specific requirements	
Requirement	*Explanation*
FORM	
Uniquely identified	A requirement holds a unique ID
Traceable to source	A requirement should be traceable to a source
Traceable to level	A requirement should be traceable to a higher level in the requirements-tree
Traceable to function and/or object	A requirement should be traceable to function or object
Objective	Requirement needs to avoid subjective terms such as “good”, “safe” and “and as soon as possible”
Linguistically correct	A requirement is linguistically and rhetorically correct
Active	A requirement should avoid the grammatical passive form
Pressing	A requirement should have a pressing syntax, using words as “should” and “must”. A requirement cannot be optional
Positive	A requirement should avoid negatives such as “not” or “no"
Prioritized	It should be clear which requirement needs to be fulfilled above others. This should be derived from the source
Concise	A requirement should be brief, minimal and relate to the core
CONTENT	
Feasible	A requirement should lead to least one solution and implementation
Necessary	A requirement should be necessary to come to a functioning and complete system
Complete	A requirement is complete when it has sufficient level of detail for its purpose
Consistent	A requirement cannot contradict other requirements
Unambiguous	A requirement can only be interpreted and explained in a single way
Verifiable	A requirement should be able to be verified
Solution-free	requirement should be free of prescribed solutions, components, suppliers, etc.
Singular	A requirement can only hold one criterion on which a requirement should satisfy
Measurable	It should be clear when the purpose of a requirement is met
Acceptable	A requirement should be acceptable to stakeholders

APPENDIX F – BEST PRACTICE GUIDELINES

#	*Guidelines*	*Explanation*
	SRS management guidelines:	*Subtotal:*
1	The document is easily changeable	Easy to apply changes/modifications. For example, Relatics is used
2	Standard document structure is defined	A standard document structure is used, little modifications were necessary
3	Standard template for the SRS is specified	A standard template is used, little modifications were necessary
4	Standard template for requirements is specified	A standard template for the presentation of requirements is used
5	Software is provided to support negotiations	Software to support communication and negotiations is used, such as Relatics
6	There is a plan for conflicts and conflict resolutions	Conflicts can easily be solved due to this plan
7	Requirements are reviewed by people with multi-disciplinary background	Or by multiple people with different disciplinary backgrounds
8	Both internal and external reviewers are involved in the validation process	External reviewers are reviewers who are not directly involved in the project and have a fresh view on the project
9	Policies for requirements management are defined	Policies on how to derive requirements
10	Define change management policies	A policy for making changes/modifications to the document is specified beforehand and applied
11	Quality of the SRS is checked	A reviewer checks the quality of the document before it is finished
12	Maintain a good relationship among stakeholders	Better satisfy stakeholder needs. Stakeholders also include internal stakeholders
13	Identify and consult all likely sources of requirements and consult system stakeholders	

	Guidelines on the SRS content:	*Subtotal:*
14	Use of the document is explained	The document should be readable for everyone, including people without background knowledge
15	Specialized terms/jargon is explained	The document should be readable for everyone, including people without a technical background. All derivations and specialized terms should therefore be explained
16	The document has a good table of contents and is well-readable	Refers to the table of contents. Help readers find information
17	Diagrams and figures are used appropriately	Diagrams and figures are used to visualize certain aspects of the project

18	System stakeholders are identified and visualized	Stakeholders are visualized according to importance and influence
19	Systems environment is modeled	A model or visualization of the system environment and boundaries is given
20	The system scope and boundaries are defined	The scope of the system and its boundaries are defined
21	The interfaces are described	A list of all of the interfaces is given
22	Interaction matrices are used to find conflicts and overlaps	Matrices such as an N2-matrix and visual representations of system context
23	Systems architecture is modeled	A model or visualization of the system architecture is given
24	Traceability policy is defined	Traceability of requirements is performed using a predetermined policy and is explained in the text
25	A list of sources is given	List of source documents
26	An FBS is used	There is a clear link between the requirements and the FBS. And the FBS is shown in the appendix
27	An SBS is used	There is a clear link between the requirements and the SBS. And the FBS is shown in the appendix

	Guidelines on the requirements:	*Subtotal:*
28	Requirements are prioritized	Each requirement has a certain priority, which shows in the SRS document
29	Verification checklist is defined	Per requirement it is defined how it is checked whether it is actually considered and applied
30	Requirements are traceable to a source	The source of each requirement is clear and easily traceable
31	Requirements are traceable to an object or function	Each requirement is traceable to a location in the FBS and/or SBS
32	Business concerns are specified in requirements	Concerns which are apparent to stakeholders are specified in the requirements
33	Each unique requirement is identified	Each requirement holds a unique ID
34	Specify systems using formal specifications	System requirements are specified using formal language
35	Formal requirements inspection is organized	There is a formal inspection/review of the requirements organized
36	Rejected requirements are recorded	Old requirements that are rejected or changed significantly are recorded and kept
37	Use a database to manage requirements	
38	Define policies for requirements management	
39	Identify global system requirements	This could be in the form of a list of standard requirements or the repetitive use of certain requirements
40	Specify requirements quantitatively	Requirements should be measurable
41	Collect requirements from multiple viewpoints	
42	Record requirements rationale	Improves requirements coverage

APPENDIX G – SRS QUALITY CONTROL CHECKLIST

Part	*Requirement*	*Explanation*
General remarks	Specialized terms/jargon is defined	The document should be readable for everyone, including people without a technical background. All derivations and specialized terms should therefore be explained.
	The document is well readable and understandable	The document should be readable for everyone.
	Language is simple, correct, consistent and concise	The document should be readable for everyone.
	The content is complete and consistent	The general content of the document is consistent with input documents.
	Diagrams are used appropriately	To visualize certain aspects.
	The proper standard template is used	The proper standard template from the client or Movares itself.
	The project-specific norms are met	Predetermined norms and goals of the project are visibly considered.
	The product-specific norms are met	The predetermined requirements are met, such as the one from the work description and work procedure of an SRS.
Lay-out	Table of contents is provided	Table of contents is logical and helps the reader in finding information.
	The document should be organized	The document is logically organized.
Introduction	General information	Introduction contains background information of the project and system, motivation, the most important measures taken, a ground of the site and satellite pictures.
	The function of the SRS is explained	The goal and function of an SRS within the engineering process should be explained, and how the document should be used.
System and system context	The goal of the projects and system are described	Also, the definition and description of the system itself and main functionality. The SBS and/or FBS of the system are also referred to in this part, important is that these models have enough depth. The definition of the system should not be limiting the scope.
	The current situation is described	The current situation describes the area within the system boundaries and direct area outside of the boundaries, the current situation in relation to (third party) objects and infrastructure in the area, land owners and users, destination plans for the area and other relevant aspects around the system boundaries such as access roads.

	The desired situation is described	The desired situation refers to the general outline of the CRS and the aspects mentioned above, except in relation to the future situation.
	System boundaries described	A clear and functional description of the system boundaries are described. This is important, since there are requirements on the system boundaries.
	System context is described	The system context encompasses the system in relation to other systems. After all, a system is always part of a bigger system. The system context is related to the architectural design, and visually presented in a star diagram with the system of interest in the middle.
	Complementary system models are developed	Models of the system boundaries, system environment and system architecture are given.
	Interfaces deliberated in detail	The interfaces are already introduced in the system context, and further elaborated in a separate part afterwards, including a schematic representation. An internal interface overview in the form of a N^2-matrix can also be provided.
	Stakeholders are identified and visualized	Stakeholders should be mentioned and schematically presented in order of importance, primary, secondary and tertiary stakeholders.
Scope	The system scope is clearly defined	The scope is related to the previously mentioned system context and describes the boundaries of the project, defining what exactly lies inside and what lies outside of the scope. No questions arise. These boundaries or aspects of the scope can be listed in a table, in which per aspect is specified what lies within and outside of the scope.
Applicable documents	A list of sources is given	This part contains list of binding and informative documents that contain source information on the system requirements that are listed later in the SRS. A system requirement needs to be able to be traced back to source documents, which are listed here.
Design decisions	A list of the important design decisions is given	Throughout the engineering phase, decisions are made concerning the design of the system. It is important to register these decisions, including the source document.
Requirements	A list of the requirements is given	List is complete. There are functional-, aspect-, interface-, and realization requirements.
	All requirements are structured	The requirements are properly structured according to type of requirement and/or function.
	The links between stakeholder requirements and system requirements are documented	There is a clear and visible link between the SRS and the input documents.
	There are only system requirements	For example, process requirements do not belong in the system requirements specification. There are only system requirements on the system.

List of modifications	List of modifications is provided	The engineering process is an iterative process in which requirements are added, removed and changed over time. Decisions on changing these requirements are kept and tracked in a list.
Verification & Validation	Verification & Validation management should be specified	The content of this part leads from the V&V management plan, this is another document which specifies the way a verification and validation of requirements and designs should take place.
Attachments	Appropriate appendices are attached	Attachments such as the FBS & SBS. These are referred to in the text.

APPENDIX H – QUESTIONNAIRE ON PERCEIVED USEFULNESS

#	Questions	Answer	Explanation if necessary
1	I would find the new manual useful in my job	Yes	
2	Using the manual enables me to accomplish making an SRS more quickly	No	I expect to be slower but produce higher quality
3	My interaction with the manual is clear and understandable	Yes	At a high level, I understand how it works, it gives me less to hold on to in detail
4	It would be easy for me to become skillful at using the manual	Yes	It's comprehensible in lay-out
5	I would find the manual easy to use	Yes	
6	Using the manual is a good idea	Yes	
7	The manual makes work more interesting	No	
8	Working with the manual is fun	No	This work isn't fun
9	In general, the organization supports the use of the manual	Yes	
10	I have the knowledge and resources necessary to use the manual	Yes	
11	The manual is compatible with other systems that I use	Yes	
12	A specific person (or group) is available for assistance when I run into difficulties with the manual	Yes	
13	I could complete an SRS using the manual if there was no one around to tell what to do as I go	No	Necessary detail is missing
14	I could complete an SRS using the manual if I could call someone for help if I got stuck	Yes	
15	I feel apprehensive about using the manual	Yes	
16	The manual is somewhat intimidating to me	Yes	
17	I intend to use the manual in the coming 6 months	Yes	If I need to make a SRS
18	I predict I would use the manual in the coming 6 months	Yes	If I need to make a SRS
19	I plan to use the system in the next 6 months	Yes	If I need to make a SRS

APPENDIX I – SRS MANUAL

On the next page starts the newly made manual for making SRSs. This is in Dutch, since Dutch is the first language of the people who will use the manual.

Handleiding SRS/SES

Dit document bevat de handleiding voor het maken van een SRS (System Requirements Specification) en eisen afleiden volgens de Movares manier van werken, voor een eenduidige manier van werken. Een SRS heet ook wel in het Nederlands een SES (Systeem Eisen Specificatie). Door het document heen zijn de activiteiten beschreven die leiden tot een kwalitatief goede SRS.

Doel van een SRS: Het beschrijven van de functionaliteit van een systeem. Systeemeisen zijn eisen welke limieten opleggen aan ontwerpbesluiten en waaraan het uiteindelijke systeem dient te voldoen.

Toegevoegde waarde van een SRS: Het SRS proces is een gestructureerde en standaard manier van werken, dat zorgt voor een betere kwaliteit van de output binnen een project en minder rework. Een SRS is belangrijk voor het vastleggen van eisen en beslissingen aan het systeem, en de herleidbaarheid daarvan. Het document vertaalt onder andere klanteisen naar technische systeemeisen, en is daarmee input voor het verdere ontwerp- en ontwikkelproces. Daarnaast helpt een SRS met het managen van aspecten en raakvlakken, en daarmee dus ook veranderingen daaraan.

Een SRS vormt de basis voor een vraagspecificatie en een verificatierapport. Het gebruik van een SRS als tussenstap tot het komen van een vraagspecificatie, zorgt ervoor dat de vraagspecificatie zeer snel gemaakt kan worden.

Start en einde SRS: Het maken van een SRS start nadat een CRS is afgerond en goedgekeurd, en wordt afgerond nadat er genoeg ontwerp beslissingen zijn gemaakt en eisen zijn afgeleid om tot het vooraf gewenste specificatieniveau te komen.

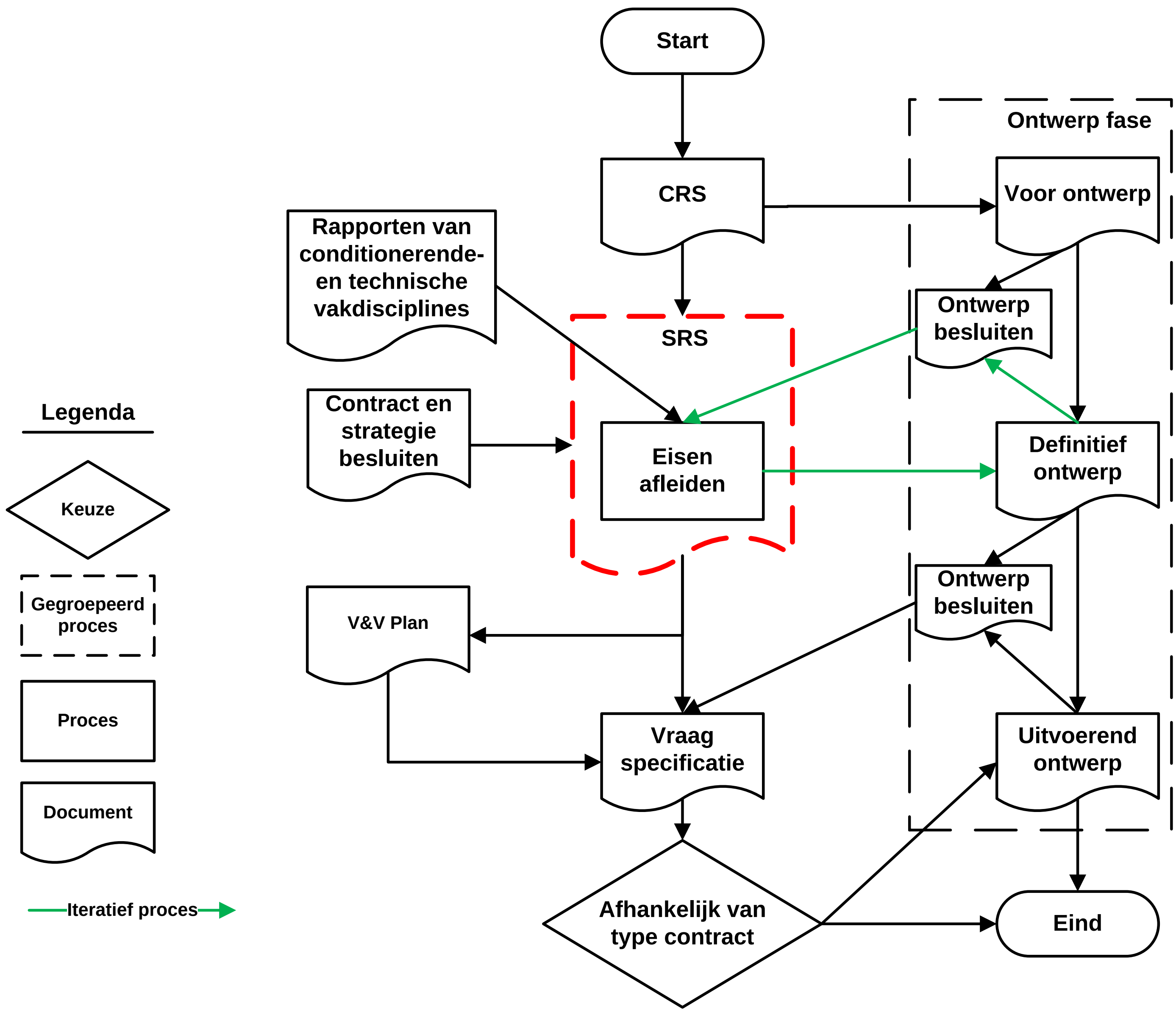
Hoe te gebruiken: Deze handleiding is zo ingericht, dat deze door alle verschillende betrokkenen bruikbaar is. Voor iedereen die een SRS maakt of eisen afleidt biedt deze handleiding een houvast door het proces heen. Mensen die geen of weinig ervaring hebben met een SRS of eisen afleiden vinden hier alles wat er nodig is over deze onderwerpen. Voor meer ervaren SRS schrijvers is deze handleiding er om hun eigen, vaste manier van werken te herzien en gemakkelijk informatie te vinden over specifieke onderwerpen of activiteiten. Per activiteit is aangegeven door welke rol deze dient te worden uitgevoerd. Eén persoon kan binnen een project meerdere rollen bekleden en andersom. Indien er verbetermogelijkheden zijn, gelieve deze door te spelen naar de afdeling Systeem Integratie.

Versie: 16-07-2017

Auteur: Arjan Laout

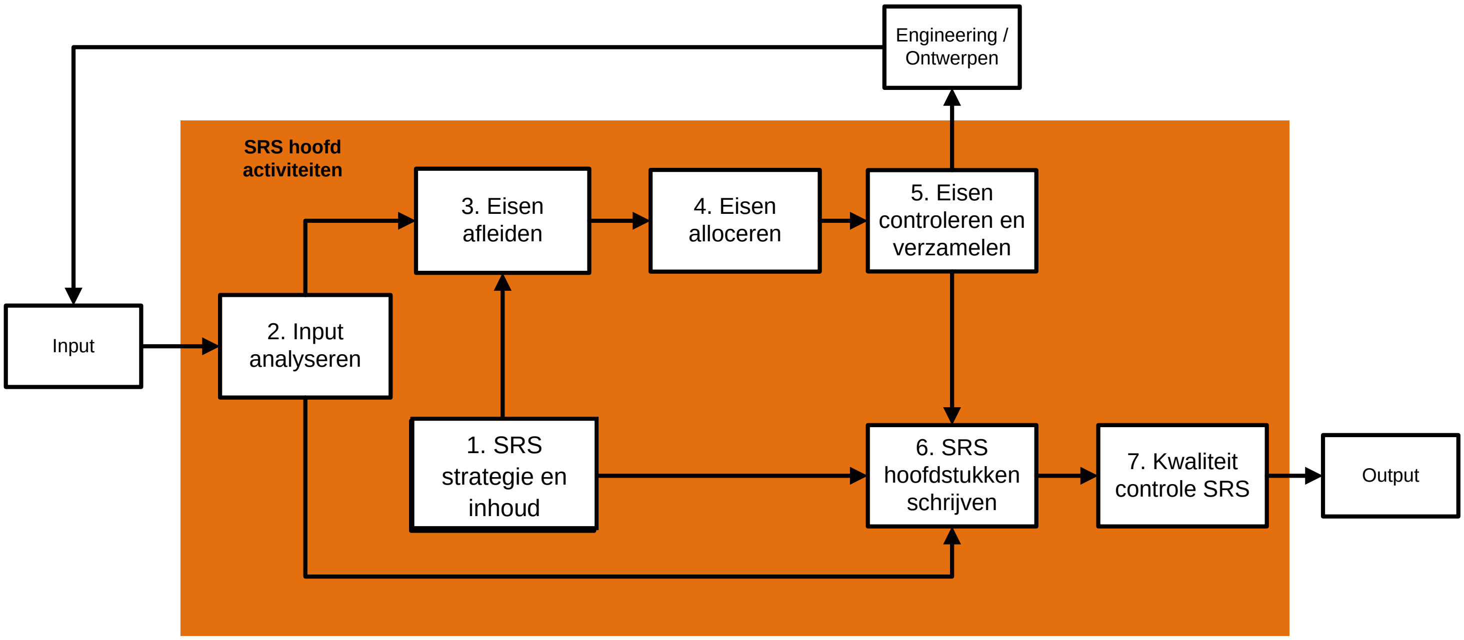
Inhoudsopgave

Context



Dit proces geeft de algemene, versimpelde weergave van de context waarin een SRS speelt. Hieruit wordt duidelijk wat input en output is voor een SRS. De iteratie met het ontwerpproces is hier ook goed zichtbaar.

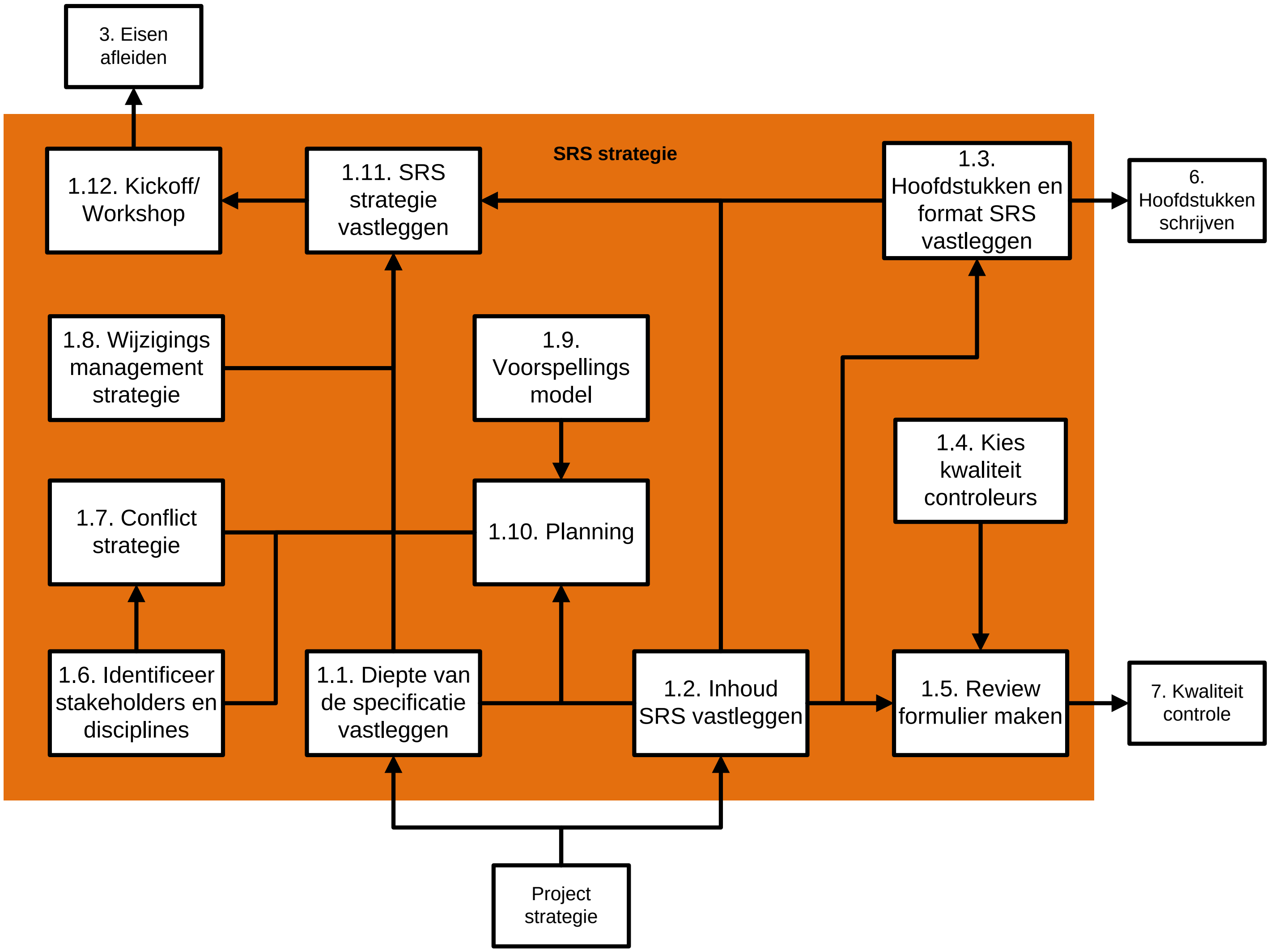
SRS Proces



Snel naar: (1) SRS Strategie & inhoud	Snel naar: (2) Input analyseren	Snel naar: (3) Eisen afleiden	Snel naar: (4) Eisen alloceren
Snel naar: (5) Eisen controleren en verzamelen	Snel naar: (6) SRS hoofdstukken schrijven	Snel naar: (7) Kwaliteit controle SRS	

SRS proces in een notendop: Dit is een versimpelde weergave van de hoofd activiteiten van een SRS proces. Allereerst wordt de inhoud en het proces van de toekomstige SRS besproken en vastgelegd. Verschillende soorten input komen op verschillende momenten binnen, deze input bestaat voornamelijk uit klanteisen, ontwerpbesluiten en conditionerende rapporten. Vervolgens worden systeemeisen afgeleid uit verschillende soorten input. Deze nieuwe eisen worden gestructureerd, verzameld en vormen weer input voor het ontwerpproces. Vanuit nieuwe ontwerpen worden weer nieuwe eisen afgeleid, Dit iteratieve proces herhaalt zich totdat het gewenste specificatieniveau bereikt is. Dan worden de eisen samen met de andere hoofdstukken in een SRS document gezet, wordt het document gecontroleerd en tenslotte als het goedgekeurd is, vrijgegeven.

1. SRS Strategie & inhoud



Voorafgaand aan het maken van een SRS, wordt één document gemaakt waarin de strategie voor de SRS is vastgelegd. Om deze strategie vast te leggen, vindt een overleg plaats. In dit overleg worden de hierboven beschreven activiteiten besproken en uitgevoerd. De inhoud van deze processen zijn zeer afhankelijk van de grootte en complexiteit van een project, zoals beschreven bij de verschillende deelactiviteiten. De RE of auteur van de SRS is uiteindelijk verantwoordelijk voor de SRS strategie.

Terug naar: SRS Proces

1.1. Diepte van de specificatie vastleggen

Wie: Requirements Engineer, Systems Engineer, Lead Engineer, Project Manager.

Wat: De diepte van de specificatie bepaalt tot welk niveau eisen worden afgeleid. Dit heeft dus te maken met de gevraagde nauwkeurigheid van het uiteindelijke ontwerp.

Waarom: Deze stap is input voor veel vervolgactiviteiten, zoals het maken van een FBS en het bepaalt wanneer er genoeg eisen zijn afgeleid.

Hoe: Het specificatie niveau kan per object verschillen, en ligt aan de behoefte van de opdrachtgever. Dit specificatieniveau dient vooraf goed bekeken te worden. Het kan namelijk zo zijn dat de opdrachtgever wilt dat op een abstracter niveau gespecificeerd wordt, zodat de opdrachtnemer vrij is om zelf met een creatieve oplossing te komen. Aan de andere kant kan zo zijn dat er bepaalde significante risico's zijn, en om die risico's in te perken dient een bepaalde functie of object verder gespecificeerd te worden.

1.2. Inhoud SRS vastleggen

Wie: Requirements Engineer, Systems Engineer, Lead Engineer, Project Manager.

Wat: In deze stap wordt vastgelegd hoe de SRS er uiteindelijk uit gaat zien.

Waarom: Vanzelfsprekend is dat de SRS een overzicht bevat van alle systeemeisen. Een SRS is echter veel breder dan dat, deze dient namelijk leesbaar te zijn als zelfstandig document voor mensen zonder kennis van het project of kennis van technische inhoud. In deze stap wordt dus ook wel de beoogde inhoud van het output document vastgelegd. Met inhoud wordt hier bedoelt het aantal elementen waar de SRS uit zal bestaan en in welke mate deze elementen worden toegepast. Deze beoogde inhoud is zeer afhankelijk van de grootte en complexiteit van een project. Een groot en complex project vereist ook een uitgebreide SRS, waar duidelijk uit blijkt hoe de eisen tot stand zijn gekomen. Hoe complexer een project, des te belangrijker de herleidbaarheid. Het risicoprofiel van een project kan helpen met het kwantificeren van de complexiteit. Waar de SRS ook zeer van afhangt is het gebudgetteerde aantal uren van de SRS. Er dient gekeken te worden naar de mogelijkheden van een SRS binnen het aantal uren.

Hoe: Per SRS wordt dus vooraf vastgesteld welke onderdelen een SRS bevat, welke attributen eisen moeten bevatten, en welke processen worden doorlopen, en in welke mate. Deze onderdelen worden nu behandeld.

Terug naar: (1) SRS Strategie & inhoud

Onderdelen SRS

De hier vastgestelde onderdelen van een SRS kunnen gebruikt worden voor het vastleggen van de hoofdstukken van een SRS. Per onderdeel dient ook de mate van toepassing worden gespecificeerd. In de volgende activiteit, *'1.3. Hoodstukken en format SRS vastleggen'* zijn deze onderdelen nog verder gespecificeerd. Uit de volgende tabel kan verwijderd worden wat niet van toepassing zal zijn bij het vaststellen van de onderdelen van de SRS.

Onderdelen van SRS	
Onderdeel:	Mate van toepassing (verwijder wat niet van toepassing is)
Inhoudsopgave	Eén pagina gereserveerd voor de inhoudsopgave.
Introductie	Achtergrondinformatie over het project, motivatie voor het project, belangrijke genomen beslissingen, een plattegrond van het systeem, satelliet/helikopter foto's. Daarnaast een introductie van de SRS, met beschrijving van het doel en de functie binnen het engineering proces.
Systeem en systeem context	Beschrijving van het doel van het project, een beschrijving van het systeem en de functionaliteit van het te realiseren systeem. Een beschrijving van de huidige en gewenste situatie. Een algemene of functionele beschrijving van de systeem grenzen. Een beschrijving van de systeem context, dus hoe het "system of interest" zich verhoudt tot andere, aangrenzende systemen (visueel weergegeven met een ster diagram en/of architectural design). Een beschrijving van de stakeholders, weergegeven in een uidiagram.
Scope	De scope is gerelateerd aan de systeem context en beschrijft de grenzen van het systeem. Er staat duidelijk uitgelegd wat binnen en wat buiten de SRS valt.
Raakvlakken	Een overzicht van de interne en externe raakvlakken. Volgt uit raakvlakken analyse. Dit is visueel weergegeven dmv een N^2 matrix.
Lijst van bron documenten	Lijst van documenten van waaruit de systeemeisen zijn afgeleid.
Ontwerp beslissingen	Een samenvatting van de belangrijke ontwerp beslissingen, of een lijst van alle documenten met ontwerp beslissingen. Hieruit kunnen namelijk eisen uit afgeleid zijn.
Eisen	Het onderdeel "eisen" is eerst tekstueel ingeleid, en de traceerbaarheid van eisen is uitgelegd (naar bron, functie en bovenliggende eis). Vervolgens volgt een gestructureerd overzicht van de eisen, per type eis (functioneel, raakvlak, aspect).
Lijst van aanpassingen	Een lijst van aanpassingen aan het systeem gedurende het project.
Verificatie & Validatie	Een uitleg over het verificatie & validatie plan (V&V plan), verificatiematrix en verificatierapport (wat aparte documenten zijn), en hoe deze zich verhouden tot een SRS. Het V&V plan is input voor deze stap.
Bijlagen	Een FBS en SBS.
Andere mogelijke onderdelen	Bijvoorbeeld een lijst met verworpen eisen.

Terug naar: (1) SRS Strategie & inhoud

Eis attributen

Het belangrijkste deel van de SRS zijn de systeem eisen. De specifieke attributen kunnen verschillen per SRS, maar moeten vooraf vastgesteld worden. Uit de onderstaande tabel kan verwijderd worden wat niet van toepassing zal zijn bij het vaststellen van de attributen van de eisen.

Attributen van de eisen	
Standaard	
ID	Uniek ID.
Naam/Titel	In één of enkele woorden.
Beschrijving	Duidelijke omschrijving van de eis, belangrijkste deel.
Bovenliggende eis	Bovenliggende eis welke de betreffende eis verder specificeert.
Brondocument	Specifieke bron van de eis. Dit kan de CRS zijn, ontwerp besluiten, etc. Er kunnen meerdere bronnen zijn.
Verificatieprocedure (V&V)	Beschrijving van hoe er gecontroleerd kan worden dat er daadwerkelijk rekening is gehouden met de eis.
Bijschrift functie	Refereert naar de functie die (deels) wordt voldaan door de betreffende eis. Dit is alleen van toepassing bij functionele eisen.
Optioneel	
Onderliggende eisen	Onderliggende eisen welke de betreffende eis verder specificeren.
Eis initiator	De actor waar de eis vandaan komt, zoals een stakeholder.
Prioriteit	Twee niveaus: 1) Het niet nakomen van deze eis bedreigt de prestatie van het hele systeem 2) Eis is belangrijk, maar het niet nakomen van deze eis zal het functioneren van het systeem niet bedreigen.
Toelichting	Extra toelichting bij een eis.
Afgeleid van klant eis (ID)	Refereert naar de specifieke CRS-eis waar de betreffende SRS-eis van is afgeleid.
Aspect	In het geval van aspect eisen, kan het specifieke aspect genoemd worden.
Raakvlak	In het geval van raakvlak eisen, refereert naar het specifieke raakvlak.

Terug naar: (1) SRS Strategie & inhoud

Uit te voeren activiteiten

Niet alleen de inhoud van een SRS wordt vastgelegd, maar ook het SRS proces, oftewel de te doorlopen stappen. Bijvoorbeeld, het maken van een '*1.7. conflict strategie*' of '*1.8. wijzigingsmanagement strategie*' is niet altijd nodig. Het kan ook zijn dat deze stappen wel nodig zijn, maar relatief oppervlakkig kunnen worden uitgevoerd. In de volgende tabel dient per activiteit de mate van uitvoering ingevuld te worden. Dit is input voor de volgende stap, '*1.3. hoofdstukken en format SRS vastleggen*'.

SRS activiteiten		
Activiteit	Mate van uitvoeren van de activiteit uit zich als volgt	Mate van uitvoeren (niet, simpel, goed)
Verificatie procedure	Mate van detail vastleggen verificatie procedure en uitleg van verificatieprocedure in de SRS	
Eisen prioriteren	Per eis vastgelegd prioriteit 1 of 2	
Conflict strategie	Onderdeel van strategie	
Verander management strategie	Onderdeel van strategie	
Intern SRS reviewen	Aantal interne reviewers, detail van reviewen	
Extern reviewen (door opdrachtgever)	Wel of niet reviewen door de opdrachtgever zelf	
FBS/SBS maken	Een FBS dient altijd gemaakt te worden, een SBS wordt bij voorkeur ook gemaakt	
Overige activiteiten	Overige activiteiten kunnen hier verder gespecificeerd worden.	

1.3. Hoofdstukken en format SRS vastleggen

Wie: Requirements Engineer.

Wat/Waarom: Sommige opdrachtgevers leveren een format voor de SRS. De format van een SRS kan ofwel de format aannemen van Movares, ofwel van de opdrachtgever. Ook een format van een opdrachtgever, kan aangevuld worden met extra normen die binnen Movares gehanteerd worden. Per project kunnen de hoofdstukken iets verschillen, maar om een Movares standaard vast te leggen, zijn de hoofdstukken vastgelegd waar de ideale SRS uit bestaat. Per SRS dient vooraf te worden vastgelegd welke hoofdstukken en onderdelen daadwerkelijk terug komen in de SRS en later verder worden uitgewerkt.

Hoe: De input voor deze keuze bestaat uit de '*onderdelen SRS*' die in de vorige deelactiviteit '*1.2. inhoud SRS vastleggen*' zijn vastgelegd. Movares, maar ook bedrijven zoals Rijkswaterstaat en ProRail hebben een eigen format vastgelegd:
Movares:
https://mint1.movares.nl/communities/Systeem%20Integratie/_layouts/15/WopiFrame.aspx?sourcedoc=/communities/Systeem%20Integratie/Bibliotheekdocumenten/Templates/SRS/Template%20SRS.docx&action=default

ProRail: http://pwredirect/open/D/3f217375-db19-41bd-9dee-fc0d75f16726/Documentsjabloon System Requirements Specification.docx

Terug naar: (1) SRS Strategie & inhoud

Titelblad

Titel, soort document, status document, auteur, datum, versie, opdrachtgever.

- **Inhoudsopgave**

1. **Introductie**
    - *Achtergrond informatie over het project, project geschiedenis, motivatie voor het project, belangrijke genomen beslissingen.*
    - *Een plattegrond van het systeem, satelliet/helikopter foto's.*
    - *Een introductie van de SRS, met beschrijving van het doel en de functie binnen het engineering proces.*
2. **Systeem en Systeem context**
    - *Beschrijving van het doel van het project, een beschrijving van het systeem en de functionaliteit van het te realiseren systeem.*
    - *Een beschrijving van de huidige en gewenste situatie"*
        - *De huidige situatie beschrijft het gebied binnen de systeemgrenzen en het directe gebied buiten de grenzen, de huidige situatie met betrekking tot objecten van derden en infrastructuur in het gebied, landeigenaren en gebruikers. Daarnaast zijn eventuele bestemmingsplannen voor het gebied en andere relevante aspecten rond de systeemgrenzen zoals toegangswegen bondig beschreven.*
        - *De gewenste situatie refereert naar de algemene richtlijnen van de CRS (klanteisen specificatie) en de in de huidige situatie genoemde aspecten in relatie tot de toekomst.*
    - *Een algemene en functionele beschrijving van de systeem grenzen.*
    - *Een beschrijving van de systeem context, dus hoe het "system of interest" zich verhoudt tot andere, aangrenzende systemen. Een systeem is altijd onderdeel van een groter systeem. Daarnaast is dit bij voorkeur visueel weergegeven met behulp van een ster diagram en/of Architectural Design (AD). Dergelijke visualisaties kunnen ook in de bijlage geplaatst worden indien deze te groot zijn.*
    - *Een beschrijving van de stakeholders van het project, weergegeven in een uidiagram. Dit onderdeel is een samenvatting van de stakeholder analyse.*
3. **Scope**
    - *De scope is gerelateerd aan de systeem context en beschrijft de grenzen van het systeem. Er staat duidelijk uitgelegd wat binnen en wat buiten de SRS valt. Bij kleinere eisen specificaties kan dit onderdeel eventueel worden opgenomen onder het hoofdstuk systeem en systeem context.*
4. **Raakvlakken**
    - *Een overzicht van de interne en externe raakvlakken. Dit onderdeel volgt uit een raakvlakken analyse en is bij voorkeur is visueel weergegeven door middel van een N^2 matrix.*
5. **Ontwerp besluiten**
    - *Een samenvatting van de belangrijke ontwerp beslissingen, overzichtelijk weergegeven in een lijst van belangrijke ontwerp beslissingen, eventueel met brondocumenten. Hieruit kunnen namelijk eisen uit afgeleid zijn.*
    - *De lijst is eerst tekstueel ingeleid, met een verklarend model hoe eisen, besluiten, systemen en verificatie met elkaar samenhangen.*

Terug naar: (1) SRS Strategie & inhoud

- *De lijst van ontwerpbesluiten bestaat minimaal uit drie kolommen met daarin ID, beschrijving en bron.*

6. **Lijst van brondocumenten**
   - *Lijst van documenten van waaruit de systeemeisen zijn afgeleid en waarnaar de betreffende eisen refereren. Bij voorkeur weergegeven is deze lijst weergegeven in tabelvorm.*
7. **Eisen**
   - *Het onderdeel "eisen" is eerst tekstueel ingeleid, de structuur is uitgelegd (de verschillende type eisen), en de traceerbaarheid van eisen is uitgelegd (naar bron, functie en bovenliggende eis).*
   - *Vervolgens volgt het belangrijkste deel van een SRS, namelijk een gestructureerd overzicht van de eisen, per type eis (per functie, raakvlak, aspect en realisatie).*
8. **Verificatie & Validatie**
   - *Een uitleg geven over het verificatie & validatie plan (V&V plan), verificatiematrix en verificatierapport (wat aparte documenten zijn), en hoe deze zich verhouden tot een SRS. Eén van de attributen van de eisen is een verificatieprocedure. In de verificatieprocedure staat per eis kort uitgelegd hoe deze gecontroleerd dient te worden op de naleving van de eis in de uiteindelijke ontwerpen en realisatie. Een verificatiematrix is een overzicht van de eisen en verificatieprocedures, en in een verificatierapport staan vervolgens de resultaten van de controles op de verificatieprocedures. Het V&V plan is input voor dit onderdeel. De inhoud van deze aparte documenten dienen hier niet besproken te worden.*
9. **Bijlagen**
   - **Definities en afkortingen**

Een lijst van gebruikt vakjargon en afkortingen. Vanuit de introductie dient naar deze lijst verwezen te worden.

- **FBS**

Visualisatie van de FBS

- **SBS**

Visualisatie van de SBS

- **Lijst van aanpassingen**

Overzicht van scope wijzigingen, belangrijke eisen, en dergelijke die gewijzigd zijn.

1.4. Kies kwaliteit controleurs

Wie: Requirements Engineer.

Wat/Waarom: Voorafgaand aan het maken van een SRS, dient al te worden vast gelegd wie de SRS uiteindelijk gaan reviewen.

Hoe: Op Mint in het KVM-plein is een lijst te vinden van medewerkers die een review uit mogen uitvoeren per vakgebied. Bij een kleine SRS volstaat één reviewer, bij grotere, complexere SRS'en dienen meerdere mensen het document te controleren. Bij voorkeur worden reviewers gebruikt met een multidisciplinaire achtergrond, of meerdere mensen met verschillende achtergronden.

Terug naar: (1) SRS Strategie & inhoud

1.5. Review formulier maken

Wie: Requirements Engineer.

Wat: Nadat is vastgelegd hoe de beoogde SRS er uiteindelijk uit gaat zien, kan het interne review formulier worden opgesteld.

Waarom: Het doel hiervan is om de vooraf beoogde kwaliteit te waarborgen en achteraf te controleren in welke mate de verschillende aspecten van een SRS zijn nageleefd.

Hoe: Hiervoor is een standaard review formulier opgesteld. In dit formulier kan worden aangegeven aan welke aspecten de SRS dient te voldoen. Dit review formulier dient te worden opgeslagen zodat het makkelijk terug te vinden is op het moment van reviewen, wat vele maanden later kan zijn. Een geschikte locatie hiervoor is in de SRS map in ProjectWise van het betreffende project. Achteraf kan dan per aspect de mate van voldoening worden aangegeven door de reviewer. Review formulier: https://mint1.movares.nl/communities/Systeem%20Integratie/_layouts/15/WopiFrame.aspx?sourcedoc=/communities/Systeem%20Integratie/Bibliotheekdocumenten/Templates/SRS/SRS%20kwaliteit%20controle%20formulier.xlsx&action=default

1.6. Identificeer actoren en disciplines

Wie: Requirements Engineer en/of Systems Engineer.

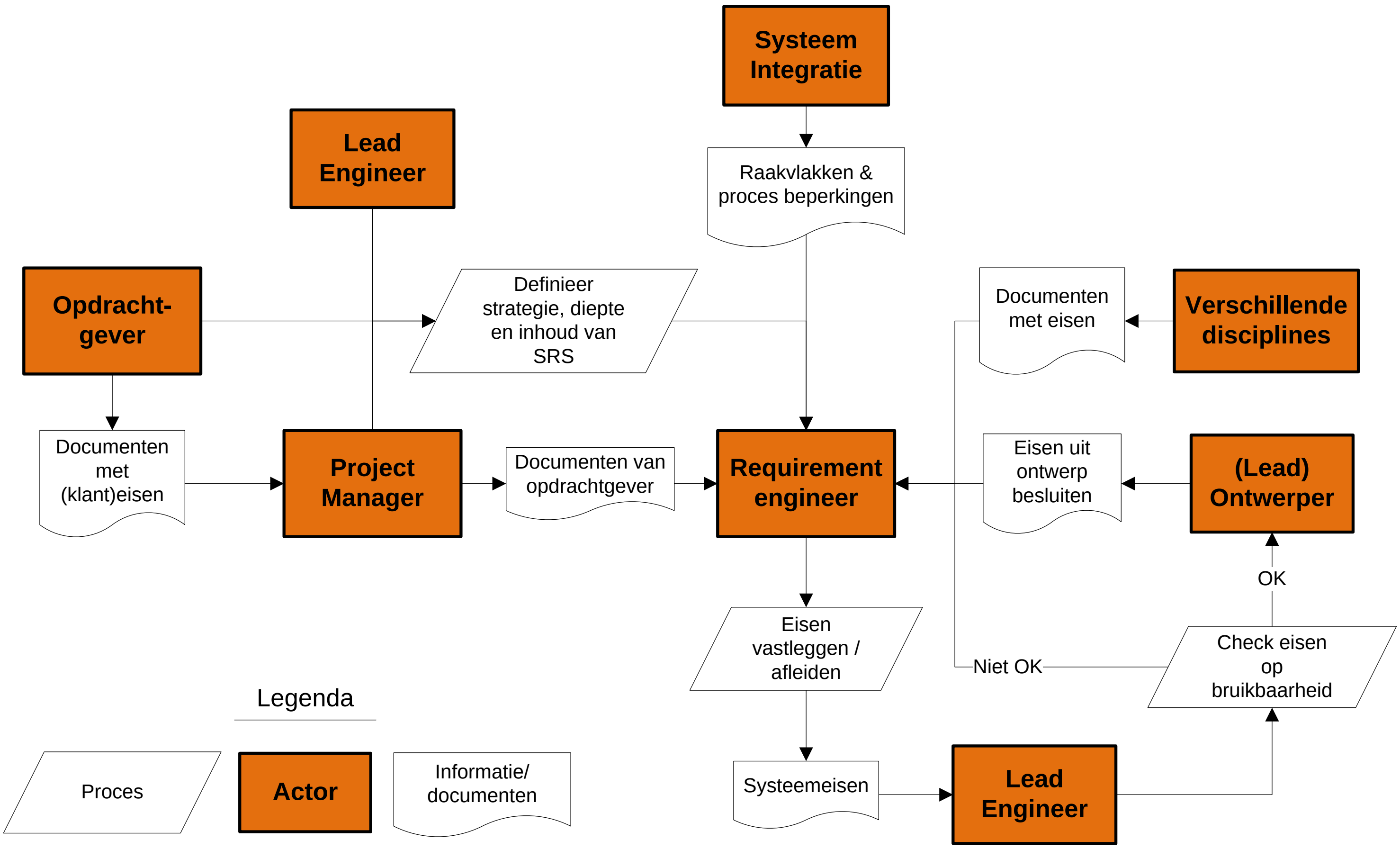
Wat/Waarom: Het identificeren van stakeholders en disciplines dient te worden gedaan voordat een conflict strategie een verander managementstrategie worden opgesteld. Met stakeholders en disciplines wordt bedoeld; Diegenen die input leveren aan de SRS of invloed hebben op het SRS proces.

Hoe: Een lijst met mogelijke relevante actoren:

- Opdrachtgever
- Gemeenten
- Provincies
- Omwonenden
- Projectmanager
- Omgevingsmanager
- Lead engineer
- (Lead) ontwerper
- Verschillende technische vakdisciplines
- Conditionerende disciplines

Deze lijst kan per project aangepast en aangevuld worden. Voor meer inspiratie is het volgende actoren diagram gegeven:

Terug naar: (1) SRS Strategie & inhoud



1.7. Conflict strategie opstellen

Wie: Requirements Engineer.

Wat: De conflict strategie is onderdeel van het document dat uiteindelijk de SRS strategie omvat. Het maken van een conflict strategie is zoals eerder genoemd niet altijd nodig, maar meestal wel. Slechts bij kleine SRS'en met een laag risicoprofiel is het niet nodig een conflict beleid op te stellen. In het geval van middelgrote projecten/SRS'en hoeft dit geen uitgebreid plan te zijn. Hoe uitgebreid een conflict strategie is, is afhankelijk van het aantal geïdentificeerde (interne en externe) stakeholders en risico's. Bij meer stakeholders en risico's is het project ingewikkelder, en is er een grotere kans op conflicten. Daarnaast blijft het altijd een afweging tussen de kosten en baten, en dus hoeveel tijd er in een conflict strategie wordt gestoken. Het kan zijn dat een uur al voldoende is. Bij grotere en ingewikkeldere projecten mag dit wel wat uitgebreider.

Waarom: Bij de meeste projecten zijn problemen met zowel interne als externe stakeholders onvermijdelijk. Met problemen wordt bedoeld; het veranderen van de scope, meerwerk, of andere veranderingen. Dan is het heel handig om hier vooraf over nagedacht te hebben en een conflictbeleid te hebben om op terug te vallen.

Terug naar: (1) SRS Strategie & inhoud

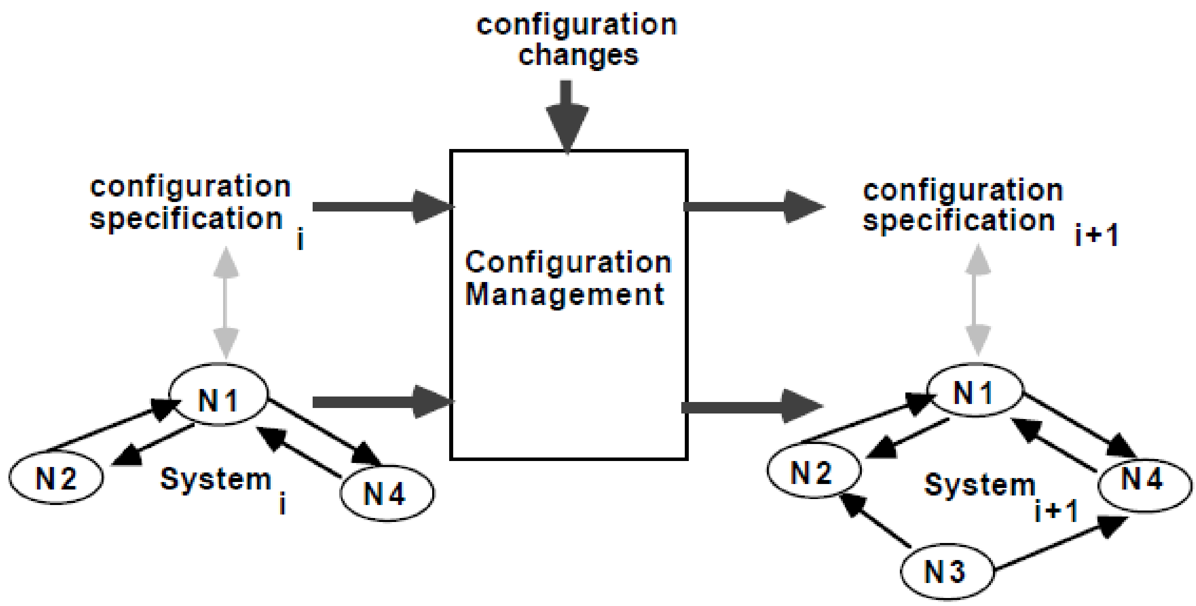
Hoe: Voor de conflict strategie dient er gebrainstormd te worden naar mogelijke risico's voor de SRS. Per actor en per discipline die al zijn geïdentificeerd, kunnen risico's worden opgesteld. Bedenk per risico hoe dit voorkomen kan worden en wat er moet gebeuren als dit probleem daadwerkelijk optreed. Mogelijke risico's zijn bijvoorbeeld het te laat leveren van input, of conflicterende raakvlakken.

1.8. Wijzigingsmanagement strategie opstellen

Wie: Requirements Engineer en Systems Engineer.

Waarom: Gedurende het project zullen er altijd dingen veranderen, daarom is het ook voor dit onderwerp verstandig om hier op voorhand over na te denken.

Wat: Wijzigingen die kunnen optreden zijn bijvoorbeeld wijzigingen van eisen, nieuwe eisen die worden toegevoegd aan het systeem en scope wijzigingen. Bij dergelijke veranderingen kan configuratie management helpen, zoals visueel is weergegeven in de volgende figuur:



Bron: Kramer, J., & Magee, J. (1990). The Evolving Philosophers Problem: Dynamic Change Mangement. *IEEE Transactions on Software Engineering* , p. 1293-1306.

Bij veranderingen is het belangrijk rekening te houden met de configuratie van het systeem en de verschillende raakvlakken tussen de deelsystemen. Als verschillende vakdisciplines verschillende deelsystemen ontwerpen, dienen deze op bepaalde momenten afgestemd te worden op elkaar om raakvlak conflicten te voorkomen. Dit zijn de zogenoemde baselines of baseline momenten.

Hoe: Op voorhand dient te worden vastgelegd op welke momenten een baseline ligt. De details van een dergelijk baseline moment kunnen worden uitgewerkt of uitgedacht als het betreffende moment nadert.

Daarnaast dient te worden vastgelegd wat er gedaan wordt met oude of verworpen eisen. Het is slim deze te bewaren, aangezien deze later wellicht worden hergebruikt, wat tijd en geld scheelt. Daarnaast geven deze eisen weer waar bewust niet voor gekozen is.

Terug naar: SRS Strategie & inhoud

1.9. Voorspellingsmodel

Wie: Requirements Engineer.

Wat: Aan het voorspellingsmodel wordt nog gewerkt, maar gaat in de toekomst gebruikt worden. Het voorspellingsmodel bestaat uit een aantal parameters die ingevuld dienen te worden. Hieruit komt op basis van historische cijfers een voorspelling naar voren met betrekking tot het aantal uur en de doorlooptijd van de SRS.

Waarom: Dit helpt met het maken van een planning en daarnaast het instellen op risico's als blijkt dat er in het verleden met soortgelijke projecten veel vertraging of rework was.

1.10. Planning

Wie: Requirements Engineer.

Wat: Onderdeel van de strategie is het maken van een planning. Als duidelijk is hoe nauwkeurig er gespecificeerd gaat worden en de risico's in kaart zijn gebracht, kan een planning worden gemaakt met behulp van het voorspellingsmodel.

1.11. SRS strategie vastleggen

Wie: Requirements Engineer in overleg met Lead Engineer en Project Manager.

Wat: Per project wordt een document gemaakt waarin de SRS strategie vastgelegd is voor intern gebruik. De opzet is dan ook functioneel.

Waarom: De strategie fungeert als referentiekader voor het gehele SRS proces. Aangezien het SRS proces vele maanden kan duren, is het verstandig een document te hebben waar de strategie en de te doorlopen stappen zijn beschreven en waarop terug gevallen kan worden bij complicaties.

Hoe: De hoofdstukken van dit document bestaan uit de activiteiten die input vormen activiteit '*SRS strategie*'. Ook hier geldt, hoe ingewikkelder het project, des te uitgebreider het document. De SRS strategie bestaat uit de volgende hoofdstukken:

- Inhoud SRS
    - Onderdelen SRS
    - Eis attributen
    - Uit te voeren activiteiten
- Diepte van de specificatie
- Actoren en disciplines
- Conflict strategie
- Wijzigingsmanagement strategie
- Planning

Terug naar: (1) SRS Strategie & inhoud

1.12. Kick-off / Workshop

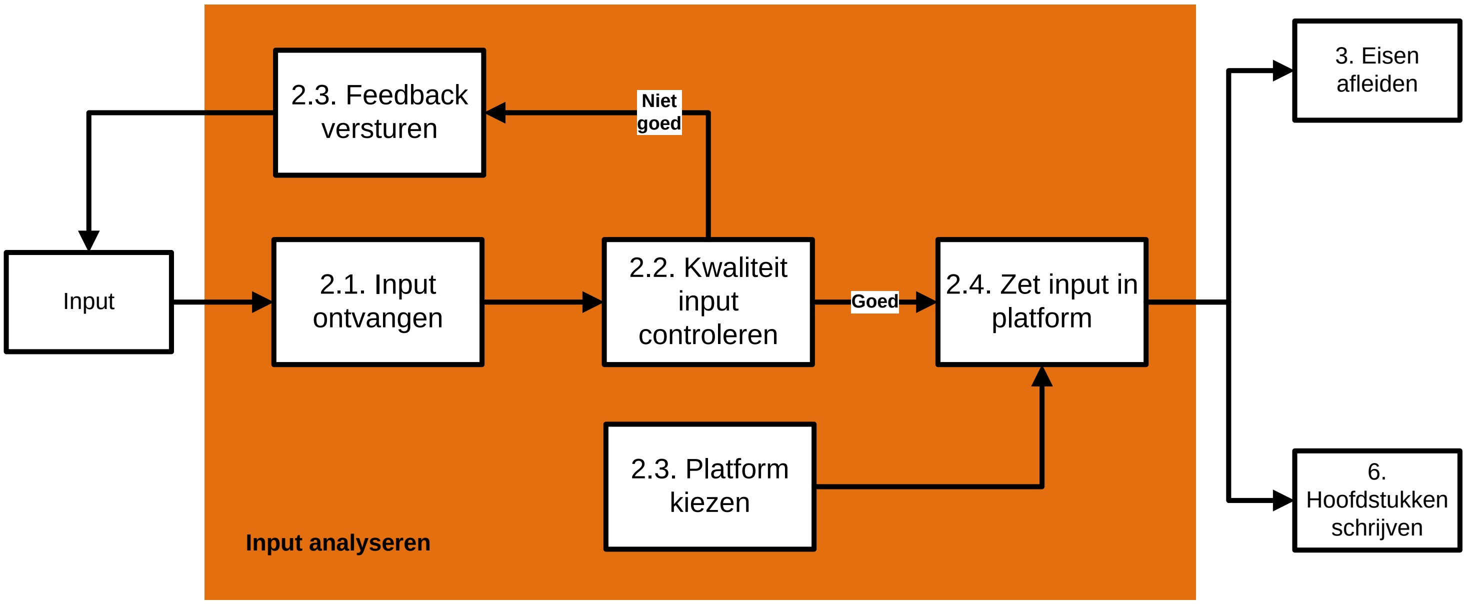
Wie: Requirements Engineer, Lead Engineer, betrokken actoren.

Wat/Hoe: In het geval van een groter of complex project, dient een SRS zijn eigen kick-off te hebben met de betrokken actoren die zijn gedefinieerd in deelactiviteit '*1.6. Identificeer actoren en disciplines*'. Nadat een strategie is vastgelegd en duidelijk is wat de bedoeling is, dienen de betrokken actoren hier gezamenlijk van op te hoogte gesteld te worden, waar ook ruimte is voor discussie. Aansluitend aan de kick-off kunnen indien nodig workshops requirements engineering of systems engineering gehouden worden.

Waarom: Bij grotere en complexere projecten loont het om eenmalig in een dergelijke activiteit te investeren, om iedereen op één lijn te krijgen en zodat men weet wat er van hen verwacht wordt. De workshops systems engineering of requirements engineering zijn nodig om ervoor te zorgen dat diegenen die eisen afleiden en/of input bijdragen aan het afleiden van eisen, zorgen dat de kwaliteit van de input en de eisen zelf goed is.

Terug naar: (1) SRS Strategie & inhoud

2. Input analyseren



Een SRS is een complex document dat input krijgt van veel verschillende actoren binnen het project. In deze hoofdactiviteit wordt deze input gecontroleerd en beheerd.

Terug naar: SRS Proces

2.1. Input ontvangen

Wie: Requirements Engineer ontvangt input.

Wat: Er dienen verschillende soorten input en bronnen van eisen te worden herkend en onderscheiden, deze zijn voornamelijk:

- Systeemarchitectuur
- Risico's
- Raakvlakken analyses
- Gemaakte ontwerpbesluiten
- CRS (klant eisen)
- FBS
- Rapporten en eisen vanuit conditionerende disciplines
- Eisen die vergunningverlenende partijen en overige recht- en belanghebbenden stellen

Het kan input betreffen waar eisen uit worden afgeleid, of input dat ondersteunend is voor het afleiden van eisen en het SRS proces, zoals een SBS en architectural design (AD).

Waarom: Indien het te lang dreigt te duren voordat bepaalde actoren input aanleveren, dienen zij daarbij structureel begeleid te worden, om zo het aanleveren te bespoedigen. Indien het lastig blijft goede input te ontvangen van een bepaalde actor, kan tot andere keuzes worden over gegaan, zoals het passeren van de betreffende actor.

De volgende stap betreft het controleren van de kwaliteit van input. Probeer al vooraf te sturen op de gewenste te leveren kwaliteit van de actor waar de input vandaan komt.

2.2. Kwaliteit input controleren

Wie: Requirements Engineer.

Wat/Waarom: Op het moment dat input documenten zijn ontvangen, dient het gecontroleerd te worden op kwaliteit, rekening houdend met het doel waar de input voor gebruikt gaat worden.

Hoe: Er zijn over het algemeen twee soorten input: 1) input waaruit systeemeisen worden afgeleid en 2) input dat ondersteunend is aan het SRS proces. Deze worden hier apart behandeld.

Input waaruit systeemeisen worden afgeleid

Wat: De perfecte input bestaat uit een lijst van eisen die reeds voldoen aan de eisspecifieke eisen. Dit is een lijst van eisen waar een gespecificeerde systeemeis uiteindelijk aan dient te voldoen. Deze lijst is te vinden onder de deelactiviteit *'4.1. Controleer op eisen aan eisen'*.

Waarom: Eén van de doelen van dit document is dat de betrokken interne stakeholders zoals technische- en conditionerende disciplines die input aanleveren waaruit eisen worden afgeleid, al zelf de eisen afleiden. Het afleiden van systeemeisen vereist namelijk vaak (technisch) inhoudelijke kennis, wat de requirements engineer meestal niet beheerst, of in ieder geval in mindere mate dan de actor waar de input vandaan komt. Hiermee wordt voorkomen dat een requirements engineer voor het verzamelen en afleiden van eisen zelf veel tijd kwijt is. Dit mag dan ook verlangd worden van diegenen die input aanleveren. Hierbij is het belangrijk de input leverende actoren eerst te overtuigen van het belang en de noodzaak van een eisen document, en dat het proces het afleiden en opstellen van systeemeisen helpt

Terug naar: (2)
Input analyseren

het ontwerpproces te ondersteunen. Onder de hoofdactiviteit *´3.1. eisen afleiden´* wordt in detail uitgelegd hoe systeemeisen dienen te worden afgeleid.

Hoe: Indien het makkelijker of noodzakelijk is dat de requirements engineer zelf eisen afleidt uit de geleverde input, dient de geleverde input van voldoende kwaliteit te zijn. Hierbij wordt gelet op:

- Duidelijkheid: De input roept geen vragen op.
- Eenvoud: Systeemeisen dienen gemakkelijk af te leiden zijn uit de input, zonder veel benodigde vakinhoudelijke kennis.
- Traceerbaarheid: Eisen dienen traceerbaar te zijn naar een brondocument.
- Functie: Eisen dienen traceerbaar te zijn naar een functie, aspect of raakvlak.
- Prioriteit: Het is duidelijk welke eisen het meest belangrijk zijn boven anderen, ten behoeve van de later te volgen deelactiviteit *´3.4. eisen prioriteren´*.

Indien de kwaliteit van de input niet voldoende geschikt is om eisen uit af te leiden van hoge kwaliteit, is de volgende activiteit het verzenden van feedback naar de betreffende actor. Ook als er wel eisen kunnen worden afgeleid, maar de kwaliteit van de input beter kan, dient er feedback in de vorm van opbouwende kritiek verzonden te worden naar de betreffende actor.

SRS ondersteunende input

Wat/Hoe: Het tweede soort input is input dat ondersteunend is aan het SRS proces. Hierbij kan gedacht worden aan bijvoorbeeld een SBS en architectural design (AD). Hierbij dient subjectief gekeken te worden naar de duidelijkheid en de geschiktheid voor het doel waar het voor gebruikt wordt.

2.3. Feedback versturen

Wie: Requirements Engineer.

Waarom: Zoals aangegeven in de deelactiviteit '*2.2. kwaliteit input controleren*', mag een bepaalde kwaliteit verwacht worden van de input die wordt aangeleverd.

Wat/Hoe: Indien de kwaliteit van de input niet voldoende geschikt is om eisen uit af te leiden van hoge kwaliteit, is de volgende activiteit het verzenden van feedback naar de betreffende actor. Ook als er wel eisen kunnen worden afgeleid, maar de kwaliteit van de input beter kan, dient er feedback in de vorm van opbouwende kritiek verzonden te worden naar de betreffende actor.

2.4. Platform kiezen

Wie: Requirements Engineer, Lead Engineer.

Wat: Deze activiteit betreft het kiezen van het platform waarin de eisen en input documenten worden beheerd.

Waarom/Hoe: De alternatieven zijn over het algemeen Relatics, MS Excel en MS Word. Dit is ook gelijk de volgorde van de voorkeur. Bij voorkeur is Relatics altijd het platform van waaruit eisen en bronnen beheerst worden. De reden hiervoor is dat Relatics als tool hier speciaal voor is ingericht, en dan ook een aantal voordelen biedt ten opzichte van de alternatieven. Dit is bijvoorbeeld het koppelen van eisen aan bronnen en functies, wat ook veel tijd bespaart bij het verwerken van wijzigingen, het werken met formats voor zowel eisen als de SRS zelf en communicatie met de opdrachtgever.

Terug naar: (2) Input analyseren

Er zijn uitzonderingen dat Relatics niet gebruikt wordt, en dat is voornamelijk het geval wanneer betrokken actoren niet voldoende bekend zijn met Relatics en het een zeer kleine en simpele SRS betreft. In dat geval kunnen eisen nog wel beheerst worden in MS Excel, aangezien in MS Excel meer mogelijkheden zijn qua bewerking, dan bij MS Word.

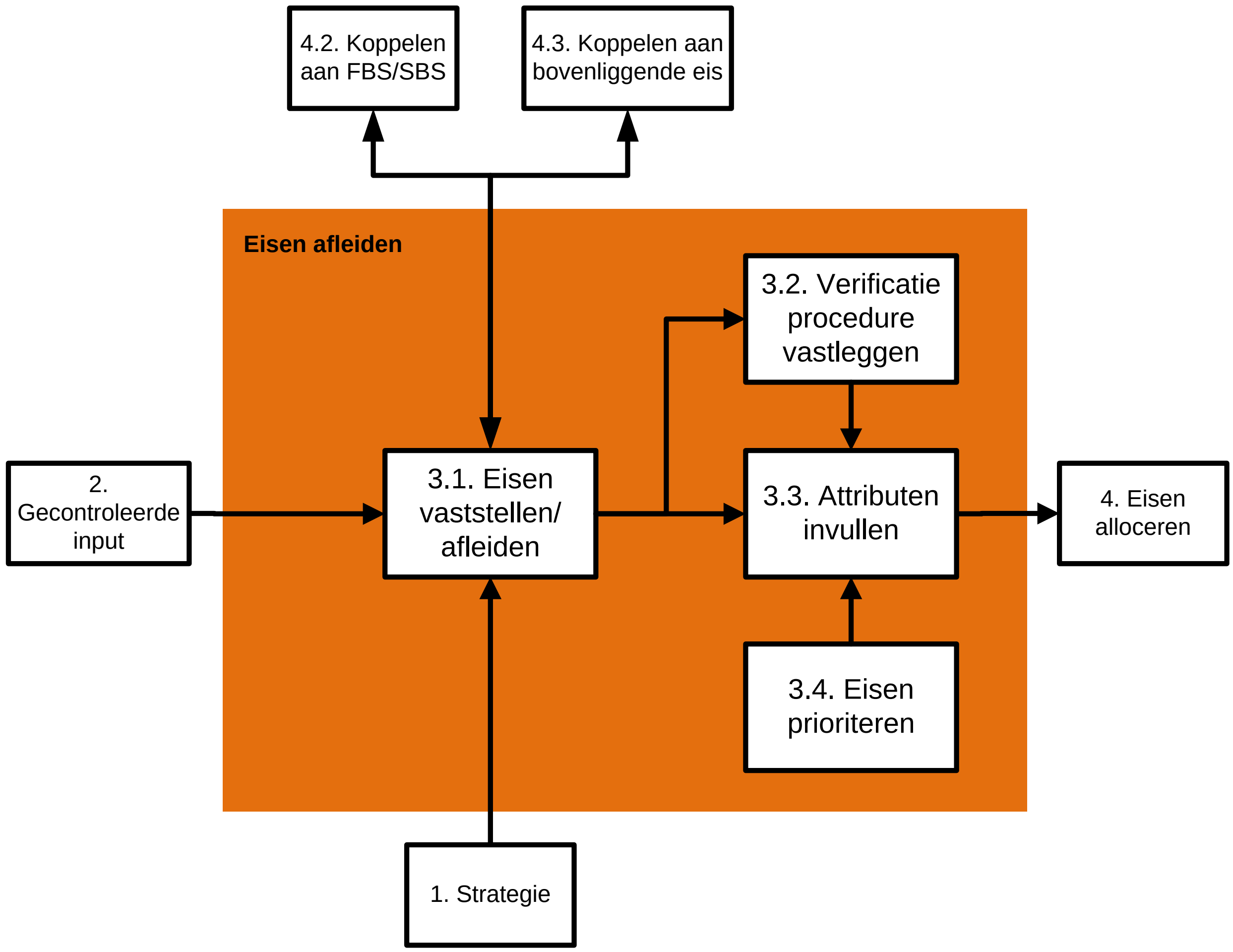
2.5. Zet input in platform

Wie: Requirements Engineer.

Wat/Hoe: Bij voorkeur worden nieuwe eisen en andere input zoals bron documenten en een FBS direct door input leverende actoren zelf in Relatics verwerkt. Als dit niet mogelijk is, worden eisen bij voorkeur in MS Excel geleverd, eventueel met tekstuele uitleg, voordat deze door de requirements engineer in Relatics worden geplaatst.

Terug naar: (2) Input analyseren

3. Eisen afleiden



Deze hoofdactiviteit is één van de meest ingewikkelde deelactiviteiten, wat ook terug te zien is in de verschillende soorten input en output. In deze hoofdactiviteit wordt de inhoudelijke invulling gegeven aan de eisen. Daarom dient deze hoofdactiviteit al zo goed mogelijk uitgevoerd te worden door de actor waar de input van afkomstig is. De requirements engineer heeft vooral een begeleidende en sturende rol in dit proces.

Terug naar: SRS Proces

3.1. Eisen vaststellen/afleiden

<u>Wie:</u> Dit ligt aan de input van de eis en staat beschreven onder “Hoe”.

<u>Wat:</u> Concreet het afleiden en/of vaststellen van de systeemeisen zelf.

<u>Waarom:</u> De eisen in een SRS zijn technische product (systeem) eisen, waaraan het uiteindelijke systeem dient te voldoen. Het afleiden of vastleggen van deze eisen is de belangrijkste en meest tijdrovende stap van het SRS proces, het is daarom ook belangrijk dat dit goed en efficiënt gebeurt.

<u>Hoe:</u> Er zijn verschillende manieren waarop een eis wordt afgeleid, welke hieronder besproken worden:

- Vanuit technische vakdisciplines: Vanuit deze disciplines worden bijvoorbeeld veel ontwerp besluiten en technische tekeningen gemaakt. De technische vakdisciplines dienen zelf al eisen op te stellen aan de hand van de gemaakte ontwerpen en ontwerpbesluiten, aangezien zij de beste vakinhoudelijke kennis hebben, ook met betrekking tot bijvoorbeeld wet- en regelgeving. Bij ontwerpkeuzes wordt een bepaald systeem verder gespecificeerd in één of meerdere deelsystemen. Hierbij wordt de SBS verder uitgebreid, en een FBS vaak ook. Aan deze nieuwe deelsystemen dienen weer nieuwe eisen gesteld te worden. Dit kunnen eisen zijn die afgeleid zijn uit één of meerdere functies van het deelsysteem, wet- en regelgeving, bestaande en extra raakvlakken die ontstaan, etc.
- Vanuit conditionerende disciplines: Vanuit deze disciplines worden onderzoeken uitgevoerd en rapporten geleverd van waaruit rekening gehouden moet worden bij het ontwerpen van het te realiseren systeem. Eisen worden afgeleid uit de bevindingen van dergelijke rapporten. De conditionerende disciplines dienen aan de hand van hun onderzoek al zelf eisen op te stellen, aangezien zij de beste inhoudelijke kennis hebben, ook met betrekking tot bijvoorbeeld wet- en regelgeving.
- Vanuit de CRS: Klanteisen en systeemeisen zijn twee verschillende soorten eisen. Klanteisen geven de behoeftes van stakeholders weer, systeemeisen vertalen deze eisen naar het technische eisen aan het systeem. In sommige gevallen kunnen (een deel van de) eisen één op één gekopieerd worden naar de systeem eisen specificatie, maar lang niet altijd. Daarom dient er per klanteis nagedacht te worden over de locatie en rol binnen de SRS. In dit geval worden de systeemeisen afgeleid door de Requirements Engineer.
- Vanuit bovenliggende eis: Een eis kan verder opgesplitst of gespecificeerd worden in één of meerdere onderliggende eisen. Vooraf in de strategie is bepaald hoe ver bepaalde systemen en eisen worden door gespecificeerd. In dit geval worden de systeemeisen afgeleid door de Requirements Engineer.
- Vanuit FBS: Een FBS is een structuur van de verschillende functies waar een systeem aan dient te voldoen. Over het algemeen zijn de meeste eisen te relateren aan eisen in de FBS, en kunnen eisen ook afgeleid worden vanuit de FBS. In dit geval worden de systeemeisen afgeleid door de Requirements Engineer.
- Vanuit aspecten: Er zijn vier verschillende soorten aspecteisen, betrouwbaarheid (R), beschikbaarheid (A), instandhouding (M) en veiligheid (S). Vanuit deze aspecten dienen aparte eisen afgeleid te worden. In dit geval worden de systeemeisen afgeleid door de Requirements Engineer.
- Vanuit raakvlakken: Elk systeem heeft zowel interne (tussen disciplines) als externe (tussen systemen) raakvlakken. Deze raakvlakken worden geïdentificeerd in een raakvlakken analyse. Gedurende het project kunnen raakvlakken veranderen. In dit geval worden de systeemeisen afgeleid door de Systems Engineer.

Terug naar: (3) Eisen afleiden

Actoren die input leveren dienen dus al zo goed als mogelijk eisen af te leiden, specificeren en relevante attributen in te vullen, aangezien zij over de meeste kennis beschikken met betrekking tot die input. De requirements engineer (RE), dus degene die de SRS schrijft en beheert, geeft hier een begeleidende en sturende rol. Het is dus niet gewenst dat de RE zelf eisen afleidt als diegene geen inhoudelijke kennis heeft over het onderwerp. De RE is wel meestal zelf de initiator van veel andere systeemeisen, waar minder inhoudelijke kennis voor nodig is.

Een eis dient uiteindelijk altijd te voldoen aan de eisenspecifieke eisen, of eisen aan eisen. Dit is een lijst met alle aspecten waar een eis aan dient te voldoen. Deze lijst is weergeven onder de deelactiviteit '*4.1. controleer op eisen aan eisen*'.

Een belangrijke valkuil is dat eisen soms het object of systeem omschrijven in plaats van de functie, dit is fout. De objectomschrijving zoals afmetingen staan al vastgelegd in de ontwerpen en hoeft dus niet nog eens extra in een eis worden omschreven en extra geverifieerd te worden. Dit kost extra tijd, geld, en is inefficiënt. De toegevoegde waarde van een SRS is juist het functionele en herleidbare aspect.

3.2. Verificatieprocedure vastleggen

Wie: Requirements Engineer en/of Systems Engineer.

Wat: Eén van de attributen van de eisen is een verificatieprocedure.

Waarom: Deze stap is belangrijk om achteraf te verifiëren in welke mate een eis wordt nageleefd.

Hoe: In de verificatieprocedure staat per eis kort uitgelegd hoe deze gecontroleerd dient te worden op de naleving van de eis in de uiteindelijke ontwerpen en realisatie. Voor bepaalde eisen is de verificatie mogelijk door te controleren of er in het uitvoeringsontwerp en/of andere tekeningen rekening is gehouden met de eis. Het V&V plan is input voor dit onderdeel.

3.3. Attributen invullen

Wie: Requirements Engineer.

Wat/Hoe: In de strategie is al vastgelegd welke attributen gedefinieerd zijn per SRS. In deze stap wordt er per eis invulling gegeven aan de verschillende attributen van een eis. Deze stap is nauw verbonden met de vorige stap, ´*3.1. eisen vaststellen/afleiden*'. Ook dit gebeurt in eerste instantie zo goed als mogelijk door diegene die de input levert.

3.4. Eisen prioriteren

Wie: Requirements Engineer.

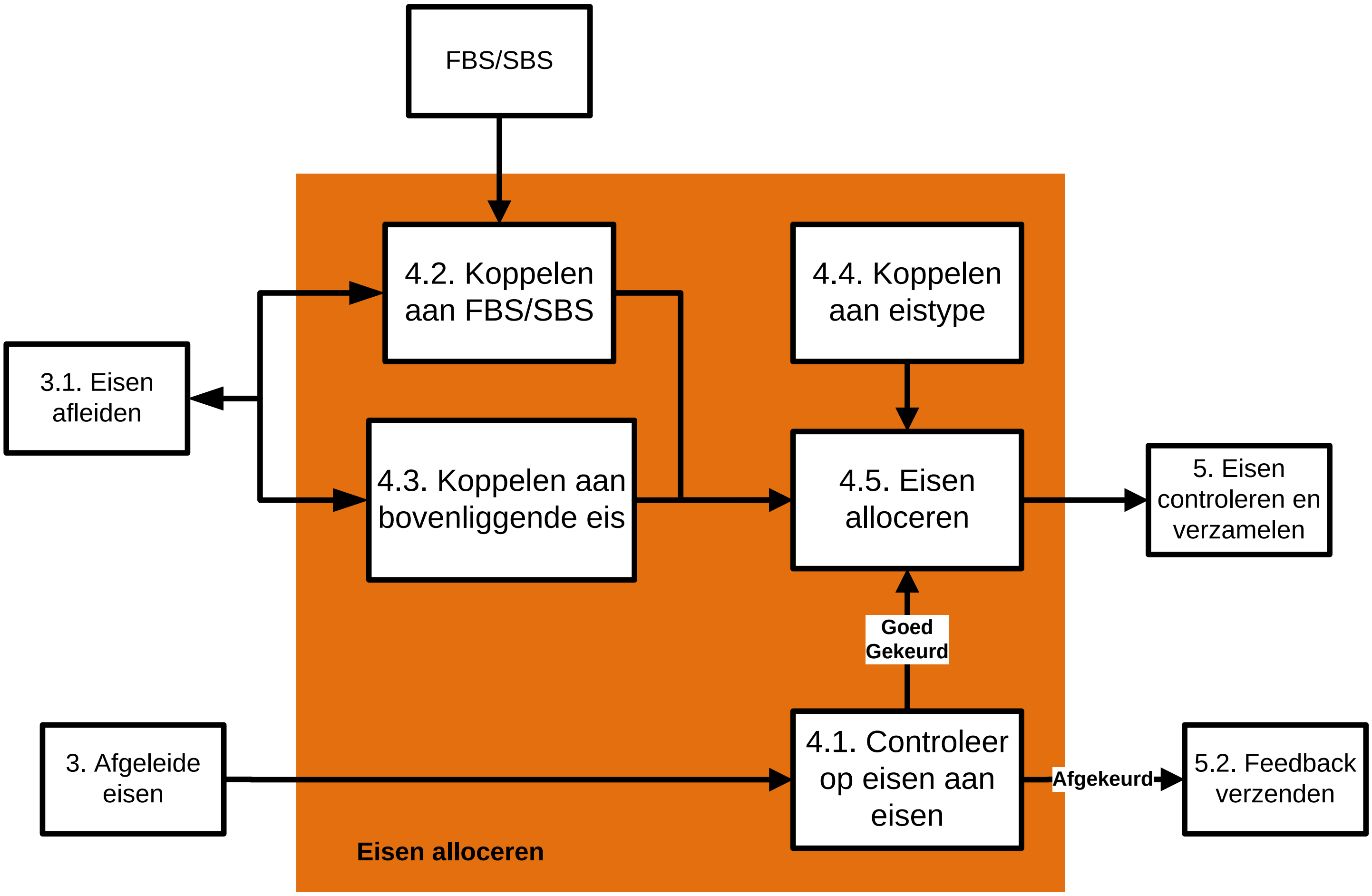
Wat: Indien in de strategie is vastgesteld dat eisen geprioriteerd dienen te worden, wordt deze activiteit bij elke eis uitgevoerd. Eisen dienen geprioriteerd te zijn naar mate van belang.

Waarom: Dan is het voor de ontwerpers duidelijk welke eisen extra aandacht verdienen ten opzichte van andere eisen.

Hoe: Hierin zijn twee niveaus gedefinieerd: 1) Het niet nakomen van deze eis bedreigt de prestatie van het hele systeem 2) Eis is belangrijk, maar het niet nakomen van deze eis zal het functioneren van het systeem niet bedreigen. Andere eisen zijn niet geschikt om in een SRS te worden opgenomen.

Terug naar: (3) Eisen afleiden

4. Eisen alloceren



Nadat een eis inhoudelijk beschreven is, dient de eis gecontroleerd, gestructureerd en gealloceerd te worden. Dat gebeurt in deze hoofdactiviteit.

Terug naar: SRS Proces

4.1. Controleer op eisen aan eisen

Wie: Requirements Engineer, Systems Engineer of Lead Engineer

Wat: Bij het controleren op eisenspecifieke eisen, dient men er wel rekening mee te houden dat de allocatie, koppeling met FBS, bovenliggende eis en eistype in sommige gevallen nog gemaakt moeten worden. Bij sommige eisen is dit al het geval, bij andere eisen dienen deze allocaties nog gemaakt te worden.

Waarom: Als de eis is opgesteld door de requirements engineer (RE) zelf, is het waarschijnlijk dat de eis al van goede kwaliteit is en dat er met allocatie al rekening is gehouden. Als eisen zijn afgeleid door mensen met weinig ervaring in systems engineering en bijvoorbeeld afkomstig zijn van een ontwerper die eisen heeft afgeleid uit zijn tekeningen, is het verstandig deze eisen te controleren nadat de eisen zijn afgeleid en onder beheer van de RE komen. Dat is de reden dat de controle stap hier al plaats vindt.

Hoe: Hieronder volgt een lijst met alle eisenspecifieke eisen waar elke systeemeis aan dient te voldoen.

Eisenspecifieke eisen	
Eis	*Uitleg*
VORM	
Uniek	Een eis heeft een uniek ID
Traceerbaar naar een bron	Een eis is traceerbaar naar een bron
Traceerbaar naar een niveau	Een eis is traceerbaar naar een hoger niveau in de eisenboom
Traceerbaar naar een functie en/of object	Een eis dient traceerbaar te zijn naar een functie en/of object
Objectief	Een eis dient subjectieve termen te mijden zoals "goed", "veilig" of "zo snel mogelijk"
Taalkundig correct	Een eis is taalkundig en retorisch correct
Actief	Een eis dient actief te zijn en de grammatisch passieve vorm mijden
Drukkend	Een eis dient een dringende syntaxis te hebben, gebruik makend van woorden zoals "moeten". Een eis kan niet optioneel zijn
Positief	Een eis dient negatieve termen zoals "niet" en "nee" te mijden, gezien dergelijke eisen moeilijker zijn om te verifiëren
Geprioriteerd	Het moet duidelijk zijn welke eisen belangrijker zijn dan anderen
Bondig	Een eis moet kort en minimaal geformuleerd zijn, relaterend naar de kern van het onderwerp
INHOUD	
Haalbaar	Een eis heeft minimaal één oplossing en implementatie
Noodzakelijk	Een eis is nodig om tot een functionerend en compleet systeem te komen
Compleet	Een eis is compleet als het genoeg gespecificeerd is voor zijn doel
Consistent	Een eis is niet tegenstrijdig met andere eisen
Eenduidig	Een eis kan slechts op één manier geïnterpreteerd worden
Verifieerbaar	Een eis kan worden geverifieerd
Oplossingsvrij	Een eis dient vrij te zijn van bepaalde oplossingen, onderdelen, leveranciers, etc.
Enkelvoudig	Een eis heeft slechts betrekking op één criterium waar een eis aan moet voldoen
Meetbaar	Het moet duidelijk zijn wanneer het doel van een eis is bereikt
Acceptabel	Een eis moet acceptabel zijn voor de betrokken stakeholders

Terug naar: (4) Eisen alloceren

Voor functionele systeemeisen is er nog een extra eis, namelijk dat functionele eisen ook daadwerkelijk functionele eigenschappen van het systeem beschrijven. Een belangrijke valkuil is dat functionele eisen soms het object of systeem omschrijven in plaats van de functie, dit is fout. De objectomschrijving zoals afmetingen staan al vastgelegd in de ontwerpen en hoeft dus niet nog eens extra in een eis worden omschreven en extra geverifieerd te worden. Dit kost extra tijd, geld, en is inefficiënt. De toegevoegde waarde van een SRS is juist het functionele en herleidbare aspect.

Indien de eis nog niet volstaat zijn er een paar mogelijkheden. Eerst dient er gekeken worden of de RE zelf de eis zo aan kan passen, dat de eis alsnog voldoet aan de eisenspecifieke eisen. Indien dit niet mogelijk is, is de eis onbruikbaar. De RE dient dan feedback te sturen naar de actor die de input geleverd heeft met het verzoek de eis aan te passen of te verwerpen.

4.2. Koppelen aan FBS/SBS

Wie: Requirements Engineer of Systems Engineer.

Wat: De meeste eisen in een SRS zijn over het algemeen functionele eisen. Functionele eisen zijn altijd gekoppeld aan de FBS, de functional breakdown structure.

Waarom: De FBS heeft een belangrijke plaats in een project. Het is namelijk zo dat een stakeholder één of meerdere behoeftes heeft. Behoeftes worden vervuld door de verschillende functies van het systeem. Functies kunnen weer verder uitgesplitst en beschreven worden in deelfuncties. Om aan een functie te voldoen, worden functionele systeemeisen opgesteld. Een ontwerper houdt vervolgens in zijn of haar ontwerp rekening met de gestelde eisen. Deze systeemeisen kunnen dan vervolgens worden gekoppeld aan een deelsysteem. Het ontwerp van het deelsysteem vervult dan de behoeftes van de stakeholders.

Hoe: Als er nieuwe eisen worden afgeleid uit nieuwe ontwerpen of ontwerp besluiten, is er soms niet direct een link gemaakt met de FBS. In dit geval dient de requirements engineer de eis te koppelen aan een functie. Hierbij kan de FBS eventueel aangepast worden.

De SBS is de system breakdown structure. Dit is de structurele boomweergave van het systeem, de deelsystemen en componenten. De SBS resulteert uit de ontwerpfase, en functioneert als overzichtelijke, gestructureerde weergave van het systeem.

4.3. Koppelen aan bovenliggende eis

Wie: Requirements Engineer of Systems Engineer.

Wat: Zoals in de deelactiviteit '*3.1. eisen vaststellen/afleiden*' is benoemd, kan een bepaalde eis worden afgeleid uit een bovenliggende eis door een bovenliggende eis verder te specificeren.

Hoe/Waarom: Als er nieuwe eisen worden afgeleid uit nieuwe ontwerpen of ontwerp besluiten, is er soms niet direct een link gemaakt met een eventueel bovenliggende eis. In dit geval dient de requirements engineer de eis te koppelen aan een bovenliggende eis.

Terug naar: (4)
Eisen alloceren

4.4. Koppelen aan eistype

Wie: Requirements Engineer of Systems Engineer.

Wat/Waarom: De belangrijkste scheiding tussen verschillende eisen is het eistype. Normaal gezien heeft degene die een eis afleidt, het eistype al bedacht voor of tijdens het afleiden. In deze activiteit dient het eistype formeel te worden vastgelegd.

Hoe: De eisen binnen een systeem zijn onderverdeeld in de volgende typen eisen:

- Functionele eisen
- Aspect eisen
- Interne raakvlakeisen
- Externe raakvlakeisen
- Realisatie eisen

Functionele eisen

De functionele eisen beschrijven de gewenste functionele eigenschappen van het systeem na oplevering.

Aspect eisen

De aspecteisen beschrijven een functie vanuit verschillende aspecten

Aspect	Toelichting
Betrouwbaarheid (R)	Eisen met betrekking tot de mate waarin het systeem een bepaalde periode zonder falen zijn functie kan vervullen.
Beschikbaarheid (A)	Eisen met betrekking tot de fractie van de tijd dat het systeem de vereiste functie kan vervullen.
Instandhouding (M)	Eisen met betrekking tot benodigde instandhoudingvoorzieningen en instandhoudingbehoefte.
Veiligheid (S)	Eisen met betrekking tot veiligheid gedurende realisatie en de rest van de levensduur.

Raakvlakeisen

De raakvlakken worden op systeemniveau beschreven (dus niet per deelsysteem).

Daar waar het noodzakelijk is om (ontwerp)beperkingen of afwijkingen van de regelgeving te benoemen worden deze opgenomen bij de eisen van het betreffende (sub)systeem. Raakvlakeisen leggen beperkingen op aan het systeem. Sommige raakvlakken zijn niet vertaald in raakvlakeisen maar in

Terug naar: (4) Eisen alloceren

functionele eisen of aspecteisen. Deze keuze is afhankelijk van of er al een keuze aan ten grondslag ligt waardoor er een systeemfunctie is ontstaan.

Hoewel in het ontwikkelproces het systeem nog niet opgeknipt wordt naar contracten is het wel belangrijk om de kritische raakvlakken tussen de subsystemen te benoemen en te beschrijven. Het betreft dus raakvlakken binnen de scope van het System of Interest. Op het moment dat besloten wordt hoe de realisatie van het systeem wordt opgeknipt naar contracten, worden de raakvlakeisen uitgewerkt op de raakvlakken tussen de contracten. Dit gebeurt in de Vraagspecificatie Eisendeel. Indien de interne raakvlakken expliciet overgelaten worden aan de ontwerpende partij, dient dit ook vermeld te worden.

Realisatie eisen

Realisatie eisen zijn eisen die gelden gedurende de realisatie ofwel die iets zeggen over de manier waarop het systeem gebouwd moet worden of de beperkingen die gelden tijdens de bouw.

4.5. Eisen alloceren

Wie: Requirements Engineer en/of Systems Engineer.

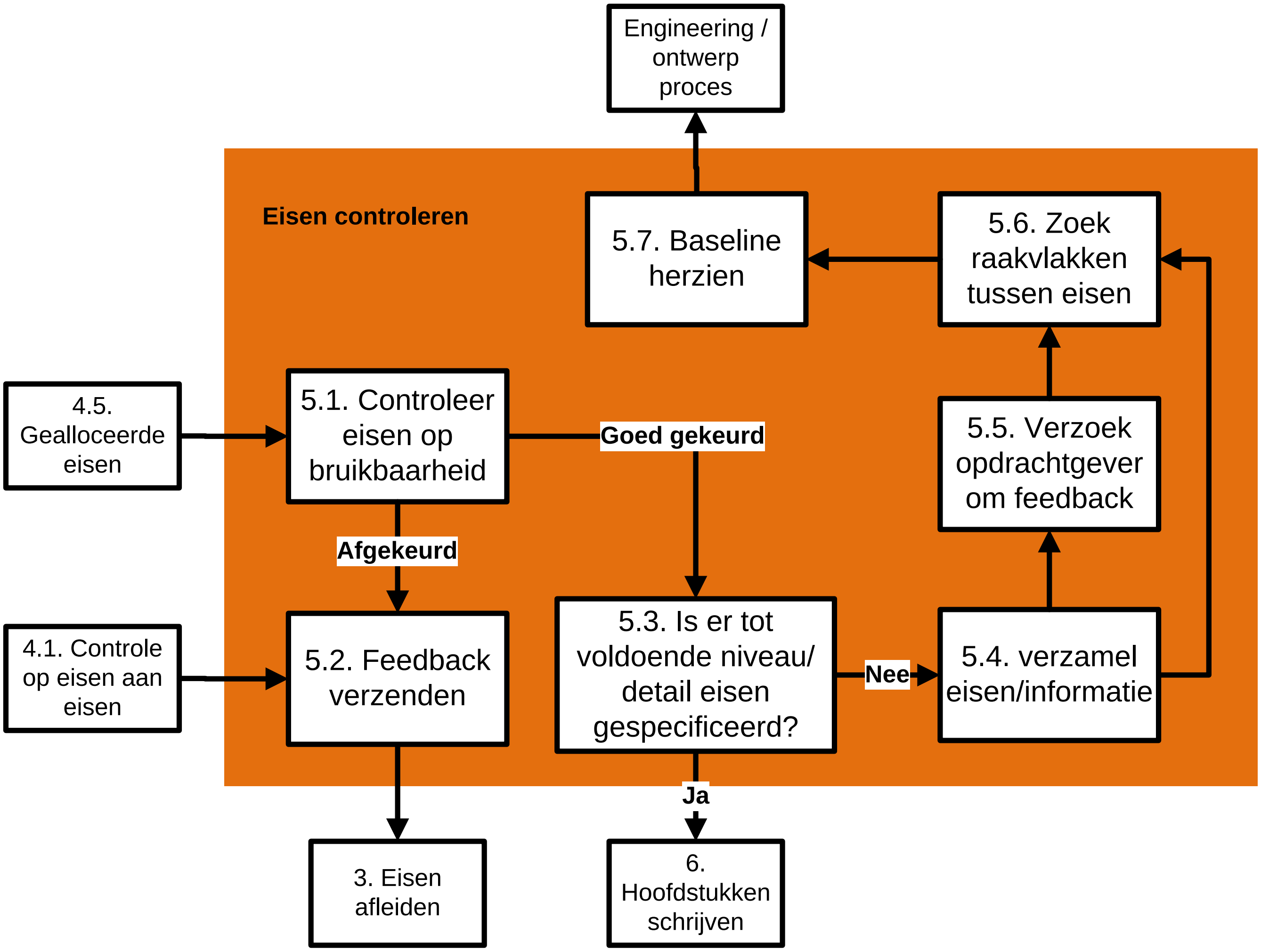
Wat/Waarom: Het doel van deze stap is het structureren van de eisen. Hier komen verschillende stappen samen. Uiteindelijk dienen eisen na deze stap overzichtelijk en gestructureerd te zijn. Nadat eisen gealloceerd zijn, horen eisen te voldoen aan alle eisenspecifieke eisen.

Hoe: De requirements engineer (RE) beheert de eisen en zorgt dat nieuwe eisen een logische plek innemen tussen de bestaande eisen en eisenboom. Als er nog lege plekken zijn in de eisenboom of onvervulde functies, raakvlakken, aspecten, etc., dient de RE daarop te sturen, om een zo goed en compleet mogelijke eisenboom te krijgen.

Een logische structuur voor eisen binnen een SRS, is om in eerste instantie eisen te scheiden op (deel)systeem en in tweede instantie op eistype.

Terug naar: (4)
Eisen alloceren

5. Eisen controleren en verzamelen



In deze hoofdactiviteit worden de nieuwe eisen voorbereid op de iteratie en terugkoppeling met de ontwerpers.

Terug naar: SRS Proces

5.1. Controleer eisen op bruikbaarheid

Wat/Waarom/Hoe: In deze stap zijn complete, volwaardige eisen tot stand gekomen. Er dient echter een laatste controle uitgevoerd te worden, namelijk of de eisen daadwerkelijk toepasbaar zijn en goed gebruikt kunnen worden door de ontwerpers die de eisen dienen toe te passen.

Wie: Deze controle dient te worden uitgevoerd door iemand met vakinhoudelijke kennis zoals een Lead Engineer of Lead Designer.

5.2. Feedback verzenden

Wie: De controleur stuurt de feedback naar de Requirements Engineer, die eventueel op zijn beurt feedback stuurt naar diegene waar de input vandaan komt.

Wat/Hoe: Indien een eis is afgekeurd, dient er feedback worden verzonden met uitleg waarom de eis niet bruikbaar is. Wellicht kan de eis dan worden herzien en doorloopt de eis opnieuw het proces. De eis kan ook compleet worden afgedaan indien noodzakelijk.

5.3. Is er tot voldoende niveau/detail eisen gespecificeerd?

Wie: Requirements Engineer en/of Systems Engineer.

Wat/Hoe: Vooraf in de strategie is vastgelegd tot welk niveau eisen dienen te worden gespecificeerd en ontworpen. Indien dit niveau bereikt is, kan de SRS worden afgerond. In dat gevalworden de eisen in het hoofdstuk '*Eisen*' geplaatst van het SRS document. Als het SRS document dan afgerond is, kan het gecontroleerd worden op kwaliteit. Dit zijn echter volgende activiteiten.

5.4. Verzamel eisen/informatie

Wie: Requirements Engineer.

Wat/Waarom: Nieuwe eisen en andere informatie worden niet direct één voor één doorgespeeld naar de ontwerpers. Deze worden eerst verzameld tot het geschikte baseline moment.

5.5. Verzoek opdrachtgever om feedback

Wie: Requirements Engineer, Project Manager.

Wat/Waarom: Het is belangrijk om de opdrachtgever betrokken te houden in het proces, hierdoor voorkom je dat deze voor verrassingen komt te staan, en voorkom je rework in de toekomst.

Hoe: Een geschikt moment daarvoor is nadat er een aantal eisen zijn verzameld en voordat deze gebruikt worden voor input om verder mee te ontwerpen.

Terug naar: (5) Eisen controleren en verzamelen

5.6. Zoek raakvlakken tussen eisen

Wie: Systems Engineer.

Wat: Tot het baseline moment, dient er gezocht te worden naar raakvlakken tussen eisen en andere informatie zoals ontwerp besluiten.

Waarom: Het kan gebeuren dat door nieuwe ontwerpbesluiten en eisen, nieuwe raakvlakken ontstaan of raakvlakken veranderen. Het is belangrijk dit voor het baseline moment uit te zoeken en af te stemmen op elkaar.

5.7. Baseline herzien

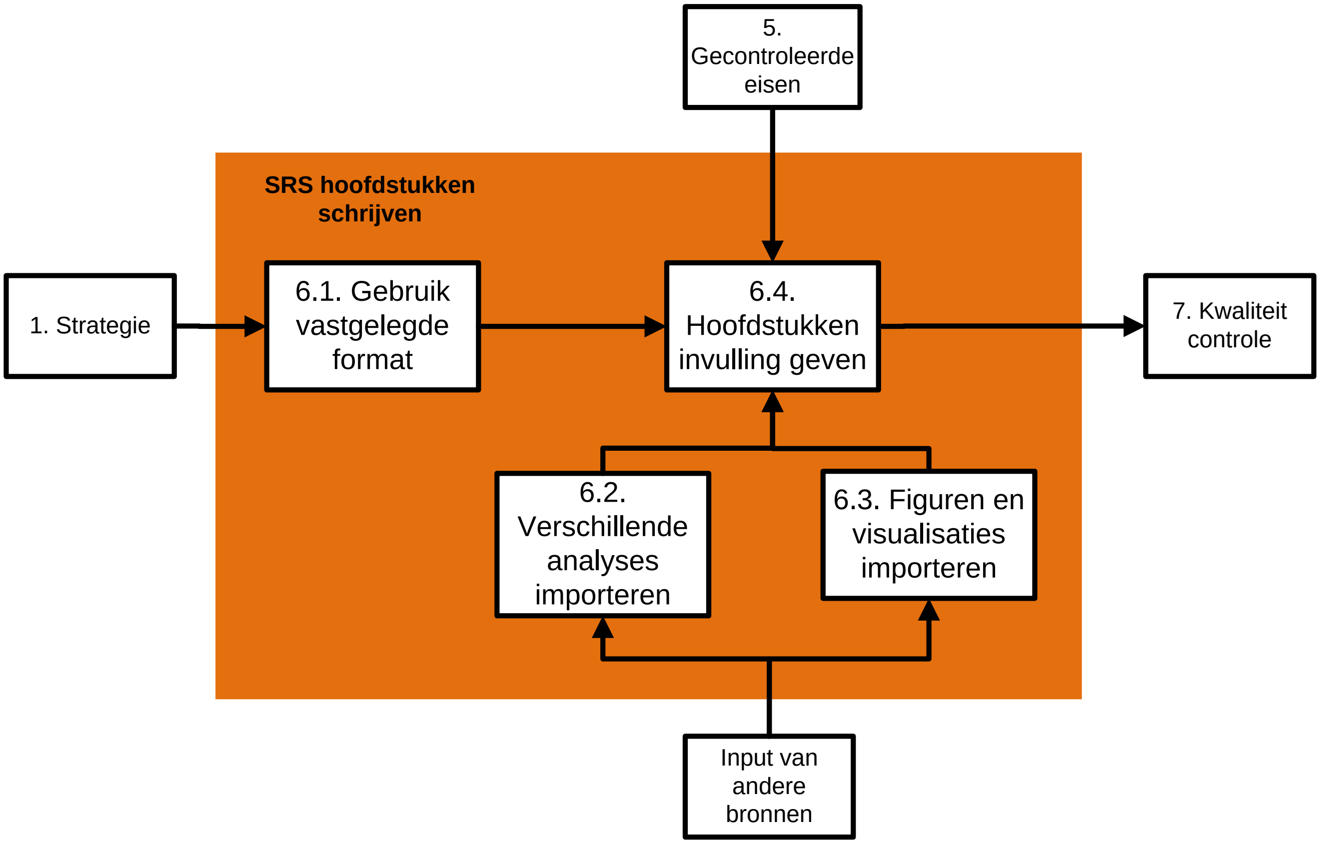
Wie: Requirements Engineer, Systems Engineer, Lead Engineer.

Wat/Hoe: Vooraf is in de strategie al nagedacht over geschikte baseline momenten. Nu het baseline moment nadert, dient deze te worden gespecificeerd naar een concreter geschikt moment. Nu dat alle informatie en eisen tot de baseline zijn verzameld, is er ook een geschikt moment om samen met de Lead Engineer te controleren of de ontwerpers en Requirements Engineer op één lijn zitten.

Waarom: Vaak zijn bij een project meerdere verschillende technische vakdisciplines en ontwerpers betrokken die van elkaar afhankelijk zijn. Onder één systeem worden vaak meerdere deelsystemen ontworpen door verschillende ontwerpers en/of ontwerpafdelingen. Deze deelsystemen hebben raakvlakken met elkaar, en ontwerpers dienen rekening te houden met deze raakvlakken. Gedurende het project komen op verschillende momenten verschillende nieuwe eisen van verschillende disciplines binnen. Deze nieuwe eisen zijn input voor de verschillende ontwerpers. Nieuwe eisen kunnen de bestaande ontwerpen flink beïnvloeden en het kan zijn dat raakvlakken veranderen of ontstaan. Voordat nieuwe eisen worden doorgegeven, is het verstandig goed na te denken over de gevolgen voor de raakvlakken tussen ontwerpers. Het is niet zo dat nieuwe eisen en aanpassingen zo snel mogelijk moeten worden doorgegeven, terwijl verschillende ontwerpers vol in hun ontwerpproces zitten. De verschillende ontwerpers en disciplines dienen afgestemd te worden op elkaar op bepaalde baseline momenten, waar ook nieuwe eisen kunnen worden geïntroduceerd. Dit is natuurlijk niet het geval voor alle eisen, maar wel met eisen die de onderlinge raakvlakken beïnvloeden.

Terug naar: (5) Eisen controleren en verzamelen

6. SRS hoofdstukken schrijven



In deze hoofdactiviteit wordt de SRS zelf gemaakt. De verschillende hoofdstukken krijgen in deze stap de inhoudelijke invulling door de auteur.

Terug naar: SRS Proces

6.1. Gebruik vastgelegde format

Wie: Requirements Engineer.

Wat/Hoe: In de strategie is vastgelegd welk format gebruikt wordt en welke hoofdstukken in welke mate ingevuld worden. Nu het document daadwerkelijk inhoud krijgt, dient men deze vooraf bepaalde keuzes toe te passen.

6.2. Verschillende analyses importeren

Wie: Requirements Engineer.

Wat/Waarom/Hoe: Een SRS bestaat niet alleen uit eisen. Er zijn verschillende analyses, rapporten en besluiten die als input fungeren voor het SRS document. Denk hierbij aan een FBS, raakvlakken analyse, stakeholder analyse, ontwerpbesluiten, etc. Zoals vooraf in de strategie is vastgelegd, krijgen deze onderdelen een gepaste plek in de SRS.

6.3. Figuren en visualisaties importeren

Wie: Requirements Engineer.

Wat/Waarom/Hoe: Figuren en visualisaties zijn meestal meer verklarend en aantrekkelijker dan woorden. Zeker bij een complex document als een SRS, dienen deze dan ook gebruikt te worden waar dat van toegevoegde waarde is. Dit is vaak ook in de strategie vastgelegd. Denk hierbij aan plattegronden, architectural design, satellietfoto's, een ui-diagram (voor stakeholders), visualiserende ontwerpschetsen, etc.

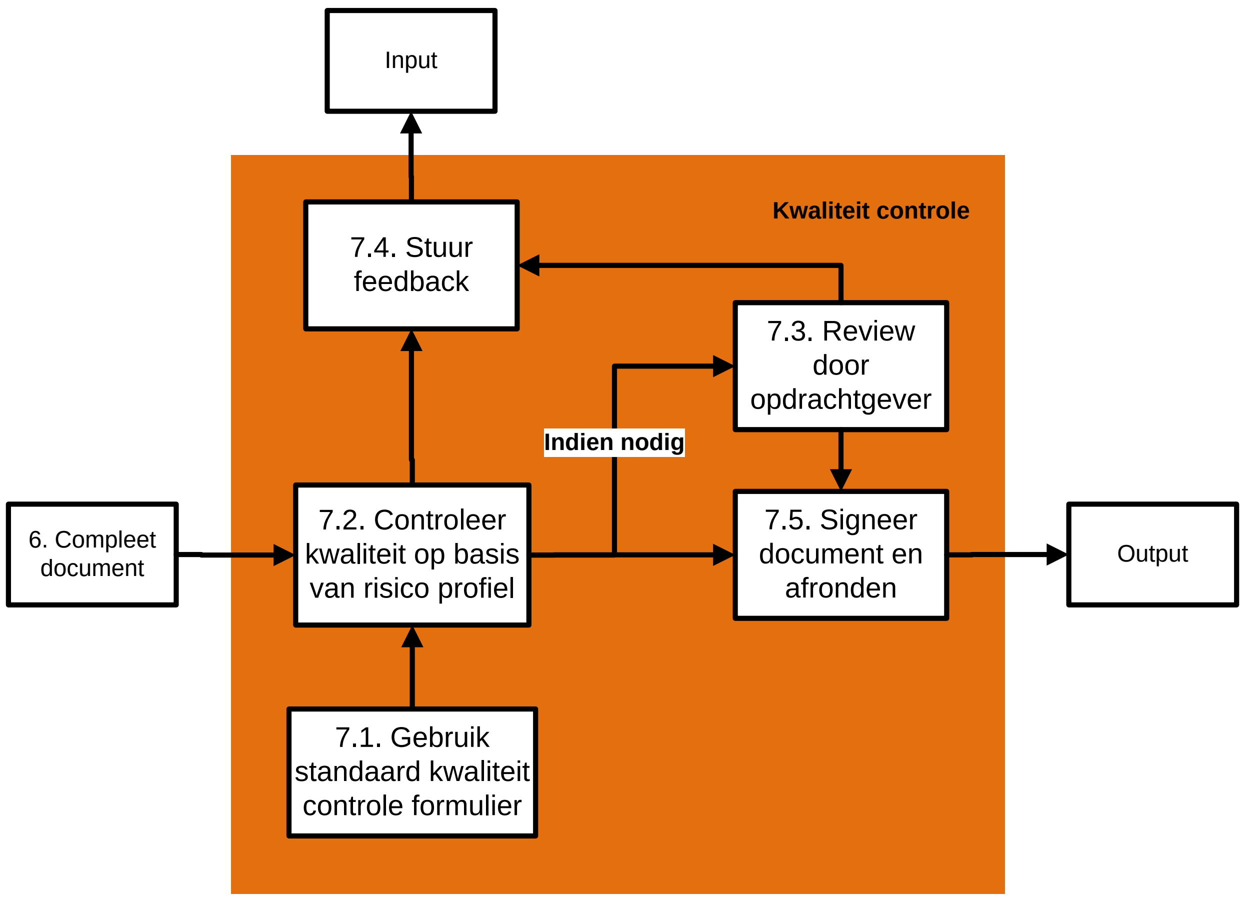
6.4. Hoofdstukken invulling geven

Wie: Requirements Engineer.

Wat/Hoe: Deze activiteit kan gedurende het project verder vervuld worden en loopt daarom parallel aan het afleiden van eisen. De hoofdstukken en onderdelen van de SRS waar invulling aan gegeven moet worden zijn bepaald in de strategie. In deze stap komen alle verschillende aspecten van de SRS samen. Deze activiteit wordt uitgevoerd door de auteur. De auteur is meestal de requirements engineer(s) zelf. Als het document compleet is, kan het gecontroleerd worden op kwaliteit.

Terug naar: (6) SRS hoofdstukken schrijven

7. Kwaliteit controle SRS



Deze hoofdactiviteit is de laatste, waarin het document wordt gecontroleerd en afgerond.

Terug naar: SRS Proces

7.1. Gebruik standaard kwaliteit controle formulier

Wie: Reviewer.

Wat/Hoe: Vooraf is het standaard review formulier opgesteld en zijn de reviewers gekozen. Deze dient nu toegepast te worden in de volgende stap.

7.2. Controleer kwaliteit op basis van risico profiel

Wie: Reviewer.

Wat/Hoe: Voor elk project is een risico profiel bepaald. Hieruit blijkt een toetsniveau waarop documenten worden getoetst, dit geldt ook voor de SRS. Aan de hand van het toetsniveau blijkt bijvoorbeeld of het document gecontroleerd dient te worden door een onafhankelijk persoon, of niet.

7.3. Review door opdrachtgever

Wie: Opdrachtgever.

Wat/Waarom/Hoe: Eventueel, als de opdrachtgever dat wilt, kan het document na goedkeuring binnen Movares ook door de opdrachtgever gecontroleerd worden, of door een vertegenwoordiger van de opdrachtgever. Hier wordt wel op aangedrongen, aangezien het belangrijk de opdrachtgever mee te nemen in het proces, en deze niet voor verrassingen komt te staan.

7.4. Stuur feedback

Wie: Reviewer, Requirements Engineer.

Wat/Hoe: Indien het document nog niet voldoende goed is bevonden, stuurt de reviewer het ingevulde formulier met feedback naar de auteur van de SRS. De auteur (of requirements engineer) dient de feedback dan te verwerken.

7.5. Signeer document en afronden

Wie: Requirements Engineer.

Wat: Als het document door Movares en de opdrachtgever goed is bevonden, kan het document gesigneerd worden en vrijgegeven worden. De SRS is doorgaans input voor een uitvoerend ontwerp en vraagspecificatie, dit hangt af van het contract.

Hoe: In ProjectWise dient het document op de juiste, daartoe voorgeschreven manier worden aangemaakt en vrijgegeven. Het document dient bewaard te worden in de projectmap "productdocumenten". Aangezien per opdrachtgever de naam van het document nog wel eens kan verschillen (SRS, SES, PvE, etc.) dient er in ieder geval in de beschrijving van het document te staan "SRS". Daarnaast dient er te staan of de versie definitief is, of bijvoorbeeld concept. Uiteindelijk staat er dus in de document beschrijving: "SRS definitief".

Terug naar: (7)
Kwaliteit controle SRS