Defining requirements in a process and system development project

Ron Lindholm

Degree Thesis

Information and Media Technology

2012

DEGREE THESIS	
Arcada	
Degree Programme:	Information and Media Technology
Identification number:	9675
Författare:	Ron Lindholm
Title:	Defining requirements in a process and system development project.
Supervisor (Arcada):	Göran Pulkkis
Commissioned by:	TeliaSonera Finland Oyj

Abstract:

This thesis work is about requirements engineering, a subdiscipline of systems and software engineering. The thesis is divided into two parts, a theoretical and a practical part. The theoretical part describes requirements engineering and the practices involved as well as common process models used, and aims to give the reader an understanding of the subject. The significance of requirements engineering in projects is also studied and inspected. The practical part is a real world case about requirements engineering, with the intention to identify a company's needs and requirements for a network information system. The creation and definition of a suitable process for similar requirements engineering tasks for future usage scenarios is included. The process was based on standard requirements engineering practices, and named as the Business Requirement Gathering Process. The obtained results from the requirements engineering case were compiled into a formally structured document, a requirements catalog, which purpose is to function as specifications for a network information system.

Keywords:	Requirements engineering, requirements, process and system development, specifications, process models.
Number of pages:	41
Language:	English
Date of acceptance:	15.11.2012

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informations- och medieteknik
Identifikationsnummer:	9675
Författare:	Ron Lindholm
Arbetets namn:	Definition av krav i ett process- och systemutvecklingsprojekt.
Handledare (Arcada):	Göran Pulkkis
Uppdragsgivare:	TeliaSonera Finland Oyj

Sammandrag:

Detta examensarbete handlar om kravhantering, från ett system- och programutvecklingsperspektiv, där kravhantering är en underkategori inom området. Arbetet är indelat i två separata delar, en teoretisk del och en praktisk del. Den teoretiska delen beskriver kravhantering och vad detta innebär, samt allmänna processmodeller inom området. I arbetet undersöks också betydelsen av att tillämpa kravhantering inom projekt. Den teoretiska delen strävar till att ge läsaren en överblick och förståelse av ämnet. Den praktiska delen beskriver en verklig kravhanteringsuppgift för ett företag, där målet var att identifiera företagets olika behov och krav för ett nätverksinformationssystem. Arbetet innehöll också planläggning och definition av en lämplig process för liknande kravhanteringsuppgifter. Processen, som döptes till ”The Business Requirement Gathering Process”, skapades genom att tillämpa allmänna standardmetoder inom kravhantering. Resultaten som slutligen uppnåddes i kravhanteringsarbetet sammanställdes som ett formellt strukturerat dokument, i formen av en s.k. kravkatalog, vars syfte är att fungera som specifikationer för ett nätverksinformationssystem.

Nyckelord:	Kravhantering, krav, process- och systemutveckling, specifikation, processmodeller.
Sidantal:	41
Språk:	Engelska
Datum för godkännande:	15.11.2012

OPINNÄYTE	
Arcada	
Koulutusohjelma:	Informaatio- ja mediatekniikka
Tunnistenumero:	9675
Tekijä:	Ron Lindholm
Työn nimi:	Vaatimusten määrittely osana prosessi- ja järjestelmäkehitysprojektia.
Työn ohjaaja (Arcada):	Göran Pulkkis
Toimeksiantaja:	TeliaSonera Finland Oyj

Tiivistelmä:

Opinneäytetyön aiheena on vaatimusmäärittely, tarkasteltuna erityisesti järjestelmä- ja ohjelmistokehityksen näkökulmasta. Opinnäytetyö on jaoteltu kahteen erilliseen osaan, teoreettiseen ja käytännön osuuteen. Teoriaosuus käsittelee vaatimusmäärittelyä ja siihen kuuluvia toimenpiteitä, vaatimusmäärittelyssä yleisesti käytetyimpiä prosessimalleja ja lopuksi tarkastellaan vaatimusmäärittelyn merkitystä onnistumisen kannalta. Teoriaosuuden tarkoituksena on antaa lukijalle yleiskäsitys ja ymmärrys opinnäytetyön aiheesta. Käytännön osuudessa suoritetaan ja kuvataan todellinen vaatimusmäärittelyn tehtävä tilaavalle yritykselle. Tehtävän tavoitteena on tunnistaa yrityksen tarpeet ja vaatimukset verkkotietojärjestelmälle. Työhön kuului myös asianmukaisen prosessin määrittely ja kuvaaminen, vastaavien vaatimusmäärittelytöiden tarpeita varten. Tulokseksi syntyi prosessi; ”The Business Requirement Gathering Process”, joka soveltaa yleisiä standardin mukaisia vaatimusmäärittelyn menetelmiä ja käytäntöjä. Lopuksi vaatimusmäärittelytehtävän tulokset koottiin muodolliseen jäsenneltyyn asiakirjaan, niin sanottuun vaatimusluetteloon, jonka tarkoituksena on toimia määrittelyinä verkkotietojärjestelmän kehittämisessä.

Avainsanat:	Vaatimusmäärittely, vaatimus, prosessi- ja järjestelmäkehitys, määritelmä, prosessimallit.
Sivumäärä:	41
Kieli:	Englanti
Hyväksymispäivämäärä:	15.11.2012

CONTENTS

Figures

Abbreviations and Definitions

BRGP	Business Requirement Gathering Process
IEEE	Institute of Electrical and Electronics Engineers
NIS	Network Information System
RE	Requirements Engineering
SRS	Software Requirements Specification
TTM	Time to market

ACKNOWLEDGEMENTS

First and foremost I would like to give my thanks to my instructor and supervisor Mr. Pasi Savilaakso, and later Mr. Sami Mikkonen, at TeliaSonera Finland Oyj for giving me the opportunity for this thesis work, as well as other superiors that have made the necessary arrangements making this work possible.

I wish to express my gratitude to my supervisor at Arcada, Dr. Tech Göran Pulkkis, for general advice and assistance regarding the whole writing process of my thesis work. I would also like give special thanks to Magnus Westerlund and Johnny Karlsson for the education I have received during their courses at Arcada.

This thesis work has been a great valuable learning experience. The thesis completes the final stage of my academic degree, making it finally possible to move forward for new experiences and challenges.

Helsinki, October 2012

Ron Lindholm

1 INTRODUCTION

The information technology branch and its industry has been growing with an exponential pace during the last few decades, yet without any signs of significant growth restrictions or slowdowns. The accelerating changes and technological development is creating new opportunities for businesses, and in order to be competitive in the field or to have that edge over other competitors, the businesses must keep evolving and adapt to the environment and its requirements.

Nowadays it's common, and becoming increasingly more so, that work and tasks are carried out as a project, which usually includes many different stages according to used models and processes. Project driven work is common, especially in the IT and software industry. To successfully drive and carry out a project requires proper planning and execution, which is crucial for large scale projects. This has been shown, and is widely known, to be particularly important in software and systems engineering, as projects are easily prone for delays and increased expenditures caused by improper planning or bad processes.

It is needed to know what kind of goals a project is aiming for and what are the underlying needs; specifications have to be set and defined. There is a specific knowledge area which includes different stages designed to solve and clarify these needs called requirements engineering, which will also be a main subject of this thesis.

1.1 The contractor, Intentions and Goals of the Thesis

This thesis work is a part of a more comprehensive large-scale project, which was initiated because of a greater need to increase quality and efficiency in the current and future processes and operations, with the intention to achieve improved cost-effectiveness. The contractor of this work is one of the dominant companies in the telecommunications industry domestically in both Finland and Sweden, as well as internationally in the Northern and Eastern Europe, with operations in other regions as well.

This work aims particularly to find out and identify what kind of needs and requirements the company's various businesses and departments are having for a Network Information System. A Network Information System (NIS) is a computer application landscape, which may consist of several different components and applications, used for managing telecommunication networks and all relevant data, e.g. connections and their attributes, network infrastructure, operational data, customer information and jobs or work requests.

This thesis work was carried out by collaborating with various departments and their corresponding people and stakeholders, to collect and clarify their requirements. The work goals included compilation of the collected requirements, specifying and defining them, and writing them formally as a requirements catalog, which was then delivered for the IT-system architects' usage.

The requirements catalog was meant to be the final product, but because of the lack and definition of a requirements gathering process, the creation and documentation of a process for it was included in the thesis work, to be also used in future similar projects.

1.2 Limitations of the Work

The thesis is divided into two different parts, a theoretical and a practical part. The theoretical part is focusing on the topic and different aspects of requirements engineering, with the intention to give the reader an overview of the subject and what kind of steps are usually involved in the process. While requirements engineering may be applied in many different areas of engineering practices and tasks, this thesis work is primarily made from a system and software engineering perspective.

The practical part is a thorough description and documentation on how making of the requirements catalog was carried out in practice. The business requirements gathering process is divided and described by different steps, which were created and used for this thesis work. The practical part is meant to be supported by the theoretical part, by following standard routines of requirements engineering.

This thesis work does not contain any specific details of the planned NIS, or other applications or tools involved and used by the company. It does not either describe in detail what kinds of requirements were collected or contain any details about the stakeholders or their identity. The information is presented in a general way, so that this thesis work could be used as a guideline.

2 REQUIREMENTS ENGINEERING (RE)

Requirements engineering (RE) should be considered as its own separate area of expertise and practice aside from systems and software engineering. Requirements engineering may also be considered as a sub discipline of systems and software engineering, or actually rather systems engineering in a more general way because requirements are relevant in every engineering task, not only in software engineering. (Hull et al. 2011)

A way to measure the degree of success of a systems and software project is the level how well the stakeholder's initial purposes and goals were achieved, including the satisfaction of the outcome. RE can be described as a process, implemented in the early stages of a development cycle or model, defining all different activities involved with the objective to establish the goals so they can be achieved as accurately as possible. The needs and requirements must be established before a system can be developed. (Nuseibeh & Easterbrook 2000)

Requirement engineering is a process driven practice, worked with in the form of project work, which may continue until the process is ended by different reasons, e.g. the goals are reached or the task is suspended or terminated. Project work is defined only to be temporary by its nature, i.e. projects are not supposed to be infinite or ongoing without limits, although projects may last and continue for several years without seeing a definitive end. The results or outcome of a project, e.g. a product or service, is necessarily not temporary as projects are generally done with the intentions to create something which lasts. (Project Management Institute 2004 p. 5)

2.1 Requirements

The definition of a requirement may be done in several ways and by using many synonyms describing a requirement. In a quite common and typical way, a requirement can be defined as a feature, function, capability, property, attribute, characteristic, behavior or constraint that is necessary and must therefore be present in order to solve some kind of real-world problem. Requirements should also be measurable, verifiable and traceable. (Kandt 2003, Hull et al. 2011)

The IEEE standard for Application and Management of the Systems engineering Process (IEEE-Std-1220-1998) tells also that a requirement can be defined as a statement, describing the constraints and characteristics of a product or process briefly in a concise form, and is essential in order for the specified product or process to be accepted. (Hull et al. 2011)

Requirements are the key of potential for quality and a successful product; meaning that requirements and their quality will correspond to the quality of a product. Improving requirements will respectively improve the quality of a product. (Hull et al. 2011)

2.1.1 Types of Requirements

The requirement types can be categorized in several different ways, as requirements originate from many types of stakeholders with diversifying backgrounds, level or area of operation, tasks, technical knowledge etc. The requirement type may be dependent on the work or project and its intentions, i.e. what is the meaning, or if there are specific needs. A typical way to categorize the requirement type on a higher level is by using three different types; business requirements, product requirements, process requirements.

Business requirements are statements from a business point of view, describing what kind of needs, goals or visions there are to be accomplished, often in order to increase the value of a business operation. Business requirements are regarded as high-level requirements, which are initial problems that should be solved. Higher level requirements are not meant to be technical or define how a problem should be solved; the meaning is to provide a concise description of what is wanted to be achieved, e.g. “we have a need to increase the customer satisfaction”, and they should be formulated so that the management and other non-technical persons would understand it.

Product requirements are requirements concerning some kind of product e.g. a system or software that is being developed, and describe its features. Product requirements may be more specifically defined as system requirements, as usually the product developed is a system of some sort, especially when relating to software engineering. A system may contain several different components, where the system defines the entirety of it. (Abran et al. 2004)

Product requirements may form a solution for a business requirement, how it’s going to be accomplished, and therefore in such a case the product requirement is a lower level requirement. A response to the earlier business requirement may e.g. be “the communication tool must identify the customer”. Features should still be described in an abstract way at this level, as the development process continues the earlier requirements will be the base and followed by other even lower level requirements, which are describing solutions in a more specific technical way. Product requirements can be further categorized as functional and non-functional requirements. Functional requirements describe the functionality and capability of what a system is required to do, while non-functional requirements describe restrictions and constraints which could affect solutions and how a system may operate, e.g. availability, maintainability, interoperability, portability, reliability, safety, security (Kandt 2003). The non-functional requirements could be even further categorized based on what they are concerning, e.g. as in in the example above. (Abran et al. 2004)

Process requirements describe the way of performing, how to handle a task, and how something should be done. In software engineering, in the process of product development, a process requirement may define what programming language, tools and software should be used, or sets other requirements which could restrict available options. (Kandt 2003)

The many types of requirements are affecting each other; they are linked together by relationships through different levels. Commonly high-level requirements are linked to several low-level requirements, even though they may be linked in the other direction as well. The way how requirements are transformed from high-level into more low-level ones is called requirements tracing, it’s the comprehension of relationships between requirements on different levels. Tracing of requirements and their relationships on different levels will allow and improve certain things, e.g. improved progress measuring, calculation of benefits vs. expenditures, impact analysis. (Hull et al. 2011)

2.2 Stakeholders

A stakeholder is anyone with the capability to affect or to be affected by actions. Generally this will include clients, customers, employees, investors and suppliers in an organization. A stakeholder can further be defined as some entity; a person, group of people, organization or some other independent being that is having some kind of interest in something, e.g. a system for usage, benefits, advantage, but also disadvantage, expenses and other things that could affect or lead to consequences. (Hull et al. 2011)

Stakeholders and identification of them is essential for success in a project, because of the interest they have, and therefore they are the highest priority source of requirements. While it is beneficial to have a large amount of stakeholders for requirements, it is common that different parties have conflicting opinions and needs, which could be problematic. Discovering potential conflicts will ensure that they can be taken into account in later phases of system development.

2.3 Requirements Engineering Practices

Standard requirements engineering practice is commonly concerned with identification of stakeholders, discovering and documenting their needs and requirements, compilation of requirements into a formal document, a so-called requirements catalog, analysis of requirements, and validation of requirements,. The requirements documentation or catalog will function to define the needs and required features in a way making it possible to implement them, e.g. by IT-architects and software engineers. Early requirements will function as a base for creating software requirements specifications (SRS), which are later to be followed in the development and implementation phases of the specific product being worked on. Specifications may also be used as an agreement with a client, to define what is going to be built.

The activities involved in requirements engineering can be broken down into several sub-areas or -processes, which generally consist of elicitation, analysis, specification, and validation of requirements. These activities are regarded as the most common and important phases in RE, and will be further handled as separate topics in this thesis work. Other phases could also be included depending on the RE process, on the scope of a project, and on its complexity. This will affect the amount of work and time required to complete the whole requirements engineering task. Examples for other phases could be documentation, system modeling and requirements management.

2.3.1 Elicitation

Requirements elicitation is regarded as a sub-area of RE, and it’s generally the first phase in RE processes. Requirements elicitation may also be described as collecting requirements or requirements gathering, although elicitation is the most commonly used terminology, because one should not expect to receive all requirements in an accurate way by simply questioning the users and stakeholders what a system should do.

The main intentions includes getting an overview of a problem which is yet to be solved, e.g. by a system or software that is going to be made according to the purposes. This is achieved by collecting requirements by various means and techniques from the stakeholders involved. The elicitation work may be done by someone who has basic knowledge of RE, e.g. a requirements specialist, software engineer or IT-architect, depending on resources and the scope of the whole project. One of the most important traits in requirement elicitation is good communication co-operation skills, as the person must be able to question the stakeholders and get them to talk openly during interviews and meetings, so that they can reveal their requirements. (Hull et al. 2011 p. 105)

Before being able to start the elicitation work itself, the possible requirement sources must be known and identified, i.e. the stakeholders and equivalent people, like the customer and end users which could be the target group for a new system. It is important to have as many sources and alternatives as possible, as it will potentially improve the end results by the amount of different viewpoints when gathering requirements across the board.

Different elicitation techniques are commonly used to collect the requirements; these may include stakeholder and expert interviews, meetings, observation, different inquiry methods and surveys. Interviews can be held individually, but it would be advantageous to arrange group interviews or meetings where people can openly discuss their point of views regarding their needs for better overall awareness. The discussion between stakeholders may also result in realization of possible conflicts, which would have a better opportunity to get solved because of the early phase in the development process.

Observation of users working in their own environment may also prove to be particularly useful for understanding their task and to realize how they handle a system or software. By getting a general overview and understanding of the tasks needed to be done, will make it easier to carry out the elicitation work and to describe the user requirements.

Here are some general technique guidelines for requirements elicitation and interviews; identify as many stakeholders as possible especially key persons, work with many types of stakeholders handling different areas of work, document everything about requirements, question stakeholders for reasons and purposes of their requirements, but don’t ever judge requirements.

2.3.2 Analysis

When the elicitation of requirements is completed, the next phase which consists of analysis of the collected requirements can be started. An important intention of the analysis phase is to verify all requirements and that they are able to be comprehended, by reviewing the collected requirements and documenting them in a structured way. The documentation should include the stakeholders, their relations, possible conflicts or constraints, and other important information related to the requirements. It is important that requirements are clearly and precisely described in order to understand them without misinterpretations. The documentation may also include the original wording of a requirement, written exactly in the same way as it was expressed or recited by a stakeholder, as well as in the native language to minimize risks associated with translation. The purpose of the requirements document is to present the requirements, their values, bring up conflicts and issues, and demonstrate the received benefits for stakeholders.

The analysis practices should include classifying requirements in different means, e.g. the requirement’s type, whether the requirement is concerning a product or a process, and also if a requirement is a functional or a non-functional requirement. The analysis may also include defining whether requirements are linked to other requirements or a result of them, e.g. higher level ones, and checking for any overlapping requirements or duplicates. Different functional areas may also be specified, i.e. what specific area or functions a requirement concerns, the range of extent and what is affected, and then classified by these attributes. The functional areas are highly dependent on the project and its scope. (Abran et al. 2004)

A commonly used method used in requirements analysis, especially in software engineering, is conceptual modeling, which means developing an abstract concept involving certain parts of the issue. The concept model should represent a real-word scenario, containing the most importantly perceived aspects, while leaving less important ones disregarded. The purpose is to help understanding the requirements and issues before actual implementation or construction. Conceptual modeling has several benefits, e.g. the completed conceptual models can be used for checking specifications and requirements. Models can be used for demonstration to help giving a general understanding of the problem and its structure. Models can be used directly to derive working systems for implementation. Models can help with the testing, debugging and verification work. Development risks may be more easily identified by using models, and models may also be reused as they are generally stable over time. Conceptual modeling may involve development of several different types of models, e.g. models for data and control flow, state, events, user interactions, object models and others (Abran et al. 2004). There are also different methods and systems used for conceptual modeling; a commonly used one is the Unified Modeling Language. (Kandt 2003)

Another thing which in general should be taken into account is requirement priority. Priority analysis is generally based on different factors, e.g. the requirement importance, implementation costs, and relationships or dependencies of requirements. The analysis may provide various benefits, like which requirements should certainly be fulfilled while other less important ones could be postponed for later implementation or skipped completely if necessary. Also during the analysis potential risks may be discovered, which should be reviewed for possible impact consequences concerning the cost, schedule and technical feasibility. (Kandt 2003)

The priority determines how important and critical a requirement is on an overall level for success. Any potential benefits should be assessed and balanced with the priority in comparison with the implementation costs. (Abran et al. 2004)

The requirement priority can be defined in collaboration with stakeholders during interviews, meetings and other methods of surveying, and then classified on different types of scales, e.g. a numerical scale or then in an equivalent way verbally with options for low, medium, high etc. A common problem in prioritizing requirements is that a large amount, or even the majority, of requirements are prioritized relatively high, because of stakeholder's opinions and the way of thinking that all requirements are necessary. A solution for too many highly prioritized requirements is using another method for prioritization, e.g. by comparing and identifying relative values between all requirements. Alternative methods as the Analytic Hierarchical Process (AHP) or the Incomplete Pairwise Comparison (IPC) could be used, or then in a more common way by first defining priority on a higher level of different components, e.g. subsystems or scenarios, depending on their requirement type and relations, and then prioritizing the requirements individually of those components. (Kandt 2003)

2.3.3 Specification

Specification of requirements is the third phase of the standard RE practices, where the specifications, which are formed and derived from the gathered requirements, should provide information of the intended outcome, e.g. a product in form of a software system, i.e. describe what it's supposed to do. All information may not or can't be displayed, but it's important that the essential information for solving the problem is present. Specifications should also be able to be used for demonstration purposes of a solution and its behavior to stakeholders or customers. (Kandt 2003)

As it is commonly known that documentation is really important in many tasks, because of the need to know what has been done and because of the need to keep things on track. The requirements specification practices may therefore include the making of several, but in general three, different types of documents; a system definition document, a system requirements specification and a software requirements specification. The documents are made for specific usage and will consist of requirements separated and categorized by their type. (Abran et al. 2004)

The system definition document could be considered as high-level requirements documentation, which describes the system and its intentions on an overall level. This documentation's target audience is primarily users of the system and customers needing a complete understanding of the system, and should therefore consist of less technically oriented terms. The system requirements specification can be defined as a common specification document, which could be used in any kind of system engineering task, meaning that it's not restricted only to software engineering. Also the software requirements specifications are usually derived from this documentation, as a system may consist of many different components, including e.g. software and hardware, and may therefore need separate documentations of the requirements and specifications. (Abran et al. 2004)

The software requirements specification (SRS) is generally in software engineering the most commonly used type of documentation, which describes the software that is going to be developed. The SRS can be defined as a collection of structured data, which reflects the requirements of the system software. The documentation should include information making it possible to assess development costs, risks and the time needed for development. Use cases may also be included, for describing the behavior and user interactions or scenarios.

The Institute of Electrical and Electronics Engineers (IEEE) association has defined and specified several standards, including standards concerning the SRS. The IEEE standard for Recommended Practice for Software Requirements Specifications (IEEE Std-830-1993) recommends several attributes that the SRS should fulfill; correctness, completeness, consistency, verifiability, prioritization, traceability, modifiability, unambiguousness. (Kazmierczak 2003)

Using many types of specification documents can provide several benefits regarding the documentation, which could be especially useful in large-scale projects with a lot of complexity that requires good documentation. (Abran et al. 2004)

There are general guidelines and principles how requirement specifications should be stated, in order to form a better uniform understanding and structure. Even though the purpose of specifications is to describe the intended outcome, the specifications should not specify how the solution is implemented, e.g. architectural or design decisions of a solution, except if the concerned requirement is of the non-functional type with the intentions to place restrictions on the intended outcome; e.g. specifying which programming language, framework or architecture should be used in this case. (Kandt 2003)

The principles include also that every requirement is completely specified describing the needed functionality, and that all requirement statements are unique. Duplicates should not be allowed, and only one requirement should be included and described per statement, meaning that requirements are possibly needed to be split in several requirement statements. Also the traceability of requirements between higher and lower level ones should be possible in a bidirectional way, including possible relationships with other requirements, because of the need to know if the derived lower level requirements are able to satisfy and fulfill higher level requirements. The requirements should also be verifiable by different methods, e.g. tests, demonstrations, analysis or inspection, so that the intention can be validated. (Abran et al. 2004)

2.3.4 Validation

The validation phase of RE should mainly consist of identification and correction of errors, faults and other mistakes that have gone unnoticed in earlier phases. The purpose is to ensure that the requirements are correct and will certainly satisfy the needs of stakeholders, followed then by approval if the validation and verification procedures are successfully completed and nothing is left unresolved.

The validation practices may include different methods, where requirements reviews are in general one of the most commonly used methods. In a requirements review the requirements and the associated documentation should be carefully inspected with stakeholders and other parties which may have interest in the project. This may include customer representatives or developers, depending on the case. (Abran et al. 2004)

Another common method is prototyping or modeling, where a conceptual model reflecting real-world usage or scenarios is used or built, with the intention to test and validate the requirement. Prototyping may provide several benefits, e.g. checking how something should work like the behavior of user interfaces and interactions, resulting in a better understanding. While there are benefits, prototyping may be expensive and time consuming, and even result in distraction from the main issue while focusing instead on minor problems. (Abran et al. 2004)

Another validation method used could be the attempt to disprove requirements by different tests. A requirement and its statement should describe the problem correctly in its wholeness, and if it would be proved false then it would be vague or inaccurate. (Nuseibeh & Easterbrook 2000)

As already is stated earlier, a requirement should be verifiable, with the possibility to ensure that the needs will be satisfied by the projected outcome. To proceed in the development process, the requirements should be assessed whether they are complete, clear, implementable, and sufficiently acceptable as required. (Hull et al. 2011)

Validation and reaching agreements about requirements may however prove to be difficult because of several reasons, e.g. the information provided and what it concerns is simply not enough for a complete understanding. Possible misunderstandings may lead to changes concerning requirements, where the original intention is altered. Therefore it's also important to review requirements with their originators. A common problem encountered when performing validation practices is the finding of a common consensus and agreement between different stakeholders, due to the reason of conflicting goals and interests. This should be solved through requirement negations, where the goal is to satisfy all the involved parties by identifying each and every win condition and to confirm they are reached. (Nuseibeh & Easterbrook 2000)

2.4 Requirements Engineering Process Models

Different kind of process models have been designed and utilized since the beginning of system and software engineering activities, especially in the commercial side by the rise of the big software giants. The continuous development and ever increasing complexity of software has led to constraints and bottlenecks on the methods used for development. Therefore also many new kinds of process models have been invented for improved development efficiency with the intention to achieve better results.

A couple of the most commonly used standard process models specifically designed for RE include the linear sequential process model, the linear iterative process model, and the spiral model, besides many others including those which are modified or suited to a specific need. Many system and software engineering process models include also phases for RE practices, e.g. the V-Model. Therefore a separate specific RE process may not necessarily be needed. Different types of process models have their own advantages and disadvantages, a certain process model cannot be commonly recommended without knowing the intentions and the details of the project.

In the field of requirements engineering, the ongoing process is typically of the iterative kind. The process continues through the entire project until its end. The industry's needs and desires of faster time to market (TTM) and shorter development periods of products has resulted that most products are in general continuously being developed, step by step, and released as upgrades or revisions which include new features and improvements. When a new revision is released the whole process starts again from the beginning, and may possibly continue until the end of the product life cycle. (Abran et al. 2004)

2.4.1 Linear Model

The linear requirements engineering process model is a very similar model when compared to the standard waterfall model, which has been relatively common in system and software engineering, because of the sequential process steps or phases used where the move to the next phase can be done when the current one is fully completed. The linear RE process model includes five main activities; concept, problem analysis, feasibility and choice of options, analysis and modeling, and requirement documentation (see fig. 1 as an example). This kind of a simple model may be used for smaller-scale projects, where it would be sufficient, but it's not recommended for large-scale projects with a lot of complexity. Several flaws may be associated with the linear RE process model, e.g. the lack of validation, user feedback, and risk management, caused by the sequential model structure which is not having iterations for different phases. (Shams-Ul-Arif et al. 2009)

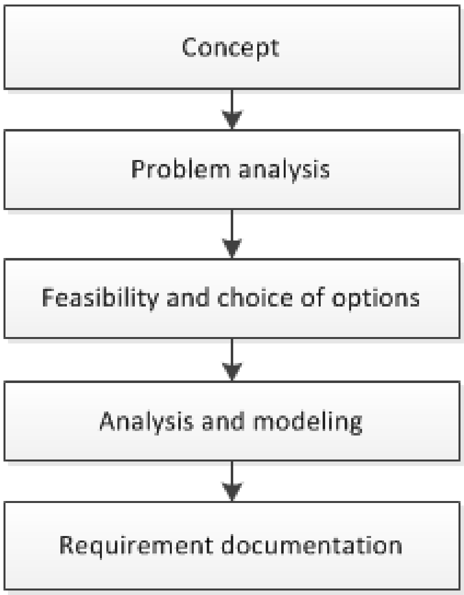

Figure 1. Linear RE Process Model

2.4.2 Linear Iterative Model

The linear iterative RE process model is a further developed version of the linear model including some iteration between the final phases, as demonstrated in fig. 2. The phases involved in this model consist of requirements elicitation, requirements analysis & negotiation, requirements documentation and requirements validation. The main difference between this linear iterative model and the completely linear model is that the validation phase is iterative, meaning it may be done as many times as required until the requirements are validated by the stakeholders and agreed upon as the final system specifications. This ensures better validation of requirements in the process. (Shams-Ul-Arif et al. 2009)

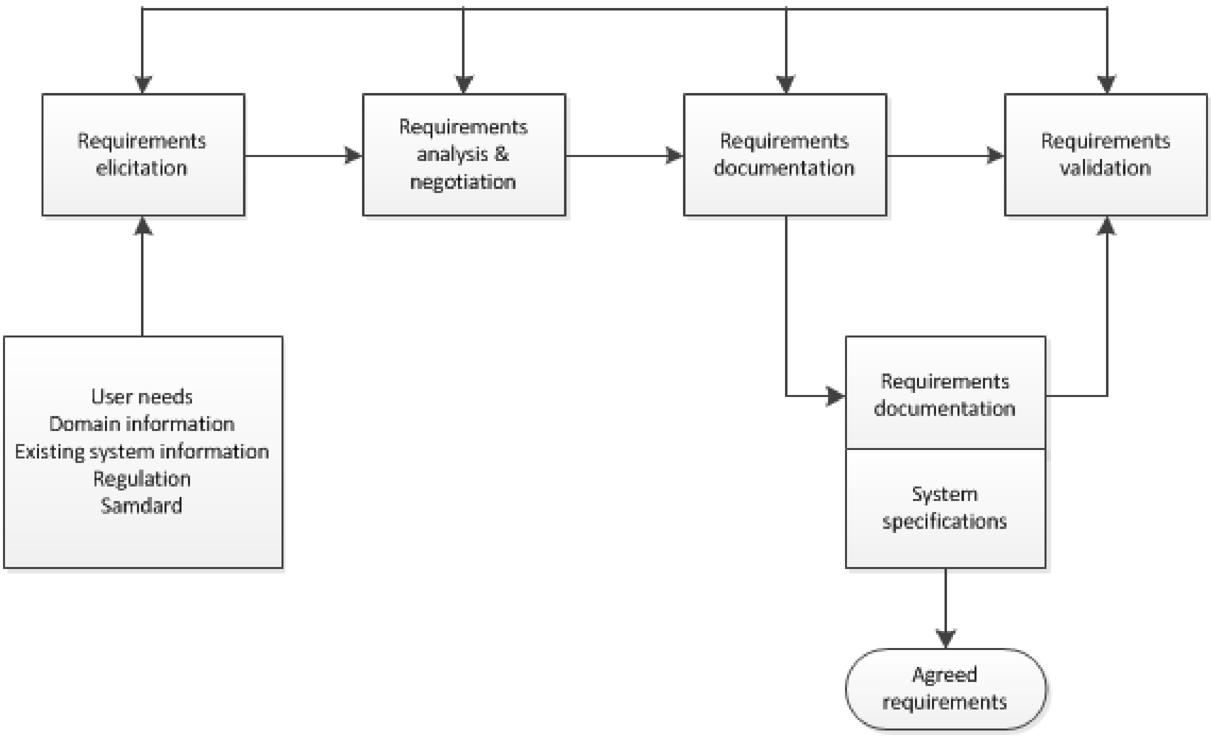

Figure 2. Linear Iterative RE Process Model

2.4.3 Iterative Model

The iterative RE process model is consisting of three main phases; elicitation, specification and validation, but because of the iterative nature of this model, the phases are in general performed multiple times until resolution as shown in fig. 3. This kind of process model is typical for software development, where new revisions of the product are continuously being developed and released, requiring shorter development and release periods. The iterative model incorporates also feedback and information from users and therefore problems have better possibility to be assessed before validation. (Shams-Ul-Arif et al. 2009)

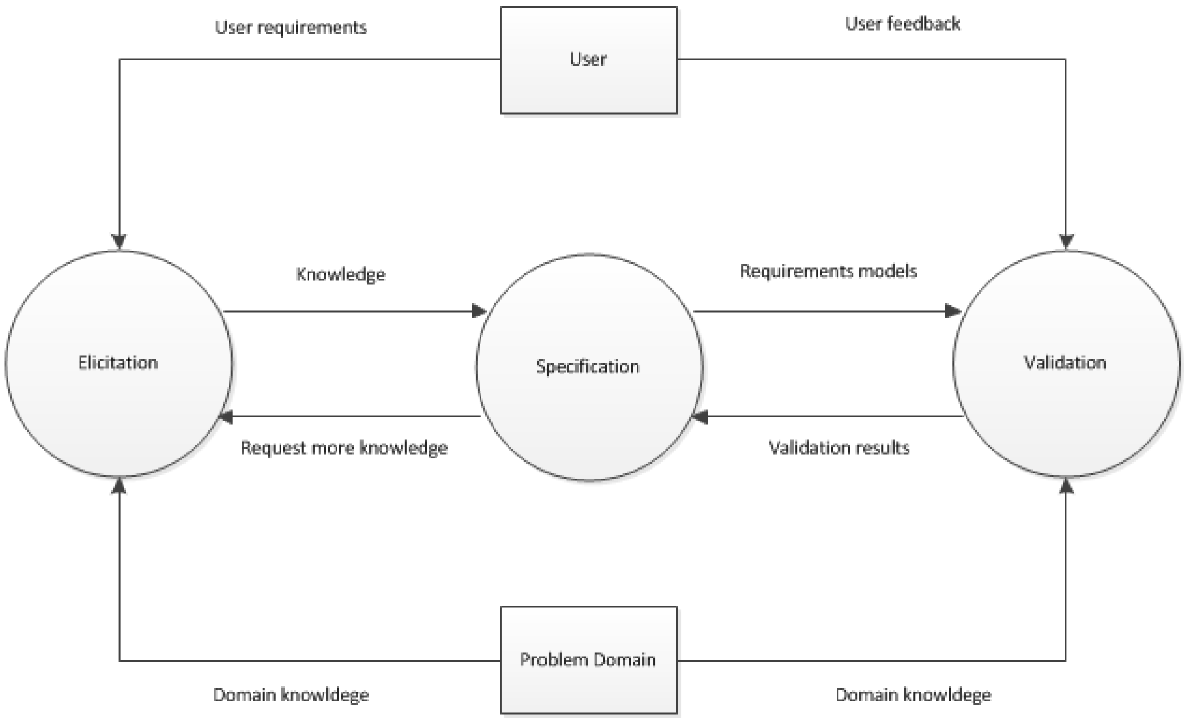

Figure 3. Iterative RE Process Model

2.4.4 Spiral Model

The spiral RE process model consists of four phases, as in the linear iterative RE process model; requirements elicitation, requirements analysis & negotiation, requirements documentation and requirements validation. The spiral RE process model works basically in a linear way, the four phases are all sequentially aligned, but because of the spiral structure as shown in fig. 4, once all phases are sequentially completed, the process starts again from the beginning forming a cycle. Thus it combines some iteration in a cyclic manner with the basic thinking of the linear process model. The spiral model is designed to handle certain things better, e.g. risk, delays requirement changes, which could have consequences affecting the cost, schedule or quality. (Shams-Ul-Arif et al. 2009)

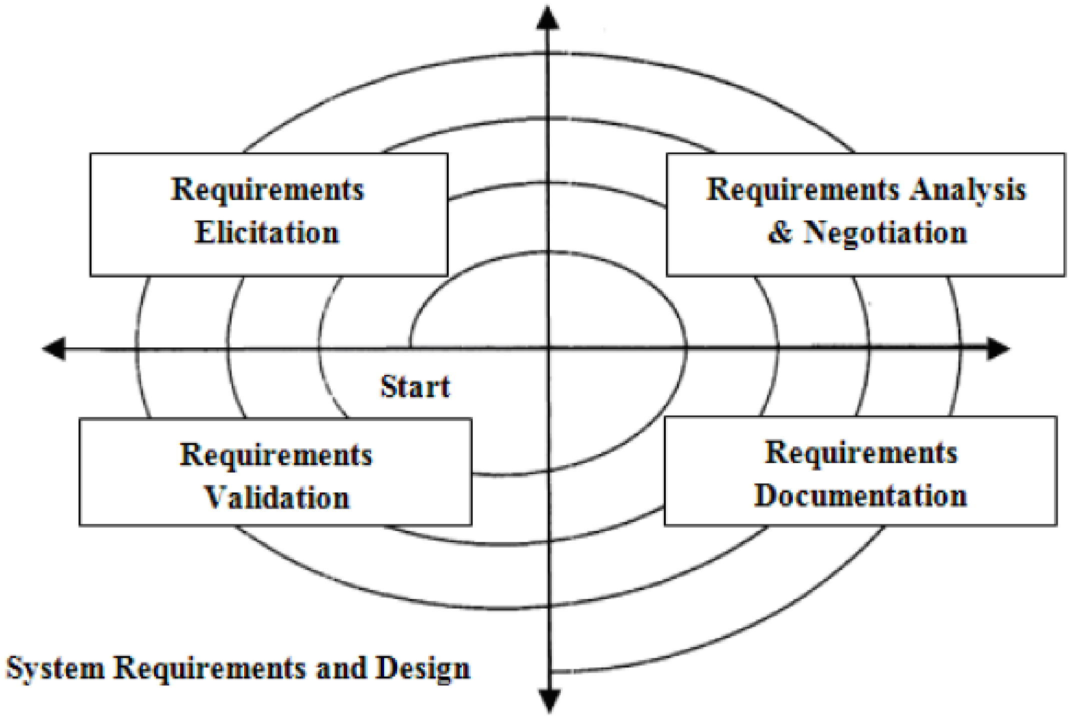

Figure 4. Spiral RE Process Model (Shams-Ul-Arif et al. 2009)

2.4.5 V-Model

The V-model is an adaption of the waterfall model used in system and software engineering. The difference concerns mainly validation, testing, quality and risk management activities, which are handled by the addition of several iterative phases to the process structure. The intention is to continuously carry out testing and verification procedures to improve the quality and minimize risks. (Kazmierczak 2003)

The development phases can be seen as different layers, which progress from the initial high-level stage further down in the process until the lower levels are reached, where the implementation work may be started (fig 5.). Each phase is also always validated back and forth, against the specified requirements, meaning that RE is playing an important role at every development phase. The process phases in the V-Model can vary depending on the intentions and the project, but the basic pattern should stay the same. (Hull et al. 2011)

As an example and for comparison to fig. 5, viewing the V-Model purely as a RE process model, the phases could be defined from the start to the end as: stakeholder requirements, system requirements, subsystem requirements and component requirements, where each phase would include respective tests and validation procedures; acceptance test, system test, integration test, and component test.

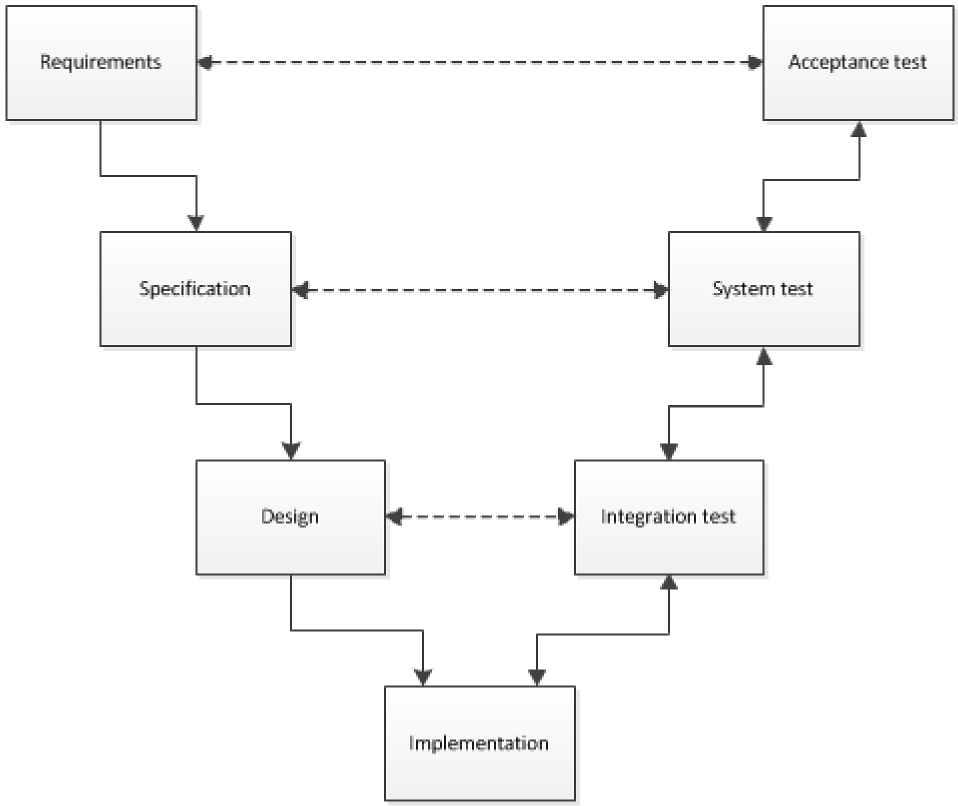

Figure 5. V-Model

2.5 Importance of Requirements Engineering

Requirements engineering may not feel very important when some kind of development project is started, especially if one is inexperienced or missing directives. Therefore RE could easily be neglected completely. Another reason for neglecting RE could be the way of thinking that instead of spending time and effort in requirements engineering, the time could be better utilized in other areas of development, e.g. coding, testing and validation. This kind of approach may work in smaller scale projects, as there are several factors involved which could affect the results. However, as the scale of the project and involved people get larger, the potential risk of something going wrong will respectively grow substantially.

If a problem or mistake, e.g. something concerning the system design or architecture, which affects a lot of different components, is introduced or made early in the development phase but not discovered until later phases, this will generally make the fixing a lot harder and more expensive. Another cause, which may result in higher expenditures when solving issues at later development phases, is the potential need for major changes in several different things. This situation requires more time spent solving the issues and possibly prolongs the project making it to miss deadlines.

A study done by the Standish Group in 1995 and 1996 (Scientific American, Sept. 1994) concludes and assembles different reasons for project failures, as demonstrated in fig. 6. The most common reasons were incomplete requirements (13.1%), lack of users involvement (12.4%), lack of resources (10.6%), unrealistic expectations (9.9%), lack of executive support (9.3%), changing requirements or specifications (8.7%), lack of planning (8.1%), didn't need it any longer (7.5%). The same study shows also project success factors, which were listed as user involvement (15.9%), management support (13.9%), clear statement of requirements (13.0%), proper planning (9.6%), realistic expectations (8.2%), smaller milestones (7.7%), competent staff (7.2%), ownership (5.3%). (Hull et al. 2011 p. 22-23)

The results (see fig. 6) show that the issues were mainly not technically related, but highly related to requirements and their management, the lack of support, resources and other problems concerning the management. (Hull et al. 2011 p. 22-23)

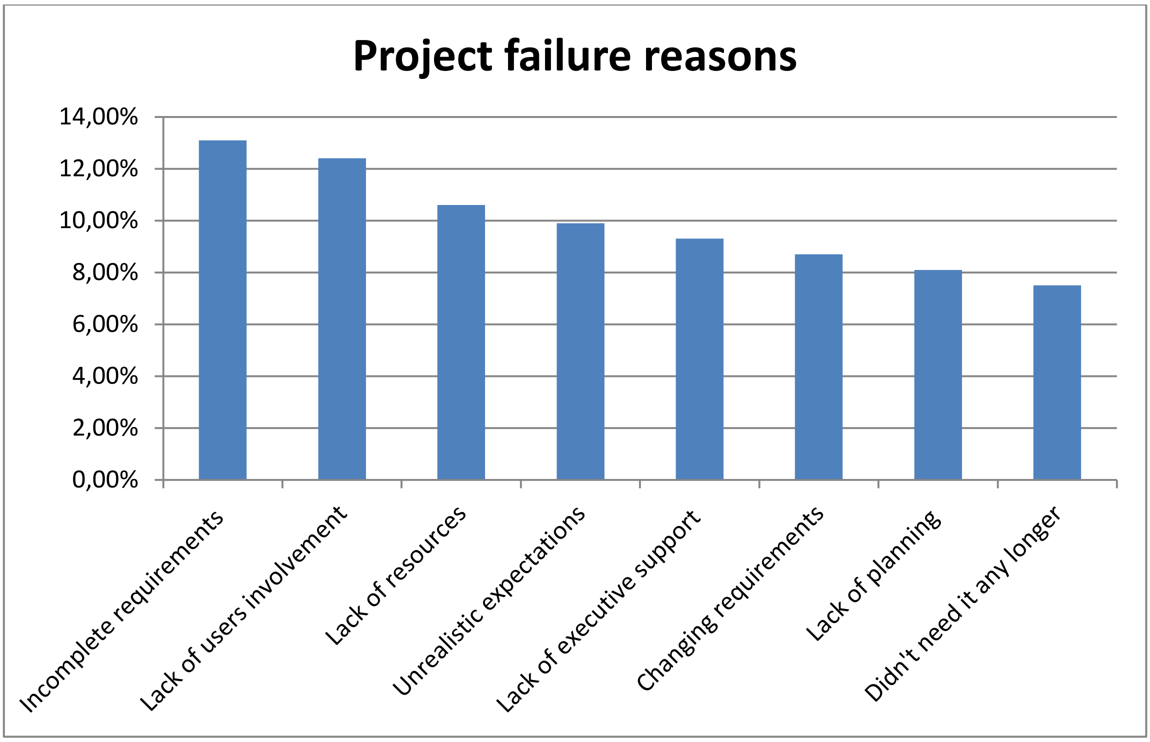

Figure 6. A chart of project failure reasons

Even though there are several RE process models which could be utilized in projects and organizations, studies have shown (Hofmann & Lehner, 2001; Lowe & Eklund, 2001) that in general the majority ignores a clearly defined RE process, because of the reason that RE is thought to be an ad hoc process, i.e. the RE practices are carried out without following a real process and the work is mostly performed by improvisation. (Martin et al. 2002)

In software engineering, the RE process and its practices have also been seen more like a certain part of a software process, which covers the entire project. While requirements and their importance for success is well known, the concept of requirements engineering as a separate knowledge area and engineering practice is not as widely known. (Martin et al. 2002)

3 A REQUIREMENTS ENGINEERING CASE

This thesis work, which is part of a more comprehensive large-scale process and system development project with the intentions to improve quality and efficiency in the current and future processes and operations, included the task to solve different needs regarding a Network Information System (NIS). This was a requirements engineering case, a task to define change requirements.

3.1 The Business Requirement Gathering Process

Because of the reason that the corresponding business unit I have been working for did not have any clearly defined methods or processes for managing requirement specifications, there was a need to define one. It is possible that other business units have readily defined processes for managing requirements, and the practice is also likely to vary due to the large amount of different units. Therefore a new process for this was planned and defined; the Business Requirement Gathering Process (BRGP).

When examining the life-cycle of a project, it should in general consist of several development phases, including possibly several subprocesses with their own phases. From a high level development perspective, in regards to this particular process and system development project which is being carried out, the Business Requirement Gathering Process is located early in the process structure as demonstrated in the Fig. 7 example. The process in this example includes phases like business requirement, IT analyze & design, IT implementation, and finally rollout. The structure itself is also linear meaning that the phases are sequentially executed when examined from a high level viewpoint.

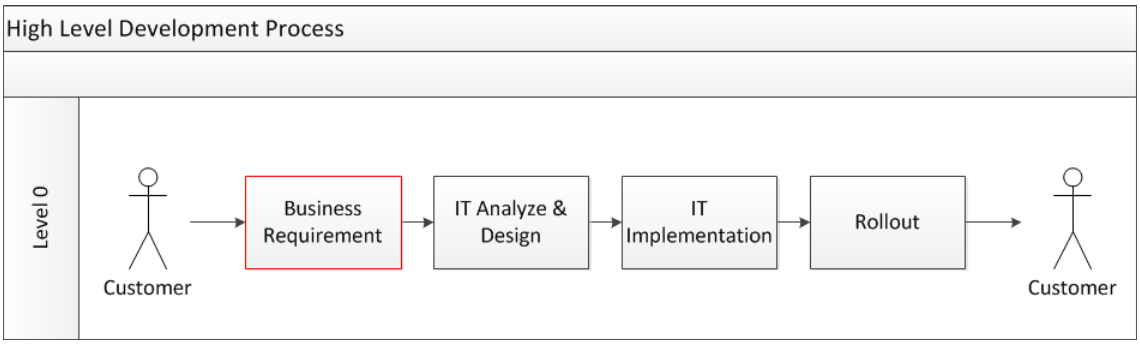

Figure 7. High Level Development Process

The Business Requirement Gathering Process (BRGP) itself was also designed as a sequentially structured process, which consists of four main phases. The included phases are elicitation of requirements, specification of requirements, prioritization of requirements, and validation & approval of requirements. These phases reflect directly to the standard RE practices, i.e. elicitation, analysis, specification, and validation, which were introduced earlier in this thesis. Fig. 8 shows the overview of the BRGP structure from a high level viewpoint, called level 1. When examining the process and its different phases in a greater detail, we proceed gradually to the following levels in the process structure.

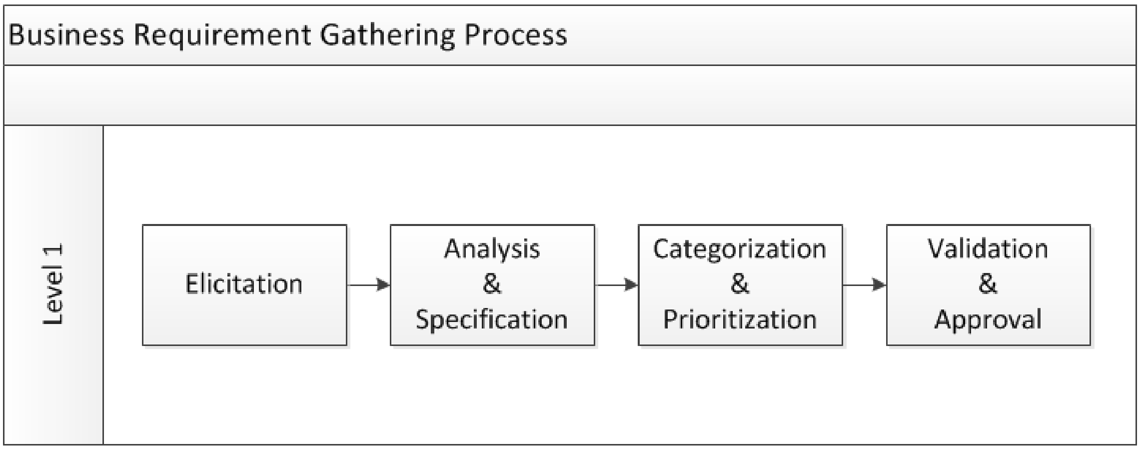

Figure 8. The Business Requirement Gathering Process

3.1.1 Elicitation of Requirements

The first phase of the BRGP is elicitation of requirements. The overview of the steps and included persons are demonstrated in Fig 9. The purpose in this case is to solve the different business needs by gathering as many requirements, and related information, as possible. The business needs and requirements will then be refined and clarified in a way that they can be later solved or implemented by different means. Several different methods to obtain and gather requirements may be used, although the elicitation task should be carried out mainly through interviews and meetings with different stakeholders and other participants of the project.

One of the most important tasks in this phase is to identify stakeholders, which are playing an important role regarding the project. These stakeholders are the key persons, e.g. usually main users, experts, and process or task owners, with extensive knowledge of their own work area. Therefore they should be included in interviews to gain the most broad and accurate information possible. Resources must be allocated, i.e. stakeholder interviews and meetings should be arranged and scheduled according to the availability and time constraints.

By the end of this process phase, a detailed documentation of the collected initial requirements and related data should be available for further usage. Also an initial listing of business benefits should be included, where each and every requirement should contain the information and reasons why it is needed and what kind of advantages it would potentially bring, especially from a business viewpoint if possible, e.g. estimated cost savings or efficiency related improvements in the operations. At this point it is advisable to have as much information and details as possible documented, unnecessary and redundant information may be trimmed later on in the specification phase.

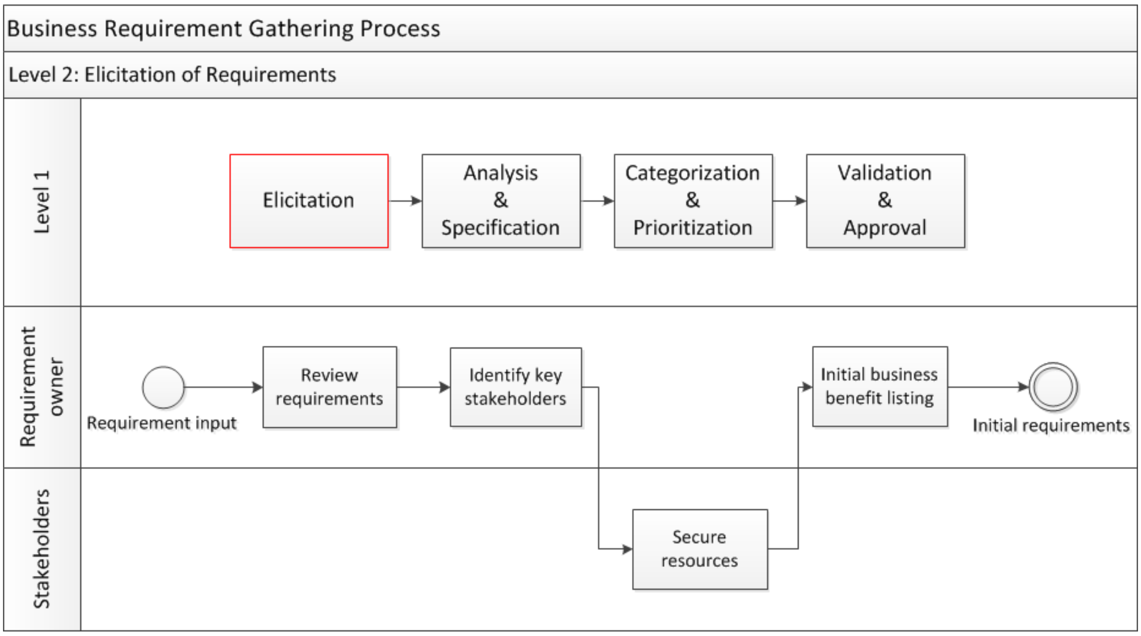

Figure 9. Elicitation of Requirements

3.1.2 Analysis and Specification

The next phase of the BRGP is analysis and specification of requirements, where the purpose is to analyze and specify the gathered initial requirements, from the elicitation phase, to assure a complete understanding and verification of the requirements. Detailed explanations of the requirement descriptions should be added to the documentation, as well as corrections or changes if necessary, e.g. as some requirements may need to be split into several ones. New requirements may yet be proposed and added, and further additions of new requirements may even be allowed until the end before final validation and approval if it's appropriate and important to the case.

The analysis and specification phase is meant to be carried out by performing additional requirement reviews, interviews and meetings with the stakeholders, including business analysts and controllers, as shown in fig. 10 demonstrating the process flow. An important thing is again to secure resources by organizing meetings and scheduling them efficiently so that all needed stakeholders could attain and be heard.

According to the process flow in fig. 10, one of the tasks is to create detailed descriptions, including explanations and clarifications, of the requirements in conjunction with business analysts. The business analysts are generally also classified as stakeholders, because they also have an interest in the project but from a different viewpoint, e.g. compared to users of a system that is developed. The next step will lead to analysis of costs and benefits and calculation of business benefits. These assessments are required to be able to further prioritize the importance of requirements and potentially attained benefits compared to implementation costs.

At this point and prior to final validation in this phase, the requirements and their scope are also meant to be verified by a second round of meetings or interviews with the requirement originators, the persons who originally proposed requirements. The intention is to review the requirements and related information documented and possibly correct faults, and to obtain approval from the originators of the requirements.

After completion of the analysis and specification phase of the BRGP process, a structured documentation of the requirements and the related information collected should be in good shape and understandable by third persons. It's still important to document a bit more than may seem necessary, although the documentation should stay structured and appear refined at this point. The documentation is important in order that the requirements and their meaning stay correct from the beginning to the end, and that they are easily understood so that they could be demonstrated and further utilized.

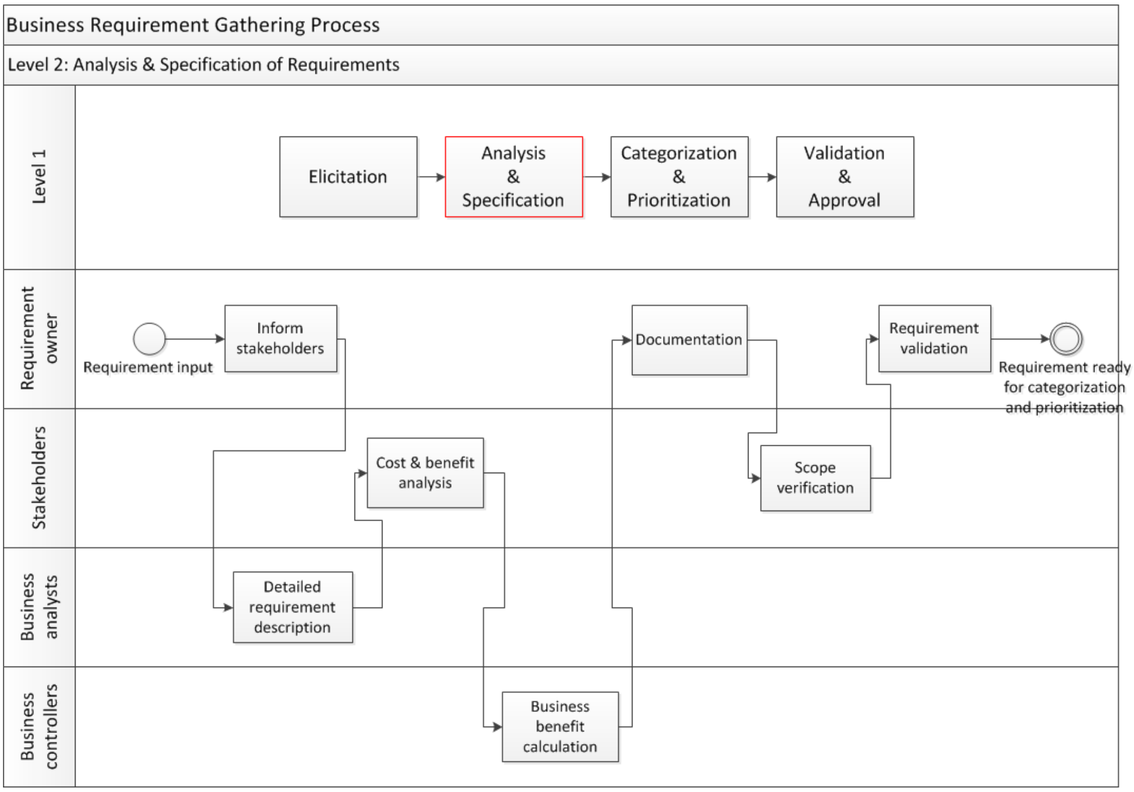

Figure 10. Analysis & Specification

3.1.3 Categorization and Prioritization

The third phase in the BRGP is categorization and prioritization of requirements. As shown in fig. 11 which demonstrates the process flow, as well as the name implies, this phase consist mainly of meetings with the stakeholders to categorize and prioritize the requirements. This should again be carried by arranging and scheduling meetings with the required persons, which may include many different types of stakeholders, ranging e.g. from users and system experts to persons from the IT- development and implementation side.

As this thesis has already described different RE practices, categorization and classification of requirements may be done in several different ways, depending on the project, its scope, and the intentions. It should be at least advisable to define the type of the requirement, e.g. is it a business requirement, process requirement or a system/product requirement. It could also be beneficial for categorization reasons to specify in detail what specific functional areas a requirement concerns, i.e. the scope.

When the requirements have been categorized, a requirements prioritization should follow. The prioritization meeting should also be done in collaboration with the corresponding stakeholders, in this case the prioritization board, which could e.g. consist of developers, system experts, and business analysts, as shown in fig. 11. Prioritization is an important task that provides several benefits, especially in large-scale projects, when the costs, benefits, and schedules among other things related to the project are assessed and determined.

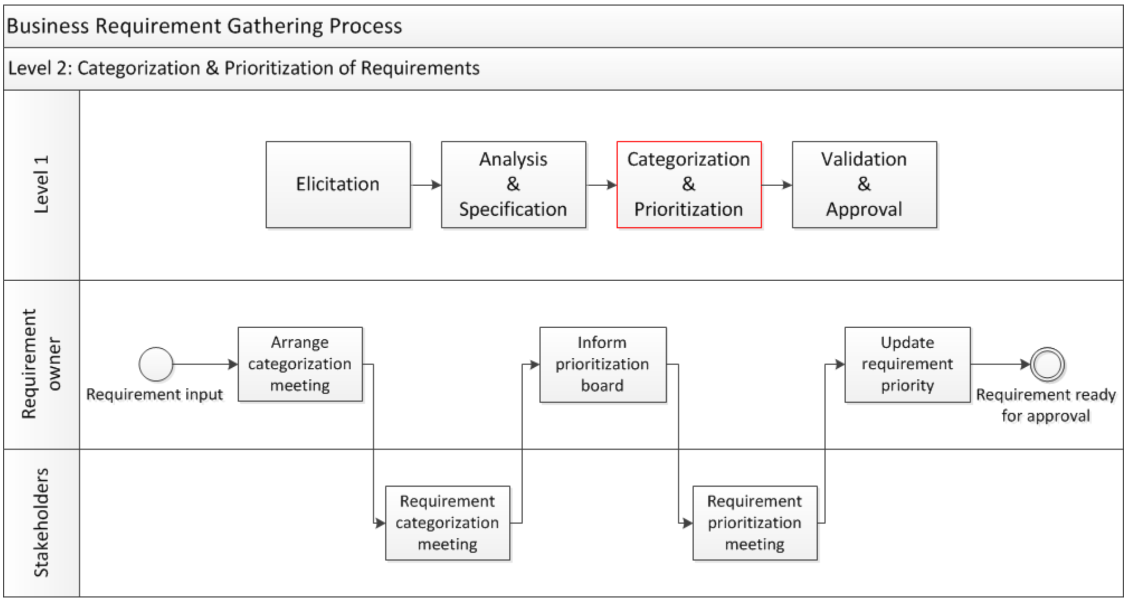

Figure 11. Categorization & Prioritization

3.1.4 Validation and Approval

The fourth and final phase of the BRGP is validation and approval. The main function of this phase is in verifying the gathered requirements with the business management, in order to get approval. After the requirements have been successfully validated, it will lead to the end of the BRGP, and the project will continue according to the high level process used, e.g. as in this case with IT analysis and design followed by implementation. The intention of this validation and approval phase is to function as a final control of the requirements.

The validation work of this phase should be carried out as shown in fig. 12, starting by informing the business management and ending with a scheduled approval meeting. During the approval meeting the gathered requirements including the documentation should be presented, in order to give a proper overview of the case. The requirements should then be reviewed thoroughly, one by one, where a requirement is either approved or invalidated/ignored with possible further instructions. After this, action should be taken according to the decisions made during the meeting.

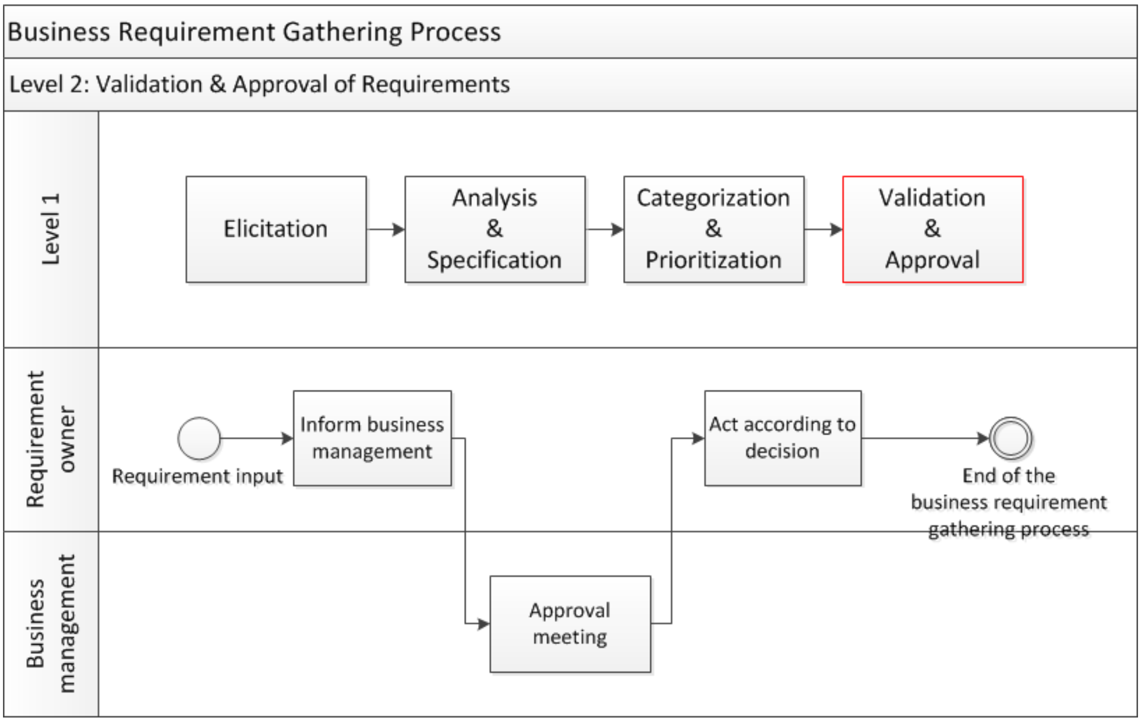

Figure 12. Validation & Approval

3.2 Compilation of the Requirements Catalog

As this thesis work also included the task to identify different needs by collecting business requirements regarding a Network Information System (NIS) and to compile the requirements into a formally structured requirements catalog, the newly defined Business Requirements Gathering Process was used and the RE practices was carried out according to the process. Although the BGRP was also developed and improved during the whole process, based on practical experience, results and feedback.

The first step was to gather business requirements through elicitation. The requirements were gathered by interviews and during meetings, which were organized with various stakeholders, usually from different business sections, seen and identified as the key persons of their knowledge area. In case a meeting required a person with another locality where they could not physically attend because of the traveling distance, the meeting was held using available net meeting tools.

After the identified stakeholders had been heard and interviewed, and the initial business requirements seemed appropriate at this stage, the next phase of the process could be started. This phase consisted of analyzing the requirements and specifying them in a greater detail, e.g. the description and the explanation of requirements was improved, the requirements were named in a suitable way, and other things concerning the documentation were improved. It was important to document the requirement owner, i.e. the person who originally suggested the requirement, in order to be able to trace back to the person if there was a need for additional information. Another essential matter was the estimation of potential benefits, as it is important to have a valid reason or justification for every requirement, which will also make prioritization easier. Most of the practices of this phase were performed in collaboration with the stakeholders, during a second round of interviews and meetings, where also the requirements and the work up to this point was verified by the original requirement owners.

The next phase consisted of categorization and prioritization of requirements. The task was performed in collaboration with various stakeholders, particularly with stakeholders from the business management and the IT architecture, design, and implementation sections.

Once the requirements were categorized and prioritized, it was time for the final phase, which was to validate the requirements and to review the entire requirements catalog. The task included to determine how each requirement should be handled, as well as to specify what the next step was. The entire requirements catalog and the related work was completed after the requirements had been validated and approved, resulting in ending of the process.

4 CONCLUSIONS

In software engineering it is common that a large amount of time is used for quality and assurance testing, to correct and fix detected flaws and issues. The later an issue is detected or a request for some kind of a change is made during a software engineering or development project, the more it will take time and resources to fix or implement it. Project failures are necessarily not caused by technical issues, the cause is in general more likely related to the management, requirements and resources.

In order solve different needs placed by the business it is highly desirable to have clearly defined goals and specifications, what is really required from the system, its users and other stakeholders. The needs and requirements must be established before development. The outcome of a project should correspond to the demands and requirements of stakeholders if it’s going to be qualified as a successful investment. Therefore high quality requirements and specifications are critical in creating a successful product that will meet user expectations. Potential benefits, risks and other factors associated with the requirements placed by the business should be assessed and balanced altogether before actual system and software development and implementation work.

Requirements engineering and its practices, which may be considered as a subdiscipline of systems and software engineering, is an important knowledge area providing several benefits when used properly in an engineering task. The purpose of requirements engineering is to clarify the needs and requirements, by bringing together different stakeholders, users affected, or anyone involved with some kind of interest in the task. The requirements may be gathered through elicitation and by using several different approaches. Other practices involved in requirements engineering is generally analysis, specification and validation of requirements.

Requirements engineering should be a process driven practice, and the same should in general apply to any kind of project work, with a clearly defined and structured process plan which is meant to be followed. The processes used may depend on the project, its scope, and several other factors. Any guidelines or recommendations of a suitable process for a specific project cannot be given without a background study of the project and what it involves.

One of the main objectives of this thesis work was to find out and identify what kind of needs and requirements various businesses and departments of a company have for a Network Information System. Because of the reason that there were not any specifically defined processes for similar requirements engineering tasks, it was also included to define a suitable process for the company. This was accomplished by the Business Requirement Gathering Process, which also follows and makes use of common standard requirements engineering practices. The process was used for the requirements engineering task in this thesis work, and the resulting requirements were documented and compiled into a formally structured requirements catalog. The requirements catalog, including all information gathered and compiled, was validated and finalized before being passed on to the IT architects.

REFERENCES

Abran, Alain; Moore, James; Bourque, Pierre; Dupuis, Robert (Eds.). 2004, *Guide to the Software Engineering Body of Knowledge*, IEEE Computer Society.

Hofmann, Hubert & Lehner, Franz. 2001, *Requirements Engineering as a Success Factor in Software Projects*, IEEE Software, Vol. 18, Issue 4, p. 58-66. Available: http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=20258 / http://www.das.ufsc.br/~romulo/discipli/cad-meto/requi.pdf Accessed 9.10.2012.

Hull, Elizabeth; Dick, Jeremy; Jackson, Ken. 2011, *Requirements Engineering*, 3rd Ed, Springer.

Kandt, Ronald, 2003. *Software Requirements Engineering: Practices and Techniques*, JPL Document D-24994. Available: http://whalen.ws/index_files/JPL_SW_Reqmts_Engr_D-24994%5B1%5D.pdf Accessed 11.9.2012

Kazmierczak, E, 2003. *Requirements Engineering*, 433-641, The University of Melbourne. Available: ww2.cs.mu.oz.au/~dmwilm/downloads/641.pdf Accessed 11.9.2012.

Lowe, David & Eklund, John. 2001, *Development Issues in Specification of Web Systems*, 6th Australian Workshop on Requirements Engineering, University of New South Wales, Sydney, Australia. Available: http://services.eng.uts.edu.au/~dbl/archive/2001-Low01e.pdf Accessed 9.10.2012.

Martin, Sacha; Aurum, Aybüke; Jeffery, Ross; Paech. Barbara. 2002, *Requirements Engineering Process Models in Practice*. CEUR Workshop Proceedings. Published 2002. Available: http://ceur-ws.org/Vol-69/paper10.pdf Accessed 11.9.2012.

Nuseibeh, Bashar & Easterbrook, Steve. 2000, *Requirements Engineering: A Roadmap*, Imperial College London, Department of Computing, University of Toronto, Department of Computer Science. Available: http://www.doc.ic.ac.uk/~ban/pubs/sotar.re.pdf Accessed 11.9.2012.

Project Management Institute (Inc.). 2004, *A Guide to the Project Management Body of Knowledge*, 3rd Ed. PMI Publications.

Shams-Ul-Arif; Khan, Qadeem; Gahyyr, S. A. K. 2009, *Requirements Engineering Processes, Tools/Technologies, & Methodologies*. International Journal of Reviews in Computing. Available: http://www.ijric.org/volumes/Vol2/6Vol2.pdf Accessed 11.9.2012.

APPENDIX: REQUIREMENTS CATALOG TEMPLATE