VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Sini Hautamäki

MANAGING REQUIREMENTS OF STANDARDS AND REGULATIONS FOR ENGINE CONTROL SYSTEM

Case Study of Storing System Requirements from DNV in Polarion

Technology and Communication
2015

VAASAN AMMATTIKORKEAKOULU
Kone- ja tuotantotekniikan koulutusohjelma

TIIVISTELMÄ

Tekijä	Sini Hautamäki
Opinnäytetyön nimi	Moottorin ohjausjärjestelmän standardien ja säädösten vaatimusten hallinta. Tapaus DNV:n järjestelmävaatimukset Polarionissa.
Vuosi	2015
Kieli	englanti
Sivumäärä	64 + 2 liitettä
Ohjaaja	Pekka Ketola

Tämä opinnäytetyö on tehty Wärtsilän Ship Power yksikön alaisuudessa toimivalle Automation & Controls –osastolle. Toimeksiantona oli kehittää hiljattain käyttöön otettuun ohjelmiston elinkaaren hallintatyökaluun, Polarioniin, rakenne luokituslaitosten esittämien vaatimusten hallintaan. Rakenteen tulisi mahdollistaa luokituslaitosten esittämien sääntöjen ja säädösten tulkitseminen vaatimuksiksi ja siirtäminen osaston normaalin prosessin mukaiseen toteutukseen.

Luokituslaitokset ovat tärkeitä laivanrakentajien yhteistyökumppaneita. Ne vastaavat osaltaan merenkulun turvallisuudesta määrittelemällä sääntöjä ja säädöksiä koskien alusten teknisiä rakenteita. Luokituslaitokset ovat organisaatioita, jotka vastaavat alusten katsastuksista ja tekevät myös yhteistyötä laivanrakentajien kanssa suunnitteluvaiheesta alkaen. Luokituslaitosten vaatimusten hallinnassa täytyy huomioida säädösten muuttuminen uusien sääntöjulkaisujen myötä sekä vaatimusten toteutuksen jäljitettävyys laitosten myöntämien sertifikaattien anomista varten.

Tässä työssä käsitellään vaatimustenhallintaa ja sen merkitystä yleisesti, määritellään luokituslaitosten vaatimusten hallintaan tarvittava rakenne Polarioniin sekä tehdään esimerkki rakenteen käytöstä syöttämällä pieni osakokonaisuus säädöksistä ohjelmaan uutta rakennetta käyttäen. Kehitettävä rakenne perustuu sekä teoreettiseen tietoon, että osaston työntekijöiltä saatuihin tietoihin ja toiveisiin. Rakennetta suunniteltaessa on myös otettu huomioon jo valmiina olevat rakenteet Polarionissa ja uusi rakenne mukautettiin toimimaan niiden kanssa.

Avainsanat vaatimustenhallinta, luokituslaitos, Polarion

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Kone- ja tuotantotekniikan koulutusohjelma

ABSTRACT

Author	Sini Hautamäki
Title	Managing Requirements of Standards and Regulations for Engine Control System. Case Study of Storing System Requirements from DNV in Polarion
Year	2015
Language	English
Pages	64 + 2 Appendices
Name of Supervisor	Pekka Ketola

The thesis was done for Wärtsilä Finland, Automations & Controls department which is located under Ship Power organisation. The assignment was to create a structure to manage standards and regulations coming from classification societies. The structure was to be implemented into recently adopted application lifecycle management tool, Polarion. The new structure should enable translating classification rules and regulations into requirements and enable transmitting these requirements into the normal development process of the department.

Classification societies are important partners for ship builders. They share the burden of maritime safety by establishing rules and regulations concerning technical structures of vessels. Classification societies are organisations responsible on vessels statutory surveys but also cooperate with ship builders from the start of design. Managing the requirements coming from classification societies is a task of taking into account that rules and regulations change over time when new rules are established and the implementation of the requirements derived from the rules has to provide traceability in order to receive the certification from the societies.

This thesis covers the topics of requirements management in general, defining the structure, the concept for managing the requirements coming from classification societies in Polarion and making a proof of concept by importing a specified part of these rules into Polarion by using the new structure. The concept created is based on theoretical knowledge of requirements management and the knowledge and wishes shared by classification experts. The existing development process was taken into account and the new structure was adopted to fit into it.

Keywords: Requirements management, requirements engineering, classification society, Polarion

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ALM	Application lifecycle management
A&C	Automation & Controls
C&P	Certification & Processes
DNV GL	Det Norske Veritas Germanischer Lloyd, a classification society
QA	Quality assurance
RFC	Request for change, work item used in Polarion
R&D	Research & Development
SteCo	Steering committee
WoCo	Work coordination

LIST OF FIGURES

LIST OF TABLES

APPENDICES

1 INTRODUCTION

Changes and escalating pace of changes are nowadays reality for businesses around the world. The up-to-date knowledge is a vital condition for the operation of the company and for the expertise of personnel. The right knowledge must be available when needed, where needed. This creates a need for information systems and procedures which can supply the knowledge and expertise for the whole organisation.

A company operating in an international environment faces many requirements. The requirements are usually demands and expectations for the process and products. The large quantity of requirements provokes a need for requirements management and procedures to ensure that the requirements are supplied for the right people and correctly implemented.

What comes to managing all the requirements related to building a ship or, like in this case, building engines for large vessels, the work load is most likely enormous. Quantity and granularity of requirements can be high. Requirements come from several different stakeholders and should be forwarded to several different places for implementation. Most importantly, all the requirements should be traceable at any time.

1.1 Background

This thesis was done as an assignment for Wärtsilä Finland plc, Ship Power. The outcome is meant for the Certification and Processes team of Automation & Controls (A&C) department within Ship Power Engines Research and Development (R&D) business unit. The Certification and Processes team is responsible for regulating the processes, quality aspects and certifications related to the development of the engine automation software and control units.

At the moment a new application lifecycle management application, Polarion, is being deployed in A&C. The application is highly editable and it needs to be modified to suit the defined use. All templates and procedures can and should be adjusted to meet the needs of the department. The first part of this process is to manage the requirements coming from different stakeholders. These stakeholders can be for example, internal customers from other departments or external operators like classification societies. For the appropriate use of Polarion there needs to be specifications for templates, workflows and other items. It is typical for this kind of industry that there are multiple standards, authorities and other stakeholders presenting requirements for processes and products of marine engine manufacturing. Now when the development of requirements management is in action, it became clear that there needs to be a separate way of managing requirements coming from classification societies. Classification societies are large regulators in maritime industry. They provide rules and regulations for ship building and control initial statutory surveys which usually are demanded for insurance of ships.

1.2 Objectives

The objective of this thesis was to create a model in Polarion to manage requirements coming from classification societies. This includes identifying, documenting, utilizing and tracing of requirements as well as creating the appropriate template and workflow for this particular type of requirement. One aspect to the assignment was to find an efficient and easy way to use the application to fit into present development process of the department. For the company view the interest was mainly on practical outcomes of the thesis which were stated:

- Identification and implementation of the high-level containers and work item type for handling of standards and regulations coming from classification societies.

- Creation of a way to transfer classification requirements into implementation to the system level.

1.3 Wärtsilä Finland plc

Wärtsilä Finland plc is a corporation which manufactures and services power sources and other equipment in the marine and energy markets. Wärtsilä produces a wide range of low- and medium-speed diesel, gas and dual- and multi-fuel engines for marine propulsion, electricity generation onboard ships and for land-based power stations. /20/

Wärtsilä is a global leader in complete lifecycle power solutions for the marine and energy markets. Wärtsilä's mission is to provide lifecycle power solutions to enhance customers' business by creating better technologies that benefit both the customer and the environment. In 2014, Wärtsilä's net sales totaled EUR 4.8 million with approximately 17,700 employees. The company has operations in more than 200 locations in nearly 70 countries around the world. /20/; /22/

Wärtsilä Finland is divided into three main businesses; Power Plants focusing on offering power station solutions on the energy market, Ship Power focusing on the marine market and developing four stroke engines for both industries, and Services which offers maintaining and customer services to both markets. The revenue of Wärtsilä Finland from the year 2014 and how it is divided among these three businesses is presented in Figure 1. /20/

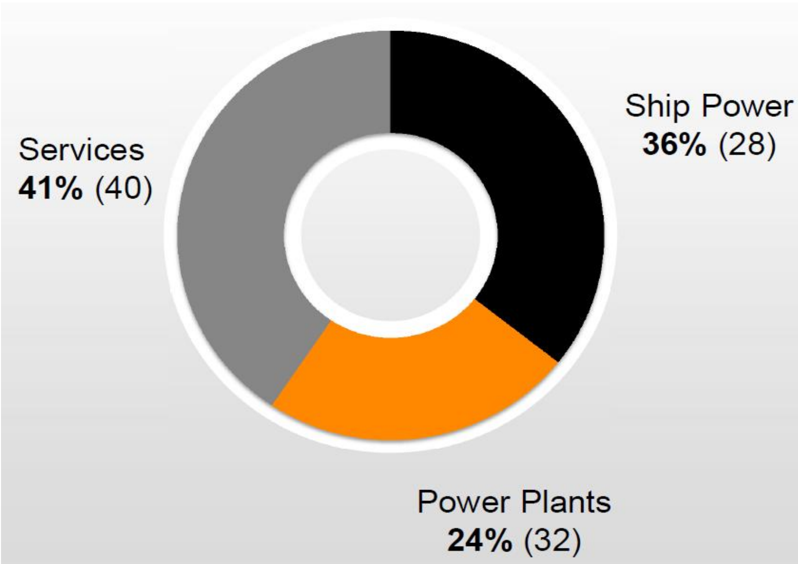


Figure 1. Wärtsilä net sales by businesses 2014. /22/

1.3.1 History

The history of Wärtsilä dates back to the year 1834 when a Wärtsilä sawmill was established in the municipality of Tohmajärvi. The history of Wärtsilä in Vaasa starts from the year 1894 when Onkilahti engineering workshop was established. Onkilahti engineering workshop started as a repair workshop and later expanded to shipbuilding among other industries. Due to financial problems it was bankrupt the ear 1935 and ended up to be owned by Wärtsilä the next year. Wärtsilä continued operation of the engineering workshop and started manufacturing diesel engines for ships there at year 1950. /21/

Over the years the company has changed its name several times. 1990 Wärtsilä merged with Lohja which also consisted of different kinds of industrial plants of metal and construction industry. From 1990 to 2000, the company was named Metra and 2001 onwards Wärtsilä again. Over the years, the company has owned several kinds of industrial holdings, but from the year 2001 onwards, the company has concentrated on large two and four stroke engines. During the 21th century, Wärtsilä has expanded around the world and did the decision to develop the services which ended up being highly profitable decision as nowadays services is responsible for the largest past of the revenue. /21/

1.3.2 Ship Power

Ship Power's strategy is to serve both shipyards and ship owners. It enhances the business of its marine and oil & gas industry customers by providing innovative products and integrated solutions that are safe, environmentally sustainable, efficient, flexible, and economically sound. Being a technology leader it is able to customize solutions that provide optimal benefits to the clients around the world. Wärtsilä Ship Power delivers everything from a single product to entire lifecycle support, from initial building to operational use, of complex systems powering ships. Figure 2 presents the parts Ship Power consists of and their percentages of production. /23/

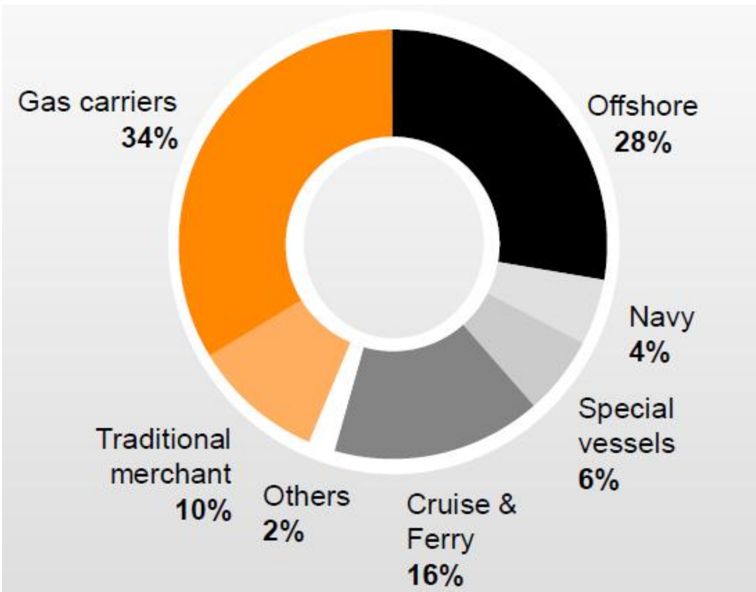


Figure 2. Ship Power production percentages by branch of industry. /23/

1.3.3 Automation & Controls

The department of Automation & Controls is a part of Ship Power Research & Development business unit of Wärtsilä. A&C is responsible for developing the engine automation system needed for controlling functions like combustion, cooling and ignition in engines. The main work is to develop software applications and hardware. In Figure 3 can be seen the organisational structure of Automation & Controls department.

The department produces applications and hardware modules for controlling the combustion and other functions in engines. This includes lots of safety aspects which mean lots of requirements coming from external stakeholders. The department consists of seven teams all responsible for producing their own application types or modules or supporting the development. The applications and modules are later composed into packages to suit different engines before delivering them to the customers. Applications and modules are also evaluated further and same application probably has several revisions. Due to the modular structure of the engine control system and the nature of constructing packages managing all these requirements can be challenging. /1/

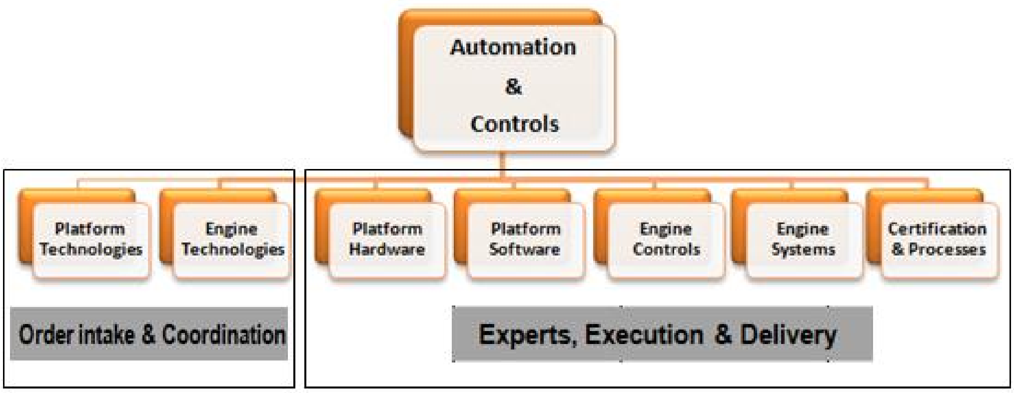


Figure 3. Organisational structure of Automation & Controls department. /1/

The responsibility of the Certification and Processes team is to supervise quality aspects, to coordinate processes, tools and guidelines used in A&C and to take care of certifications of the engine control system. As the new tool is an application designed to help application lifecycle management, it is clear that the deployment of the tool must be coordinated by this team. /1/

1.4 Classification Societies

Classification societies are organizations which provide classification and statutory services and assistance to the maritime industry and regulatory bodies. Their main concern is regarding maritime safety and pollution preventing, based on the accumulation of maritime knowledge and technology. The basic principle of classification rules is to establish clear, demonstrable and verifiable goals for ship builders to aim at. As an effect to previous a properly built, operated and maintained ship should provide minimal risk to its cargo and crew and to the environment for a specified operational life. /3/

The objective of ship classification is to verify the structural strength and integrity of essential parts of the ship's hull and its appendages, and the reliability and function of the propulsion and steering systems, power generation and those other features and auxiliary systems which have been built into the ship in order to maintain essential services on board. Classification Societies aim to achieve this objective through the development and application of their own Rules and by verifying compliance with international and national statutory regulations. /3/

As an independent, self-regulating, externally audited body, a Classification Society has no commercial interests related to ship design, ship building, ship ownership, ship operation, ship management, ship maintenance or repairs, insurance, or chartering. /3/

1.4.1 History

In the second half of the 18th century, marine insurers, based at Lloyd's coffee house in London, developed a system for the independent technical assessment of the ships presented to them for insurance cover. In 1760 a Committee was formed for this purpose, the earliest existing result of their initiative being *Lloyd's Register Book* for the years 1764-65-66. At that time, an attempt was made to 'classify' the condition of each ship on an annual basis. The condition of the hull was classified A, E, I, O or U, according to the excellence of its construction and its adjudged continuing soundness. Equipment was G, M, or B: good, middling or bad. In time, G, M and B were replaced by 1, 2 or 3, which is the origin of the well-known expression 'A1', meaning 'first or highest class'. /3/

During 19th century the concept of classification slowly spread to other countries and insurance markets. As the classification profession evolved, the practice of assigning different classifications has been superseded, with some exceptions. Today a vessel either meets the relevant Class Society's Rules or it does not. The Titanic disaster in 1912 brought safety at sea to the forefront of public concern. International classification societies played an important part in discussions on ship safety. /3/

1.4.2 Scope of Classification

Classification Rules are developed to establish standards for the structural strength of the ship's hull and its appendages, and the suitability of the propulsion and steering systems, power generation and those other features and auxiliary systems which have been built into the ship to assist in its operation. The vast majority of commercial ships are built to and surveyed for compliance with the standards laid down by Classification Societies. These standards are issued by the Society as published Rules. A vessel that has been designed and built to the appropriate Rules of a Society may apply for a certificate of classification from that Society. /3/

Implementing the published Rules, the classification process consists of technical review of the design plans and related documents, attendance throughout the construction of the vessel and its key components to the trials of the ship. Upon satisfactory completion of the above the assignment of class may be approved and a certificate of classification issued. Once in service, the owner must submit the vessel to a clearly specified program of periodical class surveys to verify that the ship continues to meet the relevant rules. /3/

1.4.3 DNV GL

DNV GL is the world's leading classification society and a recognized advisor for the maritime industry. The main concern of DNV GL is to guide safety, quality, energy efficiency and environmental performance of the global shipping industry, across all vessel types and offshore structures. /6/

The newly formed DNV GL Group became operational on September 12, 2013. Changes in ownership and strategic alignment between the two companies and their leadership provided new opportunities, and the merger was finally successful. The DNV GL Group comprises approximately 16,000 employees operating in over 100 countries. DNV GL's shared roots stretch back to 1864, when Det Norske Veritas (DNV) was founded as a membership organization in Oslo. Norway's mutual marine insurance clubs banded together to establish a uniform set of rules and procedures, used in assessing the risk of underwriting individual vessels. The group aimed to provide "reliable and uniform classification and taxation of Norwegian ships". /6/

Three years later in Germany, a group of 600 ship owners, shipbuilders and insurers gathered in the great hall of the Hamburg Stock Exchange. It was the founding convention of Germanischer Lloyd (GL), a new non-profit association based in Hamburg. GL was formed out of a desire to achieve transparency. As an independent classification society, GL was created to evaluate the quality of ships and deliver the results to stakeholders. /6/

Both organizations travelled through years developing towards the today's form. In the history the Titanic disaster in 1912 brought safety at sea to the forefront of public concern. International classification societies played an important part in discussions on ship safety. The World Wars took their toll by affecting international relationships but the inter-war period represented improvement and new growth. After the World Wars both organizations took more scientific approach to ship building. New rules based on an analytical and theoretical approach were introduced, and a significant step was taken towards establishing a dedicated research department. In the 20th century also oil findings in the northern seas, introduction of wind energy and ISO-standards have affected the development of new rules and certifications. /6/

1.5 Polarion

Polarion Software is a privately held Zurich based company operating globally with offices in Europe and North America and partners in Europe, South America and Asia–Pacific. The company hit the market with their Software Application Lifecycle Management solution in the spring of 2005. The company offers cross-platform enterprise applications that are used for requirements management, quality assurance and test management as well as application lifecycle management (ALM). The applications it provides are browser-based and can be used via a public or private cloud. The official names of the commercial products of Polarion Software are Polarion ALM, released 2008, Polarion REQUIREMENTS, released 2010 and Polarion QA, released 2012. Together these applications have over two and half million users across a broad range of industries including aerospace, automotive, medical device and systems engineering. /14/

1.5.1 Polarion ALM

The application lifecycle management is defined as the administration and control of an application from inception to its demise. It embraces requirements management, system design, software development and configuration management and implies an integrated set of tools for developing and controlling the project. The Polarion ALM platform is a fully integrated, unified solution for managing requirements and software development projects and process throughout the life cycle. It is a server-based application where everything is unified on a secure platform. /14/

Figure 4. The fields of Polarion ALM. /12/

Polarion ALM is a unified tool designed to deliver transparency to projects through real-time aggregated management information. Information exchange and testing can be synchronized which helps development teams respond faster and with better quality to demands from stakeholders. The ALM solution integrates requirements, tasks, change requests, process management, project planning and time management. Polarion's approach to ALM is built around a single source, repository-based architecture where all artifacts in the development process are stored in a software configuration management system, namely Subversion. /12/

Every requirement and specification item in Polarion is an individual "Work Item", with a lifecycle and status independent of other items in the specification. All items in the specification can be reviewed and approved independently — of each other, and of the containing document — online, using just a web browser, with all changes visible to stakeholders. The major gain with this kind of application is the ease of implementing and maintaining traceability. /12/

Polarion Requirements is the tool for requirements management in Polarion ALM. This solution is designed to help to pass an audit, compliance, or regulatory inspection with traceability that is easily implemented and guaranteed via automatic change control of every requirement from inception through reuse, to derivatives and archival. /13/

Traceability refers to the relationships between artifacts: requirements to functional specifications, and test specifications to both types of specifications, for example. In Polarion individual Work Items are easily linked to each other, after which the traceability is automatic. Traceability also means that the version history can be seen, who has done what and when, and we can compare the changed content to previous versions. /12/

1.5.2 Work item in Polarion

Work Items are the data objects, items that are a major focal point of data and activity managed in Polarion. They can be of different types, which are fully customizable and can be defined in the global configuration, the project configuration, or both. Each work item type can have its own behavior through workflow, which are a state machine controlling possible statuses and transitions between them. Work Items represent various artifacts - requirements, tasks, change requests, for example. /9/

2 DEVELOPMENT PROCESSES

When a product is created or service provided, whether it means developing software, writing a report, or taking a business trip, there is always sequence of steps to follow to accomplish the set of tasks. A process is a set of ordered tasks: a series of steps involving activities, constraints, and resources that produce an intended output of some kind. A process is more than a procedure. Where procedure is a structured way of combining tools and techniques to produce a product, a process is a collection of procedures, organized so that a product can be built to satisfy a set of goals and standards. /11/

A software lifecycle is an abstract representation of a software process. It defines the phases, steps, activities, methods, tools, as well as expected deliverables, of a software development project. It defines a software development strategy. /7/

Various identifiable phases between product's 'birth' and its eventual 'death' are known as lifecycle phases. The typical software lifecycle phases are:

1. Requirements analysis
2. System design
3. Implementation
4. Integration and deployment
5. Operation and maintenance

Requirements analysis is the activities of determining and specifying requirements. Requirements analysis is assisted by a good degree of engineering rigor, and it is therefore sometimes identified as a requirements engineering. Requirements determination proves to be one of the greatest challenges of any software development lifecycle. Users are frequently unclear about what they require from the system. /7/

System design is the modeling that takes into consideration the platform on which the system is to be implemented. System, or software, design is defined as "a description of the structure of the software to be implemented, the data which is part of the system, the interfaces between system components and, sometimes the algorithms used" /18/. /7/

Implementation is mostly programming. It is usually the longest of the development phases. In some lifecycle models, such as in the agile software development, implementation is a dominant development phase. /7/

Verification is the process of confirming that the designed and built product fully addresses documented requirements. Verification consists of performing various inspections, tests, and analyses throughout the product lifecycle to ensure that the design, iterations, and the finished product fully address the requirements. In other words, software verification is ensuring that the product has been built according to the requirements and design specifications. /7/

Integration assembles the application from the set of components previously implemented and tested. Deployment is the handing over of the system to customers for production use. /7/

Operation signifies the lifecycle phase when the software product is used in day-to-day operations and the previous system is phased out. Operation coincides with the start of product maintenance. In software engineering maintenance includes, in addition to fixing problems arising, the product evolution. /7/

There exist a number of useful lifecycle models, which are in general agreement on lifecycle phases but differ on the importance of particular phases and on interactions between them.

2.1 V-model

The development process of software is often conceptualized as a V-Model and which is compliant with and largely based on the requirements of IEC 61508, an international standard of rules applied in industry. The V-model demonstrates how testing activities are related to analysis and design. The V-Shaped lifecycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins. /7/

The process begins with requirements engineering and is then followed by design at increasing levels of detail. Design is completed prior to implementation which includes both hardware manufacturing and software coding. When implementation is completed, testing proceeds from the smallest hardware and software components to field testing of the entire system. Finally, the system is verified against the design and validated against the requirements. The simplified V-Model diagram shown in Figure 5 illustrates the basic concept. /7/

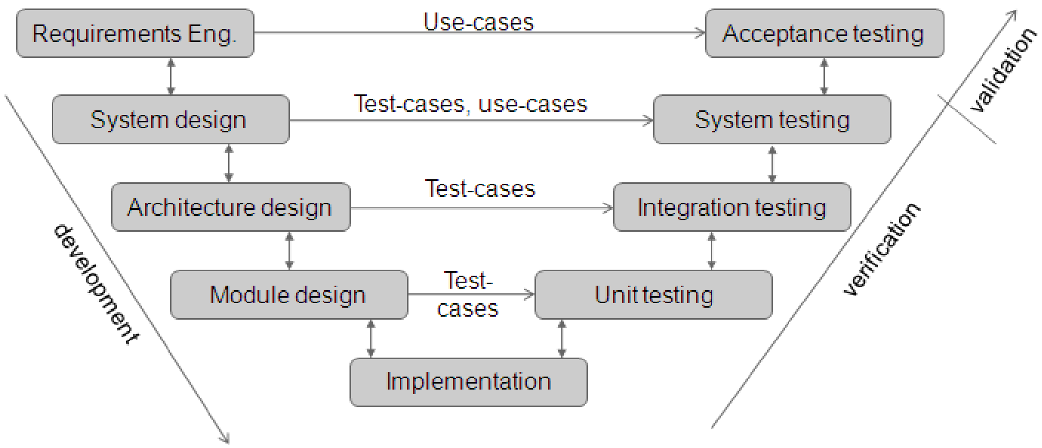


Figure 5. Presentation of the V-model. /16/

As advantages of the V-model can be seen that it is simple and easy to use, designing happens before implementation which saves time and defects are found at early stage. This model works well for small projects where requirements are easily understood. /7/

As a disadvantages of the V-model can be seen its rigidness, lack of flexibility, the lack of early prototypes as the software development is done during the implementation phase and if changes happen midway, the test documents along with requirement documents has to be updated. /7/

2.2 Agile Methods

The agile software development process, proposed in 2001 by Agile Alliance, is a daring new approach to software production. The spirit of the agile development is captured in four recommendations:

1. "Individuals and interactions over processes and tools.
2. Working software over comprehensive documentation.
3. Customer collaboration over contract negotiation.
4. Responding to change over following plan." /10/

The agile development stresses that software production is a creative activity that depends on people and team collaboration far more than processes, tools, documentation, planning, and other formalities. Unlike in other software processes, in agile development, customers work closely with the development team throughout the lifecycle. /7/

The Agile development provides opportunities to assess the direction throughout the development lifecycle. In the agile development 'conventional' integration and deployment is replaced by continuous integration and short cycles, known as sprints or iterations. In an agile paradigm, every aspect of development is continually revisited. When a team stops and re-evaluates the direction of a project every two weeks, there's time to steer it in another direction. /7/

Despite all the 'revolutionary' propositions, agile development sits well among other iterative lifecycles. An agile lifecycle may not be using the terminology of typical lifecycle phases, but it does in effect follow the normal cycle of analysis, design, implementation, and deployment. /7/

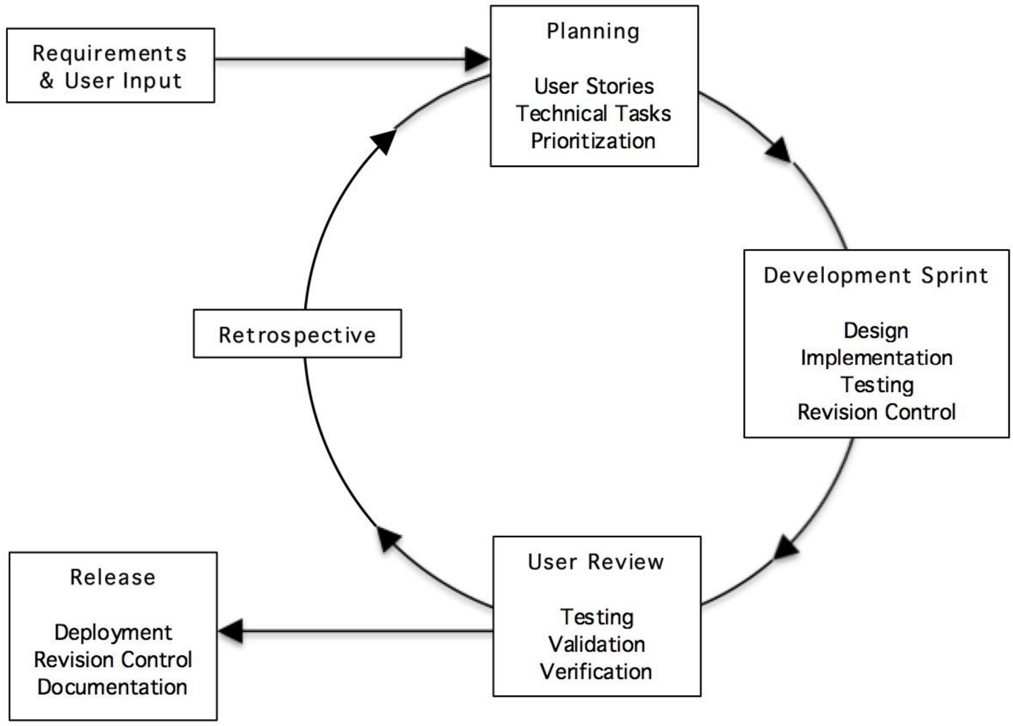


Figure 6. Presentation of agile methods. /15/

Agile methodologies can be inefficient in large organizations and certain types of projects. Many organizations believe that agile methodologies are too extreme and adopt a hybrid approach that mixes elements of agile and plan-driven approaches. /7/

3 REQUIREMENTS ENGINEERING

The domain software requirements engineering is split into requirements development and requirements management. The goal of requirements development is to identify, agree upon, and record a set of functional requirements and product characteristics that will achieve the stated business objectives. The central purpose of requirements management is to manage changes to a set of agreed-upon requirements that have been committed to a specific product release. Requirements management also includes tracking the status of individual requirements and tracing requirements both backward to their origins and forward into design elements, code modules, and tests. Figure 7 shows the structure of requirements engineering. /25/

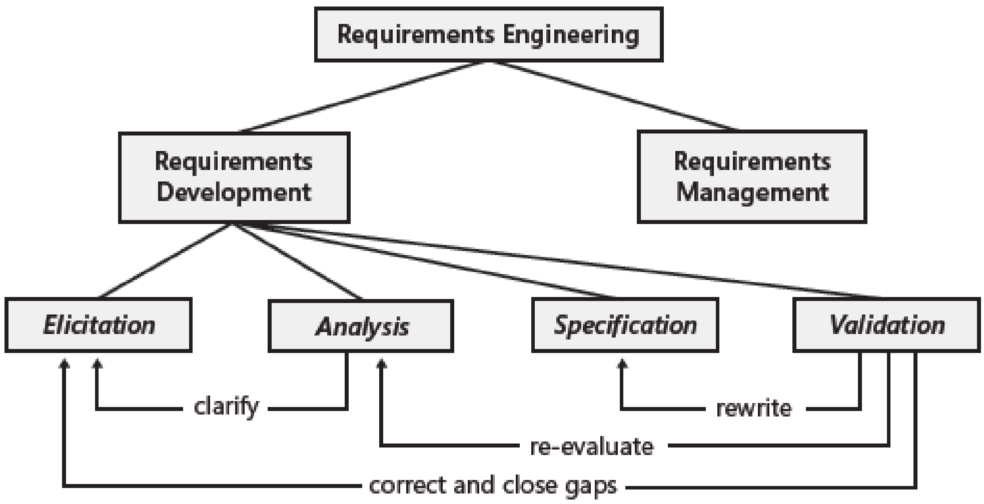


Figure 7. The fields of requirements engineering. /25/

3.1 Requirement

"Requirements are a specification of what should be implemented. They are descriptions of how the system should behave, or of a system property or attribute. They may be a constraint on the development process of the system." /19/

A requirement can be seen as a condition or capability:

- Needed by a stakeholder to solve a problem or achieve an objective.
- That must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documentation. /11/

The requirement is a statement that identifies a necessary attribute, capability, characteristic, or quality of a system for it to have value and utility to a stakeholder. Requirements can serve three purposes. First, they allow developers to explain their understanding of how the customer wants the system to work. Second, they tell designers what functionality and characteristics the resultant system is to have. And third, the requirements tell the test team what to demonstrate to convince the customer that the system being delivered is indeed what was ordered. /11/

3.2 Project and Product Requirements

Project requirements define how the work will be managed. Project requirements focus on who, when, where and how something gets done. Project requirements are generally documented in the project management plan. Product requirements include high level features or capabilities that the business team has committed to delivering to a customer. Product requirements do not specify how the features or capabilities shall be designed. /11/

3.3 Functional and Non-functional Requirements

Requirements can also be divided as functional and non-functional. A functional requirement describes an interaction between the system and its environment. They specify particular results of a system, how the system should behave given certain stimuli. Functional requirements drive the application architecture of a system. /11/

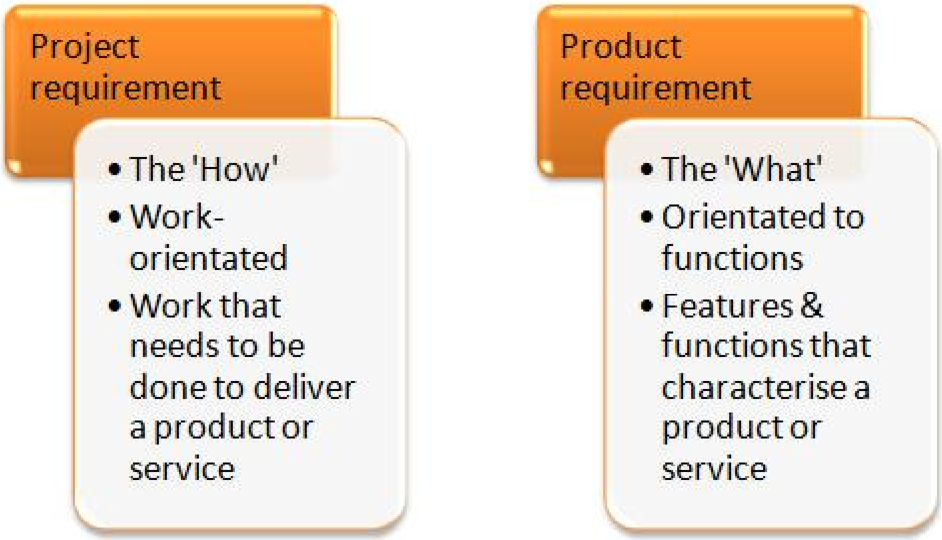


Figure 8. Project and product requirements.

A non-functional requirement or constraint describes a restriction on the system that limits our choices for constructing a solution to the problem. Non-functional requirements specify overall characteristics such as cost and reliability and drive the technical architecture of a system. /11/

3.4 Requirements Management

Requirements management is formally defined as the process for eliciting, documenting, analyzing, prioritizing and agreeing upon requirements, complemented by the control and communication of change to requirements through the project lifecycle. /4/

Requirements management is a part of overall project management. Management is the measurement, control, and monitoring of any given process which yields specific results. Management is utilized as a means of ensuring that results are predictable by applying a stable process with carefully measured steps and stages. Requirements management is the control of the process utilized to create the requirements. /4/

Requirements management is concerned with three main issues:

1. Identifying, classifying, organizing and documenting the requirements.
2. Requirements changes.
3. Requirements traceability. /8/

In requirements management controlling the resulting documents limits the ability to predict the quality and to refine those results as needed; and controlling the process allows the identification of trends, issues and risks, and exposes needed changes to achieve the desired outcomes. /4/

Requirements management helps to ensure that the end product meets the needs and expectations of the stakeholder. Requirements are defined during the planning phase and are managed throughout the entire process from high level requirements, through detailed requirements, design, build and test. Requirements management is a continuous process throughout a project. The ultimate goal of requirements management is, at the end of the day, to ensure that the final product meets the needs of the business. /4/

3.4.1 Identifying, Classifying, Organizing and Documenting

A typical system of a requirements collection consists of hundreds or thousands of requirements statements. To properly manage such large numbers of requirements, they have to be numbered with some identification scheme. The scheme may include a classification of requirements into more manageable groups. There are several techniques of identifying and classifying requirements like unique identifier, sequential number within document hierarchy and sequential number within requirements' category. /8/

Hierarchies of requirements allow defining requirements that are different levels of abstraction. This is consistent with the overall modeling principle of systemically adding details to models when moving to the lower level of abstraction. Requirements can be hierarchically structured, for example parent-child relationships where parent requirement is composed of child requirements. As a result, high-level models can be constructed for parent requirements and lower-level models can be linked to child requirements. /8/

3.4.2 Change Management

A requirement may change, be removed, or a new requirement may be added at any phase of the development lifecycle. A change may be linked to human error, but is frequently caused by internal policy changes or external factors. Whatever the reason, strong management policies are needed to document change requests, to assess a change impact, and to effect the changes. /8/

Change management involves tracking of large amounts of interlinked information over long periods of time. Without tool support, change management is doomed. The ideal decision for change management is a requirements management tool that stores and tracks the changes. This allows the developers to handle versions of models and programs across the development lifecycle. /8/

3.4.3 Requirements Traceability

Requirements traceability is concerned with documenting the life of a requirement. It is an activity of management which enables the project team members to locate the original source and destination of every requirement. It should be possible to trace back to the origin of each requirement and every change made to the requirement should therefore be documented in order to achieve traceability. Even the use of the requirement, after the implemented features have been deployed and used, should be traceable as well as all changes made to the requirements. /4/

Requirements come from different sources, like the business person ordering the product, the marketing manager or from standards. These stakeholders all have different requirements for the product. Using requirements traceability, an implemented feature can be traced back to the stakeholder or group that wanted it during the requirements elicitation. This can, for example, be used during the development process to prioritize the requirement, determining how valuable the requirement is to a specific user. It can also be used after the deployment when user studies show that a feature is not used, to see why it was required in the first place. /4/

Requirements traceability focuses on mapping the relationships between requirements and development artifacts. It is intended to promote and facilitate:

- The ability to control and measure changes during development
- The ability to make calculated steps toward the improvement of the business situation
- Full comprehension and transparency of the solution
- The quality of the solution developed.

Requirements traceability is not only from requirement forward, but it is a bidirectional and must link between all elements, from the business objectives to the implemented solution. /4/

3.4.4 Reasons to do Requirements Management

Numerous studies have examined the effects of errors in requirements on software projects. They consistently find that nearly half of the discovered defects originated as requirement errors. The typical outcome of errors in the requirements is an expectation gap, a difference between what developers build and what customers really need. Clearly, any domain that is the root cause of approximately half of the problems on software projects deserves our attention. /25/

The main reason errors in requirements are so damaging is that they cause extensive rework to correct the errors. It is well-established that the cost of correcting a software error increases dramatically the later it is discovered, as shown in Table 1. An error, omission, or misunderstanding in the requirements forces developers to redo all the work they have already done based on the incorrect requirement. Therefore, any technique that can reduce requirement defects and prevent some of this wasted effort is a high-leverage investment indeed. /25/

Table 1. Relative cost to correct a requirement defect. /25/

Stage Error Was Discovered	Relative Cost to Correct
Requirements development	1X
Design	2–3X
Construction	5–10X
System or acceptance test	8–20X
Operation	68–110X

In addition to avoiding some of the negative consequences described above, better software requirements provide numerous benefits.

- Selecting Projects to Fund. Good preliminary requirements enable companies to make effective business decisions as organizations decide which potential projects to fund. Once a project is funded, better requirements allow more sensible partition of tasks among teams.

- Facilitating Estimation. Well-understood requirements can help the team estimate the effort and resources needed to execute a project.

- Enabling Prioritization. Most projects need to make compromises to ensure that they implement the most critical and most timely functionality. Documented requirements allow the team to prioritize its remaining work.

- Developing Designs. Requirements are the foundation for design. Well understood and well-communicated requirements help developers devise the most appropriate solution to the problem. High-quality requirements also ensure that the development team works on the right problem.

- Testing Effectively. Well-defined and testable requirements allow testers to develop accurate test procedures to verify the functionality.

- Tracking Project Status. Comprehensive, traced set of requirements helps the stakeholders know when the project is done. Defined business requirements also allow the stakeholders to determine whether the project has met its goals.

- Accelerating Development. Putting more effort in developing the requirements can accelerate software development. Defining business requirements, the expected business outcomes the product will provide, aligns the stakeholders with shared vision, goals, and expectations. /25/

3.5 Requirements Development

Requirements development process, sometimes referred as requirements evolution, is the shaping of requirements through their life cycle. Requirements development is not passive process. It is an active process for capturing, understanding, derivation, exploration, analyzing, and testing of requirements. Requirements development is a process of elicitation, analysis, specification and validation of requirements as shown in Figure 7. /4/

3.5.1 Elicitation

Requirements elicitation is the practice of identifying the requirements of a system from users, customers and other stakeholders. The practice is also sometimes referred to as "requirement gathering". This stage includes lots of researching, interviewing, analyzing and validating to be done in order to generate a complete set of functional and non-functional requirements. Some requirements can be exposed by reading through documentation and diagrams and some by interviewing the stakeholders. Other examples of requirements elicitation practices are user observation, questionnaires, workshops and prototyping. /4/

The information gathered, both from individual sources and from collections of documentation, amounts to details about the goals of the business, problems to be solved, and impacts of those problems on the business and its ability to conduct its work and transactions in an effective and affordable way. These sources provide descriptions about the results that are needed, as well as the assumptions, constraints, risks, gaps, and opportunities that exist. Understanding what can be extracted from each source is a first step in a requirements development process. /4/

These requirements are needed for design and development to occur. Within the context of a project, requirements provide the overall blueprint for the end product. They form the foundation for architects and developers to design and build the new system or application. In additon requirements provide a benchmark by which the end product can be tested for quality and its ability to meet the original objectives of the project. /4/

3.5.2 Analysis

Requirements analysis is about categorizing and clarifying, defining into more detail it encompasses those tasks that go into determining the needs or conditions for a product or project to meet. At this stage the requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. At analysis stage the possibly conflicting requirements of the various stakeholders should be taken into account as well as further analyzing, documenting, validating and managing software or system requirements. /4/

At this stage it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems. Several techniques can be used to elicit the requirements from the customer. These may include the development of scenarios (represented as user stories in agile methods), the identification of use cases, holding interviews, or requirements review sessions and creating requirements lists. /4/

This stage breaks down functional and non-functional requirements to a basic design view to provide a clear system development process framework. A thorough requirement analysis process involves various entities, including business, stakeholders and technology requirements. /4/

Most likely requirements analysis occurs throughout the development lifecycle of a project. Requirements analysis and definition can refine existing requirements to determine their impact on current business processes, systems, and modifications or can be applied in future design efforts to meet evolving information technology, systems integration, and business needs and challenges. /4/

3.5.3 Specification

The specification stage is where the refined requirements are formally drafted within the document for validation process. Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. /4/

A software requirements specification is a description of a software system to be developed, laying out functional and non-functional requirements. The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements, a clear and thorough understanding of the products to be developed is needed. /4/

The specification stage is about writing the requirements and revising the future state, business rules, and use cases. The key deliverable from the specification stage is the draft of the requirements specification document itself. This document must contain enough detail to allow the engineers to create complete designs, the developers to generate complete code, and for the testers to be able to design testing and validate test results against it. This means that the content must be consistent, accurate, and logically complete. It also means that the requirements document describes what the system must do when all is well as well as what the system must do when specific criteria are only partially met or not met at all. /4/

Testable requirements can be measured, are definitive, and have clear parameters for the functionality to be performed. Untestable requirements tend to be ambiguous in nature and leave unanswered questions for other team members to fill in with assumptions. The problem is that the developers and testers may all be making different assumptions about how to fill in those ambiguities. The difference in these assumptions will directly result in defects within the solution. /4/

3.5.4 Validation

Validation stage delivers final and complete requirements to architects, developers and testers for the next steps in the project life cycle. It is the most critical stage in the requirements development process, it is the culmination of all of the requirements into coherent form, the presentation to the audiences, the establishment of the future baseline, and the negotiation point for the final requirements. The validation not only creates complete deliverables but also brings all team members to the point of mutual understanding about the expectations set out in the requirements for what is to be built. /4/

Validation is the process of confirming the completeness and correctness of requirements. Validation also ensures that the requirements: 1) achieve stated business objectives, 2) meet the needs of stakeholders and 3) are clear and understood by the developers. Validation is essential to identification of missing requirements and to ensure that the requirements meet certain quality characteristics. Validation addresses each individual requirement to ensure that it is:

- Correct – accurately states a customer or external need.
- Clear - has only one possible meaning.
- Feasible – can be implemented within known constraints.
- Modifiable – can be easily changed, with history, when necessary.
- Necessary – documents something customers really need.
- Prioritized – ranked as to importance of inclusion in product.
- Traceable – can be linked to system requirements, and to designs, code, and tests.
- Verifiable – correct implementation can be determined by testing, inspection, analysis, or demonstration. /4/

3.6 Requirements Management Tools

Requirements management tools provide storage and linkages between requirements, rules, and processes and typically connect with testing software to increase the ability and efficiency of traceability across the life cycle of the project. In the absence of a formal application, any requirements management tool could be considered an established system for collecting requirements and establishing traceability across all related requirements components. Regardless of the tool being utilized, it will require customization for each project during the planning and preparation stage to ensure that appropriate metrics can be extracted once requirements activities have begun and for accurate measurement and monitoring of performance and progress. /4/; /24/

Ideally, the requirements management tool provides change control, version control, traceability, and benchmarking. By planning the management tools and templates for the entire requirements phase more consistent results and predictable quality can be achieved as well as ongoing activity reports created and general productivity increased. /4/

Requirements management tools solve the problem of managing requirements for multiple releases. The tools overcome many of limitations of document-based storage. Commercial requirements management tools allow requirements to be stored in a single master location that is always current and is accessible via the internet to authorized stakeholders. /4/

A requirements management tool can be vital, especially in large projects, to translate and transform business needs into products. Requirement management tools are not requirements development tools. A Requirement management tool cannot help on defining business goals or develop the “right” requirements. /4/

A good requirements management tool can help to:

- “Manage history of change, record argument behind changes and define requirements baselines

- View and define requirements attributes and provide permissions to team members to change attributes
- Define and track links between requirements to analyze the impacts of future changes
- Provide team members with access to collaborate on certain projects
- Enables the re-use of requirements in as many projects as necessary." /2/

One of the strongest arguments for a requirements management tool is the functionality it provides for the traceability of requirements, which is often necessary to be compliant with industry standards set by industry regulators. The increased complexity that full transparency demands comes with a heavy price and that is the burden of administration. In the case of requirements management, administration comes in the form of maintenance of the requirements documentation. When setting up requirements types, it needs to be considered what is needed. Requirements types, such as those necessary to comply with industry standards are not optional, but many others are. /24/

4 PRESENT STATE AND PROBLEM STATEMENT

4.1 Problem Statement

At the moment, in the A&C department there are not only more and more requirements coming in, but that these requirements come from different stakeholders and are of varying types. All these requirements need to be introduced to the process and managed somehow. As the requirements can vary a lot by the type as well as by the source, they need to have different identifications and contents.

Requirements coming from different stakeholders end up in one product and one product always has requirements coming from several different stakeholders. This brings the department in a situation where tracing these requirements is really difficult without the help of requirements management tool. When applications and hardware modules are composed into a package for an engine to be delivered to a customer, it would be good to be able to trace the classification requirements of that particular engine and to verify them against the rules to get the certification amended by the classification society.

A requirement management tool has been introduced to the department in 2012. The progress of using this tool has been quite slow and today the department is in a situation where some teams are using the tool but all in different ways. A plan for a common system level requirements management process is created as well as templates for most of the needed work items, but these are not completely in use. The plan of the department for the future is one day to be able to handle the whole development process in Polarion.

As the structure for processing normal system requirements already exists, the task of this thesis is to concentrate on classification requirements and how to convert them to system requirements. System requirements already have a stated Polarion structure which allows them to be managed throughout the development process.

The objective of this thesis is to create a model to manage the rules coming from the classification societies, turn them into requirements, and manage them in Polarion. There is a need to separate these from normal system requirements as the implementation process of these varies from normal system requirements as with these there is nothing to be negotiated. This will be accomplished firstly by creating the needed structures for processing the classification requirements processing in Polarion, in other words by creating the concept. Secondly by introducing defined part of DNV regulations into the structure, in other words making a proof-of-concept. The last part of the thesis is to create needed guidelines and documentation for the appropriate use of the structure.

This can be split into tasks of:

- identification and implementation of the high-level containers and work item types for handling the requirements derived from the rules of the classification societies
- creation of a way to transfer classification requirements into implementation to the system level

To accomplish the statements above the scope of this thesis is to:

- collect lots of knowledge and compose a good comprehension about requirements management
- learn to use and develop Polarion
- find the needs for managing classification rules
- create structure for classification requirement management
- learn to write good quality requirements
- write description and guidelines for the new structure

Within this thesis a concept for classification requirements management in Polarion should be produced. A proof of this concept is proposed by implementing a defined set of regulations into Polarion using the structure and work item created. The proof-of-concept should provide more information about the structure and assist the further development of the tool.

4.2 Present State of Requirements Management in Polarion at A&C

The understanding of processes and work items in A&C and in Polarion is vital before starting the creation and implementation of structure. The structure needs to be created before the requirements development process can begin. In general, preparing for requirements management can be seen as setting up the tool and templates that have been identified as necessary for the project.

The situation with the high level container structure when this thesis was started can be seen in Figure 9. It was diagnosed that requirements originating from classification societies or from functional safety should be identified separately from the normal system requirements but it was not decided how.

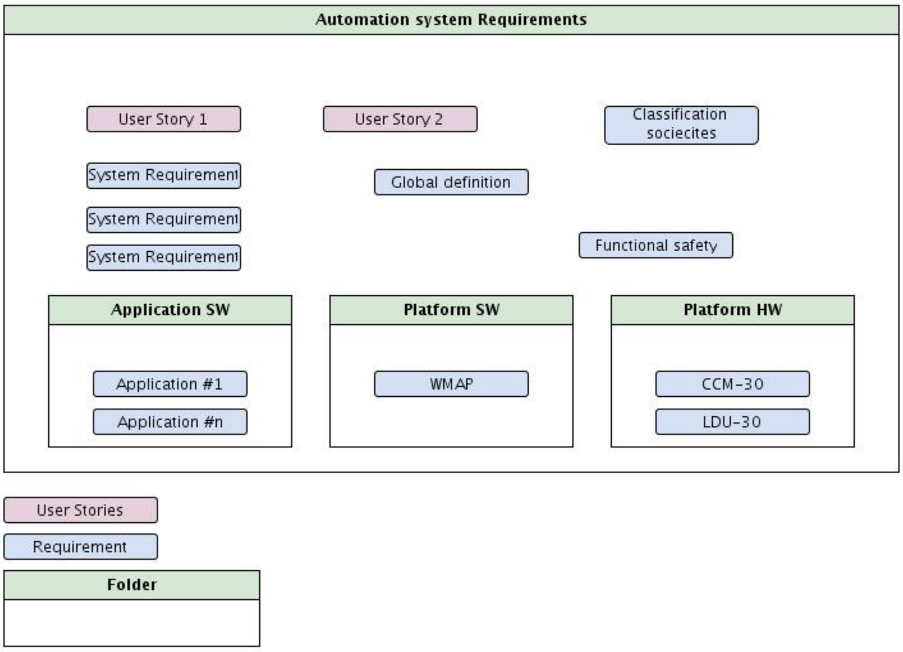


Figure 9. Overview of requirement container structure in Polarion at A&C.

The work items flow stated and agreed in Polarion at the moment is shown in Figure 10. The requirements are introduced to the system at the point which is marked with red in picture. All the work processes start with the Request For Change (RFC) coming to the A&C. This is the trigger for the process of developing software or hardware components in the department. Work requests (WR) are created based on the RFC and given for teams for analysis. During the analysis User Stories (US) and system Requirements are drafted to make a good view of work load needed to fulfill the RFC. This is the stage where the knowledge of standards and regulations should be available. If the decision is made to fulfill the RFC, new team specific WRs' are created as well as USs' and requirements on defining levels.

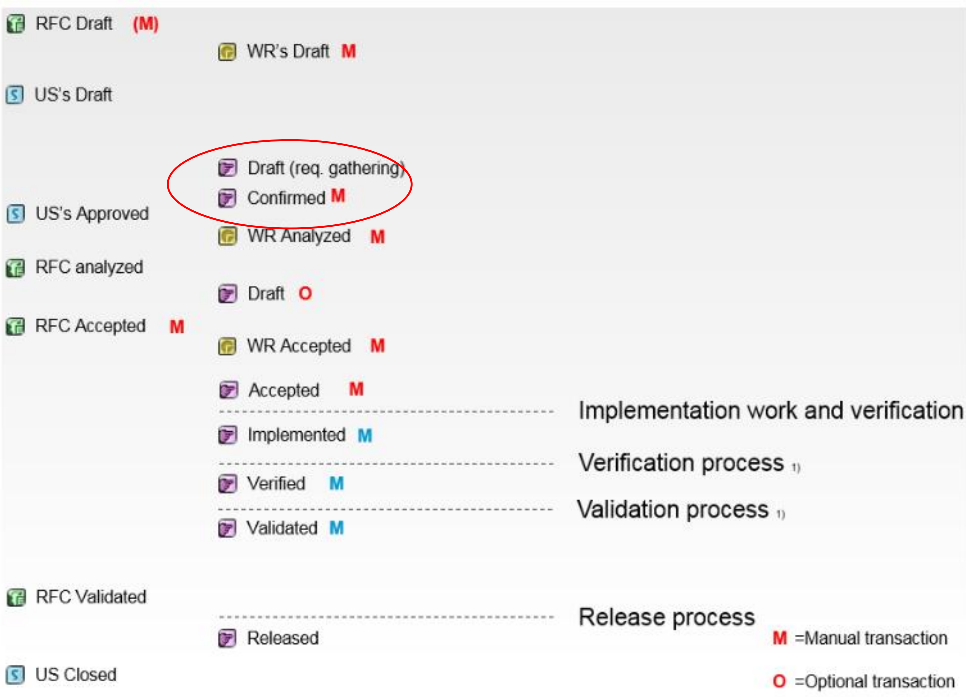


Figure 10. A&C's stated work items flow in Polarion.

5 CREATING THE CONCEPT FOR MANAGING CLASSIFICATION RULES

5.1 Converting Rules into Requirements

Classification rules are published as large rule books or nowadays usually as e-books which can be found on the web sites of each classification society . The rule books tend to be quite massive as they need to contain all the rules considering ship building. When talking about one certain part of ship building, in this case about building the engines of the ship, the rules considering this part can be scattered around the rule books and can be sometimes difficult to find. Requirements can be quite easily gathered from classification rule books but will need analyzing and specification.

Only relevant information will be stored in Polarion, there is no need to import the entire rule books to Polarion as this would not bring any added value to the process. The requirements identified from standards will have their own requirement type and container. These standard requirements will be then duplicated to the system requirements to be implemented in the normal development process of A&C. The idea of how the classification requirements are introduced to the development process is shown in Figure 11.

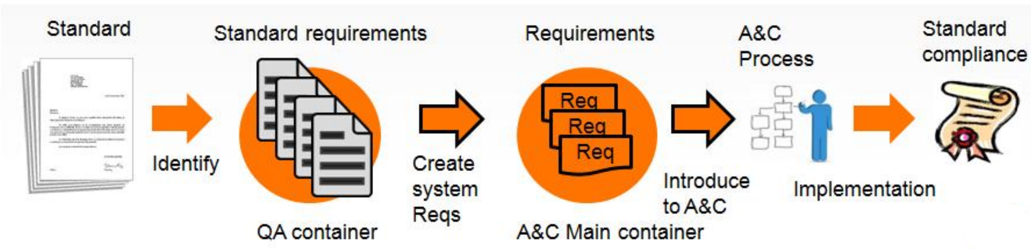


Figure 11. Introduction of standard requirements to the development process.

The C&P teams is responsible for finding the rules related to A&Cs' products from the rule books, converting these rules into good quality requirements and introducing them to the development process. Creating a specific type and container for the classification requirements allows the classification requirements to be maintained by the team separately from the normal system requirements and them to be traced comprehensively for the certification.

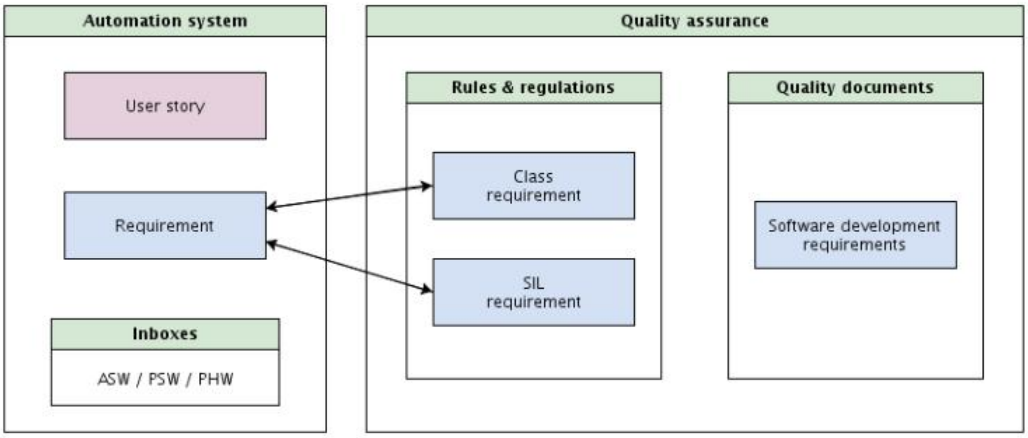
5.2 High Level Structure for Managing Classification Requirements

The classification requirements differ from the normal system requirements not only by their origin, but also by some other features; with classification requirements there is nothing to be negotiated, they need to be fulfilled in specified validity time to get the classification certificate but in the same time cost effects, impacts and implementation need to be documented. Due to this, it was seen that requirements originating from the classification society rules and functional safety standard should have an own container for storing and handling.

The container for classification and functional safety requirements was named Quality assurance, which refers to administrative and procedural activities implemented in a quality system so that requirements and goals for a product, service or activity will be fulfilled. The ISO 9000-standard refines quality assurance as "*part of quality management focused on providing confidence that quality requirements will be fulfilled*". This QA container will hold, at least at the moment, a container for Rules and regulations and other container for Quality documents which is going to be a place for documentation of the processes' quality aspects. /17/

The classification requirement type has a different workflow and template than the normal system requirement that is used for work coordination and implementation. These requirements will also have their own requirement identifier for them to be easily separated from the normal system requirements and to create a basis for traceability. These particular type requirements are then later duplicated to the system requirements for implementation.

Figure 12 shows the high level structure of quality assurance container and the relations to the A&C main container. The arrows describe the duplication of the QA requirements into the system requirements in the automation system. To create this kind of high level structure allows not only to manage work items in separate places but also to configure default settings for each folder, or a project as called in the Polarion language.



Container/Work item	Description
Automation system	Main container for whole A&C (AC Main)
Requirement	System requirement
Quality assurance	Container for quality and certification related work items
Rules & regulations	Container for rules and regulations related work items
Class requirement	Requirement conducted from classification standards
SIL requirement	Requirement conducted from functional safety standards

Figure 12. High level structure of classification requirement container and relations.

5.3 Workflow

"Workflow may be considered a view or representation of real work. The flow being described may refer to a document, service or product that is being transferred from one step to another. Workflows may be viewed as one fundamental building block to be combined with other parts of an organisation's structure such as information technology, teams, projects and hierarchies". /26/

Workflow is a way to manage changes and add traceability of requirements. It allows work items to have different statuses which simulate different phases a work item goes through during its life. Polarion supports a great variety of searches and thus work items can be found by the status they have. This will, for example allow work items that are marked with status *accepted* to be found. Polarion and other requirements management tools demand workflow for all work items. In Polarion there is a default workflow for all work items, but it is recommended to create a new one for new work items to suit the needs of particular work items.

The workflow of classification requirements maintained in their own folder is different from the normal requirements. With the classification requirements, there is nothing to be negotiated. They do not need, in the rules and regulations container, to go through the same stages of workflow as the system requirements.

The life of a classification rule goes through the phases: the classification society gives a pre-warning of becoming new rules or old ones about to change, usually half a year before. After the stated time, the rule becomes in force and should be fulfilled. In a same way rules can become obsolete with half a year warning time. With this known lifetime of classification requirement, the workflow for it was created as can be seen in Figure 13.

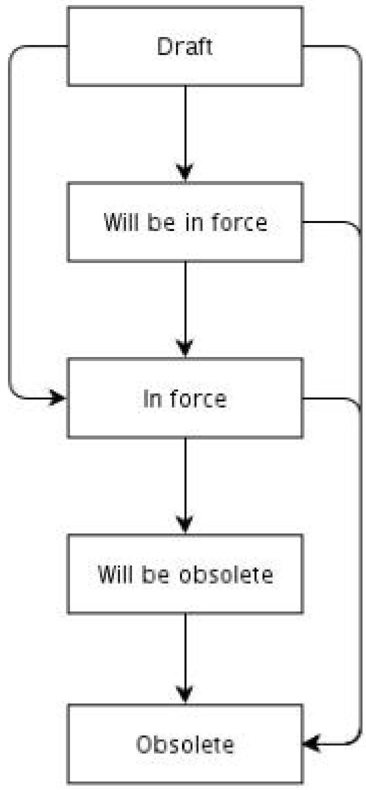


Figure 13. Workflow of classification requirement.

When the requirement is drafted, it has status *draft*. When it contains all the needed information it can be given status *will be in force* or *in force*, depending on the point of life it has in the rule book. After being *in force* and when classification society gives warning of that particular rule to change or to end its validity, the requirement is given status *will be obsolete* which is then turned into *obsolete* when the rule is no more valid. Also, in case the requirement was somehow incorrect, it can be put straight in the *obsolete* status at any stage.

Setting up a workflow is a large part of change management of requirements. Stating workflow as above can be seen as knowledge that requirements do change during the process. It allows requirements, for example, to be set to status *Obsolete* when they are not anymore valid and to replace them with new ones. Stating the steps of workflow as above allows requirements to be traced through the different lifecycle steps. As Polarion automatically saves all the changes done in work items the requirements can be traced by the workflow steps and the dates when changes were done.

In requirements management done without requirements management tools, it is important to freeze *baselines*, which means setting up the certain point where requirements are thought to be gathered well enough and changes are prohibited, or at least need special treating. With Polarion baselining is a built-in feature; all changes to work items are saved and thus baselines can be afterwards searched at any specified time point.

5.4 Classification Society Requirement Template

The work item template is the form where all the information about the work item is written and stored. The template offers the fields to be filled in order to transmit all the relevant information forward. The fields in the template also allow requirements to be searched via the information they contain. For example, a search can be done to find all the requirements that have status *in force* or are required by DNV, or fulfill both search criteria. The template is created by the XML-editor with enumerations attached to it. In Figure 14 there is an example of the XML-language, defining the layout of the template.

To create the template for the classification requirement it needs to be clear what information is needed. In Polarion, there are some default work item types and templates but for the classification requirements, there was need to create a new form.

In this template the interpretations of different standards are grouped and all the classification societies having the same content in their standards are mentioned in the template. Most classification societies base their rules on the same origin, the IACS standards, which causes them to have many partly or fully similar rules. The reason to do it this way is to reduce the amount of requirements. This will ease the work load of the system designers as they can just implement the requirements that are in the system, there is no need to check whether the requirements comply with some other. On the other hand, this may cause the work of C&P team to be more complex. The requirements gathering and specifying will demand more complex thinking on when and how the different rules are combined into a single requirement. Choosing to implement the classification requirements into Polarion this way will most likely cause more changes in the requirements, but this was seen as a better way due the simplicity of Polarion structure and the reduced amount of requirements.

```
        <section>
            <field id="categories"/>
        </section>            <section>
            <field id="ENGINE_TYPE"/>
        </section>
    </vertical>
    <vertical>
        <section>
            <field id="status"/>
            <field id="resolution"/>
        </section>
        <section>
            <field id="VALIDITY_START_DATE"/>
            <field id="VALIDITY_END_DATE"/>
        </section>
    </vertical>
        <section>
            <field id="type"/>
            <field id="author"/>
            <field id="project"/>
        </section>
    <vertical>
        <section>
         <field id="CLASS"/>
        </section>
```

Figure 14. Example of defining template layout with XML language.

At this stage of creating the template, it is better rather to have less fields than needed than too many. Working with the requirements management tool it is always easier to add fields or any other information later than to delete information that is already in the system. In the tools like this, all information is related to other information and deleting something can have surprising effects.

The template needs to contain the information about requirement type, whether it is a system or a process requirement, for it to be duplicated to the correct project. It needs to have the information about which engine type it applies to for developers to know where to address the requirement. The status of the workflow has to be present as well as the date for validity to start and to end. Also to know all the classification societies requiring this are needed to be known. As the template mentions all the societies stating the same requirement, the start of validity will indicate the date when the first society starts the validity. The validity end date is the date when the last one of the classification societies retires the rule. This allows defining the time period when the requirement shall be acceptably implemented.

The next field of the classification requirement template, *description*, contains the requirement itself, written as good a quality requirement to allow it to be implemented and tested. The next fields contain information which is needed for the system level requirement. These are for the basic information transmitted to system level; who is requiring this and why. In Figure 15 there is a classification requirement template filled in. In the template fields for comments and linked work items can also be seen that Polarion automatically creates into templates.

Polarion always gives an ID for each work item created. The ID consists of a project prefix, in this case CLASS, and a running number. The IDs are distinctive for each work item and can never be re-used. In order to users to separate work items from each other or to find work items related to particular subject, the requirements need to be given titles and to be grouped. The tree view of the requirements is structured by creating higher level *parent* requirements which have *child* requirements refining them. This will gather several requirements related to the same issue into one category which is easier to perceive and trace. Figure 16 shows the list of classification requirements with IDs', titles and the structure shown in the Polarion tree-view.

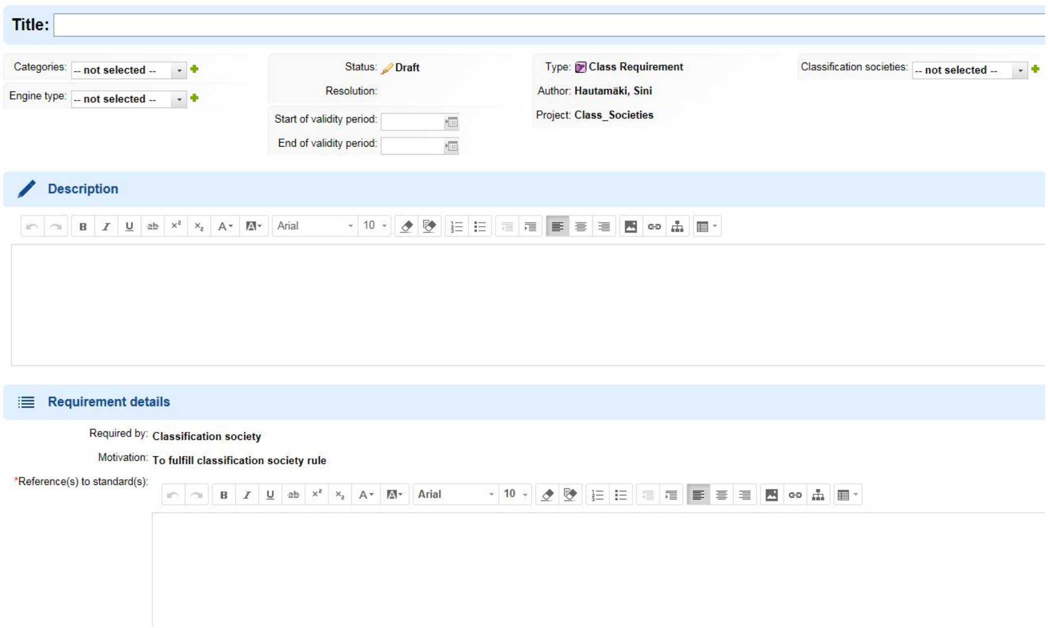


Figure 15. Classification requirement template.

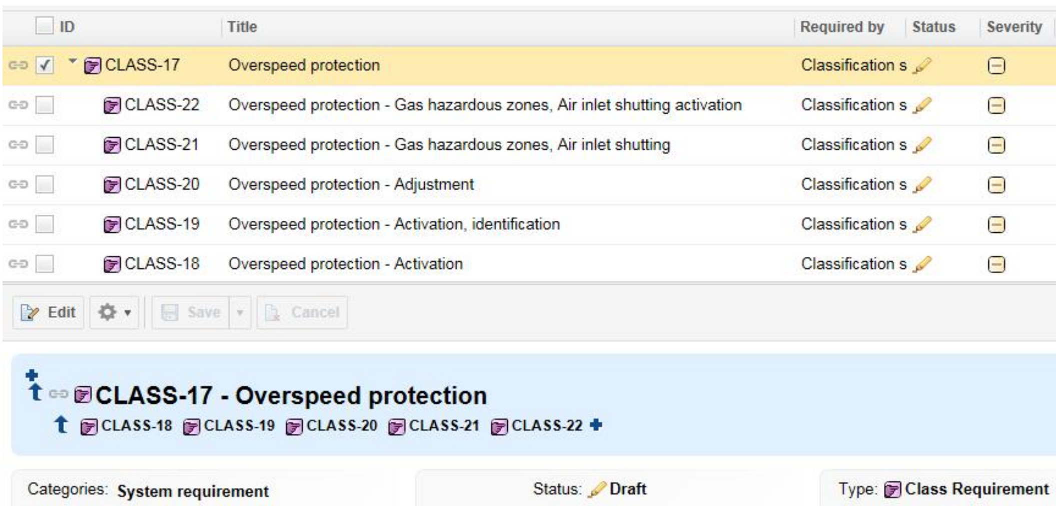
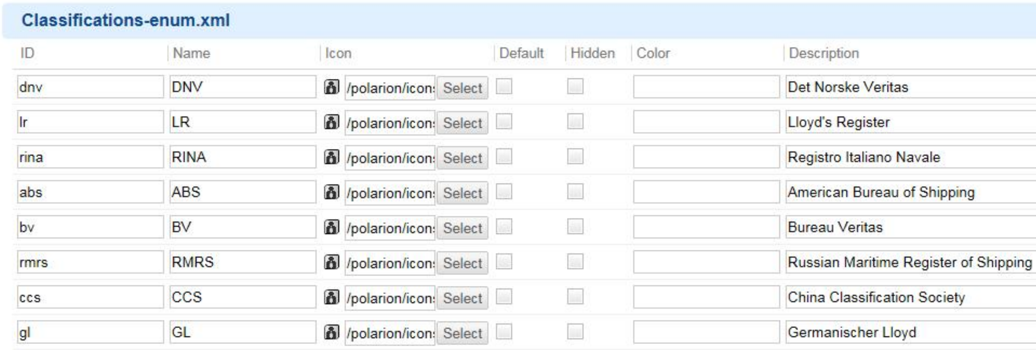


Figure 16. List of requirements in Polarion.

Several enumerations were used when creating the classification requirement template. Enumerations are basically lists of options that can be selected and applied as a value to some field of a work item. Typically, enumeration values appear as choices in a drop-down list. For this template, several enumerations were used; for example, the lists for engine types, classification societies, or the status. In Figure 17 the enumeration for classification societies can be seen.



Classifications-enum.xml

ID	Name	Icon	Default	Hidden	Color	Description
dnv	DNV	/polarion/icon: Select				Det Norske Veritas
lr	LR	/polarion/icon: Select				Lloyd's Register
rina	RINA	/polarion/icon: Select				Registro Italiano Navale
abs	ABS	/polarion/icon: Select				American Bureau of Shipping
bv	BV	/polarion/icon: Select				Bureau Veritas
rmrs	RMRS	/polarion/icon: Select				Russian Maritime Register of Shipping
ccs	CCS	/polarion/icon: Select				China Classification Society
gl	GL	/polarion/icon: Select				Germanischer Lloyd

Figure 17. Enumeration for classification societies.

5.5 Duplicating Requirements

The classification requirements need to be duplicated to the system level for implementation. System requirements have their own template with its own distinctive features, workflow, and fields. The duplication includes choosing the values transferred for system requirements and choosing the link role between these requirements.

There are certain values that are essential in the system requirements created from the classification requirements. The system requirement conducted from the classification requirement needs to have clear information about the requiring party to highlight the fact that these are requirements that must be fulfilled and that there are no possibilities to negotiate about the aspects included. For this the fields of *required by* and *motivation* are set as default non-editable values already in the classification requirement template and are copied straight to the system requirement template. The most important part to be copied to the system level is the description, the requirement text, which is already copied as a default by Polarion.

Duplicating work items is a built-in function in Polarion. Thus, it is widely used form of requirements traceability. The duplication of the classification requirements to the system requirements is an easy process that goes through with few clicks. Polarion automatically copies the title, description, required by, and motivation fields. Other fields can be chosen to be copied

Link roles are important parts of requirements traceability. They are paths between requirements which allow implemented features to be traced back to the stakeholder that requested them. The requirements as well as other work items also have the links *refines* to the work items of the same kind, which means that requirements can form hierarchies, where the lower level requirement refines the higher level requirement.

The link role between QA and system requirement is *relates to*. This is the default by Polarion. No specific needs for this link were found other than showing that these are related to reach traceability. It was considered to name the link role differently, to create a specific link role for requirements based on rules and regulations, but as there are already other values highlighting the source of these requirements, it was seen unnecessary. In Figure 18 the link roles between requirements are presented.

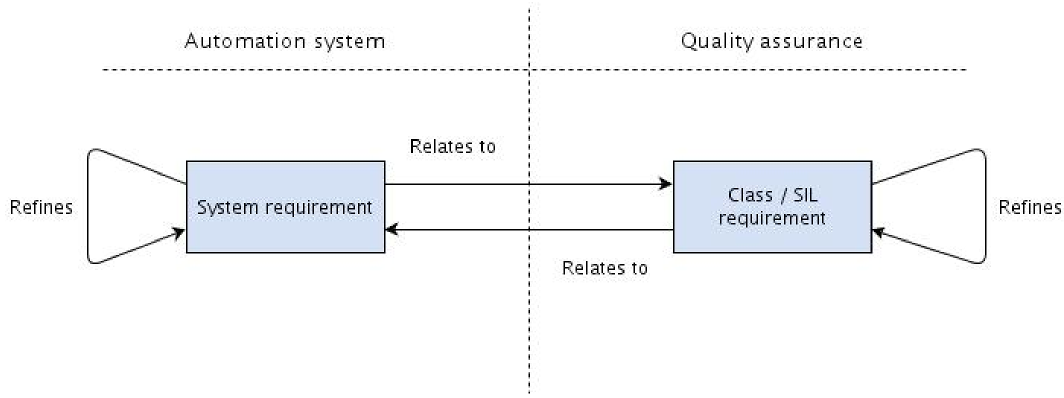


Figure 18. Links of quality assurance requirements.

5.6 Guidelines

The guidelines were written to instruct users to fill in the template created and guide them to duplicate the classification requirements to the system level. The guidelines to assist the use of the created concept are meant for people who are already familiar with Polarion and requirements development, who know how to write good quality requirements, but need guidelines how to use the particular classification society requirement template.

The guidelines were written as a Polarion document. In this thesis it was agreed to create only short how-to descriptions due to the restricted time. The guidelines were created mainly as tables and figures and additional short texts as these were found to be more informative and easier to comprehend than long describing texts and allow organizing the document more effectively. Short sections also give more opportunities to insert informative headings in the material.

6 PROOF OF CONCEPT - CREATING CLASSIFICATION REQUIREMENTS INTO POLARION

The following tasks were done by using the concept created earlier. In Figure 20 the folder structure view in Polarion can be seen where all the work is started. This shows the folder structure specified earlier.

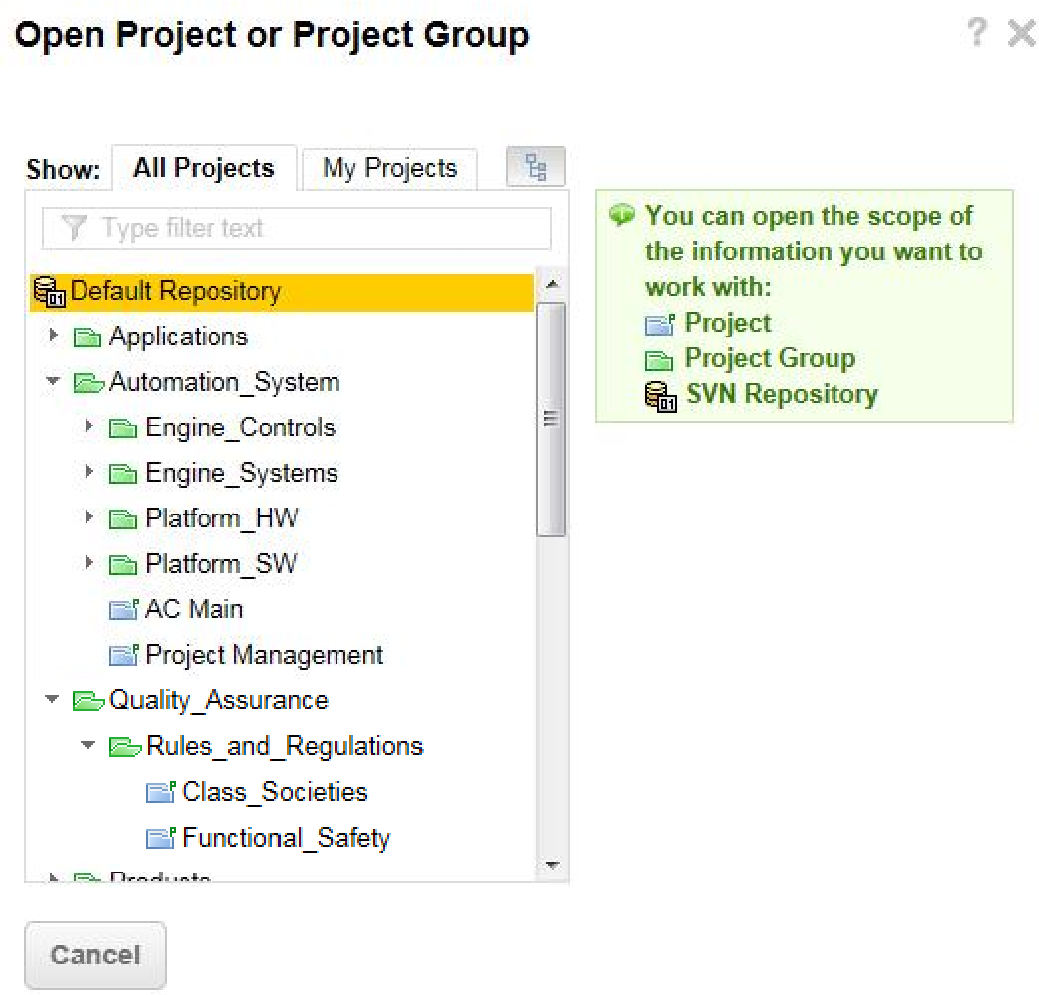


Figure 19. Folder structure in Polarion.

6.1 Requirements Development

The following text is a straight reference from DNV's rules considering overspeed protection. Requirements development is taken into use to transform the rules into good quality requirements for the engine control system. This means going through the phases of elicitation, analysis, specification, and validation as follows.

"E 300 Overspeed protection

301 A separate overspeed protective device is required. The overspeed protective device may be substituted by an extra speed governor that is completely independent of the first governor and acting without delay.

302 Activation of the overspeed protection device shall cause either engine shutdown or limitation of rpm. This applies to both systems if the overspeed protection device is substituted by an extra speed governor. Activation of the overspeed protection system shall be identified in the control room.

303 The overspeed protective device shall be adjusted to ensure that the engine speed cannot exceed the maximum permissible speed as determined by the design, but not beyond 120% rated speed except for diesel engines driving generators where 115% of rated speed applies.(IACS UR M3.1.2 and M3.2.5)

304 For engines operating in areas defined as gas hazardous zones or spaces (see applicable class notation), an additional device that automatically shuts the air inlet in case of overspeed. The shutting device shall activate at the same speed level as does the overspeed protective device required in 301. For engines with turbochargers that can suffer overspeed due to a sudden shut of air intake, the shutting device should be between the turbocharger and the engine." /5/

Elicitation is the practice of identifying the requirements of a system from stakeholders. This thesis considers classification requirements which can be quite easily found from the rule books published by the classification societies. These requirements can be exposed by reading through the documentation. The identified important parts of the rules are highlighted with yellow.

The requirements analysis phase is about categorizing and clarifying, defining the requirements into more detail. After this phase the requirements should be defined to a level of detail sufficient for the system design. The requirement identified from the text at the earlier stage should be now modified into more detail. For the purposes of requirements management at A&C this phase also meant choosing the right requirements for the implementation in case there was a possibility to choose and leaving out the ones that do not comprehend the development in the department. In this case there was also a need to clarify the rule about the rated speed limit with an additional comment about direct propulsion engines.

The specification stage is where the refined requirements are formally drafted within the document for validation process. In this phase the text from the rule book was transformed into requirements that can be written into Polarion as class requirement work items. Five requirements were found from the overspeed protection part of the rules.

1. Separate overspeed protective device is required
   - 1.1. Activation of the overspeed protection device shall cause engine shutdown
   - 1.2. Activation of the overspeed protection system shall be identified in the control room
   - 1.3. The overspeed protective device shall be adjusted to ensure that the engine speed cannot exceed 115% of rated speed (In case of direct propulsion engine the engine speed shall be adjusted to ensure that the engine speed cannot exceed 120% of rated speed.
   - 1.4. Engines operating in areas defined as gas hazardous zones or spaces an additional device that automatically shuts the air inlet in case of overspeed is required between the turbocharger and the engine
   - 1.5. The air inlet shutting device shall activate at the same speed level as the overspeed protective device required in 1.3. (115% or 120% of rated speed).

Validation is the process of confirming the completeness and correctness of requirements. In this case the validation of requirements meant that person responsible for classification aspects validated the requirements. If there was any doubt that the rules are not understood correctly, the normal process would be to communicate with the classification society in question.

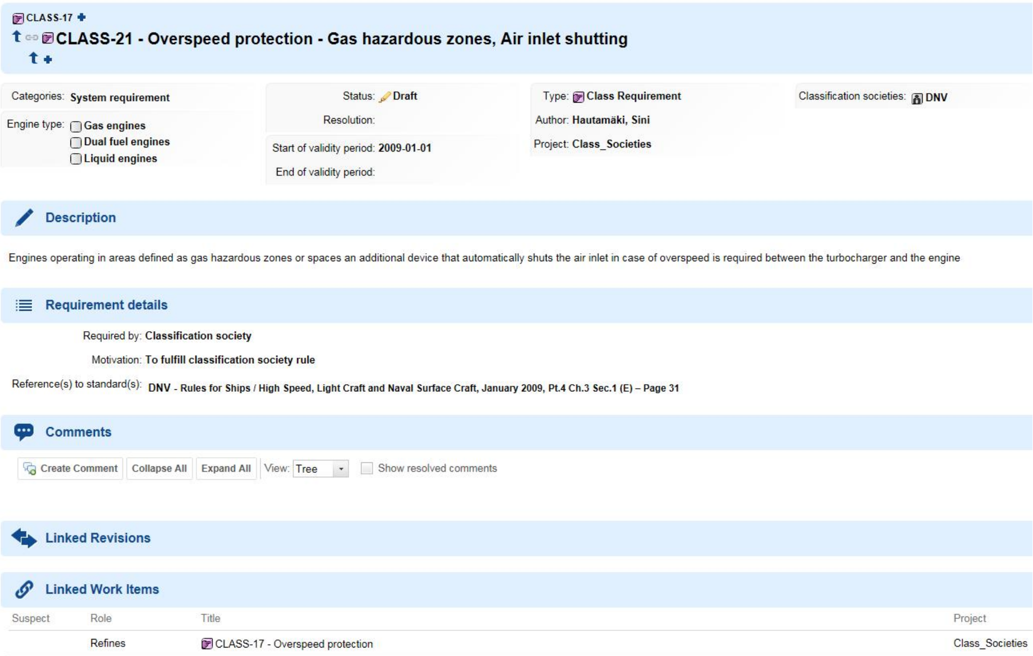


Figure 20. Example of classification requirement with information added into it.

6.2 Requirements Management

To properly manage such large numbers of requirements, they have to be numbered with some identification scheme. The scheme may include a classification of requirements into more manageable groups. This part of rules formed one group of overspeed protection requirements. The group includes one higher level requirement which has five lower level requirements refining it. This allows these to be handled as a group and thus helps to organize, identify and trace these requirements.

In this case when working with requirements management tool, every requirement created is assigned with an identifier which consists of a Polarion project specific prefix, in this case Class, and running number. This is a way to all work items in the tool to have unique IDs' for the system and users to separate work items from each other. The requirements were also given names to help the usage. As there is going to be hundreds or thousands of requirements in the system, it was decided that the name should contain first the field of design and the further specification of certain requirement. This helps the users to find requirements considering certain parts of design via the search tool or by looking the lists. The tree view of overspeed protection requirements list is shown in Figure 21.

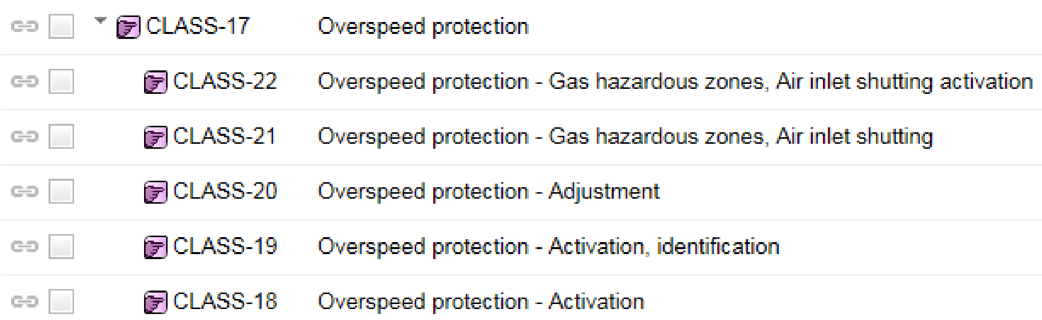


Figure 21. The list of overspeed protection requirements in Polarion.

6.3 Tracing Requirements

All the information inserted into template, including all the fields with their information as well as links can be searched. In Figure 22 an example of tracing work items with searches can be seen. The arrows in the matrix describe the links between work items. Searches can be generated into different views and this kind of matrix view is beneficial when handling large amount of work items and it is important to see the links. Multiple searches can be placed for the rows or columns to find the needed work items.

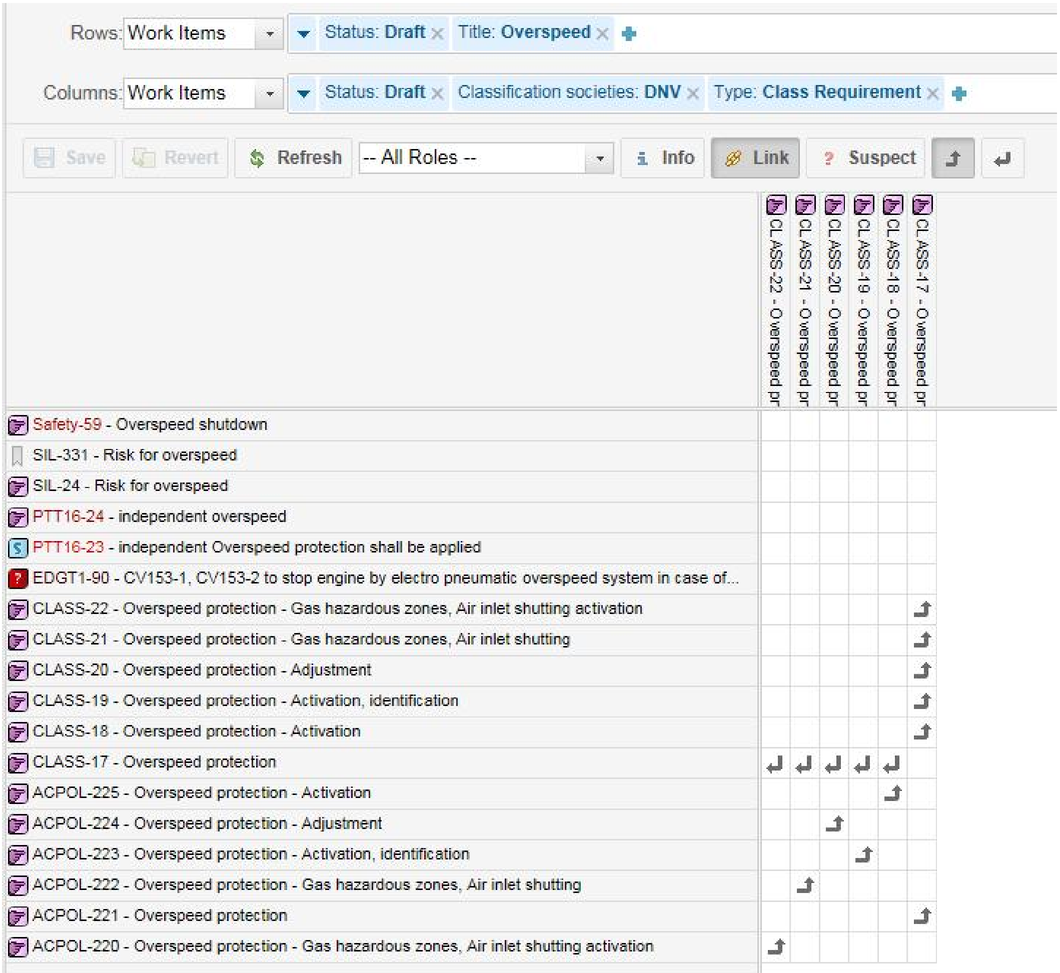


Figure 22. Matrix view executed with searches in Polarion.

To be able to execute searches in this way can play a large role when applying for classification certification. Stakeholders can be easily shown the rules that were valid at a certain point of time and whether they were implemented or not. The searches also serve the developers of control system as they can find the requirements with different themes and statuses.

7 CONCLUSIONS AND EVALUATION

7.1 Conclusions of the Classification Requirements Concept

Requirements are important parts of development processes. Introducing the requirements can be seen as setting up the targets of developing products. This will guide the development through the process to the desired outcome. To fulfill the task the use of requirements management tool is essential as number of these requirements can be enormous.

The assignment of this thesis was to develop a concept for managing requirements originating from classification societies in an application lifecycle management tool. To manage requirements of standards and regulations can be seen as an important part of developing the engine control system and beginning to manage these in the specified requirement management tool can prove to be huge leap towards a better quality and more transparent development process. The use of this kind of tool can also make certification processes easier as it brings requirements traceability into new level.

Classification societies are important operators in maritime safety and regulators amending certification for statutory surveys and thus it can be seen as a valuable task to create a concept for managing requirements of rules they produce. Classification requirements vary from normal requirements from their source and some other features, they, for example, have higher demands on traceability and different validity aspects than normal requirements, so it is well argued for them to have their own category.

The concept was created by combining theoretical knowledge and knowledge gathered by discussing with experts in the A&C department. The knowledge gathered was then fitted into the existing Polarion structure. The process started with getting familiar with the tool and requirements management in general as well as the development process and requirements management in the department. The concept was developed mainly in the order documented in this thesis. As there were discussions with the experts all the way during the thesis process as the concept was being developed, there was no separate validation process for the concept. At the end, when concept and guidelines were ready, the content was walked through with the C&P team members and was validated that way.

In many ways, creating a concept for requirements management in a specified tool is a task to get familiar with the environment it is going to be used. Polarion is a highly editable tool and to adapt it to work in a specific environment is a task of getting familiar with the environment, in this case with the development process in use at A&C. The capacity of Polarion is huge but a lot of time and effort was needed to develop it to suit the needed use and to cover the whole development process. This thesis covered just a small part of the possible use of Polarion.

As the overall development of Polarion at A&C is still going on, the new Polarion structure and the way-of-working are not fully rolled-out in the department, there was no possibility to truly test the usability of classification requirement concept. The usability of classification requirements was only tested inside the C&P team. The true usability of the concept will be seen during a few years of time as the use of Polarion spreads to cover the whole development process.

7.2 Evaluation of the Thesis Process

Writing this thesis was a jump into unknown. To fulfill the assignment there was lots of knowledge to be gathered. All the fields included in this thesis were unfamiliar and it took quite a long time to deeply understand what is meant by requirements management, how the department handles development processes and how Polarion and other requirements management tools work. The thesis process started with getting familiar with the tool, with requirements management and with the A&C department.

The combination of using written sources and discussions with experts worked well with developing the concept. Theoretical knowledge worked as a guideline for discussions aiding to ask the right questions. There was no specified plan for arranging the interviews with experts in fields covered in the thesis. In the beginning of the process all the aspects of assignment were so unknown that it felt difficult to schedule interviews. If the phases of development and questions to be asked do not exist, it seems a difficult task to create a schedule for interviews. The discussions were held when there was need for them. Luckily I was positioned to sit close to the experts in this field and they were always keen to answer my questions. If the thesis process would have covered longer period of time, creation of a proper schedule could have been possible.

As a conclusion, it can be said that the process of writing this thesis was most of all a process of learning. I did not only learn about the topics written here but also writing this trained my English skills to a whole new level and a getting familiar with new company introduced me to a new way of working.

The time felt a bit short, I could have used a lot more in aim to understand the different fields of the thesis better. But as the work continues, the process of learning can be kept going.

REFERENCES

/1/ Åkerman, J. AC Info event October 2014 presentation. Wärtsilä intranet. Accessed 18.2.2015. Access rights needed

/2/ Benefits os requirements management. 2015. Intland. Accessed 11.2.2015. http://intland.com/blog/requirements/benefits-of-requirements-management/

/3/ Classification Societies - What, Why and How? 2011. IACS. Accessed 30.1.2015. http://www.iacs.org.uk/document/public/explained/Class_WhatWhy&How.PDF

/4/ Davis, B. Mastering software project requirements. 2013. J. Ross Publishing

/5/ DNV - Rules for Ships / High Speed, Light Craft and Naval Surface Craft, January 2009. P. 31.

/6/ DNV GL company presentation. 2015. DNV GL. Accessed 11.3.2015. http://www.dnvgl.com/about-dnvgl/

/7/ Maciaszek, L., Liong, B. Practical Software engineering. 2005. Pearson

/8/ Maciaszek, L. Requirements analysis and system design, 2001, Pearson

/9/ Managing requirements your way. 2015. Polarion software. Accessed 5.3.2015. http://www.energie-und-technik.de/fileadmin/media/whitepaper/files/158_managingrequirementsyourway.pdf

/10/ Manifesto for Agile Software Development. 2001. Beck, K. et al. http://www.agilemanifesto.org/

/11/ Pfleeger, S. Software Engineering, 1998, Alan APT

/12/ Polarion ALM fact sheet. 2015. Polarion software. Accessed 5.3.2015. https://www.polarion.com/downloads/Polarion_ALM_Fact-Sheet.pdf

/13/ Polarion requirements fact sheet. 2015. Polarion software. Accessed 5.3.2015. https://www.polarion.com/downloads/Polarion_REQUIREMENTS_Fact-Sheet.pdf

/14/ Polarion software. Wikipedia. 2015. Accessed 12.2.2015. http://en.wikipedia.org/wiki/Polarion_Software

/15/ Presentation of agile methods. 2015. SSG Software Services Group. Accessed 4.4.2015.
https://confluence.aps.anl.gov/display/SSG/Agile+Software+Development

/16/ Presentation of the V-model. 2015. Innosophia. Accessed 4.4.2015. http://innosophia.com/wiki/index.php5?title=V-Model

/17/ Quality assurance. Wikipedia. 2015. Accessed 2.4.2015.
http://en.wikipedia.org/wiki/Quality_assurance

/18/ Somerville, I. Software engineering, 2001, Addison-Wesley, p 56

/19/ Sommerville, I., Sawyer, P. 1997. Requirements engineering: A good practice guide. John Wiley & Sons

/20/ Wärtsilä company presentation. 2015. Wärtsilä. Accessed 16.2.2015. http://www.wartsila.com/about

/21/ Wärtsilä company, history. 2015. Wärtsilä. Accessed 16.2.2015. http://www.wartsila.com/about/history

/22/ Wärtsilä Corporation result presentation 2014. Wärtsilä. Accessed 16.2.2015. http://www.wartsila.com/docs/default-source/investors/financial-materials/interim-reports/resultpresentationq414.pdf?sfvrsn=4

/23/ Wärtsilä Ship Power presentation. 2015. Wärtsilä. Accessed 18.2.2015 http://www.wartsilareports.com/en-US/2014/ar/this-is-wartsila/ship-power/

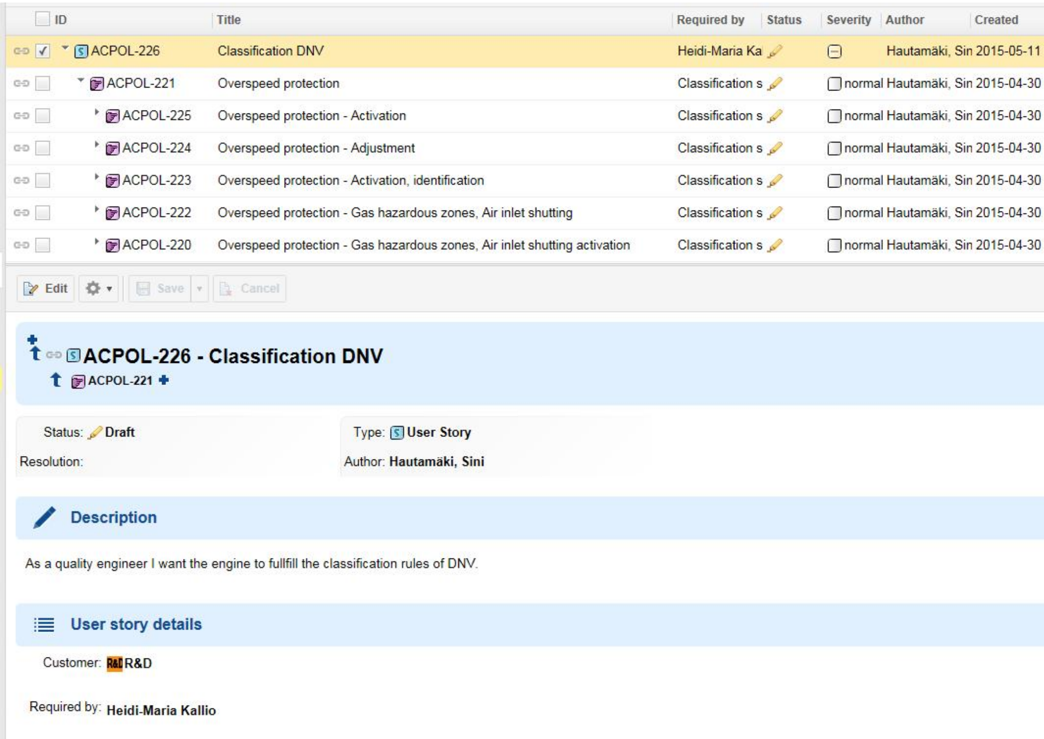
/24/ Wiegers, K. 2006. Testing the Requirements**.** Microsoft Press. (http://www.jamasoftware.com/wp-content/uploads/documents/wiegers-getting-the-most-out-of-requirements-management-tool.pdf)

/25/ Wiegers, K. 2006. More about software requirements. Microsoft press

/26/ Workflow. Wikipedia. 2015. Accessed 26.3.2015. http://en.wikipedia.org/wiki/Workflow

APPENDIX 1

Example of User story in Polarion



APPENDIX 2

Example of system requirement duplicated from classification requirement.