CHALMERS

Requirements Engineering and Management
A development of a Requirements Management Metrics portal

Master of Science Thesis in Networks and Distributed Systems

ALEXANDER JONSSON
KEBING HOU

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Göteborg, Sweden, 2009

Requirements Engineering and Management
A development of a Requirements Management Metrics Portal

Alexander Jonsson
Kebing Hou

Examiner: Arne Linde

Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden

Requirements Engineering & Management

A development of a Requirements Management Metrics Portal

Alexander Jonsson
Kebing Hou

Abstract

This thesis report describes the benefits of employing metrics within Volvo Requirements Management and Project Management. It further describes the effectiveness of implementing these metrics in an automated project measuring system such as the prototype developed. Disciplines such a Systems Engineering, Requirements Engineering and Requirements Management were extensively studied in order to have enough knowledge to propose useful metrics. A generic workflow for Volvo Complete Offer and Embedded Software departments was also proposed which could serve as the basis for creating new metrics that meet the organisation's goals. The prototype developed, Requirements Management Metrics Portal, gathers data (i.e. requirements) from the databases provided by Volvo, calculates the metrics using predefined base-measures and visualizes the results in charts as the chosen graphical presentation format.

Keywords: Requirements Engineering, Requirements Management, Measures, Metrics, Process measurement.

Preface

This master thesis, elaborated by Kebing Hou and Alexander Jonsson, is a required document in order to conclude the Master Programme within the Computer Science and Engineering Department at Chalmers University of Technology.

The work was carried at the Volvo IT in the Electrical and Systems Engineering department located in Lindhomen Science Park, Gothenburg.

We would like to thank our supervisor at Volvo IT, Fredrik Berglund, for the invaluable support and dedication he gave us and our thesis work. We also want to thank the rest of the guys at Electrical and Systems Engineering that provided great support and input during the thesis work period, namely Sasko Cuklev, Carsten Lindgren and Mikael Hansson. Moreover, we are also grateful to our examinator Arne Linde for his support and advice.

Finally, we would like to thank our families and friends for the the love, motivation and support they have given us throughout the thesis work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Requirements Management Process

Requirements Management (RM) is the process that involves eliciting, documenting, prioritizing, analyzing, agreeing on requirements, controlling changes and negotiating with involved stakeholders. RM is the process of increasing the value of requirements after the requirements elicitation process has taken place. In order for RM to keep a high level of quality in the requirements, changes in requirements need to be meticulously analysed to acquire an understanding of how much effort and resources have been used to make this change. This is a good method to cope with changes in requirement because it prevents the requirements from deterioration leading to requirements that are little meaningful.

Within the Volvo Group, several Business Areas use prescribed processes, methods and tools for RM. For instance, this includes setting prerequisites for individual development projects, establishing complete vehicle requirements and detailing requirements on vehicle modules, systems and components (See Figure 1.1). In a typical vehicle projects thousands of requirements are established documented and followed-up. For instance:

Project prerequisites:

- This product should be leading in fuel-consumption.
- This product should be leading in its segments.

Complete Offer requirements:

- Fuel consumption should be less than...

Vehicle Module requirements:

- The drag coefficient of the cabin should be less than...
- The system shall weight less than...

This process is a vital instrument to secure the functional and quality standard of all development activities. Consequently, this process is highly connected to the project management function.

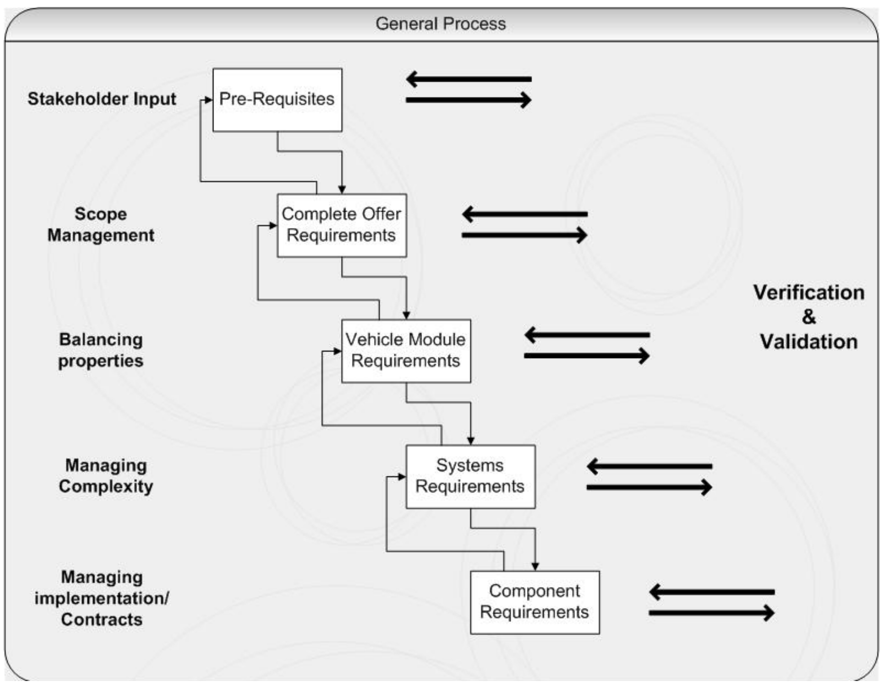

Figure 1.1: General Process model

To effectively manage this process, Requirements Managers need to apply measurement processes. Measures employed in a project could be number of requirements items, change frequency and fullfillment status. Applying measurement processes allows the project managers to acquire an overview of the project, align review-meetings and tune the development process. However, at present time there is no available solution that can automatically summarize this information to the project management group, and compiling the data manually is very time-consuming. Therefore, Volvo proposed this thesis which is about designing and implementing a solution that will allow project managers to analyse a project's health status. They will be able to analyse: Where are we now?, Is the project still on the right track?, What corrective actions should we take?, What aspects of the ongoing development project can we improve?

1.2 Dimensions RM

Dimensions RM is a Requirements Management tool which allows the user to manage requirements effectively through a graphical user interface [12]. Furthermore, Dimensions RM allows the user to create visual application prototypes that shows exactly what needs to be done and generates full traceability reports showing how the requirements are linked to each other. Volvo, uses this tool to manage its project requirements. Dimension RM is an important helping tool for this thesis work since it provides all the data the prototype needs. The prototype and Dimensions RM will be strongly linked to each other in the sense that the Dimensions RM will output the data needed while the solution will take that data as input. Indeed there are other tools that Volvo use as SE-tools for example, but in this case, Dimensions RM served its purpose by providing the exact data the prototype needed.

1.3 Global Development Process

The Volvo Corporate Global Development Process (GDP) product development structure allows the organisation to effectively and within short time periods develop a product. These two mentioned attributes are imperative to keep a highly competitive product development methodology and above all, to keep the customers satisfied.

Since the amount of man-hours may vastly differ to realize a product change, the principal objective of GDP is to deliver the right product with the right quality, at the right time, cost and risk level with features that meet or exceed customer expectations.

1.3.1 Phases

The GDP is divided into phases which represent an important focus in the project:

- Pre-Study Phase
- Concept Study Phase
- Detailed Development Phase
- Final Development Phase

1.3.2 Gates and Project Decision Points

Finally we arrive to one of the main parts of the GDP structure, namely the Gates. These gates act as checkpoints in the project work where the Project Management makes sure the gate standards have been met. Furthermore, Gates exhibits the preparations for the next gate and update the project prediction of final delivery.Moreover, at the project decision points, the project is *accurately* examined by the Decision Body from a corporate, strategic and profitability viewpoint. If project is approved, then more funding is released for next decision point. One vital aspect for the gate audits is to look at requirement agreement, requirement acceptance, and requirment fulfillment status.

Figure 1.2: Global Development Process model

At gates that precede a project decision point, the gate is opened by the Project Steering Committee which also recommends a decision to the Project Decision Body. The Project Decision Body either approves or rejects the project and likewise for its future funding.

This process gives an excellent perception of how Volvo carries out their project development. Understanding the GDP process and its terminology is important for this thesis because it provided the means necessary to carry out the work having in mind the development process.

1.4 Scope

The objective of this thesis work is to introduce a RM Metric Portal that will act as a dashboard for the Requirements Manager, Project Manager or whoever is responsible for a project within the above mentioned areas. The work will be performed in the following way:

- Compiling and visualizing different possible RM metrics based on relevant literature and interviews with experienced engineers.
- Through illustrations gather experience and pre-requisites from potential users.
- Agree on a suitable approach for Analysis, Design and Implementation together with Volvos systems analysts.
- Design and develop a portal that compiles and visualizes RM data gathered from Volvos RM tools and databases.
- Evaluate delivery of the application.

1.5 Delimitations

This thesis report is limited to disciplines such as Systems Engineering, Requirements Engineering and Requirements Management. When developing the metrics, many ideas were not considered since they often did not meet goals in reality. Therefore metrics discarded were not documented in this thesis.

Time was a limiting factor because there was no possibility of designing the application to support other RM tools such as SE-Tools used by other departments. Time also limited the amount of metrics that were implemented in the prototype.

Chapter 2

Method

This chapter describes how the thesis work is done by dividing it into phases.

2.1 Pre-study

In order to cover the scope outlined in Chapter 1, it was necessary to study and obtain knowledge about Systems Engineering, Requirements Engineering, Requirements Management, measures, measurement processes and metrics. To collect books and articles from Chalmers Library and searching for information within the Volvo knowledge databases turned out to be an effective way of gathering information. Although it was slightly confusing with different viewpoints from different theories in the beginning, pre-study did help understand concepts and principles of multiple areas.

2.2 Workshops

Having workshops was an effective approach to combine knowledge, experience and ideas from multiple areas. Current Volvo requirements management processes were discussed. These workshops took place within the Volvo facilities where we all gathered and brainstormed together on possible solutions for metrics. The idea of having workshops was to create useful metrics together with experienced engineers at Volvo.

2.3 Metric Construct Approach

A bottom-up metric construct model (as shown in Figure 2.1) was employed for building some of the metrics presented in this thesis. To clarify "some", we mean that the metrics construct approaches were different since they were developed for two different departments within Volvo. These departments are explained further down.

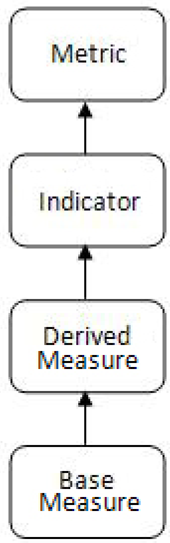

Figure 2.1: Metric construct model

- A base measure is a measure, defined by a specified measurement method, of a measurable property or characteristic of the requirements management process. A value for the measure is produced by executing the measurement method.
- A derived measure is a measure or a quantity that is defined as a function of two or more base measures and/or derived measures. A derived measure captures information about more than two measurable properties[10].
- An indicator provides an estimation or evaluation based on an analysis model which is an algorithm involving two or more base measures

and/or derived measures. In the following chapters, we will see that a graphical chart performs as an indicator for implementation.

- A metric performs as a collection of indicators, interpretations, and recommendations provided to the decision maker as an output of the measurement process[10].

  One example of building a metric for a software development with this model is shown in Figure 2.2. Base measure Effort and Size count the number of hours spent on coding and the number of semicolons in the code. The derived measure, productivity, is the result of applying the measurement function on base measures. After comparing a threshold with the derived measure productivity, an indicator productivity evaluation is produced and serves as the ultimate metric together with criteria information.

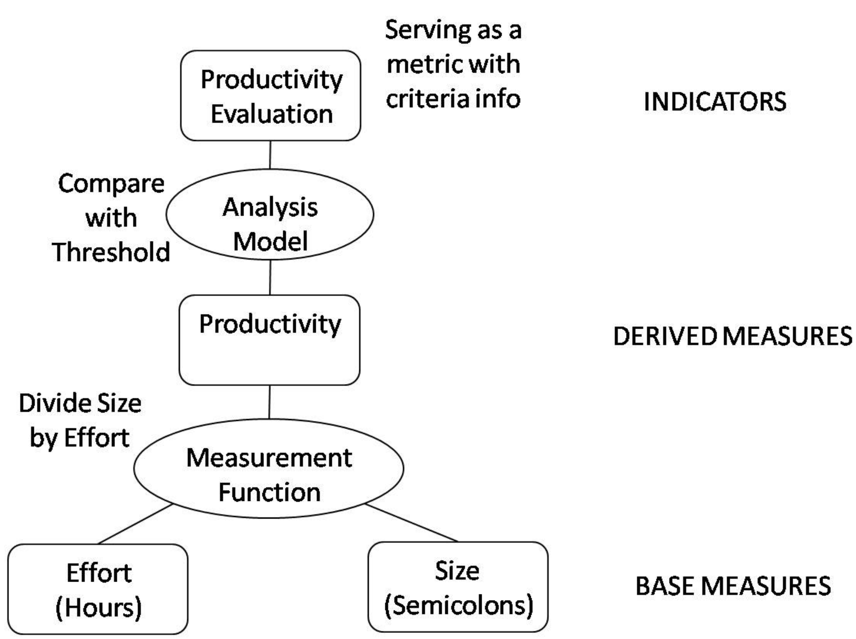

Figure 2.2: Metric construct — Productivity example

2.4 Implementation

With the purpose of developing an RM metrics portal for Volvo Requirements Management and project management, a software engineering method waterfall model was employed in this phase. The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design (validation), Construction, Testing and maintenance as shown in Figure 2.3.

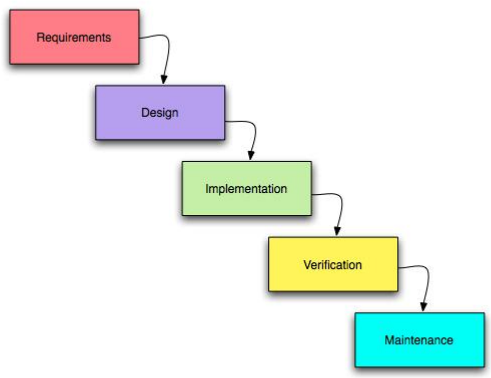

Figure 2.3: Waterfall model

- Requirements: Software Requirement Analysis is also known as feasibility study. Requirements for RMMP were established by interviewing Systems Engineering and Requirements Management engineers, studying RM tools and investigating proposed metrics from workshops.
- Design: In System Analysis and Design phase, the whole software development process, the overall software structure and its outlay are defined. The metric construct model described in last section, the intranet environment at Volvo and the manipulation on RM tools were taken into account while doing the RMMP design.
- Implementation: With the help of ASP.NET, which is a web application framework applied for development, the project was divided into different layers. This enables parallel coding in layers and improves the reusability of the code.

- Verification: Different testing methods are available to detect the bugs that were committed during the previous phases. Unit Tests were done by the two authors of this thesis while coding and integrated tests were done by engineers via Volvo intranet after RMMP was deployed on the server.
- Maintenance: RMMP is currently running on the server at Volvo measuring the test project on a production server. Configurations require to be done when the test project is changed or replaced. In addition, updates are not available for this prototype.

Chapter 3

Theory

3.1 Systems Engineering

Complex projects are usually separated into different process areas which are closely related to each other. One common separation is shown in Figure 3.1.

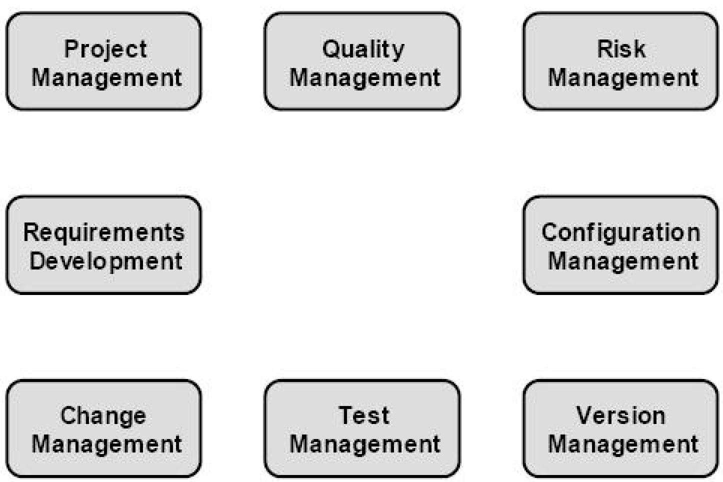

Figure 3.1: Separation of project activities into different process areas

In general, this separation is called systems engineering. The function of systems engineering is to guide the engineering of complex systems[8]. It is easier and more reliable to establish estimation and plans with the whole complex project divided into a set of manageable pieces. People with different experiences may carry out activities using different processes and methods in corresponding areas.

3.2 Requirements Engineering

Requirements engineering is equivalent to the term requirements development shown in Figure 1.1 as a part of the separation. It is a set of activities with respect to identifying and communicating what shall actually be built and act as a bridge connecting the needs of the whole system and all kinds of stakeholders with the development process.

The classic V-model indicates various stages of development and the relationship between requirements and testing. The whole development process is also viewed in terms of layer by the V-model. Requirements engineering addresses the concern to each layer as shown in Figure 3.2. Although slightly different processes may be used at each level, the basic pattern of requirements use is the same[7].

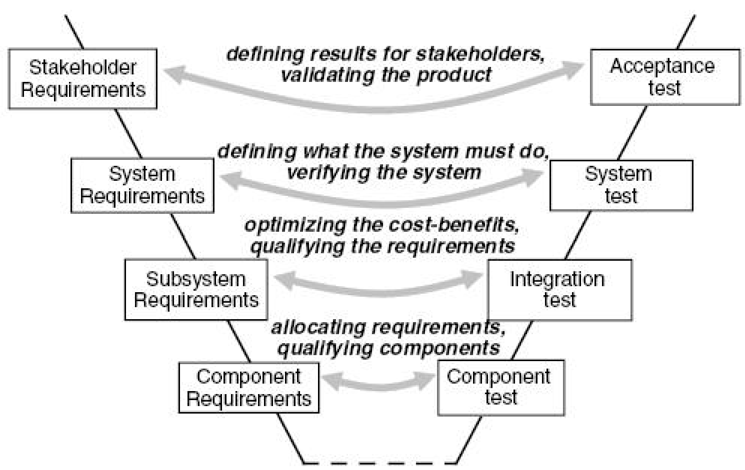

Figure 3.2: Requirements engineering in V-model layers

The main process carried out in requirements engineering is the requirements definition process, which contains two sub processes: definition of scope and definition of requirements.

3.3 Requirements Management

Requirements Management acts as a collection of systems engineering processes that interfaces with requirements engineering. The RM process keeps track of all requirements changes and configurations. It also follows up requirement fulfillment and V&V (validation and verification) status.

3.3.1 What is requirements management

Requirements management is defined as the set of activities which ensures that the requirements information is always up to date and can be accessed by all project staff that may benefit from it. In other words, requirements management integrates all relevant pieces of information from all the other systems engineering disciplines.[6]

3.3.2 Why requirements management

Requirements engineering, which involves no evaluation of requirements, only cares about issues such as how to formulate the visions found inside the heads of the stakeholders, whether a requirement is too expensive or not, whether a requirement makes sense or not, etc. A lot of problems, such as the ones given as examples below, may arise if the development process runs without requirements management.

- Requirements are not feasible
- Requirements are not testable
- Requirements have to be evaluated every time stakeholders propose a change request without documenting changes
- We cannot tell the difference between old and new versions or what is improved without documenting relevant updated information

With many other existing similar problems during the process, requirements management reduces the risk and complexity and makes the process more organized.

3.3.3 Methods in Requirements Management

Some of the most common methods of requirements management are given as follows:

- Identifiability
- Filterability
- Traceability
- Linking
- User rights
- Baselining

Identifiability: Each single requirement is supposed to be identifiable. This can be achieved by assigning each requirement a unique number. Since reassigning identifiers after requirements are deleted may cause misunderstandings and ambiguities, adding a unique prefix to the identifier for each individual specification enables all requirements to be really uniquely identifiable.

Filterability: RM keeps the information together and only extracts what is needed at the moment. If people share several pieces of information it is necessary that they can all extract them.

Traceability: Traceability covers two important aspects: the first aspect is traceability between different pieces of information at a moment. The second aspect is traceability of one piece of information over time.

Linking: Linking is the documentation of relationships between different pieces of relevant information associated with requirements. For example, linking the requirements information to the test information means documenting somehow which requirements will be tested by which test cases and which test cases cover which requirements[6].

User rights: Since all project members share common information, rules regarding the administration of this information are necessary. For example, test managers normally only have read access to requirements. Hence it is important to define an information access policy to make sure that data is only seen and edited by a specific set of project members according to predefined rules.

Baselining: Baselines are commonly used in organised projects. A baseline performs as a snapshot of a collection of requirements at a particular time. Once requirements are baselined, for instance, in a certain phase, they are not supposed to be changed anymore. A requirement is frozen with the version it holds when baselined. However, new versions may be elicited without changing the baselined versions after baselining. Differences among baselines are observed by reports in some way when evaluations take place. An example is shown in Figure 3.3 where there are four requirements with a couple of individual versions. Initially all the requirements are baselined with version 1 as Baseline 1 and each requirement with version 1 cannot be changed anymore. New versions however are developed over time for all requirements and at the point when Baseline 2 takes place, each requirement has a new version except for Requirement 3. In this case current requirements with current versions are baselined as Baseline 2.

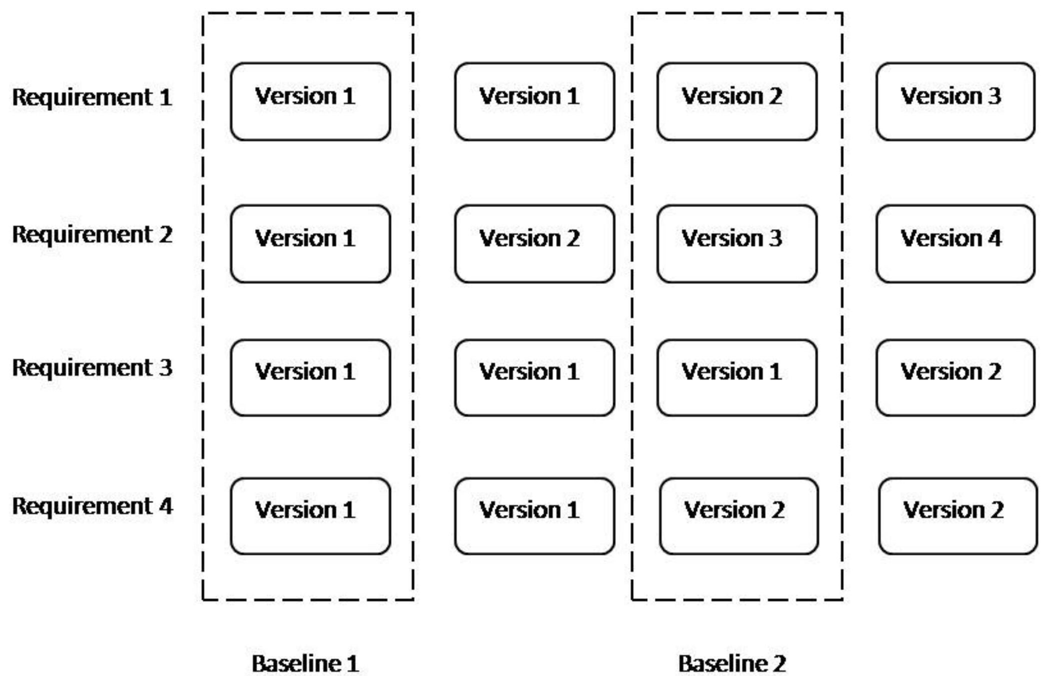

Figure 3.3: Baselining with developing requirements

3.3.4 Requirements Volatility

Stakeholders are the principal contributors and creators of requirements within the development of a system/product. Within every (initial) phase in a project, requirements are subject to change due to the fact that stakeholders often have various needs and goals which, in many cases, might produce conflicts. These conflicts can be interpreted as changes and errors that need measurement. When requirements have many changes over time, they have a tendency to be highly volatile [9].

According to Sommerville [13], stakeholders may also have contradicting interests which produce even more volatility in requirements. These requirements must then evolve to reflect this changed view of the system in development. Furthermore, the volatility on requirements depend on several factors. For instance, one might be organisational complexity, the process maturity of the company, the phase of the life cycle, the volatility of the market, etc[9]. It seems logic to assume that the more complex system/product is being developed, the higher the volatility due to an increased number of interacting components.
Boehm[4] proposes incremental development in order to control requirements volatility. When using single increment, it is very hard to postpone the stakeholders requests which results in a vast amount of effects distributed through the product and project's schedules. Incremental development in the other hand allows each increment to follow a plan less prone to changes[9]. However, requirements will change but this method will make it easier to handle the volatility of requirements.

3.3.5 Requirements Traceability

Traceability of requirements performs documenting links between different requirements. Requirements are elicited from different sources, therefore the possibility to trace back to the origin of each requirement should be ensured. Changes made by anyone responsible should be documented as well. Gotel and Finkelstein[5] define requirements traceability as follows:

> *"The requirements traceability is the ability to describe and follow the life of a requirement, in both a forward and backward direction, i.e. from its origins, through its development and spec-*

ification, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases"

Traceability of requirements is important because it follows the changes a requirement has gone through. It also identifies when the change occured and the reason behind it. Furthermore, traceability also provides information which helps to determine if all the relations between a number of requirements have been addressed.
Implicitly speaking, requirements traceability enables a requirement manager to trace a specific requirement from its creation to implementation along with all the corresponding changes made throughout its lifecycle. Moreover, when a change is made on one requirement it might propagate changes to other requirements that were linked to this one, which gives place for traceability which also asissts in assessing the impact of these changes made on requirements as well as assisting in planning changes, estimating efforts in changes among other things. Additionally, using requirements traceability allows requirements to be prioritized according to the stakeholders' needs which serve as basis in the requirements elicitation phase.

3.4 Requirements management tools

Nowadays there are various requirements management tools applied in different areas. In some cases, multiple tools are used at different levels in a complex project. Requirements management activities are achieved by the functionalities of these tools in varying degrees, which enables efficient requirements management in industrial environment.

Dimensions RM is a software solution developed by Serena for requirements management. It provides an enterprise-wise platform interfacing with Oracle database and helps create, manage and track requirements data throughout the development process. Dimensions RM enables tracking requirements across different organizations and understanding the impact that various requirements have on cost, schedule, and time-to-market can have a dramatic effect on project success[11]. Traceability reports, supported by this tool, make it effective to evaluate changes of the requirements and show relationships among the requirements. Users may facilitate change management by processing change requests in a systematic way and communicate with team members through comments, polls or E-mail notification.

3.5 Metrics and Measures

Metrics and measures form an essential part of measurement within Requirements Management and Engineering. Below we present an introduction to the previously named terms to acquire an increased perception around the subjects.

3.5.1 Measures

To have an understanding of which requirements are essential for a project one must employ the use of measures. Measures are important for the development of a product since they give a status of the project to the development team which highers the probability of producing a satisfactory result in the end product. Error identification is vital for a project because it saves resources to detect errors in the earlier phases than in the later phases of the development. Measures can eliminate these errors.

Definition of Measures

- The concept of measure is introduced as the result of counting otherwise quantifying an attribute of a process, project or product.
- Measures are numerical values assigned to attributes according to defined criteria.
- The raw data from which indicators are calculated.[2]

A few examples of a measure would be:

1. total number of requirements.
2. total number of code lines.

3.5.2 Metrics

Definition of Metrics

By definition a metric (a.k.a indicator) is a measure or a combination of measures that provides insight into an issue or concept. Metrics are often comparisons, such as planned versus actual measures, which are usually presented in graphs or tables. Metrics can describe the current situation (current indicators) or predict the future situation (leading indicators) with respect to an issue[2].

"You can't control what you can't measure." — Tom de Marco

In order for the project to be successful, one need to create a set of accurate metrics that provide the information needed to meet the goals set by the organisation. Additionally, analysis, identification and compilation of relevant data for evaluation and validation is also part of creating quality metrics. Validation of data is imperative to make sure the entities given by the customer meet the goals, which should reflect the organisational needs.

3.5.3 Types of metrics identified

During the pre-study period of the project a few sort of metrics were identified and categorized as presented below.

Product Metrics

This type of metrics measure the quality of an end item. The item could be anything from the shipped product to the system design specification document to a quantifiable measure of service performed.[2] In a Software Engineering perspective, one can say that they deal with characteristics of the source code, measure requirements, size of program and design among other. This metric is divided into subcategories and only the most relevant are listed:

- Size Metrics: measure the size of the software as Lines Of Code (LOC), function points, etc.
- Complex Metrics: allows us to manage/control the process of the software development and measure the complexity of the program.
- Halstead's Product Metrics: Halstead proposed a set of metrics and a program vocabulary which included unique operators and unique operands in a program.
- Quality Metrics: measure the software quality in every phase of the development
- Defect Metric: measures the the faults and failures found in the system.

Process Metrics

Process metrics measure the success of an ongoing activity which gives an overview of the process stability and improvement possibilites. Furthermore, historical data can be used as a basis for comparisons for similar future projects. A couple of brief examples of this sort of metric might be time, persons involved in the development and rework factors etc. These metrics are used to improve the process and the management of the requirements of a development.[2]

Requirements Metrics

Requirement metrics validate the written requirements against the actual requirements, they evaluate if the requirements are complete or not.[1]
Some examples of possible metrics used for requirements engineering or management could be:

- Size of the requirements.
- Requirements traceability.
- Requirements completeness.
- Requirements volatility.

3.5.4 Quality characteristics of Metrics

A quality metric should be meaningful (i.e., it represents project progress or an expected performance of a system), easy to retrieve, understand and therefore helpful to make a decision. There are various metrics a Requirements Manager can make use of in order to obtain an objective measure of the status of a project. This gives opportunity for the manager to take cautional steps and if necessary, avoid the project going off track. Other metrics can be indicators of how well the project is doing, (i.e. a project could be ahead of schedule) allowing a delivery of increased business value than expected.

Precise metrics are always desirable, but we should be careful with *false precision*. Metrics that count things (i.e. calls answered, monthly sales, number of changes on X) can often be measured precisely, but if they do not adjust the motivations of the managers and employees with the long-term goals, they could become counterproductive.

3.5.5 The Importance of Metrics

Measurements are an extremely important part of quality assurance. In order to be sure the quality standards set by the customer or the organisation in need of the metrics have been met, the periodic results must be measured to form a basis for a decision as to whether improve and/or correct a process or project. According to Hood [6], the results should be measured against a predefined benchmark. Since benchmarks change very rarely, the measurement succeeds in producing a relatively "objective" evaluation of the results. Without measurements one can only guess what the outcome of a work process will be.

"To measure is to know " — J.C.Maxwell

Why metrics

Organizations usually perform measurement for one of the following reasons[2]:

- *Characterize* or gain an understanding of their processes, project progress and/or products and establish baselines for future assessment of these.
- *Evaluate* or determine project progress with respect to plans
- *Predict* resources, schedule and performance to support planning and trades.
- *Identify* improvement opportunities for progress, processes and/or products, such as roadblocks to progress, root causes of problems in products, and inefficiencies in processes.

Although the above sentences make a clear point, we would like to further explain the reasons why one would want to employ the use of metrics in a project.
As a product development project begins, requirements start pouring in. It is highly possible that most of the requirements will be changed during the project life cycle. Moreover, changes to the requirements are prefferable to be expected in the early phases of the project as the stakeholders, development team and solution providers reach consensus of what the system/product should and should not do. The simple reason to why changes are preffered to be performed early during the project, is to save resources and to avoid

having the project going off track potentially leading to project failure. Of course this is all relative to the size of the project but in extreme cases a vast amount of economic resources are spent on a project that is ultimately cancelled.

3.5.6 Measurement Process

Insight into the progress of a project or system in development is acquired by ensuring a solid measurement process. Insight provides the decision maker with information that allows him/her to take the correct decisions. These decisions should be made having in mind that the measures used solve current issues at hand and that those issues may change with time. The activities that form part of the measurement processes will be discussed further down through this thesis report in the corresponding section.

3.5.7 Attributes of Metrics

Controlling and management units must be conscious that a single metric can at most measure a part or one aspect of reality. In many cases the values of single individual metrics are often insufficient to allow a rock-solid judgement. Therefore Colin [6] proposes to carefully select the metrics and describe their respective attributes:

- Goals supported by the metric
- Customers of the metric
- Interval of measurement
- Data or measurements used
- Unit of measurement
- Data source (Effort required to capture/reliability)
- Interpretation of results
- Strengths and weaknesses of the metric
- Prerequisites for measurement
- Presentation format of the metric

3.6 RM Metrics Examples

Possible Metrics			
Metric	**Formula**	**Attribute**	**Purpose**
CO Requirements Growth	$N_G + N_R + N_Y$ (3.1)	Characterize, Predict, Identify	To assess the variations in the number of requirements in time.
CO Requirements Volatility	$\frac{N_G + N_R + N_Y}{N_{Ch}}$ (3.2)	Identify, Evaluate	Measures requirement changes over a time period with the sole purpose to analyse the reasons for change as well as the rate in which requirements change.
CO Requirements Rate of Acceptance	$R_{N_Y} + R_{N_G} + R_{N_R}$ (3.3)	Charactertize, Identify	Measures the level of acceptance for each requirement for each responsible involved.
Test Coverage	$\frac{N_C}{N_T}$ (3.4)	Evalute, Identify	To measure the ratio of requirements that have been linked to test cases over time.

ESW Requirements Verification	$$\frac{N_V}{N_T} \quad (3.5)$$	Evalute, Identify	To easure the ratio of requirements that have been verified over time.
ESW Addressed Requirements	$$\frac{N_A}{N_T} \quad (3.6)$$	Evalute, Identify	To measure the ratio of requirements that have been addressed over time.

Table 3.1: RM Metrics Examples

Where:

- N_Y is the total number of YELLOW requirements.
- N_G is the total number of GREEN requirements.
- N_R is the total number of RED requirements.
- N_{Ch} is the total number of CHANGED requirements.
- N_C is the total number of requirements that have been linked to test cases.
- N_V is the total number of requirements that have been verified.
- N_A is the total number of requirements that have been addressed (linked to requirements in the layer below).
- N_T is the total number of requirements.
- R represents a type of role. The roles responsible are:
  1. Complete Offer Requirements Owner - CORO
  2. Stakeholder - Stkh
  3. Solution Provider - SP

- R_{N_Y} is the total number of YELLOW requirements for a specific role.
- R_{N_G} is the total number of GREEN requirements for a specific role.
- R_{N_R} is the total number of RED requirements for a specific role.

3.6.1 RM Metrics Research

Annabella Loconsole

Annabella Loconsole is a former PhD student who wrote her thesis about RM measures. The goal to achieve stated in her thesis, *Definition and validation of requirements management measures*, is to improve the management of requirements by using software measurements. In her work, she takes a quantitative approach in requirements management which she describes as "to monitor the RM activities and requirements volatility through software measurement". The research presented in her thesis is founded on software engineering and software measurement. According to Loconsole, Software measurement allows for defining the degree of success or failure quantitatively, for a product, a process or a person. She also states that when a project manager needs to measure a project he/she has to define software measures and a model associated with these measures. The model has to describe entity and attributes being measured, the domain and range of the measures and the relationship among the measures. Moreover, she also talks about validation of measures whose purpose is to show that the measures defined for a special purpose are indeed useful in practice.

Annabella Loconsole's PhD thesis focus lies on Software Engineering and related product development. The measurement discussed in her thesis reflects attributes that any project manager within mostly any field requires to measure the status of a project. Even though she looks at things from a software engineering perspective, one can apply the same principles for defining measures and measurements in other fields or industries. For example, measures such as number of requirements are vital in other fields too since it is imperative to measure the number of requirements variations over time in order to acquire a better picture of how volatile the requirements are. In general, this PhD thesis has given valuable input that may be applied within Systems Engineering measurement processes.

INCOSE

The International Council on Systems Engineering (INCOSE) is a not-for-profit membership organization founded to develop and disseminate the interdisciplinary principles and practices that enable the realization of successful systems. Measurement Working Group (MWG) within INCOSE has been investigating Systems Engineering measurement which includes RM measurement. Its research focuses on the measurement process, purposes of measurement and proposing measures for different areas of Systems Engineering. The measures it establishes such as Review Rate and Requirements Stability are related to RM and provide a Systems Engineering perspective for measure production in other specific RM environments.

INCOSE provides guidance on how to effectively use measurement, avoid its misuse, select good measures, obtain the benefits from correct use of measurement, and find references to other resources that discuss more specialized topics in measurement. Furthermore, it provides a insight into the measurement process which defines planning of what will be measured, how the measurement will be performed, how the data will be analyzed, what reporting is needed, what actions will be taken for the results, and who is responsible for each of these activities[2].The principles and infrastructures covered by INCOSE give fundamental supports to all measurement activities including measurement program.

3.7 The Goal/Question/Metric process

The Goal/Question/Metric method is widely used to identify useful metrics for a project. It was been successfully applied for process improvement and software metrics programs. The main goals of this metric creation process is to provide project control and process improvement. In other words, in order to improve processes one have to define measurement goals which are elaborated first as questions which later, after a bit of work, turn into metrics. The metrics created from the questions will be the answers to these questions. Furthermore, the GQM paradigm helps define metrics appropiate to a stated information need. It requires definition of *who* need to know *what* information, and *why* and *when* they need to know it [14].

According to [3], the GQM approach is based upon the assumption that for an organization to measure with purpose it must first specify the goals for itself and its projects, then it should find a link between the goals and the data to define those goals and eventually provide a framework for interpreting the data with respect to the stated goals.

The GQM process provides several steps:

1. State the information goal: Identify the stakeholders and investigate what each type wants to know and wants to do with the information.

2. Ask the question: What question(s) are relevent to ensure that the goals have been met?

3. Identify the specific parameters that must be measured to answer the question posed in step 2.

4. Apply the metrics selected, evaluate their usefulness, and go to step 1 or 2 when indicated.

To further elaborate the above steps one can see that the goal is on top of the hierarchical structure as shown in Figure 3.4 which is divided into questions that in the end will address how a particular goal will be met. A goal can also be refined into manageable subgoals. Each question is then refined into metrics, which can be both subjective or objective. The same metric can be used to answer other questions under the same goal [3].

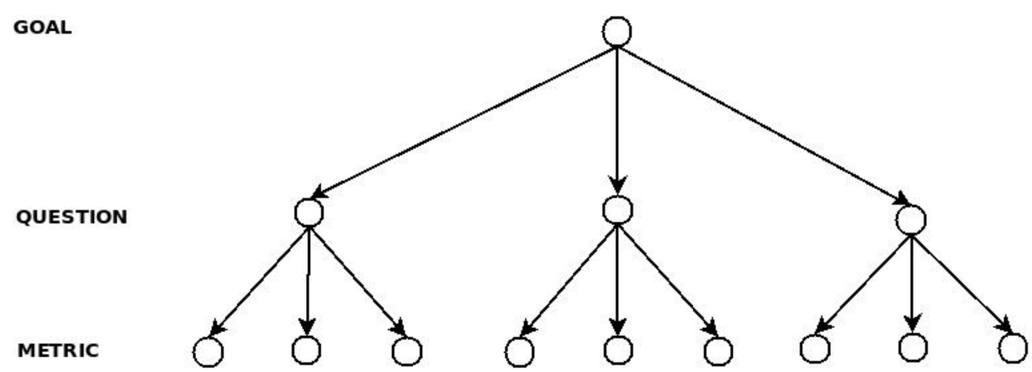

Figure 3.4: The Goal Question Metric approach

Chapter 4

Metrics for Volvo

In the initial phase of the project, it was planned to develop metrics and the RMMP for Complete Offer (CO) only but as the project evolved, other Volvo departments, such as Embedded Software (ESW), became aware of the great significance of the RMMP and the use of metrics. Therefore, the RMMP was redesigned to fit the needs from both divisions.

4.1 WorkFlow

Requirement evolution process varies according to the selection of requirements management theories and tools. Practical issues, such as cost and project complexity, should be taken into account when the evolution process is being established. To some extent, the metric construct depends on the evolution process following a certain work flow.

4.1.1 Basic work flow

In order to build powerful requirements management metrics, the evolution process of Dimensions RM is applied in this thesis as a basic work flow which does not support necessary metrics sufficiently. Thus a new work flow was proposed that would optimize the working process. This implies constructing additional metrics that are strongly related to the workflow. For instance CO requirements are banlanced during the development process but no activities are taken to measure balancing progress. If a metric indicating balancing progress is needed, the work flow is proposed to be improved in some way.

The basic work flow processes requirements using a set of statuses as shown in Figure 4.1.

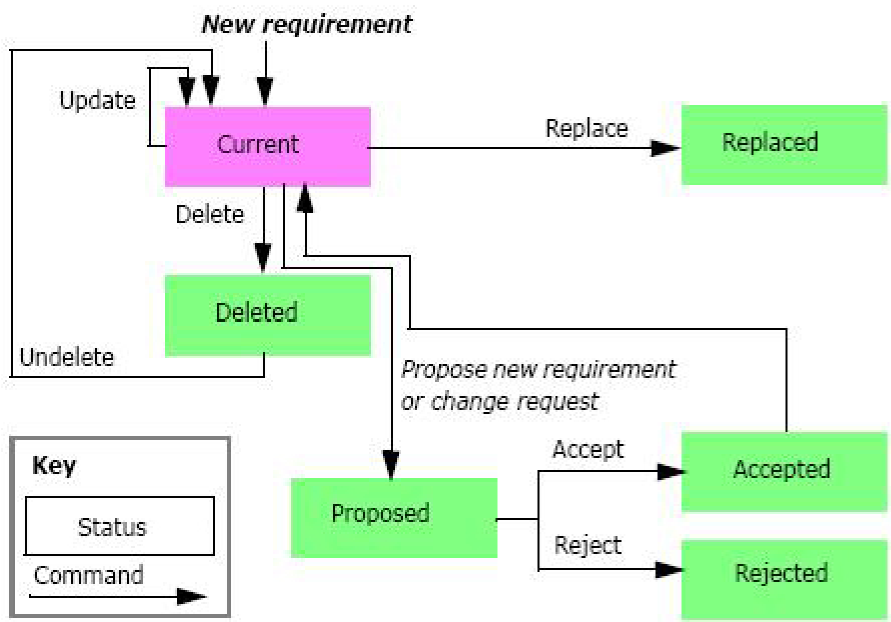

Figure 4.1: Basic requirement evolution work flow from Dimensions RM

The statuses are defined as follows:

- Current: The requirement is the most recent or current version.
- Replaced: The requirement has been replaced by a newer version.
- Proposed: A change request has been made to either change the current requirement or create a new requirement.
- Accepted: A change request was accepted.
- Rejected: A change request was rejected.
- Deleted: The status of the requirement is changed to Deleted, but the requirement remains in the project. The prior version of the requirement, if any, is given a status of Current[11].

Once a new requirement is created it may be updated, replaced and deleted accordingly. When someone submits a change request on a requirement, one copy of the requirement with the status of proposed is created. The status of the proposed requirement becomes rejected when the change request is rejected. On the other hand, the status of proposed requirement becomes accepted when the change request is accepted. In the meantime, a copy of the requirement is created with the status of current.

4.1.2 Proposal of a generic work flow

Although requirements can be elicited and managed with the basic work flow, the management process cannot be effectively measured for industrial needs. In this case, a generic work flow is proposed and(as shown in Figure 4.2) built for creating metrics. There is no doubt that as more activities are involved the management process has been improved according to the generic work flow.

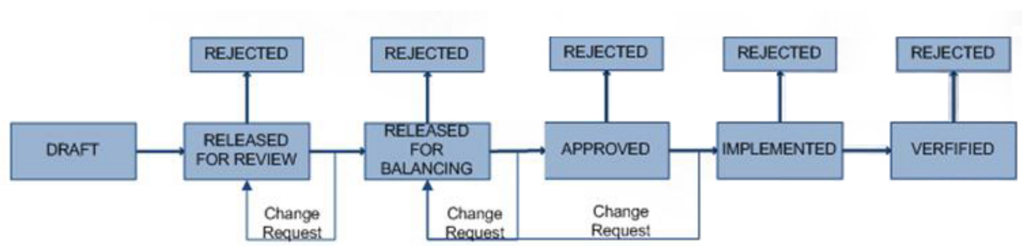

Figure 4.2: Proposed generic requirement evolution work flow

Requirement engineers are responsible for composing the requirement drafts which are checked by requirement managers. Qualified drafts are released for review taken by requirement managers, solution providers and customers. The requirements that everyone has agreed on are released for balancing. Anyone who does not agree proposes a change request and the status of the corresponding requirement goes back to released for review waiting for review again. During balancing, successfully balanced requirements become approved and the requirements to be changed go back to the status of released for balancing. Approved requirements are implemented and verified afterwards. If any change request is proposed during implementation, the status of the corresponding requirement goes back to released for balancing.

Requirements may be rejected at every stage throughout the whole work flow except for the drafts.

4.2 Complete Offer

The workshops performed together with engineers at Complete Offer were done with the purpose to brainstorm, discuss and propose metrics that would fit CO's goals and needs. At first it was thought to analyse, propose and create metrics employing a top-down approach by, for example, using the GQM method. However, this was not the case as many ideas and proposals poured in, the approach used was both top-down and bottom-up. The main reason that facilitated the metric creation process using both approaches was the extensive experience of the engineers, since they knew perfectly *what* was needed.

4.2.1 Proposed Metrics

Although the workshops were truly productive, there was not enough time to brainstorm and create more metrics. Every discussion of a proposed metric covered various issues that could affect the way of work of the organisation. So when creating these metrics, we had to take into considerations several factors such as, how useable is this metric?, is this metric feasible to construct?, is the metric too complex?, do we have the data available for this metric?, etc.

When creating base-measures/metrics, it is important to choose goals that match the organisations goals. Additionally, starting simple and taking time to learn about the metrics one wants to introduce and to analyse their strengths and weaknesses is always a recommendable approach. But most importantly is to create metrics whose *limitations* we do understand.

Below we present the metrics proposed, created and used in our thesis work. We employ the model presented by [6] to present the metrics proposed in order to give the presentation a well-defined structure. in order to save space we have implemented the proposed model within a table.

Requirements Growth

Requirements Growth measures the total amount of requirements for each project phase or date point. The attributes of this metric are shown in B.1.

Requirements Acceptance

Requirements acceptance measures the level of acceptance of requirements. Each requirement is either marked GREEN, YELLOW or RED by each participant in the review. These participants/roles are: Complete Offer Solution Provider, Stakeholder and Complete Offer Requirements Owner. Each and one of these roles may fully agree, agree but with restrictions/modifications, or totally disagree to a requirement's specification. The attributes of this metric are shown in B.2.

Requirements Acceptance metric panel for CO has a navigation menu to the far left where all the different levels within Complete Offer can be viewed as shown in C.7. These levels represent a specific area of requirements within Complete Offer requirements area specification. A user can click on any level to get the Acceptance rate automatically visualized. When visualizing, the RMMP shows three pie charts, which represents the percentage of Acceptance for each role involved in the requirement review.

Requirements Volatility

Requirements volatility measures the rate of change of requirements. This allows to set boundaries of how much volatility can exist during a project. The attributes of this metric are shown in B.3.

4.3 Embedded Software

Embedded Software provides software for ECUs to all truck brands within the Volvo Group. Based on LDC requirements and system design they specify (in Dimensions RM), implement, test, integrate and maintain embedded software systems. In order to improve their progress, in particular the RM process, there is a need to implement metrics.

4.3.1 Proposed Metrics

ESW engineers had a number of base measures that need to be aware of in the process.With the base measures, analysis was done regarding the metric construct model from Chaper 2, the process being carried out at Embedded Software as well as part of the proposed work flow described in Chapter 2. As a result, a number of metrics were constructed in the workshops.

ESW takes care of two types of requirements which are Logical Design Component (LDC) requirements and Software Component (SWC) requirements. The requirements can be baselined whenever needed.

Requirements Growth

Requirements Growth indicates the project size over time by measuring the amount of requirements. It may be used for both monitoring current projects and reviewing historical projects. The attributes of this metric is shown in B.4.

Requirements Growth metric panel presents Requirements Growth for both LDC requirements and SWC requirements. A chart showing the total number of LDC and SWC requirements over time gives users a basis of evaluating the project size and identifying the correctness of the ratio between LDC and SWC requirement size as shown in C.1. When a user clicks on a date point in the chart, detailed data would be presented in the table on the right top. The interval of the chart is optional with a list of weekly, monthly, six months and yearly on the right. This panel also enables users to export charts and raw data to Microsoft Excel with the button and options on the left. Selecting a chart to export and the format (only chart or chart with raw data) from the lists on the left, a user would get the exported data in Microsoft Excel as shown is C.2.

Another bar chart shows the number of LDC requirements for each baseline. Each baseline corresponds to a vehicle unit, which means the requirements for a specific baseline are all the requirements for the corresponding vehicle unit the moment they are baselined. The requirement size for each baseline can be evaluated according to the corresponding vehicle unit size.

Requirements Volatility

A baseline freezes a collection of requirements when it comes to a specific stage of the project. One project usually establishes a large number of baselines. Requirements may be changed or deleted from one baseline to another. New requirements may be added to the system as well. Requirements Volatility indicates the stability of requirements by measuring changes that take place among baselines. The attributes of this metric are shown in B.5.

Requirements Volatility is implemented to indicate the number and ratio of new, changed, unchanged and deleted LDC requirements between consecutive baselines using a column chart as shown in C.3. There is a list of vehicle components such as VMCU (Vehicle Module Control Unit) for users to select, therefore the requirements linked to the selected component can be measured in term of volatility. Requirements that are baselined and linked to the selected component would be presented as a component is selected. RMMP enables users to observe the changes from one baseline to another with the visualized chart.

Requirements Acceptance

Requirements are reviewed in order to analyse its feasibility and are either accepted or rejected. Reviews lead to change requests in some cases. Requirements Acceptance indicates the acceptance degree by measuring review results as projects are running. The attributes of this metric are shown in B.6.

Requirements Acceptance for ESW is implemented to indicate the number of accepted and non-accepted LDC requirements linked to the vehicle component which is selected from the list on the left as well as the number of new, changed, approved and rejected SWC requirements over time using two stacked area charts as shown in C.4. When a user clicks on a data point in either of the charts, detailed numbers and ratios on that day will be presented in an additional pie chart on the left.

Requirements Verification

In the verification stage, requirements are verified by using various methods among which different methods require different effort and staff. Requirements verification indicates an overview of verification methods applied to all requirements. The attributes of this metric are shown in B.7.

Requirements Verification is implemented to indicate the relative number and ratio of each verification method for SWC requirements using a pie chart as shown in C.5. Users can tell how much effort should be put into the present project from the proportion of verification methods since different methods require different resources.

Addressed Requirements

There are links among requirements and LDC requirements are normally linked to SWC requirements while developing. LDC requirements that have been established links to SWC requirements become addressed requirements in requirements management. This metric measures the addressing degree of requirements indicating a certain aspect of project progress. The attributes of this metric are shown in B.8

Addressed Requirements metric panel provides a column chart presenting the number and ratio of LDC requirements which have been linked to SWC requirements in the database as shown in C.6. Like Requirements Volatility, a vehicle component can be selected from a list and requirements that are baselined and linked to the selected component would be presented. Users can evaluate the addressing progress based on the columns in the chart.

4.4 Metrics proposed for future use

The metrics presented below were not implemented since there was not enough time. However, they are presented as proposals for future reference because integrating these additional metrics provides more means for the engineers to measure and control the progress of a project.

4.4.1 Potential rate of change

In reality, Requirements are elicited in pyramid structure where requirements are addressed in layers as shown in Figure 4.3.

In general, one requirement is normally linked to one or more requirements from both higher and lower layers. Although the relationships between linked requirements vary, it is clear that a changed requirement would potentially cause changes on the requirements linked to it and this might happen repeatedly. In other words, changes are possible to propagate in requirements. Measuring potential change rate may help with resource allocation and stability evaluation. However, compared to changed requirements, potentially changed requirements are relatively difficult to measure. One of

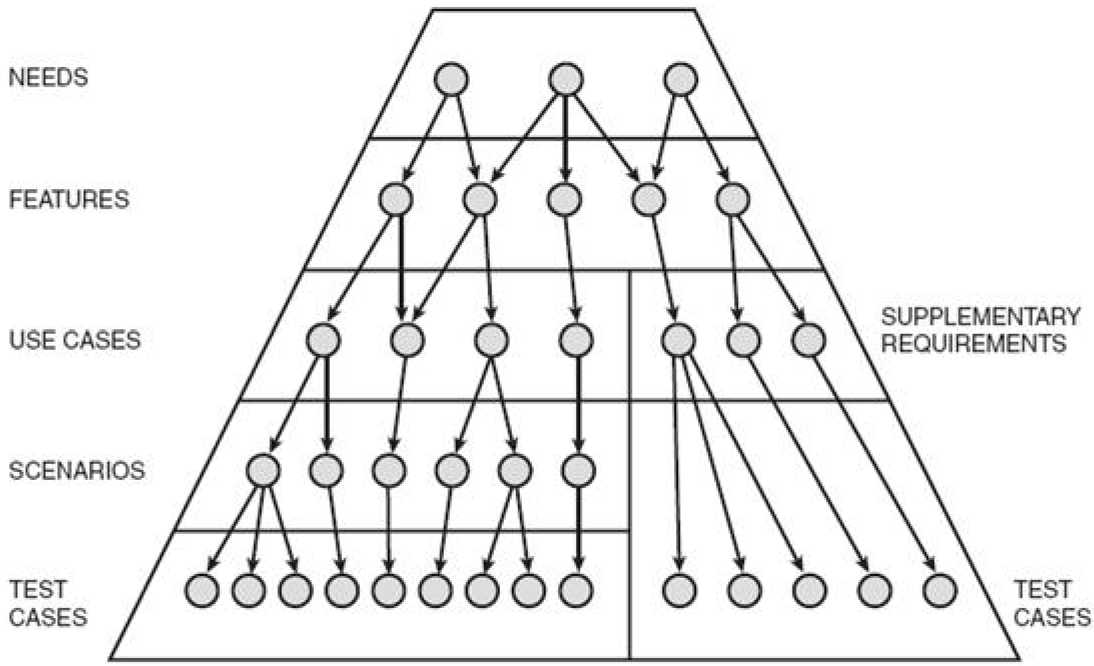

Figure 4.3: Requirements pyramid

reasons is this potentiality might be traced to the very top or very bottom layers if there is no limited degree on measurement. Besides, manually measuring potential change requirements might cause more effort to check if the potential changes exist. Due to the infeasibilities, metric potential change rate is proposed for potential future use in this thesis.

4.4.2 Requirements Maturity

The maturity in a requirement is improtant to measure since the maturity indicates whenever a requirement is ready to be moved onto the next phase in the project in relation to the connection these have to the GDP gates. The requirements maturity metric is necessary to identify when a requirement has evolved from being a broad statement in aproject scope document to a real tangible object that needs to be identified in use cases or included into a requirements specification document.

4.5 Metrics Abuse

If given the opportunity, organisations can abuse metrics. When creating a metric, it is vital to consider every positive and negative aspects of this metric and to modify it if necessary. The main reason why one should be careful with introducing metrics is *abuse*. There are several ways one could abuse a metric, for instance, to gather one type of data and then treat it as it were another type or to give incentives in an inproper manner to task/project responsibles.

Let us look at a brief example of how one can abuse metrics within the industry: Suppose project managers in a company have the task to assess the fulfillment status of each requirement, in other words they have to make sure to specify the fulfilment status of the requirements (i.e. how complete the requirement really is). Then suppose the company gives the project managers an incentive for every fulfilled requirement they manage to specify within each review phase. This will cause the project managers to set a very high fulfillment status to requirements even though perhaps they are not really fulfilled at the level specified maximizing his/her pay while minimizing productivity and making bad use of the metric.

Chapter 5

System Analysis

This chapter describes the analysis of the system developed which is a web-based tool that allows the project responsibles to acquire a project health status which is visualized through graphs and charts. The application takes as input data generated by Dimensions RM, which are thousands of requirements, compiles the data, and then displays the data in a graphical format.

The requirements section describes the user needs by presenting facts about what the system shall and shall not do. This is presented with enough clarity allowing anyone with an engineering background to fully understand the customer needs. The requirements are gathered from the Software Requirement Specification (SRS) document, which you can consult in APPENDIX XSRS.

5.1 Motivation

Engineers today have a time-consuming task when it comes to analyzing the requirements in order to ensure the project is on track. Currently, the process for analyzing a requirement's attribute is done manually, by personally doing the work or checking with other project responsible for their statuses. This turns into, as said before, a very time-consuming and, not least, resource consuming process. The use of measures and metrics to measure a project's health status is essential and effective, but it is more effective when those metrics are integrated into a system that performs the measurement processes in an automated fashion.

By employing such a system, Volvo's requirements management and project management processes will vastly improve. The time, and effort spent (i.e. man hours) today on manually analyzing a project's status can, with the help of an automated system, not only save time, but also increase the organisation's profitability in the long term. Having this in mind, it becomes clear that an automated project measuring system is in need.

5.2 Requirements

The main purpose of the application is to allow a project reponsible, such as a Project Manager(PM) or Chief Project Manager (CPM), to measure how well they are doing in a specific project, in other words, to consult the project's health status. The RMMP will assisst PMs and requirements managers, which work within Embedded Software(ESW) and Complete Offer(CO) divisions of this organization, to effectively measure the progress of a project. The RMMP will allow the PM or requirements manager to make effective decisions and take corrective actions when needed by analysing the data displayed to him/her. The data displayed to the managers are metrics which RMMP calculate by gathering diverse predefined base measures and combining them forming useful metrics.

A relevant goal when analyzing the system's requirements in terms of user needs is to make a relatively simple yet functional Graphical User Interface. The customer needs a straight forward and simple user interface with which he/she can work. Too complex application interfaces tend to be confusing and tiring.

One of the main customer requirements is the ability of exporting both the charts and data in ASCII format to, for example, a Microsoft Excel sheet document. The main reason why Excel is chosen as the designated document where data is exported is because Volvo uses Microsoft based products, so in order to make things easier, a document type which is widely supported within the organisation is selected. Unfortunately, there was only time to implement this function *locally*, which means the export function will only work from the computer or server where the application is installed. More about why this decision was made is discussed in the discussion section.

Another relevant requirement worth mentioning is the type of the application desired by the user. In the initial phase of the analysis, it was thought to construct the application as a stand-alone program, but as discussions took place with the customer/stakeholders the desire for making the application totally web-based increased. The reason behind this is straight forward because making the application as a stand-alone program would imply the use of more resources in terms of human effort when installing the application in every indicated machine. Also, stand-alone applications tend to be platform-dependent, while web-based can be platform-independent and reachable from any computer within Volvo's network.

In order to make it possible for the RMMP to perform the measurements, metrics are to be calculated. A metric is defined by the manager by defining base measures which are useful for a specific project. These base measures are then combined to create a powerful Metric. Then, after having calculated the necessary metrics, the RMMP will plot the data and display it to the manager in charts as the graphical presentation format.

5.3 Systems with similar functions

The RMMP is a tool that is new to the Volvo Corporation and independent of any other system currently in use. Furthermore, the RMMP will provide new and effective ways of managing and controlling a project. Having said this, there is, at the moment, no other automated tool within the organisation that can direcly compete with the RMMP in terms of functions, objectives and goals. However they do have other means of generating metrics but they are rather rudimentary methods compared to the RMMP since the metrics are manually created. By using this tool, PMs and RMs are given the opportunity to optimize their way of managing the project by finding errors early in the progress and correct them.

There are however, other tools that might offer functionalities to create metrics. One of those tools is DOORS, which allows the user to check the amount of requirements in the database. This method performs its function but it is also highly manual, which takes more time and effort than necessary.

Another tool, IBM Rational Jazz, which is a new technology platform for collaborative software delivery, offers also similar capabilities for creating metrics as the above mentioned methods. It does the job of creating a metric but still not as automated as the RMMP, where the RMMP not only does present current data but also historical data which is highly valuable in requirements management and project management.

5.4 Usability

Even though the RMMP is only a prototype, analysing the user interface requirements was given a high priority. This was done in order to achieve high usability in the system. The user desired a simple user interface so the final versions of the GUI were indeed simple and had the functionality required to perform the operations in a straight forward manner. The system is intended to be used by project responsibles such as a project manager or chief project managers that are not experts in computing or software. These users are engineers with relatively basic knowledge about software and computers.

Chapter 6

System Design

The present chapter describes the system environment where RMMP is running, development tools, Volvo network structure as well as class design in detail. In addition, system implementation and deployment are described at the end of this chapter.

6.1 System Design

RMMP is designed as a web based application since it is convenient to access the application via ubiquitous web browsers without installing software on client computers. Besides, it is also easy to maintain and update the application in a web based manner without distributing software.

6.1.1 System environment and development tools

- Operating system: It is designed that Windows Server 2003 is installed on the server side and users access the portal using Windows XP with Internet Explorer 6.0 on the client side.

- Database: The production servers use Oracle database where the requirements reside. However RMMP does not interact directly with the databases. Instead, it makes use of a number of CSV (Comma Separated Values) files extracted from the databases. As a prototype, RMMP interacts with the CSV files containing all necessary data for producing metrics. It is ensured that the running production databases cannot be corrupted by incorrect operations on RMMP.

- Development tools and language: Visual Studio 2008 from Microsoft is used as an IDE (Integrated Development Environment) for RMMP development. It is mainly because that ASP.NET 3.5, included in Visual Studio 2008 as a web application framework, provides a simplified development process. Furthermore, with powerful UI (User Interface) components integrated in Visual Studio, it is easy to illustrate metrics produced from statistical data graphically. Having extended supports provided by Microsoft is another reason of choosing Visual Studio for development. Multiple programming languages are supported by this IDE, such as Basic, Java, C++, C#, etc. As an up-to-date object-oriented language, C# is selected for programming.

- Web server: Considering the compatibility with ASP.NET applications, IIS (Internet Information Services) is employed as the web server for RMMP. IIS is the worldąŕs second most popular web server offering streamlined processing, configuration error control and low deployment cost. Meanwhile, IIS is also available on the server with Windows Server 2003.

6.1.2 Network Structure

RMMP is designed to be deployed on an internal web server connecting production servers and intranet as shown in Figure 6.1. Databases with requirements are running on production servers for a number of Volvo departments to carry out requirements evolution and management. Dimensions RM scripts are used for generating CSV files. Executing specific scripts on the web server, relevant data, for example, a set of requirements with a specified status, is extracted from the databases to one or more CSV files. The application accesses the files, calculates the data, produces desired metrics and eventually presents the metrics graphically to web browsers through intranet.

Requirements are stored in the databases with attributes, such as created time, status, etc. The scripts are created manually within Dimensions RM specifying the requirements to be extracted and the attributes to be retrieved for each requirement.

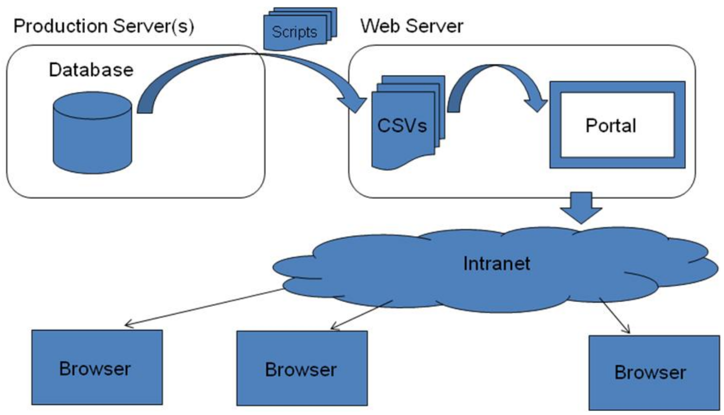

Figure 6.1: Volvo network structure

6.1.3 Application Architecture

A common-used modal with four layers is applied for the application architecture as shown in Figure 6.2. With the system divided into layers, the flexibility and reusability are improved.

- Presentation Layer: With the job of generating web pages containing dynamic content, this layer consists of various UI components presenting the output from business layer in a user-friendly way and delivering the input from clients to business layer. The most important UI component in this layer is Microsoft ASP.NET Chart Control which offers powerful functionalities for visualizing calculated results and a large number of configurable visualization features.

- Business Layer: This layer consists of services provided by the classes designed in the model. These services can be directly used as classes in the code or wrapped in a web service. The input from presentation layer is involved in the data processing from the services.

- Data Access Layer: Apart from the input, data access layer also delivers data that is retrieved from data layer to business layer for data processing. ADO.NET is included in this layer for connecting business and data layer. ADO.NET is a set of computer components that used to connect to data sources and retrieve, manipulate and update data. It is part of the base class library included with .NET framework.

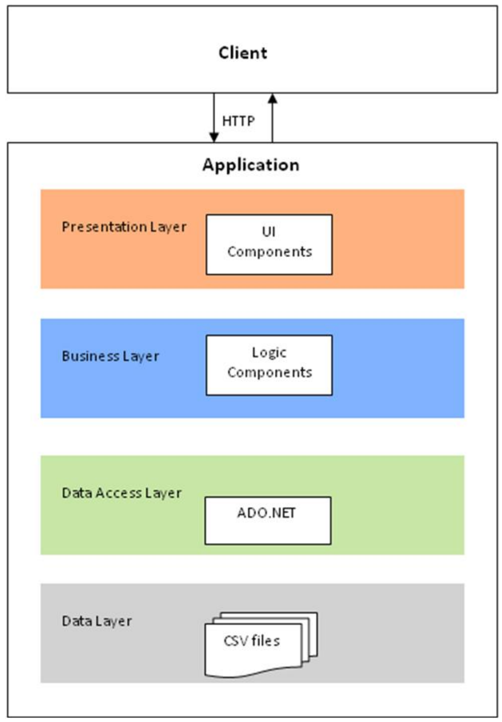

Figure 6.2: Application Architecture

- Data Layer: Data access layer accesses data from data sources lying in data layer. Data layer includes different data sources accordingly such as Microsoft SQL Server, Oracle database, CSV and XML. In this application architecture, CSV files act as the data source in this layer providing raw data for producing metrics.

6.1.4 Class Design

Classes offering key services all lie in business layer. Since the main functionality of RMMP is producing useful metrics based on requirements from production databases, the metric construct model described in Chapter 4 may be used as a basis for establishing the class design. In general, three base classes are designed corresponding to the three modules from the metric construct model and derived classes that inherit the base classes collaborate for metric production as shown in Figure 6.3.

- Base Measure:This is a base class for generating base measures. Each

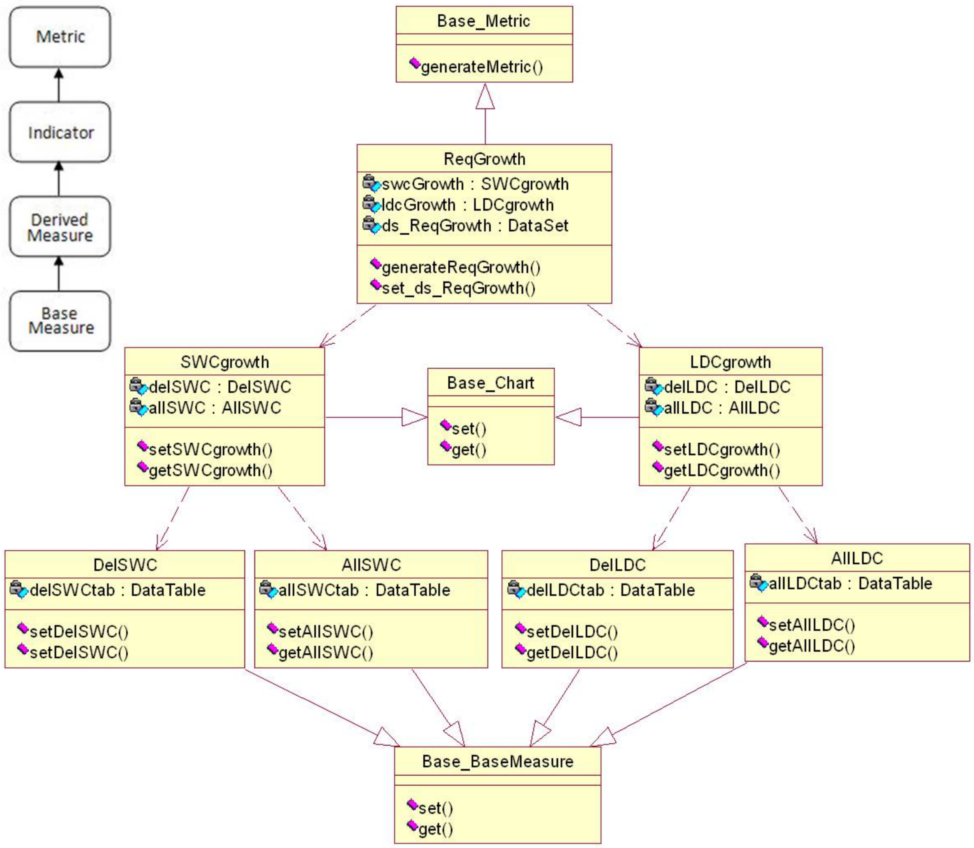

Figure 6.3: Class diagram and metric construct model

derived classes inheriting this base class generates an individual base measure. In Figure 6.3, class AllLDC and DelLDC both inherit class BaseMeasure and generate the number of all LDC requirements and deleted LDC requirements respectively by calculating the data delivered by ADO.NET from data access layer.

- Chart:This is a base class for generating charts, which corresponds to indicator in the metric construct model. A derived class inheriting class Chart calculates one or more base measures provided by derived classes inheriting BaseMeasure. During this procedure, derived measures (in the metric construct model) may be generated and used together with base measures for producing charts. In Figure 6.3, class LDCgrowth inheriting Chart calculates the total number of working LDC requirements by subtracting the number of deleted LDC requirements from the number of all LDC requirements since deleted requirements are still left in database with the status of deleted and cannot be counted as working requirements.

- Metric:This is a base class for generating metrics, which corresponds to metric in the metric construct model. A derived class inheriting class Metric integrates one or more charts provided by derived classes inheriting Chart with additional information (e.g. evaluation criteria). In Figure 6.3, class ReqGrowth integrates both LDC and SWC growth chart as one chart to be visualized by presentation layer.

6.2 System Implementation

This section describes how the system is implemented in four layers and application deployment on the server.

6.2.1 Implementation in presentation layer

With the well-developed framework provided by ASP.NET for web application development, presentation layer can be developed with powerful integrated UI components and interface with business layer in a simplified way. The mainly used UI component is Microsoft ASP.NET chart control which is responsible for visualizing all metrics of RMMP. In most cases, coding is only needed while binding the data from business layer to the chart controls for visualization and other work such as specifying the type, colour and interval of the chart controls is done by manual configuration in a graphical interface offered by Visual Studio. Each UI component corresponds to an object initiated automatically therefore the communication between objects from business layer and UI components from presentation layer becomes the communication between objects, which makes developers focus on logic development more.

6.2.2 Implementation in business layer

The main logic functionalities are all implemented in the objects in business layer. Based on the class design, each metric is produced by the collaboration of objects working at different levels. Objects of class BaseMeasure call the APIs of ADO.NET from data access layer in order to access data in CSV files and offer the retrieved data to Objects of class Charts for calculating. Algorithms are addressed in objects of class Charts doing comparison, calculation and combination with the data offered by objects of Class BaseMeasure. Once data for plotting a chart is produced, it is delivered to an object of class Metric which combines the delivered data with other information such as evaluation criteria to form the ultimate metric. The data of the ultimate metric is then bound to UI components by presentation layer.

6.2.3 Implementation in data access layer

As a bridge between business layer and data layer, data access layer applies ADO.NET where a number of manipulations on data source such as CSV are encapsulated. ADO.NET enables developers to connect to the CSV file by specifying the path and manipulate the data using SQL language from objects in business layer. Retrieved data is accessed as an object by other business objects, which meets the need of object-oriental development.

6.2.4 Implementation in data layer

CSV files obtained from Dimensions RM by running predefined scripts are stored in the file system on the server. Scheduled batch files which call Dimensions RM to run scripts update the CSV files according to the required interval.

6.2.5 Deployment

The application has been deployed on IIS after development and running currently as Volvo required. Since web applications do not require installing software on client computers, internal users are able to access RMMP with web browsers (Internet Explorer 6.0 currently) to measure a test project.

Chapter 7

Discussion and Conclusion

7.1 Discussion

Making research about Requirements Management and Requirements Engineering gave us a broader picture and deeper understanding of how requirements in product development projects are engineered, supported, maintained and realised. It has become clear to us that in requirements engineering there are no requirements evaluation processes, it is simple and straight forward – requirements engineering gives life and form to the concepts proposed by the stakeholders. Requirements management is an indispensable requirements management process for Volvo and its Business Areas. This is because it increases the value of the requirements.

Documentation in Requirements Management is extremely important, specially when it comes to changes made in a requirement. If we consult the work flow proposed to Volvo, we can see it contains several change requests that can be performed if necessary. The point with documenting change requests in this particular work flow, or generally in requirements management for that matter, is to know if a certain requirement has already been changed in an earlier phase of the project. If we do not document the changes, even if the requirement was not previously implemented, in other words rejected, the organisation would then have to spend additional man-hours and economic resources on evaluating the same requirement all over again. Most importantly, if no documentation would have been carried out, the use of metrics such as Requirements Volatility would have been useless since the requirements change documentation is the basis for performing measurement on how volatile a requirement *currently* is and *has* been.

Learning the requirements management tools used within Volvo, such as Dimensions RM, was a wise thing to do because that gave us great insight of how the requirements are actually managed and documented. This tool also helped us to understand even further how the requirements were linked to each other. The usage of this tool made our work possible because it provided the information needed in order to implement the metrics in our application. For instance, in order to implement the metric Requirements Acceptance for Complete Offer, we had to extract, from Dimensions RM, all the YELLOW, GREEN and RED requirements for each role involved in the reviews of the requirements. Then only minor calculations were performed by the RMMP in order to visualize the requirement acceptance. Similarly, Dimensions RM also provided, in a swiftly manner, data such number of deleted and number changed requirements for calculating the Requirements Volatility metric. It did however not provide the number of *NEW* requirements which forced us to calculate this manually, hardcoding it in the RMMP. This is a shortcoming that could be addressed by setting a new status to each requirement in Dimensions RM, where the flag *New*, as in new requirement, is set. Currently, the engineers at Embedded Software do not make use of such status, but it is clear that introducing this new status or similar, would boost their working processes performance and also give them the ability to create additional useful metrics, hence improving the control and overview of their projects.

The phase of creating metrics did indeed enrich our knowledge about the usefulness and importance of these. We did however find the process of metric creation a bit overwhelming as it involved being totally familiar with the subjects (i.e. metrics, measures, measurement processes and Requirements Management) as well as with Volvo processes. Nevertheless, working with the Volvo engineers helped to maintain a relatively high level of creativity and participation during the workshops. We also realise that it is not an easy task to apply RM metrics theory into practice. The reason is simple, engineers might not be used to employ metrics to measure a project's health status. This can make it difficult to introduce metrics in a field or industry where no one employs them. Consequently, the engineers new to measures and metric might have a tough time adapting his/her way of work in order to make the best use of these metrics. Furthermore, updating an attribute for each requirement in the database takes 30 seconds $\times$ 5000 requirements = 5 manhours. It therefore becomes clear that it is difficult to create a business-case on adding an attribute just to acquire a metric. Another scenario could rise when introducing an automated project measuring system such as the RMMP, because this might imply the introduction of new workflows, new requirements attributes, or simply modifying the existing data in order to make things work with the new system and its metrics. As discussed earlier, engineers working with an existing working process might find themselves reluctant to such a system due to the significant additional work it would take to adapt to this new system. It is understandable that such a change will be difficult to perform over a day. However, when the moment comes where such a system is being used in real time projects and the engineers have adapted to an acceptable level to the system and all that it implies, the project processes will progress in a swiftly and controlled manner.

Having a background in Software Engineering helped us in the analysis, design and implementation phases of the software system. We are confident in the fact that we achieved to construct a satisfactory functional system. There are, as always in a software development project, roadblocks which might make life difficult. One such roadblock was the compilation of relevant data from the databases to visualize the Requirements Acceptance metric for ESW. An example of such data is the name of the baselines as well as the dates when they were created. This was quite challenging since this data was nowhere to be found using Dimensions RM functionalities like the data export function nor its web services. Fortunately, we got access granted to the Oracle database where, at last, the data needed was found, and exported to a CSV file.

Other problematic situation encountered was when implementing the export-function. This function, as explained in previous sections, exports the charts as JPEG illustrations as well as raw data in ASCII format to a Microsoft Excel file. The initial version of this export function worked *locally*, i.e. on the same computer where the prototype was running. This worked fine since no scripts were needed the be sent from through the network . However, this situation would change as soon as we tried to implement a global version. Implementing a global version implied having java scripts being sent over the Volvo network, from the server to a client browser. Volvo, as any other well known international company, are fully aware of the risks scripts and other malicious code imply. Thus, there are heavy regulations and rules which impeded us from finishing the global implementation of the export function in time. From a network security view, it is well known that sending scripts, no matter the language, is a potential threat to a network and its nodes. Furthermore, we would have needed additional time to analyse the risks and understand the threats in order to implement a *robust* export function that could cause no harm to its environment.

Time, was a factor that limited our work in a few ways. The previous paragraph described one example but it also limited the implementation of the RMMP when it comes to visualizing all the metrics created for Complete Offer such as Requirements Growth and Requirements Volatility. Time also limited the variation in the visualization of the Acceptance metric. For example, strongly we believe, that using a line chart would better describe the acceptance rate over time for each of the participating roles.

7.2 Conclusion

There are two main results we would like to talk about briefly. The first is the creation of useful metrics and the second is the ability of the RMMP to successfully visualize the metrics.

One of the main long-term goals for this thesis project was to create metrics that are useful to Volvo's project development processes. Research shown in this thesis report has proven that developing metrics which meet the organisation's goals thrivingly allows the project managers to measure and control a project in their favour. Correspondingly, the effort, creativity and work put during the metric creation phase gave fruitful results, not only because we kept a high interest and devotion into the matter, but mainly because of the valuable input and support we received from the engineers at Volvo. The input received was valuable because it came directly from people that d fully understand the Volvo processes as well as experience with reference to *what* is needed to improve the current project processes. Having this said, we are certain that the metrics created during the thesis project will have a positive impact on Volvo's project processes if they are implemented.

In the final moments of the project, we managed to implement all the metrics discussed for ESW and one metric for CO into the RMMP. We are pleased with the RMMP capabilities of displaying the metrics as it was planned. The engineers at Volvo will have the opportunity to try out the RMMP as they please during a long period. This will give them the opportunity to get familiar with an automated measuring system for their projects which in turn will push the them to think more about how useful metrics and measurement processes can really be. Furthermore, enablers and and hinders were also identified. For instance, for the RMMP to perform successfully and display the correct data, the organisation enables swift access to data in the database which is easily elicited and displayed to the user in form of charts. Another enabling feature is that the prototype is completely platform-independent, which means the user can have direct access to the application from any computer within Volvo's internal network while at the same time, as discussed above, the application has easy access to the database.

A hinder that acts as an obstacle for the RMMP to show the correct results is that updating requirements attributes takes time because they need to be balanced according to the benefits. This might cause the system to display data that is not yet updated and allow the project manager/requirements manager to make decisions based on this. In addition to the hinder previously explained, user experience hinders could also prevent correct decision making. For instance, analyzing the requirements growth curve can be a complex procedure if you are not familiar with the stongly related project processes and phases. Consequently, a user could misunderstand the growth curve which could lead the user to think that the requirements growth is growing too fast when in reality it is an indication that the project has reached a review phase.

In addition to the already implemented functions and metrics, the RMMP could support the ability of having the user specify and create custom metrics by choosing from, for example, a long list of available base-measures. Likewise, the user could also be able to create custom base-measures to combine them later into metrics. The ideal way of doing this would be to design and implement a type of wizard that would, with simplicity, allow the user to perform the previously mentioned custom metrics operations.

Another feature that would be useful for the RMMP to provide, is the ability to read input from other widely supported formats such as Microsoft Office products: Word and the like. This would higher the usability of the system and broaden its capabilities of service.

Finally but not least, we would recommend to implement the metrics that are suggested in the section of "Metrics for Future use". Having these metrics implemented could amplify the effectiveness and usability of the RMMP and give the engineers at Volvo further capabilities of measuring and controlling the project's health status.

Bibliography

[1] Mohammed Javeed Ali. Metrics for requirements engineering. Master's thesis, Department of Computing Science, Umeå University, June 2006.

[2] Patric Antony, Jennifer Dunn, Dr. William Farr, Dr. Donna Rhodes, Garry Roedler, Cathy Tilton, and E. Richard Widmann. *Systems Engineering Measurement Primer*. INCOSE, Seattle, WA, USA, 1998.

[3] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. The goal question metric approach. Technical report, Department of Computer Science, College Park, Marlyand and Kaiserslautern, Germany, 1994.

[4] B.W. Boehm and Papaccio. *Understanding and Controlling Software Costs*. IEEE Transactions on Software Engineering, 1988.

[5] Finkelstein A. Gotel, O. An analysis of the requirements traceability problem. Technical report, Department of Computing, London, United Kingdom, 1993.

[6] Colin Hood, Stefan Fichtinger, Urte Pautz, and Simon Wiedermann. *Requirements Management: The Interface Between Requirements Development and All Other Systems Engineering Processes*. Springer-Verlag Berlin Heidelberg, Oberhaching, Germany, 2008.

[7] Elizabeth Hull, Kenneth Jackson, and Jeremy Dick. *Requirements Engineering*. Springer London Berlin Heidelberg, USA, 2005.

[8] Alexander Kossiakoff and William N. Sweet. *Systems Engineering*. John Wiley and Sons, Inc., Hoboken, New Jersey, USA, 2003.

[9] Annabella Loconsole. *Definition and validation of requirements management measures*. PhD thesis, Umeå University, January 2008.

[10] John McGarry, David Card, Cheryl Jones, Beth Layman, Elizabeth Clark, Joseph Dean, and Fred Hall. *Practical Software Measurement: Objective Information for Decision Makers.* Pearson Education, Inc., 75 Arlington Street, Suite 300 Boston, USA, 2002.

[11] SERENA. *SERENA DIMENSIONS RM 10.1.4 User's Guide.* Serena Software, Inc., 1900 Seaport Blvd., Redwood City, USA, 2008.

[12] SERENA Software. `http://www.serena.com/Products/dimensions-rm-requirements-management/index.html`.

[13] Ian Sommerville. *Software Engineering.* Pearson Education Limited, seventh edition, 2004.

[14] Ann Wilbut, Gayle Towers, Tom Sherman, Dan Yusawaka, and Sue Shre ve. *Metrics Guidebook for Intergrated Systems and Product Development.* INCOSE, Seattle, WA, USA, 1995.

Appendix A

Software Requirements Specification

Requirements Management Metrics Portal
Software Requirements Specification

Alexander Jonsson
Kebing Hou

October 8, 2009

A.1 Introduction

A.1.1 Purpose

The purpose of this Software Requirements Specification document is to describe the requirements specification for a Requirements Management Metrics Portal.

The target audience for this document is intended to be the prospective delevopers as well as System- and Bussines Analysts belonging to the client organization.

A.1.2 Scope

The software product to be produced is a Requirements Management Metrics Portal, which we will refer to as RMMP throughout this document.

The RMMP will assist project managers and requirements managers, which work within Embedded Software and Complete Offer divisions of this organization, to effectively measure the progress of requirement rekated work. RMMP will allow the Project Manager or Requirements Manager to make effective decisions and take corrective actions when needed by analysing the data displayed to him/her. The data displayed to the managers are metrics which RMMP calculate by gathering diverse base measures and combining them forming useful metrics for this organization's projects. Furthermore, the manager should be able to export the data into a file to have valuable information saved and collected separately.

RMMP can be used in any environment where managers need to have a highly acceptable overview and control of a project's progress within an organization. RMMP's main objective is to measure, with the help of useful metrics, the progress of a project. Moreover, the goal of RMMP is to higher the effectiveness when it comes to project management. By having increased control of the project progress a Project Manager can save time and resources by making useful predictions, based on data provided by RMMP, on how well the team is doing in a project and what needs to be done in order to improve the process.

A.1.3 Definitions, acronyms, and abbreviations

- Activity: this defines the various processes within Software Development Analysis and Design. For instance such activities could be: analysis, design, implementation, testing, debugging and so forth.
- Artifact: physical representation of the result of an activity. A physical representation could be: SRS, Use Case model, Domain model, activity diagrams, UML diagrams and so forth.
- SRS: Software Requirement Specification.
- RMMP: Requirements Management Metric Portal.
- IIS: Internet Information Services.
- PM: Project Manager.
- RM: Requirements Manager.
- LDC: Logical Design Component
- SWC: Software Design Component

A.1.4 References

Hull, E., Jackson, K. and Dick, J. (2005) *Requirements Engineering*, 2^{nd} edition, London: Springer

A.1.5 Overview

From now on, the document contains an Overal Description of RMMP, a description of the Specific Requirements related to RMMP as well as a Classification of the Functional Requirements.

A.2 Overall Description

A.2.1 Product Perspective

The RMMP is a tool new to the client organization and independent of any other system in use by the organization. Furthermore, the RMMP will provide additional functionalities as well as new and effecitve ways of managing and controlling a project. Having said this, there is, at the moment, no other automated tool within the organisation that can directly compete with the RMMP in terms of functions, objectives and goals. However, they do have other means of generating metrics. By using this tool, PMs and RMs are given the opportunity to optimize their way of managing the project by finding errors early in the progress and correct them.

System Interfaces

The RMMP is a web-based application which is integrated to the organisation's intranet. The system consists of the following components:

- Client Module

Interfaces

The Client Module must provide a graphical user interface that is available through the Microsoft Internet Explorer browser.

Hardware Interfaces

The above described components must be capable to run on personal computers and the like.

Software Interfaces

The Client module should be web based and should run within Microsoft Internet Explorer.

Communication Interfaces

The Client Module , which is deployed at a web server, should communicate through the network over a TCP/IP connection.

Memory Constraints

The Client Module must operate within 64 MB of memory.

Operations

The Client Module should be easy and simple to use for the PMs and/or RMs. No higher technical skills should be required for the user in order to operate the tool properly.

Maintainability

System code documentation should be supported in order to make a follow-up of the work in a later phase.

Site Adaption Requirements

No site adaptation requirements required.

A.2.2 Product Functions

The RMMP will enable the PM or RM to effectively measure the current progress in a project. It will allow the user to have a relatively clear overview of an ongoing project. This will assisst the managers to take corrective actions in time and to possibly predict scenarios which may have a negative impact on a project.

In order to make it possible for the RMMP to perform the measurements, Metrics are to be calculated. A Metric is a measure of quality and can be used to improve the product quality and productivity. A Metric is defined by the manager by creating and/or defining base measures which are useful for a specific organisation or project. These base measures are then combined to create a powerful Metric. Metrics are often used by the managers as comparisons, such as planned versus actual measures.

The RMMP, with the help of these customized metrics, can describe the current situation or predict the future situation with respect to an issue in a project. Early error identification is vital in a project because it saves resources to detect errors in an early phase rather than in a later phase.

Moreover, the RMMP, after having calculated the necessary metrics, will plot the data and display it to the manager as charts. In addition to this visualization function, the RMMP will also allow the manager to export the data displayed into a file. The file type will be a predetermined file type that is supported within the office environment within the organisation.

A.2.3 User Characteristics

The intended users of this software product are Project Managers and/or Requirements Managers. The users should be highly familiar with Requirements Management and have a mature perception of the significance and use of Metrics for measuring a project's progress.

A.2.4 Constraints

The RMMP should be independent of any other software which the organisation has to pay for in order to make full use of the RMMP.

A.2.5 Assumptions and Dependencies

A specific support for ASP .NET based applications should be available on the hardware intended to host the software application.

Internet browsers which are fully compatible with the software application should be available through the users personal computers stationed within the organisation's facilities.

A.2.6 Apportioning of Requirements

No apportioning of requirements is required.

A.3 Specific Functional Requirements

A.3.1 The RMMP shall run within Internet Explorer 6.0 or higher.

A.3.2 When the user accesses the RMMP, the RMMP shall display a dropdown list with all the projects.

A.3.3 The RMMP shall provide a list of projects for selection.

A.3.4 When the user selects a project, the RMMP shall display a predefined list of metrics.

If the user selects the Acceptance metric, the RMMP shall display three column charts

where each chart:

- shall present the number of reviewed and non-reviewed LDC requirements per week.
- shall present the number of accepted and non-accepted LDC requirements per week.
- shall present the number of reviewed and non-reviewed SWC requirements per week.
- shall display, whenever a week for a column for reviewed requirements in the chart is selected, a pie chart showing percentage of reviewed requirements.
- shall display, whenever a week for a column for acceptance of requirements in the chart is selected, a pie chart showing percentage of accepted requirements.
- shall display, whenever a week for a column for acceptance of requirements in the chart is selected, a pie chart showing percentage of accepted requirements.

If the user selects the Growth metric, the RMMP shall display a line chart where each chart:

- shall display the total number of LDC and SWC requirements per week.

In addition to the initial chart, the RMMP shall:

- display a list of all the LDC baselines with a Key Word field with its corresponding display button
- display a list of all the SWC baselines with a Key Word field with its corresponding display button.
    1. One or more baselines shall be selectable.
    2. The Keyword field shall accept input of the type string.
- display a line chart presenting the total number of (LDC or SWC) requirements for eah baseline.

If the user selects the Volatility metric, the RMMP shall

- display a list of all the LDC baselines with a Key Word field and its corresponding display button.
- a list containing all the SWC software released baselines with a Key Word field and its corresponding display button.
    1. One or more baselines shall be selectable.
    2. The Keyword field shall accept input of the type string.
- display a line chart presenting the number of new, changed and deleted (LDC or SWC) requirements between two consecutive baselines.

If the user selects the Test Coverage metric, the RMMP shall

- display a column chart that presents the number of SW-C requirements connected to test cases per week.
- shall display, whenever a week for a column for reviewed requirements in the chart is selected, a pie chart showing percentage of reviewed SWC requirements connected to test cases.

If the user selects the Verification metric, the RMMP shall

- display a line chart which presents the relative number of each verification method for SWC requirements per week.

If the user selects the Addressed Requirements metric, the RMMP shall

- display a column chart which presents the number of LDC requirements not addressed to SWC requirements per week.
- shall display, whenever a week for a column for reviewed requirements in the chart is selected, a pie chart showing percentage of LDC requirements requirements not addressed to SWC requirements.

A.3.5 The RMMP shall provide one option for exporting data to a file

A.3.6 The RMMP shall acquire input data from a CSV file

The RMMP shall read the CSV file every night at 23:59 hours

Appendix B

Metric Attribute Tables

The presentation format of the metrics shown in each table were made for illustrative purposes only and differ from the actual screenshots taken from illustrations provided by the prototype.

Requirements Growth	
Goals	• Precise measurement of requirements size • Allow the PM to identify errors and prevent them • Allow PM to improve project processes
Customers	• CO Project Managers • CO Requirements Managers
Intervals of measurement	• Weekly - per project phase.
Data or measurements used	• Total number of requirements.

Data source	• Project RAW Data available in the databases.
Interpretation of results	• The growth rate is important to analyse to improve the pr ocess by observing the amount of new requirements introduced compared to historical data.
Strengths	• Objective measurement of the number of requirements in the database. • Indication of the amount of new requirements created at any given time point.
Weaknesses	• To analyze the growth curve kan be tricky if the analyzer is not familiar with the working processes and project phases. A misunderstanding could lead the user to think that the requirements growth is growing too fast when in reality it is an indication that the project has reached a *review* phase.
Prerequisites for measurement	• Raw data to be measured should be made available in the designated database.
Presentation format	• A chart representing the metric is shown in B.1

Table B.1: Complete Offer Requirements Growth

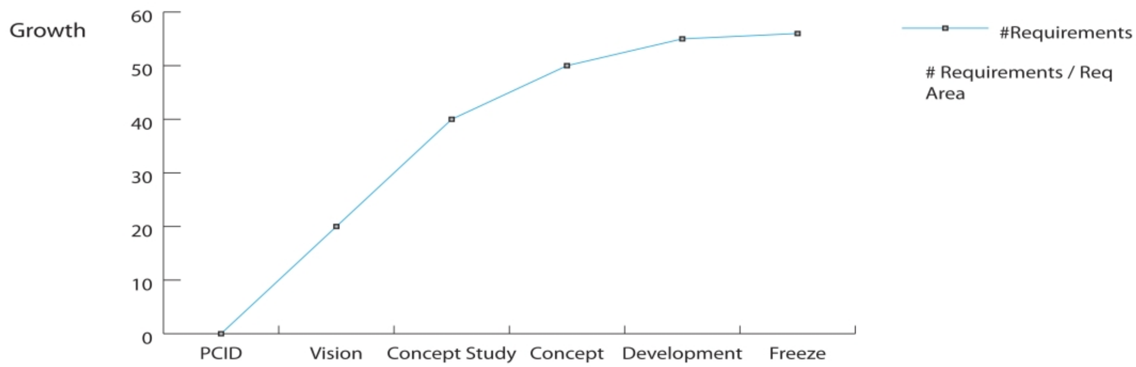

Figure B.1: Presentation format of Requirements Growth for CO

Requirements Acceptance	
Goals	• Precise indication of the level of acceptance at a current date point for each role involved. • Indication of how well written requirements are for the project. If too many requirements are RED or YELLOW then it indicates that requirements need to be reviewed/rewritten.
Customers	• CO Project Managers • CO Requirements Managers
Intervals of measurement	• Weekly

Data or measurements used	• Total number of RED marked requirements. • Total number of GREEN requirements. • Total number of YELLOW requirements. • Total number of RED, GREEN and YELLOW marked requirements.
Data source	• Project RAW Data available in the databases.
Interpretation of results	• The more RED and YELLOW marked requirements found in database, the more review those requirements need. • The more GREEN marked requirements, the faster one can proceed to the next phase in a project. • The amount of GREEN marked requirements are the basis for deciding how well written and accepted requirements are. The higher the acceptance, the faster one moves on to the next phase.
Strengths	• Objective measurement of the acceptance level of requirements.

Weaknesses	• The requirements might not always be updated, since they are usually updated a few days after a review has taken place. Consequently, the metric will not measure up-to date requirements.
Prerequisites for measurement	• All requirements in the database should be made available and marked GREEN, YELLOW or RED.
Presentation format	• A chart representing the metric is shown in B.2

Table B.2: Complete Offer Requirements Acceptance

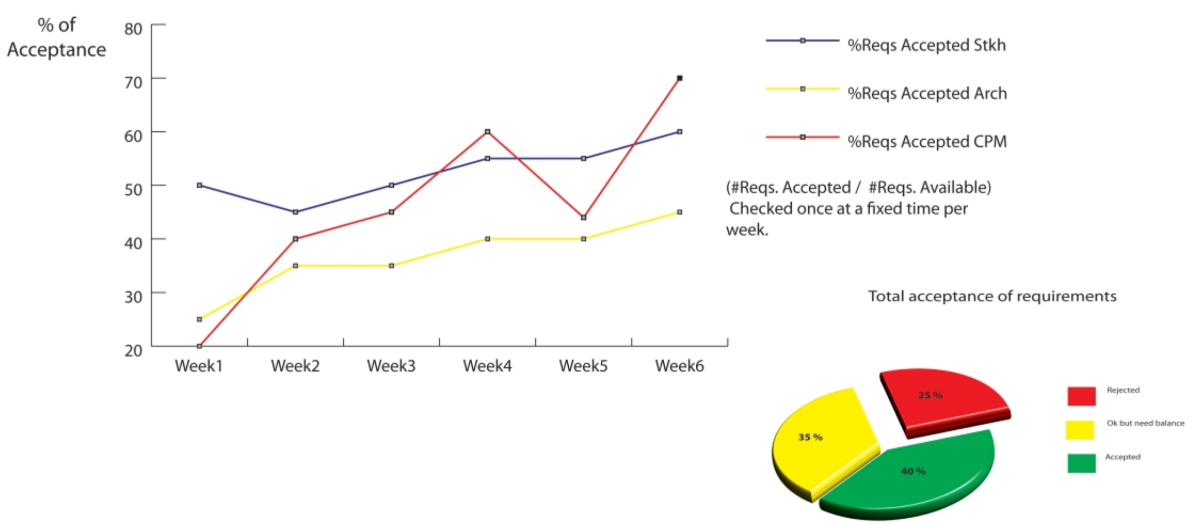

Figure B.2: Presentation format of Requirements Acceptance for CO

Requirements Volatility	
Goals	• Precise measurement of requirements change rate. • Allow the detection of exceeded data collection which suggests a deep analysis of the project is needed. • Gives place for further analysis in order to understand the reasons why the amount of excessive data and to take corrective actions in time.
Customers	• CO Project Managers • CO Requirements Managers
Intervals of measurement	• Weekly.
Data or measurements used	• Number of new requirements. • Number of deleted requirements. • Number of modified requiremenets.
Data source	• Requirements data from Requirements Managers responsible.

Interpretation of results	• Low volatility mihg mean that the requirement engineering process is being stalled. • In early phases, the volatility is expected to be high. • If the data is far more than allowed, then a deep analysis and corrective actions are needed to address the issues at hand.
Strengths	• Objective measurement of the number of requirements being changed per project phase, entity or time point. • Indication of excessive data crossing boundaries set in an early phase of the project.
Weaknesses	• Information might not always be updated in the database so outdated requirements may be measured leading to unprecise volatility rates
Prerequisites for measurement	• Requirements of the project are made available by the Requirements Manager responsible.
Presentation format	• A chart representing the metric is shown in B.3

Table B.3: Complete Offer Requirements Volatility

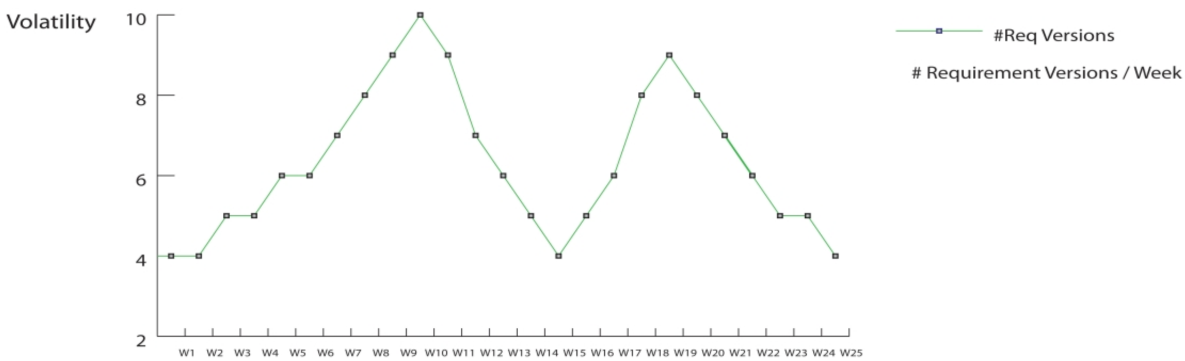

Figure B.3: Presentation format of Requirements Volatility for CO

Requirements Growth	
Goals	• Estimate the size of the system requirements to arrange appropriate resource allocation • Determine the correctness of the ratio between LDC and SWC requirements size
Customers	• ESW Project Managers • ESW Requirements Managers
Intervals of measurement	• Optional for users - weekly, monthly
Data or measurements used	• Total number of LDC and SWC requirements

Data source	• Project RAW Data available in the databases
Interpretation of results	• Used to understand the system progress in terms of size • Review of this metric can help evaluate the ratio between the amount of LDC and SWC requirements
Strengths	• This metric requires no extra work flow but all the requirements available in the databases
Weaknesses	• It takes more time to produce the metric as the requirement size is growing
Prerequisites for measurement	• Raw data to be measured should be made available in the designated database
Presentation format	• A chart representing the metric is shown in B.4

Table B.4: Embedded Software Requirements Growth

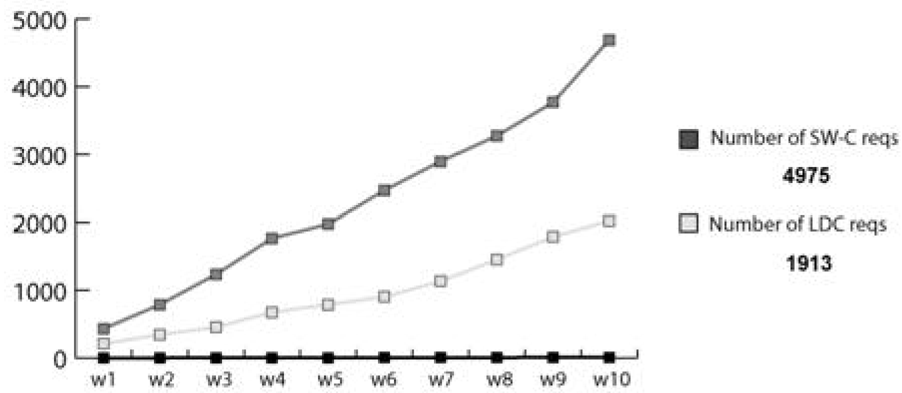

Figure B.4: Presentation format of Requirements Growth

Requirements Volatility	
Goals	• Evaluate the stability of LDC requirements from baseline to baseline • Estimate risks in time during development • Assist with resource allocation
Customers	• ESW Project Managers • ESW Requirements Managers
Intervals of measurement	• Every baseline
Data or measurements used	• The number of new, changed, unchanged and deleted LDC requirements

Data source	• Project RAW Data available in the databases
Interpretation of results	• Corrective activities should be taken if an abnormal change rate arises
Strengths	• This metric requires no extra work flow but regular requirements composition process
Weaknesses	• The responsible source for the changes are yet provided by this metric
Prerequisites for measurement	• Raw data to be measured should be made available in the designated database • Requirements have been baselined in the database
Presentation format	• A chart representing the metric is shown in B.5

Table B.5: Embedded Software Requirements Volatility

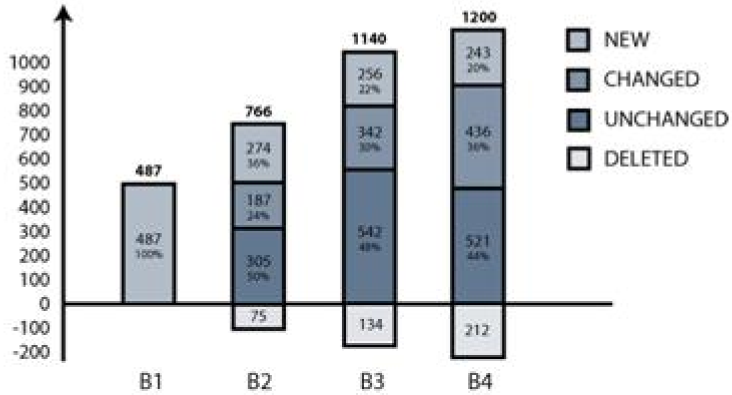

Figure B.5: Presentation format of Requirements Volatility

Requirements Acceptance	
Goals	• Evaluate the acceptance progress of requirements elicitation
Customers	• ESW Project Managers • ESW Requirements Managers
Intervals of measurement	• Weekly
Data or measurements used	• The number of accepted and non-accepted LDC requirements • The number of new, changed, approved and rejected SWC requirements

Data source	• Project RAW Data available in the databases
Interpretation of results	• The proportion of accepted or approved requirements determines the acceptance degree in requirements elicitation • Changed requirements are the requirements that have been reviewed and changed due to some reason and yet approved
Strengths	• This metric can be part of gate examination criteria in GDP described in Chapter 1
Weaknesses	• Each status needs to be updated for each requirements and this process takes time
Prerequisites for measurement	• Raw data to be measured should be made available in the designated database • Engineers manipulate the requirements following a specific work flow
Presentation format	• Two charts representing the metric are shown in B.6

Table B.6: Embedded Software Requirements Acceptance

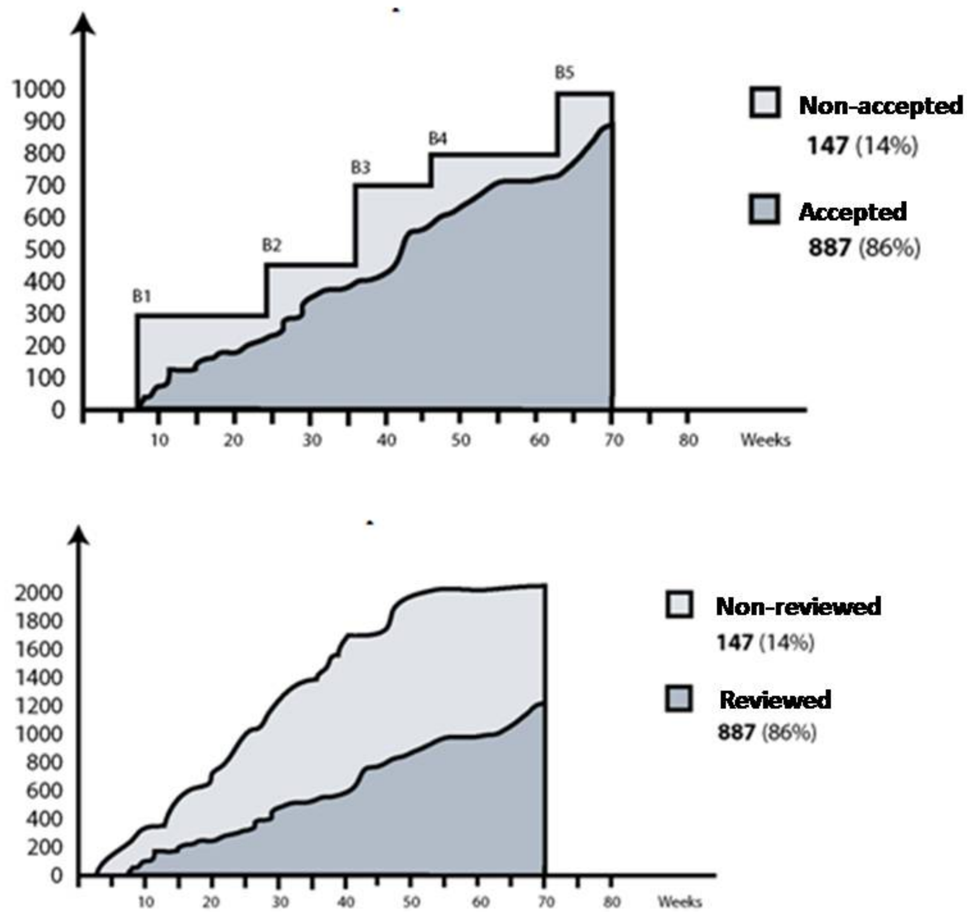

Figure B.6: Presentation format of Requirements Acceptance

Requirements Verification	
Goals	• Help understand the effort to be put into verification • Assist with resource allocation

Customers	• ESW Project Managers • ESW Requirements Managers
Intervals of measurement	• None
Data or measurements used	• The number of relative number of each verification method for SWC requirements
Data source	• Project RAW Data available in the databases
Interpretation of results	• Resource should be allocated according to the verification method proportion regarding all requirements
Strengths	• This metric can be produced quickly
Weaknesses	• When new verification methods are added into the system, more work need to be done in the process for producing this metric

Prerequisites for measurement	• Raw data to be measured should be made available in the designated database • Verification methods have been specified for requirements
Presentation format	• A chart representing the metric is shown in B.7

Table B.7: Embedded Software Requirements Verification

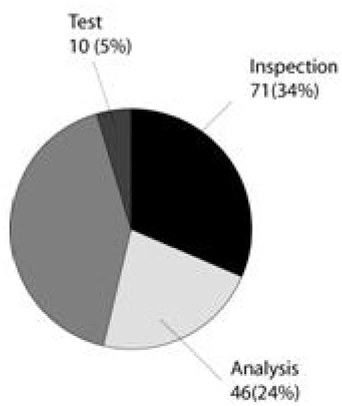

Figure B.7: Presentation format of Requirements Verification

Addressed Requirements	
Goals	• Evaluate LDC requirements progress in terms of linking
Customers	• ESW Project Managers • ESW Requirements Managers
Intervals of measurement	• None
Data or measurements used	• The number of LDC requirements that have been addressed (linked to SWC requirements in the database)
Data source	• Project RAW Data available in the databases
Interpretation of results	• A low rate implies more effort should be put
Strengths	• This metric can be produced quickly
Weaknesses	• This metric counts the number of requirements that has one link, which does not mean that it is fully addressed

Prerequisites for measurement	• Raw data to be measured should be made available in the designated database • Links are built up among requirements
Presentation format	• A chart representing the metric is shown in B.8

Table B.8: Embedded Software Addressed Requirements

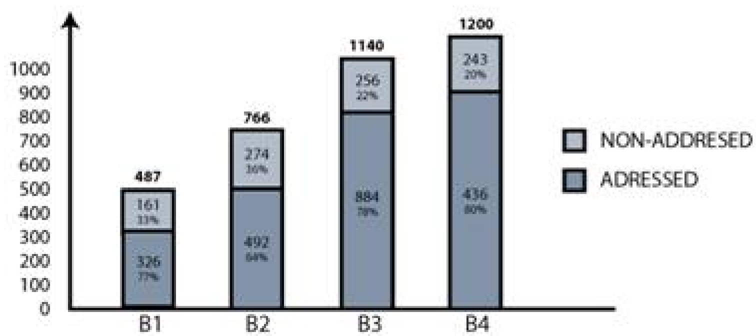

Figure B.8: Presentation format of Addressed Requirements

Appendix C

RMMP User Interfaces

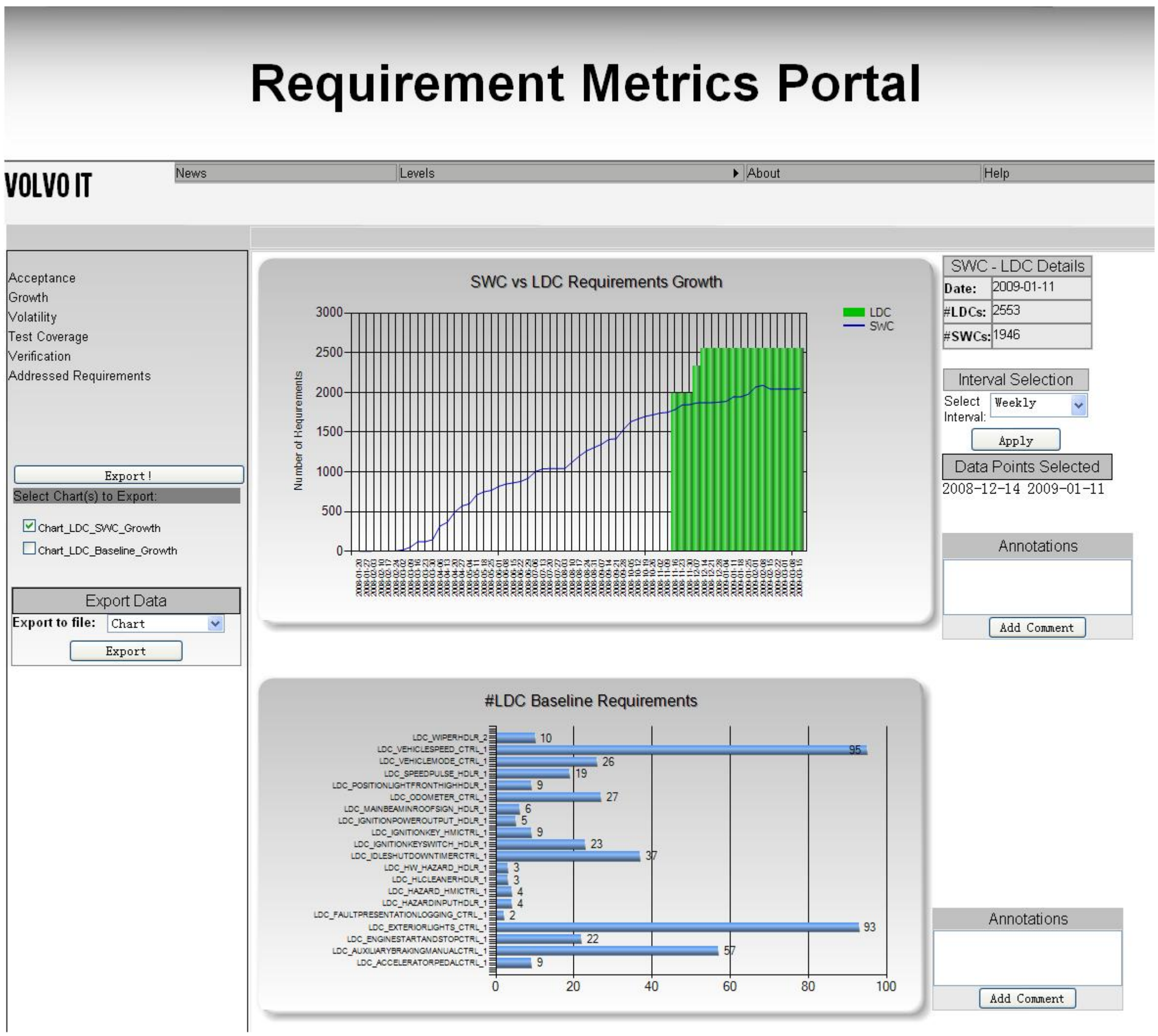

Figure C.1: Interface of Requirements Growth

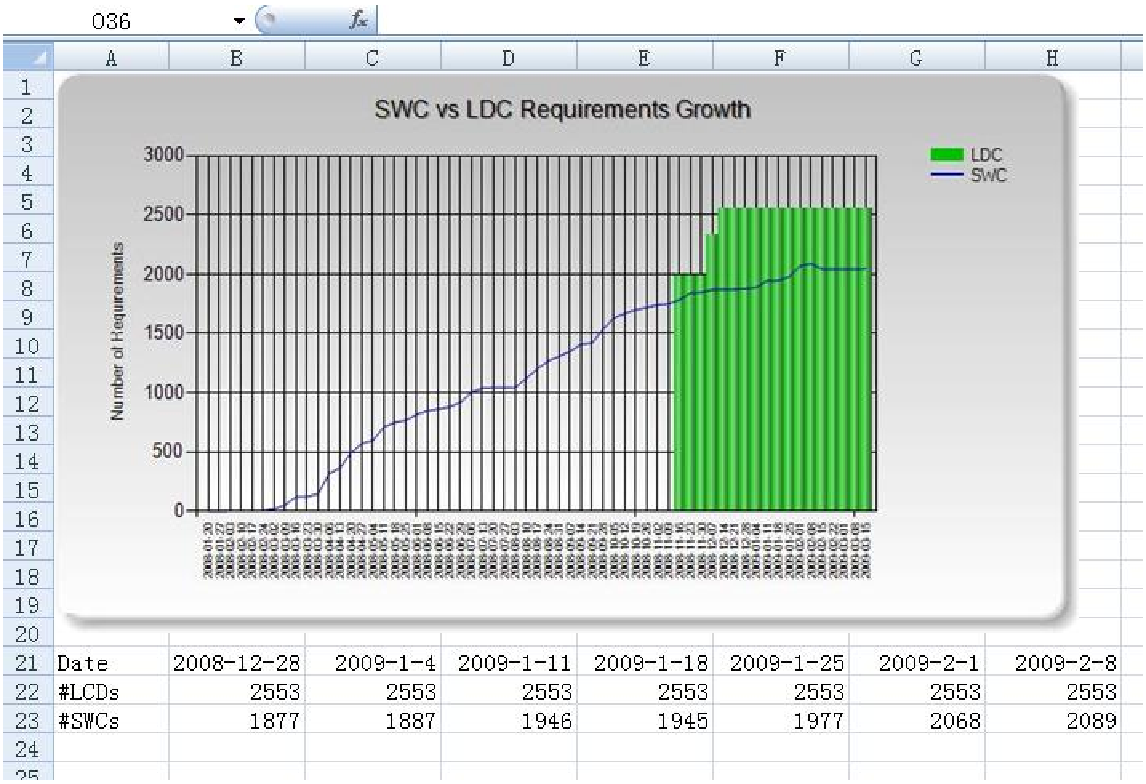

Figure C.2: Exported chart with raw data

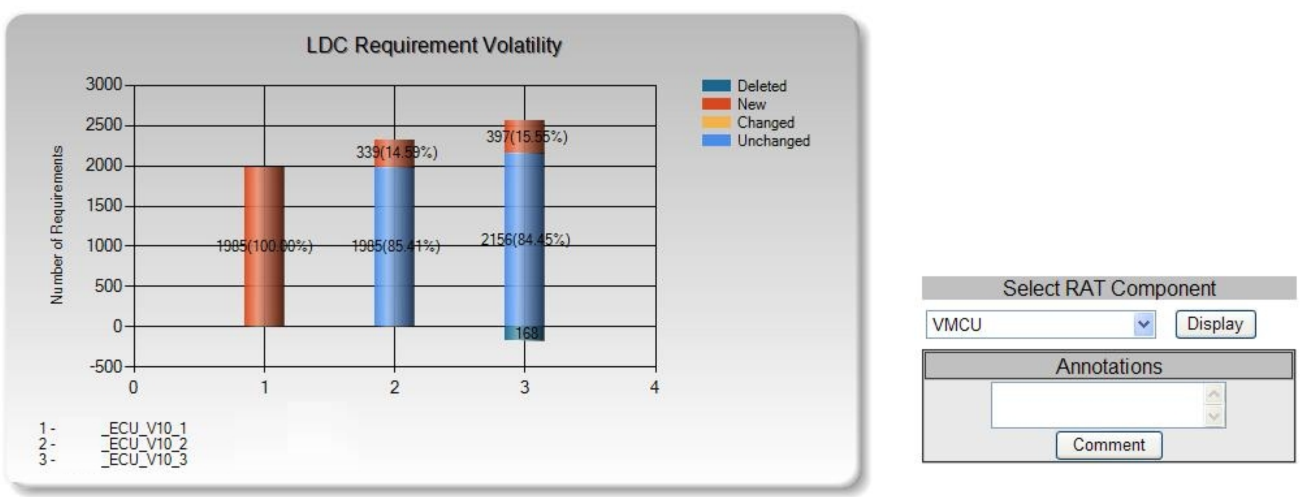

Figure C.3: Interface of Requirements Volatility

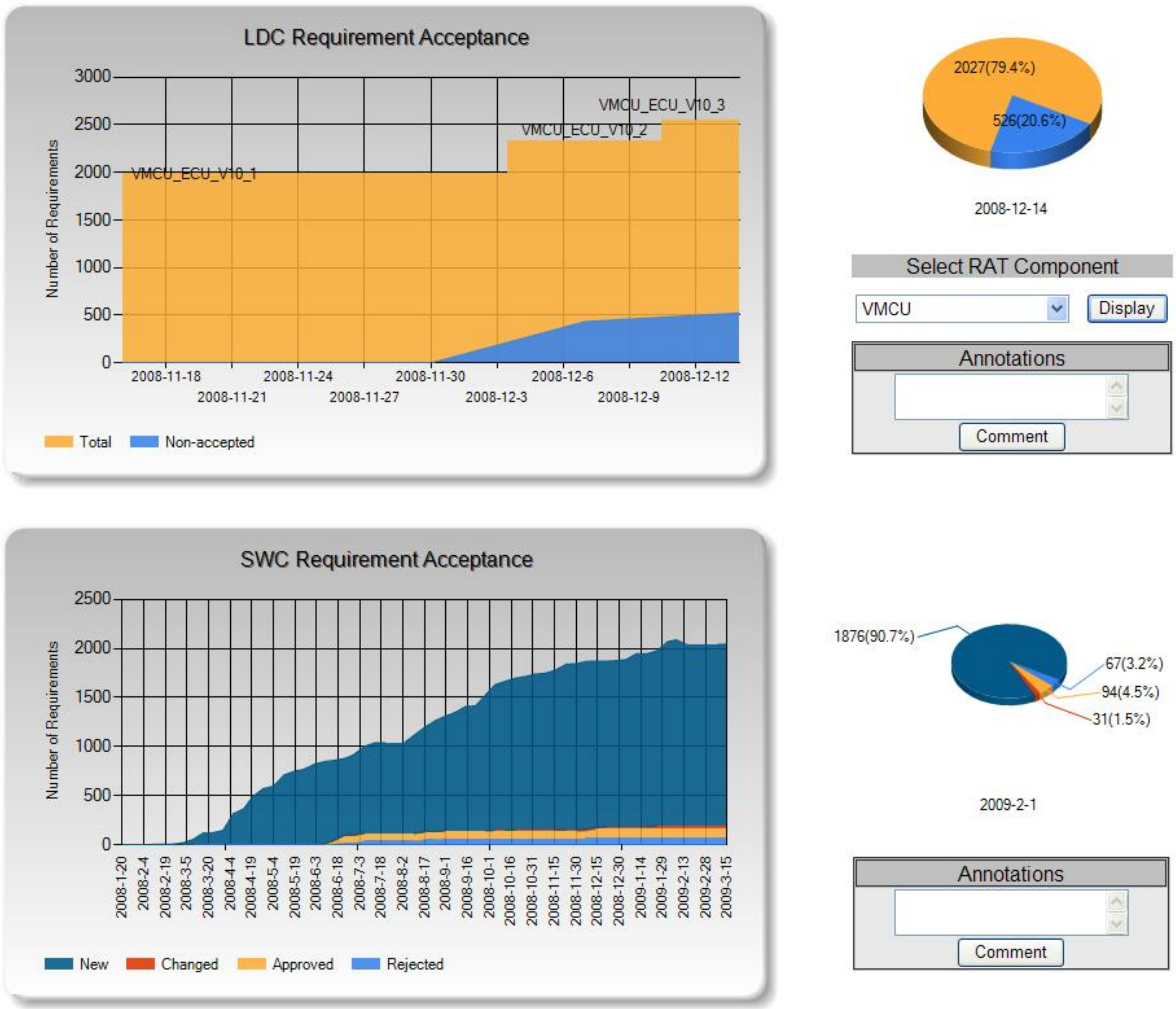

Figure C.4: Interface of Requirements Acceptance for ESW

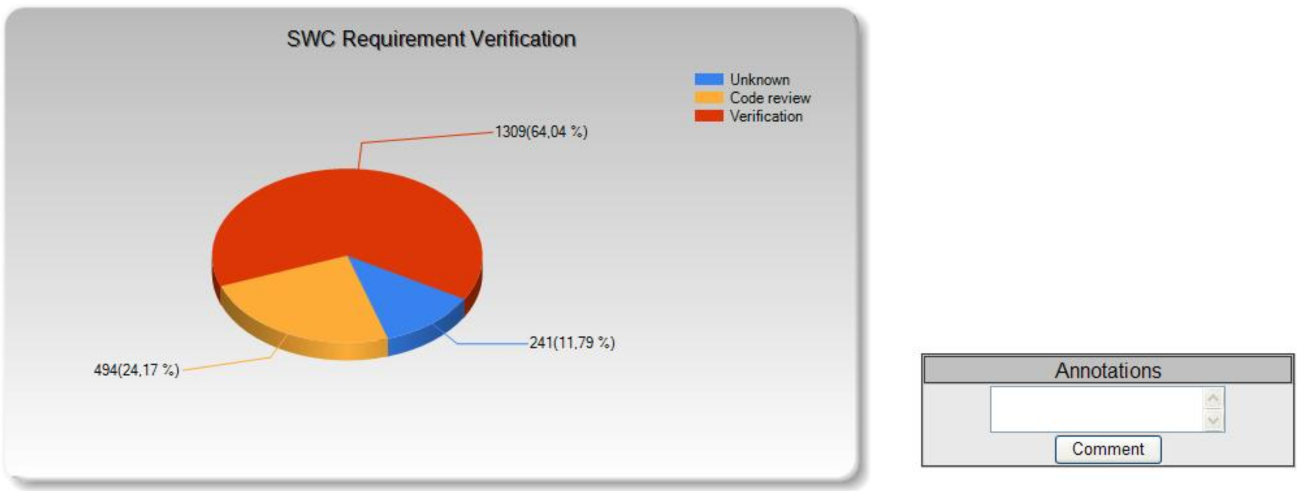

Figure C.5: Interface of Requirements Verification

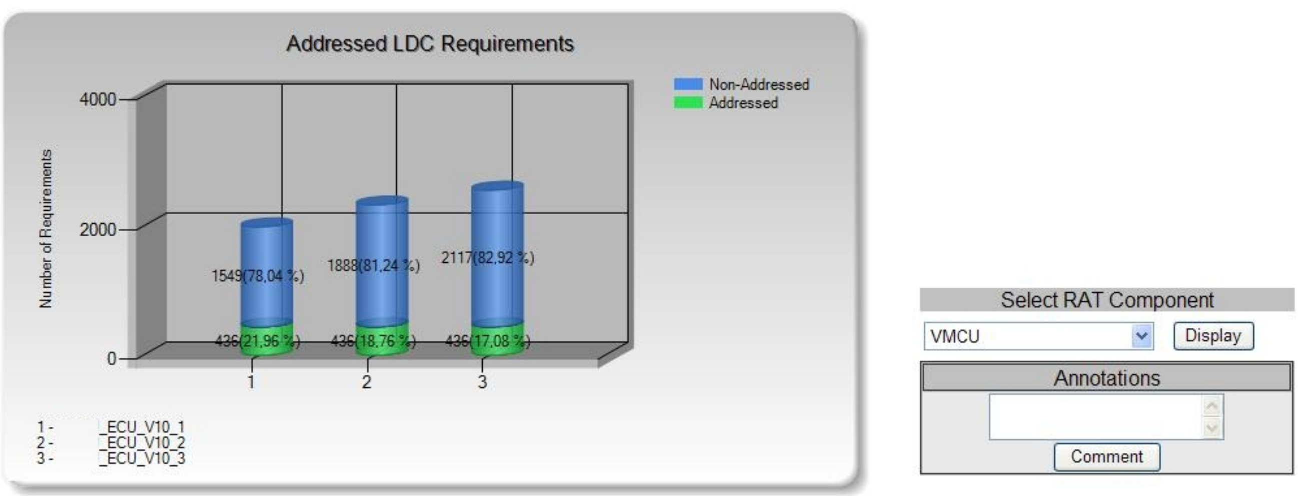

Figure C.6: Interface of Addressed Requirements

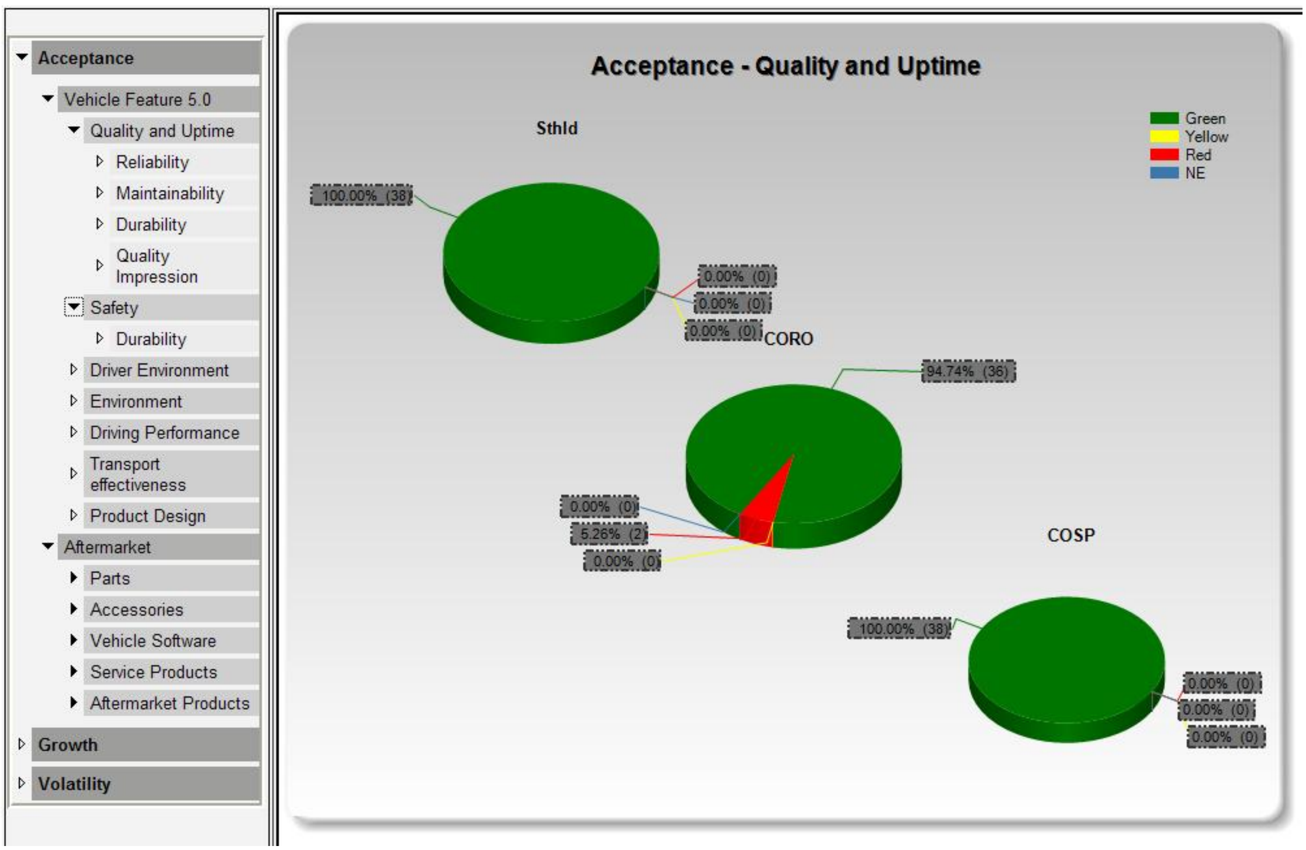

Figure C.7: Interface of Requirements Acceptance for CO