Master Thesis
Software Engineering
Thesis no: MSE-2010-08
April 2010

An Analysis of the Cost for Adhering to the ECSS Standards in the Space Industry

Niclas Torstensson

School of Engineering
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author(s):
Niclas Torstensson
Address: Halssmyckev. 9D, 443 36 Lerum, Sweden
E-mail: niclas.torstensson@gmail.com

External advisor(s):
Joakim Ekwall
RUAG Space AB
Address: 405 15 Göteborg
Phone: +46 (0) 31 735 00 00

Erika Hult
RUAG Space AB
Address: 405 15 Göteborg
Phone: +46 (0) 31 735 00 00

University advisor(s):
Dr. Robert Feldt
School of Computing

School of Engineering
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Internet : www.bth.se/tek
Phone : +46 457 38 50 00
Fax : + 46 457 271 25

ABSTRACT

The introduction of standards in software development is more or less common practice today and regarding software for space applications, standards are one of the primary mechanisms to ensure a sufficient quality level. The space industry has special requirements in terms of reliability and dependability and the European Cooperation for Space Standardization (ECSS) standards is a tool used to fulfill them. The use of standards provides many benefits however it also comes with a compliance cost, in terms of, for example, additional documentation and activities. For making the right decisions on which development and quality assurance activities to focus on, it is important to know not only their added value but also their costs.

In this paper a method is presented for a Cost of Standard Compliance Analysis (CoSCA) in software development. It consists of seven steps and is based on a model which divides the costs into four different types based on the actual reason for conducting activities required by the standard. The four different types of compliance cost are quality-adding, confidence-adding, adherence and development necessary costs. The adherence costs are those costs that are required by or follow from the use of a standard but do not add any immediate value to the functionality, quality or quality assurance of the software. As a result of a CoSCA analysis, the cost for complying with the standard for all the separate development activities is calculated. Initial results from applying the method to a company, that develops software for the space industry and complying with the ECSS standards, are presented. The result gives an indication of potential optimization possibilities and for the top ten adherence activities examples are presented. The evaluation showed that the method is practical and usable with an acceptable level of effort and that it helps pinpoint development activities with high adherence costs.

Keywords: standard, cost of standard compliance, method, space industry, ECSS

Contents

1 INTRODUCTION

The possibilities presented by the introduction of Internet have made it a lot easier for companies to establish themselves on a global market [FRI05] and as the world shrinks the demand of high quality increases. A way to ensure that a product meets a certain level of quality is to use standards [TOU02], however there are other reasons for using a standard as well, for example a standard can be required externally by the customer rather than internally. When speaking of standards in general a common interpretation is product standards which many people come in contact with in their daily life, for instance mp3 music files. Product standards can be described as a set of requirements that a product have to fulfill regarding connection protocol, data format or even design. However standards can also be used in development with the intention to establish a known process. Under that circumstance standards are viewed [FEN93] as a set of requirements which ideally can be related to specific entities in the process. A more general definition is published in the Oxford Encyclopedic English Dictionary defining a standard as "an object or quality of measure derived as a basis or example or principle to which others conform or should conform or by which the accuracy or quality of others is judged"[FEN98]. According to [EMM99] the practices enforced by a standard are often defined in terms of constraints for documents, for example documentation of user requirements or development plans, however an earlier study [PFL94] shows that many standards are not standards at all. They can more correctly be described as guidelines and in those cases the organization in question is recommended to objectively address their goals and only apply relevant parts of the guidelines [PFL94].

The use of standards and technical regulations in software development is more or less common practice today. In year 2000 it was reported [EMA00] too be more than 250 software engineering standards in use and that figure was expected to grow. They are developed to protect customer safety [MAS05] or to achieve other quality goals [AHM09] and they can be introduced by stakeholders, development companies, governments, unions such as EU or even by private initiatives to ensure or grant a higher level of quality. However all standards are not equally effective [FEN98] and many software standards do not explicitly state their resulting benefits. Making it even harder is the fast changing nature of software development and thus the accuracy of a standard in different domains [BOE00]. Assessing the benefits is left to the user and their means of measuring it. Attributes commonly implied are reliability, maintainability of code and productivity of personnel [FEN93]. There are a large number of available standards for software development both generic and case specific and many of them are results from years of efforts by practitioners and academic experts. Others are developed by professional bodies and applicable for certain categories of software [TOU02], one example is the European Cooperation for Space Standardization (ECSS) standards.

In the European space industry the ECSS standards are commonly required to be applied, especially when the customer is the European Space Agency (ESA). The European space industry has had different development standards in work since the early 1980's and at current time revision C of the ECSS standards is in use. The space industry is one market with extra focus on quality and high-dependable software and they are no exception when it comes to using standards [AHM09]. Like all software development today the space industry is pressured [AHM09] to deliver software with a higher quality to a lower cost, resulting in a need for optimization of their methods and standards. Standards often need tailoring to be effective within a company, but they have proven [TOU02] to be more than a necessity. However even when regarding them as a positive advantage it is important to remember that parts of them, different parts depending on the circumstances, are without contribution.

A company in the European space industry with ESA as their main customer is RUAG Space AB, and in this paper they have been chosen to represent the space industry. They specialize in on-board equipment for satellites and space launchers including computer systems, antennas and microwave electronics and adapters and separation systems. With ESA as their main customer the result is a mandatory acceptance of the ECSS standards' content and it forces them to conduct a series of activities they would not normally conduct. Activities that come with many benefits, however they also come with a cost of compliance [FEL09]. Costs that the company would not have otherwise, e.g. specific documentation requirements, verification activities needed to ensure that the standard is being followed and other activities required by the standard [TOU02]. Moreover the initial cost of introducing a standard can be substantial [MAS05]. As mentioned standards are a common tool in software development today and their attributes are in general accepted, however there are not many attempts made to evaluate the actual success regarding the result [EMA00]. All standards are not equally effective and their resulting benefits are not always clear [FEN93].

1.1 Motivation

A method to analyze a standard's impact and cost would go a long way towards a more effective development process and in this paper a Cost of Standard Compliance Analysis (CoSCA) is performed for that very reason. An evaluation of the effectiveness can be of high importance when appliance to a new standard is considered or for future tailoring, of an already applied standard. New editions of a standard can also be expensive to implement, especially if it force technological changes [FEN98]. To ensure compliance with a standard measures have to be implemented, to assess that the effort has desired effect [MUL07]. This can be considered as a problem since the high demand of thorough documentation and confirmation of conformance with the standard takes resources away from the actual quality assurance activities together with verification and validation [AHM09]. In the Netherlands an approach to locate the administrative burden for a country imposed by regulations was introduced and it resulted in a reduction plan estimated to save €4.1 billion in 2007, approximately 25 % of the administrative burden, and similar results have been reported in Denmark [SCM05A]. The administrative part cannot be ignored since it is a confirmation of sort, that all quality aspects have been accounted for, however it can be made more effective and the cost can be lowered.

The cost introduced by efforts to improve the quality must be financially justified and software quality is to be viewed as an investment that should bring a higher financial return then cost [SLA08]. Therefore the appliance of a standard in development must not be imposed but rather managed, since total compliance with all parts in the standard may not be desirable [EMM99]. In the space industry political and market pressures forces companies like RUAG to investigate possible improvements to increase the quality and at the same time develop software to a lower cost [AHM09]. Most quality experts agree that cost of quality efforts should be tailored to fit the specific company's process and not just copied [SCH06]. Knowledge of the costs in relation with their benefits is necessary [PRE09] to be able to minimize those with no adding value. Knowing the cost is also highly beneficial when speaking with the stakeholders. It gives a good edge in negotiations to tell a customer, that there is no problem following a specific standard only that the cost of doing so is “this high” “because of”. It is also indicated [SLA08] that financial justification for each software quality effort gives a better basis for adaption possibilities in the development process. This by stating [SLA08] that it is possible to spend too much on software quality and it might be beneficial to reduce quality of an effort if the cost is high. Traditionally quality costs have been separated as prevention-related, appraisal-related and failure-related [FOS96] and they can all be related to conformance, which is described in detail in chapter two. This approach is also partly applied in the VAMOS framework.

VAMOS is a management and optimization framework for verification and validation activities and was recently created in the context of RUAG’s development process. At the current time it is not implemented, but preparations are being made to include, at least part of it, in the process. In VAMOS the cost reducing effort is essential and it has proven [PRE09] to work by locating process steps where improvement is possible, however to give even better proposals, the approach can be complemented. Analyzing the cost of compliance for standards and restrictions have been shown [PFL94] to be of great importance and the VAMOS framework by itself does not give a complete picture of the activities' costs in a company with projects tightly restricted by standards. Therefore it can be complemented with a detailed cost of compliance analysis to give more accurate improvement propositions and an analysis dependent on more than proposed measurements also grant a higher quality [PFL94]. As an example, to be able to reduce administrative costs they need to be pinpointed and measured [SCM05B]. A difficulty when attempting to use measurements in any cost assessment is the lack of information about cost, in companies. They are not detailed enough to reach a level that will allow us to understand the implications on the cost from, in this case, using a standard [PFL94]. So by performing an analysis of cost of compliance for standards used at RUAG, based on both measurements and expert estimations, quality will be added to VAMOS along with strengthening the bargaining position and increasing the understanding of the standards reviewed. Research has shown [FEL09] that RUAG experience the ECSS standards to have a large impact on software development and a somewhat negative effect on their efforts. However the research also states that RUAG’s assessment regarding ECSS contribution to quality is positive which makes an analysis of the cost of ECSS compliance attractive.

The motivation behind the creation of the CoSCA method can be summarized as followed:

- Knowledge of the standard's effectiveness.
- The cost introduced by changes in new editions of the standard.
- Locate unnecessary administrative cost.
- Locate parts in the development process which does not contribute to the quality or confidence, thus can be limited or removed to save time and money.
- A complement to the VAMOS framework to help in improvement of the verification and validation process.

1.2 Methodology

The goal is to create a CoSCA method for cost assessments, which can be applied in a company or department either as stand-alone or together with the VAMOS framework. Furthermore the concept will be applied at the software department in RUAG Space AB and evaluated. The analysis will be conducted based on one smaller and one larger software project at RUAG, to account for process differences, and the results combined into a more general result. Since the result shall be of a general character the projects to analyze will be chosen based on how well they represent the most common working order. Furthermore the choice will depend on available data in the different projects. As it is now some projects conducted at RUAG have more detailed time reporting activities than others and obviously the selection shall be made within these projects.

Finding the most beneficial approach for the analysis will be based on a thorough literature review in combination with experience, taking into account current standards, the project process and available data. A stated [BOE00] conclusion is to use more than one technique and compare the result, to produce realistic results. Taking this into consideration the general idea is to use data from time reports in combination with cost estimates. In this case estimation is not considered as an elicitation technique instead it is used as a synonym for best guess which is what interviews, observations as well as expert assessments mostly is all about. The use of collected data from the projects is a given choice since it is already available and therefore easily can be included in the analysis. However the time report data is not collected within the projects with this analysis in mind and therefore it will most likely not be complete. Estimations by experts are for that reason included as a separate step in the analysis to ensure a complete set of data. Experts are important to include in the analysis since a baseline for reference is necessary to review any success with confidence. If there is nothing to compare to, assessing the result of a possible improvement is impossible [PFL94]. In software development at RUAG there are no projects or efforts without regulations and therefore the knowledge and experience of their experts is the most cost effective way to gain usable assessments

A preliminary approach has been developed (see figure 1) based on available resources at RUAG. The standards have been divided into separate requirements and matched to existing project activities. This is regarded as the key stage in [PFL94] and related articles [FEN98]. Measurements will then be produced for the activities with an historical analysis of current projects' process documentation and logs. In the software cost estimation area it is called [HEE92] estimates based on reasoning by analogy. After the data is collected the standard compliant part for each activity has been estimated by experts [HEE92] and the result compared with the measurements. The specific approach to use regarding the estimation has been decided based on experience and a review of the most common software engineering approaches. In the event of nonexistent measurements and no good basis for estimates, the cost has been estimated indirect from other projects. The expert estimates and existing relevant measurements are then the foundation for the chosen approach. An interview with a RUAG executive along with studies of other cost analyses has determined in which format the result shall be presented and thus which data to collect.

An analysis of current standards and recent updates to them has been conducted to determine how general the result will be. The analysis is based on documented guidelines for the ECSS standards along with expert assessments. Already known is revision C of the ECSS standards, which has been under integration into RUAG's process for the past year and applied in all new projects for 2010.

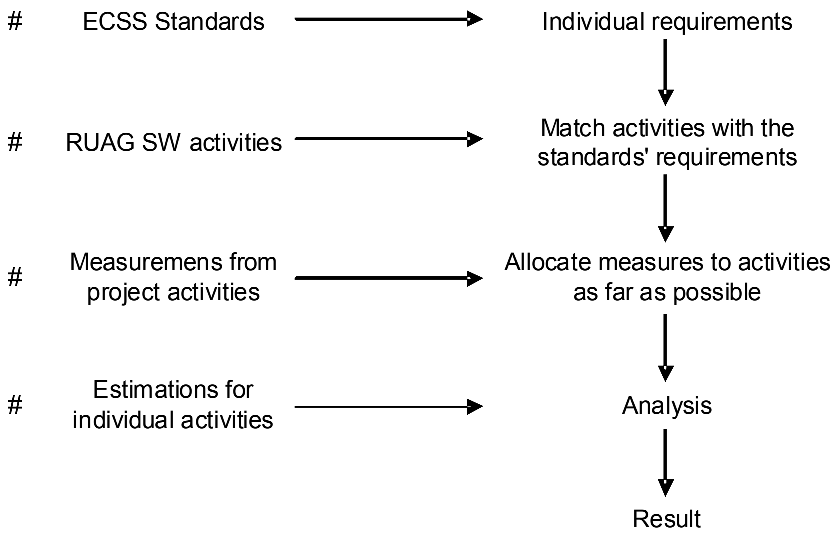

Figure 1 Preliminary approach

1.3 Limitations

Limitations of the study are set according to priorities made by RUAG and the amount of available time. The analysis of the ECSS standards is limited to software related development and furthermore by not addressing operational requirements presented in the standards. The operational phase of the software includes process implementation, support during operation and operational testing in the intended environment [ESA03A] and this is deemed to be out of scope for software development at RUAG. Man-machine interface and in-flight software restrictions are two other sections in the ECSS standards, which are related to software but not relevant here. Software developed at RUAG is not required to include an interface for man-machine communication and even though the resulting products are included in space shuttles and satellites they are not controlling the actual flight and for that reason in-flight restrictions are excluded. For RUAG's part this means that their software development section is included in the analysis along with parts of the system design and the verification and validation sections that are related to software.

The ECSS standards requires a fair number of documents to be planned and created and by doing so introducing a cost, however templates constructed for continuously use for the benefit of the standard is not considered as a cost. It is an initial cost that is no longer relevant, instead only the cost for completing the templates is considered. The cost for updates of the templates with regard to revision C is estimated and held as a separate issue together with other affected costs.

1.4 Structure of the Thesis

The thesis consists of four major parts. The first part consists of an introduction of the subject along with the affected ECSS standards and RUAG Space as a company, to give the reader a general understanding of why and how the analysis is conducted. The initial approach to the problem is in other words proposed and motivated. In the second part related research is presented to investigate possible appliance in the coming analysis. The third part is divided in two chapters, where a detailed description of the approach for cost of standard compliance analysis is proposed in the first chapter, followed by how it is applied in a case study at RUAG. Finally in the fourth part the result from the case study together with discussions and conclusions drawn from result is presented.

One appendix is included with details of the estimations made regarding activities related to the ECSS standards.

2 INDUSTRIAL CONTEXT

As mentioned in the introduction RUAG Space AB is representing the European space industry in this paper and the focus is the cost for complying with the ECSS standards.

2.1 RUAG Space AB

RUAG Space AB, former Saab Space AB, was recently bought by the RUAG group owned by the Swiss confederation with their central office in Swiss, Emmen. RUAG Space AB is located in Gothenburg and Linköping, Sweden specializes in on-board equipment for satellites and space launchers including computer systems, antennas and microwave electronics and adapters and separation systems. They are part of the space division within RUAG Holding AG and have been in the field since the late 60's. RUAG in general is a technology group focused on defense and aerospace technology. The defense section is focused on weapons, command and control and simulation systems while the aerospace sections is a specialized in aero structures, in servicing and outfitting airplanes and helicopters and developing and manufacturing satellites and launch vehicle technologies.

In this paper, the software section at RUAG Space and sections related to it (see figure 2) are included in a case study, performed based on a constructed CoSCA method. The related sections are system design and verification and validation, however not their entire sections, only the parts with activities related to the software development. For system design it is the creation of the requirement baseline which includes a software system specification and software interface requirements from which the software section creates their technical specification. The verification and validation related parts are all activities were software testing is being prepared, performed or documented.

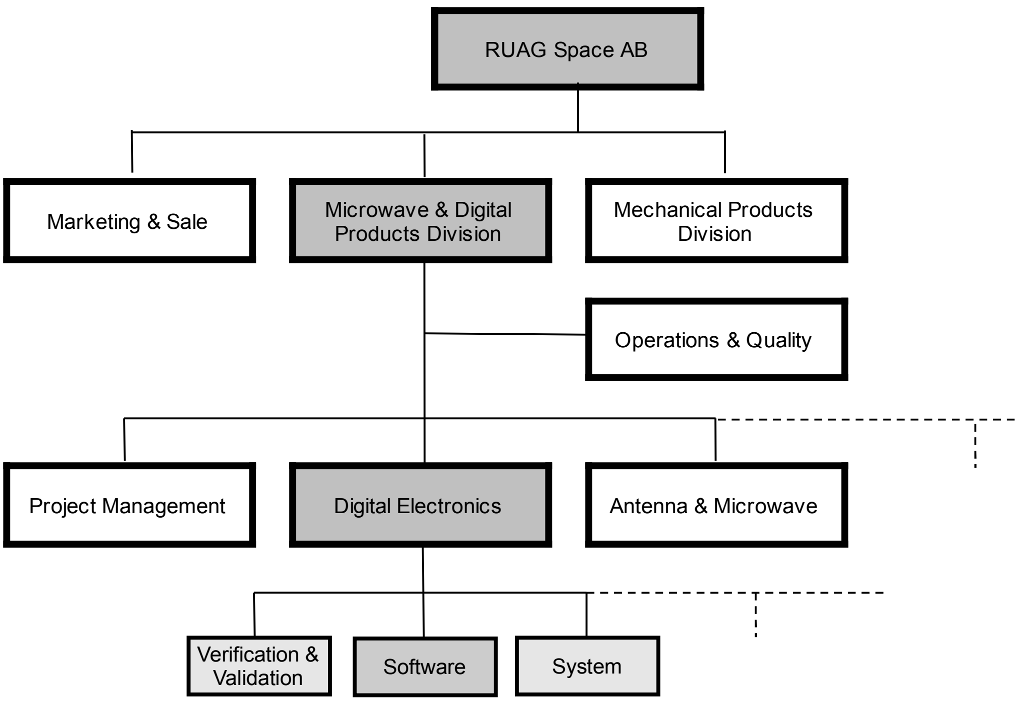

Figure 2 RUAG Space AB Organization

2.1.1 RUAG’s process

In software development RUAG’s process can be described by a V-model [PRE09], see figure 3.

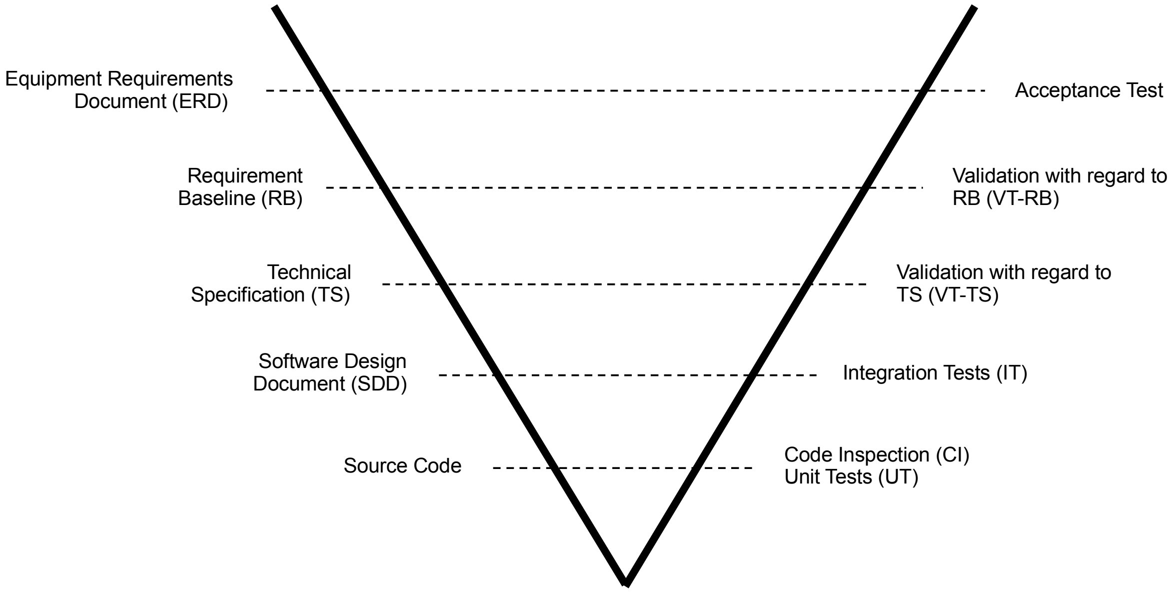

Figure 3 RUAG development process *(V-model perspective)* [PRE09]

It starts with an Equipment Requirements Document (ERD) where the high level requirements are specified, based on the customer's requirement specification. Based on the ERD the Requirement Baseline (RB) is created to describe, in more detail, the high level requirements of the project. The software part is described in a Software System Specification (SSS) and it focuses more on what shall be done and not how to do it.

So far the system section is the one responsible however the responsibility changes to the software section when the Technical Specification (TS) is addressed. The changes in responsibilities between the sections can be viewed in figure 4. In TS the software requirements are refocused to become more solution-oriented and describing parts of the technical implementation together with inputs and outputs. The requirements are documented in a Software Requirements Specification (SRS) and the interface described in a Software Interface Control Document (SW-ICD). Using the Unified Modeling Language (UML) and a real-time specific profile the next step includes creating the design and document it in a Software Design Document (SDD).

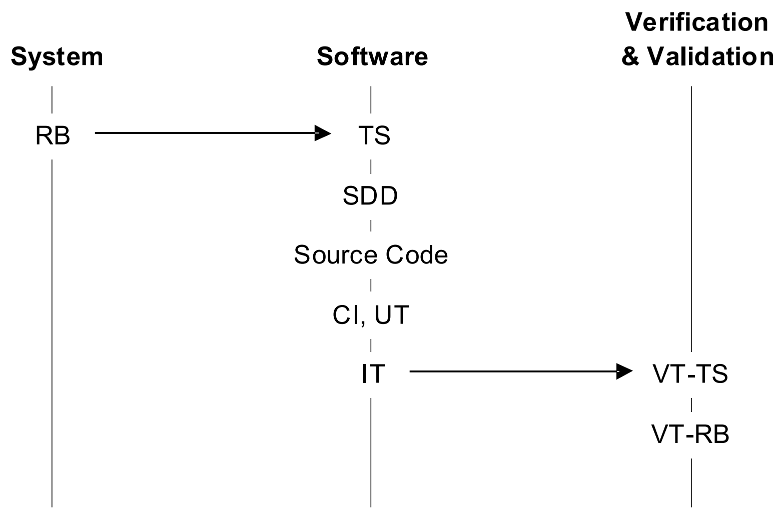

Figure 4 Responsible sections

Following the design is the implementation and development of the source code and validation tests which are conducted simultaneously. Code inspection (CI) and unit tests (UT) are then performed to separately verify the different modules implemented. After the modules are verified they are integrated and tested (IT) again to confirm that they are compatible.

The Software must also be verified against the requirements to confirm that the customer will have what asked for and this is done by testing with reference to TS and RB. When performed acceptance tests are conducted by the customer as a last step in the process to ensure that everything is in order.

2.2 The ECSS standard

European Cooperation for Space Standardization (ECSS) is as mentioned an initiative to create a more user-friendly collection of standards for use in all European space activities, by concentrating on coherency and on a common goal [JON97]. The European space industry has had different standards in work since the early 1980's and in 1994 the ESA Council decided to replace the ESA Software Engineering Standards (PSS-05-0) with the ECSS system of standards. The PSS system was used as a mandatory input in the preparations of the new system and the European Space Agency (ESA) along with European national space agencies and the European space industry are all represented in ECSS team [JON97]. Its intent is to help companies not to miss important aspects by concentrating on project level and not organizational level, by providing flexibility and making it possible to apply it with other standards and by introducing tailoring options [AHM09].

In the area of software development two standards are of major interest, the ECSS-E-ST-40C and the ECSS-Q-ST-80C where the 'C' in the end stands for revision C of the standards. The ECSS standards updated from revision B to C in March 2009 and when this analysis was set into motion RUAG was still in the process of integrating the changes into their process along with affected documents. References in templates and in project documentations at RUAG were all referring to chapters or parts of the revision B edition and for that reason the analysis was conducted on revision B.

However it was complemented by a short analysis of the differences between the revisions including potential effects it could have on the result.

In revision B the relevant standards are foremost the ECSS-E40B and the ECSS-Q80B standard. The ECSS-40 standard is divided in two parts ECSS-40Part1B and ECSS-40Part2B and it covers the entire development process, from requirement to delivery and maintenance. In the first part titled “Software – Part 1: Principles and requirements” the space system software engineering process is described and requirements for details are presented. They are divided according to internal processes such as management, design and verification with references to required files and documents. In the second part titled “Software – Part 2: Document requirement definition” the required files and documents are presented more thoroughly and detailed guidelines for the more important documents are given. The ECSS-Q80B “Software product assurance” standard complements the E40 standard with a set of quality requirements related to the software development and maintenance. The process lifecycle set in place by ECSS for software development can be viewed in figure 5. The numberings in the different boxes are a reference to chapters in ECSS-40Part1B and the abbreviations at the bottom aim at different reviews with collections of documents beneath.

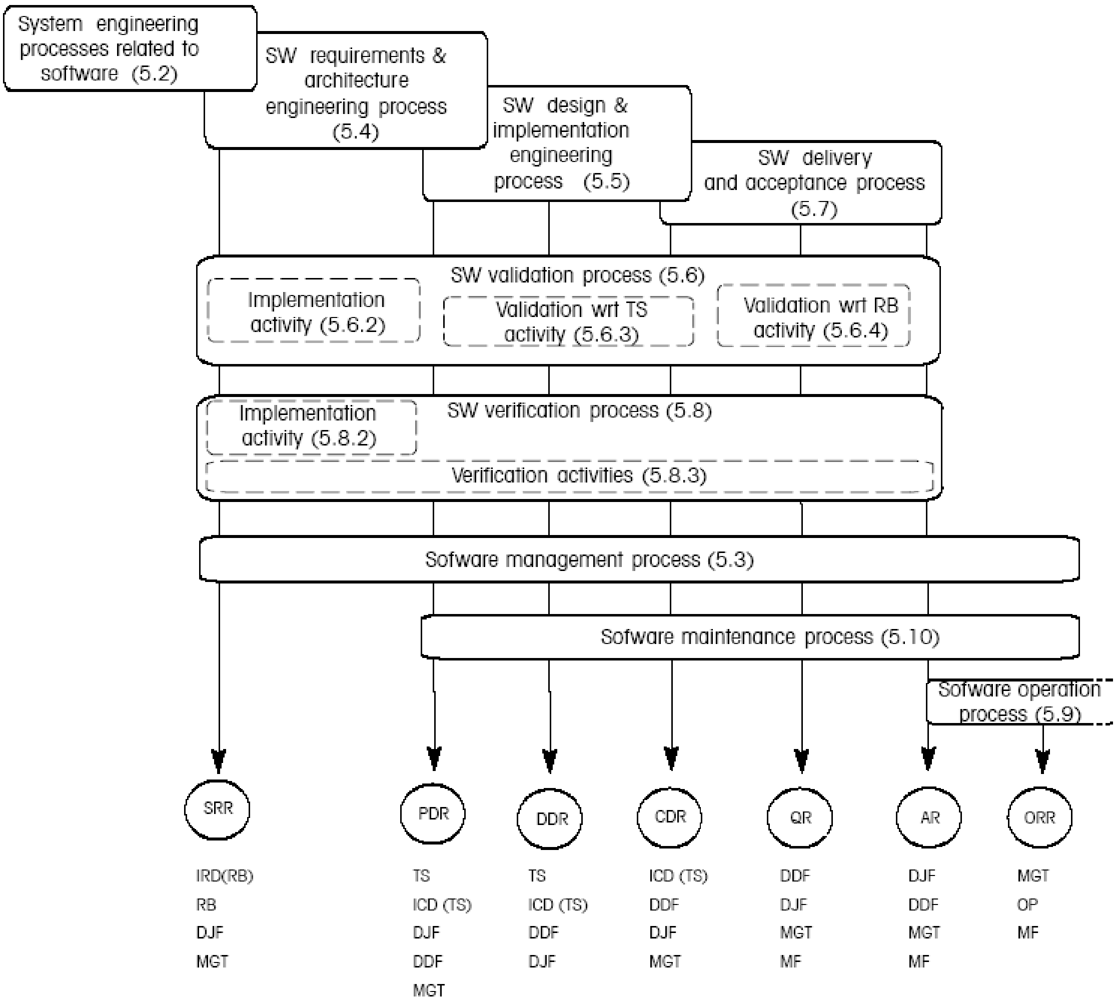

Figure 5 Overview of the ECSS software life cycle process [ESA03A]

3 RELATED WORK

Quality has a major part in software development and since increasing the quality can be one reason to use a standard, they are related to some extent. For that reason an attempt was made to find useful approaches from methodologies concerning quality and cost related to modern economics. The promising parts of that is presented in chapter 3.1 follow by, more standard and cost specific approaches, from the software engineering field, in chapter 3.2.

3.1 Quality and Cost

The cost of quality refers to the extra cost an attempt for higher quality brings and it is not directly related to analysis of a standard, however some of the same principles used in a cost of quality analysis can be applied. A possible approach is to divide quality costs into prevention-related, appraisal-related and failure-related (P-A-F) and from that a relation to conformance can be established, as shown in figure 6, with internal dependencies between the costs. The curve presented is referred to as the Lundwall-Juran curve [FOS96] and conformance can be described as level of perfection.

Prevention costs are introduced by planning, training and other activities aiming to prevent defects before they occur, appraisal costs can be categorized as inspection or assessment costs and failure costs are those costs that can be related to mistakes resulting in a non-conformance [FOS96]. A more general division of quality costs is conformance and non-conformance [SLA08] which is applied in the Lundwall-Juran curve without stating it. Prevention and appraisal cost are the fundamentals of conformance and failure-related- and non-conformance costs can be considered as synonyms [SLA08].

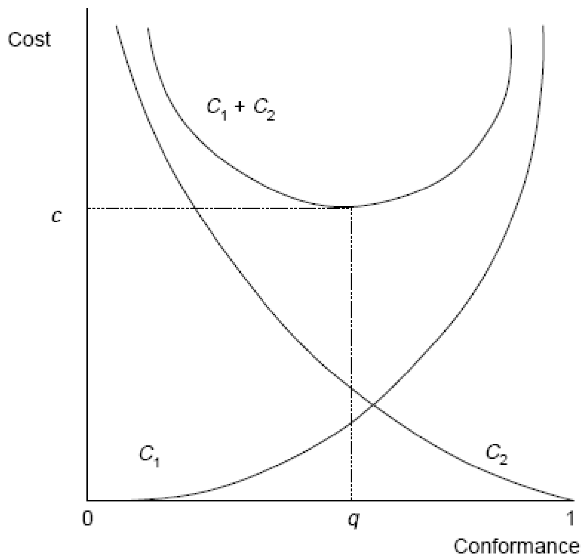

Figure 6 Basic Lundwall-Juran curve [FOS96]

In the Lundwall-Juran curve (figure 6) C_1 represent prevention and appraisal costs and C_2 represent failure costs and $C_1 + C_2$ describes the total cost related to quality. The thesis of the Lundwall-Juran concept is; as conformance improves, the prevention and appraisal costs increases while the failure costs decrease [FOS96]. A value of 1.0 in conformance equals to 100% (the maximum) and as can be view in figure 6, a minimum cost level exists for quality suggesting an economic quality level q which is lower than 1 [FOS96]. However alternatives have also been observed in companies where the quality has been close to 1 with a lowered quality cost [FOS96] and this implies variation possibilities for the Lundwall-Juran curve. Regardless the main conclusion suggesting an optimal economical quality level less than 1 is still valid since the prevention and appraisal costs summarized will have the opposite trend of failure costs. All together this implies a need to assess which activities to perform and by regarding it as an investment of higher quality, some economical models can be applicable.

Modern theories regarding economics have shown that an asset's value is decided by calculating the present value of future payments [FRY03]. This can be applied when calculating the potential of an investment by enabling a comparison between the return and the investment. There are some methods that can be used and return on investment (ROI) and net present value (NPV) are two of them. In the ROI approach the result including financial incomes is divided by the total capital and it provides a percentual figure describing the rate of return with regard to total capital. Total capital is everything a company owns together with all depts, such as bank loans and amount due to suppliers. The problem with these methods when applied in investment calculations concerning IT is the fast changing nature of development processes [FRY03], where software development are included. If applied in the assessment of introducing a standard in development one approach could be too; divide the man-hours saved after the introduction with the total allocated time for a project before the introduction. Regardless of the probability of a correct result the approach concentrates on calculation of a potential investment and can therefore not be applied in this analysis. Moreover rate of return is not of interest since it does not describe or can be related to the percentage of time spent on noncontributing activities. However possible use can be found in a related approach called return of software quality.

Quality metrics concerning software development have been introduced over the years and return on software quality and cost of software quality are two of them. In general however very little has been published on the subject and many experts agrees in the proclamation that cost of quality programs shall be customized for every company [SCH06]. The P-A-F approach [FOS96] in turn or similar ones can be used as a basis for calculations even though the structure of the models will differ from company to company [SCH06].

3.2 Standards and Cost

To gain results with the highest confidence when assessing the cost or benefits with standards, one must compare it with a world without any standard dependencies or restrictions. In RUAG's case this is not possible since only one reality exist and the only other way is to model the differences in outcome for when a standard is applied and when it is not, which is one of the most difficult things to do in cost analyses [HAR99]. Creating a baseline to compare against requires thorough knowledge of the projects being analyzed including experience with similar projects without the same particular restrictions and the result will still to some degree be hypothetical [HAR99].

In general two models exist for analysis of standards or restrictions and one newly developed combining the two of them. We have the SMARTIE (Standards and

Methods Assessment using Rigorous Techniques in Industrial Environments) project [FEN93] and then there is the Standard Compliance Model (SCM) [SCM05A], focused on administrative burden. At RUAG along with the VAMOS framework an idea to combine the SMARTIE and SCM approaches where proposed [PRE09]. The VAMOS project is a management and optimization framework for verification and validation activities relying on measurements of fault data and costs. All three approaches are more thoroughly described in chapter 3.2.2 to 3.2.4.

In all cases the different types of cost have to be separated to be able to make any assessments and P-A-F is one way to accomplish that. Another proposed division [FEL09] is to divide them into three different categories in an attempt to separate the positive and negative costs.

(1) Quality related costs
(2) Confidence related costs
(3) Adherence costs

Quality related costs are costs introduced by activities that add to the quality of the software by removing existing or prevent future defects. As long as it results in a higher benefit than cost, it is considered to be of a positive nature and not a cost to be removed. The activity itself can be optimized to make the process leaner or in an attempt to reduce the cost, however it cannot be removed without lowering the quality. Quality related costs can also be divided further and as mentioned earlier the P-A-F scheme [FOS96] is a possible approach, however by using P-A-F confidence related costs and the adherence costs will be considered as part of quality and not separated. The decision to divide based on quality or not depends on the intention behind the assessment or analysis that is taking place. The P-A-F breakdown is not useful if the focus is directed on adherence and not quality.

Confidence related costs are costs which actually increase the confidence in the software. The benefits are not in the sense of high or low quality instead they are related to marketing and faith in the product within the organization and for customers. If the estimated benefit from marketing the product or gaining the customers trust is higher than the cost, it is not to be eliminated. Finally there are adherences costs introduced by the standard that does neither of the above and hence are of no obvious benefit. They can be categorized as administrative or process costs forced upon the company without any contribution. The primary problem is that the standard requires them and therefore they may not be possible to completely remove. Some standards have tailoring possibilities with varied flexibility, however if elimination of the cost is impossible what should be done is to reduce as much effort spent on the task as possible.

3.2.1 Cost calculation techniques

Models for cost calculations without the aspect of standards are more common, however the use of them can be limited depending on available resources, for example, time, money and personnel. Parts from various cost calculation methods can always be combined, even if they are developed for another purpose the general structure allows for a wide range of use and since many of them have been tested on multiple occasions the quality they provide can be confirmed. Commonly used techniques are interviews, estimates, surveys, observations and report data collection, where interviews and surveys are the general approaches when primary data sources are not available [FRY03]. Validation or analysis of processes within a company conducted by an external party can best be accomplished by expert reviews, simulations, surveys, or case studies [FOS96].

When selecting an appropriate method to use the choice depends on how the problem is formulated, the purpose of the research and as mentioned available resources [FRY03]. Observations is a useful tool to get a general understanding of a process or for elicitation of requirements, however it will not result in any specific data and if special anomalies are of interest it can be very time consuming to wait for them to occur. Report data collection or data measurements on the other hand have no other contribution except data and since facts are the most reliable basis for an analysis it is mostly what to strive for. When measurement collection is not possible the second best is expertise-based techniques, which is a fairly common approach even though it is considered second, due to the nonexistent facts [BOE00]. Another word for it is estimation and its intent is to captures the knowledge and experience from experts in the area being analyzed. Expert estimations are based upon the expert's prior knowledge and experience form projects, and their outcome. It is important to take the experts' error in judgment into consideration, in the analysis [BOE00]. In an attempt to minimize the errors in judgment made by the experts the Delphi technique and the Work Breakdown Structure (WBS) technique have been developed [BOE00]. The Delphi technique was originally made for making assessments regarding the future but it is also applicable for estimations concerning present time. To begin with the estimations are done individually without any consulting between the participants and the result is collected and distributed to all the participants. As a next step estimations are performed individually once again and this time with the knowledge of the others' answer, resulting in more closely matched answers [BOE00]. The WBS approach in turn is not about the actual estimations instead it focuses on the project elements and sorting them in a hierarchy to simplify estimations [BOE00].

Estimates are a common approach and often used as a part of another technique, for instance within interviews. There is nothing forbidden against mixing the techniques the only thing to remember is why they are chosen in the first place to prevent them from being a hindrance for one another. Software cost estimation approaches can be divided into four different categories [HEE92].

(1) A parametric approach
(2) Expert estimates
(3) Reasoning by analogy
(4) Available capacity
[HEE92].

Estimates using a parametric approach can, as an example, be the sum of other calculated costs, which make it more reliable in the way that there are figures supporting it [BIT01]. This is supported in WBS' hierarchic structure, where the underlying layers combined shall match the whole. Another way is for experts in a given area to perform the estimates and reliability then goes hand in hand with the experts' trustworthiness and ability of remembering historical facts. The danger with such estimations is that the expert's rule of thumb later on can become accepted as a fact and then used in an inapplicable situation [HEE92]. Reasoning by analogy is jet another approach where the key is to find similarities with other project or situations and base the estimation on analyses of them. It is the most common technique used and it is more reliable since it deals with facts and not rule of thumb. However it requires a thorough analysis of the similar projects to establish the differences and their effect. Estimations can also be made based on available capacity, by in reverse estimate the spare time and then subtract it from the total. The danger here is stated in Parkinson's Law, that work has tendency to always fill the time appointed and therefore there is no spare time [HEE92]. Each of these four techniques can be used separately however they are often combined for a better result. Better result means higher quality and since a lot of research shows that all the estimation techniques used today contribute a low level of quality they need to be combined. A combination of techniques is also preferable since all of them have their pros and cons and two or more together would strengthen each other [BOE00]. Every use of these techniques can then be approached from two different directions. First there is the top-down approach [HEE92] where the total cost of the entire project is calculated or estimated and then divided among the current activities. Second there is the bottom-up approach [HEE92] where every activity or component is estimated individually and then combined for a complete picture. Estimates are used when there is no absolute answer, if no measurements exist or if they are too expensive to collect. Actual measurements are of course to be preferred if they exist. In New Mexico individual cost estimates were used to calculate the cost of compliance with a new drinking water standard and the result was assessed as reliable [BIT01]. For every instance three separate estimates were completed [BIT01].

Interviews are another technique mentioned and it can be constructed with many different goals in mind. They can be preferred [ALO05] in case studies where the most important result is understanding and not the exact figures, the figures are of course important not only the main goal. However interviews can also be constructed with data collection as its main purpose and preferred [FRY03] when more information than just the questions is sought after. A discussion around the subject can sometimes give as much as the questions them self and even if assumption often have to be made the result will proffer [FRY03]. Interviews are also a good choice when specific persons are considered as more important sources then others [ALO05][FRY03]. Surveys on the other hand are to be chosen if it is more important to have a strong statistical basis [FRY03]. In the time it takes to conduct one interview many times that number can be held subject to a survey. As an example, when compliance with product standards in some developing countries was assessed there were a large number of people to take into consideration and a survey was the best choice [MAS05]. It is also necessary to have a large number of participants to gain a credible result. Moreover it is important to include a fall out in the answers and to put effort in the design of the survey [FEL94]. In a conducted survey sent out to dentists in the US only 35 % answered [FEL94].

3.2.2 The SCM method

SCM [SCM05A] is a method for pinpointing a country's, company's or a project's administrative burden when imposed with regulations. By doing so it is later on possible to reduce the cost for parts with no value. From the beginning it was created to help governments improve the conditions for their business communities however it can be used on any level due to the detailed and clear relation to possible actions in its results. For input it relies on activity-based measurements which make it possible to observe the changes in administrative burdens and not only calculate it for a specific setting [SCM05B]. Some years ago it was the most widely used methodology for measuring administrative costs [SCM05B]. Administrative burdens are not the only costs introduced by regulations, however it is a central part and often costly and therefore there is much to gain by making it more effective. In other words it is important to minimize the administrative burdens in order for a business to grow and improve their competitive position [SCM05A].

The basic idea of SCM [SCM05A] is to break down the regulations of a standard into more manageable pieces so they can be measured and then combined to a percentage of the unnecessary administrative cost. Of interest when measuring are obligations to provide information and data to the public, or in a project the customer. Information obligations does not necessary mean that the information must be distributed, it can also mean that it only have to be available upon request. The information obligations in turn can be divided into one or more data requirements to represent every single element of information. To provide each element with correct data the SCM method helps to estimates the cost of such undertakings on the basis of price, time and quantity parameters.

Activity Cost = Price x Quantity

Formula 1: The basic SCM formula [SCM05A]

Price is calculated by multiplying tariff with the time spent, where tariff is the wage cost plus additional costs in the form of material or a percentage of basic overhead administration costs. Quantity in turn is calculated by multiplying the size of affected businesses with the frequency of how many times per year/project the activity must be completed. The size of the affected businesses in a project is simply the number of simultaneous appearances of the activity. This gives the extended formula:

Activity Cost = Price x Quantity = (Tariff x Time) x (Size x Frequency) [SCM05B]

From the government perspective the administrative burdens are measured by conducting in-depth interviews with a target group of businesses and they are asked to provide time and money spent on each administration activity [SCM05A]. In a project or company the different businesses would be translated to groups or sections instead and the measurements collected by interviewing people responsible for the different obligations.

3.2.3 The SMARTIE project

The SMARTIE project's aim [FEN93], when created, where to provide a measurement based approach for assessing software engineering standards, allowing companies to evaluate the efficiency of a standard in their own environment. It is a two step approach [FEN93] where the first stage addresses the need to divide the standard into single requirements, followed by a measurement-based assessment in the second step.

The identification of standards that most likely will help to improve some aspect of the development is the intention and it is stated [PFL94] that SMARTIE will help by providing potential benefits from using a specific standard, how objective the measurements are, related costs and an assessment of the cost in comparison with the benefits. The key stage to achieve this is to decompose the standard into smaller components [FEN98]. In the process of identifying single requirements the classification of products, processes and resources related to requirements in the standard are used along with attributes stated in the standard as benefits or results [FEN93].

The single requirements will successfully pass the assessment if it can be shown that an increase in conformance to the standard results in increased benefits to such extent that it covers the costs for introducing the standard [FEN93]. This assessment is done for every single requirement and it provide a good understanding of what to focus on in the standard if it is implemented.

When an attempt was made to apply SMARTIE method it resulted in assessments that only to a small extent could be called objective [FEN93]. For traditional engineering standards the benefits are often clearly stated in contrast to software related standards where the purpose behind it is only vaguely described [FEN93]. No attributes that can be measured is presented and notions like “improved quality” or to provide a “standard approach” are the only profits mentioned. To be able to assess the benefits [FEN93] was forced to define their own means for measuring and it resulted in reliability and maintainability of code and productivity of personnel. A conclusion drawn [PFL94] was that relating the standard to an actually cause of a problem was almost impossible and in general it was deemed hard to evaluate a standard with any certainty before it is introduced. The appliance of SMARTIE when assessing a standard already in use will in other words result in a different conclusion.

3.2.4 The VAMOS project’s cost of compliance model

The VAMOS project [PRE09] is focused on reduction of development costs while at the same time keeping the quality. In other words it is a management and optimization framework for verification and validation activities since quality of software to a great extent depends upon those [AHM09]. It relies on measurements of fault data and costs and requires fault type classification scheme as an additional input. It was primary developed for industry and therefore it has to be light-weight in sense of necessary changes to the process and which measurements to collect. VAMOS is an iterative evidence-based process for improvements that reduces the overlap between verification and validation activities. When different activities finds the same fault it is a waste of effort and reducing that overlap will therefore lessen the cost without affecting the quality. Finding the faults as early as possible are also beneficial to the development and by having a fault-slip-through analysis as part of the framework VAMOS can increase the quality and/or productivity [PRE09]. Moreover it helps in deriving and selecting process improvements and presents figures that can be used to prioritize among candidate improvements. The motivation behind it was the rising demand in space industry of shorter development time at a lower cost and still keeping the quality.

At RUAG along with the VAMOS framework an idea to combine the SMARTIE and SCM approaches where proposed [PRE09]. The reasoning was that the existing methods were too limited by them self, regarding the breakdown of standard costs, benefits and drawbacks and the integration into VAMOS. The combination resulted in a measurement based method dependent on the same measures collected in the framework. The first step in the method is to decompose the standard into so called mini-standards, based on relation to entities or the objective they fulfill. Next cost and benefits for each single standard requirement are assessed with different approaches dependent on if they are considered to be of an appraisal or preventive nature. Since the measures used are the same as in VAMOS there would only be a small additional cost for calculating the cost of compliance, if VAMOS where to be used. However the VAMOS framework needs to be verified in a real industry setting before any statements regarding the quality of the measurements can be made.

4 Cost of Standard Compliance Analysis

In this chapter the the CoSCA method is presented and motivated, and the applicable and non-applicable parts from related work are addressed.

The absolute best would have been to use an existing method for the analysis and based on the analysis of related work the SCM approach had potential. It provides a good work structure by first breaking down the standard into more manageable pieces and then conduction in-depth interviews with key personnel. The reason for not using it as a complete method in this analysis can be explained by its goal to locate administrative burdens. The fact that it was initially intended for countries and not projects does not affect the decision but it use more complex calculations then needed. Moreover this analysis must have a broader view and administrative costs are not the only ones of interest, missing is unnecessary and unwanted process activities and their costs. For structure, the initial breakdown of the standard will be used and its necessity is also strengthened by the SMARTIE method where a similar appliance is described. In-depth interviews will also be a part of the analysis appearing as estimation interviews. Attempts have been made [PRE09] to use measurements and other data at RUAG resulting in the conclusion that they must be complemented for a complete view. That said the possibility was still investigated and estimations or similar means of information were deemed necessary. The useful measurements found were included as part of the analysis but they were not complete and not detailed enough.

The SMARTIE approach can, as described, be divided into two parts were the first one regards a decomposition of the standard, which has already been included in the analysis as written above. In its second part a measurement based approach is proposed and as discussed it is not applicable in this cases. VAMOS present an interesting approach since it is based on the SCM and SMARTIE approach together with the quality cost perspective, using the P-A-F scheme to separate the costs. Moreover it is constructed with RUAG as a case company however the adherence cost which is of primary interest in this analysis is only referred to as a cost to be decided case by case and because of that VAMOS needs to be complemented. In the P-A-F scheme the quality costs are in focus and costs with a direct connection to development, adherence and confidence are omitted.

From the different alternatives presented to a measurement-based approach the resources limited the choice to interview and expert estimations. Observations, surveys and simulations were all excluded since neither time nor manpower was available to apply them. Interviews and expert assessments were combined in an approach referred to as estimation interviews. Following the SCM methodology in-depth interviews is to be used but to gain the data sought after specific estimations are necessary and therefore expert assessments are needed. However in the related work including SCM the benefits from interviews were regarded as high and by interviewing experts in the area and ask them to provide estimations for every single activity or requirement, no benefits will be lost.

To summarize, the applicability of the evaluated methods and techniques presented in related work, are addressed in table 1.

Method / Technique	Applied	Reason
P-A-F division	No	The P-A-F division focus on quality and not adherence cost.
ROI / NPV calculation	No	Focus on return of potential investment and not adherence cost.
Return of Software Quality / Cost of Software Quality calculation	No	To little have been published and conclusions drawn from appliance of them states that they must be modified for every company's process.
Quality / Confidence / Adherence division	Yes	Separates Adherence in a useful way that is easy to display and work with.
Estimation approach including Delphi / WBS and Top-down / Bottom-up approaches	Estimation: Yes Delphi: No WBS: Partly Top-down: No Bottom-up: Yes	Since no complete data over all standard related activities exists, expert estimates is the best choice, based on available resources. The Top-down and WBS approach were rejected in favor for Bottom-up since it would have been more time consuming to build a complete hierarchy. The activities where however divided into manageable pieces for the estimations and related to different areas, partly in appliance with WBS.
Interview approach	Yes	Used to include comments emerged from the estimations. Moreover it is used in the SCM method, which is also applied in the analysis.
Observation approach	No	Not used since the result from it is too general.
Survey approach	No	Required larger personnel resources then available.
Report data collection approach / Measurement approach	As far as possible	All data available was used as part of the analysis since it represent actual facts.
SCM method	Partly	Large parts of this method is of use in the analysis with exception from the measures to collect and the calculations, thus the basis of it will be included. Both the initial simplification of the standard and the in-depth interviews will be applied.
SMARTIE method	Partly	To rely on measurements alone is not possible in RUAG's case since the data does not exist, however the statement to split the standard into smaller components confirms the conclusions made regarding SCM.
VAMOS cost of compliance method	No	Relies on data that does not exist and uses the P-A-F division, which is not applicable in this analysis.

Table 1 Evaluated methods and techniques

Applying the different approaches and techniques, the analysis method used will have seven steps as described in table 2.

The seven steps of CoSCA	
Label	**Description**
(1) Find Standard(s)	Establish which standards are to be assessed and their relevant parts.
(2) Sub-divide	The relevant parts shall be divided into single requirements.
(3) Map to activities	The individual requirements shall be referenced to activities performed in the project.
(4) Evaluate relations	Analyze and evaluate the connections established in (3).
(5) Existing cost data?	Investigate existing cost data for activities. What can be used and to what extent.
(6) Estimate TCV	Estimation of the activites using the Type-Cost-Value (TCV) model.
(7) Calculate CoSC	Calculation of Cost of Standard Compliance (CoSC).

Table 2 The seven steps of CoSCA.

Establish which standards are to be assessed is a given one and the necessity of dividing the standard into more manageable pieces has been addressed. The only thing to add is the level of division necessary and it is simply; that they should be detailed enough to be estimated. In other words the expert must be able to relate actual work to the requirement, to estimate the time spent. For this reason it is recommended to match the requirements to activities which are actually performed in the project. As a fourth step these relations need to be evaluated to ensure that all relevant parts of the standard are represented and if available cost data for the activities needs to be collected. The cost data will then be compared against the estimations to assess the quality of the analysis and to help calculating the cost for standard compliance. The TCV model used in the estimations is presented in chapter 3.1.

4.1 A Type-Cost-Value (TCV) Model

This model was created to separate the different types of activity cost from each other and to determine their contribution. The resulting figures in this case will be in percentage and displaying the relations between the different types of cost, for each activity. Important to notice is the value part of the estimations, since it is the value of the activity costs that are estimated. In other words, if an activity is regarded as valuable the estimations will reflect that, regardless of a high or low cost.

The intended use of this model is in estimations of cost and value for a project's activities. Its size and complexity is constructed with the intent to be repeated for every activity and to be complemented by activity reports presenting man-hours for a given project. Reports with man-hours in focus are often mandatory in projects due to management and the salary aspects. The relation to specific activities however varies and mostly the only relation that can be made is to a collection of activities. For this reason man-hour estimates are included in the model, as a separate step to be estimated before or after the contribution of the different reasons. Man-hours is a typical cost estimation and it can work as a confirmation of the correctness, later on, of all the estimated values since some activities have measured man-hours connected to them. If the reported man-hours are a good match to the estimated man-hours the likelihood that the other estimations are correct increases.

The different types of cost used in this model, that a standard introduces can be separated into three types [FEL09]:

- Costs that adds to the quality of the product.
- Costs that adds to the confidence of the product. Confidence refers to costs that increase the external or internal understanding and trust in the product.
- Adherence costs that are imposed by the standard with no contributing value.

As a minimum these three types have to be included in the model, since the purpose of the entire analysis is to determine the cost of compliance with the ECSS standards. However even if they nicely represent the standard aspect there are many other purposes for an activity that can be relevant to consider, for instance in-house research, legacy or process specifics. At least they must be represented in the result to give the other types of costs a complete basis to relate to. For this reason the model will have four different types of cost to choose from, the three presented above plus one additional, see table 3. The fourth reason was named “Development necessity” in an attempt to reflect its purpose, to cover all parts performed with product development as the primary reason. Furthermore this limitation is due to the fact that there are a lot of activities that have to be assessed and more choices will highly increase the effort.

Reason for an activity
1. Quality adding
2. Adherence to standard
3. Confidence adding
4. Development necessity

Table 3 Type-Cost table

Regarding value, as a part of the model name, it refers to the contribution of a given activity. It is from the estimated values that the percentage displaying the relation between the types in table 3 is calculated. For this part the one hundred dollar method [HAT08] is introduced and using that the experts subjected to estimation interviews will be asked to split one hundred dollar on the four reasons of an activity, based on their impact. This approach has been use with success on previous occasions [REG01] and as a prioritization technique it is regarded as quick and easy to perform and well suitable when the requirement or activities are more detailed [HAT08].

Applying this model, table 4 shall be completed for every project activity and summarized to give a percentage of cost of compliance with standards connected to the project. Important to specify before beginning the estimations, is the project to which the expert shall relate in the assessment. A possible introduction can be; “For development activity A and the four activity types below, to what degree would you distribute 100 dollar to the different types of reasons for which the activity was conducted in project P”.

TCV Estimation Form	
Activity	
Man-hours	
Reason	***Value (100*** **total*****)***
1. Quality adding	
2. Adherence to standard	
3. Confidence adding	
4. Development necessity	

Table 4 Final TCV activity estimation form

In the end of the interview an estimation of the total time for the project in question shall be estimated, to be compared against the actual figure and help in calculating the adherence, quality and confidence cost for the entire project. The total time spent in the project is necessary to include since the activities being estimated do not represent all the activities in the project, only those with a connection to the standard. Moreover a notation shall be made for every activity considered hard to estimate by the expert, including the reason why. By so doing it will later on be easier to evaluate possible differences in the result.

A static pre-pilot-evaluation of the model has been performed to ensure understanding and workability by assessing every step of the potential estimation approach and following that a pilot-evaluation was conducted before the actual estimations.

4.1.1 Pilot-evaluation

A pilot run on five random activities was performed with a RUAG employee, to evaluate the TCV model approach. The goal of the evaluation was to determine if the TCV activity estimation form (table 4) is understandable and usable and moreover if the interview approach to the estimation process is a good choice. It resulted in some modifications and requests for clarification however the assessment of using expert estimations as the primary input was good. The largest change was the removal of a fifth reason from the TCV activity estimation form (table 4), where it initially was included as a type of cost. It was named “Other reasons” with the intention to capture reasons that could not be linked to the other four types, but it was deemed to be too abstract and was removed. Instead the subject shall be told in the beginning of the interview that if neither of the reasons matches, a separate reason for that specific activity shall be stated. Furthermore reason one and three in the TCV estimation form were hard to separate for the one subjected to the estimation and a more detailed definition became necessary. For this reason all the remaining reasons where defined as shown in table 5.

TCV Reason Definition	
	Man-hours
	The total time spent on an activity, including updates of documentation when performed.
	Reason
1.	**Quality adding**
	An activity that increase the quality of the actual software, for example by increasing the time between failures or eliciting necessary functions.
2.	**Adherence to standard**
	An activity performed only because the standard requires it.
3.	**Confidence adding**
	An activity that increase the customer's confidence in the software, by giving them insight in the progress and helping them to understand the level of quality. Moreover an activity that add to an internal (within RUAG) confidence, by increasing the general understanding of the software and confirming a level of quality.
4.	**Development necessity**
	An activity that is necessary to perform to be able to develop the software.

Table 5 Final reason definition

The use of the 100-dollar method for the estimations was also a success. It was easily applied and understood by the interview subject and changing the amount to a higher or lower figure was never necessary to discuss. Together the modifications and clarifications resulted in an update of the TCV activity estimation form and a better understanding of how to approach the subject.

5 CASE STUDY - APPLYING THE CoSCA METHOD

To use the constructed TCV model as intended the first step was to select estimation as the mean of information. The choice was complemented with the decision to use man-hour measurements from a RUAG project for comparisons as far as possible to increase the quality of the study. Following that the first step in the CoSCA method is to establish which standards to include in the analysis and for this the ECSS-E-40B and ECSS-Q80B standards are chosen.

As said the ECSS standards and their requirements become the focus after the general approach was established. The standards had to be broken down and sorted in a logical way for the estimations. To start with they were digested into single requirements which could have taken an extreme amount of time if it was not for their internal hierarchy and identification numbers. When trying to divide the requirements into more manageable settings the first attempt was to split it based on the required reviews presented. The reviews required by ECSS are

- Software Requirement Review (SRR),
- Preliminary Design Review (PDR),
- Critical Design Review (CDR),
- Qualification Review (QR),
- Acceptance Review (AR),
- Operational Readiness Review (ORR)

and all except AR and ORR are relevant for RUAG's software section. For each review the standard requirements in turn were divided with respect to a general file belonging. This resulted in a division where many of the requirements were sorted into more than one review and by so making the estimation and measurement matching more difficult. The varied range of the subject of the requirements in a specific file was also too wide to get a good overall picture. To solve these problems a second attempt to sort the requirements was made. Instead of dividing based on reviews the requirements were divided based on their relation to documents, which in turn were required by the standards. That way the overview become much more lucid and no requirement doublets were made with a few well documented exceptions. What was not possible though was the single minded document connection. Instead a combination of basic requirement connection and document connection was introduced since far from every requirement can be related to a document in a direct way. Documents along with individual requirements were then allocated to the activities as visualized in figure 7.

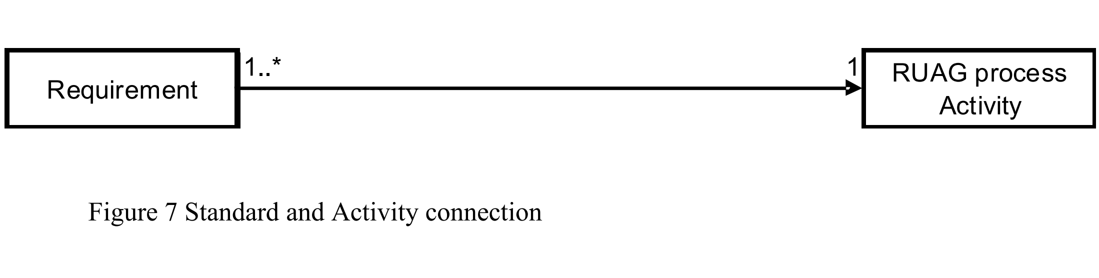

Figure 7 Standard and Activity connection

To find activities used in development, RUAG's process documentation were analyzed and the result contemporized based on information from informal interviews with key personnel. This was believed to be the difficult part and in some cases it were, however RUAG has adapted their entire process to the ECSS standard so well that many connections were obvious. The constructed table of activities with ECSS relations was then evaluated and corrected together with a RUAG responsible. The final result can be viewed in appendix A.

All the activities in turn were divided based on section belonging and then person responsible. Beyond the software section, part of the system development section is involved and validation is performed by a yet another section.

Selection of appropriate projects for analyze of man-hour measurements followed the activity and standard matching. The measurements shall as mentioned be used as a quality confirmation of the estimations and therefore is it essential to choose projects with detailed activity reports. Another condition to include in the selection process is the up-to-date factor. If the projects are too old the measurements would most likely not be accurate since the process has evolved and experience has changed. Based on these conditions and the choice of experts, two projects at RUAG were proposed, where one was considered as small and the other one as large. Moreover two experts were chosen, one from each of the projects, and referred to as expert A and expert B. The measurements were derived from some of the projects' report documents and allocated to related activities or a collection of activities when needed. The reason for not conducting the analysis on one single project was the different backgrounds of the experts available. The interview subjects were selected based on availability and knowledge with the first as the largest factor. For both of the interviews the subject in question was or had been an object manager with insight into all the activities even those a little outside the software section's scoop.

Before the actual estimations a pilot run on five random activities was performed as described in previous chapter. Next step was to apply the TCV model and conduct two estimation interviews with key personnel at RUAG. One day prior too each estimation interview the TCV estimation forms for all the activities were remitted to the expert as recommend [FRY03] to allow more carefully prepared answers.

After the estimations the data was analyzed and the final result presented in tables showing the percentage for the different type of costs, see table 9 and appendix A. A comparison between the measured and the estimated man-hours, in total, confirmed a high quality of the data. Moreover a comparison of the inputs from the two interviews was performed and the probability of correct results was assessed as high since the answers were fairly similar. However some deviations between the estimates exist and a third estimation interview was conducted, with an expert C and expert D, to limit the margin of error. Expert C and D split the effort and completed the third estimation together and they based their estimations on a normal sized project. The existing deviations mainly concerned the estimated values regarding the different reasons behind the activities and by adding a third party an acceptable [BIT01] average could be calculated. Differences in man-hours did only to a smaller extent need further assessment. Depending on estimation and process experience, differences in the estimations were expected and therefore some additional work was expected and planned. However the details were not decided beforehand since there was no way of knowing the outcome and the extent of the deviations.

Using the relation between the activities as a guide, the measurements without an activity belonging were sorted and together with the estimates an average for every activity in man-hours was calculated. Finally the cost for complying with ECSS was calculated with reference to different types of cost presented in table 3. The entire process described above is summarized in figure 8.

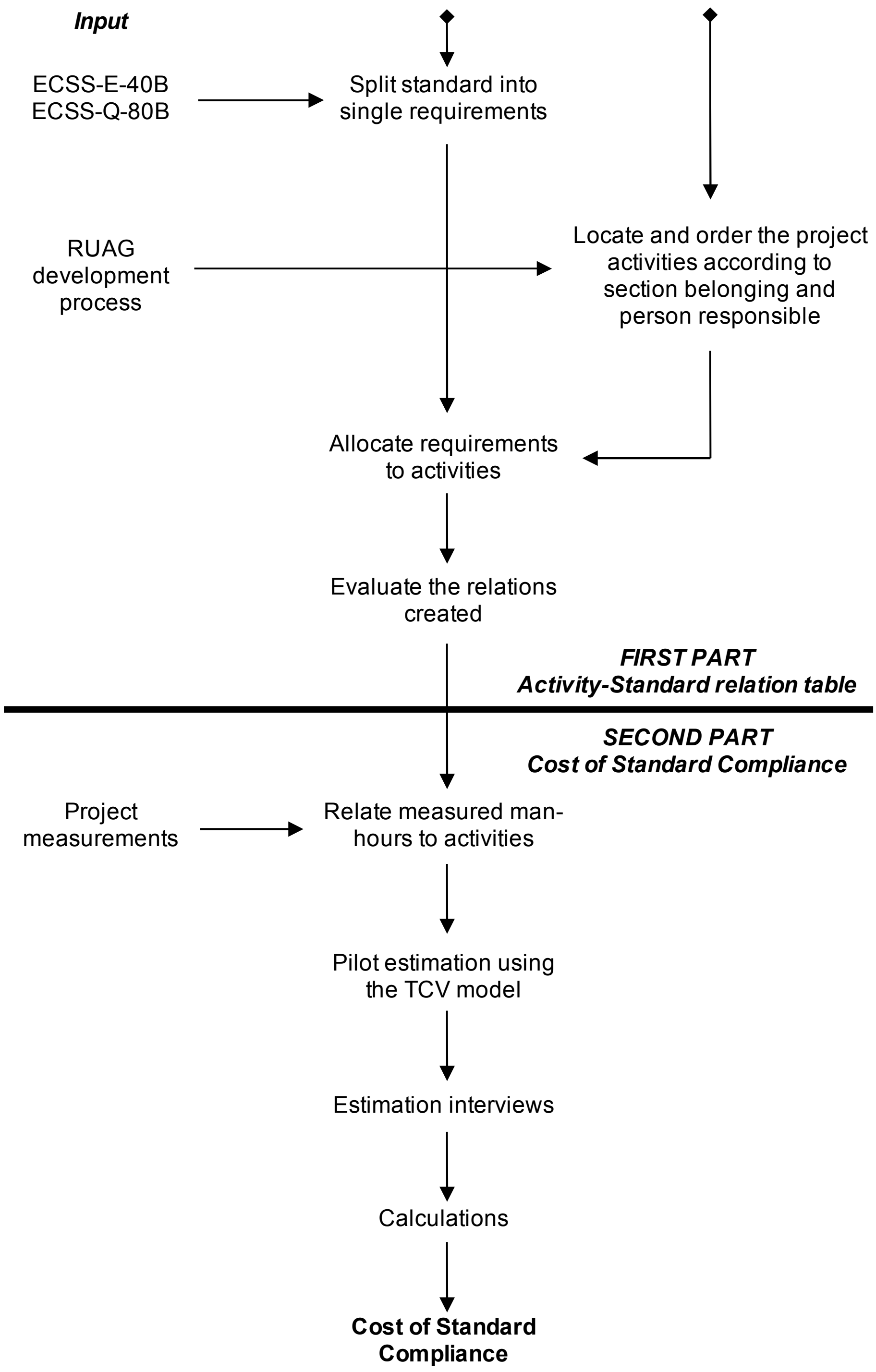

Figure 8 Cost of Standard Compliance process

5.1 Lesson learned

As said some unnecessary work dividing the standard based on reviews were made in the beginning and some effort could have been saved by reversing the matching process and start with the development process instead. This most likely would have led to an earlier choosing of the process-document-requirement approach. The lesson learned in this case is that if the process to match the standard against is already using the standard in question, the adaption process itself most likely will alter the process in such a way that the standard can be referenced in it. That said step four in the CoSCA method is still believed to be necessary in future assessment. The evaluation of the connections between activity and standard requirements became a part of the trial and error division attempts but it was still conducted.

6 RESULT

As mentioned there were two sets of initial estimations made for all the activities related to the ECSS standards and they were later on complemented with additional estimates for parts where significant differences existed between the initial estimates. However noticeable is the high level of similar estimates done by the different experts, concerning adherence cost, which largely exceeds the initial expectations. This concerns both man-hours and purpose estimates and with the complemented estimates the result is therefore deemed to be of high quality.

Important to take into consideration in this case is the fact that the activity list does not represent all the activities performed during a project at RUAG. It simply states all the activities that are relevant when calculation cost of compliance, therefore when calculating the percentage of cost of compliance an estimate of the total project time is used. To explain in figures the reason "adherence to standard", from the final results, is used as an example.

$$(1)\ 8147 / (2)\ 10650 = X$$

$$X * (3)\ 21{,}7\% = 17\%$$

The resulting total of man-hours from the estimations (1) was divided by the estimated total project time (2) and this figure multiplied with the initial percentage of the selected reason (3). The initial percentage represents all the estimated activities while the new one represents all activities performed in a project. This is summarized in formula 2.

$$\frac{\text{Estimated man-hours}}{\text{Total project time}} \times \text{Initial percentage}$$

Formula 2 Percentage calculation

The resulting differences between the initial estimates can be view in table 6. It shows the percentage of the different reasons calculated from the activities included in the estimation. Of the 115 activities assessed in the analysis the percentages displayed hint at the real reason why they are conducted and the man-hours are the estimated total of these activities.

Initial estimation results

Reason	Small Project	Large Project
Quality adding	41%	13%
Adherence to standard	20%	24%
Confidence adding	5%	20%
Development necessity	34%	43%
Man-hours in total	6150h	9500h

Table 6 Initial estimation results

The deviation between the estimates in table 6 can seem like a lot especially between the reason quality adding and confidence adding and to a larger extent it depends on interpretation of the reason. Quality and confidence is closely linked and their relation to an activity can be hard to estimate. One of the experts interpreted confidence as something that is often a part of a quality activity and therefore deserved part of the quality percentage. The other interpretation was that every quality activity which does not give a physically contribution is to be considered confidence adding. No one of them are wrong according to the reason definition in table 5, thus a lesson learned is to be even more thorough in the definitions and explain by example before the estimations. To solve the problem by combining them into a single reason would not be beneficial even though the adherence to standard cost is of most interest the differentiation between quality and confidence are important since both of them are needed. Regarding the remaining reasons their definitions were afterward described as clearly stated and the resulting figures concludes the statement.

The complementary estimates effectively helped in closing the gap between the initial estimations and mainly with regard to differences in reason. For every single activity the need for an additional estimation was assessed and then performed if necessary. The estimations were, as described, divided between two experts C and D, where one's area of expertise was quality and where the other one had a more general knowledge basis. Activities related to quality and confidence with differences in their initial estimations was given a strong additional input and the reasons become easier to differentiate.

In the initial estimates the largest deviations concerned man-hour estimates for design, detailed design and production of software unit tests. However since the size of the two projects included in the estimations were different a higher figure was to be expected from the larger project. That said a high figure in man-hours can have an unwanted impact on the final calculations and to minimize the margin of error they were included. Regarding reason assessment the largest differences were found in activities regarding software documentation and code verification reports along with the production of software unit tests. All-in-all 16 activities were assessed as absolutely necessary to be included in the complementary estimations (see table 7) and out of these, 14 were related to reason estimates. The differences in interpretation of the reason regarding quality and confidence related activities were evaluated prior to the third estimation round and based on that the related activities were included for the additional assessment. Out of a total of 115 activities 54 were estimated a third time with more focus on how to differentiate between quality and confidence The initial intention after the first two analyses were to estimate all the activities a third time, however the availability of experts limited the estimations as described above.

For a few of all the estimations performed difficulties or hesitation in the expert's assessment was noted, foremost concerning man-hours. Depending on the level of uncertainty the final calculations were made with favor to the other estimations performed for that activity.

Activity	Reason for a third assessment	Result
Software criticality analyses report: - SW dependability and safety analysis - List of critical software components	Foremost differences in reason assessment but with regard to the man-hour estimate as well.	No answer was given and an average was calculated based on initial estimates.
Make sure that review decisions are implemented.	Differences in reason assessment	Close match to one of the initial assessments resulting in a calculation with favor to them.
TRR - Test Readiness Review	Differences in reason assessment	No answer was given and an average was calculated resulting in close to a draw between the reasons.
SW Verification Report (SVR) - Software architectural design & interfaces verification report	Differences in reason and man-hours	Helped in closing the gap between the initial estimates and an average from all three assessments were calculated.
SW Verification Report (SVR) - Software documentation verification report	Differences in reason assessment	Helped in closing the gap between the initial estimates and an average from all three assessments were calculated.
SW Verification Report (SVR) - Software code verification report (Code inspection)	Differences in reason assessment	Close match to one of the initial assessments resulting in a calculation with favor to them.
Detailed design verification as a part of code inspection: SW detailed design verification report	Differences in reason assessment	Close match to one of the initial assessments resulting in a calculation with favor to them.
SW Unit Test Plan (SUTP)	Differences in reason assessment	Helped in closing the gap between the initial estimates and an average from all three assessments were calculated.
Produce SW unit tests	Differences in man-hours	Helped in closing the gap between the initial estimates and an average from all three assessments were calculated.
Perform SW unit testing: SW Unit Test Report (SUTR)	Differences in reason assessment	Helped in closing the gap between the initial estimates and an average from all three assessments were calculated.
For every change due to noncompliance to requirements: SW Problem Report (SPR)	Foremost differences in man-hour estimates but with regard to reason assessment as well.	Helped in closing the gap between the initial estimates and an average from all three assessments were calculated.
SW-QR - Software Quality Review	Differences in man-hours	No answer was given and an average was calculated based on initial estimates.
Implement validation scripts	Differences in reason assessment	Helped in closing the gap between the initial estimates and an average from all three assessments were calculated.
Verifying that the work performed by the verification team follows the approach laid out in the SValP, and that the produced tests are adequate for testing the SW.	Differences in reason and man-hours	Close match to one of the initial assessments resulting in a calculation with favor to them.
Ensure that suitable SW PA provisions are included in the contract and review of all changes to the contractual requirements concerning SW PA issues	Differences in reason and man-hours	Helped in closing the gap between the initial estimates and an average from all three assessments were calculated.
SW Product Assurance Report (SPAR)	Differences in reason and man-hours	Close match to one of the initial assessments resulting in a calculation with favor to them.

Table 7 Activities necessary to estimate a third time

From the complementing estimations regarding quality activities the calculations showed that 95% of them were close matches to the initial estimations which increase the confidence for the remaining activities together with the ones assessed.

The top 10 candidate activities for adherence to standard cost are presented in table 8. All of these activities have a 40 % adherence or higher related to them, meaning that 40 % or more of the activity are performed because the standards require it. With exception of activities with 100 % adherence, the activities shall not in any way be completely removed from the development process if tailoring of the standards allows it. However larger parts of them are not contributing any value to the development and if there is no contribution the recommendation is to limit the time spent on those parts. The figures presented for number nine in the list should be regarded with care since larger differences in the estimations exist and it resulted in a rough average, see table 7. For the other activities the experts agreed in their assessment with one additional exception concerning man-hours for SW-QR (number two). However the deviation was not large and the average calculated can be used without further assessment.

Top 10 adherence to standard cost activties

	Activity	% of adherence	Unnecessary time spent
1.	SW Requirements Specification (SRS)	50%	**125h**
2.	SW Quality Review (SW-QR)	90%	**81h**
3.	SW Integration Test and SITR	70%	**70h**
4.	SW Product Assurance Report (SPAR)	40%	**68h**
5.	SW System Specification (SSS)	40%	**60h**
6.	SW Reuse File (SRF)	55%	**50h**
7.	Validation testing w r t RB	75%	**45h**
8.	Validation report evaluation w r t RB	100%	**20h**
9.	SW criticality analyses report	51%	**13h**
10.	SW Integration Test Plan (SITP)	55%	**11h**

Table 8 Top 10 adherence to standard cost activities

The hours presented in table 8 are an average calculated from estimations and measurements in combination, as explained in the methodology. The differences between projects are also taken into consideration and an average time for completing a project is calculated in the same fashion. For the activities in table 8 a total of 543 hours is spend without contributing to the process because the standards require it.

A contradiction to one of the activities has also been proposed [AHM09] based on a survey stating that if RUAG was not dependent on ECSS, they would like to spend more time with integration testing. This is a direct opposite to number three in table 8, which shows that 70 % of the activity is conducted simply because ECSS requires it. One possibility of a united front is if the survey result focuses on the actual integration testing and the 70 % adherence focuses on the Software Integration Test Report (SITR). In both cases this is a possible interpretation, however in a post-interview it was stated by example that SITR is relatively cheap and the unnecessary cost refers to the integration testing. This said the earlier assessment is proven wrong since larger parts of RUAG's software development develop separate drivers and not complete products. Integration between the drivers is not common and for that reason integration testing would have been approach in a more effective way for the parts actually integrated, if the standard allowed it.

The post-interview mentioned was conducted to gather examples for the activities in table 8 in an attempt to give more specific improvement proposals. The examples are not validated in any additional interviews and must be regarded as possible reasons or part thereof and not used as facts. They are presented below with numbering referring to the activities in table 8 and can be used as input to discussions concerning the related activities.

(1) The level of the requirements in SRS and SSS are unclear and it results in two similar documents. Some double requirements can also be made between SRS and the software interface control document (SWICD),
(2) During the SW-QR the same tests conducted during TS is performed again.
(3) In software development RUAG usually develops separate drivers for different purposes and not complete products and for most of the low-level drivers there is no integration.
(4) Some of the data reported in SPAR have already been reported in other documents. For instance metrics collected during code-inspection, which are reported in the code-inspection report.
(5) The requirements are to low-level. Connected to number (1).
(6) Most of the results presented in the SRF are already known since all the code to be reused has been developed in-house.
(7) Testing with regard to TS, which predates RB, are conducted on real hardware and hence no new aspects are presented in RB.
(8) The specific analysis has already been conducted and many of the tests are automated.
(9) The customer gives the criticality.
(10) The same as number (3).

Regarding the entire development process and all activities enforced by the standards, the percentage representing the different reasons are presented in table 9. As an example, based on activities required by ECSS; the reason for carrying out 17 % of the total time spent in a project, conducted at RUAG, are to increase the quality of the product.

Final results	
Reason	**Results**
Quality adding	17%
Adherence to standard	17%
Confidence adding	12%
Development necessity	31%
Man-hours in total	8147h
Total project time	10650h

Table 9 Final results

The percentages are calculated according to formula 2 and are a representation of the reasons behind the activities related to software development, enforced by the ECSS standards.

77% of the total time spent in a project is represented in table 9. However the remaining 23% can affect all of the figures except the adherence to standard percentage since all related activities regarding it, have been accounted for[1]. One way to create a total of 100% could be to simply increase the percentage for the "development necessity" reason. Most likely the majority of the remaining hours can be linked to it however not all of them. Quality for instance, is of great importance in development today and to assume that the ECSS standards' aspect is the only quality relation, would be a grave mistake. Furthermore the goal with this analysis is not to elicit the reason behind all the activities performed in a RUAG project. The focus shall be on the ECSS relation and for that reason the best way to present the result is by limit it to the activities presented in appendix A.

In general ECSS is regarded as something positive at RUAG even though some parts are referred to as a waste of time. The use of both SSS and SRS, and tests performed with regard to them are two examples, as presented in relation to table 8. An earlier evaluation [FEL09] of the experience with ECSS separates the positive and negative aspects as quality and software development efforts. At RUAG and one additional company in the European space industry included in a study [FEL09], they think that ECSS positively affects the quality but have a somewhat negative effect on costs related to software development.

The total time for conducting this analysis using the proposed method is estimated to approximately three weeks, as presented in table 10. A re-assessment of ECSS-E-40 and ECSS-Q-80 at RUAG or any other company that can relate to the activities presented in appendix A, will be a significant faster process. The first three steps in table 2 can then be disregarded and the analysis can be performed by conducting step 4-7.

[1] To the best of knowledge.

Estimated time for CoSCA.	
Steps	**Estimated time**
(1) Find Standard(s)	1 hour - 1 day
(2) Sub-divide	3 days - 1 week
(3) Map to activities	1 week +
(4) Evaluate relations	1 -2 day(s)
(5) Existing cost data?	1 -2 day(s)
(6) Estimate TCV	½ day per estimation
(7) Calculate CoSC	1 - 2 day(s)

Table 10 Estimated times for CoSCA.

Important to take into consideration when applying the estimates in table 10 is that they are based on some prior knowledge and experience with the standard(s) in question and a familiarity with the company's development process.

6.1 Resulting consequences

Some activities connected to specific parts of the standards have been identified for potential improvements. The exact elements of those activities have not been addressed and are left to future work. However a proposal for future assessments, using the CoSCA method, is to ask for examples when an activity has more than 25 % adherence cost attached to it. This will strengthen the figures presented and give the reader a better idea of where and how to focus. In the presented case study it was only conducted on the top 10 adherence activities, but it was well received by company representatives and improvement propositions were easier accepted and understood.

Some hints have been noted regarding limitations in the tailoring of ECSS. In [ESA09c] it is stated that modifications and removal of ECSS requirements are allowed if justified. If the project's stakeholder(s) and the company have a good relation and are confident that they will deliver some deviations have been allowed [AHM09]. If the stakeholders on the other hand are limited by technical and general knowledge they can require ECSS applicability without tailoring for any single ECSS requirement [AHM09]. Revision c of the ECSS standards was published during 2009 and some updates have been made concerning software development. Regarding tailoring possibilities not much have changed and the updates appears to be even more detailed than before. It have been stated [AHM09] that the detailed requirements on documentation specifics and the necessity to prove that the company is compliant with ECSS takes resources away from other activities and since the latest update of ECSS is even more demanding the cost of compliance can be expected to increase as well.

7 DISCUSSION

The differences in the result from the two initial estimations were as mentioned, complemented by an additional estimation interview. However even if an average from three independent sources were calculated the deviations still exist and has to be addressed. In a first analysis of the result it has been shown that all the activities related to the deviations have a small impact on the final result and they can all be related to differences in experience. The deviation between the results can be considered small and in general the time spent on the activities can be considered proportional to the project's size. In other words the percentual figures calculated remains the same regardless of size of the project, since the relation between the estimated hours for most of the activities follows the relation between projects' size.

Regarding strength and weaknesses of the chosen approach the choice of using experts to estimate the result can be regarded as both. If the activity tables are in place the effort of conducting the analysis is fairly small in contrast with an introduction of measurement tools in development. The foremost weakness with relying on estimations is that they are only as good as the expert. Measurements are preferable when available however noticeable is the high level of similar estimates done by the different experts. A weakness with the method worth mentioning is the matching between activities and requirements. In an initial assessment it was noted as a major risk, saying it might be too hard to find a relation for all the requirements. However it did not become an issue since the process itself have been altered to meet the standards and the lesson learned in this case must be that if the process to match the standard against is already using the standard in question. The adaption process itself would most likely have altered the process in such a way that the standard can be referenced in it. In cases like this, when the process used is heavily adapted to the standard to be investigated it might be easier to start with the process itself and map it to the standard rather than the other way around. This may not be the case for other industries and if no clear relations can be made the matching process will be much more complex.

The calculated result presented in the case study is considered to be representative for most of RUAG's project and using estimates based on different projects of somewhat different size, a confidence interval of 15,5-20 %, with confidence level 90 %, could be calculated. This confidence interval represents the "adherence to standard" cost and the varied results from the initial estimations will not change that since the same relation between the different types of cost are used for the calculation of the minimum as well as the maximum value. The variable is the estimated man-hours and hence their effect on the costs.

As said some unnecessary work dividing the standard based on reviews were made in the beginning and some effort might have been saved by reversing the matching process and start with the development process instead. This would most likely have led to an earlier choosing of the process-document-requirement approach displayed in figure 7.

The need to relate to a specific project in the estimation was not initially specified and the unspoken intention was to have a more "general" project as reference. However after the first estimation the subject had unintentionally estimated based on a specific project and a "general" or "normal" project was categorized as to abstract. For that reason the clause to relate the estimations to specific projects was added to the model. In this case it resulted in two different project relations for the initial interviews and a third for the additional estimations. The upside to include multiple projects in the analysis is that it is possible to make conclusions regarding how well the results represent different projects. As a downside it allows less validation with a comparable approach and the end result might have been non-conclusive if it was not for the similar results reported despite different projects. The entire method is developed with one project in mind but can as well be applied for multiple projects within a company if necessary. When the result was analyzed it showed that the reason behind an activity was similar independent of project and that experience and knowledge were larger influences. The recommendation is however to focus the analysis on one project and choose experts within that project for the estimations. This simplifies the calculations and the analysis does not need to include any aspects of process differences between the projects, if the estimations are not conformable.

The estimated time to perform a similar analysis is presented in table 10 with the prerequisite of some prior knowledge and experience of the standard and the development process in question. Some prior knowledge and experience refers to an actual participation of a similar project restricted by the standard being assessed. For an external part to conduct the same analysis the time would increase significantly to perhaps two or three times the estimations for step 2-5. The other steps would remain relatively the same. If a person with extensive experience and knowledge in the relevant process and standard where to perform the analysis the time could not be expected to decrease by half since all steps are sill necessary, however the time spent on step 2-5 would be less.

In discussions concerning the SMARTIE approach an extremely low figure was presented referring to an objective assessment of conformance and that for many of the requirements only a subjective assessment was possible. This caused some initial concern however the cost and estimation approach used in the CoSCA method limit the problem since they are focused on reasons behind activities performed. The standard is not assessed to see if it should be implemented but to separate and locate the different types of cost introduce by it.

Using an experience-based approach has been motivated in chapter two and has been stated as useful [BOE00] in the absence of a complete collection of data to analyze. The obvious drawbacks of the estimations have also been addressed since they only are as good as the expert and years of experience does not necessary mean high estimation skills [BOE00]. Asking the same expert to estimate the same number of activities more than once may result in as many different answers as times they are asked. Moreover one or even two experts may guess wrong for the same activity but the likelihood for them to give the same wrong answer is not high. For that reason some of the estimations made might be wrong but the total figure must represent a true picture of the cost deviation.

7.1 Possible improvements to the TCV model

The TCV model was used successfully in the performed estimation interviews and the general idea has been proven to work. Improvements though are always possible and the most obvious one is the re-definition of the quality and confidence reasons, which was necessary to limit the number of interpretations that was made.

Another possibility is to split confidence into internal and external “confidence adding” and estimate with five different types of cost instead of four. This would enable a differentiation between stakeholder and the company and might be useful when dealing with customers with varied confidence in the company.

Interpretation of a standard presents another problem since they can be translated differently by different people and organizations [AHM09]. A company may state that they follow a specific standard while the customer interprets it in a second way which would result in noncompliance for the company. At RUAG new reviewers from the customer for the different reviews have resulted in a lot of rework due to interpretation of what the ECSS standards requires [AHM09]. If the experts have different views of the standard their estimations may differ and conclusions may be hard to make. It might be a good idea to add some general questions regarding the standard, the company process and the expert's experience to the initial proceedings of the TCV model to be used in the assessment and calculation of the final results.

Another part that can be added to the initial proceedings of TCV is the aspect of noncompliance [EME00] and how the project or company handles a notice of violation (NOV). The word NOV may not be used but in some way noncompliance with the standards must be handled and how that is managed affects the administrative cost [EME00].

8 CONCLUSION

This paper present a new method for calculating the cost of complying with a software development standard in a project that is or has had it implemented and used. It is based on a simple model of the type of value and associated cost of the development activities of a company. Moreover the primary source of information is interviews with experts in the associated project and estimations done during them, regarding reason and man-hours. An initial investigation at a company in the space industry showed that the method is practical and usable with an acceptable level of effort.

The initial investigation involved a static, pre-pilot assessment of the method and its forms, a pilot evaluation, followed by two full estimation interviews. The estimated man hours per activity was complemented with actual data from the company's project management system. Furthermore a third estimation interview was performed for the activities with a larger deviation between the two estimators, together with all activities related to quality and confidence of the product. Overall there was a high level of agreement in the estimations which was unexpected due to differences in experience and mostly because the estimators based their assessments on different projects. The agreement was seen both in estimated man-hours and in the estimated level of contribution of an activity. With the complemented estimates the result is therefore deemed to be reliable. From the complementing estimations regarding quality activities the calculations showed that 95% of them were close matches to the initial estimations which increase the confidence for the remaining activities together with the ones assessed. All-in-all this points to the practical usability of the method and presents possibilities to apply it in multiple projects and not just at RUAG Space.

Important to take into consideration in this case is the fact that the activity list does not represent all the activities performed during a project at RUAG. It simply states all the activities that are relevant when calculation cost of compliance, therefore when calculating the percentage of cost of compliance an estimate of the total project time need to be used. Based on these estimations it is shown that 17% of a project was spent on efforts to adhere to the ECSS standard that did not add any value to the developed software product, neither by adding quality to the product itself or by adding confidence that a certain level of quality has been attained. Based on comparison with other projects a confidence interval of 15,5-20% can be estimated.

Time to conduct a similar analysis in a project can be estimated to approximately three weeks, if it is performed by someone with prior knowledge and experience in the standard and the development process used within the project. Most of the time in the analysis are spent by sub-dividing the standard and connect the resulting parts to activities performed in the project, thus the level of knowledge and experience have a large impact on the total time needed.

9 FUTURE WORK

For future work a more detailed analysis of the presented candidates for potential improvements needs to be made in an effort to locate the specific parts of an activity that can be optimized and tailored. Moreover the CoSCA does not address how it can be used to monitor the effect of efforts made to minimize the costs. A comparison between analyses performed over different periods of time can of course reveal this, but to get more frequent updates an approach needs to be established.

Another task that is left to future work is how the argumentation with customer and ESA can be constructed, to motivate certain tailoring of ECSS and to make them aware of the adherence cost.

With regard to VAMOS the next step is to test it in a real industrial environment and if the effects ECSS have on the process is to be included, a CoSCA is necessary. The data already exists if it is introduced at RUAG's software department and a complete table of the final activity estimations is presented in appendix A. When VAMOS has been introduced the result needs to be combined with the CoSCA results and by so doing locate potential optimization possibilities.

10 References

[AHM09] E. Ahmad and B. Raza, "Towards Optimization of Software V &V Activities in the Space Industry," *Blekinge Institute of Technology,* Jan. 2009. [Online]. Available: http://www.bth.se/fou/cuppsats.nsf/all/27c723432e8963a8c1257 5680034d427/$file/Thesis-MSE-2009-02.pdf [Accessed: Oct. 5, 2009].

[ALO05] O. Aloui and L. Kenny, "The Cost of Compliance with SPS Standards for Moroccan Exports: A Case Study," *Agriculture and Rural Development,* 2005.

[BIT01] K.Bitner, B. Thomson and B. Chwirka, "The cost of compliance with a new drinking water standard for arsenic in New Mexico," *New Mexico Geology,* pp. 10-12, Feb. 2005.

[BOE00] B. Bohem, C. Abts, and S. Chulani, "Software Development Cost Estimation Approaches – A Servey," *Annals of Software Engineering,* vol. 10, no. 1-4, pp. 177-205, Nov. 2000.

[EMA00] K. Emam and I. Garro, "Estimating the extent of standards use: the case of ISO/IEC 15504," *The Journal of Systems and Software,* vol. 53, pp. 137-143, 2000.

[EME00] R. Emery, M. Charlton and J.L. Mathis, "Estimating the Administrative Cost of Regulatory Noncompliance: A pilot Method for Quantifying the Value of Prevention," *The Radiation Protection Journal,* vol. 78, no. 2, pp. 40-47, May 2000.

[EMM97] W. Emmerich, A. Finkelstein, C. Montangero, and R. Stevens, "Standards compliant software development," *International Conference on Software Engineering Workshop on Living with Inconsistency, (IEEE CS)*, pp. 1-8, 1997.

[EMM99] W. Emmerich, A. Finkelstein, C. Montangero, S. Antonelli, S. Armitage, and R. Stevens, "Managing standards compliance," *Software Engineering, IEEE Transactions on*, vol. 25, no. 6, pp. 836-851, 1999.

[ESA03A] ECSS STD ECSS-E-40 Part 1B, "Software Part 1: Principle and requirements," ESA-ESTEC, Requirements & Standards Division, 2003.

[ESA03B] ECSS STD ECSS-E-40 Part 2B, "Software Part 2: Document requirement definitions (DRDs)," ESA-ESTEC, Requirements & Standards Division, 2003.

[ESA03C] ECSS STD ECSS-Q-80B, "Software product assurance," ESA-ESTEC, Requirements & Standards Division, 2003.

[ESA09A] ECSS STD ECSS-E-ST-40C, "Space engineering: Software," ESA-ESTEC, Requirements & Standards Division, 2009.

[ESA09B] ECSS STD ECSS-Q-ST-80C, "Space product assurance: Software product assurance," ESA-ESTEC, Requirements & Standards Division, 2009.

[ESA09c] ECSS STD ECSS-S-ST-00C, "ECSS System: Description, implementation and general requirements," ESA-ESTEC, Requirements & Standards Division, 2009.

[FEL94] M. Feldman, "What is the cost of compliance?," *The Journal of the American Dental Association*, vol. 125, no. 6, pp. 682-686, 1994.

[FEL09] R. Feldt, E. Ahmad, B.Raza, E. Hult and T. Nordebäck, "Evolving the ECSS standards and their Use: Experience based on Industrial Case Studies," *Data Systems in Aerospace*, 2009, in submission.

[FEN93] N. Fenton and S. Page, "Towards the evaluation of software engineering standards," in *Software Engineering Standards Symposium, Proceedings.*, pp. 100-107, 1993.

[FEN98] N. E. Fenton and M. Neil, "A strategy for improving safety related software engineering standards," *Software Engineering, IEEE Transactions on*, vol. 24, no. 11, pp. 1002-1013, 1998.

[FOS96] T. Foster, "An examination of the relationship between conformance and quality-related costs," *International Journal of Quality & Reliability Management,* vol. 13, no. 4, pp. 50-63, 1996.

[FRI05] T. Friedman, The World is Flat: A Brief History of the Twenty-First Century, Farrar, Straus & Giroux, 2005.

[FRY03] G.Frykmer and B. Liedén, "Bedömningar av IT-investeringar inom offentliga sektorn," *Handelshögskolan vid Göteborgs Universitet: Institutionen för Informatik,* Oct. 2003.

[HAR99] W. Harrington, R. Morgenstern and P. Nelson, "On the Accuracy of Regulatory Cost Estimates," *Resources for the Future,* Jan. 1999.

[HAT08] S. Hatton, "Choosing the “Right” Prioritisation Method," *Australian Conference on Software Engineering,* vol 19, pp. 517-526, 2008.

[HEE92] F.J. Heemstra, "Software cost estimation," *Information and Software Technology,* vol.34, no. 10, pp. 627-639, Oct. 1992.

[JON97] M. Jones, U.K. Mortensen and J. Fairclough, "The ESA Software Engineering Standards: Past, Present and Future," *3rd International Software Engineering Standards Symposium (ISESS '97),* pp.119-126, 1997.

[MAS05] K. Maskus, T. Otsuki and J. Wilson, "The Cost of Compliance with Product Standards for Firms in Developing Countries: An Econometric Study," *World Bank Policy Research Working Paper 3590,* May. 2005.

[MUL07] S. Müller and C. Supatgiat, "A quantitative optimization model for dynamic risk-based compliance management," *IBM J. RES. & DEV,* vol. 51, no. 3/4, pp. May/July 2007.

[PFL94] S. L. Pfleeger, N. Fenton, and S. Page, "Evaluating software engineering standards," *Computer*, vol. 27, no. 9, pp. 71-79, 1994.

[PRE09] P. Preissing, "A Software V&V Framework for the Space Industry," *Technische Universität München*, 2009, in submission.

[REG01] B. Regnell, M.Höst, J. Natt och Dag, P. Beremark and T. Hjelm, "An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software," *Requirement Engineering,* vol. 6, pp. 51-62, 2001.

[SCH06] A. Schiffauerova and V. Thomson, "A review of research on cost of quality models and best practices," *International Journal of Quality & Reliability Management,* vol. 23, no. 6, pp. 647-669, 2006.

[SCM05A] International SCM Network, "Deliviring reductions in administrative burdens: An executive summary of the SCM method," *Administrative burdens,* 2005. [Online]. Available: http://www.administrative-burdens.com/filesystem/2005/11/ab_declaration_a5-_print_versie_177.doc [Accessed: Sept. 1, 2009].

[SCM05B] International SCM Network, "The International SCM Manual: Measuring and reducing administrative burdens for businesses," *Administrative burdens,* 2005. [Online]. Available: http://www.administrative-burdens.com/filesystem/2005/11/international_scm_manual_final_178.doc [Accessed: Sept. 1, 2009].

[SLA08] S. Slaughter, D. Harter and M. Krishnan, "Evaluating the Cost of Software Quality," *Communications of the ACM,* vol. 41, no. 8, pp. 67-73, Aug. 1998.

[TUO02] W. Tuohey, "Benefits and Effective Application of Software Engineering Standards," *Software Quality Journal,* vol. 10, pp. 47–68, 2002.

Appendix A – ECSS Software activities

The requirements forcing the activities presented are elicited from the ECSS standards titled

- Software – Part 1: Principles and requirements [ESA03A],
- Software – Part 2: Document requirement definition [ESA03B],
- Software product assurance [ESA03C].

The activities are limited to software related development at RUAG and by so doing operational requirements along with the acceptance review milestone are excluded.

In the table below all the activities are presented with final calculations in percentage describing the reason behind them along with a man-hour average. All the activities are sorted, for each topic, by most unnecessary time spent at the top. The reason calculations are done by calculating an average between the different estimated values with special consideration to comments made by the experts during the estimation interviews. As an example comments about being unsure were taken into consideration in favor of the other answers. Moreover if two out of three answers are closely matched and the remaining one differs a lot, the average was set with favor to the first two. However these differences are not displayed below only the final result from the calculations.

The reason reflected upon can be viewed in table 5 together with their definition. In the activity table the reason abbreviations are

- Quality adding (Q),
- Adherence to standard (A),
- Confidence adding (C),
- Development necessity (D).

Activities	Man-h	Q	A	C	D
SYSTEM					
Establish RB SW System Specification (SSS)	150	0	40	0	60
Software criticality analyses report: - SW dependability and safety analysis - List of critical software components	25	5	51	26	18
Establishing the Integration Plan - SW/HW Integration Plan (IP) SW and HW	80	0	15	0	85
Perform HSIA = HSIA HW SW Interaction Analysis (HSIA)	80	0	10	0	90
Establish RB System configuration items list - System partition with definition of items	3	0	0	0	100
Establish RB System configuration items list - System requirements allocated to items (SW, HW, Human) (Traceability to system partitioning)	25	0	0	0	100

Activities	Man-h	Q	A	C	D
Establish RB SW Interface Requirements (SWIRD)	0	0	0	0	0
Establish RB Interface management procedures	0	0	0	0	0
System design related to SW = System design	20	0	0	25	75
Verification of System design to system requirements conformance	5	0	0	50	50
System requirements to partition(system design) traceability	5	0	0	50	50
Ensuring that system level documentation takes software aspects into account, for example the User's Manual	25	0	0	35	65
MANAGEMENT & DEVELOPMENT					
Produce SW unit tests	1000	52	22	18	8
Establish TS(SRS/ICD): SW Requirements Specification (SRS)	250	0	50	0	50
SW Verification Report (SVR) - Software code verification report (Code inspection)	400	25	25	30	20
Make detailed design: SW Design Document (SDD) - Detailed Design (DDD)	475	0	20	0	80
SW-QR Milestones and technical review reports Software design and test evaluation report	90	0	90	10	0
Software Reuse File (SRF) Analyse potential reusability = SRF (draft) Analyse potential reusability =SRF	90	5	55	5	35
Define SW architectural design: SW Design Document (SDD) - Architectural Design (ADD)	280	0	15	0	85
SW-CDR Milestones and technical review reports	140	10	25	50	15
Software maintenance phase: Maintenance records	50	0	70	30	0
Detailed design verification as a part of code inspection: SW detailed design verification report	100	15	32	48	5
Schedulability analysis on architecture and on source code: SW Schedulability Analysis Report (SAR)	200	15	15	20	50
SW-PDR Milestones and technical review reports	140	13	18	38	31

Activities	Man-h	Q	A	C	D
Acceptance test plan	20	0	100	0	0
Acceptance test report	20	0	100	0	0
Software traceability matrices	50	0	40	60	0
SW Verification Report (SVR) - Software documentation verification report	107	20	17	30	33
SRR Milestones and technical review reports	100	13	18	38	31
Perform SW unit testing: SW Unit Test Report (SUTR)	35	6	47	47	0
SW Configuration File (SCF)	70	12	23	38	27
SW Unit Test Plan (SUTP)	27	8	57	8	27
All reviews and inspections shall be based on written procedures: Checklists	30	40	50	10	0
Plan SW verification: SW Verification Plan (SVerP)	30	0	38	38	24
Prepare SW Maintenance Plan (SMP)	25	0	45	40	15
SW Integration Test Plan (SITP)	20	0	55	25	20
SW Development Plan (SDP)	50	5	22	5	68
Procured software component list	10	0	100	0	0
SW Configuration Management Plan (SCMP)	25	0	40	15	45
SW Verification Report (SVR) - Software architectural design & interfaces verification report	48	13	20	30	37
Report SW budget: SW Budget Report (SBR)	90	15	10	50	25
Software deliveries: SW Release Document (SRelD) SW delivery at QR	80	0	10	10	80
SW Verification Report (SVR) - Software requirements verification report	40	10	20	25	45

Activities	Man-h	Q	A	C	D
SW Development Plan (SDP) *- Test environment and Requirements on the test equipment*	30	0	25	0	75
Establish TS(SRS/ICD): SW Interface Control Document (SWICD)	145	0	5	0	95
Analyses, Inspection and Review of design(RoD) verification report with respect to TS	45	55	15	15	15
Analyses, Inspection and Review of design(RoD) verification report with respect to RB	45	55	15	15	15
Establish the delivery procedure before the first delivery of software to the Customer.	30	0	15	0	85
TRR - Test Readiness Review	10	15	25	35	25
SW Verification Report (SVR) - Software integration verification report	9	0	30	25	45
Software behaviour verification	5	0	50	50	0
For every change due to noncompliance to requirements: SW Problem Report (SPR)	40	32	2	13	53
Make sure that review decisions are implemented.	50	10	0	20	70
Ensure that the members of the software object organization know, understand, and follow this SDP.	15	25	0	0	75
Setting up the development environment and all tools and scripts needed during the development	111	0	0	0	100
Review the RB = Comments on RB	50	0	0	0	100
DR is responsible for ensuring that the members of the design and implementation team understand the architectural design and implementation strategies documented in the SDD	20	0	0	0	100
Ensure that all the controlling documents and the agreements with the Customer are available for the software object organization.	10	25	0	0	75
Ensure that the members of the software object organization know, understand, and follow the SCMP.	5	0	0	70	30
SW User Manual (SUM)	90	0	0	0	100
For every change of requirements: SW Change Request (SCR) SW Change Proposal	40	45	0	20	35

Activities	Man-h	Q	A	C	D
Customer approvals of documents	0	0	0	0	0
The Head of the Software Department is responsible for providing the project with trained personnel For each individual, the Head of the Software Department maintains and executes a training plan	20	0	0	0	100
QUALITY ASSURANCE					
SW Product Assurance Report (SPAR)	170	27	40	33	0
SW Product Assurance Report (SPAR) Assess whether a training plan Assess whether the project members recognise and understand the SW plans relevant to their work by interviewing them Ensure that the methods and procedures stated in the SW plans are applied	SPAR				
SW Product Assurance Report (SPAR) - Assess whether the SW design as expressed in the SW Design Document is compliant with the SW Design Standards	SPAR				
SW Product Assurance Report (SPAR) - Assess whether the complexity and modularity of the design meet the quality requirements by analysing relevant product metrics w r t the target values.	SPAR				
SW Product Assurance Report (SPAR) - Assess whether there is an appropriate level of information for maintenance activities; and feedback the assessment result to the design team	SPAR				
SW Product Assurance Report (SPAR) - Assess whether the code analysis and verification are performed according to procedures addressed in [SDP] by inspecting samples of the performed code inspections and unit tests	SPAR				
SW Product Assurance Report (SPAR) - Assess whether the statement coverage, decision coverage, test coverage are achieved according to the objectives defined	SPAR				
SW Product Assurance Report (SPAR): - Assess whether methods and tools to be used are mature by checking, in the beginning of the life-cycle phase, the development and test environment, tools and methods in the SDP / SVerP / SValP (and other plans); and by checking the configuration status of these issues in the SW Configuration File	SPAR				
SW Product Assurance Report (SPAR): - Assess the characteristics of the SW product by collecting product metrics and analysing the metrics and checking whether they meet their targets The SW organisation will collect data for the code metrics. For these types of data, a static analysis tool will be used	SPAR				
SW Product Assurance Report (SPAR) or/and Software problem reports (SPR) - Assess whether software containing deactivated code and/or configurable code has been verified specifically, to ensure that the verification requirements are met	SPAR				
SW Product Assurance Report (SPAR) - Assess whether the tests and validation are performed according to the tests and validation procedures by inspecting the procedures. Assess the tests and validation results; especially, by checking w r t success criteria	SPAR				

Activities	Man-h	Q	A	C	D
SW Product Assurance Plan (SPAP)	45	37	30	33	0
SW Product Assurance Plan (SPAP) *- Definition of measures and verification activities in the SPAP*	SPAP				
SW Product Assurance Plan (SPAP) *- Define metrication programme* *- Software metrics that will be collected*	SPAP				
SW Product Assurance Plan (SPAP) *- Define assurance activities, to ensure the product meets the technical specification reuirements*	SPAP				
The resources used to perform the SW PA function	SPAP				
Audit plan	SPAP				
Plan and implement the collection of the metrics defined in [SPAP]. Collect software metrics and report to SPA.	25	37	43	20	0
The SW PA Manager contributes to the Project Manager's regular progress report with a section for SW PA issues	19	0	50	33	17
Ensure that suitable SW PA provisions are included in the contract. Review all changes to the contractual requirements concerning SW PA issues	25	7	33	10	50
SPR reviewboard	25	35	25	32	8
Assess whether the SW Problem Reports (SPR) / Nonconformance Report (NCR) are raised and managed according to [SCMP]. Check that the SPRs and subsequent actions are properly closed out.	15	40	40	20	0
Assess whether the pre-conditions of each formal SW review according to [SDP] are met by checking that all documentation required by [SDP] for the current phase has actually been produced, completed and is provided in the deliveries (data packages)	9	27	56	17	0
The SW PA Manager will establish SW PA requirements for the next-lower level suppliers	20	25	25	50	0
The SW PA Manager will monitor the next-lower level suppliers' conformance to the SW PA requirements	20	10	25	65	0
Assess whether critical software has been regression tested after any change and assess whether there is a need for additional verification of critical software after any change	15	40	33	27	0
Assess by inspection, in the beginning of the project; - whether the procedures and project standards include provision for all classes of software in the project, - comply with the relevant plans and contractual requirements, - are reviewed for suitability and for the availability of resources to implement them and, - are finalized and available before starting the activities using them	10	26	47	27	0
SW Release Document (SRelD) - Assess the status of NCRs / SPRs ahead of SW-CDR / SW-QR; check especially that no major NCRs / SPRs are open going into the SW-CDR / SW-QR	8	27	56	17	0

Activities	Man-h	Q	A	C	D
Assess whether the SW dependability and safety (RAMS) activities are planned and performed in sufficient time Assess whether the recommendations from the dependability and safety responsible are analysed and agreed by the SW team and properly implemented	10	37	43	20	0
Assess whether the selection and management of reused items (SW Reuse File (SRF)) are in conformance with the SW PA requirements	9	10	45	45	0
The SW PA Manager will assess whether the management of software risks and critical items is performed as defined A systematic risk assessment, reduction and control of risks will be performed in achievement of required technical performance	10	27	40	33	0
Perform Audits	10	25	35	40	0
Assess whether the TS/RB-validation test procedures have been executed for the same SW version without raising major NCRs / SPRs; unless a proper justification is provided	7	40	40	20	0
Identify incoming alerts issued by the Customer or ESA and check the handling of them. Identify and check the handling of SW problems (non-conformances) that could lead to alerts being issued to the Customer of ESA	20	60	13	17	10
The SW PA Manager will ensure that the procured software is correctly classified for dependability and safety criticality	5	0	50	50	0
Software process assessment records: - The SW PA Manager will assess the effectiveness of the processes used during the SW life cycle -Analyse the data to understand the strengths and weaknesses of the employed processes. -Suggest possible improvements of the employed processes. -Determine technology advancement needs. -Report the suggested improvements. -Report any problem experienced with the employed processes -When needed, initiate process improvement actions	10	45	15	40	0
Numerical accuracy analysis	0	0	0	0	0
Physical Configuration Verification: -Examine the physical media used for the software delivery, to verify their content and the labelling	0	0	0	0	0
The line organisation appoints the SW Product Assurance Manager (SPA)	1	0	0	0	100
Software process assessment plan: - A process assessment procedure shall be developed, documented and applied. - Collect process and quality data (metrics)	5	50	0	50	0
TEST					
Implement validation scripts	1500	33	7	3	57
Perform SW integration testing: SW Integration Test Report (SITR)	100	15	70	15	0
VR is responsible for verifying that the work performed by the verification team follows the approach laid out in the Software Validation Plan, and that the produced tests are adequate for testing the software	200	33	23	40	4

Activities	Man-h	Q	A	C	D
Perform validation testing w r t RB: SW Validation Test Report (SVTR) - RB Validation	60	25	75	0	0
Specify validation tests: SW Validation Testing Specification (SVTS) - TS Validation - RB Validation - Testing feasibility report	350	40	10	10	40
SW Validation Test Report (SVTR) - Validation report evaluation with respect to RB	20	0	100	0	0
STEL (SW test event log)	9	0	90	10	0
Perform validation testing w r t TS: SW Validation Test Report (SVTR) - TS Validation	70	40	10	0	50
Plan SW Validation: SW Validation Plan (SValP)	30	58	18	12	12
SW Verification Report (SVR) - Validation report evaluation with respect to TS	20	0	0	50	50
Independent software verification & validation plan (ISVV)	5	0	0	100	0
Independent software verification & validation report (ISVV	10	0	0	100	0