CHALMERS

Implementing a Software Verification and Validation Management Framework in the Space Industry

Master of Science Thesis Software Engineering and Technology

BOGDAN MARCULESCU

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, September 2010

Implementing a Software Verification and Validation Management Framework in the Space Industry

BOGDAN MARCULESCU

Examiner: ROBERT FELDT

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

Department of Computer Science and Engineering
Göteborg, Sweden September 2010

Abstract

Software designed for use in space related applications has a particular set of requirements regarding dependability and reliability, due mostly to the high cost of the finished products and to the difficulties of performing maintenance. In this setting, verification and validation activities become increasingly important and increasingly expensive. At the same time, the industry is under pressure to reduce expenditure without compromising the quality of its products. One option to reduce cost without reducing the overall quality was to develop a framework for the optimization of verification and validation activities. One such framework, called VAMOS [1], was developed at an earlier stage. The current paper presents the implementation of the framework, together with the modifications that were made to ensure success and the reasoning behind those modifications.

Keywords

Software Engineering, Measurements, Verification and Validation, Framework Implementation, Space Software Industry

Acknowledgements

The author would like to take this opportunity to thank some of the people that made this work possible.

I would like to thank Dr. Robert Feldt for his advice, ideas and feedback, both when applying existing ideas and when developing new concepts.

I would also like to thank Erika Hult of RUAG Space Sweden AB and all the employees of RUAG Space Sweden AB for their support, interest and for the ideas they contributed.

Finally, I would like to thank my parents, family and friends for their moral support and for their confidence.

Table of Contents

Chapter 1 - Paper

Implementing a Software Verification and Validation Management Framework in the Space Industry

Bogdan Marculescu
Computer Science Engineering.
Chalmers tekniska högskola.
Göteborg, Sweden
bogdanm@student.chalmers.se

ABSTRACT

Software designed for use in space related applications has a particular set of requirements regarding dependability and reliability, due mostly to the high cost of the finished products and to the difficulties of performing maintenance. In this setting, verification and validation activities become increasingly important and increasingly expensive. At the same time, the industry is under pressure to reduce expenditure without compromising the quality of its products. One option to reduce cost without reducing the overall quality was to develop a framework for the optimization of verification and validation activities. One such framework, called VAMOS [1], was developed at an earlier stage. The current paper presents the implementation of the framework, together with the modifications that were made to ensure success and the reasoning behind those modifications.

KEYWORDS

Software Engineering, Measurements, Verification and Validation, Framework Implementation, Space Software Industry

INTRODUCTION

The space industry has special requirements in terms of software, linked mostly to limitations of the industry itself. There is little margin for error, the costs are extremely high and the costs of failure are higher still. Software failure in the space industry could lead to loss of expensive equipment, costly setbacks in the respective space programs and, in the case of manned missions, may even lead to loss of life. To add to these difficulties, maintenance becomes extremely difficult and costly after deployment. All these difficulties lead to Verification and Validation being an important component in the development for software for space applications. Moreover, the industry in under both political and market pressure to reduce costs and development time. Given the importance of Verification and Validation activities, optimizations in this are offer a high potential for improvement by significantly reducing cost without reducing the overall level of quality.

In this context, an ongoing collaboration between Swedish universities and aerospace companies seeks to find ways to optimize cost and quality. One of the results of this collaboration was the development of a framework for the management and optimization of verification and validation activities in space software development [9]. The framework, called VAMOS [1] – meaning Verification and Validation Management and Optimization, emphasized a constant evaluation and improvement cycle. This cycle aims to improve the quality of the verification and validation activities while reducing the overall costs of these activities. Following development, the framework was tested on historical data. The decision was thus made to proceed to the implementation of a prototype of this framework. The academic goals of the prototype are to validate the framework and to provide additional information regarding practical problems that may arise. From an industry point of view, the prototype has the benefit of showing potential improvements and cost savings as well as beginning to perform its primary goal: that of optimizing and managing verification and validation activities.

OVERVIEW OF VAMOS

VAMOS is a software verification and validation management and optimization framework defined in [1] and [2]. It seeks to improve the overall performance of the verification and validation process by reducing the overlap between various validation activities and allowing the selection of the most appropriate validation activities in a given setting. VAMOS aims to achieve these results by using an iterative process consisting of four steps. The result of iterating through these steps provides constant improvement, while at the same time monitoring the results of the measures being taken to ensure that the measurement fit the goals and needs of the company.

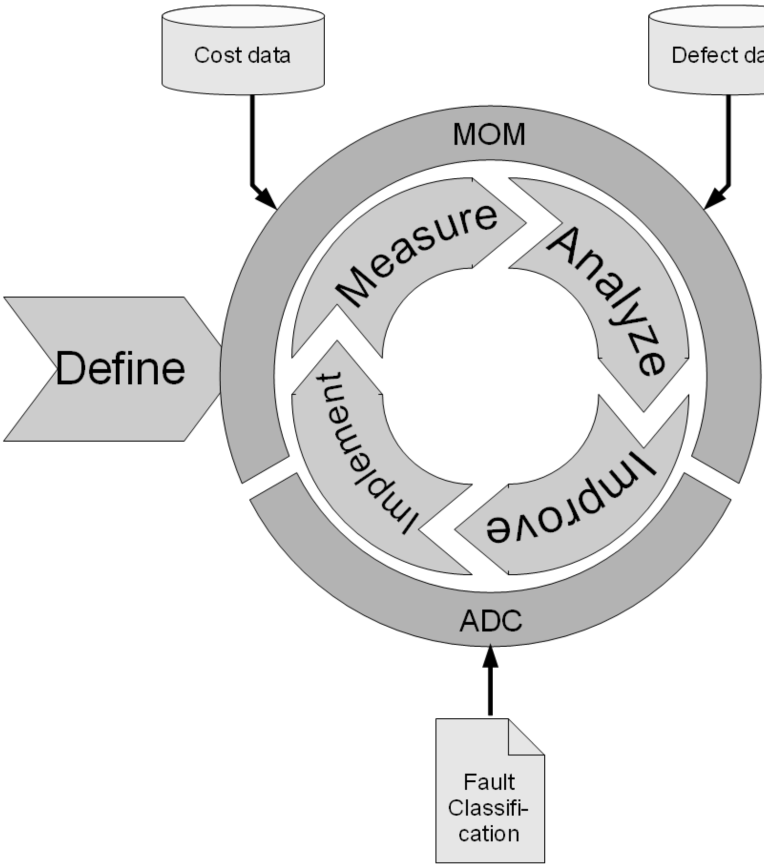

Figure 1. The VAMOS framework [1]

The steps of the VAMOS framework are:

- Analysis
- Improvement
- Implementation
- Measurement

In addition to these steps, the framework has a series of auxiliary devices to allow calibration and evaluation. An extra step Definition, allows the framework to determine the company's specific goals and information needs and to factor them to the activities of the framework.

Two quality gates are also defined, their purpose being to derive efficient cost and defect measurements and a domain specific fault classification [1]. These quality gates control the information that can be used by VAMOS and ensure that the information is suitable for use.

One of the main ideas underlying the VAMOS framework is the notion of fault slip-through. This means that there is a certain delay between the time a fault is introduced in a system, the time where discovery would be most beneficial, i.e. removal costs associated with the fault would be smallest, and the time the fault is actually found and removed. Associated to this notion is that of Removal and Regression Cost – the cost incurred in removing the fault and then performing all the tests that are needed to ensure the quality level of the module being tested. The later in the process a fault is discovered the more cost is incurred by redoing all the testing that had been performed up to that point. While this is not a hard and fast rule, usually the sooner in the process a fault is found, the lower the removal and regression costs associated with it.

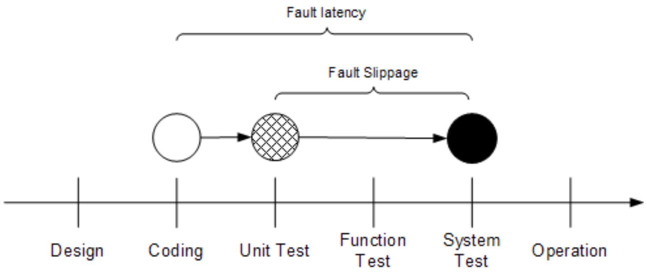

Figure 2. Fault Slip-through

This makes an appropriate combination of verification and validation activities even more important, and thus the management of such activities gains in importance too.

To allow the framework to be adapted to the specific needs of the company, several mechanisms for adapting the framework to the specific needs of the company exist. This flexibility is achieved by adapting the information filters in the quality gates to receive information of a lower accuracy. This approach provides less accuracy in the data being obtained, but enables the framework to be applied in situations where the exact regression and removal costs cannot be measured, the information cannot be collected in a cost effective manner or some other practical issues prevent the implementation of the quality gates in full.

METRIC PROTOTYPE

Reasoning

In order to obtain more information about the costs associated with implementing and introducing a new measurement in the process at RUAG, it was decided to develop and implement a prototype metric. The selection of the metric was done by company decision makers in accordance with the goals and priorities of the company and the amount of time that was to be devoted to this activity.

The area of interest selected was that of Code Inspection Sheet analysis. During the course of development, a software module has to pass a verification activity called code inspection. This activity consists of a number of checks performed on the code by tools and human inspection. The code inspection is performed in such a way as to look for specific faults, items that are likely to occur, that are more damaging or that can best be discovered by this means. The results of the Code Inspection activity are recorded on Code Inspection Sheets. These sheets have information regarding the faults that are being considered, called checkpoints, which of these was passed, failed or unavailable, the code of the module and comments recording problems, possible solutions and suggestions.

The main reason this activity was selected for a prototype measurement was the concern that most of the faults being discovered were not affecting code functionality and, as a result, the cost associated with this activity did not have a significant influence in terms of quality. Research exists [3], that shows that more than 75% of the number of defects found during code inspection were so-called beautification faults, i.e. faults that relate to code readability and maintainability, but have little effect on functionality. These findings seemed to support the concern that the activity had little impact on code quality.

Development

The idea behind the development of this tool was that the number of actual faults being found was less relevant, since a single code inspection could find a large number of faults of a certain type. The decision was made to focus not on the numbers of faults, but on the numbers of rejections associated with a certain type of fault. A module being rejected will have to go through the entire module testing and code inspection process again. This has the effect that the cost associated with a rejection is attributable more to re-testing than to removing the fault. Since in the particular case of code inspections, there is no additional cost associated with finding the defect. The effect of this is to keep the overall removal and regression cost constant, regardless of the number of faults. This is especially true for readability or maintainability faults. It was deemed that measuring the number of times each fault category caused a rejection was more relevant to the issue of cost than the number of faults.

The tool was developed to analyze all the code inspection sheets in a project, to store data regarding the module and relate that data to the numbers each checkpoint was passed, failed or not applicable. The tool itself was designed to be flexible and adaptable to several uses and to rely on as little human input or effort as possible, in order to keep the flexible costs associated with running the tool as low as possible.

Results

From an industry perspective, the results showed that, while so-called “beautification” defects were high in number, relative to other defects, they caused fewer rejections. It was found that less than 40% of the modules being rejected were also rejected for non-functional checkpoints. Moreover, it was found that no module was rejected solely for non-functional reasons. The significance of this is that, while non-functional, “beautification”, faults are found in high numbers, they have little effect in terms of costs. This proved that code inspection had a significant impact on code quality and showed the value of the activity for the company.

In terms of results significant from an academic point of view, the development of the tool provided valuable information into the costs of developing a new system, albeit a relatively simple one. The development process also showed the limitations that a measurement is likely to face in terms of available input and the way is which the type of available input is likely to affect the results being obtained. Moreover, the results were obtained and analyzed in a particular way, one that differs from previous approaches. This showed both the effect that a different approach may have on results and the importance of selecting the correct approach, in order to obtain results relevant in the context. The focus, in this case, on cost rather than number of faults changed the nature of the problem significantly and helped improve the relevance of the results. An additional benefit was to show the importance of the effort associated with analyzing the results: the results of any tool have to be interpreted and the cost of this interpretation has to be taken into account.

Conclusion

Concerns regarding validity and the level and type of analysis needed to use the results still remain. Some are mentioned and briefly explained in the annex relating to this item. Despite these concerns, the tool has proven useful and has provided valuable data, both from an industrial perspective and from an academic one.

VAMOS IMPLEMENTATION

This section describes the prototype implementation of the VAMOS framework. It includes the lessons learnt from this implementation and the changes being proposed to VAMOS as a result of these lessons.

The Define step

In accordance with the company's stated priorities, it was decided that the prototype implementation of VAMOS was to be performed for the Integration Phase of RUAG's process.

The implementation began with the Define step of the framework: the investigation into the process and activities of the company.

RUAG is implementing Integration Driven Development, so the Integration Phase is the central part of their process. The implementation of Integration Driven Development is, however, fairly recent. As a result there are areas where the process is not well defined and where information is unavailable. This is made worse by the complexity of the Integration Phase itself, since this is the moment where different components are brought together. The interaction between different departments makes information gathering about this area all the more difficult. In addition to all these difficulties, the process itself is fluid, subject to constant modification and improvement.

All these factors conspire to make the Integration Phase a difficult area to research and measure accurately. Moreover, given the relative lack of information regarding the area, new information needs may be discovered and priorities may shift during the course of the study. It is a type of setting that requires considerable flexibility in order to provide useful measurements and to collect relevant data.

The investigation itself was an exploratory study into the company's activities. It began with an analysis of the various documents describing the integration phase and the various activities and artefacts related to the Integration Phase. This was followed by a series of unstructured and semi-structured interviews with key persons involved in the Integration Phase.

To deal with all the complexities described earlier, it was deemed that a model was needed, to encapsulate all the information gathered so far and to allow new information to be incorporated as it became available. The model is created as a result of investigations prior to the project and is constantly improved after the project starts. New information, that becomes available as a result of investigating the ongoing project, is used to refine and improve the model.

The resulting model is a description of the various activities and the departments to which they belong, their timing, the documents being produces and potential sources of information.

In addition to creating and refining the model, the Define step is where the priorities and information needs of the company are defined. Since the information needs and priorities vary at several levels in the company, it is important to determine the information needs and priorities for all stakeholders of the Integration Phase. This was achieved mostly by unstructured and semi-structured interviews with the stakeholders, since documentation provides little information about priorities and intentions.

The stakeholders considered for this implementation were the management level and the project level. This analysis is company specific and different companies may have different structures, not just different priorities.

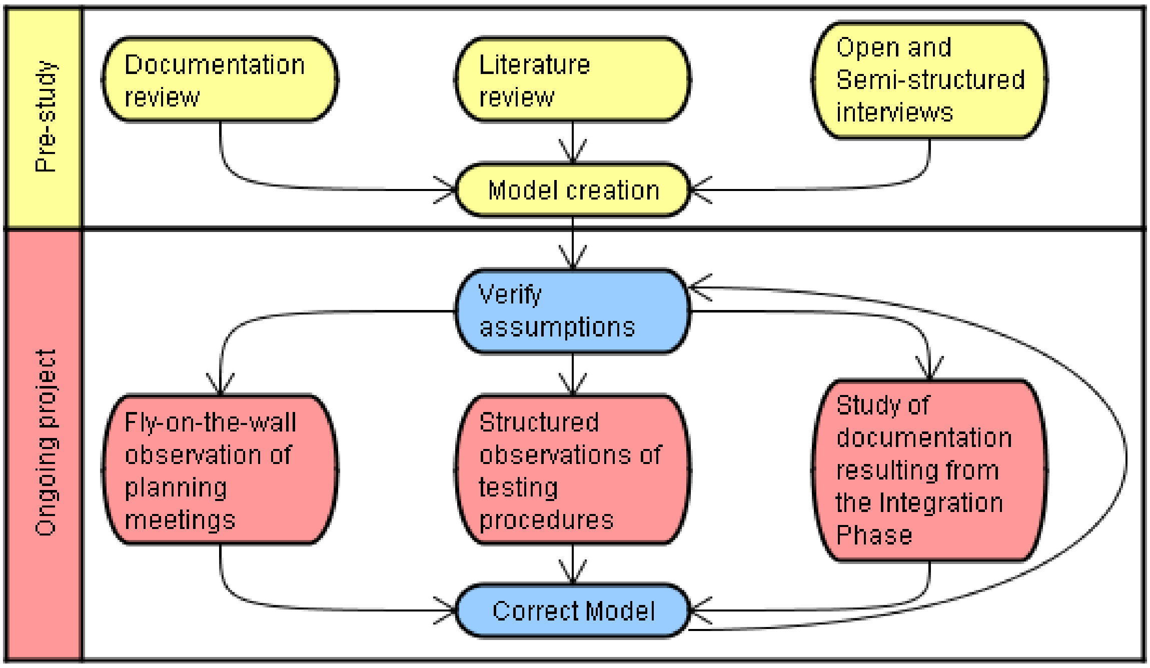

Figure 3. Research Methods for Model Development and Improvement

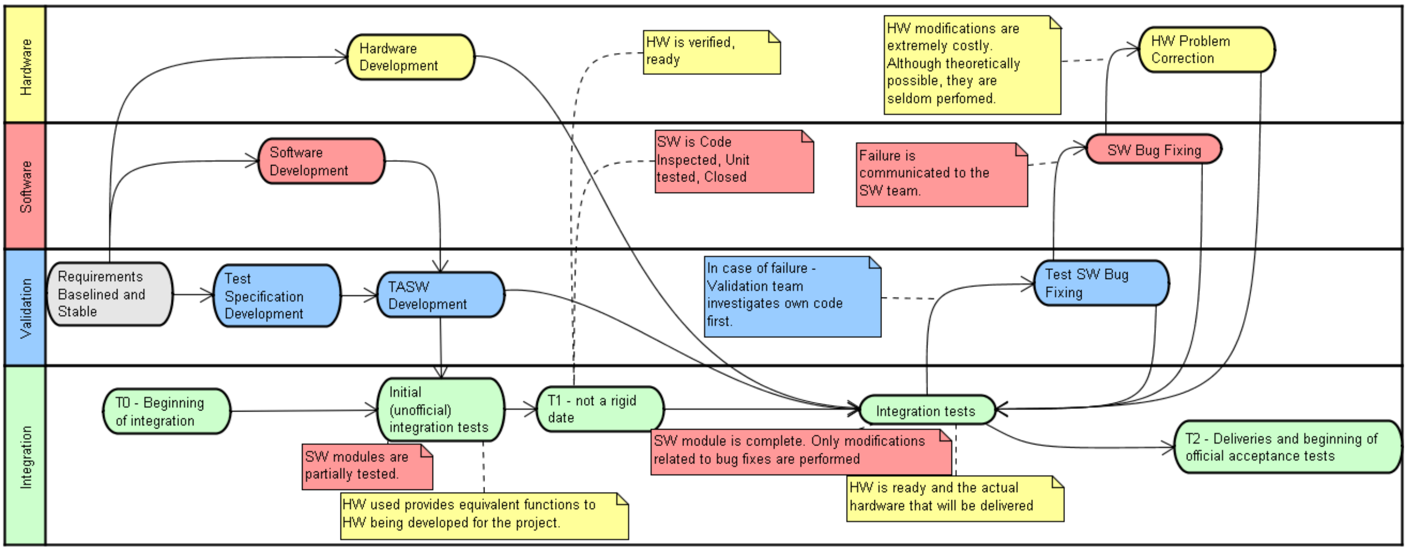

Figure 4. The Integration Phase Model

The management level is the level where decisions are made regarding all the projects. This includes major decisions such as delivery deadlines, resources assigned to each project and the priority each project receives. At the management level, the priority is cost reduction. The information need at this level focuses on the types of problems being encountered and the cost associated with each of the problem types. At this level, decisions are made that affect all the projects of the company, decisions regarding the process to be used and techniques and technologies that are to be implemented. The importance of the decisions being made at this level means that the information need focuses on accurate and well analyzed information. Lessons learned from past projects, analyzed post-factum and considering the context and peculiarities of each project for the basis for the decisions being made at this level. The crucial factors at this level are the quality of the information and the analysis.

The information need at this level is relatively easy to elicit and analyze. Since the management positions are generally the first contact one gets in a company, their priorities and goals are clearly stated from the start. Moreover, at this level there are mechanisms for determining and disseminating the goals and priorities throughout the company.

In particular, the management level information needs identified in this implementation were:

Process evaluation and improvement. Information obtained during the Integration Phase can help evaluate and improve the development process. Evaluation of the process during the Integration Phase is currently difficult, given the relatively little available information. Once more information is available this Phase, bottlenecks might be identified and so can potential improvements. Given the adoption of Integration Driven Development, potential improvements of the development process during the Integration Phase may have a considerable impact on the overall process.

Validation Activity effectiveness and cost data. One of the key principles of the VAMOS framework is the constant evaluation of the Validation Activities, their effectiveness and cost, and the notion of adapting and changing the selection of Validation Activities being performed to better fit the needs and requirements of each project. At this moment there is little practical information regarding the effectiveness and less still regarding the cost associated with each Validation Activity. From a perspective external to any project, data regarding these two areas could greatly improve the understanding of each Validation Activity, its strengths and weaknesses, as well as the cost, effort and practical considerations. Estimation and evaluations that are now purely theoretical could benefit from the strength of practical application and experimental results.

Fault number, type and cost data. This type of information is needed outside the project as well as within the project. Within the project, this information is important for evaluating the current status and identifying problems at the earliest opportunity. Outside the project, this type of information is useful as a means of identifying recurring problems across projects, finding areas related to faults already identified and that could benefit from existing solutions.

The second level is the project level. This is the level where project-related decisions are being made. This includes the planning of activities within the project and their priorities, the assignment of resources to each of the activities and handling of internal deadlines.

To determine the information needs at the project level, i.e. the information needs, goals and priorities of the project team, the Define phase interviews were designed to include questions relating to what information is available, information would be needed or might be useful, but is not available at the time.

The information need at this level is concerned with timing and early identification of delays. Information about the types of faults and problems being encountered is interesting, but as an input to delay assessment and prevention. Early identification of faults and delays can improve the overall response of the project team and allow the proper planning decisions to be made to ensure that the project is on schedule. The timely availability of the information is thus more important than accuracy and detailed analysis. Post-factum analyses of the delays are not as useful since the time when countermeasures would have been most effective has already passed. Moreover, information about delays and causes of the delays is based mostly on estimations and is treated as estimation. As a result, increasing the level of accuracy of the information being provided would not increase its usefulness, but rather introduce delays and require additional resources.

The exact items of information need identified for this level are:

Estimation and conformance to estimation. For any project, accurate estimations of the time and effort needed for the activities involved are necessary in order to request the needed resources and to ensure timely deliveries. The Integration Phase, in particular, is a part of the project that relies heavily on estimation. Moreover, estimating for the Integration Phase is harder do to, since it's difficult to estimate the time needed to find and remove an unknown number of unknown faults. Therefore, recording the estimations made and checking the conformance of these estimates to the actual times is an integral part of the information need.

Evaluation of the accuracy of previous estimations. In addition to controlling the Integration Phase conformance to schedule by means of estimation, the estimation process itself should be evaluated and improved. The first step in evaluating the estimation process is to record the accuracy of estimations currently being made. There is an added complication in the Integration Phase. This is caused by the fact that the activities are not known in detail as the Phase begins. An unexpected level of faults, both in terms of number and in terms of difficulty might throw off even the most reasonable estimates. It thus becomes very important if a certain delay was caused by faulty estimation or by factors that could not be foreseen and therefore could not be accurately estimated and accounted for. Once the causes for the delays are being identified, a more accurate evaluation of the estimation process can be made.

Improvement of the estimations are being made is the expected consequence of this evaluation of the evaluation procedure. Once the accuracy of estimation has been evaluated, the benefits in terms of accuracy improvement are weighted against the effort needed to achieve those results. This too is not as straightforward as might be expected. Since these are just estimations and since unexpected faults can throw off even the most accurate estimations, the decision to be made has to take into account additional factors.

Expected fault types and numbers, based on past experiences. Experience offers a lot of information in terms of types and numbers of faults that can be expected. Just like estimates, these numbers are far from accurate, but can provide a level of accuracy that is just enough for practical purposes. Information regarding the types and numbers of faults is available in the Integration Phase. Since this information is not recorded at the time, however, it is hard to draw any conclusions at the end of the Phase. Recording this information, even at a very basic level of accuracy, may provide useful information regarding what faults can be expected for a certain module or project.

Determining the information needs and priorities at this level is quite important, since it directly impacts the planning, activities and resource allocation that goes on at project level. Since any decisions taken regarding validation activities have to be implemented at project level, it becomes important to provide decision makers at this level with motivation and useful information too. Moreover, it is important to show that the Define phase, as implemented here, showed that there are differences between the information needs, goals and priorities at different levels.

Motivation has been identified, both in previous studies [6], and in the current analysis, as a major factor in the success of any measurement. Thus, motivating the project team to correctly and consistently collect the needed data is vital in order to obtain useful information. A major factor of motivation is the usefulness of the changes and measures being implemented to the project team.

Another priority of the Define phase is the identification of the information sources available during the Integration Phase. The information sources were identified by studying the documentation available, unstructured and structured interviews with personnel and by structured observations of various planning meetings. Due to the complex nature of the Integration Phase, information sources were of an unconventional type. The lack of a centralized system for collecting data and documents meant that the only information sources that would cover all the activities within all the departments involved in the Integration Phase were the Daily Integration Meeting and the documents associated with it. These information sources can provide estimation data, planning and timing data, as well as information regarding the problems that have been encountered and the effort needed to solve them. The information obtained from these sources consists mostly of expert estimations and evaluations and can be regarded as being reliable and of a high quality, albeit lacking in accuracy.

The implementation of the Define step provided a large quantity of information regarding the company and the specific context in which the VAMOS framework is to be implemented. Information about the process as well as the priorities of the company proved useful in selecting the appropriate validation activities and implementing them in the proper manner. The Define phase, as implemented here, is much more detailed than that described in the VAMOS framework. The reason for this is the need to adapt the information gathering aspects of the framework to the context of the implementation and to provide the company with all the relevant information that can be obtained. The implementation of the Define phase of the framework did help develop a toolkit of research methods that can be used in such a setting. This toolkit could provide a useful addition to the Define phase and provide a starting point for future framework implementations in different settings.

CHANGES AND UPDATES TO VAMOS

During the implementation of the VAMOS framework a number of lessons were learned and a number of modifications were made to the framework itself, based on those lessons. These issues have been gathered under a single heading, to provide an overview of the modifications and of the problems that caused them.

Goal and priority uniformity

As discussed earlier, one of the assumptions of VAMOS was that goals and priorities are known and uniform throughout the company. The initial investigations in the Define phase showed that this is not always the case. The differences between the goals and priorities at different levels may cause the measurement selection to be inadequate. In addition, a mismatch between the information needs, goals and priorities at the project level and the measurements being implemented may cause a decrease in the motivation to perform those measurements consistently and accurately, thus reducing their usefulness and even providing misleading results.

The way this problem was dealt with in this implementation was by extending the scope of the Define phase to include determining the goals, priorities and information needs both at management level and at project level. In other organizations there may be more levels to consider. Each of these levels has its own priorities and goals and each has its own influence on how a measurement is to be implemented and used. Since the Define phase of the framework is concerned with obtaining all the necessary information to ensure the proper functioning of VAMOS, the solution was to extend the scope of the Define phase to include all the additional information described here, without altering the phase’s purpose.

Goal and priority stability

The Define phase is meant to be an initial step that defines all the information needed to successfully implement the framework. The assumption being made is that, once this information is obtained, the company’s goals and priorities are more or less stable.

During the course of the current prototype implementation, however, it was discovered that the goals and priorities might change significantly. As more information becomes available, decision makers may realize that in some areas the information is insufficient, while in others resources are being wasted gathering information that is not needed or used. This is especially true in cases where an exploratory study is used to obtain initial information into a certain area. Exploratory studies are conducted specifically to determine in more detail the information need and information availability for a certain area. In this perspective, re-assessing the information needs, goals and priorities of the company is part of the framework implementation work. Even in situations where the measurements are fairly stable however, keeping track of changing priorities and information needs would help to ensure that all information needs are satisfied with no resources wasted on measurements that are not required.

The solution used for this implementation was to constantly re-assess the information needs and priorities as the implementation proceeded. Close cooperation with the decision makers within the company helped decide on the suitability of certain decisions and modifications and ensured a good fit between the information being provided and the information need. This approach is appropriate especially for exploratory studies, where the information needs change rapidly as more information becomes available. Moreover, this level of flexibility in an exploratory setting can provide excellent results by allowing a complete shift in the focus of any information gathering process. Once a determination is made that a certain area of study can provide better information and better cost savings, the new information can be acted upon immediately.

While this approach was suitable in the exploratory setting presented here, it will be unnecessarily costly in situations where the changes take place less often. The kind of close cooperation with decision makers presented above is a useful tool in the given setting, but it is also quite a costly tool. Re-assessment of priorities and goals takes a lot of time and effort, and may quickly prove to be a waste of resources if the changes don’t happen often enough. In this situation, a solution would be to change the Define phase into a Define process. This would be a process independent of the implementation of the framework. The result of the current implementation of the Define phase is a model of the process of the area being studied, with information needs, goals, priorities and information sources attached. The Define process would be charged with identifying changes in the information needs, goals and priorities in the company, and then ensuring that the model resulting from the initial Define phase is kept up to date. This too is a potentially expensive solution and possible subject for further research.

Issues regarding acceptance and reliability

As discussed earlier, acceptance and motivation have a significant result of the outcome of validation activities and measurements. The current implementation of VAMOS seeks to investigate the possibility of improving motivation and acceptance by shifting the burden of decision making from the researcher to the company. This approach has the benefit of making the company own the measurements and validation activities framework and ensure direct involvement in data collection, maintenance and further development. Since the goal is to ensure a good fit between the company's information needs and the information being provided by the measurements, turning ownership of the measurements to the company itself is a step closer to achieving the goal in a consistent and reliable manner.

To implement this approach, the company decision makers have to be provided with accurate information about their options, tailored to the specific situation of the company. Moreover, to enable an informed decision to be made, additional information needs to be provided on each measurement regarding costs and benefits, as well as risks associated with the measurement and the degree to which implementing the measurement is likely to affect their current activities and their personnel.

The VAMOS framework has a quality gate, the Measurements Options Model or MOM, which seeks to help the researchers choose between several measurement options in a given situation. The focus of the information in the MOM is on cost and accuracy, while other factors are to be dealt with by the person implementing the framework. By shifting the burden of decision to the company, additional information needs to be provided. Since cost is not the only factor affecting the decision, all the other issues that have been identified, need to be analyzed and addressed.

The solution for this problem is to develop the MOM tool and adapt it for the new setting. The new MOM becomes a toolbox of measurements, containing as much information as possible about the measurements themselves. In addition to the MOM, an additional tool was developed called the Company Measurement Options or CMO. The purpose of the MOM is to select the measurements that are appropriate in terms of the information needed and available, the level of accuracy being demanded of it and the level of resources available for its implementation. After this selection is made, it is the CMO that would help determine the fit between company specific issues and the measurements initially selected. For example, a measurement could be appropriate in terms of costs and accuracy, but entail changes in the process that the company is not willing or able to make. The CMO would provide the decision makers with the information needed for their decision and a certain level of analysis of the available data.

Flexibility and adaptability

The VAMOS framework provides flexibility in terms of changing the selection of the measurements to be implemented. Since more measurements offer the same type of information, a choice can be made based on the needed level of accuracy, cost as well as any other of the factors mentioned earlier. While this level of flexibility is extremely useful, especially in the exploratory setting of current implementation, other areas have also benefitted from additional flexibility.

The first such example is that of the framework itself. VAMOS provides a number of mechanisms for adapting to the needs of the particular setting. While these mechanisms are useful, additional changes might also be required. One such example was discussed earlier in this paper: the need to expand the scope of the Define phase, so as to provide all the information needed for the current implementation. While the need to keep the framework as general as possible means that some areas are under-defined, it is important to note that the framework itself can and should be changed to fit the needs of the implementer. Practical issues cannot always be foreseen and planned for, so changes are inevitable. Such changes should be documented, their reasons explained and their results analyzed, but the possibility to change the framework without giving up its benefits is an important aspect of VAMOS. The modifications made as part of this implementation are just one attempt to provide the needed information as efficiently as possible.

The second level of flexibility being investigated in the current implementation is that of the measurement itself. In the VAMOS framework, measurements are the building blocks of a measurement system. They have their own characteristics, benefits and drawbacks and they can be combined in any number of ways to provide the information needed. They are not, however, seen as flexible themselves. If a measurement does not fit the information need, it is replaced with another or a combination of others, which do. One drawback of this approach is that replacing a measurement is a costly option. Apart from costs associated with the development of the new tools needed for the measurement and the testing of these tools in the specific company conditions, there are less tangible costs of training personnel and allowing them time to adapt to the new approach, as well as drops in confidence, if more measurements are replaced or if the replacements themselves prove inadequate.

The problem is even more significant in the exploratory type of setting this prototype was implemented in. Since initial information was scarce, information needs and priorities were hard to estimate in advance and difficult to plan for. With each iteration of the framework, new information became available and changes needed to be made. Rather than repeat the selection process and replace the prototype measurement, it was decided that the measurement itself should be allowed a level of flexibility. This enabled the measurement itself to be adapted to the specific setting, during the course of data collection, improving the quality of the measurement itself and of the data being collected. Certain measurements have their own constraints and limitations that may not allow such an approach. For those that can be adapted, though, the benefit is the development of a customized set of measurements, adapted specifically to fit the information need of the company, while making the best use of the available information sources and collection resources.

The MOM and the CMO in practice

The need and purpose of the two tools, the updated MOM and the CMO, was described earlier. A detailed description of the tools themselves, the information they contain and the meaning of that information is available in the annex dealing with this issue.

While the tools are quite similar, the focus of each is on different aspects and different users. The MOM is meant to be used by the implementers of the framework and tries to provide them with the information they need to make the initial selection. The CMO is meant to provide the decision makers with the information and the analysis results they need. The tools try to take into consideration and analyze as many of the factors affecting the decision as possible and provide all the information it has available on those factors. The CMO takes into consideration the company's process and information focus, as well as any other particular needs of the company. This is achieved by using the model developed and maintained as part of the Define phase.

Once the Define phase is complete, the process of selecting the measurements to be implemented began. First, based on the MOM and information regarding the information needs, goals and priorities of the company, an initial selection was made of the measurements proposed for implementation. The measurements thus selected were analyzed further, taking into account the model of the company's process, goals and information sources. The purpose of this analysis is to find how each of the measurements would be affected by the specific conditions of the company, and how it would affect the company's process. Additional benefits, drawbacks and possible risks associated with each measurement are also analyzed.

A secondary selection can be performed at this stage, to eliminate measurements that do not fulfill requirements of resources, accuracy, process or any other factor. The purpose of this second selection is to provide decision makers with options that can reasonable fulfill their goals and expectations.

Once all the information in the CMO has been gathered and analyzed, a workshop was held with decision makers within the company. The purpose of the workshop was to analyze the options from the perspective of the company and to decide which of them were to be included in the prototype implementation. For this reason, all levels of stakeholders were involved in the choice, to ensure that information needs at all levels are represented.

In practice, the MOM and the CMO require large amounts of data and analysis in order to be useful. Information regarding the measurements themselves, the data sources, the company's goals and priorities, the process and a host of other factors is needed in order to properly use these tools. Careful analysis is also needed to ensure that all the conclusions being drawn can be supported by evidence. Since results of these tools, in the form of data and analyses, are going to be used for decision making, ensuring the quality of the results is a priority. All the effort that goes into properly implementing and using the CMO is justified by the need to ensure that the decision regarding the selection of measurements is an informed one. While the implementer of the framework no longer makes the decision regarding the measurement selection, they still have the responsibility of providing the appropriate levels of information and analysis to enable the company to make the right decision.

FRAMEWORK ITERATIONS

Given the low level of information previously available, the decision was made to implement a flexible measurement. The goals for this measurement were:

Planning and scheduling information

Keeping track of schedules and delays and providing early warning regarding possible delays. Given the high level of interaction between the different components of the Integration Phase, early identification of delays in one area may provide significant warning and allow planning improvements. Planning and scheduling information in mostly relevant at project level, allowing the Integration Responsible and the Project lead to change and adapt plans as needed, in order to ensure the deadlines are met.

Identifying the most significant problems

In this context, the most significant problems are those causing delays. This is an important issue at both project and management level. At project level, knowledge of what problems to expect at different stages helps with planning and resource allocation. At management level, fault type information can improve the overall development process.

While the goals of the measurement were decided during a high level measurements workshop, the practical details of the measurement implementation were left to the Integration Responsible. The responsibility of this role for the entire Integration phase ensures that the information needs at this level are the most relevant and the role is a suitable owner for the measurement.

From the point of view of the VAMOS framework, only one iteration was performed. The measurement implemented was found suitable for the purpose and provided useful information. While changes to the measurement were performed, they were due to the flexibility of the measurement and not dictated by the VAMOS framework.

EXPLORATORY MEASUREMENT FRAMEWORK

The exploratory nature of the investigation into the Integration Phase meant that some existing measurements were unsuitable. In such a context, a metric would benefit from flexibility and the possibility to change focus as new information becomes available. The need for flexible measurements, especially in the context of exploratory research, has offered an interesting dilemma. A generalized version of a tool for exploratory research would be a useful addition to any toolkit. However, the uncertain nature of the exploratory study and the lack of information make such a generalized tool difficult to develop. One solution is the Exploratory Measurement Framework, the purpose of which is to enable the development of flexible measurement in a particular setting.

The Exploratory Measurement Framework, EMF, is an iterative approach to developing light weight measurements for a given context.

The input consists of information about the context, the information needs and the data sources. For the purposes of this implementation all the necessary inputs were provided by the analyses performed as part of the VAMOS framework.

The Define phase of VAMOS provides information about the company's process. In this particular case, the model of the Integration Phase, developed as part of the implementation of the VAMOS framework, provided all the useful information regarding the roles, activities and artefacts linked to the Integration Phase. This information is used to ensure that the measurement is as lightweight as possible and has as little an impact as possible on the existing process.

Another component of VAMOS, the CMO, provides useful information in terms of alternative measurements available, their costs, benefits and other relevant factors. This is useful information when the measurements is chosen and defined, as well as during the process of redefining and modifying the measurement. Additional ideas, concepts and methods can be adopted from existing measurements and incorporated into the current project.

The CMO analysis regarding data sources is useful in terms of both Measurement definition and redefinition and Data collection and analysis. Knowledge of the available data sources is vital in ensuring that only those measurements are selected that rely only on information sources that are available and appropriate in terms of accuracy and reliability. Data collection and analysis also depends on detailed knowledge of the available data sources. This knowledge can be derived from the detailed analysis performed as part of the CMO component of the VAMOS framework.

The evaluation of the measurement and its suitability with respect to Information needs depends on knowledge of those information needs. This knowledge is derived both from the analyses of the VAMOS framework, in the case of high-level information needs, and from the Local information needs. The owner of the measurement, in this implementation the Integration Responsible, is in charge of defining local information needs and ensuring that the measurements are modified in accordance with those needs.

The Exploratory Measurement Framework itself consists of 4 stages.

- **Measurement definition and redefinition.** The metric in questions is first defined, taking into consideration the information needs, both local and those identified by VAMOS, data sources and other considerations such as process impact and motivation. After the first iteration, this stage is where the measurement is modified and re-designed, based on the information becoming available as a result of the previous iterations. Changes in the degree to which the previous implementation fits the information needs, the data sources available or in the information needs themselves are factored in the development of the new version.

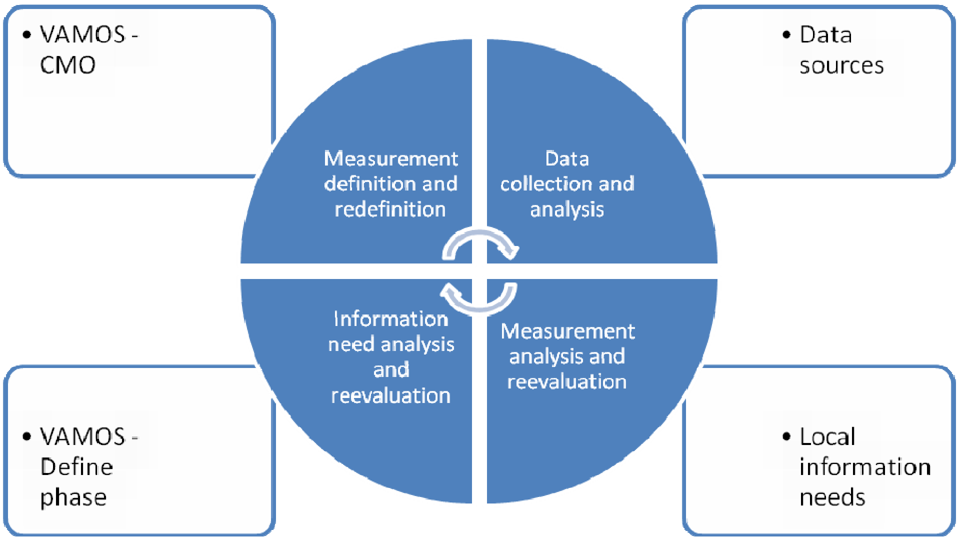

Figure 5. The Exploratory Measurement Framework

- **Data collection and analysis.** The measurement is implemented in this stage, data is collected and analyzed. It is important that the measurement be implemented in a real setting, so that the practical obstacles as well as the benefits can be identified. An added benefit to the implementation of the metric in a real setting is that it builds confidence in the metric and is useful practice in its use.

- **Measurement analysis and reevaluation.** The performance of the measurement is analyzed, with respect to existing information needs, expected results, implementation obstacles and opportunities and any other ideas and information that become available during the course of implementation.

- **Information needs analysis and reevaluation.** The information needs are reevaluated an analyzed in the light of the new information becoming available. The information needs that are most often updated are local information needs. However, by adopting the proposed modifications to VAMOS, especially making the Define step a continuous and permanent process, information needs at the higher levels can be updated as well. Since VAMOS has the tools to identify and analyze higher level information needs, the framework itself is responsible for pushing the needed information to the EMF. The Measurement analysis and reevaluation and the Information needs analysis and reevaluation stages can be performed simultaneously and with a high level of exchange of information between them. Together they form the basis for the proposed modification to the tool in the following iteration. This is also where the decision can be made that further modifications are no longer needed and the metric has stabilized.

EMF IN PRACTICE

The measurement

The measurement developed with the EMF is a flexible information collection tool, the purpose of which is to obtain certain types of information from planning meetings. The measurement itself consists of a set of two or three questions related to the delays encountered and a classification of possible reasons for the delays. The classification is displayed on the Integration Board, to allow all the people involved at the meeting to understand what kind of information is needed and to suggest improvements. The people present at the Integration Meeting, as described in the RUAG process breakdown, are those responsible for different aspects of the Integration Phase. Each of the people involved in the Integration Meeting is in the best position to evaluate the situation of the aspect they are responsible for, how much of a delay they have in their area and the likely reason for that delay. They are also in a position to know how delays in other departments are likely to affect their activities.

In keeping with the overall focus on flexibility of the implementation of the VAMOS framework, the measurement is to be implemented in a flexible manner. The initial form of the measurement can be changed, both in terms of the questions being asked and in terms of the delay reason classification. As described in the CMO, the Owner is the Integration responsible, so it’s his responsibility to define the information need and it’s his prerogative to decide on changes in terms of the questions to be asked, the frequency and exact timing of the data collection and the solving of any other practical issued that might arise.

The implementation of the measurement changed in the course of the application, with the classification of the delays being updated as a result of practical information becoming available. An example of such a change is related to the category “Planning problems”. Practical implementation revealed that the category was too broad, including both planning problems that could be foreseen, i.e. incorrect estimates or improper planning, and planning problems that could not be foreseen, e.g. use of new technologies and approaches that are hard to plan for and thus create difficulties. The problem was solved by adapting the classification to have distinct categories for each of those and make the distinction clear. The problem was all the more important since, during the project under study, a technological approach was used for the first time by the project team and delays associated with this technology were considerable. By making the distinction between the two categories, the delays caused by the new technologies were not misinterpreted. Another example of modification was the addition of the “Infrastructure problems” category, when the necessity for such a category became apparent.

The decision to change belonged to the Integration Responsible, with all the people attending the Integration Meeting having the possibility to propose changes. Data collection was performed twice a week, with the exact date when the data was to be collected also at the discretion of the Integration Responsible. As stated before, the reason for this decision was to allow the Integration Responsibility to shift the exact date in such a way as to avoid the times when the schedule was busier. This also had the effect of making the measurement easier to accept for the people attending the meetings. The lightweight approach, together with the flexibility to change the data collection time, helped obtain the necessary information without creating its own delays in situations where time was essential for meeting deadlines.

The final form of the measurement is that of a table, containing the categories and the areas of responsibility that take part in the Integration meeting. For each collection date, the responsible for each area states the delay they have had since the time of the previous data collection and split that delay according to the causes. The classification of the possible causes is displayed and an additional document containing the definition of each category is available to all those attending the meeting. This means that the meaning of each category is only discussed if there are doubts in specific cases and that the increase in meeting time due to data collection is only of a few minutes. For a meeting of about 10 – 12 people, the data recording time was consistently under 3 minutes.

The answers will be in terms of man days of delays that were incurred. The purpose is to identify problems that cause the most significant delays. For each of the questions mentioned there are a couple of sample answers. The purpose of these is to define the level of detail, to better explain the kind of answers expected and to help with the classification of the causes to some extent.

The problems that are to be recorded are unpredictable faults or problems that fall outside the scope of regular planning efforts. This means that the delay caused is the major unit of the effort, with due corrections for the number of people involved. The time spent on a fault that did not cause delays (due to slack in the planning or reserve resources available) is not recorded because it is considered to not have put the project behind schedule and thus that project is still on track.

During the implementation process, the need for additional information became apparent. Certain situations caused delays, although the causes for the delays were not immediately apparent. These delays had to be recorded and analyzed as well. These problems are written on the Integration Board as Key Issues, and are often the cause of considerable delays affecting several areas of responsibility.

This type of situation is hard to deal with in the manner presented above in a clear and accurate manner. This is the reason an auxiliary tool was developed. The tool takes its name from the problem it's trying to solve, Key Issue Recording Tool, and it consists of a table where each of the key issues currently under investigation are noted by means of a unique ID and every participant in the meeting states the delay this has caused. When the investigation is finally closed and the cause of the delay discovered and classified, the exact numbers and time breakdown are available for an accurate and relevant further analysis.

Results

The data seems to indicate considerable delays, about 48% of all delays, to be found within the Software department. This result should be, however, provided with the appropriate context. Data collection began after the hardware design and development had finished, thus hardware problems and delays in that area remained unrecorded. Moreover, delays in the hardware department, late in the development process, tend to cause increasing delays for all departments affected by the changes.

Two conclusions can be drawn from this result. First, the results show that Timesharing and Planning problems for new technologies are the leading causes of delay. When it comes to applying new technologies and processes, some delays are to be expected. Relative inexperience in the new technology makes estimation difficult, while unforeseen problems that arise during the course of practical application may cause extensive delays. It is to be expected that delays relating to this area will diminish, as the technology become more familiar and the more common problems are overcome.

The second conclusion that can be drawn is a direction where further research can be performed. The high delay associated with the software department can be investigated in further detail, to determine more exact causes for the delay. This is the type of result that is to be expected from exploratory studies: an area of the process that requires investigation in more depth. A more detailed breakdown of the problems encountered by the software department should, however, be the subject of another study.

DISCUSSION AND CONCLUSIONS

The current implementation is a prototype, with certain restrictions and limitations. First, time constraints restricted the number of measurements implemented within the framework. The second limitation is that the framework was limited to studying a single phase during the development process, albeit a vital phase.

Regardless of the scope limitations of the framework, the viability of the framework was proven. Moreover, the Integration Phase, that the current implementation phase was focused on, is a particularly complex and difficult phase to analyze. The useful results provided in this context shows that the VAMOS framework is a useful tool and its flexibility enables it to perform adequately even in less than ideal circumstances, where little information is available and where information needs may change frequently.

Overall, the VAMOS framework, with all the additions, updates and additional tools attached to it, is a powerful means of developing and managing verification and validation activities. The current project proves the practical validity of the framework and provides useful information about the practical aspects, benefits and obstacles that were encountered. These are valuable lessons that will benefit both the current implementation and any further implementation attempts.

REFERENCES

[1] J. Schulte, A Software Verification & Validation Management Framework Industry for the Space Industry, Blekinge Institute of Technology 2009.
[2] P. Preissing, A Software Verification and Validation Management Framework for the Space Industry, Technische Universität München, 2009.
[3] M. V. Mäntylä, C. Lassenius, *What Types of Defects Are Really Discovered in Code Reviews?,* IEEE Transactions on Software Engineering, VOL. 35, NO. 3, MAY/JUNE 2009
[4] L.-O. Damm, L. Lundberg, and C. Wohlin, *Faults-slip-through - a concept for measuring the efficiency of the test process*, Software Process: Improvement and Practice, vol. 11, no. 1, pp. 47-59, 2006. [Online]. Available: http://dx.doi.org/10.1002/spip.253
[5] M. A. Wojcicki and P. Strooper, *An iterative empirical strategy for the systematic selection of a combination of verification and validation technologies*, in Software Quality, 2007. WoSQ'07: ICSE Workshops 2007. Fifth International Workshop on, 2007, p. 9. [Online]. Available: http://dx.doi.org/http://dx.doi.org/10.1109/WOSQ.2007.4
[6] M. Umarji, Understanding developer resistance to software metrics,
[7] A. Rainer, T. Hall, N. Baddoo, *Persuading Developers to „Buy into" Software Process Improvement: Local Opinion and Empirical Evidence,* Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE'03)
[8] M. Staron, W. Meding, C. Nilsson, *A framework for developing measurement systems and its industrial evaluation*, Information and Software Technology 51 (2009) 721–737
R. Feldt, B. Marculescu, J. Schulte, R. Torkar, P. Preissing, E. Hult, *Optimizing Verification and Validation Activities for Software in the Space Industry*, DAta Systems In Aerospace (DASIA), June 2010.

Chapter 2 – Introduction

Software in the aerospace industry

The space industry has special requirements in terms of software, linked mostly to limitations of the industry itself. There is little margin for error, the costs are extremely high and the costs of failure are higher still. Software failure in the space industry could lead to loss of expensive equipment, costly setbacks in the respective space programs and, in the case of manned missions, may even lead to loss of life. To add to these difficulties, maintenance becomes extremely difficult and costly after deployment. All these difficulties lead to Verification and Validation being an important component in the development for software for space applications. Moreover, the industry in under both political and market pressure to reduce costs and development time. Given the importance of Verification and Validation activities, optimizations in this are offer a high potential for improvement by significantly reducing cost without reducing the overall level of quality.

In this context, an ongoing collaboration between Swedish universities and aerospace companies seeks to find ways to optimize cost and quality. One of the results of this collaboration was the development of a framework for the management and optimization of verification and validation activities in space software development [9]. The framework, called VAMOS [1] – meaning Verification and Validation Management and Optimization, emphasized a constant evaluation and improvement cycle. This cycle aims to improve the quality of the verification and validation activities while reducing the overall costs of these activities. Following development, the framework was tested on historical data. The decision was thus made to proceed to the implementation of a prototype of this framework. The academic goals of the prototype are to validate the framework and to provide additional information regarding practical problems that may arise. From an industry point of view, the prototype has the benefit of showing potential improvements and cost savings as well as beginning to perform its primary goal: that of optimizing and managing verification and validation activities.

RUAG Aerospace Sweden AB

The thesis was done at RUAG Aerospace Sweden AB, a subsidiary of the Swiss- and German-based company RUAG [11]. It specializes in highly reliable on-board satellite equipment including computer systems, antennas and microwave electronics and adapters and separation systems for space launchers. RUAG Space in Sweden has been active on the space equipment arena since the late 60´s and has over the years developed world-class products in the areas of Computers and Data Handling Systems, Antennas, Microwave Electronics, Payload Adapters and Separation Systems, Satellite Structures and Sounding Rocket Guidance Systems.

The headquarters and location for design and manufacture of digital electronics, microwave electronics and antennas is in Göteborg, Sweden and employs 320 people (in December 2009) [11].

Typically up to five projects are developed simultaneously in varying team sizes of about 10 people. The software projects range from 10 KLOC to 100 KLOC depending on the project and are developed mainly in C but with some low level parts written in assembler [1].

Background

VAMOS was developed as a framework for the efficient management and optimization of verification and validation activities [1]. The framework was developed in a real-world software development environment, in cooperation with RUAG Aerospace AB. The VAMOS framework was developed and tested on historical data, but was never validated in an active project.

Its involvement in the development of the VAMOS framework made RUAG Aerospace AB an obvious candidate for a prototype implementation. The academic purpose of the prototype would be to validate the framework and to obtain additional information about the practicalities of implementing such a framework in an active project.

RUAG has recently adopted the Integration Driven Development process. The focus of this process is the development of software, hardware, testing equipment and all other auxiliary components in parallel. As these components are developed, they are integrated into the final product and any integration problems or compatibility errors are solved as they are discovered. The Integration Phase is the most important and the longest phase of this process. It is the phase where all the components are designed, developed, tested and integrated into the product. Given the focus on integration of various components and the subsequent involvement of various departments in the company, the usual approaches to metrics and measurements were difficult to implement in a cost-efficient in this setting.

From an industrial perspective, the purpose was to conduct an exploratory investigation into the Integration Phase, provide a useful set of metrics for the purpose and provide a framework for managing the metrics that were proposed. It was decided that the VAMOS framework's focus on cost efficiency and optimization would provide a better alternative.

Contribution

The thesis contributes a prototype implementation of the VAMOS framework, along with all the lessons learned as a result of this implementation and proposed changes to the framework that are the result of these lessons.

In addition to this, a measurement development framework was developed that is better suited to providing the information needed by the company. The interaction between the verification and validation activity management framework and the measurement development framework has also been analyzed and illustrated.

Aims and objectives

The aims and objectives of the thesis are derived from two main sources. The need to provide practical validation of the VAMOS framework in a real-world setting, as part of an active project provided the academic aims. The information needs associated with the company provided the industrial aims. The overall aims and objectives of this thesis are as follows:

Investigate the Integration Phase. The company's main aim was to obtain additional information about the Integration Phase and to determine how measurements could best be implemented in the Integration Phase.

Implement the VAMOS framework. The prototype implementation of the framework would provide validation for the framework as well as useful lessons about the practical obstacles, difficulties and opportunities associated with the implementation.

Use and, if needed, develop the appropriate measurements. In order to obtain the required information about the Integration Phase it was known that specialized measurements would have to be used. If no measurements were found appropriate in terms of accuracy, cost or any other factors, changes could be brought to the measurements and all new measurements developed to fulfill the information needs.

A consideration of a more flexible and adaptable measurement concept. Given the exploratory nature of the study, it was deemed important to have as flexible an approach to measurement implementation as possible.

Research questions

1. In order to achieve the aims and objectives of the thesis, the following research questions would have to be answered.
2. How can a prototype of the VAMOS framework be implemented in an industrial setting?
3. What modifications need to be made to the framework to make implementation possible and cost efficient and to minimize the effect on the company's process?
4. What measurements does the framework need to use?
5. How does the metrics flexibility and adaptability translate into practical tools and methods?
6. How can the concepts of flexibility and adaptability be adapted to measurement development? How would an agile measurement development framework be defined?

Expected outcomes

- A practical example of VAMOS implementation
- An updated version of VAMOS, incorporating the lessons derived from the practical implementation
- A practical example of a flexible measurement and its adaptation mechanism
- A flexible measurement development framework

Chapter 3 - RUAG process description and status

Introduction

During the implementation efforts for the VAMOS framework, it became clear that more information was needed about the Integration Phase then was available at that time. This lack of information was due to several factors:

The process is relatively new. RUAG has been switching to an Integration Driven Development approach relatively recently. Therefore there are a number of areas where the process is ill-defined. Furthermore adapting and accepting the new process by the users might be hampered if any more changes are added to the process before that is completely understood.

The Integration Phase is complex by nature. By definition the Integration Phase is where all the components of a system are brought together. Faults and imperfections are then ironed out, interfacing and interaction problems have to be sorted out and the final result of the phase is a working system. Since the company produces customized systems, the complexity of this phase is further increased by the changes in requirements and system characteristics from project to project. These changes entail significant modifications in terms of not just Hardware and Software, but also in terms of the way tests are defined and coded, what test equipment and test tools are to be used.

The lack of exact definitions and extensive documentation for the Integration Phase. While coding, review and test procedures are very strictly defined and documented, little documentation on the Integration Phase can be found. The Integration Phase is defined as the period of time when hardware, software, test equipment and test software are all made to function together. Stringent tests at the end of the Integration Phase ensure the quality of the product. There is little record, however, of the kinds of faults met during this Phase, what common problems and solution are encountered, what kinds of steps and resources are needed to ensure the proper and timely delivery of the product.

Constant refinement and updating of the process. While adapting to the realities of software development is a plus for any process, this makes the process somewhat harder to define. Constant experimentation and updating of various practices may lead to information becoming obsolete quite quickly.

From the point of view of the implementation of the VAMOS framework, the need arose for a clear understanding of what the important dimensions were during the Integration Phase, what the information needs of the various stakeholders were and what information was available.

Methods

In order to define the Integration Phase in a clear and detailed manner several research methods were used. The purpose of each method was twofold: first to obtain the maximum amount of relevant information about the Integration Phase. The second purpose was to corroborate any information obtained from other sources, thus both ensuring the accuracy of the available information and giving a different perspective on the same information.

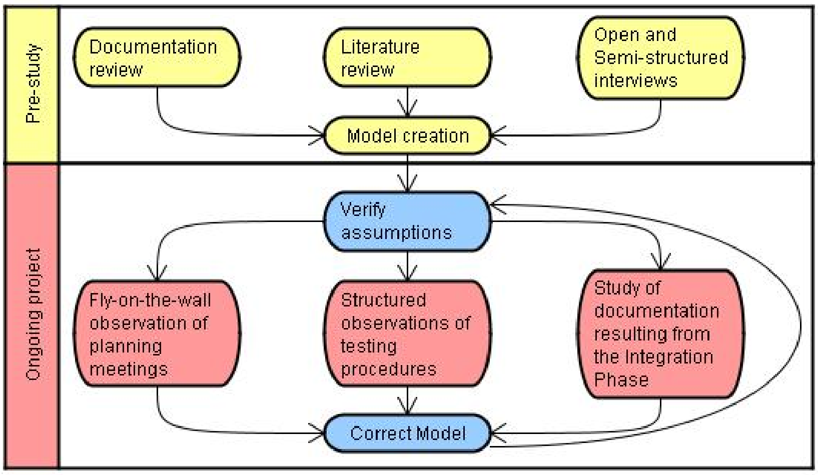

Figure 6. The research process

The research of the Integration Phase was split into two major components:

The first step was to conduct a pre-study. This took place outside of any project. The purpose of the pre-study was the development of a model of the Integration Phase. Assumptions could then be verified and possible improvements proposed, on the basis of this model. The pre-study consisted of:

Documentation review. A review of available documents regarding the official view of the Integration Phase. This included official process descriptions, documents from past projects that had used the approach and other, company specific documents.

Literature review. Available literature on the notion of Integration Driven Developments provided an initial description of the theoretical concepts behind the Integration Phase. While this initial description differed from the situation on the ground, it did provide a useful starting point. Further literature was useful in identifying other research methods, considerations regarding the application of other research methods and a host of other issued. The literature review provided the theoretical backbone of the study.

Open and semi-structured interviews. During the course of this study, 5 semi-structured interviews and a number of open interviews were performed in order to obtain information about the Integration Phase. While theoretical and official documentation existed, an exploratory study by means of interviews proved a useful tool in identifying particularities of the application of this Phase at the company, identifying information sources and information needs that were not mentioned in the official documentation and providing a more detailed account of what goes on during this stage for each of the persons participating in the interviews. While their views were shaped by their own roles in the projects they had been involved in, they provided useful insight into the practical difficulties, problems and solutions demanded by integrating different components into a single system.

Defining the model. Once all this information had been elicited, a model of the Integration Phase was created. The purpose of this was to synthesize available information into a more intuitive format and to create a clearer, more understandable view, while at the same time keeping track of the important factors defining the Phase.

The second step was the verification of the assumptions made. In order do to this, a further study was conducted on an ongoing project, while that project was in its Integration Phase. This study of an ongoing project consisted of:

Verifying assumptions and **Correcting the model**. These were not distinct steps, but rather an ongoing process by which the model and its underlying assumptions were constantly corrected as a result of the observations made and the information obtained during the study.

Fly-on-the-wall observations of planning meetings. To further expand the available information about the Integration Phase and to confirm the degree to which the official planning process fits the reality in the field, fly-on-the-wall observation sessions were setup for the Software daily meetings and the Daily Integration meeting. These sessions were strictly intended for observation, thus no interference was permitted with the meeting. The purpose of these sessions was to obtain more information about the way the Integration Phase and various activities within it are planned on a day-to-day basis, to determine how much information is available and how much information is used during the planning process and to determine the level of information exchanges between participants at the meeting.

Structured observations of testing procedures. Observation sessions were setup to allow the study of testing procedures. This was a contextual inquiry, with the observer actively interacting with the people involved in the testing procedure in order to obtain additional information. This was undertaken in order to provide a clearer understanding of the testing procedures and testing environment, and the limitations they place on the process and measurements that can be implemented.

Study of documentation resulting from the Integration Phase. The Integration Phase does produce a number of documents, used mostly for planning and progress monitoring purposes. These documents are sometimes left out of the official documentation, some because they are the result of internal tools and procedures and others because they are the result of newly implemented practices or practices under test. The access to these documents proved useful in identifying information sources and information availability during the Integration Phase.

Results

The result of all these efforts was a model of the Integration Phase. During the study, the model was refined and updated as additional information became available.

The model describes the Integration Phase in terms of the interactions between several areas of activity. The Software and Hardware areas deal with the design, development and delivery of software and hardware. The Validation area is concerned with the design and development of test specifications, coding the tests, designing and developing the test equipment and the test software. The Integration area is concerned with conducting the integration tests, and coordinating the activities of the others.

The Integration Phase is composed of several Integration Steps. Each such step integrates new modules to the existing system. Thus the process is an iterative process of accumulation of functionality.

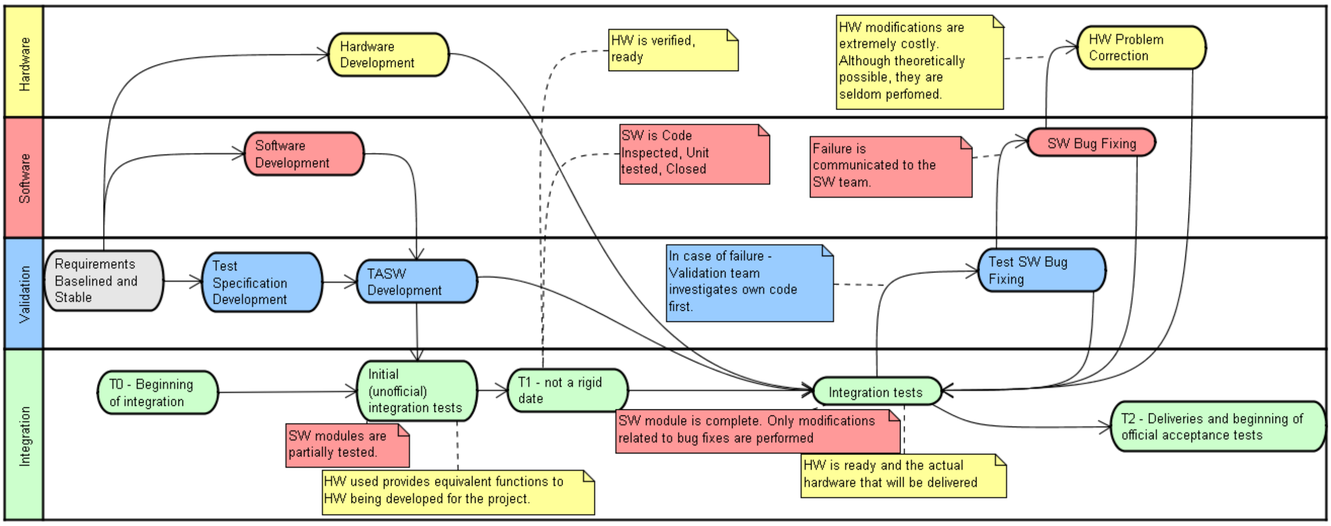

Figure 7. Model of the Integration Phase

At the beginning of the Integration Phase, before any of the integration steps are started several conditions must be fulfilled. The requirements, design and interfaces of the system must be set. While modifications are possible afterwards, mostly due to fault removal, ideally they would be minimal. The anatomy of the system, i.e. the exact components and relationships between them, and the integration order are also decided. The integration order is composed of a series of steps, called integration steps, which describe which modules are to be integrated into the system at that point. Several integration steps may run in parallel or with different time offsets. This is part of the concept of Integration Driven Development.

The beginning of each integration step is marked by a tollgate called T0. After this tollgate the software and hardware components for that integration step are developed. The validation team begins developing the test specifications. The development and review of test specifications may result in changes being made to requirements. Once the test specifications are completed, the tests are coded.

When the software is almost complete and estimated to be stable, development begins on the Test Application Software (TASW). The decision regarding the stability of the software is made by the software team and the information is communicated via informal channels. Changes in software will trigger changes in the TASW at a later date, but it is estimated that the time gained by starting the TASW implementation earlier offsets delays incurred as a result of such changes. When both are completed, the TASW is built together with the software into a single unit.

Several conditions have to be met before the next tollgate. The software had to be ready, unit tested, code inspected and closed. The hardware has to be ready and reviewed.

Once all these conditions are met, the integration step is at tollgate T1. There is no strict deadline or date for the T1 tollgate. It is considered that T1 has been reached when all the previous conditions are met and all departments are ready to move on to testing.

After T1 the Integration tests are conducted. No changes may be made to either software or hardware, unless they are part of fault fixes. The testing is usually performed by the validation team. When a fault is found the investigation starts with the tests themselves and test software. Other departments and the people responsible for the hardware and software components being tested are involved into the fault investigation as it progresses. The fault investigation and the communication between all the people involved is done on an informal basis and results in no documentation apart from commit comments for any changes in code or specifications.

Planning

Integration Driven Development requires a high amount of planning, in particular short-term planning. Since the company's focus was the analysis and improvement of the integration phase, one of the obvious places to start was an analysis of the daily Software Planning Meeting and Integration Planning Meeting. The different departments will be discussed separately, with a focus on the impact decisions made at each level have on the integration process.

The Daily Software Planning Meeting

The Software Planning meeting is a daily stand-up meeting, supported by a dedicated planning board for the project being studied. Attending this meeting are all the members of the software team for the project in question. The whiteboard is used to describe the current status and activities planned. This board is a snapshot of the plan. Planning changes are performed several times a day, by people responsible for the work being re-planned. The more important changes, those that affect deadlines, other activities or tollgates and other important dates are discussed and approved at the daily meeting.

Changes are not recorded, however, and information regarding changes and the reasons behind them is lost. The recording of changes is considered unnecessary, since the purpose of the board is to provide an overview of the current situation.

Activities recorded on the board are fine-grained, detailed estimations of the activities that need to be performed. The planning process also requires that estimated and actual times are recorded for each activity. The process is quite new, though, and detailed estimations are not always made. There is a pattern regarding estimations, with more experienced and confident developers being more likely to estimate the duration of their activities more often. Activities similar to those performed before are also estimated more frequently. Recording the actual times happens less frequently, mostly when there is some discrepancy between the estimation and the actual time. Information is insufficient to draw more conclusions.

A detailed activities list exists and is printed out and displayed next to the board for all the daily meetings. The list is updated any time there is a change in the status of the activities listed, i.e. when activities are added or their status changed. These changes are quite frequent so the Status of activities list is updated on a daily, or at the very least weekly, basis.

Similar planning meetings exist for other departments. Those are performed independently and some differences do exist. Some departments have a single board for all their projects, as opposed to a dedicated board. There are also differences in terms of the level of detail of the work breakdown, with other planning boards containing more coarse-grained information or just the deadlines and tollgates.

The Daily Software Meeting is chaired by the Software Responsible. That person is responsible for the software aspect of the project in question. The meeting itself consists of a brief status report from each team member. Each has to specify what activities they have completed the previous day, what will be done in the current day and to ask any questions they may have. While short, the meeting provides the starting point for more detailed discussions between team members after the conclusion of the meeting.

The Daily Integration Planning Meeting

The Integration Planning Meeting is a daily stand-up meeting, the purpose of which is to coordinate the activities of the various departments. Since the Integration Phase brings together all the components of a system, it is essential that activities are synchronized as much as possible. Attending this meeting are people responsible for key activities and areas from all the departments involved. The departments involved are: Software, Hardware, Test Equipment, Verification and Validation. Some departments may have more than one key activity. In this situation the department will have a person present for each key activity.

The board supporting the Integration Meeting is used for the current project only. Due to the nature of the meeting, the work breakdown is coarse-grained, with little information in addition to deadlines and delivery dates.

The Integration meeting is supported by several documents. The first is the Integration Plan, with is updated daily and displayed on the board. It contains information regarding the timing for the tollgates for the various modules of the system.

The Anatomy is printed out and displayed. This contains information regarding the structure of the system, the modules making up the system and the integration step that each module belongs to.

The Activities list shows the activities planned for at the current time with additional information regarding the activities' status, the person responsible for each activity, the description and links to the time these activities were decided and to any documents describing, supporting or related to the activities. This list is also displayed on the board, and is updated frequently.

The Test status sheet is also displayed on the board. It contains information regarding the status of each test. This document is the basis for an overall view describing the projects' status in relation to the remaining time, the number of tests coded and the number of tests integrated so far.

The meeting itself consists of a brief status evaluation made by the Integration Responsible, who also chairs the meeting, regarding the overall view, timing and planning changes and general issues. Each participant then presents their own status, changes in planning and deliveries, reasons for any delays. If changes in planning are necessary or foreseen, the issues are discussed in detail between those responsible for the areas affected by those changes. Each participant also has a chance to ask questions regarding planning or technical issues. If these questions cannot be answered on the spot, for various reasons, they are then recorded and answered when the appropriate information becomes available.

Overall integration perspective

The Integration Phase has its own particularities with respect to how information is recorded and communicated, as well as what information is needed. Information communication issues will be discussed below, together with information exchange systems used in other departments and their suitability for use with integration.

The Integration Phase is, by definition, the phase where various components of the system come together and interact to perform all the required functionality. This also implies interactions between different departments with different areas of activity and different supporting systems. The activities of all these departments need to be synchronized and, as a result, information needs to be communicated easily within departments.

Each department has solved the problem of internal communication in its own way, using its own system and fulfilling its own needs. From an Integration perspective this means that there is no central hub that collects information from all the departments and makes it available to everyone else. This is particularly important when a fault is discovered, as fault investigations will have to be performed and will involve people from several departments.

Currently the communication problem is solved on an informal basis, by direct contact. This approach is the easiest way to communicate, with no additional systems or facilities needed. It does however have several disadvantages. First of all, the information exchange is not recorded in any way. Thus automating information collection, information collection at all in fact, is quite difficult. Another disadvantage of this approach is that sometimes, when information is communicated across departments, the exact boundaries of responsibility for fixing various bugs become blurred.

The ticket system is a system used by the Software department to record some of the information needed for everyday activity. The system stores items, and information about those items such as the date it was created and the creator, the status and additional comments. While the system can be used for fault tracking, it is flexible enough to allow other types of data to be recorded. In the project currently being studied, the ticket system is used to record activities, their status and people responsible for them and other planning-relevant information.

Adapting the ticket system to record integration-relevant information has several disadvantages. First of these is related to using the system. Currently the ticket system is being used only in the software department. Adopting it to record integration information would mean introducing it to all the departments. Furthermore, the software department itself has no standard format for using the ticket system, with different projects collecting different kinds of information.

Integration Phase information need

In terms of information need, each actor involved needs different kinds of information in order to perform their daily work. In order to better describe the information needs of different stakeholders, two levels have been defined.

The information need within the project defines the information needed by the project team members for their day to day activities. This type of information is usually project related and project specific, as well as being relevant in the short term. The information needs indentified for the project currently under study are listed below.

Estimation and conformance to estimation

For any project, accurate estimations of the time and effort needed for the activities involved are necessary in order to request the needed resources and to ensure timely deliveries. The Integration Phase, in particular, is a part of the project that relies heavily on estimation. Moreover, estimating for the Integration Phase is harder do to, since it’s difficult to estimate the time needed to find and remove an unknown number of unknown faults. Therefore, recording the estimations made and checking the conformance of these estimates to the actual times is an integral part of the information need.

Evaluation of the accuracy of previous estimations

In addition to controlling the Integration Phase conformance to schedule by means of estimation, the estimation process itself should be evaluated and improved. The first step in evaluating the estimation process is to record the accuracy of estimations currently being made. There is an added complication in the Integration Phase. This is caused by the fact that the activities are not known in detail as the Phase begins. An unexpected level of faults, both in terms of number and in terms of difficulty might throw off even the most reasonable estimates. It thus becomes very important if a certain delay was caused by faulty estimation or by factors that could not be foreseen and therefore could not be accurately estimated and accounted for. Once the causes for the delays are being identified, a more accurate evaluation of the estimation process can be made.

Improvement of the estimations are being made is the expected consequence of this evaluation of the evaluation procedure. Once the accuracy of estimation has been evaluated, the benefits in terms of accuracy improvement are weighted against the effort needed to achieve those results. This too is not as straightforward as might be expected. Since these are just estimations and since unexpected faults can throw off even the most accurate estimations, the decision to be made has to take into account additional factors.

Expected fault types and numbers, based on past experiences

Experience offers a lot of information in terms of types and numbers of faults that can be expected. Just like estimates, these numbers are far from accurate, but can provide a level of accuracy that is just enough for practical purposes. Information regarding the types and numbers of faults is available in the Integration Phase. Since this information is not recorded at the time, however, it is hard to draw any conclusions at the end of the Phase. Recording this information, even at a very basic level of accuracy, may provide useful information regarding what faults can be expected for a certain module or project.

Information need external to the project

Information obtained during the Integration Phase can also be relevant on a higher level, outside the project limits.

Process evaluation and improvement

Information obtained during the Integration Phase can help evaluate and improve the development process. Evaluation of the process during the Integration Phase is currently difficult, given the relatively little available information. Once more information is available this Phase, bottlenecks might be identified and so can potential improvements. Given the adoption of Integration Driven Development, potential improvements of the development process during the Integration Phase may have a considerable impact on the overall process.

Validation Activity effectiveness and cost data

One of the key principles of the VAMOS framework is the constant evaluation of the Validation Activities, their effectiveness and cost, and the notion of adapting and changing the selection of Validation Activities being performed to better fit the needs and requirements of each project. At this moment there is little practical information regarding the effectiveness and less still regarding the cost associated with each Validation Activity. From a perspective external to any project, data regarding these two areas could greatly improve the understanding of each Validation Activity, its strengths and weaknesses, as well as the cost, effort and practical considerations. Estimation and evaluations that are now purely theoretical could benefit from the strength of practical application and experimental results.

Fault number, type and cost data

This type of information is needed outside the project as well as within the project. Within the project, this information is important for evaluating the current status and identifying problems at the earliest opportunity. Outside the project, this type of information is useful as a means of identifying recurring problems across projects, finding areas related to faults already identified and that could benefit from existing solutions.

Conclusions

The analysis of the current process being used for software development at RUAG has a number of limitations. First of these is that the analysis was focused on a specific phase of the process, rather than seeking evaluate the entire process. This means that conclusions regarding information availability and needs are not really applicable in other areas of the process. As stated in the description, other phases have means of detailed fault reporting and data centralization.

The second limitation is that the analysis was focused on the Integration level information availability and needs and was undertaken with the needs of the VAMOS implementation in mind. This focus means that the analysis is suited for the needs of this implementation, but is not a complete description of the process.

In spite of these limitations, a large amount of useful information came out of this analysis. The results were vital in selecting the appropriate options in terms of measurements as well as selecting the appropriate criteria to describe and prioritize those measurements. Since VAMOS emphasizes the need to tailor the solutions being applied to the needs of the company or organization where they are applied, similar analyses must be performed every time the framework is to be implemented. Furthermore, information needs might change, both due to the new information becoming available as a result of the measurements and due to changing priorities and focus during the regular activity of the company. For this reason, regular analyses of the current processes and they extent to which the information available as a result of applying the process fits the information requirements are a useful addition to any measurement and process analysis and improvement programme.

Chapter 3 - Code Inspection Sheet Analysis Tool

Introduction

Code inspections are a method of code verification involving the use of static analysis tools and manual review to ensure the quality of the software being produced. This type of verification can be performed early in the process and defects discovered at such an early stage are easier and cheaper to remove.

Code inspections are performed in a structured manner, defined by a Code Inspection Sheet. This document describes the steps that are to be performed, the tools that are to be used and the types of faults that the reviewer seeks to identify. The results of each of these components, both the use of analysis tools and the manual analysis of the code are recorded in the Code Inspection Sheet, together with comments and the decision of whether or not the code has passed or failed the code inspection. Code inspections are also part of regression testing, since any modifications to a software module triggers a new code inspection. This results in a large quantity of code inspection sheet documents being produced for each project. The tool described in this chapter seeks to centralize the information available in these documents and present it in a way useful to decision makers within the company.

Context

This tool was developed in the context of the application of the Framework for the Management and Optimization of Verification and Validation Activities (VAMOS) [1] at RUAG Space Sweden AB. The framework was developed as part of a Master Thesis at RUAG and validated on historical data. It was decided that a prototype implementation would provide useful data about the application of this framework, the difficulties encountered during implementation as well as potential modifications and optimizations to the framework itself.

The VAMOS framework seeks to describe a means of managing an optimizing verification and validation activities. One of the main concepts that the framework relies on is that of fault slip-through, i.e. the concept that certain go uncaught at the appropriate time. The later discovery and fixing of these faults is more costly, due to impact on the rework of other modules and the costs associated with performing regression testing. A solution to this problem is the specialization of the Validation Activities, with each activity focusing on a particular kind of fault and aiming to identify as many as possible of those faults. The faults that each Validation Activity is to focus on are selected depending on the cost of fixing and regression testing, the importance and seriousness of the fault and costs associated with allowing the fault to slip through to later stages.

During development at RUAG Space Sweden AB, each module is unit tested and then code inspected, based on the application of certain tools and visual inspection according to a list of items that are of interest (referred to as checkpoints). In order for a module to pass Code Inspection, the code must fulfill all the conditions, i.e. it must pass all applicable checkpoints. If any checkpoint is failed, then the module is sent back to the developer that wrote it, the problems found are fixed, the module is unit tested again and then code inspected again. The checklist for the code inspection is determined at the beginning of the project and it forms a template. Each code inspection instantiates the template and results in a Code Inspection Sheet, where each of the checkpoints in the template is marked as Passed, Failed or Not Applicable. For a code inspection to be passed all checkpoints must be Passed or Not Applicable.

There were two major concerns about this stage. The first concern was that the Validation Activity found and rejected modules based on faults that were not serious enough to warrant a redo of the unit testing. This is to say that the Validation Activity found considerable numbers of "beautification" or documentation faults. These are text format, readability or comment faults that, although requiring fix and important for documentation and maintenance purposes, do not require a repeat of all the unit tests, since no functionality is modified as part of the fix. Rejecting a module based solely on readability faults meant that time and resources were spent redoing unit tests and code inspections without need. While Code Inspection Sheets do contain that information, there are several sheets per module and a number of modules per project, so an automated way to perform the verification was needed.

The second concern was that the Code Inspection procedure may catch faults slipping through from previous phases. This was considered both a concern and an opportunity. It was a concern because design and requirements faults caught at this stage had already caused considerable loss in terms of wasted effort on inappropriate modules. It was considered an opportunity because, although there was time lost to catch such faults in Code Inspections, it was certainly cheaper than letting them slip through further. A more detailed analysis of the Code Inspection Sheets may have yielded the appropriate information, but the same problems applied as before. The large number of Code Inspection Sheets, particularly when trying to analyze entire projects or more than one project, prevented this evaluation from being done by hand.

Reasoning

Literature regarding Code Inspections seemed to lend credence to the fact that most faults found during this stage are readability and so-called "beautification" faults. According to some sources [3] up to 75% of faults found were of this type. Previous efforts with this type of approach yielded similar results. An analysis conducted during the development of VAMOS[1] also indicated that large numbers of beautification and readability faults were found, lending even more weight to the arguments.

Concerns at the company regarding the Code Inspection procedure were mostly connected to cost of repeating the unit tests and code inspections rather than with the cost associated with correction of the faults themselves. This is mostly due to the fact that readability and beautification faults are quickly and cheaply fixed.

This mismatch between the research results available, showing the prevalence of minor faults in terms of ratio to normal faults, and the focus of the company on the cost associated with regression, retesting and repeating the code inspections lead to the consideration of a different approach to the code inspection sheet issue.

Each code inspection sheet identifies a number of faults, a lot of them of the same type. This was considered an interesting result, considering that several readability and/or beautification faults could exist in the same sheet. In practice, the regression cost associated with retesting and redoing the code inspections would only have to be paid once for all those faults. Moreover, since the failure of once checkpoint leads to the failure of the entire code inspection and to repeating the entire process, those minor faults may not even impact cost at all. This insight led to a new approach to evaluating code inspection sheets.

Rather than evaluate the prevalence of readability and beautification faults in terms of their ratio to total number of faults, the new approach was to evaluate these faults based on their effect on the cost regression, retest and repeating the code inspection. Since only a failed checkpoint can lead to incurring additional cost, the decision was made to check how many times each checkpoint was failed. The reasoning behind this decision was that only failed checkpoints cause the rejection of a module and so, in terms of costs, only rejected modules are significant. This decision changed the entire approach to solving this question.

Development and architecture

The Code Inspection Sheet Analysis Tool had as inputs the Code Inspection Sheets of one or more projects, the expected outcome was a list of the checkpoints in those projects and the number of times each was passed, failed or not applicable. The input Code Inspections Sheets are text documents, identical in form, thus making processing them considerably easier.

The first step was the development of a model for the tool. The thought was that the tool will be flexible enough to allow both the expansion of the initial prototype to fit potentially changing requirements as well as allowing for the subsequent creation of related tools, potentially re-using parts or modules of the prototype, or providing the basis for an entire family of tools of related functionality.

The model

Since the purpose was to count and evaluate the state of checkpoints, the model of the system was to revolve around the notion of Checkpoint. The tool would build-up a library of checkpoints it had encountered, and use that to compare against the checkpoints in the project currently under analysis. The goal of this approach was to limit the number of duplicates that would result from slightly different formulations of the same checkpoint. For each Code Inspection Sheet processed, details such as the date, the module it referred to, headers or other files linked to the current module would be stored. This information would then be available for display.

The model was to provide a maximum degree of flexibility, by separating those parts of the model that will have to change in order to adapt to other requirements or to changing inputs. Thus, the Module actually processing the input files was left out of the model altogether, since that will change every time the input format changes. Other modules that will require modification were also separated within the module. Notions like the Checkpoint Library were added to the model even though the initial prototype would not make full use of their possibilities, mostly due to the interesting development options they opened.

The prototype

While the architecture of the prototype tool could be easily derived from the model being proposed, decisions would have to be made about the instantiation of the model. One of the primary decisions was that of the programming language in which the tool prototype was to be developed. Given the particularities of the model, an object oriented language was considered to be best fitting for this application.

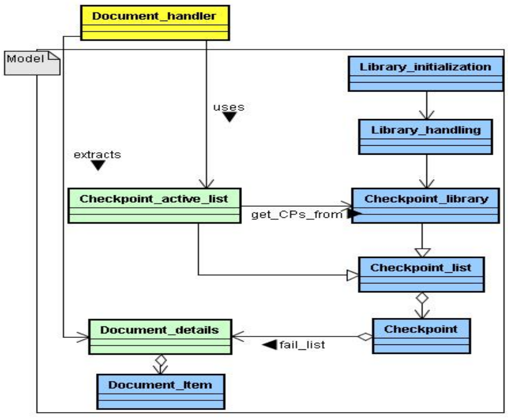

Figure 8. The Model of the Code Inspection Sheet Analysis Tool

The language chosen was Ruby, partly due to it being an object oriented language, and partly due to the relative ease of developing and of interacting with the environment. Given the large number of interactions with the file system and various information sources in various formats, the ease of use of this language was deemed an important benefit.

Storing the results of the analysis was to be done in the form of marshaled Ruby objects, pending a decision regarding what form should they be displayed in to the user. Once this decision was taken, necessary information would be displayed, but the underlying Ruby objects were still to be preserved. This was to provide the developer with flexibility regarding the information to be displayed, while at the same time give access to all the information stored, should that prove necessary or useful.

The prototype implementation is thus a set of Ruby classes. It takes its input from an .xls file containing the folders where the Code Inspection Sheets for the projects to be analyzed are stored. It then opens each of those files in turn, perform the required analysis and move on to the next. Results are stored in the form of marshaled Ruby objects and displayed in an .xls file.

The results file contains:

- the date and time when the analysis was run
- the number of directories and files processed
- the list of all the software modules considered in the analysis. These correspond to the files given as input, but not to the projects that are part of the analysis, since a single project may have multiple modules that are analyzed separately.
- the list of all the non-duplicated checkpoints. For each of these the tool displays the number of times the checkpoint was passed, failed and not applicable, the modules that were failed, the folder and file that was failed.

Analysis and discussion

Analysis of the results

The results obtained by the use of the Code Inspection Sheet Analysis Tool required some analysis before they could be used to address the problems that had lead to the creation of the tool. Below is a table of the results, showing the distribution of rejections across the checkpoints in terms of percentage of the total number of rejections.

Table 1. The Code Inspection Sheet Analysis Tool Results

Checkpoint	Percent of all rejections (%)
Check that the object implements all requirements traced to it from [SRS], no more and no less. This includes checking that module and operation 'What' descriptions are consistent with the requirements traced to the module.	6,8
Check that the code is compliant to the design according to the [SDD]. This includes checking the 'What' and 'How' descriptions and the implementation constraints themselves, both for modules and operations. This also includes general design requirements and guidelines from [SDD] which is applicable to all modules.	0,8
Check that the external interface and the internal interface are consistent, including descriptions.	1
Check that the identified usage constraints are correct and sufficient. Especially the following questions shall be answered: Is the operation re-entrant or not? Can the operation be called in parallel with any other operation? Note: This checkpoint only applies to internal BSP interfaces, since RTEMS dictates the external interfaces for the BSP. However, the BSP patch must not alter an external interface or its usage constraints.	6
Check that the usage constraints of used objects and operations have been followed and are properly resolved or inherited by the using object and/or operation.	7,8
Check that appropriate data structures and algorithms have been chosen with respect to required accuracy and performance, and that no actions and information is unnecessarily duplicated. Has the simplest algorithm been chosen?	1,6
Check that constraints identified in [HSI] are not violated.	2,4
Check that resources and critical regions are handled in a correct way. Are deadlocks avoided? (E.g. is more than one resource locked at a time (OK if always locked in the same order), is a process waiting for an event, i.e.	1,8

semaphore, FIFO, etc., where it locks out the signalling process).	
Check that the MISRA 2004 coding standard is followed. Use separate check list below.	1,8
Check that the Internal rules of the coding standard are followed. Use separate check list below.	3,9
Check that the applicable coding standard for assembler is followed.	0,2
Check that the C and/or Assembler code passes the compilation without any unjustified warnings. Note: The only warning that must not be justified is for using designated initializers, see [SDP].	0,2
Check that the C code passes FlexeLint without any unjustified warnings.	4,9
Check that the C code passes Splint without any unjustified warnings.	1,3
Check that the C code passes RuleChecker without any unjustified warnings.	0,7
Check that the C code metric limits are not violated without justification.	2,4
Other findings (e.g. bugs, inconsistencies or coding standard violations not covered by other checkpoints)	26,9
Check that only used interrupts are unmasked.	0,2
Check that all operations are re-entrant if feasible.	0,3
MISRA Rule 8.7: Objects shall be defined at block scope if they are only accessed from within a single function.	0,5
MISRA Rule 10.6: A "U" suffix shall be applied to all constants of unsigned type.	1,5
MISRA Rule 11.4: A cast should not be performed between a pointer to object type and a different pointer to object type.	0,3
MISRA Rule 16.9: A function identifier shall only be used with either a preceding & or with a parenthesised parameter list, which may be empty.	0,2
MISRA Rule 16.10: If a function returns error information, then that error information shall be tested.	0,7
MISRA Rule 17.4: Array indexing shall be the only allowed form of pointer arithmetic. Note: Array indexing for pointers are allowed, see [SDP].	0,3

MISRA Rule 19.4: C macros shall only expand to a braced initialiser, a constant, a parenthesised expression, a type qualifier, a storage class specifier, or a do-while-zero construct.	0,3
MISRA Rule 19.7: A function should be used in preference to a function-like macro.	0,2
MISRA Rule 20.2: The names of standard library macros, objects and functions shall not be reused. Note: NULL is allowed, see [SDP].	0,2
Internal rule 200: Abbreviations	1,1
Internal rule 300: Indentation	0,8
Internal rule 402: Module implementation prologue	0,3
Internal rule 406: Code comments	1,1
Internal rule 901: Visibility	0,5
Check that the code is compliant to the design according to the [SDD]. This includes checking the 'What' and 'How' descriptions and the implementation constraints themselves. This also includes general design requirements and guidelines from [SDD] which is applicable to all modules.	11,6
Check that the pseudo-code is followed in the code.	0,2
MISRA Rule 1.2: No reliance shall be placed on undefined or unspecified behaviour.	0,2
Internal rule 405: Local operation prologue Check parameters for reasonable descriptions	0,2
Check that comments are correct and sufficient and not redundant with the code. (R/Com.Gen.5, R/Com.Gen.6)	0,2
Check that the provided interface is minimised. (R/Pgm.Visibility.1) (Note: Extra symbols might be needed for test purposes.)	0,2
Check that the code is compliant to the 'What' and 'How' descriptions.	0,3
Check that the identified usage constraints are correct and sufficient. Especially the following questions shall be answered: Is the operation re-entrant or not? Is the complete range of the input parameters valid? Is the output parameters always valid Can the operation be called in parallel with any other operation?	0,7
Check that operations stated to be re-entrant (according to the usage constraints) are implemented as re-entrant.	0,5
Check that all parameters have adequate parameter descriptions.	0,3

Check that comments are correct and sufficient and not redundant with the C-code.	1
Check that the layout and the typographical and lexical convention are according to the coding standard, the glossary and the [SDD].	0,2
Check that the provided interface is minimised, i.e. only types, constants and operations that needs to be exported shall be included in the specification.	0,3
Check that no unnecessary objects are included, i.e. an object should only be included when a declaration in the object is used directly. Also check if it is sufficient to include the object only in the body and not in the specification.	0,5
Check that constants are used in favour of numerical literals and that the constants are assigned correct values. Check that constants declared inside operations are static. Check that all const data is located in the default section in RAM.	0,3
Check that all expressions are easily understood (e.g. operator precedence, lazy evaluation, nesting depth).	0,3
Check that the C code passes SE_Tools without any unjustified warnings.	1
Check that the C and/or Assembler code passes the compilation without any unjustified warnings. Note: Warnings present already when compiling the original code need not be justified, but should be noted here.	0,8
Check that the layout and the typographical and lexical convention are according to the coding standard, the glossary and the SDD.	0,5
Check that static variables not initialised at declaration are initialised in a initialisation operation. Also check that there is a usage constraint that ensures that this operation is called before any operation using the static variable.	0,3
Check that the object implements all requirements traced to it from the SDD and the SRS, no more and no less.	0,3
Check that the ICD layout and content is consistent with the module specification.	0,7
Internal rule 405: Local operation prologue Check parameter description	0,7
Total	100

First question to be answered was the degree to which modules are being rejected based on readability and beautification faults, rather than faults affecting the functionality of the code. To provide an answer to this question, two types of checkpoints were of main concern.

First there were certain checkpoints that were obviously concerned with readability, documentation and "beautification" problems (e.g. indentation concerns). Overall modules rejected for these reasons were found to be about 10% of the total number of rejections. This finding supported the decision to look into code rejections rather than absolute number of faults. While in terms of absolute numbers, such faults are prevalent, they are seldom serious enough to warrant rejection.

The second type of checkpoint of concern was one labeled "Other". It was feared that this checkpoint was the disproportionately rejected and, due to the unclear definition of the types of faults that fit under it, the faults that caused the rejections were largely cosmetic in nature. In terms of numbers of times this checkpoint was failed, it seemed that these fears were largely justified. While the situation was nowhere near as serious as previously thought, in some cases up to 25% of the modules being rejected also had a rejection for this checkpoint. To further analyze the modules being rejected partially due to this checkpoint, a list of those rejected modules was obtained from the Tool. Some 30% of the Code Inspection Sheets that had failed this checkpoint, randomly chosen, were then analyzed by hand. The purposes of this analysis were: to determine what kinds of faults cause this type of rejection and to determine whether or not this checkpoint was the only one causing the rejection of any of the code inspection sheets. The results of this analysis were, first, that all the faults falling under this category were cosmetic. Since all functionality faults fit under other categories, all cosmetic, readability and most of the documentation related faults fit under this category. This seemed to lend weight to the idea that code was being rejected based solely on cosmetic reasons. This further analysis, however, also determined that, out of the randomly picked sample of 30%, all had other checkpoints failed as well. This meant that while that module was being rejected for cosmetic faults, it was also rejected for functionality faults as well. Not a single example was found of a module being rejected for reasons relation solely to readability or beautification of the code.

The results are based on an investigation of about 1000 code inspection sheets, in 14 separate modules. The modules are part of 6 projects, varying in complexity. The projects taking part in this study cover the range of products that RUAG produces, in order to avoid potential biases introduced by certain kinds of projects.

The second question to be answered related to the degree to which faults introduced in the requirements and design stages could be found during code inspection, thus limiting their slip-through. This was considered useful because, if achieved, it could limit the number of faults of that nature found during integration. Since faults found during integration incur considerable cost in terms of fixing and regression testing, it was considered worthwhile investigating a better way to discover such faults. While there are checkpoints in the Code Inspection Sheets that relate to previous documents, mostly the Software Design Document (SDD) and the Software Requirements Specification (SRS), it was discovered that their formulation reflected more the degree to which the current module is in accordance with those documents, and less the correctness of those documents themselves. To the purposes of implementation and code inspection, the SDD and the SRS were considered to be correct. Since no question was raised as to the correctness of these documents, it was thought that faults in there documents were harder to catch at this stage. Moreover it was thought that, due to the level of detail that the implementation and code inspection stages work on, it is also theoretically difficult to discover design and requirements faults. A brief literature review yielded little more information about the possibility of discovering design or requirements faults at the implementation or code inspection levels. While this is a matter that has not been researched in at any length here, it was considered that, to the purpose of this study, too little information is available to draw other conclusions than the ones presented here.

Overall the results showed that some improvements can be made in terms of classification of code inspection faults, particularly related to the treatment of beautification and cosmetic faults. The lack of prospect for significant improvements, particularly in terms of fault slip-through, did however suggest that the benefits that are possible are not enough to justify deep changes in the process of code inspection, particularly considering the disturbances those changes would cause on the current process. While the result of this analysis is mostly negative, confirming the suitability of the current process rather than finding dramatic improvements, it does serve as an example of how information need is the driving force behind measurements and the tools performing those measurements.

Analysis of the tool

The tool presented here is a prototype implementation. It was designed to fulfill a very specific role in a given context. That having been said, the model behind it seeks to offer the maximum amount of flexibility and to provide room for expansion of the concept. Three potential expansion ideas will be presented and commented here.

First, the checkpoint identification problem is one that can be improved upon. In the prototype implementation, checkpoint identification is done by means of the description of that checkpoint. This approach was dictated by the current situation and is in no way the only available approach. The problem with this style of checkpoint identification is that checkpoints differing only in name and not intention or meaning are treated as different checkpoints. A solution to this issue is the adoption of a checkpoint ID number, which would form the basis of the differentiation between checkpoints, regardless of their description. This would ensure that only checkpoints with different meanings are assigned different IDs and thus minimize duplicates. This facility is supported by the prototype implementation as well, but not currently used.

Second, the Checkpoint library offers the possibility for expansion. The checkpoint library stores all checkpoints processed so far by the tool, so it could form starting point for a Code Inspection Sheet Template Generation Tool. The users would have the possibility to add and remove checkpoints from the library and then use existing checkpoints to create other Code Inspection Sheet Templates. This would help improve standardization across different projects in terms of Code inspection sheets as well as minimize checkpoint duplication even further.

The third potential expansion idea, emerging from discussions with employees of the company, was the possibility of refining the Checkpoint library to allow updating of certain checkpoints. This would be an expansion of the Checkpoint library enabling it to analyze historical data, while relating that data to current classifications. While the expansion is not a clearly crystallized idea, it would involve links between various checkpoints and allow linking results from various versions of a certain checkpoint to that checkpoint's latest form. It could bridge the gap between historical data and current data classifications and remove some of the barriers in analyzing historical data.

Validity threats

As discussed earlier, a certain amount of analysis is required in order for the results to be meaningful. This analysis is performed based on a considerable amount of assumptions regarding the amount of time and effort it takes to remove each fault, as well as the relative importance and impact of each fault. These assumptions may be misleading or may have been misunderstood thus leading to a certain inaccuracy in the results. Given how the results are not meant to be extremely accurate, but rather an estimation, this is not a major concern.

A more serious concern is that of the type of information being recorded. Right now the tool records the failure involving a certain category of faults. The exact number of faults that have led to that failure is not recorded. Particularly for more difficult problems, requiring more time and effort for removal, this may lead to a considerable loss of accuracy. The assumption made in this case was that, for a given problem, the actual cost is incurred due to the time and resources needed to indentify the problem itself and to perform the necessary regression testing, not the fault removal itself. It is conceivable that, in certain cases, fault removal costs may increase and outweigh the other components. This problem, coupled with the fact that the exact number of faults in that category may be unknown, may produce misleading results.

As stated earlier, the purpose of this tool is to provide an estimation that can be used for decision support. Thus, the emphasis was more on providing "good-enough" information in a fast and automated way. The definition of "good-enough" information is very hard to pin-point in any accurate manner, which makes this objective rather more undefined. Overall, the results given by the tool were considered appropriate, given the information needs and even considering the possible inaccuracies. The results were considered "good enough" in this case and for this company, although caution is needed if this tool is to be applied in a different setting. While the implementers of the tool can provide the necessary information regarding the limitations of the tool, the ultimate decision regarding the acceptability of the assumptions being made lies with the decision makes in the company.

Conclusions

The Code Inspection Sheet Analysis Tool provided useful data both in terms of the immediate questions it sought an answer to and in the wider scope of the VAMOS implementation.

In terms of immediate answers, it provided useful information regarding the performance of code inspections, the faults found as a result of this Validation Activity as well as providing more confidence that the code inspection activity is cost-effective and does not reject code solely due to readability or cosmetic issues.

In the wider scope of the VAMOS implementation the Tool provided information regarding to several areas. First and foremost it provided useful information regarding the cost and effort required for the implementation of a measurement tool and limitations of such tools due to available inputs. Moreover, the change is the approach to solving the Code Inspection problem did serve to show that the way a question is asked determines the kind of answer obtained. This insight will serve later to show the importance of determining information need in a clear and accurate way. While many of these lessons only gain importance when viewed with full knowledge of the difficulties encountered during the implementation of VAMOS, they serve as a useful glimpse into the difficulties that any such tool implementation is likely to encounter.

Chapter 5 - Experiences implementing VAMOS

Attempted implementation

In order to validate the VAMOS framework, the decision was made to implement it in part of RUAG's process. The decision to limit the validation to one stage was caused by several considerations, the most important of which was the assessment of the cost and risk entailed by the "big-bang" application of an untested framework. Therefore it was decided that a pilot implementation on a part of the process would provide the necessary data to validate the framework in an active environment, would help iron out any problems met in the application of the framework as well as providing confidence in the further expansion of the framework implementation.

The choice of stage where the framework was to be implemented was made by the company, based on their own evaluation of where additional information would provide most benefit. The phase thus chosen was the Integration Phase.

The VAMOS framework relies on the existence of a set of VAs in the process it seeks to improve. Moreover, the VAs that VAMOS assumes to be in place also have to provide at least some level of information regarding fault classification and removal and regression cost. Although these requirements are understandable in terms of the proper functioning of the framework, there is no clearly defined set of actions to be undertaken if this information is not available or if the VAs are not in place.

The "Improve" step of the framework bears the responsibility for deriving meaningful improvements, quantifying them, and then deciding on which to implement [2]. It is thus assumed that the goals that guide such decisions are clear at all levels, correspond between different levels and are well disseminated throughout the organization. This was deemed a dangerous assumption to make. It was decided that a detailed analysis of the Integration Phase was needed in order to determine:

- **The exact nature of the process**. Given the complexity of Integration in general, and in particular, the Integration Phase as implemented at RUAG, more information was needed before any modifications could be successfully considered.

- **The information need at various levels**. It was recognized early in the exploratory analysis of the Integration Phase that different stakeholders have different information needs. Thus, all these needs had to be defined in order to have a clearer view of what stakeholder would have an interest in which particular VA.

- **The information sources available**. Since imposing a top-down decision model on the choice of VAs would entail significant change in the process, it was decided to determine what information is already available. Since process changes were already underway, and the process itself was relatively new, information sources were not known by all stakeholders involved. This information was deemed necessary before any changes were to be required.

A more detailed description of the analysis of the Integration Phase, as it is performed at RUAG, the methods used for the analysis and the results that were obtained can be found in the separate chapter dealing with that issue.

Problems with the implementation

Several problems were encountered while implementing the VAMOS framework to the Integration Phase. Some of these were severe enough to suggest changes that could be made to the framework itself and perhaps a change in some of the underlying principles of the framework.

One of the first problems encountered relates to the company's goals, resources and priorities. The framework describes a "Define" phase where the company's goals, resources and priorities are defined. The results from this phase are then used to select the most appropriate validation activities and to optimize those validation activities that are to be implemented. The company's goals are seen as being fairly stable, well known and shared at all levels of the company. The information need within the company is also seen as being unaffected by time and additional information becoming available. The information obtained during this phase is also regarded as being sufficient to make decision at a later stage about other validation activities.

Goal and priority stability

The analysis of the Integration Phase, the context where this first step of VAMOS was to be applied, showed that some of these assumptions are not as reliable as previously thought. The first aspect of this problem is that of priority and goal stability in time and with respect to additional information. Initially, the main focus was on the Code Inspection Validation Activity, its efficiency, cost effectiveness and the types and distribution of faults found. As the initial results of the Code Inspection Sheet Analysis Tool became available, the focus changed. These results showed that the Code Inspection Validation Activity was an effective and efficient tool and, while there was some scope for improvement, the benefits that could be gained from assigning more resources to this area were less than potential benefits elsewhere. Moreover, potential improvements in this area were not likely to affect other validation activities, so any changes would be highly localized and unsuitable for generalization to other activities. Thus, as this information became available and was validated by other information sources, the priorities of the company changes accordingly.

It becomes apparent then, than the priorities and goal of the company might change, either with time or as a result of information becoming available, or as a result of changes in process, market or any other factors. It also becomes apparent that implementing new validation activities, measurements and process changes is likely to provide the kind of information that would change a company's goals and priorities.

A solution to this problem is to consider the change of goals and priorities as a natural occurrence during the course of a company's activity, rather than a constant or an exceptional event. The "Define" phase, that aims to define a company's goals and priorities, turns from an initial step that is outside the regular operation of the framework, into an integral part of everyday activity.

The first way of integrating the "Define" phase into the day to day activities of the framework is to move the phase as a whole, as it is currently defined, as a fifth step of VAMOS. This approach would ensure that regular evaluations of the goals and priorities of the company would keep the framework up to date with any changes and would keep the evaluations and recommendations of the framework relevant to the company.

This approach does, however, have a number of drawbacks. All the steps in the framework relate to the validation activities, while the "Define" phase would have a completely different target.

Although the other phases depend on the company's goals and priorities as determined here, there are few other links, either in terms of requirements, methods or results. In addition to the conceptual problems, performing such and evaluation of the goals and priorities with every iteration of the framework would be costly and inefficient. Changes in goals and priorities are important and need to be taken into account, but they happen less regularly than optimizations to the validation activities. Once a set of validation activities are set in place, minor optimizations would be a frequent occurrence whereas changes of goals and priorities would be an exceptional event.

The second option would be to include the "Define" phase in the "Analysis" phase. This would link the analysis of the results of each validation activity with the goals and priorities and the process by which they are elicited. While this removes the conceptual problem, the cost of re-doing the analysis of goals and priorities is still high.

The third option is to create a separate process of definition of company priorities and goals that would run parallel to the main steps of VAMOS. This would ensure that any changes or updates in terms of goals and priorities are taken into account, analyzed and factored into the framework, while reducing the activities of eliciting this information for times when changes actually occur. Once an initial definition of the company goals, priorities and processes is performed, any changes can be described with respect to the initial view. Thus, a full re-evaluation is needed much less often, and costs are reduced even further, without compromising the framework's suitability and relevance to the company specific factors.

Goal uniformity throughout the organization

One of the most obvious results of the analysis of the Integration Step is that there are different goals and different information needs at different levels in the organization. Thus, it is unrealistic to assume that a single set of "Company goals, priorities and information needs" can be used as rules to select the most suitable validation activities and to perform optimizations.

At different levels, different decision makers have different information needs. A simple example that came forward as a result of the Integration Step analysis is the difference between the information need of managers within the project and that of managers outside the project.

Within the project, the most important kinds of information are those related to timing, schedules and deadlines. Decisions taken here have an immediate impact only on the current project. At this level it is important to have up to date information regarding delays or possible delays, reasons for the delays and how these reasons impact other activities within the project and their adherence to schedule and estimation regarding problems that can be foreseen and planned for. Reasons for delays are important only to the extent to which they impact other activities within the project or can be planned for and avoided in later stages of the same project. At the project level, the emphasis is on timely availability of the information rather than its accuracy. Thus, at this level, it is considered more useful to have a rough estimation of foreseeable delays as soon as possible so counter-measures can be enacted. At this level, changes occur very quickly, so this creates a fluid work environment where a lot decisions are based on estimations and assumptions. Timely availability of information is, at this level, more relevant than high accuracy.

Outside the project, the situation is reversed. Outside the project the focus is on decisions regarding process and overall decisions that influence all projects. The focus thus shifts from scheduling and timing information to the problems being encountered the degree to which the existing process and validation activities can successfully solve these problems in a timely and cost effective manner. Since the decisions here have an impact on several projects, it becomes much more important to thoroughly understand the situation and applying a correct solution to existing problems. Time pressure is reduced and the accuracy of the information upon which decisions are based becomes much more important. The information need outside the project is shaped by the need to improve the existing process, making it more cost effective and better suited to any project. Thus, a clear understanding of the problems being encountered and their causes and an accurate description of their impact upon the company's activities becomes a priority at this level. Most of this data is collected and summarized after the projects have ended, when all the data is available and conclusions regarding the effectiveness of various measures have been drawn. The environment, in this case, is much more stable and quick decisions are less important.

From the point of view of the implementation of VAMOS, both of these levels are significant. The outside-project level is where decisions regarding the process itself are made and the validation activities are selected. Information need at this level determines what types of validation activities are needed for both quality assurance and a good understanding of the process. Given the broader scope of decisions made at this level, good practices can be adopted on a wider scale. A wider application will yield better results and more data that can be used to optimize both the process and the selection of validation activities. This is also the level at which management decisions, especially regarding resource allocation are made. These decisions have to be supported by accurate and relevant data, thus the information need at this level has to be taken into account by the framework.

The project level decisions and information needs are far more localized. This level is, however, of vital importance for the framework for two distinct reasons. First of all, all the information to be collected has an immediate impact. Data collected at this level directly affects the planning of various activities within the project on a short-term basis. Thus, the validation activities, the way they are planned and executed and the resources assigned to each, are directly influenced by the information being collected at this level. The information need at this point, and especially the focus on speed and availability of the information rather than accuracy, means that the optimization options that VAMOS recommends need to be suited to these priorities.

To address this issue, the phase of the VAMOS framework that defines the information need and priorities of the company should be conducted such that it ensures that the information need at all levels is properly defined. For this purpose, the information sources need to be considered, such that they are not biased towards one of the level. For example, if interviews are to be conducted to determine the information need, then the interviewees need to be both management and project leaders, in order to cover both areas.

A set of tools for the "Define" phase of VAMOS could also be defined. While not all companies can agree to all these tools and not all tools are appropriate for every type of setting, it would offer implementers a starting point. Such a "Research toolbox" can be updated and optimized itself, with more information being added with respect to the effectiveness and limitations of each of the tools being described.

Issues regarding acceptance and reliability

An issue that needs to be taken into account in any implementation of Validation Activities, and particularly any form of measurement is that of user acceptance and reliability of the data collection process.

Many factors influence user acceptance of certain measures. While not all factors can be planned for, some issued need to be taken into account before any validation activity or related measurement can be considered. In “Understanding developers’ resistance to software metrics” [6], the motivation associated with a software metric is characterized by 4 types of factors:

- Ease of use
- Attitude
- Usefulness
- Control

The VAMOS framework describes a Measurements Options Model, the purpose of which is to find the benefits and drawbacks of each measurement being proposed and to enable a choice between them. The measurements included in the MOM are currently defined and ranked according to the level of effort they require, with distinction made between the fixed effort of setting up a measurement and the effort associated with the regular collection of data.

The evaluation of effort required only partly covers the issue of ease of use. In addition to concerns regarding the costs associated with each metric, other factors need to be taken into account. The first of these is the level of analysis that the results need in order for them to be relevant. A high level of analysis, performed by specialists, would add significantly to the costs associated with a measurement, while not adding to the costs of the measurement itself.

In addition, some measurements require significant process changes before they can be implemented. These changes may result in an increase in the costs of establishing and using a measurement or validation activities, an increase that does not depend on the item itself. These factors cannot be evaluated before the company’s current process and activities are known. They are however quite relevant in terms of acceptance and cost. While not strictly connected to each of the validation activities, there factors are essential when the decision to implement that activity is taken. Improper analysis of these factors may increase the cost significantly, while reducing the relevance or acceptance of each of the measurements.

The accuracy evaluation of each of the measurements in the MOM is related to the Usefulness factor in the above. While an evaluation of accuracy is useful and necessary in such a decision, it is not quite enough for an informed decision. Related to the issue of accuracy is the cost of increasing or otherwise changing the accuracy of the measurement. For example, a more flexible measurement, that can improve its accuracy when required and at a lower cost may be preferable. This is especially true for situations where the measurement is used for an exploratory study of certain aspects of the company’s activities or when the initial information is insufficient to formulate detailed information requirements.

Also related to the Usefulness factor is the issue of Robustness. In addition to the accuracy of the measurement, it is important to know the extent to which the measurement can be relied upon in less than perfect conditions. During the course of practical application of a measurement, conditions may be less than perfect. The Robustness item would measure the level of discipline required from the people involved in collecting the data and the extent to which the accuracy and reliability of the measurement would be affected by inconsistent data collection and breaches in the discipline. The effect of delays in data collection or improper detail in the data being collected, for example, might affect some measurements more than others.

The Attitude factor cannot be measured exactly. Some estimations can still be made, especially regarding the expected level of acceptance that the measurement will meet with those involved in collecting and using the data. Factors likely to influence the level of acceptance can also be stated, to provide as much information as possible and to allow an informed decision to be made.

The current Measurement Options Model is meant to be a toolbox that enables the person implementing the framework to make a decision regarding the measurements to be implemented, i.e. decide which of the tools to use. The decision, however, lies with the person applying the framework and introducing the measurements rather that with the people working with those measurements and using the data resulting from them. In order to ensure a higher level of acceptance and, with it, a higher level of reliability of the measurements, the decision should be taken by the company. This approach would ensure that the selection of measurements is consistent with the company's priorities and would also clarify why each measurement is important. Moreover, the decision makers within the company would become owners of the measurements. This would reinforce the idea that the measurement is important and needed within the company and that the information it provides was deemed useful by someone actively involved in day to day work.

From this perspective it becomes very important that the decision being made is an informed one. The Measurements Options Model thus becomes more than a toolbox, it helps describe each of the items and present their benefits and drawbacks and thus provides as much relevant information as possible to the decision makers.

In practice, the process of deciding on a selection of measurements would change also. After identifying the information need at various levels, a preliminary selection would be made for each of the information needs. This selection would take into account all measurements, i.e. items in the Measurements Options Model, that could fulfill the information need and fit the information sources that have been identified. The list of all the measurements selected in this preliminary step is then presented to the owners of the information need that has been identified.

Since the selection is meant to provide them with the information they require, it would be them making the choice regarding the exact tool to use. This is also because the exact level of effort they are willing to spend on the measure and the exact level of accuracy that is needed or acceptable, may vary greatly and may be hard to explain and convey to someone outside the organization. Once the selection has been made, the items selected are implemented in close cooperation with the company, especially with the people that would be actively involved with the measurement.

The framework should strive to provide a selection of measurements that fits each company, its purposes and process and the information needs identified at each level. These factors would improve the quality of the measurements and the data they provide in addition to providing benefits in regarding the level of acceptance.

Flexibility and adaptability

The VAMOS framework focuses on offering flexibility in terms of selecting the exact measurements that are to be employed. In certain situations, however, flexibility may be needed at other levels as well.

First of all there is the level of the framework itself. While some changes are already included in the framework, more can be achieved by allowing changes to the framework, its activities, phases and the tools it uses. This would make the framework itself more adaptable to the companies it analyzes without compromising any of the benefits it provides. To illustrate this increase in flexibility a number of changes will be proposed. All these changes are the result of the practical application of the framework in an industry setting. Increasing the flexibility of the framework itself can easily be achieved by allowing practical conditions and constraints to influence decisions regarding the use of tools and the implementation of the phases described by VAMOS. Complementary to this approach, a mechanism for the adaptation of the framework to unforeseen practical problems could be included in the framework itself. This approach would allow practical issues to change the framework in a structured manner and would provide a more detailed record of the problems being faced, the solutions found to deal with them and the reasoning behind the modifications being made.

The second level of flexibility that was found useful in practice was that of the measurements themselves. The current study was an exploratory one, with little information being available from the start. In this type of setting priorities may change, as more and more information becomes available. One such example is that of a measurement providing the necessary type of information, but at a level of accuracy higher than required and with a greater cost and at a later date. The simplest and least costly solution to this problem would be that of changing the measurement to reduce the accuracy to a level that is considered sufficient, thereby also reducing cost. Little re-training of the people conducting the measurement would be needed in this case.

This change may, however, have its own cost, or there might be inherent limitations to what the measurement may be able to achieve, regardless of cost. This information needs to be available when the measurement is selected.

This approach has benefits on several levels. First of all the costs associated with selecting and implementing an entirely new measurement are reduced. Selecting a new measurement can be quite a costly process of defining information need and information availability, gathering and presenting the information relevant to the choice and then rating the available measurements based on this data. Implementing an entirely new measurement is costly in terms of developing or purchasing the tools for it and training the people that will collect it. Moreover, when a new measurement is selected to replace an older one, all the information collected up to that point is unusable, since the entire approach has changes. The cost of constant changes in process, both time and resources and the effect such constant changes have on morale, can be reduced by enabling certain changes at a low level.

The second benefit is also connected to costs. By adding flexibility to the measurement itself, a closer fit can be achieved between the measurement and the information need it tries to fill. Any measurement that is more accurate than needed or more detailed than needed is necessarily more costly than needed. The measurement could be adapted to provide information that is accurate and detailed enough to be adequate, with no excessive resources spent on items that will not be used. The reverse is also true, making a measurement more accurate or making it available faster, should the information need shift in that direction. The focus is on the fit between the information need and the information provided, rather than ensuring a uniform implementation of the measurement in all situations.

The third benefit is the possibility to consciously plan for an exploratory type of setting. Rather than selecting a batch of measurements and implementing them all at once, the approach can be a progressive one. With more data becoming available, the information need may change and decisions may be taken to change the level of accuracy of the current selection of measurements, complement the existing selection by introducing new types of measurements, remove measurements that are not relevant to the information needs or any combination of the above. An exploratory setting being analyzed with less flexible measurements runs the risk of incurring large costs without offering a better fit between information need and information availability.

Lastly, ensuring that only necessary data is being collected, removing parts of a measurement that are irrelevant or not used and making sure that all measurements fit some information need would benefit in terms of morale. Information collection is done at the expense of a developer’s time. If that information is not used, that time would have been wasted. The belief that one is required to perform activities that are compulsory, but not relevant, can sap morale and may damage other data collection efforts that are relevant.

The underlying objective of introducing flexibility at the measurement level is that of providing “good enough” information. The idea behind this approach is to provide the appropriate amount of information, of good enough quality, at the appropriate time and with minimal cost. Any increase in the accuracy or detail of the information can be provided, but only if such an increase is needed and if the costs associated with these requirements are understood and accepted.

Chapter 6 - The Measurement Options Model and the Company Options Model

Concept and purpose of the Measurement Options Model

The VAMOS framework describes a Measurement Options Model (MOM), a tool that helps the implementer select the appropriate measurements. The MOM is a list of measurements, each of these measurements having linked to additional information aimed at facilitating the choice between them. The additional information contained is:

- **Fixed effort.** The effort, measured in time or as a monetary value, necessary for the initial implementation of the measurement. This includes resources needed to develop or purchase tools, for their installation and for the training of personnel in their use. [1]
- **Variable effort.** The effort, measured in working time, needed to ensure the correct collection of data. For examples, in the case of a bug-tracking tool, this includes logging problems found and writing their description, evaluation of the potential costs or the later logging of the actual Removal and Regression Costs (RRC) or the Setup and Execution Costs (SEC). [1]
- **Accuracy.** Each measurement has certain limits regarding accuracy, with some measurement being inherently more accurate than others. While many factors influence the level of accuracy of a certain measurement, this evaluation describes the likely accuracy of a measurement when properly applied and correctly and consistently performed. [1]

Based on this additional information, the implementer of the framework can choose the measurements they find most suitable for accomplishing the goals set by the company. The developers of this tool have also identified a number of drawbacks. First among these is that different Validation Activities have different capabilities and different objectives. Judging two measurements solely on the basis of cost and accuracy will lead to misleading results and even to situations where the recommended Validation Activity cannot be implemented in a given setting [1]. A second problem identified is that of the lack of a clear definition of the measurement options [1]. Since the way each measurement is to be introduced and used may differ depending on the particularities of the company, it is difficult to make recommendation in the absence of extensive information regarding the company. The framework itself cannot make any generalized recommendations.

The more detailed MOM-Defect Measurement Options (MOM-DEMO) focused on one particular type of measurement option. While the analysis there is appropriate for any measurement that fits that category, it is not appropriate for other types of measurements. This, in effect, makes it less useful for a generalized framework, since the type of measurement needed is not known in advance. In different measurement sets, combinations may appear in which at least one of the measurements is not of the type that MOM-DEMO can analyze.

As a result of the implementation performed as part of the current project, an updated version of the MOM tool has been created, along with additional tools aimed at supporting the MOM.

The Company Measurement Options

During the course of the initial investigation into the processes and activities at RUAG, for the purpose of implementing VAMOS, it became clear that there were different information needs at different levels. Prioritizing between there information needs is a decision that cannot be made by the implementer, but only by the company. This shift in the burden of decision also meant that all the information available for this decision should be made available and included into the MOM.

To deal with the lack of clear definition at framework level the decision was made to create the Company Measurement Options (CMO). The CMO is an extended version of the Measurement Options Model. Thus, the MOM is an implementation and company independent tool. It contains all the information about a measurement that does not depend on the context. This includes a description of the measurement, evaluations of the setup cost (previously fixed effort), cost of maintenance (that comprises both the previous variable effort as well as any effort needed to deal with the maintenance of the tools and any other costs incurred as a result of the collection of data). In addition to those factors, brief analyses regarding the risks that may endanger proper data collection, the robustness of the measurement and the limitations of the measurement itself.

In the new approach the MOM is only part of a wider tool called CMO. The CMO is specifically created to suit each company and contains information and analyses that cannot be available before an analysis of the activities that are to be measured is performed. Once all the needed information about the company and its processes and activities has been collected and analyzed, a brief selection of measurement ensures that only potentially relevant measurement are selected for the CMO.

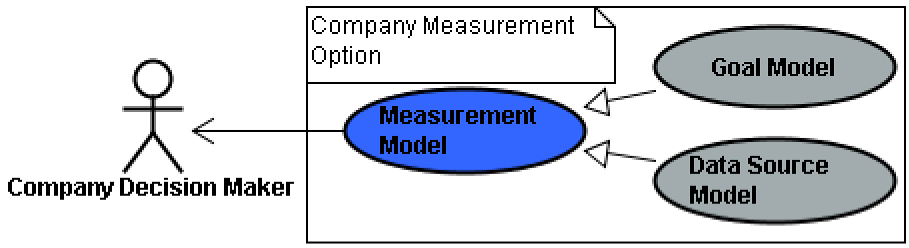

Figure 9. Overview of the CMO

In the new approach the MOM is a general list of the measurements that can be used in various situations. The MOM becomes a repository of measurements and information about those measurements, aimed at storing as much information as possible about these items. The information stored here enables the initial selection of the suitable measurements. The Company Measurements Option contains a module that is similar in regards to the type of information that the MOM contains, with some additions. In spite of the similarity of the two, they have different functions and serve different purposes.

The Company Measurement Options is aimed at providing the company decision makers with the appropriate information. The Measurement Model is the main part of the CMO and the one that contains the information about the Measurement Options.

To support the Measurement Model, two more models are contained in the CMO. The Goal Model lists the goals of the current implementation of the framework and some additional information regarding these goals. The goals are then linked to the Measurement Model, to ensure that all the measurements that are being considered refer to a goal and, as a result, are relevant in terms of the information need at some level in the company. The goal taxonomy is as follows:

Goal taxonomy

- Primary owner
- Stakeholders
- Priority
- Intent
    - Schedule
    - Time
    - Quality
    - Size
    - Derived
- Achievability determination mechanisms

Primary owner – describes the person setting this goal, who defines the goal and whose ultimate responsibility is to check whether or not the goal was achieved.

E.g. A department manager sets a goal to improve the quality of delivered software within the next year. It is his responsibility then, as owner of this goal, to describe the goal clearly, to decide what mechanisms will be used to determine if the goal was achieved and to decide whether or not the goal was achieved. The owner might not be involved in the actual implementation details, but he bears ultimate responsibility for defining and checking the goal.

Stakeholders – describes persons whose activities are influenced by the goal or who are interested in the outcome of the goal.

E.g. The quality manager and the division manager are both, albeit different, stakeholders in the example above. The quality manager is interested in the goal since it is his job to achieve it and it will directly affect his work. The division manager is a stakeholder, since the successful achievement of the goal could provide useful lessons for other departments in his division as well as providing the necessary resources for the achievement of this goal.

Priority - describes in detail the benefits that this goal will bring and puts these benefits in perspective with respect to the other activities of the company, in order to better prioritize resource allocation. Unlike problem report priorities, it is not just a number, but rather an analysis of the changes that the successful achievement of this goal will bring, and the conformance with company strategy.

E.g. For a company aiming at a reduced time to marked, the goal of improving the quality of the software might not receive such a high priority, since the analysis might reveal that the costs entailed might not be in accordance with company strategy.

Intent – describes the general intention behind the goal. If the goal is part of a larger purpose, then its place and influence in achieving this purpose is described.

E.g. The improvement of the quality of delivered software may be part of a larger purpose of pushing into premium markets. In this situation, an analysis follows on how the level of achievement of the goal will help towards the larger aim. The improved quality would translate in saving in terms of maintenance and improved image for the company and its products. Conversely, failure to achieve this goal might hamper company moves into premium markets, since it would affect both maintenance and customer satisfaction and fidelity to the brand.

Achievability determination mechanisms - describes the mechanisms that will be used to determine whether or not the goal was achieved. This is a description of what indicators or measurements will be used to determine whether or not the goal is achieved, as well as an analysis on how the mechanisms chosen fit the intent described earlier on. The purpose of this is to ensure that the measurements being performed are in accordance with the goals, that all needed measurements are known and that only needed measurements will be performed.

E.g. The improvement of code quality over the next year with an aim to pushing into premium markets entails a certain set of measurements. If efficiency measurements are part of these mechanisms, an analysis is needed to determine if the efficiency measurement is in accordance with the intent of improving quality for new markets. It might be the case that efficiency is not an overriding priority at this time. In this case it is removed from the mechanisms presented here.

The Goal model aims to identify the information need at all levels, to associate these information needs with the stakeholders involved in the data collection and usage and to link the information needs to the measurements being proposed. The link between the measurements and the goals is designed to provide a means of allowing the importance and priority of the goal to influence the choice of measurements. Measurements associated to a goal of higher importance and priority will answer a more important question and provide greater benefits.

The Data Source Model contains information about the activities and artifacts that contain all or part of the information that is sought. While most of the information sources are fairly obvious, completing this model will allow previously overlooked data sources to be considered and would provide a clearer overview of where the information is coming from. Moreover, an assessment of the sources of information would also show the types of information that are available and the kinds of measurements that would be suitable to collect all this information.

Data Source taxonomy

- Type of data
- Accuracy
  - Low
  - Medium
  - High
- Updating frequency

- Reliability
- Reliability/Accuracy increase requirements
- Collection method
    - Manual
    - Automated
    - Automatable

Type of data – describes the kind of information that is accessible through the data source. A single data source may yield several types of information, while other data sources may be focused on a single type. The purpose of this is to ensure that information available in each data source is the kind of information that would help accomplish the goals.

E.g. Planning meetings at project level are a primary source of information for items such as planning, timing across the project and so on. Thus, measurements relating to this data source are only relevant if one of the goals is related to planning and timing information. If none of the high priority goals are linked to planning or timing, then measurements collecting data from this source lose some of their importance. If, on the other hand, timing and planning information at a project level relates to a high level goal, then measurements that collect data from this source will also receive a higher priority.

Accuracy – describes the level of accuracy that can be expected from the data source, as well as a brief analysis of factors that may affect the accuracy of the data source. These factors may vary from how official the data source is, whether the information is obtained by estimation or by other means as well as any mechanisms there are for ensuring data accuracy.

E.g. Information present in planning meetings is official, both based on official documents and forming the basis for official documents. Planning and management decisions are based on information presented here. The planning information is, however, estimated. From this point of view the accuracy of information at most planning meetings is just medium. In some situations, planning estimations are regularly updated. This helps to increase the overall accuracy of the current data source. While this might not modify the rating, since the rating system is quite coarse-grained, mentioning all these factors would be beneficial in selecting the most appropriate measurements and data sources.

Updating frequency - describes the frequency with which a data source is updated. The more often updated data is more suitable for measurements aimed at showing an instantaneous evaluation of the situation. For meetings, the updating frequency is the frequency at which the meetings take place.

E.g. A daily planning meeting, when considered as a data source, can be considered to be updated daily. Documents on the other hand may be updated weekly or less. The significance is that two reasonably close data points may provide an accurate estimation of progress being made in the case of a frequently updated data source, while the same estimation of the same process made on the basis of a less frequently updated data source may suggest no change whatsoever.

Reliability – describes the level of confidence that the data source is properly updated and used. Documents that are frequently used and planning meetings usually have high reliability ratings since the meetings have to be held and the documents have to be kept up to date. Less used documents or informal data sources may provide different results.

E.g. A planning meeting for the project is a reliable data source, since the planning meetings have to be held in order for the project to progress. Frequently used documents and documents used in the planning process are more likely to be up to date and contain correct and accurate information.

Reliability/Accuracy increase requirements – describes the effort needed to increase the accuracy or reliability of a data source. A brief analysis of the reasons for that estimation is also included.

E.g. An increase in accuracy of the data obtained from a database may have a low cost, if the additional data can be obtained from information already in the database. Consider a tool that uses a defect database to extract the total number of defects in a given module or in a given period of time. Increasing the accuracy to provide a breakdown by type of those defects would have a low cost, provided that the defect type information exists in the database. If that information does not exist, increasing the accuracy in that manner would mean changing the structure of the database, changing the way data collection is performed and other costs, thus making the increase quite costly.

Collection method – describes the method by which the data can be collected from this data source. Automated describes a data source for which an automated data collection tool is already available; automatable means that the data collection can be automated, but no automated tool exists at the moment; manual describes a situation where data collection cannot be automated with reasonable cost and accuracy.

E.g. Defect counts collected from databases are automated, i.e. tools already exist, or at least automatable, i.e. the tool can be developed at reasonable cost. Data collection of information from planning meetings, on the other hand, cannot be automated. Collection of that data has to be performed manually, at least initially. After the information is stored electronically, automated tools can be used for further analysis.

The Measurement Model is the main part of the CMO. It contains a set of measurements that have been selected based on their suitability for achieving the specified goals based on the specified data sources. In addition to listing the measurement options, the Measurement Model also stores additional information, useful in making the most appropriate selection, and links to the other Models.

Metric taxonomy

General

- Goal
- Prerequisites
- Description
- Type
    - Process
    - Product
    - Project
    - Resources
    - Hybrid
    - Other?
- Risks

- Robustness
- Control and conformance mechanisms

Effort and accuracy

- Setup effort (how much time, who is concerned)
- Maintenance effort (how much time, who is concerned, how often)
- Accuracy
- Level of analysis needed:
    - Limited: automatable analysis
    - Medium: data can be analyzed without reference to other documents.
    - High: analysis with reference to other documents
    - Extensive: analysis by experts, relying on other sources as well
- Degree of process change required for implementation

Stakeholders

- Primary beneficiary (owner)
- Primary performer
- Other stakeholders
- Accuracy change cost
- Performer acceptance
- Beneficiary acceptance
- Additional motivating/de-motivating factors
- Metric visibility level and limitations

General

Goal – this connects a metric or measurement to a certain goal. No metrics should be performed unless they are needed or requested and the information gathered is not used in some way.

Prerequisites – describes any type of document, information source. goal, other metric or any other artifact needed to successfully implement this measurement.

Description – a description of the metric, how it is to be introduced, how it is to be performed, the steps that need to be followed, etc.

Type – A choice exists here between the categories mentioned above. The categories are not fixed, but open to additions and improvements. The type, in this context, describes what aspect is being measured.

Risks – This is a brief analysis of the potential risks in adopting the metric. The risks are described and analyzed in order to offer the company a clear view of what to expect when adopting this metric, what kind of difficulties may arise and to check if the risks involved are acceptable for them.

Robustness – describes the impact on accuracy unsuitable application of the metric has. The unsuitable application refers to minor discrepancies between the actual application and the ideal model.

E.g. Measuring the number of lines of code is robust with respect to following timing constraints: delaying the measurement for a day or two, especially in advanced stages of development, will not harm accuracy too much. The same measurement, however, is very sensitive to different definitions of line of code: i.e. does this definition include comments or empty lines, does it check for proper formatting first, etc. Measuring different modules with different standards may decrease accuracy to the point of making the results unusable.

Control and conformance mechanisms – describes the mechanisms that might be used to ensure the reliability of the data obtained with this measurement.

E.g. For a measurement involving recording of certain data by the developers, the project manager might remind everyone at the daily meetings to record that data or might ask them if they have recorded it on the previous day. This will reinforce the idea that the measurement is important and might convince developers to conform to the procedure set for this measurement.

Effort and accuracy

Setup effort – describes the effort needed to put the measurement in place. This includes any tools that need to be developed, any training that would be needed, etc. This item includes an estimation of the overall effort needed and an estimation of what people and what roles are needed, as well as any other consideration that will affect the effort of successfully implementing the measurement.

Maintenance effort – describes the effort needed to perform the measurement as needed. This includes the time spent, the people involved in performing the measurement and checking compliance and the frequency with which the measurement, or certain parts of it, will have to be performed. This includes considerations such as level of automation of the measurement, time spent collecting it, tools available and any other consideration that will affect the effort of performing the measurement.

Accuracy – describes the level of accuracy that the measurement will provide at the specified level of effort and while following the methodology provided in the description. This is a general assessment that will provide the company with more insight into possible benefits, as well as contrasting these benefits with the costs associated with the measurement.

Level of analysis needed –within the context of the options presented above, this item describes to what extent the information gathered will have to be analyzed before being presented to their intended audience. By default this refers only to the primary beneficiary of the metric, but the definition can be extended to include the extent of the needed analysis for presentation to other stakeholders, together with considerations regarding the effort needed for such an analysis and relevance to the activities of the stakeholders in question.

Degree of process change required for implementation – describes an in-depth analysis of the changes that have to be made to the overall process in order to adopt a new measurement. This can be a separate category (as here) or it can be included in the setup costs for the metric.

Stakeholders

Primary beneficiary (owner) – describes the person that will receive the results of this metric. This person is the owner of the metric and is ultimately responsible for the appropriate use and dissemination of the data resulting from this measurement. As a result, the owner will ensure that the measurements are in accordance with his/her needs and will change the measurement when appropriate. This person represents the information need.

Primary performer – the person in charge with performing the measurement and delivering the results. This person is in charge of the actual implementation of the metric, will ensure that the measurements are carried out in accordance with the description, the information requirements and the needs and instructions of the owner. This person represents the information source.

Other stakeholders – any other persons involved with performing the measurements, interested in the results of the measurement or affected by the outcome of the measurement. These include other people involved in performing the metric, in analyzing results or whose activities may be influenced by these results.

Accuracy change cost – describes the costs (or benefits) entailed by modifying the level of accuracy for this metric. The increase in the accuracy level may require additional resources, while the decrease in required accuracy may free up resources. Since this dependence is not linear, it is important to know the costs associated with performing accurate measurements, such that informed decisions can be made about the level of accuracy needed with respect to the resources the company is willing to spend on the measurement.

Performer acceptance and **Beneficiary acceptance –** describe the level of probable acceptance from the 2 main parties involved. These will describe probable levels of acceptance, main factors influencing and any other considerations that might affect the acceptance of the measurement. This should be a rather detailed analysis, given the importance of the morale factor in any metrics and measurements implementation project.

Additional motivating/de-motivating factors – describes any other factors that might influence the level of motivation for the people involved. Incentive plans can be described here, as can any other factors that can be foreseen or are deemed important.

Metric visibility level and limitations – describes the limits of the audience for this measurement and its limitations. While the intended audience is the owner, many other people in the organization might be interested in the results. This item describes how visible should the metric be at other levels than the intended audience. It also describes the limitations of the measurement with respect to these levels, and puts the results of the measurement in perspective for people at these levels. It is important that results are not misinterpreted or mishandled, and this item seeks to reduce that risk.

The measurements selected for inclusion in the CMO have to pass an initial evaluation regarding their suitability, given the goals and information sources described earlier. Once this initial selection has been performed, the information presented above is obtained and added to the Model. A second selection may follow then, removing those measurements that are not compatible with the company's priorities. The results are then presented to the company decision makers for selection and approval.

The CMO emphasizes the shift in the burden of decision from the implementer of the framework to the company decision makers. As a result of this approach all the relevant information must be provided, since the implementer's previous experiences with measurements will not be influential to the decision unless clearly stated. A side-effect of this approach is to provide a lot of information regarding the priorities and goals of the company, as well as the reasons behind certain decisions. This information will be useful to the implementers in the application of the measurements. The additional information regarding the goals and information needs at various levels may also be useful to the company itself. As discussed earlier, knowledge about the information needs and goals at various levels is not usually widespread in the company.

Implementing the Company Measurements Options

The completion of the "Define" phase and the detailed description of the process used at RUAG are described in another chapter, as are the information needs regarding the Integration Phase. This in-depth study enabled the Data Source Model and the Goal Model to describe the company correctly and accurately.

With the Data Source Model and the Goal Model suitably provided with information, the creation of the Measurements Model can be performed. To reduce the number of measurements that have to be analyzed and ensure that only those measurements that are relevant to the goals are investigated in any depth. Once this preliminary selection has been performed, a detailed analysis of each of the measurements shows how they can be applied in the specific context of this company, what their adoption would cost and how it would affect the process and the activities of the company. An example of how each of the components of the CMO would look like is presented below, applied to the Code Inspection Sheet tool, data sources and goals.

Data Source

Table 2. CMO view of a data source - example

Number	Data source	Type of data	Accuracy	Updating frequency
1	Code Inspection Sheets	Defect, defect classification, defect numbers and distribution.	High. The Code inspection sheets are official documents, are regularly performed and inspected.	Irregular. The code inspection is performed once a code module or modification to a code module is completed.

Reliability	Reliability/Accuracy increase requirements	Collection method
High. The documents are official and regularly used.	Medium-high. Eliciting information not in the code inspection sheets would require extensive modifications to the sheets and process. Information that can be derived from existing data is easy to collect.	Automatable. A tool already exists for collecting certain kinds of information from this source. Other tools can also be developed.

Goal

Table 3. CMO view of an Information Goal - example

No.	Goal description	Owner	Stakeholders	Intent	Achievability determination mechanisms
1	Code inspection activity evaluation	Software Department Lead	Software Department Lead, Project Leader.	Quality, Time. The information would indicate whether or not too much code is being rejected for minor reasons (as recorded in the code inspection checklist).	An analysis of the results would indicate the degree to which code modules are being rejected on the basis of various checkpoints.

All this information is used to create the Measurement Model entry for the Code Inspection Sheet Tool.

Measurement

Table 4. CMO view of a measurement - example

Number	1
Metric name	Code Inspection Sheet Analysis tool
Goal	1.
Prerequisites	Consistently formatted and performed code inspections.
Description	Analysis of the code inspection sheet to identify reasons for rejection in terms of failed checkpoints.
Type	Product.
Risks	1. Changes in code inspection sheet format. Anomalous rejections may cause incorrect results.
Robustness	Medium - small changes are needed to cope with changes in CI sheet format.
Control mechanism	None (Automated).
Setup effort	High. Development of the system.
Maintenance effort	Low. Running cost is low. Minimal modifications for format changes.
Accuracy	High. Due to automation
Analysis	High: analysis with reference to other documents.
Degree of process change	None.
Owner	Project manager.
Performer	Automated
Stakeholders	Management, Process analysis, project leader.
Accuracy change cost	High. Increased accuracy requires tool redesign and concept redesign.
Performer acceptance	N/A
Owner acceptance	Medium. The limitations associated with the approach are known.
Motivating/ Demotivating factors	

Limitations	Limited to code inspection checkpoints and to the outcome of code inspections. Given the difference between projects, data may be difficult to interpret accurately
Perfomance frequency	One-time collection at the end of the project.

For this specific situation, five measurements have been presented to the company during the measurements workshop. These measurements were:

- **Use spreadsheet to collect fault information.** Every time a fault is discovered, an entry is made in a spreadsheet form, recording information about the fault. Not all the information is available, however, at the time of discovery. In most cases, the exact type of the fault is hard to determine and there is no way to estimate removal time and cost. This additional information would be recorded when the fault has been removed and the fault investigation has been concluded.

- **Use spreadsheet to collect timing estimation information.** A planning document is created that stores the timing estimation information that is discussed during planning meetings. Such a document would have to be updated several times a day, as new information becomes available and existing data is updated. The estimations would then be compared against the actual time needed to perform certain activities.

- **Flexible information collection (at meetings).** The information needed in most cases exists in the company and is discussed during planning meetings. This includes planning and estimation information, fault classification information and delay information. The purpose of this tool is to allow the people interested in certain information, in this case the Integration Responsible, to collect the information they need from the meetings is a structured form. The tool itself is meant to be as flexible as possible, so as to ensure that only information being used is actually collected. The tool is designed to allow the key stakeholder to change the type of information being collected, the level of detail needed as well as how often the information is collected with a minimum in cost and preparation. The tool is of an informal nature, with an emphasis on flexibility, adaptability and low data collection costs, which make it particularly suitable for exploratory studies.

- **Survey.** A survey is conducted with questions that reflect the information needs at the time of the survey. Information concerning a variety of issues can be collected, depending on the selection of questions to be included. When compared to the Flexible information collection (at meetings), this tool does have some additional costs. Questions for the survey have to be prepared and pre-tested, to ensure their suitability and to ensure that all those involved understand them in the same manner. Moreover, completing the survey takes time and results are more costly to interpret. Mechanisms for ensuring that the answers are of the appropriate quality also have to be put in place. Ensuring that the survey is completed in a timely manner, that all questions are properly understood and answered, would also incur additional costs.

- **A list of questions to think about.** In keeping with the focus on lightweight tools, suited to exploratory studies, this tool consists of displaying the questions that key stakeholders are interested in. During the meetings, if any of the participants have information regarding any of those issues, they would step up and provide that information, which would then be recorded and analyzed. The scope of those questions and the way they are defined poses certain problems, particular to this tool. Keeping the scope of the questions very clearly and strictly defined would make it difficult to fit the tool for the purposes of exploratory research, since only planned types of information would be collected. On the other hand, a loosely defined scope for the questions makes it difficult to ensure that all those involved have the same understanding of the questions and the information that is required. A suitable balance must be reached and maintained, in order for such a tool to provide results.

A more detailed view of all these tools, from a CMO perspective follows below.

Table 5. CMO Measurement Options.

Number	4	5
Metric name	Use spreadsheet to collect fault information	Use spreadsheet to collect estimation information
Goal	3.	2.
Prerequisites		
Description	An Excel spreadsheet is used to record certain information about each fault found (sampling can be introduced to minimize workload).	An Excel spreadsheet is used to record information regarding schedules, tasks and estimated durations. The information based on the information obtained during the daily meetings (e.g. SW, Integration, etc).
Type	Process. Product.	Process.
Risks	1. Inadequate reporting might lead to inaccurate results. 2. Additional effort might be rejected by users.	1. Inadequate reporting might lead to inaccurate results. 2. Additional effort might be rejected by the performer.
Robustness	Low. Requires discipline in terms of data entered for any results. Inconsistent or anomalous entries are hard to detect.	Low. Requires discipline in terms of data entered for any results. Inconsistent or anomalous entries are hard to detect.
Control mechanism	Human. The use of the tool would have to be verified on a regular basis.	Human. The project manager or responsible for that part of the project (e.g. SW, TE, HW etc) would have to be relied upon to collect the data.

Setup effort	Low. The option is relatively simple to introduce, although it will require training in order to achieve good results.	Low. The option is relatively simple to introduce, although it will require motivation and a feeling of the usefulness of the metric to acheve good results.
Maintenance effort	Medium. Running cost is significant, as each fault has to be handled individually.	Medium. Running cost is significant, as after each meeting changes in estimation information would have to be recorded.
Accuracy	Medium-High. Usually accuracy is high, but sampling biases and inconsistent data collection might reduce it significantly.	Medium-High. Accuracy is high, although dependence on other estimations and inconsistent data collection may affect it.
Analysis	High: analysis with reference to other documents.	Medium: data can be analyzed without reference to other documents.
Degree of process change	Medium-Low. This method requires the training of the staff in the new approach and ensuring the proper collection of data.	Low. This method requires little change in the actual process, aside from the added task of recording the data for the project manager or meeting leader (SW responsible, Integration responsible, etc)
Owner	Department manager, Project manager	Project manager.
Performer	Project team.	Responsible for the area in question.
Stakeholders	Project manager, project team.	Project manager, project team.
Accuracy change cost	Low. The adding of additional data to the system requires comparatively little change.	Low. The adding of additional data to the system requires comparatively little change.
Performer acceptance	Low. Additional cost associated with daily tasks may result in low acceptance by the performers. The performers would also have to be informed as to the intention and the use of the metric.	Low. Additional cost associated with daily collection of this data may result in low acceptance by the performer.
Owner acceptance	Medium. Sampling and data collection biases or perceived biases may lower confidence.	High. Information would be useful and relevant in terms of process analysis.
Motivating/ Demotivating factors	- increased cost associated with recording the data - perceived uselessness or ignoring of the data - lack of information at the moment of fault discovery - improper data collection may be seen as more expedient. + actual usage of the data may improve motivation in collecting the data.	- increased cost associated with recording the data - perceived uselessness or ignoring of the data - improper data collection may be seen as more

Limitations	Due to the human factor, collection of the data may be inaccurate or problematic. Motivation of the measurement performers is an important part of adopting this metric.	Due to the human factor, collection of the data may be inaccurate or problematic. Motivation of the measurement performers is an important part of adopting this metric.
Perfomance frequency	With each fault found or selected during the sampling process.	Daily (after the meeting) or weekly (this may result in a loss of accuracy).

Number	12	13	14
Metric name	Flexible information collection (at meetings)	Survey	A list of questions to think about
Goal	2. 3.	2. 3.	2. 3.
Prerequisites	Regular meetings of the project team.	A very clear and specific set of questions to answer. The questions themselves should be clear to everyone involved, and they should not leave room for other interpretations.	The information needs have to be clear. The team needs to be confident and comfortable about communicating the information to their colleagues.
Description	The Integration Responsible (although it can be aplied to other areas) collects the data he is interested in. Once the questions stabilize, the process can be formalized.	A survey is created, with questions about timing, faults found, time taken to fix and so on. The survey is periodically (and automatically?) sent to the project team, and the replies collected.	A list of questions is given to the project team. At each meeting, if a project team member has new and relevant information about any of the questions, he communicates it to the rest of the team.
Type	Process. Product.	Process. Product.	Process. Product.
Risks	The meetings could last too long as a result, leading to time waste and drop in interest.	The survey may have a low response ratio. That might affect the accuracy of the results. Automation of collection may also distance the owner from the measurement and relevance might diminish	Information may be lost if team members don't feel comfortable with relating their experiences.
Robustness	High. The method is designed to be extremely flexible and adapt to the information need.	Medium. A low response ratio will affect accuracy, but useful information may still be derived.	Medium. Individual personality will affect the measurement. Useful information can be gathered, though, even in those conditions.

Control mechanism	None (The method is driven by the person collecting the information and who has the information need).	Automated and Human.	Human.
Setup effort	Low. Little preparation is needed, especially for the initial, exploratory, part. A few iterations might be needed before the questions are completely understood by the entire team.	Low. Once the survey questions are set, the actual survey development would not require much in the way of resources. Developing an automated way of sending and receiving the questionaires might be required once the method is proven.	Low. Little preparation is needed, once the informatino need has been identified. A few iterations might be needed before the questions are completely understood by the entire team.
Maintenance effort	Low. The running cost is just the time taken to record the information. Changes are encouraged, and so incur little cost.	Low. Sending and collecting the questionaires could be automated.	Low. The running cost is just the time taken to record the information. Changes are encouraged, and so incur little cost.
Accuracy	Medium-Low. The method relies on estimation, and that decreases accuracy significantly. However, since the data is collected quite often, there should be little information loss.	Medium-High. In general, the accuracy depends on the questions, and has the potential of being high. Low response ratios may affect the accuracy, though.	Medium-Low. The method relies on estimation, and that decreases accuracy significantly. However, since the data is collected quite often, there should be little information loss. In addition, question formulation and team members' personalities may affect the outcome.
Analysis	Medium: data can be analyzed without reference to other documents.	Medium: data can be analyzed without reference to other documents.	Medium: data can be analyzed without reference to other documents.
Degree of process change	Low. The introduction of the questions would fit smoothly into the current process. Changes in the questions would also cause little impact on the process.	Low. The questionnaire would not change the daily routine of the project team.	Low. The introduction of the questions would fit smoothly into the current process. Changes in the questions would also cause little impact on the process.
Owner	Integration Responsible (Can be extended to other roles as well).	Integration Responsible (Can be extended to other roles as well).	Integration Responsible (Can be extended to other roles as well).

Performer	Integration Responsible (generally the same role as the Owner)	Integration Responsible (generally the same role as the Owner)	Integration Responsible (generally the same role as the Owner)
Stakeholders	Integration responsible, project team.	Integration responsible, project team.	Integration responsible, project team.
Accuracy change cost	Low. Within the limitations of the method, there is little cost for changing the detail of the questions.	Low. Within the limitations of the method, there is little cost for changing the detail of the questions.	Low. Within the limitations of the method, there is little cost for changing the detail of the questions.
Performer acceptance	Medium-High. There is additional effort associated with data collection. This will be offset by the fact that the performer also has the information need, and is thus motivated to conduct the measurement.	Medium-High. There is additional effort associated with data collection. This will be offset by the fact that the performer also has the information need, and is thus motivated to conduct the measurement. Automating collection might reduce effort.	Medium-High. There is additional effort associated with data collection. The effect might be offset by the chance to discuss achievements and results.
Owner acceptance	High. Information would be useful on a daily basis and within a wider context.	High. Information would be useful on a daily basis and within a wider context.	High. Information would be useful on a daily basis and within a wider context.
Motivating/ Demotivating factors	- there is some cost associated with collecting the information + the information driver is also performing the measurement, so misunderstandings or mismatches are avoided + the method is highly flexible, allowing the modification of information collected and the introduction of new information	- there is some cost associated with developing the questionnaire and (as an option) the automated collection - automation of information collection might distance the owner from the measurement + would take up less of the meeting time + the method is quite flexible	- team members' personalities might inhibit them from sharing their experiences + the chance to discuss achievements, results and discuss solutions + the method is quite flexible
Limitations	The method is meant to be a starter, an exploratory search into the information need. Once the information need is clearly established, more formal and rigid methods can be employed.	The survey is limited in the amount of information it can collect and in the accuracy of that information. Biases may appear, since a small number of active respondents might generate opinions out of proportion with their numbers.	The method relies on the team members volunteering information. Low motivation may be a limiting factor.

Perfomance frequency	Once or twice a week.	At any regular interval (suggest once or twice a week).	At any regular interval (suggest once or twice a week).

Once all the selected measurements have been analyzed from these perspectives, it became evident that several of the measurements are more appropriate for the goals prioritized by the company than the others. Once all the information has become available and all the analyses were complete, a meeting is held with the decision makers, i.e. usually the owners of the goals in question and some of the stakeholders. The purpose of the meeting is to present the findings, propose measurements and allow the stakeholders and decision makers within the company to make the decision. This is also an opportunity for all the people involved in the process to come up with additional ideas and proposals. Since each company has a specific set of problems, means and priorities, ideas coming from people actively involved in the process could be extremely valuable. Sharing all the information that has been obtained up to the present time is also valuable way of obtaining more information and verifying the current model.

The measurement information may require re-doing after such a meeting, in light of the new information. Once the decision has been made regarding the selection of measurements to implement, the CMO becomes a list of possible alternatives.

In the implementation of the current prototype, the 5 measurement options presented above were discussed. All the measurements presented were lightweight, low cost and suitable for the stated goals of the company, so the discussion focused on less definable issues such as user acceptance, a best fit between specific practices and the data collection methods and minimizing the impact that the introduction of the measurements would have on existing processes and activities at the company.

The survey and the fault information collection were rejected on the grounds that they would impact the company's process to a greater degree that was acceptable. Introducing new documents and new activities in the Integration Phase is a considerable burden on the process. The emphasis these tools have on written documents also means added cost in the initial stages of implementation, when changes are inevitable, mostly due to the availability of new information. In addition to the real burden, adding the new activities to the process would also create perceived burdens which would hinder the acceptance of the new measurements and thus diminish their usefulness.

The List of questions to think about tool was also rejected, mostly due to the fact that it is an unrefined concept. While the idea does have potential, it requires significant amounts of testing and practice, meaning an increase in the cost and time needed for implementation. Additional considerations would make this method difficult to adapt to. The method relies heavily on the willingness of team members to participate and on their similar understanding of the questions and the level of detail and type of information required. These can be fulfilled if the measurement is met with acceptance from the users, but acceptance requires time and practice.

The decision was to start implementing the Flexible information collection (at meetings) tool immediately, in the context of the Integration Meetings. The goal of the tool was that defined in the CMO: to obtain more information about the Integration Phase, specifically regarding the delays, the amount of the delays and the overall causes for the delays. The exact questions that were to be asked, the form of the metric and the exact nature of the expected results were left undefined at the meeting.

Defining them in more detail would have limited the flexibility of the tool without improving the quality of the resulting information.

The Spreadsheet for collecting estimation information was also considered a useful measurement, and the decision was taken to start the implementation work on that, as well. While full implementation was unlikely, due to timing constraints, it was decided that work on refining the spreadsheet, the fault classification and other foreseeable practical aspects could begin.

Comparison with existing work

In the paper [8], several recommendations are made regarding the adoption of the ISO standard and the experience of the setup of a metrics system in Ericsson SW Research, Ericsson AB, Sweden. This aims to compare the recommendations and considerations in that paper with the CMO and the way the CMO is structured.

- Reason in terms of information needs rather than what can be measured.[8]

The CMO was built around the idea of indentifying long term measurement goals and associating each goal and each proposed metric with an Owner. The purpose of this is to define the person most interested in obtaining that data and thus the person needing the data. Furthermore, during the interviews preceding the development of the CMO, particular emphasis was placed upon the information need at various levels in a project, their identification and classification. This emphasis on information need resulted in the definition, for each metric, of owners, type of information provided and owner confidence or acceptance of the metric.

- Use indicators as the main information products provided to the stakeholders, not base or derived measures. [8]

Given the level at which the CMO is defined, it is impossible to provide more than just a framework. In order, however, to ensure that the cost and effort of transforming the result of each metric into something meaningful for the stakeholder, the CMO defines for each metric the "Analysis" category. This seeks to define the amount of analysis that is needed before the direct information can be used. Given how this example was focused on the integration phase, the information need is mostly in the area of fault slip-through. While this may be hard to automate, analysis by experts, with various degrees of need for other documents is a fair measure of the effort needed to provide the stakeholders with the appropriate indicators. As an example, numbers and types of faults collected during the integration phase might be misleading or irrelevant in the absence of additional information. Such additional information includes:

  - Similar data from other projects and a comparison of the results
  - An evaluation of the special factors that affected this project and their likely impact on the results
  - An analysis of potential changes to the process and their likely impact on the following project.

Given the nature of these comparisons, it is considerably harder to automate them. The data can still be used, however, after some amount of expert analysis. This category seeks to quantify, albeit roughly, the effort spent on that analysis.

- Use customized measurement systems rather than off-the-shelf metric tools.[8]

The CMO is a framework for selecting the appropriate metrics, rather than a measurement system. As a result, the very goal of the CMO is to fulfill this recommendation, by enabling each company to select those metrics that are relevant for them, to customize both the metrics themselves and the metric systems they use. Furthermore, the purpose of VAMOS as a whole is to provide a tool to not only customize existing metrics systems for company use, but to enable continuous customization and adaptation in the way each company applies the metrics, the information collected and its use.

- Build own competence [8]

The recommendation emphasizes the importance of creating a competence group inside the company, rather than hiring external consultants for the performance and adoption of metrics. The CMO and, at a larger scale VAMOS, seek to create a continuously updating and improving measurement system. This continuity relies on the internal competences, information needs and decision making abilities within each company. The purpose of both these frameworks is to create a measurement system that the company can use, customize and improve based solely on its own needs and capabilities. This is a long term goal that can only be achieved by building a competence base within the company, and relying on that base rather than the, by definition short term, involvement of outside consultants.

Other considerations are mentioned in the paper, such as:

- Automation of metrics collection
- Providing means for assessing whether indicators can be trusted or not
- Ability to integrate the measurement systems with existing infrastructure [8]

All these are handled by the way the information for each metric is structured. Collection mechanisms, Control mechanisms and the Degree of process change seek to address these issues, while at the same time, providing the company with the flexibility to adopt certain options if they think that the benefits outweigh the drawbacks and justify the needed effort.

Conclusions

The MOM has proven to be a valuable tool in the implementation of the VAMOS framework. The extension of the MOM and the development of the CMO have made the shift of the burden of decision to the company itself possible. This can only be achieved by providing the company decision makers with all the information needed to make an informed decision. Moreover all this information needs to be properly analyzed and linked to all other relevant items of information.

The MOM and CMO combination aims to fulfill several roles. First, the MOM itself aims to be a repository of knowledge and experiences about various measurements and validation activities, their potential benefits, limitations and experiences related to their implementation. Since the MOM gathers shows data related to the measurement itself rather than the company, data gathered here could be used in other settings as well. This benefit is also the source of the MOM tool's greatest weakness. Given the general level of the data it aims to gather, the MOM can be used to select measurements that may be appropriate in a given setting. It cannot guarantee that the measurements will work as theorized and it cannot give practical details regarding their implementation. The levels of effort and cost presented in the MOM are overall estimations and generalizations of previous experience.

The second role the combination seeks to fill is that of evaluating each measurement in the context of a given company and presenting a small number of options that are both suitable and achievable by the company. This role cannot be filled by the MOM, due to its general nature. It is for this role that the CMO was developed. For each company, a CMO can be drawn that will find suitable measurements for the goals of that particular company, to fulfill the information need of that company and that can be performed with the resources of that company. While seemingly similar in form, the MOM and the CMO are on opposite ends of the generalization scale: the MOM takes a general view, focused on measurements; while the CMO takes a detailed view, and focuses its efforts on the company.

Chapter 7 - Measurements implementation and the Exploratory Measurement Framework

Introduction

The Company Measurement Options, as described in previous chapters, enables the company to select the measurements they find most relevant and that would bring the greater benefits. The analysis performed as part of the CMO requires a large amount of information and tries to offer the best match between measurement options and the information needs of the company. All this analysis, however, cannot guarantee that the chosen measurements are practically suitable in the given setting. To provide validation for the CMO approach in the current project, a measurement was selected for implementation by means of the CMO process.

The selection process

During the preliminary analysis for the completion of the CMO, one of the information needs that were identified was that of planning and the comparison between planning estimations and the actual time taken. In addition to the actual delays, a necessary item of information was the reason that the delays in question were incurred. A short list of measurements that fit those information needs and that can be applied on the information sources that were available was prepared. This list, with all the additional information that is part of the CMO, was then presented as part of a Measurements Workshop. During this workshop, a decision was taken regarding the measurement to be implemented.

The measurement selected was to function in the context of the Integration Phase, the main part in the Integration Driven Development process used by the company. This poses certain limitations on the available sources of information, since the different departments cooperating in the Integration Phase seldom have the same systems and documents in place. In addition to the limitations placed on the information sources, implementing the measurement in the context of the Integration Phase also sets the overall scope. The level of detail in the information that the measurement provides has to match the level of information needed for the proper functioning of the Integration Phase. Too little detail and the information would be useless; e.g. saying that the delays are caused by factors in the Integration Phase does not offer any means of narrowing down the likely cause and doing anything to improve the situation. Too much detail and the information would be hard to collect and harder still to act upon at the Integration Phase level; e.g. finding detailed information about software department specific issues is not something that can be solved at the Integration Phase level.

The problem at this point is still that the definition of both the level of information needed and the exact boundaries of the scope are rather broadly defined. There is no way of knowing this information exactly without actually putting the measurement into practice. For this purpose the decision was taken to implement a prototype measurement. By this means, any practical difficulties in the implementation could be identified, as well as helping to better define the exact type and detail needed in the information being provided.

The EMF approach – Exploratory Measurement Framework

In order to deal with all the uncertainties of the situation, a flexible approach was decided upon. The goal of this approach is to allow the maximum level of flexibility at the level of the measurement itself. In practical terms, the purpose is to apply the principles of agile software development to the development of measurements appropriate for a given situation.

Given that the focus of this approach if that of providing the higher possible level of flexibility, an exact set of steps and a precise framework for it is hard to provide. The philosophy behind it, however, relies on a few critical points.

- **The measurement's owner is the decision maker.** In the CMO, a role was defined of Owner for each measurement. The Owner is the one who will use the information derived from the measurement. In effect he/she is the person that has the information need that the measurement is trying to fulfill. As a result, they have the ultimate power of decision over whether or not the information provided is sufficient, what focus the measurement should take and what items are to be added to the measurement or removed from the measurement. Regardless of what the implementer might think is the best practice; it's the measurement's owner that is the "client" in this case and, therefore, it's their needs that matter most. The underlying principle is that since the owner has the information need, they are the only ones to know when the information they are provided with fulfills their expectation. Moreover, as more and more information becomes available, information needs may change, as may the focus of the measurement itself. In this situation too, it's the owner that has the ultimate power of decision regarding the measurement, since they are in the better position of knowing what information they require in the new setting.

  A second reason for this point is that the owners themselves might not have all the information at the beginning of the measurement implementation process. As more information becomes available, their own information needs become clearer to them. The situation may arise where information previously thought important was irrelevant while other information has gained in importance. The measurement should be changed such that the owner's information needs are fulfilled, adapting the measurement as often as needed.

- **The measurement is not fixed.** While individual measurements are the building blocks of any measurement programme, they are not to be seen as immutable. Certain measurements and measurement ideas have limitation related mostly to the manner in which they collect data and the types of data they are designed to collect. These limitations cannot be changed without changing the basic metric.

Within those limits, however, the measurement itself can be changed as often as possible.

This is particularly true for measurements aimed specifically at being as flexible as possible. In the current implementation, the primary goal was to put in place a set of measurements to enable an exploratory study of the Integration Phase. The type of measurement best suited to this kind of study is a flexible measurement to which this kind of philosophy is obviously beneficial. Even to more rigid settings, however, the idea of adapting the measurements is beneficial and usually followed, albeit unofficially. This point tries to emphasize the need to change and record changes in any measurement, to improve the quality of the information being provided, the understanding of the context and the limitation of what the measurement can provide. Even something as clearly defined as a bug tracking software can be change to suit changes in information needs. This may be minor changes, such as providing a view of fault distribution in addition to number, or greater changes, such as adapting the tool to count only fixed bugs or to collect other types of information.

While the measurement itself becomes less standard as a result of this idea, it should provide a better fit between the information being offered and the information need of the various stakeholders. Adapting to a changing set of information needs in a fluid environment might prove a better option than following a strict plan, yielding more relevant and useful results.

- **Do not ignore less conventional information sources.** The Agile approach emphasizes the importance of individuals and interactions over processes and tools. This item is guided by the same underlying philosophy. A lot of measurements focus on documents, databases and processes. While this yields a lot of useful information, there are situations where such data sources do not have the appropriate information or cannot deliver it in a timely manner.

  Individual and interaction centric measurements may contain useful information and, in some cases, may be the only source for a particular type of information. In other cases, information from these sources may provide the necessary information in a timelier manner than any other sources.

  All information sources and all collection mechanisms have their limitations. In order for a measurement to provide information that a decision can be based upon it needs to be relevant, accurate enough and well analyzed in the context that it is meant to functions. All these limitations have to be clearly understood and factored into the analysis. Once all these sources, the information they can provide, the limitations they have are understood and the proper analysis is performed, these sources can provide useful information. Moreover, information derived from planning meetings or other less conventional sources may be available much sooner, albeit with some reduction in accuracy. This is perfectly acceptable, however, in certain circumstances, when time is more important than accuracy.

En example of this is when dealing with planning and planning relevant information. In many cases such information is based on estimation, so accuracy is not a central factor. Often, however, the availability of such estimations at the proper time can affect management decision. As such, timely availability of the information can be beneficial to the company's activities.

- **Focus on building competence for collecting the measurement within the company.** The measurement process should be a continuous activity that provides relevant results for the company. Ensuring the relevance of the results can be done initially by changing the measurement to fit the information needs of the owners. But in the long run it is important that the company develops the competence to collect and adapt the measurement to its needs. This can be achieved by involving stakeholders as much as possible in the development and collection of the measurements. To this purpose the CMO has information regarding items that are likely to motivate or de-motivate stakeholders, as well as how the measurement can affect their activities and how it might change the existing process.

  The importance of building competence within the company is closely related to ensuring that the measurement set that has been selected for implementation will continue to function properly. Moreover, in addition to ensuring that the competence for collecting the measurement is build, the ability to modify and adapt the measurement to local needs should also exist within the company.

  The overall aim of this recommendation is to provide the company with a working measurement as well as the tools to keep that measurement working with as little external effort as possible. A functioning measurement is not just one that functions now, but one that can be adapted and updated to keep it relevant and useful for as long as possible.

While trying to make a measurement as flexible as possible does make it difficult to propose exact steps to be followed while implementing, the above items are provided as guidelines. With every implementation the context and circumstances differ, so the measurements being implemented should fit the given situation as closely as possible. The purpose of the items is to provide some basic principles regarding how such an objective can be achieved. It's important to emphasize that the items are no more than suggestions and are themselves subject to change. The context and circumstances may dictate other priorities, in which case each of the items will have to be adapted to the certain situation.

Overview of the EMF

The EMF uses the same concepts of Goal, Question and Metric, with a different approach with regards to the creation of the model. The goals are analyzed and defined first, using information gathered as a result of the VAMOS framework's Define Phase. The potential metrics that are likely to fulfill the goal are analyzed next. The VAMOS framework's CMO addition provides the information needed to ensure that only the appropriate metrics are factored in as alternatives. The local information need is the main factor in the development of a metric within the Exploratory Measurement Framework. Since the purpose is to adapt the metric to information needs, both local and those identified by VAMOS, they drive the changes to the metric.

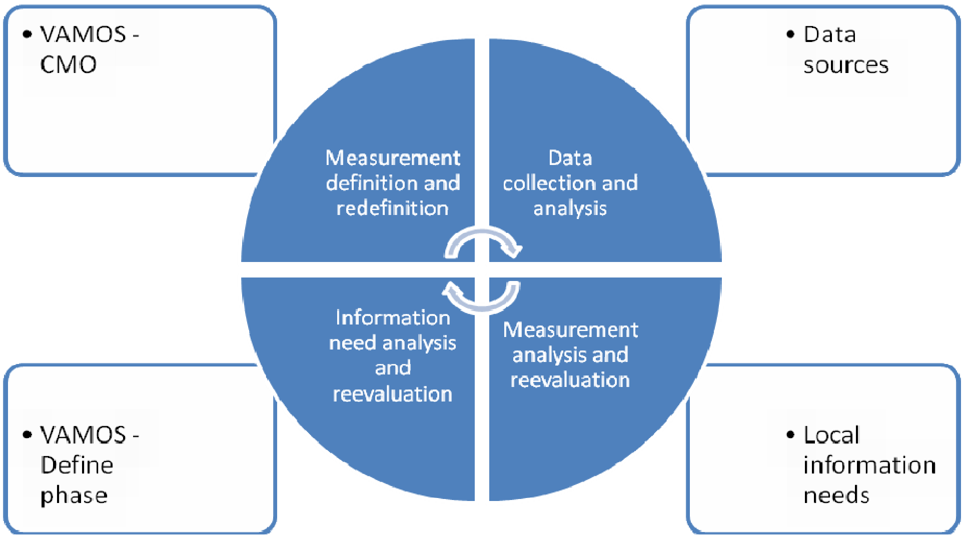

Figure 10. The Exploratory Measurement Framework

In the current implementation, the owner of the local information needs, i.e. the Integration Responsible, was also the owner of the EMF metric. The reason for this was to ensure that the local information needs are the main consideration behind the changes brought to the metric. This was all the more appropriate in the given setting. The overall goals were loosely defined: to find out more information about the Integration Phase and the problems met during this Phase. Thus, the local information needs were a more refined version of the higher level goals.

In the case of conflict between the higher level goals and the local information needs, a compromise needs to be reached and a clearer definition of the expected results might be required. Such discrepancies and contradictions, however, form a problem of their own that is outside the scope of this tool.

With both the goals defined and the potential metrics stated and analyzed, an iterative process is started, that consists of 4 stages.

- **Measurement definition and redefinition.** The metric in questions is first defined, taking into consideration the information needs, both local and those identified by VAMOS, data sources and other considerations such as process impact and motivation. After the first iteration, this stage is where the measurement is modified and re-designed, based on the information becoming available as a result of the previous iterations. Changes in the degree to which the previous implementation fits the information needs, the data sources available or in the information needs themselves are factored in the development of the new version.

- **Data collection and analysis.** The measurement is implemented in this stage, data is collected and analyzed. It is important that the measurement be implemented in a real setting, so that the practical obstacles as well as the benefits can be identified. An added benefit to the implementation of the metric in a real setting is that it builds confidence in the metric and is useful practice in its use.

- **Measurement analysis and reevaluation.** The performance of the measurement is analyzed, with respect to existing information needs, expected results, implementation obstacles and opportunities and any other ideas and information that become available during the course of implementation.

- **Information needs analysis and reevaluation.** The information needs are reevaluated an analyzed in the light of the new information becoming available. The information needs that are most often updated are local information needs. However, by adopting the proposed modifications to VAMOS, especially making the Define step a continuous and permanent process, information needs at the higher levels can be updated as well. Since VAMOS has the tools to identify and analyze higher level information needs, the framework itself is responsible for pushing the needed information to the EMF. The Measurement analysis and reevaluation and the Information needs analysis and reevaluation stages can be performed simultaneously and with a high level of exchange of information between them. Together they form the basis for the proposed modification to the tool in the following iteration. This is also where the decision can be made that further modifications are no longer needed and the metric has stabilized.

It is important to note that even when the metric is stabilized, the constant analysis of both metric and information needs has to continue. The frequency of the iterations, and thus the analyses, can be reduced for the purposes of cost reduction, but regular reevaluations should be performed. This will ensure that the metric continues to be relevant and to provide useful information. Constant reevaluation would lead to frequent minor changes that would both improve the quality of the results and the overall confidence level in the metric.

EMF in practice

Context

Once the Measurement Workshop was held the measurement to be implemented was chosen. The information needs considered of the higher priority were those related to planning information, namely the measurement of delays and relating delays in the integration work to possible causes. At the time of the adoption of the measurement there was little information about the types of problems encountered during the Integration Phase, the extent to which each problems caused delays, the frequency with which the problems were encountered or the seriousness of each problem. Some knowledge relating to possible problems did exist in the company, but it was incomplete and un-quantified.

The effort to fulfill the information needs as stated above was made more difficult by the fact that all the information about these, although discussed and analyzed during the development process, was not written down or measured in any way. Thus, the classical approach of using automated tools to look into documents or analyze databases was less feasible and costs associated with such a method were likely to be quite great.

A complicating factor was that during the Integration Phase, the pace of the work varies greatly. In practical terms, this means that in some situations there was time to do all the needed data collection and analyses while in others time was too short. Insisting on data collection in situations where deadlines were looming and where the pace of work became frantic was likely to produce rushed analyses with an uneven quality and inconsistently accurate. Moreover, insisting on such analyses would have raised morale problems, since most of the employees prioritized completing their tasks in a timely manner over collecting information regarding reasons for potential delays. The measurement that would be selected would have to be quite robust, in order not to allow the odd gaps in collection to affect the overall result. In more fast-paced situations the data collection should be postponed, to allow the personnel to focus on the urgent matters of meeting the deadlines. This postponement, however, should not affect the overall quality of the information being obtained and should have little impact on the rest of the data collection process.

A final factor in the selection of the measurements was the emphasis on availability of the information and the reduction in costs, in terms of time and effort, over the accuracy of the measurement. The measurement needed would have to be light weight, i.e. with a low cost, both in terms of the initial setup costs and regarding collection and maintenance costs. Given that the information need prioritized was related to planning issues and considering that there was little information available at the time on the subject, accuracy was considered a secondary factor. This enables the use of data sources considered less accurate, such as meetings and planning estimates, while reaching the accuracy goal. It was stated that the planning related information that will be obtained from this metric will be an estimate and will be used as an estimate, therefore it needs to be no more accurate than that. But the information does need to be available as quickly as possible, to enable the decision making process to be improved.

The decision was made to focus on the one data source that contained all the required planning, estimate and problem information that was required: the daily Integration planning meeting. As stated previously, all the required information and analysis expertise already existed at this stage, problems and their causes were discussed, what the measurement had to do was to collect all this information in a light weight manner.

The measurement

The measurement proposed for this role is a flexible information collection tool, the purpose of which is to obtain certain types of information from planning meetings. The measurement itself consists of a set of two or three questions related to the delays encountered and a classification of possible reasons for the delays. The classification is displayed on the Integration Board, to allow all the people involved at the meeting to understand what kind of information is needed and to suggest improvements. The people present at the Integration Meeting, as described in the RUAG process breakdown, are those responsible for different aspects of the Integration Phase. Each of the people involved in the Integration Meeting is in the best position to evaluate the situation of the aspect they are responsible for, how much of a delay they have in their area and the likely reason for that delay. They are also in a position to know how delays in other departments are likely to affect their activities.

Table 6. CMO analysis for the Flexible Information Collection Tool

Number	12
Metric name	Flexible information collection (at meetings)
Goal	2. 3.
Prerequisites	Regular meetings of the project team.
Description	The Integration Responsible (although it can be aplied to other areas) collects the data he is interested in. Once the questions stabilize, the process can be formalized.
Type	Process. Product.
Risks	The meetings could last too long as a result, leading to time waste and drop in interest.
Robustness	High. The method is designed to be extremely flexible and adapt to the information need.
Control mechanism	None (The method is driven by the person collecting the information and who has the information need).
Setup effort	Low. Little preparation is needed, especially for the initial, exploratory, part. A few iterations might be needed before the questions are completely understood by the entire team.
Maintenance effort	Low. The running cost is just the time taken to record the information. Changes are encouraged, and so incur little cost.
Accuracy	Medium-Low. The method relies on estimation, and that decreases accuracy significantly. However, since the data is collected quite often, there should be little information loss.
Analysis	Medium: data can be analyzed without reference to other documents.
Degree of process change	Low. The introduction of the questions would fit smoothly into the current process. Changes in the questions would also cause little impact on the process.

Owner	Integration Responsible (Can be extended to other roles as well).
Performer	Integration Responsible (generally the same role as the Owner)
Stakeholders	Integration responsible, project team.
Accuracy change cost	Low. Within the limitations of the method, there is little cost for changing the detail of the questions.
Performer acceptance	Medium-High. There is additional effort associated with data collection. This will be offset by the fact that the performer also has the information need, and is thus motivated to conduct the measurement.
Owner acceptance	High. Information would be useful on a daily basis and within a wider context.
Motivating/Demotivating factors	- there is some cost associated with collecting the information + the information driver is also performing the measurement, so misunderstandings or mismatches are avoided + the method is highly flexible, allowing the modification of information collected and the introduction of new information
Limitations	The method is meant to be a starter, an exploratory search into the information need. Once the information need is clearly established, more formal and rigid methods can be employed.
Perfomance frequency	Once or twice a week.

In keeping with the overall focus on flexibility of the implementation of the VAMOS framework, the measurement is to be implemented in a flexible manner. The initial form of the measurement can be changed, both in terms of the questions being asked and in terms of the delay reason classification. As described in the CMO, the Owner is the Integration responsible, so it's his responsibility to define the information need and it's his prerogative to decide on changes in terms of the questions to be asked, the frequency and exact timing of the data collection and the solving of any other practical issued that might arise.

The implementation of the measurement changed in the course of the application, with the classification of the delays being updated as a result of practical information becoming available. An example of such a change is related to the category "Planning problems". Practical implementation revealed that the category was too broad, including both planning problems that could be foreseen, i.e. incorrect estimates or improper planning, and planning problems that could not be foreseen, e.g. use of new technologies and approaches that are hard to plan for and thus create difficulties. The problem was solved by adapting the classification to have distinct categories for each of those and make the distinction clear. The problem was all the more important since, during the project under study, a technological approach was used for the first time by the project team and delays associated with this technology were considerable. By making the distinction between the two categories, the delays caused by the new technologies were not misinterpreted. Another example of modification was the addition of the “Infrastructure problems” category, when the necessity for such a category became apparent.

The decision to change belonged to the Integration Responsible, with all the people attending the Integration Meeting having the possibility to propose changes. Data collection was performed twice a week, with the exact date when the data was to be collected also at the discretion of the Integration Responsible. As stated before, the reason for this decision was to allow the Integration Responsibility to shift the exact date in such a way as to avoid the times when the schedule was busier. This also had the effect of making the measurement easier to accept for the people attending the meetings. The lightweight approach, together with the flexibility to change the data collection time, helped obtain the necessary information without creating its own delays in situations where time was essential for meeting deadlines.

The final form of the measurement is that of a table, containing the categories and the areas of responsibility that take part in the Integration meeting. For each collection date, the responsible for each area states the delay they have had since the time of the previous data collection and split that delay according to the causes. The classification of the possible causes is displayed and an additional document containing the definition of each category is available to all those attending the meeting. This means that the meaning of each category is only discussed if there are doubts in specific cases and that the increase in meeting time due to data collection is only of a few minutes. For a meeting of about 10 – 12 people, the data recording time was consistently under 3 minutes.

The answers will be in terms of man days of delays that were incurred. The purpose is to identify problems that cause the most significant delays. For each of the questions mentioned there are a couple of sample answers. The purpose of these is to define the level of detail, to better explain the kind of answers expected and to help with the classification of the causes to some extent.

The problems that are to be recorded are unpredictable faults or problems that fall outside the scope of regular planning efforts. This means that the delay caused is the major unit of the effort, with due corrections for the number of people involved. The time spent on a fault that did not cause delays (due to slack in the planning or reserve resources available) is not recorded because it is considered to not have put the project behind schedule and thus that project is still on track.

The current classification

Below is the current classification of the potential causes for delays, as of version 4 of the measurement, and sample answers. Sample answers were deemed more useful for practical purposes than lengthy theoretical explanations. The document below is available to all participants at the meeting and other stakeholders.

1. Specifications are contradicting or incomplete

    - A module is delayed for 8h. The cause is that the specifications they were to follow did not contain all the necessary information.

    In this case the number recorded is 8h of delay (for one person only).

2. Specifications are changed

    - A change in the specifications causes a software module to be delayed 4h. Two people worked on that module. Since the specifications changes are to be expected, to a certain extent, additional time spent by the specifications responsible will not be counted here.

In this case the time to be recorded is 8h (4h of delay for 2 developers). The reason for the changed specifications and the impact of the changes on specifications work will be dealt with if any delays are incurred there.

3. Incomplete or incorrect manuals (includes incomplete, inaccurate or unintelligible information)

    - The manual for, or any other description of, HW component X lacked information about the interface for a given function Y. The software module that needed to interact with that interface was delayed so that the missing information could be obtained from the HW department. Two people were working of developing the software module and their work was delayed for 3 hours while they gathered the information they needed.

In this case the cause is 3. Incomplete or incorrect manuals, time spent on the problem is 6h (2 people x 3 hours)

    - The manual for HW component X gave certain information about the interface to the function Y. This information had been changed and thus the information in the manual no longer conforms to reality. The software module Z, that needs to interact with that interface has been completed and delivered, but failed the integration tests, due to the interface errors. The problem was identified and fixed, but the delivery to that module is now delayed by 8h.

    In this case the delay is 8h.

4. HW problems

    - Fairly self explanatory

Time to be recorded is the amount of the delay for all the people affected by it. The estimates for the time effect on each of the departments are to be made by the responsible for each area. If replacing a HW board delays the work of 4 people for 2h, for instance, the number recorded will be 8h.

5. SW problems

    - Fairly self explanatory

Time to be recorded is the amount of the delay for all the people affected by it. If removing a software failure delays the work of 4 people for 2h (by delaying the availability of inputs for the validation department, for instance) the number recorded will be 8h.

6. Timesharing, unavailability of resources (personnel, equipment, other departments, licenses, etc.)

    - A software module was delayed for 4 hours because two of the people involved were requested to work on different projects (or modules; Say a test rig is only available then and takes priority. In this case, if the software module is put on hold and suffers delays, the cause is still resource overlap). They were called upon to do so at different times, so at no time were they both unavailable.

The effort to be recorded here is 4h. Since only one person was absent at any one time, there is no need to add that effort as extra.

Note: even if the people had to spend less than 4 hours working on the other projects, the delay caused by their absence will be recorded.

- A software module is ready for testing. No test equipment is available at the time so the module is delayed 5h, waiting for the test equipment to become available.

The effort recorded is 5h, since that is the delay the overlap caused. If the module had to wait for 6h for the test equipment to become available, but a 1h delay was expected and planned for, the amount recorded will be 5h.

7. Planning problems for known work

    - A document has to be ready at 14:00, and is estimated to take 1h. All the inputs are ready and the task assigned at 8:00. The person responsible for the document delays doing it until 13:00, at which point the text editor crashes and is down for an hour. The document is delayed by 1h.

In this case the cause is faulty planning, not an infrastructure fault. The effort is 1h.

Note: If the person responsible for the document is called to work on another project from 8:00 to 13:00, rather than voluntarily postponing, this becomes a resource overlap problem. The problem caused a 1h delay, so the time recorded is 1 hour, even though the person had to work at a different project for more than 1 hour.

- Developing a software module involved the writing of a document that takes about 2h. The developer estimates his time without the document and, as a result incurs a delay of 1 hour.

In this case the reason is faulty planning and the time to be recorded is 1 hour (namely the delay caused).

8. Planning problems for new technologies

- The Flash is a technology currently being employed for the first time. This causes delays due to inexperience in working with this particular technology. These delays are recorded in a separate category.

9. Infrastructure problems

- Repeated crashes of the text processor application caused a delay of 2h in the delivery of a document. The fault was fixed within 3 hours, but a 1h delay was planned for.

In this case the time recorded is 2h, the delay imposed on the plan.

- Test equipment had to be restarted and re-initialized repeatedly during the testing procedure. This lead to a situation where a test suite planned to take 3 hours took 5 instead. The particular scheduling of test equipment use also lead to an overall delay of 4h.

In this case the record will show a 2 hour delay due to infrastructure problems. The reasoning in this case is that the time plan cannot capture details of the use of particular equipment. Moreover, the need to re-start and re-initialize the test equipment is one that is discovered well into the testing process, and thus cannot be planned for. It cannot, thus, be considered faulty planning. The additional 2h, leading to a 4 h overall delay, will be assigned to Resource overlap, since availability of the test equipment would have avoided the further delay.

Notes on the current classification

While the document describes the data collection in terms of man hours, the decision was made by the Integration Responsible to switch data collection to man days. The reason is the data being collected is easier to estimate in this form for the people attending the meeting. The added accuracy of using man hours for the purpose is irrelevant since the increased estimation difficulty would negate that benefit.

Additional developments and tools

Another practical aspect of the application of the measurement is that of ongoing failure investigations. These are situations where problems have already arisen and caused delays, but their causes have yet to be identified. These problems are written on the Integration Board as Key Issues, and are often the cause of considerable delays affecting several areas of responsibility. The ability to collect information regarding these issues is a useful addition to the tool.

This type of situation is hard to deal with in the manner presented above in a clear and accurate manner. This is the reason an auxiliary tool was developed. The tool takes its name from the problem it's trying to solve, Key Issue Recording Tool, and it consists of a table where each of the key issues currently under investigation are noted by means of a unique ID and every participant in the meeting states the delay this has caused. When the investigation is finally closed and the cause of the delay discovered and classified, the exact numbers and time breakdown are available for an accurate and relevant further analysis.

Results

The purpose of the tool is to obtain more information about the reasons some of the delays occur. A table of the results can be seen below, expressed in terms of percentage of the overall delay.

Table 7. Results of the Flexible Information Collection Tool

		Integration Work	System Specifications, Manual, Prerequisites	HW - Processor Module	HW - MotherBoard, Telemetry	Software	Test Equipment, Tools, Infrastructure	Test Specifications, implementation, debugging and integration	Validation - Test Application Software (TASW)	Validation - PC-side test development - p1	Validation - PC-side test development - p2	Validation - develpoment of Boot Validation TASW and PC	Validation - reviews	Problem Total
1	Contradicting or incomplete specifications					8								8
2	Changed specifications													0
3	Incomplete or incorrect manuals						2							2
4	HW problems													0
5	SW problems					8			4					12
6	Timesharing, unavailability of resources (personnel, equipment, other departments, licenses, etc.)			4		12		10			2	4	4	36

7	Planning problems for known work											4		4
8	Planning problems for new technologies			8		20	2		6					36
9	Infrastructure problems								2					2
	Dept Total	0	0	12	0	48	4	10	12	0	2	8	4	100

The data seems to indicate considerable delays, about 48% of all delays, to be found within the Software department. This result should be, however, provided with the appropriate context. Data collection began after the hardware design and development had finished, thus hardware problems and delays in that area remained unrecorded. Moreover, delays in the hardware department, late in the development process, tend to cause increasing delays for all departments affected by the changes.

Two conclusions can be drawn from this result. First, the results show that Timesharing and Planning problems for new technologies are the leading causes of delay. When it comes to applying new technologies and processes, some delays are to be expected. Relative inexperience in the new technology makes estimation difficult, while unforeseen problems that arise during the course of practical application may cause extensive delays. It is to be expected that delays relating to this area will diminish, as the technology become more familiar and the more common problems are overcome.

The second conclusion that can be drawn is a direction where further research can be performed. The high delay associated with the software department can be investigated in further detail, to determine more exact causes for the delay. This is the type of result that is to be expected from exploratory studies: an area of the process that requires investigation in more depth. A more detailed breakdown of the problems encountered by the software department should, however, be the subject of another study.

The EMF in relation to other tools and approaches

The Exploratory Measurement Framework relies heavily on concepts derived from the Goal Question Metric approach [10]. The notion of Information Goal as applied in the EMF is similar to the Goal in GQM, as is the notion of metric. The difference arises from the way these concepts are implemented.

The GQM approach is based on a top-down method of creating the model. The goals are defined first, then the questions that need to be asked in order to answer the goals and, last, the metrics that are needed to answer the questions. The purpose is to clearly define what information is needed and then to ensure that all the metrics needed to provide that information are implemented.

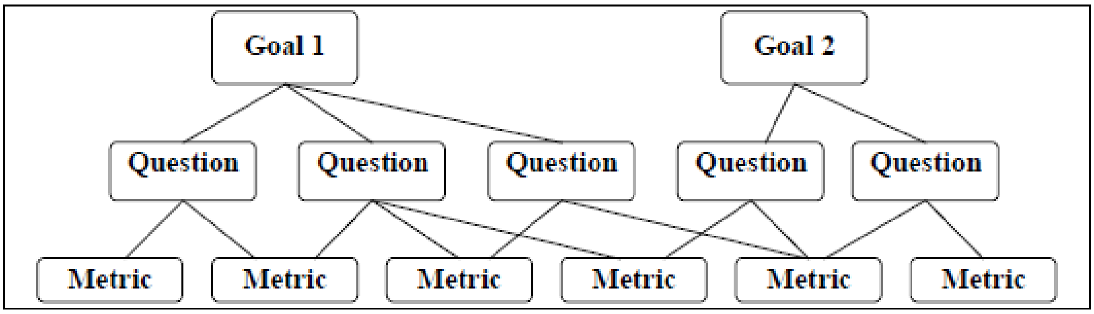

Figure 11. The Goal Question Metric approach [10].

The GQM approach suffers from the disadvantage that once the goals and questions are defined, the metrics needed to provide that information may be too expensive, impractical or even impossible to implement. In this situation, the goals and questions need to be reevaluated, their priority and scope analyzed in the context of the new metric information. This means that the entire process might be more costly, in terms of time and resources, than might be initially foreseen.

Moreover, the definition of the goals takes place at a higher level than the analysis regarding the cost and possibility of implementing the metrics. This means that considerations such as impact of certain metrics on the process, issues regarding motivation and mechanisms for ensuring reliability of the metrics might be overlooked, in turn affecting the resulting information.

The same concepts of Metric, Question and Goal are also used in a measurement system developed for Ericsson SW Research and defined in [8]. That system, however, creates the model in a bottom-up way: the measurements that can be implemented and automated are defined, the questions they answer are stated and then everything is linked to the goals they fulfill and the stakeholders that are interested in pursuing those goals. This approach seeks to ensure that all the measurements that are implemented can be linked to a goal and a stakeholder. Thus, no measurements are implemented that cannot be linked to a goal and therefore are not useful.

This approach solves the problem identified with the GQM: by starting the analysis from the metrics, it ensures that all the metrics that are being proposed and implemented can be implemented at an acceptable cost. Those metrics that do not answer any question and that cannot be linked to any goal are simply dropped, with no additional analysis required.

The bottom-up approach does suffer from the drawback that some of the goals may remain partially or completely unhandled. Since the system defines the metrics and then links them to the questions that they can provide answers for, there is a chance that some of the questions will receive incomplete answers or remain unanswered at all. This problem persists when linking the questions to the goals they are trying to achieve.

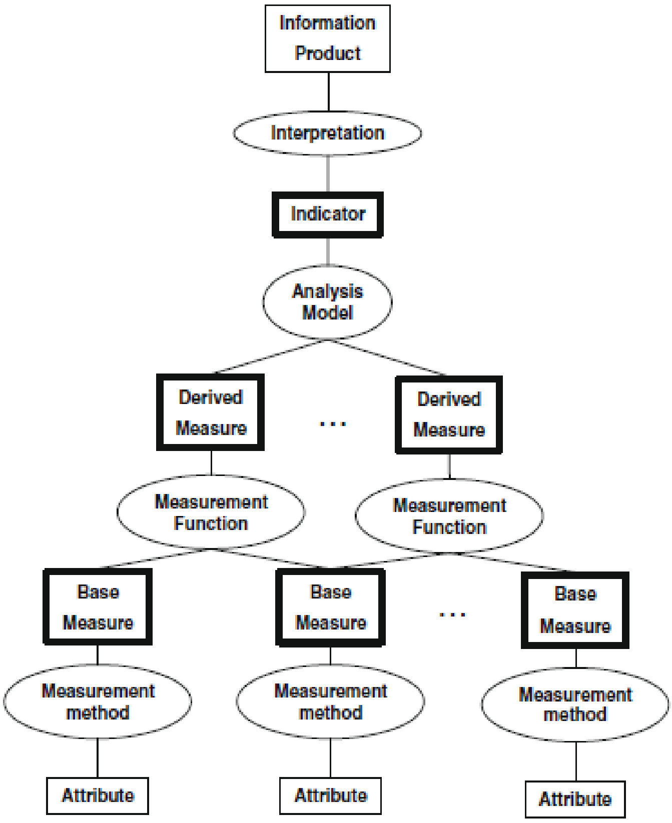

Figure 12. The measurement information model - adapted from ISO/IEC 15939:2007 [8].

EMF and VAMOS

While both frameworks rely on an iterative approach aimed at improving the quality of the measurement process, they work at completely different levels of abstraction. VAMOS is a high level tool that enables the users to alter the selection of measurements. It provides the tools for high level analysis of the setting where the measurements are to be implemented, the company itself, and the measurements and can collect all necessary data by means of these tools.

The EMF seeks to provide flexibility and adaptability to lower level metric development. The purpose is to provide a tool that, based on the information needs, priorities and data sources available, can provide a useful measurement and can modify that measurement to constantly fit the requirements. The EMF has few tools to collect and analyze information. Instead it relies on higher level frameworks, such as VAMOS, to provide the appropriate information, as well as on the expertise of the metric owner and stakeholders. This makes the EMF one of the tools that VAMOS, or other higher level frameworks, can use, rather than a standalone metric.

From the perspective of VAMOS, EMF is an approach to metric development appropriate for certain situations. During the operation of the VAMOS framework, the EMF requires certain information from the framework, especially regarding the process and the information needs that have been identified, but the internal workings of EMF are not visible since they function at a different level of abstraction. The metrics that are identified and analyzed during the operation of VAMOS are not identified as EMF, but rather are clearly defined in terms of overall purpose and methods. The EMF is mentioned as the means by which the metric is to be developed, but not as a metric itself.

From the EMF perspective, VAMOS is a key source of information regarding process and information needs. While VAMOS is a key stakeholder and information provider, the owner of the EMF and the person ultimately responsible for it is a single individual, identified when the decision to use the EMF is taken. The purpose of this is to clarify the responsibility and to clearly identify the decision-makers.

Conclusions on the Implementation

The implementation of the measurement showed the viability of the EMF approach and the VAMOS framework as a whole. The initial research, analysis and the decision process, implemented according the recommendations of the updated version of VAMOS, helped make the selection of measurement relevant and interesting, both from an academic and an industry perspective.

The implementation itself showed the benefits in applying the EMF approach to an exploratory study in an area of activity where little information is available. The flexibility underlying the approach allowed the measurement to continue to be relevant even in the rapidly changing context caused by the availability of new information. The implementation of this prototype provided a lot of useful practical data that could not be foreseen and helped confirm some of the assumptions that the VAMOS framework and the EMF approach are based on. While additional projects and implementations will, undoubtedly, provide more data and propose improvements, the both concepts have been validated.

Chapter 8 - Discussion and Conclusions

This section compares the initial research questions and expected outcomes against the actual results of the thesis. This is meant to discuss the overall accomplishments, benefits and validity threats to the entire work. In-depth individual analyses of the benefits and drawbacks of each of the components are included in the appropriate chapters.

Overall view of the VAMOS implementation

The overall VAMOS implementation was aimed at proving the viability of the VAMOS framework in the given context. The current implementation was limited both in scope and time.

One limitation of scope was that of scale: the framework was applied only to the Integration Phase, rather than the entire process. The other was the little information that was available on the Integration Phase, that lead to the need to perform an exploratory study into the area, developing and implementing a measurement that would provide the initial information needed. The use of the concept of Fault Slip-Through is difficult to validate within the confines of these limitations.

The limitations did, however, allow this implementation to explore and validate the framework's capability of selecting and implementing new frameworks. In a situation where measurements had already been in place, the testing and validation of this part of the framework would have proven difficult to achieve. The current implementation proved the VAMOS framework's suitability for allowing the proper research, selection and implementation of measurements, even in contexts where little information is available. These aspects of the framework have been validated and a roadmap to implementation of new measurements has been created.

Overall, the implementation was successful, proving the viability of the concept and providing useful information for further efforts. Identifying information needs, their stability and ownership are crucial to any measurement effort and this implementation has validated some of the methods of obtaining these items of information.

The Code Inspection Sheet Analysis Tool

The Code Inspection Sheet Analysis Tool can be seen both by itself and in the context of the VAMOS framework. By itself, the tool provided useful information about the code inspection validation activity, the costs associated with it and its effectiveness. The tool provided a new approach to measuring the impact and the effectiveness of code inspection, by dispensing with the existing approach of counting the faults and focusing on the number of rejections that each fault causes. While the approach can be improved upon, the tool was deemed sufficiently accurate for the purpose and development on it has been stopped. One reason for this is the low scope for improvement in the area of Code Inspection. Since code inspection can only catch implementation faults, even major improvements in the area cannot translate into major cost reductions.

From a VAMOS perspective, the tool had benefits beyond the results it produced. Development of this tool proved the importance of properly understanding the information need and the goals of the company. The company's focus on the cost aspect dictated the approach taken for the measurement and that, in turn, heavily influenced the results. Proper understanding of the company's goals and information needs became a priority and led to changes being proposed for VAMOS itself.

In addition, development of the tool provided valuable information in terms of information needs and goal stability. As more information became available about the results provided by the tool, the priorities of the company shifted. As the Code Inspection activity was found to be appropriate in terms of effectiveness and efficiency, the focus shifted to other areas that were considered more wasteful. Since improvements in terms of Code Inspection were reduced in scope and impact, the goals of the company changed. This proved a useful lesson regarding the need to constantly evaluate the company's information needs, goals and priorities and the need for flexibility within the framework, to allow for the changing goals and priorities. This lesson was later incorporated and used both in the updated version of VAMOS and in the Exploratory Measurement Framework.

The Exploratory Measurement Framework

The Exploratory Measurements Framework developed from the need to allow the information needs and goals that drive a measurement to adapt to the information becoming available and to changing environments. This is especially important in situations where little information is already available and few if any measurements are already in place. This setting requires and exploratory study to be performed and this, in turn, requires the flexibility to adapt the measurement and the goals driving it to the information becoming available.

The concept of adaptable measurements, as applied to an exploratory setting, and the Exploratory Measurements Framework as a tool were validated by this implementation. In an unstable context of an exploratory study, the measurement developed by means of this tool provided useful information both in terms of practical issues and in terms of the further development of the study. In practical terms, the tool provided useful information regarding the delays that were encountered and their causes. While the information was based on estimates, it is important to note that these are expert estimates and that additional accuracy was deemed unnecessary in the given setting. In terms of information relating to the further development of the study, areas that had the most delays were identified as subjects for further research. This meant that only those areas that show the highest delays, and thus the highest potential for improvement, will be investigated further, reducing the costs associated with the study.

Overall, the Exploratory Measurement Framework has proven the viability of the concept. While additional information is still useful and further research into the area is required, the concept has been validated and has proven a useful addition to this study.

Research questions

1. How can a prototype of the VAMOS framework be implemented in an industrial setting?

The prototype implementation is presented throughout the paper. Some of the steps taken were modified, with elements added or removed as necessary. Each of these modifications, as well as the reasons for making them and the results of the modifications are discussed in detail in the respective chapters. Overall, the prototype validates the framework, shows that the framework can be implemented successfully and can provide useful information.

2. What modifications need to be made to the framework to make implementation possible and cost efficient and to minimize the effect on the company's process?

The modifications proposed for the framework are discussed in more detail in the chapter "Changes and updates to VAMOS". Some of these modifications were included in the current implementation, while others are yet to be validated in practice. In both cases, the particular details of each implementation are to be decided depending on local factors.

3. What measurements does the framework need to use?

The measurements that were needed during this implementation are presented under the "Measurement implementation" chapter. Due to time constraints, only one measurement was implemented completely. However, several measurements were proposed, analyzed and presented to the company. The analyses for these measurements are available as part of the MOM component of VAMOS.

4. How does the metrics flexibility and adaptability translate into practical tools and methods?

A flexible metric was needed for the exploratory investigation of the Integration Phase. Such a metric was practically implemented as part of this thesis and its results are presented in the "Measurement Implementation" chapter. The tool itself was proven practically to be useful addition to the process and an important information collection mechanism.

5. How can the concepts of flexibility and adaptability be adapted to measurement development? How would an agile measurement development framework be defined?

The EMF, Exploratory Measurement Framework, describes a general framework aimed specifically at developing metrics according to agile principles. The framework is described in detail under the chapter "Exploratory Measurement Framework".

Validity threats

The current implementation was aimed at providing the VAMOS framework with a practical implementation, the lessons that result from such and effort and the proven validity provided by the practical results.

Admittedly, the current implementation is a prototype. It was conducted on a specific part of the process, on a relatively small scale in both scope and resources. More iterations are needed to prove the current implementation's long term validity. Moreover, more practical implementations would be very useful in investigating the effect various processes, environments and company organizational cultures have on the implementation. The current implementation is focused on the aerospace industry. The particularities of this industry have influenced both the development of the framework and the implementation efforts. An interesting follow-up would be an attempt to implement the framework in other industries where software reliability is a vital factor.

Conclusions

Overall, the current project proves that the VAMOS framework is a useful tool in analyzing and managing validation activities. More work is needed for a conclusive and a more general validation. The current project does, however, prove the validity of the framework, albeit in a restricted environment. The development of additional tools, meant for use with VAMOS, prove the framework's flexibility and adaptability. The information resulting from this implementation can be included in further attempts and lead to further development and refinement of the framework, thereby increasing both its usefulness and its applicability. The development of additional tools, meant for use with VAMOS, prove the framework's flexibility and adaptability.

References:

[1] J. Schulte, *A Software Verification & Validation Management Framework Industry for the Space Industry*, Blekinge Institute of Technology 2009.

[2] P. Preissing, *A Software Verification and Validation Management Framework for the Space Industry,* Technische Universität München, 2009.

[3] M. V. Mäntylä, C. Lassenius, *What Types of Defects Are Really Discovered in Code Reviews?,* IEEE Transactions on Software Engineering, VOL. 35, NO. 3, MAY/JUNE 2009

[4] L.-O. Damm, L. Lundberg, and C. Wohlin, *Faults-slip-through - a concept for measuring the efficiency of the test process*, Software Process: Improvement and Practice, vol. 11, no. 1, pp. 47-59, 2006. [Online]. Available: http://dx.doi.org/10.1002/spip.253

[5] M. A. Wojcicki and P. Strooper, *An iterative empirical strategy for the systematic selection of a combination of verification and validation technologies*, in Software Quality, 2007. WoSQ'07: ICSE Workshops 2007. Fifth International Workshop on, 2007, p. 9. [Online]. Available: http://dx.doi.org/http://dx.doi.org/10.1109/WOSQ.2007.4

[6] M. Umarji, *Understanding developer resistance to software metrics***,**

[7] A. Rainer, T. Hall, N. Baddoo, *Persuading Developers to „Buy into" Software Process Improvement: Local Opinion and Empirical Evidence,* Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE'03)

[8] M. Staron, W. Meding, C. Nilsson, *A framework for developing measurement systems and its industrial evaluation*, Information and Software Technology 51 (2009) 721–737

[9] R. Feldt, B. Marculescu, J. Schulte, R. Torkar, P. Preissing, E. Hult, *Optimizing Verification and Validation Activities for Software in the Space Industry*, DAta Systems In Aerospace (DASIA), June 2010.

[10] V. Basili, G Caldiera, D. Rombach, *The Goal Question Metric approach*,

[11] RUAG Holding, "RUAG Aerospace Sweden AB", Website: http://www.space.se, 2010, last accessed: 2010-09-11.