A Template for Communicating Information about Requirements and their Realization

Mira Kajko-Mattsson and Jaana Nyfjord

***Abstract*—Structured and disciplined communication is a prerequisite for effective management of requirements. In this paper, we investigate what requirement management information is communicated within a software development cycle. We do this by studying the management of requirements information within one Canadian organization. Our results show that most of the information as designated in our template is recorded by the organization studied.**

***Index Terms*—requirements specification, tool, lightweight and heavyweight software development.**

I. Introduction

To aid in maximizing the quality of the development process, one should provide guidance for what information to collect about requirements management and how to structure it. Usually, such guidance is provided in form of templates.

In this paper, we investigate what requirements management information is communicated both within lightweight and heavyweight software development. We do this by creating a template of information required for describing and managing software requirements within a development cycle and by finding out how it is implemented in one Canadian company. We call our template *Software Requirements Management Template* (SRMT). Our primary goal is to elicit information that is needed for communicating information about requirements and their implementation within the development cycle. However, we do not aim at distinguishing which information is used in different development approaches. Our secondary goal is to find out the state of practice within the organization studied using the *SRMT* template as a basis.

The remainder of this paper is as follows. Section 2 describes the research method taken when conducting our study. Section 3 briefly presents the *SRMT* template covering information required for communicating[1] requirements and their realization. Section 4 describes the requirements information communicated within the organization studied. Finally, Section 5 provides concluding remarks.

Manuscript received January 7, 2008.

Mira Kajko-Mattsson is with the Department of Computer and Systems Sciences, Stockholm University/Royal Institute of Technology, Forum 100, SE-16440. Kista, Sweden. (Phone: +46-8-162000; fax: +46-8-7039025; e-mail: mira @dsv.su.se).

Jaana Nyfjord is with the Department of Computer and Systems Sciences, Stockholm University/Royal Institute of Technology, Forum 100, SE-16440. Kista, Sweden. (E-mail: jaana@dsv.su.se).

[1]By communication, we mean both oral and written communication.

II. Research Method

This section describes the research method taken during our study. Section II.A lists and describes the research steps. Section II.B describes the organization studied.

A. Research Steps

As a first step, we decided to create the *SRMT*. Hence, we started our work by studying current literature in search of publications suggesting any templates. Unfortunately, we were not very successful. The only publications we could find were [1][2][3][5] and the templates they suggested were quite coarse-grained. They mainly concentrated on suggesting general templates for how to describe requirements in the initial development phases, but not on how to communicate them during the whole development process cycle. Hence, these publications did not provide us with enough support for describing and managing requirements within development. They only constituted a starting point for outlining the first out of eight clusters of our preliminary template (see *General Requirements Description* cluster in Figure 1). This preliminary template was then complemented with the information found in various publications such as [4][8][9].

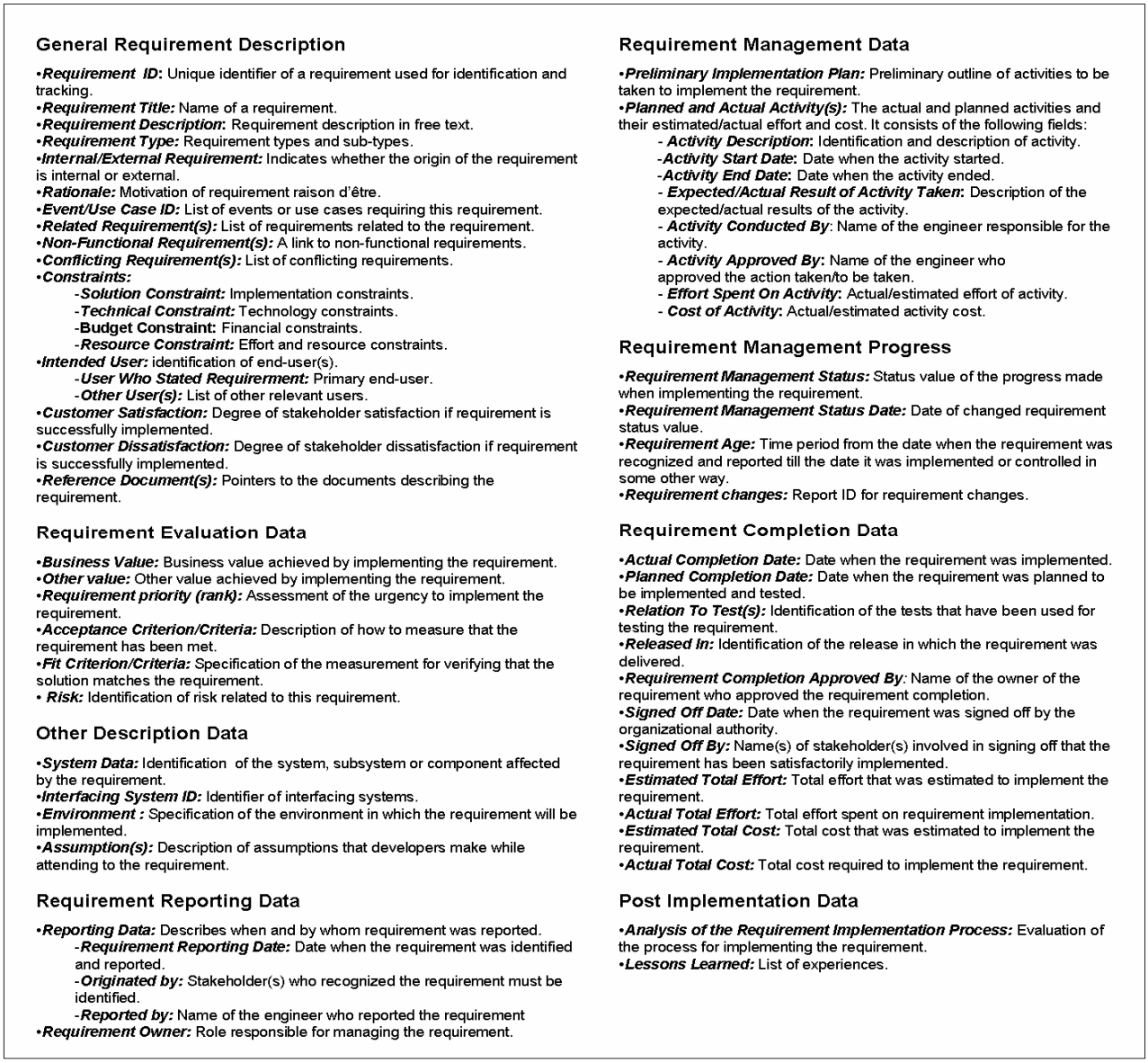
As a next step, we created a questionnaire. As illustrated in Figure 2, the questionnaire was open-ended and semi-structured. It focused on finding out the type of requirements information that was managed in our company.

The questionnaire consisted of two groups of questions, (1) introductory questions and (2) questions concerning the management of requirements information.

To cover the template, 130 questions were created. Due to space restrictions, we cannot list them all. However, the majority of them were structured according to the following pattern (1) does your organization record this information (attribute) 2) could you please provide an example, (3) if yes/no, please motivate why.

Not all types of information (attributes) were amenable to this pattern. Hence, the pattern had to be complemented with questions specific for each attribute studied. Examples of these questions can be found in Section B under *Complementary Questions* for specific fields in Figure 2.

As a next step, we interviewed one representative from our Canadian software company. For confidentiality reasons, we do not name this organization. It is however briefly described in Section II.B. The results from the interview have helped verify the usefulness of our template.



General Requirement Description

•***Requirement ID***: Unique identifier of a requirement used for identification and tracking.
•***Requirement Title:*** Name of a requirement.
•***Requirement Description***: Requirement description in free text.
•***Requirement Type:*** Requirement types and sub-types.
•***Internal/External Requirement:*** Indicates whether the origin of the requirement is internal or external.
•***Rationale:*** Motivation of requirement raison d'être.
•***Event/Use Case ID:*** List of events or use cases requiring this requirement.
•***Related Requirement(s):*** List of requirements related to the requirement.
•***Non-Functional Requirement(s):*** A link to non-functional requirements.
•***Conflicting Requirement(s):*** List of conflicting requirements.
•***Constraints:***
-***Solution Constraint:*** Implementation constraints.
-***Technical Constraint:*** Technology constraints.
-**Budget Constraint:** Financial constraints.
-***Resource Constraint:*** Effort and resource constraints.
•***Intended User:*** identification of end-user(s).
-***User Who Stated Requirement:*** Primary end-user.
-***Other User(s):*** List of other relevant users.
•***Customer Satisfaction:*** Degree of stakeholder satisfaction if requirement is successfully implemented.
•***Customer Dissatisfaction:*** Degree of stakeholder dissatisfaction if requirement is successfully implemented.
•***Reference Document(s):*** Pointers to the documents describing the requirement.

Requirement Evaluation Data

•***Business Value:*** Business value achieved by implementing the requirement.
•***Other value:*** Other value achieved by implementing the requirement.
•***Requirement priority (rank):*** Assessment of the urgency to implement the requirement.
•***Acceptance Criterion/Criteria:*** Description of how to measure that the requirement has been met.
•***Fit Criterion/Criteria:*** Specification of the measurement for verifying that the solution matches the requirement.
• ***Risk:*** Identification of risk related to this requirement.

Other Description Data

•***System Data:*** Identification of the system, subsystem or component affected by the requirement.
•***Interfacing System ID:*** Identifier of interfacing systems.
•***Environment :*** Specification of the environment in which the requirement will be implemented.
•***Assumption(s):*** Description of assumptions that developers make while attending to the requirement.

Requirement Reporting Data

•***Reporting Data:*** Describes when and by whom requirement was reported.
-***Requirement Reporting Date:*** Date when the requirement was identified and reported.
-***Originated by:*** Stakeholder(s) who recognized the requirement must be identified.
-***Reported by:*** Name of the engineer who reported the requirement
•***Requirement Owner:*** Role responsible for managing the requirement.

Requirement Management Data

•***Preliminary Implementation Plan:*** Preliminary outline of activities to be taken to implement the requirement.
•***Planned and Actual Activity(s):*** The actual and planned activities and their estimated/actual effort and cost. It consists of the following fields:
- ***Activity Description***: Identification and description of activity.
-***Activity Start Date***: Date when the activity started.
-***Activity End Date***: Date when the activity ended.
- ***Expected/Actual Result of Activity Taken***: Description of the expected/actual results of the activity.
- ***Activity Conducted By***: Name of the engineer responsible for the activity.
- ***Activity Approved By***: Name of the engineer who approved the action taken/to be taken.
- ***Effort Spent On Activity***: Actual/estimated effort of activity.
- ***Cost of Activity***: Actual/estimated activity cost.

Requirement Management Progress

•***Requirement Management Status:*** Status value of the progress made when implementing the requirement.
•***Requirement Management Status Date:*** Date of changed requirement status value.
•***Requirement Age:*** Time period from the date when the requirement was recognized and reported till the date it was implemented or controlled in some other way.
•***Requirement changes:*** Report ID for requirement changes.

Requirement Completion Data

•***Actual Completion Date:*** Date when the requirement was implemented.
•***Planned Completion Date:*** Date when the requirement was planned to be implemented and tested.
•***Relation To Test(s):*** Identification of the tests that have been used for testing the requirement.
•***Released In:*** Identification of the release in which the requirement was delivered.
•***Requirement Completion Approved By***: Name of the owner of the requirement who approved the requirement completion.
•***Signed Off Date:*** Date when the requirement was signed off by the organizational authority.
•***Signed Off By:*** Name(s) of stakeholder(s) involved in signing off that the requirement has been satisfactorily implemented.
•***Estimated Total Effort:*** Total effort that was estimated to implement the requirement.
•***Actual Total Effort:*** Total effort spent on requirement implementation.
•***Estimated Total Cost:*** Total cost that was estimated to implement the requirement.
•***Actual Total Cost:*** Total cost required to implement the requirement.

Post Implementation Data

•***Analysis of the Requirement Implementation Process:*** Evaluation of the process for implementing the requirement.
•***Lessons Learned:*** List of experiences.

Figure 1. Our software requirement management template

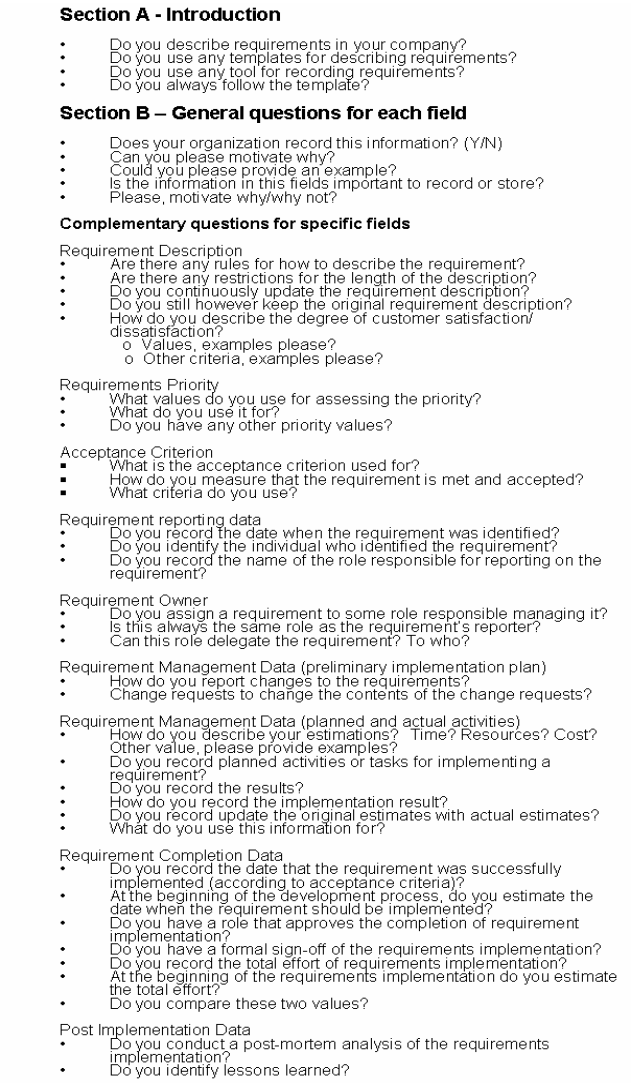
B. Organization Studied

Regarding the organization studied, we interviewed one representative (a process owner) of one Canadian systems development organization. The company was selected according to its relative ease of access, i.e. by the convenience sampling method [7]. The company develops products ranging from ground stations for satellite radar systems to e-commerce applications. It uses both lightweight and heavyweight software development processes.

III. TEMPLATE

The *SRMT* consists of eight clusters of information, each dedicated to a particular requirement aspect. As listed in Figure 1, each cluster covers a set of attributes bearing on coherent information. Below, we briefly describe the clusters.

- **General Requirement Description** describes basic requirement information needed for identifying, understanding, and classifying requirements [1][3] .
- **Requirement Evaluation Data** describes the data essential for evaluating and prioritizing the requirements [4][5].
- **Other Description Data** provides the context of the requirement and its management process [4]. It covers data regarding products, methods, projects, and the like.
- **Requirement Reporting Data** records when and by whom the requirement has been identified and to whom it has been assigned [4].
- **Requirement Management Data** communicates information about the requirement management process. It covers both planned and actual actions taken to implement the requirement, identifies roles involved in these actions, records effort required for implementing the requirement, and the effectiveness of the implementation activities[1][3].
- **Requirement Management Progress** tracks the status of the requirement implementation process essential for



Section A - Introduction

- Do you describe requirements in your company?
- Do you use any templates for describing requirements?
- Do you use any tool for recording requirements?
- Do you always follow the template?

Section B – General questions for each field

- Does your organization record this information? (Y/N)
- Can you please motivate why?
- Could you please provide an example?
- Is the information in this fields important to record or store?
- Please, motivate why/why not?

Complementary questions for specific fields

Requirement Description
- Are there any rules for how to describe the requirement?
- Are there any restrictions for the length of the description?
- Do you continuously update the requirement description?
- Do you still however keep the original requirement description?
- How do you describe the degree of customer satisfaction/ dissatisfaction?
  - Values, examples please?
  - Other criteria, examples please?

Requirements Priority
- What values do you use for assessing the priority?
- What do you use it for?
- Do you have any other priority values?

Acceptance Criterion
- What is the acceptance criterion used for?
- How do you measure that the requirement is met and accepted?
- What criteria do you use?

Requirement reporting data
- Do you record the date when the requirement was identified?
- Do you identify the individual who identified the requirement?
- Do you record the name of the role responsible for reporting on the requirement?

Requirement Owner
- Do you assign a requirement to some role responsible managing it?
- Is this always the same role as the requirement's reporter?
- Can this role delegate the requirement? To who?

Requirement Management Data (preliminary implementation plan)
- How do you report changes to the requirements?
- Change requests to change the contents of the change requests?

Requirement Management Data (planned and actual activities)
- How do you describe your estimations? Time? Resources? Cost? Other value, please provide examples?
- Do you record planned activities or tasks for implementing a requirement?
- Do you record the results?
- How do you record the implementation result?
- Do you record update the original estimates with actual estimates?
- What do you use this information for?

Requirement Completion Data
- Do you record the date that the requirement was successfully implemented (according to acceptance criteria)?
- At the beginning of the development process, do you estimate the date when the requirement should be implemented?
- Do you have a role that approves the completion of requirement implementation?
- Do you have a formal sign-off of the requirements implementation?
- Do you record the total effort of requirements implementation?
- At the beginning of the requirements implementation do you estimate the total effort?
- Do you compare these two values?

Post Implementation Data
- Do you conduct a post-mortem analysis of the requirements implementation?
- Do you identify lessons learned?

Figure 2. Our questionnaire

monitoring and controlling requirements[4]. It records the status value, the date when the requirement changes status values, the overall requirement implementation progress status value, and the requirement age.

- **Requirement Completion Data** covers information about the completion of the requirement implementation process [4][8]. It records planned and actual completion date, roles involved in approving and signing off the completion, and the total effort spent on requirement implementation.
- **Post Implementation Data** holds information on the post-mortem analysis of the requirement implementation process. The analysis results should provide an important feedback for improving the future requirements management.

IV. Interview Results

In this section, we present the results of our study. Our presentation is structured according to the eight clusters and their attributes as outlined in the *SRMT* template in Figure 1.

The organization studied documents the requirements in two ways: in a requirements management tool and in a separate document called requirements specification. Requirements specification mainly describes the requirements, but not the information about their realization. The tool, on the other hand, does both. Hence, some of the information in these two sources overlaps. When presenting our results, we present the results as recorded in the tool.

Many of the attributes as identified in Figure 1 are not always explicitly distinguished in form of a field in the tool. They may however be recorded in free text together with other information (other attributes). When presenting our results, we will point this out by stating that the attribute is recorded in free text.

A. General Requirement Description

Due to the fact that the organization studied follows the IEEE standard guidelines [2], it records the majority of the attributes as listed in the *General Requirement Description* cluster. The only attributes that they do not record are *Rationale*, *Budget Constraints*, *Resource Constraints*, *Customer Satisfaction* and *Customer Dissatisfaction.* Some of these attributes however may be recorded under different guises in later phases. Below, we report on how the attributes are managed.

- *Requirements ID:* All the requirements are uniquely identified with an ID. Usually, the ID corresponds to a numerical value. Some of the requirements however may be identified with an alphanumerical value, where a letter indicates the requirement type (functional, non-functional, or other).
- *Requirements Title*: In addition to an ID value, each requirement is identified with a title in the organization studied. A title is a short name of the requirement. It usually consists of several keywords. It is very helpful in doing manual searches in the tool. It allows one to quickly browse through requirements list without having to read the whole requirements description.
- *Requirements Description*: The organization studied describes its requirement in free text in an explicitly dedicated field for this purpose. The organization does not pose any restrictions on the description. The only restrictions they have concern the wording and the description length. The descriptions should use the words *Shall* or *May*. They should be short; one-sentence or two to three sentences per requirement at the most. If the description is longer, then probably the requirement has to be further broken down.
- *Requirements Type*: The organization studied classifies their requirements into three categories: *Functional*, *Non-functional*, and *Specialty*. *Specialty* requirements concern specific aspects of the system, such as domain, construction, and other system requirements. The descriptions of the three requirements types do not differ much. The merely follow the same pattern. However, the relationship between them is not formally managed. It is recorded in free text in the original functional requirement, e.g. "*this function has to perform according to XX specifications*".
- *Rationale*: The attribute for describing the rationale behind each requirement is not used by the organization studied. The interviewee did not even recognize this attribute and its purpose.
- *Even/Use Case ID and Reference Documents*: The use cases are always identified. Generally, the use cases and the *Operational Concept* [6] are produced first. They then

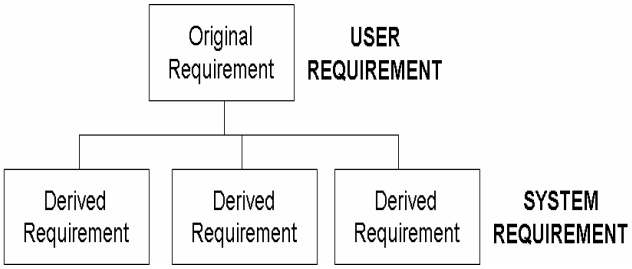


Figure 3. Relationship between original and derived requirements

provide a basis for specifying the requirements. Together with other relevant documents, they are identified as *Reference Documents*.

- *Related requirements*: Relationships among the functional requirements are always identified. In the tool, the related requirements are identified as a link. In a requirements specification document, a high-level requirement is described in one section. Its related lower-level requirements are described in its subsections.
- *Conflicting requirements:* The organization studied manages information about the conflicting requirements. Usually, however, they start identifying the conflicting requirements in the design phase where they encounter conflicts. If a conflict occurs, then a comment is added in the free text describing the requirement and the conflict.
- *Constraints:* The organization studied only indicates design or technical constraints. Budget and resource constraints are not very applicable on the requirements specification level. They are however more applicable in other higher level documents such as the *Operational Concept* [6].
- *Intended Users:* Information about the *Intended Users* is common, especially in IT type projects. The identification of the end-users is however implicitly provided by linking use cases to requirements.
- *Customer Satisfaction and Customer Dissatisfaction*: The organization studied does not collect information about customer satisfaction and dissatisfaction. To satisfy the customers, they mainly prioritize the requirements (as *Critical* or *Optional*) and create acceptance test specifications. The acceptance test specifications however are not recorded together with the requirement specifications.

Finally, we would like to point out that the organization studied distinguishes between two types of requirements: *Original* and *Derived*. As illustrated in Figure 3, the original requirements correspond to high-level requirements as provided by the customer or other internal role within the organization. The derived requirements, on the other hand, correspond to system requirements. They are derived from the original requirements. They correspond to the developers' understanding and interpretation of the user requirements. Because they are for internal use only, they are expressed in technical terms.

In order to trace the derived requirements to the original requirements, the organization studied relates them in a parent-child relationship. As can be seen in Figure 3, the original requirement is a parent, whereas the derived requirements are the children.

The original requirement description is kept unchanged. The reason is to create a fallback opportunity so that one can follow the history of a change. By seeing what was changed over time and why, one may avoid misinterpretation later on in the project. Therefore, the original requirements should not be modified. They should always stay intact. All modifications to them must undergo a formal change and approval process. This is due to the fact that changes to the original requirements may impact customer satisfaction, project scope, budget, or other factors. Hence, its change generally requires more formalism.

B. Requirement Evaluation Data

The organization studied uses only *Requirements Priority* and *Acceptance Criteria* in the *Requirements Evaluation Data* cluster. The other three attributes, *Business Value*, *Other Values* and *Fit Criteria* are not used. They may however be managed in other forms. Below, we report on the results for each of the attributes:

- *Business Value* is not recorded in the requirement document. The business value is recorded in the business case, which is a separate document produced at the business and product planning levels [6]. However, it strongly affects the value of the requirement priority.
- *Other Values:* This attribute is not used at all. The interviewee could not think of any other values that might be recorded in the requirement document.
- *Requirements Priority:* The organization studied prioritizes all their requirements by assigning either *required* or *optional* values to them. These values constitute a basic and minimum level of stating the priority. The organization also uses additional way of prioritizing requirements by assigning to them the implementation priority value. This value depends on various aspects such as whether the requirement is critical for initial operation, whether it provides a basis for negotiating the scope with the customer, and other aspects.
- *Acceptance Criteria:* The organization studied manages information about the *Acceptance Criteria*. This attribute however is not part of the current requirements management tool. Another tool is used for this purpose. At a minimum, the acceptance criteria correspond to the descriptions of the acceptance procedures. These procedures may either correspond to analysis, inspection, test or demonstration.

C. Other Description Data

The organization studied uses only two attributes in the cluster *Other Description Data*. They are *System Data* and *Interfacing System ID*. Using them, one identifies the system, subsystem or component affected by the requirement, and the interfacing systems. Regarding the attributes, *Environment* and *Assumptions*, they are not used within the organization studied.

D. Requirement Reporting Data

None of the attributes in the *Requirement Reporting Data* cluster are fully utilized within the organization studied. The *Reporting Data* attribute and its sub-attributes are only used in a few cases depending on the project team and their needs. Regarding *Requirement Ownership*, it is implicitly implied by other information. A requirements owner is the role[2] who

[2] A role may correspond to one or several persons.

owns an entire system component rather than an individual requirement. All the requirements allocated to that component are automatically owned by this role. Usually, a component gets allocated to one team.

E. Requirement Management Data

The organization manages most of the attributes in the *Requirements Management Data* cluster. By large, all the planning in the studied organization is done on a component and not on a requirement level. A component represents a group of related requirements or part of a system. Once one has allocated requirements to components, one starts planning their implementation using the attributes as designated in the *Requirements Management Data* cluster as listed in Figure 1.

F. Requirement Management Progress Data

The organization studied does not record the information as defined in the *Requirement Management Progress* cluster. It only tracks the status of the requirement implementation progress via the existing requirements management tool.

G. Requirement Completion Data

The organization records all the information in the *Requirement Completion Data* cluster. However, the information is recorded on a component level, not on an individual requirement level. For each component, they record the planned and actual completion dates, roles involved in approving and signing off the completion, and the total effort spent on the component implementation.

If they wish to track the completion data for individual requirements, then they have to do it manually. Doing it however does not belong to their ordinary procedure. Once they have analyzed the requirements and assigned them to the components, they just keep track of the status of the components.

H. Post Implementation Data

Regarding the information in the *Post Implementation Data* cluster, the organization studied does not record post-mortem analysis or lessons learned for individual requirements. The information may however be recorded for components.

The organization conducts post-mortem analysis on the project level once the project is completed. Part of this analysis involves tracking what areas of the requirements have changed, e.g. in comparison to other areas where the requirements were quite stable, and if they were successful in managing the scope. The analysis results provide an important feedback for improving the future requirements management for the same type of requirements.

They also identify lessons learned. The lessons learned are continuously considered especially in projects of iterative and agile nature.

All the information resulting from the analysis is recorded in a report that is kept in a common repository so that others can go back and read the lessons learned.

V. Concluding Remarks

In this paper, we have created a preliminary template, *Software Requirements Management Template (SRMT),* covering information about software requirements and their realization during a software development cycle. We then evaluated it within one Canadian company. This has helped us to evaluate our template and establish a state of practice within this company.

Our results show that all the attributes as suggested in our template are highly relevant both within heavyweight and lightweight software development. Many of them however, were not explicitly recorded. They might however be implicitly provided in other forms or in other documents or tools. This concerns attributes such as *Business Value, Requirements Ownership, Lessons Learned, Post-Mortem Analysis,* and other.

Some of the important attributes that have been suggested in many well-known standards and models were either not implemented or recognized by the organization studied. This concerns *Rationale, Customer Satisfaction* and *Customer Dissatisfaction. Fit Criteria, Assumptions,* and some of the attributes in the *Requirement Reporting Data* and the *Requirement Management Progress* clusters.

Despite the fact that these attributes are not used within the company studied, we do not modify our template. We motivate this with the following:

- *Rationale*: Many times, one needs to understand why a certain requirement needs be implemented. Hence, its *raison d'être* needs be provided. This helps the organizations understand the reason and intent behind the requirement and thereby assign a right priority value to it [10].
- *Customer Satisfaction* and *Customer Dissatisfaction:* This attribute indicates the degree of customer satisfaction/dissatisfaction if the requirement is/is not implemented. It indicates the customer priority, the value on which the development organization bases their own development priority value (see *Requirement Priority* in the *Requirement Evaluation Data* cluster). It also provides a basis for creating acceptance tests and evaluating the fulfillment of the requirements [11].
- *Fit Criteria*: Software developers must be provided with a set of criteria aiding them in assuring that they are building the right product. Hence, it is important to record fit (acceptance) criteria. Together with the *Customer Satisfaction* and *Customer Dissatisfaction* values, they constitute a basis for creating tests and evaluating the fulfillment of the requirements [11].
- *Assumptions:* In many large systems, the operational domain is unbounded. The software system, on the other hand, is finite [10]. Hence, there is a gap between the system and its operational domain. It must be bridged by assumptions. These assumptions help understand how one reasoned when developing the system.
- *Requirement Reporting Data*: This cluster contains attributes such as *Reporting Date* and *Originated By*. Both of them are very important for managing requirements. The reporting date indicates the age of the requirement and together with the requirement priority value, it constitutes an important basis for planning development. Regarding the *Originated By* attribute, it identifies the stakeholder who originated the requirement. Admittedly, the organization studied identifies the originator via use cases. However, due to its importance, we believe that this information should be

more visible. It (1) facilitates contact with the requirement originator to resolve any conflicting issues, (2) enables the delivery of the implementation to the right customer, and finally (3) it substantially increases customer satisfaction [4].

- *Requirement Management Progress:* The development process is usually divided into several phases. To enable an effective planning and monitoring of the development process, each development phase should be thoroughly identified and assigned a status value. This enables (1) determination of the development progress, (2) control of the amount of work that has been done and that remains to be done for a certain release, (3) control of the workload of each engineer/team (4) improved process discipline, (5) comparison of the planned and actual results, and other important controls [4], and other benefits.

VI. Epilogue

This study has been made within only one company. Still however, it provides a valuable feedback for preliminarily evaluating the usefulness of the *SRMT* template in the industry. It also provides a basis for further studies of the requirements management information. Hence, we cordially invite the software community to conduct similar studies in order to extend and evaluate our template.

ACKNOWLEDGEMENT

Many thanks to the anonymous company and the interviewee for contributing with valuable data to this study.

REFERENCES

[1] Atlantic Systems Guild, "Volare Requirements Specification Template". [Online] Available at: http://www.systemsguild.com/GuildSite/Robs/Template.html. Accessed in December 2007.

[2] IEEE, *IEEE Guide to Software Requirements Specifications (Std 830-1993)*. The Institute of Electrical and Electronics Engineers Inc., New York, NY, 1993.

[3] Higgins S.A et. al, "Managing Product Requirements for Medical IT Products". Proceedings of Joint International Conference on Requirements Engineering, 2002, pp 341-349.

[4] Kajko-Mattsson M, *Corrective Maintenance Maturity Model: Problem Management*. Doctoral Thesis, Stockholm University/Royal Institute of Technology, Sweden, 2001.

[5] Managing Requirements, "Templates and Guidance". [Online] Available at: http://www.jiludwig.com/Template_Guidance.html. Accessed in December 2007.

[6] Nyfjord and Kajko-Mattsson, "Degree of Agility in Pre-Implementation Process Phases". Technical report, Department of Computer and Systems Sciences, Stockholm University/KTH, Sweden.

[7] Robson C., *Real World Research*. Blackwell Publishing, 2002.

[8] Texas Department of Information Resources, Software Requirements Specification Template, DIR Document 25SR-T1-0. [Online] Available at: http://www.dir.state.tx.us/pubs/framework/gate2/sdlc/srs/25SR-T1-0.doc. Accessed in December 2007.

[9] Wiegers K. E, "Software Requirements Specification for Project". Available at: http://www.processimpact.com/process_assets/srs_template.doc. Accessed in December 2007.

[10] Lehmna M.M. and Ramil J.F., "Software Evolution in the Age of Component Based Software Engineering", IEEE Software, Vol. 147(6), 2000, pp. 249-255.

[11] Sommerville I., *Software Engineering*. 7th Ed. Addison Wesley, 2006.