POLITECNICO DI TORINO
Repository ISTITUZIONALE

A methodology for supporting Requirement Management Tools (RMt) design in the PLM scenario: an user-based strategy

Original
A methodology for supporting Requirement Management Tools (RMt) design in the PLM scenario: an user-based strategy / Vezzetti, Enrico; Violante, MARIA GRAZIA. - In: COMPUTERS IN INDUSTRY. - ISSN 0166-3615. - (2014). [10.1016/j.compind.2014.05.001]

Availability:
This version is available at: 11583/2546536 since:

Publisher:
ELSEVIER

Published
DOI:10.1016/j.compind.2014.05.001

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 October 2024

A methodology for supporting Requirement Management Tools (RMt)design in the PLM scenario: an user-based strategy

Abstract

In the current "mass customization" scenario, product complexity is increasing significantly due to the necessity to answer as quickly and effectively as possible to many different costumer needs but maintaining costs under control. In this scenario, requirements management becomes a fundamental features for the entire product lifecycle, as enterprises need to have a complete and clear idea of the market for succeeding in developing and supporting the right and innovative product. Moreover, considering that product lifecycle is characterized by many "trade-off", so that product features are often negotiated in order to fulfil to conflicting requirements, it is important to support the "traceability" of the entire lifecycle "negotiation" process. For this reason, PLM platform has to provide suitable methodologies and tools able to efficiently support the design and management of large set of complex requirements. Requirements Management Tools (RMt) embedded in PLM solutions help keeping specifications consistent, up-to-date, and accessible. At present, there are different possible solutions, but a shared PLM integrated seems not to be available. In order to fill this gap, this paper has developed an user-based strategy, based on Kano methodology, so on "user satisfaction", in order to define a structured set of guidelines to support the design of the features of an integrated PLM requirement management tool.

Keywords

Requirements management, Customer requirements, Product lifecycle Management, Kano, satisfaction.

1. Introduction

Innovation and new product development are essential for most companies to sustain future revenue growth. Customers demand more new products in shorter time intervals, often customized to their own needs. They want more attractive designs, better performance, better quality, lower prices, and instant availability. To meet these needs, companies have to be able to collaborate closely within their own organization and with partners and suppliers located in various parts of the world. At the same time companies have to manage increasing product and manufacturing complexities due to a quickly growing number of environmental and regulatory rules and requirements. Using a Product Lifecycle Management (PLM) strategy to manage product data, and to integrate and automate business processes generally results in efficiency improvements, which consequently enable companies to develop more new products, shorten time-to-market, reduce costs, increase productivity, and improve the quality of products and procedures. The PLM approach support the connection between single processes and the whole product lifecycle process management (Table 1).

Among these processes, the *Requirements Management* one enables users in requirements-driven product development environments to define, develop, capture, and incorporate product requirements of all kinds, customer, technical, regulatory, etc., in a single dynamic repository.

Today, as the product projects are getting more and more complex, the management of their requirements is getting more essential. In order to have right requirements, it is necessary to use the right tools and methodology. To remain competitive in current rapidly changing business climate, it is important to identify customer needs and transform them into design of customer-oriented RM application. So it is imperative that customers play an active part in the development process in order to deliver a high quality application. Conversely, poor understanding of customer needs and inaccurate assumptions made during the analysis of customer needs may adversely influence design and manufacturing of the products in terms of quality, lead time, and cost [1]. As the competition

for new markets increased, customer satisfaction also became a key factor for business success and a major concern of the companies. Satisfaction ratings are being used as an indicator of the performance of services and products and help to formulate strategies of the companies [1]. Hanan, M. and Karp, P. [2] have stated that "Customer satisfaction is the ultimate objective of every business: not to supply, not to sell, not to service, but to satisfy the customer needs that drive companies to do business." The objective of the product specification definition activity is to turn customer needs into a product specification [3]. Customer needs represent the "problems" the product must solve, e.g. functional features, durability, etc., while a product requirement specification is a formalized specification of customers' requirements, considering also performance and cost constraints. An efficient requirements management comprehends the tracking of evolving requirements over the entire product lifecycle and the identification of impacts from changes [4]. It is necessary to guarantee the agreement of the parties, recognize the deviations or changes in the project requirements, and renegotiate them. So the key point is gathering the right requirements and keeping them updated in order to accomplish the project and to have the right product.

Table1 - Processes usually automated with PLM systems.

Process	PLM systems
• Portfolio Management • Proposal Response	Sales and Marketing
• Early Sourcing • Component and Supplier Management	Sourcing
• Product Support Analysis and Planning • Technical Information Creation and Delivery • Performance Analysis and Feedback	Customer Service and Support
• New Product Development and Introduction (NPDI) • Program Management • Project Management • Requirements Management • Change Management (ECR/ECO)	Management
• Concept Development • System Design • Detailed Design • Configuration Management • Variant Design and Generation • Verification and Validation • Design Outsourcing	Engineering
• Quality and Reliability Management • Regulatory Compliance	Quality Assurance and Regulatory Affairs
• Manufacturing Process Management • Tooling Design and Manufacture • Manufacturing Outsourcing	Manufacturing

In order to reach this aim it is necessary to support the Requirement Management with the right methodologies and integrated PLM platform tools.

Focusing the attention on tools, it is possible to say that at present different solutions that help keeping specifications consistent, up-to-date, and accessible are available. These have been growing steadily in recent years. There are so many tools and correlated methodologies currently on the market that claim to support the requirements management process (or part of it). In the technical literature, a high number of evaluation frameworks exists to assist practitioners in

selecting requirements management tools. For example Hoffman et al. [5] present a role-oriented perspective on the required features of a requirements management tool and offer a comprehensive catalogue of requirements for requirements management tools based upon this work. Gotel and Mader [6] provide a mini-tutorial with an high-level guidance on designing a requirements management solution and selecting a requirements management tool. Hoffmann et al. [5] present a requirements catalogue to help: users to compare and select requirements management tools; tool providers to direct future tool developments. On the basis of specific industrial scenarios, the paper of Beuche et al. [7] derives important requirements that have to be observed if requirements management tools are to be usefully applied to product lines.

Most of these studies provide a list of features through which a number of leading commercial and open-source tools can be compared, but they are strictly linked with the specific vendor solution rather than with the real needs of the Requirement Management tasks and stakeholders (users or costumers who pay for the system; developers who design, construct, and maintain the system; users who interact with the system to get their work done).

So, in order to support the definition of a complete set of features that the RM tools have to provide for supporting efficiently the entire product lifecycle, this paper wants to propose an objective methodology, disjoined by the specific vendors, that focuses the attention on the real needs of RM process. For reaching this aim it is necessary to involve in the study the RM stakeholders and capturing their point of view as "users". Each user has a unique view on the "system", so users must be involved in the tool analysis and design. So users' activities have to be linked, their needs and perspectives have to be considered from earlier design stages, and their achieved degree of satisfaction has to be tracked [8]. The level of satisfaction is ultimately dependent on the fulfilment of user needs. The quality of a product or service is a key element in creating user satisfaction. If we do not demand that the requirements meet certain quality criteria, then it will be more difficult to search for quality in later development phases [9]. Since the impact on user satisfaction is different for each user requirement, it is important to determine which attributes of a product or service bring more satisfaction than others.

For that reason a Kano model has been employed for designing our strategy, for capturing the RM users voice and for identifying which attributes drive them to real satisfaction. The paper is organized as follows: in section 2 and 3, we present a brief introduction to Requirement Management and Kano model; in section 4, we explain our methodology and in section 5 we present the obtained results.

2. Requirement Management

Schwaber, C. and P. Sterpe [10] provide a definition of requirements management: "*The storage of requirements, the tracking of relationships among requirements, and the control of changes to individual requirements and groups of requirements*". More in detail, requirement management could be stated with four different tasks. The first one is requirements elicitation. This is the process through which the product developers(designers) discover, review, articulate and understand the users' (potential customers) needs and project constrains on development activities [11]. This phase is dedicated to convert tacit and subjective needs into explicit statement. Usually users are not good at describing what they need; this is because in the first step users' needs must be caught and organized to prevent ambiguity or misunderstanding from arising before significant work is done. Since there are many kind (lead users, professional,…) involved, the ability of the analyst is to understand each need and to resolve the conflicts that may arise. The second task is the requirements analysis that supports the developers to arrive to a definition of users' needs in order to define product features. This phase requires analyzing needs and understanding preferences; it involves a classification and prioritization of requirements, as these are not equally important. The analysis involves the refinement of requirements through many stages in order to achieve a more detailed description of requirements. The third task is requirements specification that consists in the

creation of a structurally concrete and precise specification of product requirements based on functional knowledge that has been elicited from users [12]. The last one is requirements verification and validation that checks the quality of the requirements and whether requirements specification reflects users' needs. Validation is an evaluation process to determine if it meets requirements or not; instead verification is defined as a process used to establish if product reflects specification requirements [12].

For supporting an efficient Requirements Management process, firms need to have specific tools. These can assist organizations in defining and documenting requirements by allowing them to store requirements in a central location. Project teams can then access the requirements to determine what is to be developed, and customers can access the requirements to ensure that their needs were correctly specified. To understand the working of a RM tool, CPDA cites the example of an Aerospace program [27]:*"it starts with several thousand customer requirements at the system level, and several hundred thousand at the lowest level defining components, with an equivalent number of verification and test items. In a perfect world, the cascading requirements and related information will be defined once, and relied on for the duration of the project. In reality, requirements are always on the move. Users often change their mind, or develop a better understanding of the targeted needs after looking at the initial design; market drivers change over the duration of the development project; authorities keep adding new constraints related to environmental or safety concerns; and sometimes the project encounters difficulties that require a revision of the initial targets. All those issues necessitate that the requirements management tool has the ability to deal with change management, change impact analysis, revisions, options, baselines and effectivities"*.

3. Introduction to Kano model

The Kano model of customer satisfaction is a useful tool to categorize product attributes based on how they are perceived by the customers and their effect on customer satisfaction [13, 14]. By meeting the user basic quality needs, it provides the foundation for the elimination of dissatisfaction and complaints. By exceeding expectation it creates a competitive advantage and leads to innovation [25]. By the use of the theory of attractive quality for the analysis of improvement opportunities in products and services[14] and the model of excitement and basic quality [1,15], it is possible to take into consideration the asymmetrical and non-linear relationship between performance and satisfaction (Fig.1).

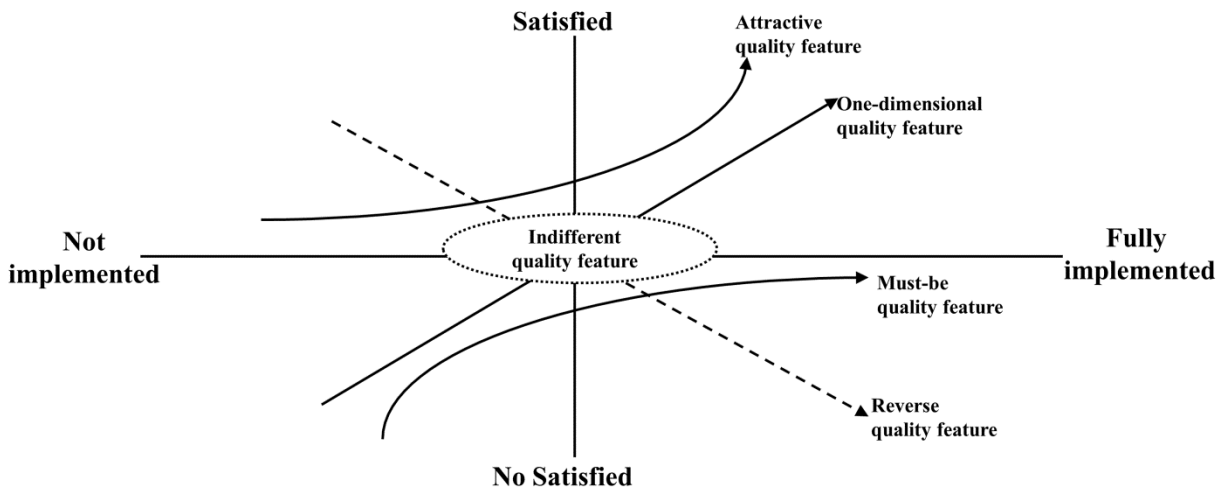

Figure 1 - Kano model of user satisfaction.

For this reason in the Kano model, positive (functional) and negative (dysfunctional) questionnaires are conducted to collect the satisfaction difference per item from the interviewees, and to judge the specific quality of each item represented according to the "Kano evaluation form" (Table 2) [14].

Table 2 - Kano evaluation form.

		Dysfunctional				
Functional		*I like it*	*I expect it*	*I'm neutral*	*I can tolerate it*	*I dislike it*
	I like it	Questionable	Attractive	Attractive	Attractive	One-dimensional
	I expect it	Reverse	Indifferent	Indifferent	Indifferent	Must-be
	I'm neutral	Reverse	Indifferent	Indifferent	Indifferent	Must-be
	I can tolerate it	Reverse	Indifferent	Indifferent	Indifferent	Must-be
	I dislike it	Reverse	Reverse	Reverse	Reverse	Questionable

Table 3 shows Kano quality categories. According to them, it is possible to classify an item (or attribute, feature,...) of the analysed product/service.

Table 3 – Quality category.

Quality category	**Description**
Attractive	Attractive quality features are the product criteria which have the greatest influence on how satisfied a user will be with a given product. Attractive features are neither explicitly expressed nor expected by the user. Fulfilling these features leads to more than proportional satisfaction. If they are not met, however, there is no feeling of dissatisfaction.
One-dimensional	These one-dimensional quality features are positively and linearly related to user satisfaction. The user satisfaction is proportional to the level of fulfilment of these features: the higher the level of fulfilment, the higher the user's satisfaction, and vice versa. One-dimensional features are usually explicitly demanded by the user.
Must-be	This is the basic criteria of a product/service. If these features are not fulfilled, the user will be extremely dissatisfied. On the other hand, as the user takes these features for granted, their fulfilment will not increase his satisfaction. The user regards the must-be features as prerequisites; he takes them for granted and therefore does not explicitly demand them. If they are not fulfilled, the user will not be interested in the product, service or process at all.
Indifferent	An attribute whose presence or absence does not cause any user satisfaction or dissatisfaction.
Reverse	An attribute whose presence causes user dissatisfaction, and whose absence results in user satisfaction.

From an operative point of view, in order to collect all the necessary information for the analysis, it has been necessary to employ a questionnaire. The questionnaire design has been implemented by the formulation of a pair of questions for each product feature for which you desire user feedback. The first question in each pair of questions for a product feature refers to a situation in which the feature is met, and it is worded in a format similar to the following: "If [the product] satisfied [feature x], how would you feel?" This is the *functional* question. The second question in

each pair refers to the case where the feature is not met. This is called the *dysfunctional* question and it is worded in a format similar to the following: “If [the product] did not satisfy [feature x], how do you feel?” For each question, the user can answer in one out of five different ways: (1) I like it; (2) I expect it; (3) I am neutral; (4) I can tolerate it; (5) I dislike it. By combining the two answers in the Kano evaluation table, the items of the product/service can be classified into one of six quality categories: attractive quality (A), one-dimensional quality (O), must-be quality (M), indifferent quality (I), reverse quality (R), or questionable result (Q). If the user answers, for example, “I like it” as regards “If [the product] satisfied [feature x], how would you feel?” in the functional form of the question, and answers “I dislike it” as regards “If [the product] did not satisfy [feature x], how do you feel?” in the dysfunctional form of the question, the combination of the questions in the Kano evaluation table finds an “O”, indicating that “feature x” is a one-dimensional quality item from the perspective of users (Fig. 2). If combining the answers yields category I, this means that the user is indifferent to this “feature x”. The user does not care whether it is present or not. The user is, moreover, not willing to spend more on this feature. Category Q stands for a questionable result. Normally, the answers do not fall into this category. Questionable scores signify that the question was phrased incorrectly, or that the person interviewed misunderstood the question or crossed out a wrong answer by mistake. If looking up the answer in the evaluation table yields category R, this “feature x” is not only not wanted by the user but he/she even expects the reverse.

		Dysfunctional question: **“**If [the product] did not satisfy [feature x], how do you feel?”				
Functional question: If [the product] satisfied [feature x], how would you feel?”		*I like it*	*I expect it*	*I’m neutral*	*I can tolerate it*	***I dislike it***
	I like it					**One-dimensional**
	I expect it	Reverse	Indifferent	Indifferent	Indifferent	Must-be
	I’m neutral	Reverse	Indifferent	Indifferent	Indifferent	Must-be
	I can tolerate it	Reverse	Indifferent	Indifferent	Indifferent	Must-be
	I dislike it	Reverse	Reverse	Reverse	Reverse	Questionable

Figure 2 – Functionality of Kano evaluation table

In order to classify each feature in the Kano quality category, it is necessary to process the results on the basis of the highest response frequency obtained from the questionnaire. But if two or more Kano quality category had obtained the same frequency value, the authors have considered the following order M>O>A>I (that is the classification with the greatest impact on the product) to decide the quality category.

In order to find out which feature can influence user satisfaction, the user satisfaction (CS) index is calculated [15]. The CS index is indicative of how strongly a product feature may influence satisfaction or, in case of “non-fulfilment”, user dissatisfaction. The formulas to calculate the extents of satisfaction and of dissatisfaction are the following:

extent of satisfaction: $$\frac{A+O}{A+O+M+I};$$

extent of dissatisfaction: $$\frac{O+M}{(A+O+M+I)\cdot(-1)}.$$

The extent of satisfaction (or positive CS index or Better value) ranges from 0 to 1. If the value is close to 1, it means that the feature has a positive effect on increasing user satisfaction; when the extent of dissatisfaction (or negative CS index or Worse value) is close to - 1, it means that the feature can decrease user satisfaction.

4. Methodology

Our proposed methodology based on the Kano model follows two steps. The first step is the identification of the RM tool features and the last one is the development and administration of the "Kano-questionnaire".

Step1: Identification of the RM tool features

Starting from the International Council on Systems Engineering (INCOSE) survey of requirements modelling tools, it has been possible to have a first idea about how software vendors assess their product, working on 80 questions ranging from software support to modelling capabilities [24]. Together with the INCOSE data, Vrinat M.'s survey has been taken into account too [26]. He conducted an analysis over eight requirements management tools and evaluated them against one-hundred criteria. In order to complete the questionnaire design, the authors have considered an evaluation work based on demonstrations and discussion with tools vendors [11], a critical evaluation of existing tools conducted by James L. [12] and Grady, J. [16], a user cases scenarios analysis evaluated by Gabb A. P. el al. [17] and material coming from a work on tool selection criteria conducted by Daimler Chrysler [23] and Jones, D.A., et al. [18], Schwaber, C. and P. Sterpe, [10], and Regnell, B., et al. [19].

By the use of a cluster analysis approach [20], [21] on the information obtained by the previous surveys and evaluations, it has been possible to provide a first features set of RM tools (Table 4). In addition, starting from the four phases (elicitation, analysis, specification, verification and validation) in which RM is structured within a company, we have identified which features of Table 4 best support these tasks.

Requirement elicitation involves seeking, uncovering, acquiring and elaborating requirements. So storing all source documents, notes, and observations from stakeholders interviews or workshops in a database, and building a tracking list of those documents permit a requirement identification and capture [26]. In this phase, the documents and requirements can be organized in tree structures or views, as well as the identification of stakeholders with their roles and responsibilities can be saved, and that data can be managed. The requirements can be imported from multiple sources and formats – text in a Word file, parameters in an Excel spread sheet, etc. In addition it could be necessary to import of requirements from another project for re-use. So features as, for example, ***R3*** *– Requirement identification & Capture,* ***R4*** *– Capturing system element structure,* ***R9*** *– Linking and tracing,* ***R10*** *– Requirement hierarchies,* ***R12*** *– Storing,* ***R14*** *– Reusability of Requirements,* ***R15*** *– Workflow capabilities,* ***R18*** *– Requirements Definition features,* ***R21*** *– Integration with other life-cycle tools,* ***R25*** *– Ms Word Support* could support this phase.

With regard to "Requirement analysis", it involves the refinement of requirement through the decomposition of high level descriptions into more details which may entail building models, evaluating feasibility, analysing overlaps or conflicts between requirements, and negotiating

priorities. Refining or deriving requirements through the flow-down and flow-across processes must be facilitated with the automatic linking of related requirements and the inheritance of attributes [26]. So this phase may necessitate features such as ***R3*** *– Requirement identification & Capture,* ***R4*** *– Capturing system element structure,* ***R5*** *– Handle a large set of documents,* ***R6*** *– Being able to support complex set of document,* ***R8*** *– Extensibility,* ***R9*** *– Linking and tracing,* ***R11*** *– Identify inconsistence,* ***R12*** *– Storing,* ***R13*** *– Comparing,* ***R16*** *– Reporting and Analysis.* The feature ***R18*** *– Requirements definition features* supports the "Requirement Specification" phase. Capabilities in support of "verification and validation" phase are: *(a)* consistency checks for isolated items such as a requirement or a function, for any lack of attributes, for un-allocated requirements, and for un-justified functions; *(b)* links between requirements and their associated documents covering tests or specifications; *(c)* creation and storage of customizable views of requirements; *(d)* access for large numbers of stakeholders and users for reviewing and commenting on requirements; *(e)* access rights for users and groups for each requirement, including full access, read only, and read-write; *(f)* history of all changes for each requirement; *(g)* storage and management of the review and inspection of the results, including information on how it was done and who was responsible; *(h)* traceability of verification and validation cases to requirements; *(i)* traceability of the validation cases to validation procedures; *(j)* and storage and management of verification and validation plans and procedures [26]. So for this phase, it is important to provide some features such as ***R4*** *– Customizable,* ***R4*** *– Linking and tracing,* ***R11*** *– Identify inconsistence,* ***R12*** *– Storing,* ***R17*** *– Change Management,* ***R20*** *– Requirement Validation Capabilities,* ***R22*** *– Security capabilities,* ***R23*** *– Integration with Web,* ***R24*** *– Collaborative Working.*

Other features such as ***R1****- Easy to use & minimal training* and ***R2*** *– Simple Framework* cover aspects related to user interface of RM tools. In fact an easy-to-use interface must serve a broad range of occasional users who are not experts with the specific software to achieve the full payoff of requirements management. A simple user interface represents a key aspect for the broad acceptance needed for a successful implementation.

Table 4 - Features of RM tools and their description.

Feature of RM tool	Feature Code	Description
Easy to use & minimal training	**R1**	The RM tool is user friendly when it is easy to use with minimal training.
Simple framework	**R2**	A User interface that enables the inexperienced user to interact successfully with the tool.
Requirement identification & Capture	**R3**	A clear definition of requirements is necessary as the requirements drive cost, schedule, skills required, resources required, verification plans and schedules and operational procedures. The consequence of poor requirements is that the user does not get what they want, when they want it and for the price that is competitive. ▪ Input document enrichment/analysis: using existing document information (such as glossary, index, etc.), aid the user in requirements analysis, identification of requirements, etc. ▪ Input document change/comparison analysis: the ability to compare/contrast two different versions of a source document. ▪ Automatic parsing of requirements: a mechanism for automatic identification of requirements by key words, structure, unique identifiers, etc. to create

		requirements from the text. ▪ Interactive/semi-automatic requirement identification: the ability to identify requirements from a text file via interactive means such as mouse highlighting of the requirement text or prompting by the system "is this a requirement?" ▪ Manual requirement identification: a manual means of identifying or creating requirements. ▪ Batch mode operation: a mechanism for inputting/identifying requirements from outside of the tool. ▪ Batch-mode document/source-link update: does the tool have the ability to update existing linked documents from new/changed versions of the source documents without having to re-establish traceability links. ▪ Requirement classification: does the tool have the ability to classify/categorize requirements during identification.
Capturing system element structure	**R4**	Once the requirements have been captured, the allocation of requirements to sub-system elements takes place. The tool must capture these elements so links/allocations can be made to those sub-systems elements ▪ Graphically capture systems structure: Can the tool graphically capture system implementation (such as architecture, functional decomposition, WBS - Work Breakdown Structure, etc.) and display them graphically such that requirements can be linked to them. ▪ Textural capture of systems structure: Can the tool textually capture system implementation (such as architecture, functional decomposition, WBS, etc.) and display them textually such that requirements can be linked to them.
Handle a large set of documents	**R5**	The tool must be able to manage large of requirements. Increased numbers of users, end users, developers, subcontractors, product features, external system interfaces, etc. come along with increased number of requirements generated in the Requirements Engineering - RE process
Being able to support complex set of documents	**R6**	The tool must be able to complex sets of requirements.
Customizable	**R7**	The users must be able to customize the standard views without changing the template. The user interface of the tool also must be customizable with a standard script language.
Extensibility	**R8**	The RM Tool must be adaptable and extensible to the needs of the organization or project. The tool must provide an open and well-documented object model and an API which makes all data and functions accessible to

		extensions. The user interface of the tool must be extensible with a standard script language.
Linking and tracing	**R9**	The tool must maintain traceability, as the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origin, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases). Tracing individual requirements to other system components helps ensure that your team does not inadvertently overlook any requirements during implementation. You can define links between different kinds of requirements and between requirements in different subsystems. When analyzing the impact of a change proposed in a specific requirement, the traceability links reveal the other system elements that the change might affect.
Requirement hierarchies	**R10**	Without the capability of identifying and managing dependencies among requirements, keeping track of elaborate requirements hierarchies that include both parent/child relationships and arbitrary relationships can be downright gruelling.
Identify inconsistence	**R11**	The RM tool should allow the user to identify inconsistencies such as unlinked requirements or system elements (orphans).
Storing	**R12**	The RM tool must support baselines. A baseline is the state of a (specified subset of the) requirements database fixed at a given point in time. Baselining requirements is like taking a snapshot of their state at a point in time both individually and in aggregate and then applying a label to it. Requirements management tools should store baselines in a secure repository accompanied by information about the act of creating the baseline potentially in the form of electronic signatures.
Comparing	**R13**	Because individual requirements and the collection of requirements that correspond to a development effort will change over time, requirements management tools need to include baselining capabilities to determine the differences between various baselines.
Reusability of Requirements	**R14**	The reuse of requirements across projects permits to build up a repository of all of the requirements that have ever been fulfilled — to make it easy to search across all of these requirements — and thus to raise awareness of redundancy in software assets. So enterprise-level requirements can be linked to project-level requirements. Copying and pasting requirements from project to project, parent/child relationships with inheritance among enterprise- and project-level requirements are true support for reuse of requirements.

Workflow Capabilities	**R15**	To automate the processes that surround requirements change, requirements management tools should include workflow capabilities, including graphical utilities for workflow design and form design and extensibility to support initiation of arbitrary internal and external events. Workflows can help to implement a certain RE process and can improve consistency and standardization of the requirements.
Reporting and Analysis	**R16**	The tool must generate freely configurable change reports. These reports should relate to views, baselines and generated documents. The tool could analyze changes to provide information about the project status. The tool should be able to analyze requirements. Examples are linguistic analysis, analysis of the link structure, analysis of project progress and risk management.
Change Management	**R17**	The tool must offer the possibility to handle formal change requests. This function must be customizable to the change process of the users. Especially in the late phases of a project, a restrictive change management is important. The straightforward changes in earlier project phases must be possible as well.
Requirements Definition Features	**R18**	To help elicit and specify requirements. While many requirements management tools offer support for inputting requirements through bulk import or even customized forms, they do not help to elicit or to specify requirements.
Decision Support Capabilities	**R19**	To facilitate prioritization and selection of requirements. Once requirements have been defined, the task of determining which will be fulfilled and in what order remains.
Requirements Validation Capabilities	**R20**	To assess the quality of requirements. Many IT organizations expect their requirements management tools to measure or even to improve the quality of their requirements. But requirements management tools are agnostic about requirements contents: They store, associate, and version requirements without any respect to their quality.
Integration with other life-cycle tools	**R21**	Requirements management tools manage relationships among requirements, for example between business requirements, functional requirements, and technical specifications. Integrations between requirements management tools and tools for activities like software configuration management, build management, and test management are required for this chain of traceability to extend through the lifecycle without significant manual effort.
Security capabilities	**R22**	Specification of who has authorised access to the system and under what circumstances that access is granted. It is important to set access permissions for users. Web access lets you share requirements information with all team members, even if they are geographically separated

Integration with Web	**R23**	The tool should have a web interface or another browser-based client that makes it unnecessary to install a client application for occasional users. Web interfaces offer a reliable and easily manageable possibility to work with the requirements. They are interesting for collaboration with external partners ("extranet") and for internal users that use the tool only occasionally.
Collaborative Working	**R24**	Many users should be able to work on the same data at the same time. Of the many users working on an object, only one must be able to apply changes. If a user changes an object, it should refresh automatically in the user interfaces of the other users.
Ms Word Support	**R25**	The RM Tool should support Ms Word as an input mechanism, whether that's through utilities that parse Word documents to import requirements and associated metadata, plug-ins that expose requirements management toolbars within Word, or a Word-like interface in the requirements management tool.

Step 2: Design and administering of the "Kano-questionnaire"

To design the questionnaire, it is necessary to formulate a pair of questions for each identified RM tool feature for which a customer feedback is desired. The appendix A shows the constructed questionnaire. Then, the Kano questionnaire was distributed to RM tools users in the field of automotive, aircraft and defense, who have experience in these tools indispensable to gather, analyze, document and manage requirements of "complex systems/products". The Kano questionnaire was administered via e-mail with a letter of introduction, explaining the questionnaire purpose, sent together with some operative instructions.

5.Results and Discussion

The results obtained analyzing the survey results, developed on thirty feedback, have been quantified in table 5.

Table 5 –Results.

Feature of RM tool	Feature Code	A	O	M	I	Result
Easy to use & minimal training	**R1**	20%	30%	7%	43%	I
Simple framework	**R2**	17%	23%	27%	33%	I
Requirement identification & Capture	**R3**	0%	30%	70%	0%	M
Capturing system element structure	**R4**	43%	23%	23%	10%	A
Handle a large set of documents	**R5**	77%	23%	0%	0%	A
Being able to support complex set of documents	**R6**	77%	13%	10%	0%	A

Customizable	**R7**	10%	47%	40%	3%	O
Extensibility	**R8**	10%	53%	33%	3%	O
Linking and tracing	**R9**	0%	23%	77%	0%	M
Requirement hierarchies	**R10**	0%	33%	80%	0%	M
Identify inconsistence	**R11**	30%	43%	27%	0%	O
Storing	**R12**	13%	40%	47%	0%	M
Comparing	**R13**	10%	57%	33%	0%	O
Reusability of Requirements	**R14**	0%	40%	60%	0%	M
Workflow Capabilities	**R15**	37%	30%	30%	3%	A
Reporting and Analysis	**R16**	10%	47%	30%	13%	O
Change Management	**R17**	3%	53%	43%	0%	O
Requirements Definition Features	**R18**	7%	40%	57%	0%	M
Decision Support Capabilities	**R19**	13%	13%	13%	57%	I
Requirements Validation Capabilities	**R20**	0%	33%	67%	0%	M
Integration with other life-cycle tools	**R21**	77%	23%	0%	0%	A
Security capabilities	**R22**	67%	17%	17%	0%	A
Integration with Web	**R23**	23%	10%	13%	53%	I
Collaborative Working	**R24**	43%	23%	33%	0%	A
Ms Word Support	**R25**	0%	23%	23%	53%	I

The features of RM Tools have the percentage of fulfilment showed in Figure 3 and classified in the following way:

Dissatisfiers or Basic Needs or Must-be needs are expected features of a product or service (legible forms, correctly spelled name, basic functionality). These needs are typically "unspoken." If these needs are not fulfilled, the user will be extremely dissatisfied. For RM tools, examples of "unspoken" needs are: ***R3****-Requirement identification & Capture,* ***R9****-Linking and tracing,* ***R10****-Requirement hierarchies,* ***R12****-Storing,* ***R14****-Reusability of requirements,* ***R18****-Requirements definition features,* ***R20*** *- Requirements validation capabilities*.

Satisfiers or Performance Needs or One-dimensional needs are Standard features that increase or decrease satisfaction by their degree (cost/price, ease of use, speed). These needs are typically "spoken." For RM tools, one-dimensional features are: ***R7*** *- Customizable,* ***R8****–Extensibility,* ***R11*** *- Identify inconsistence,* ***R13*** *- Comparing,* ***R16*** *- Reporting and analysis,****R17*** *- Change management.*

Delighters or Excitement Needs or Attractive needs are unexpected features that impress users and earn the company "extra credit." These needs also are typically "unspoken." For RM tools, Attractive Features are: ***R4****-Capturing system element structure,* ***R5****-Handle a large set of documents,* ***R6****-Being able to support complex set of documents,* ***R15****-Workflow capabilities,* ***R21****-Integration with other life-cycle tools,* ***R22*** *- Security capabilities,* ***R24****-Collaborative working.*

Indifferent Features in RM tools are: ***R1****-Easy to use &minimal training,* ***R2*** *- Simple framework,* ***R19****-Decision support capabilities,* ***R23*** *– Integration with web,* ***R25****-Ms Word support*. In this case, user is indifferent to whether the feature is present or not.

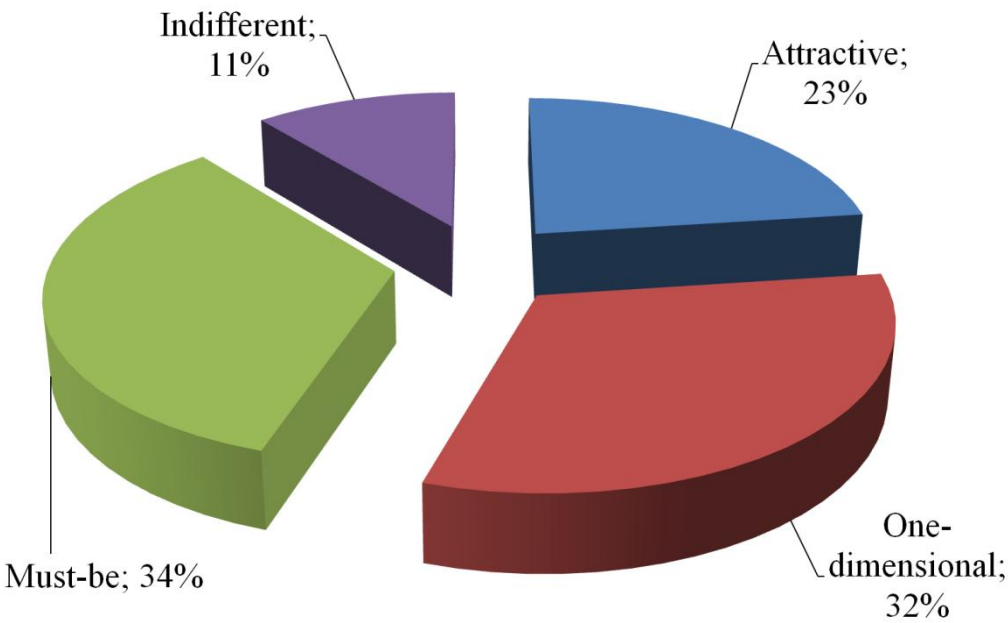

Figure 3 –RM tools features fulfilment percentage.

In table 6 and figure 4, how each feature can influence user satisfaction is described. The extent of satisfaction (or positive CS index or Better value) ranges from 0 to 1; the closer the value is to 1, the higher the influence on user satisfaction. In table 5 these values are highlighted. A positive CS index which approaches 0 signifies that there is very little influence. At the same time, however, one must also take the negative CS coefficient into consideration. If it approaches -1, the influence on user dissatisfaction is especially strong if the analysed feature is not fulfilled (in Table 6) the values closer to -1 are highlighted). A minus sign is put in front of the CS index of user dissatisfaction in order to emphasize its negative influence on user satisfaction if this feature is not fulfilled.

Table6- User satisfaction coefficient.

Feature Code	Feature Name	Better Value	Worse Value
R1	Easy to use & minimal training	0,50	-0,37

R2	Simple framework	0,40	-0,50
R3	Requirement identification & Capture	0,30	**-1,00**
R4	Capturing system element structure	0,67	-0,47
R5	Handle a large set of documents	**1,00**	-0,23
R6	Being able to support complex set of documents	**0,90**	-0,23
R7	Customizable	**0,57**	**-0,87**
R8	Extensibility	**0,63**	**-0,87**
R9	Linking and tracing	0,23	**-1,00**
R10	Requirement hierarchies	0,29	**-1,00**
R11	Identify inconsistence	**0,73**	**-0,70**
R12	Storing	**0,53**	**-0,87**
R13	Comparing	**0,67**	**-0,90**
R14	Reusability of requirements	0,40	**-1,00**
R15	Workflow capabilities	**0,67**	**-0,60**
R16	Reporting and analysis	**0,57**	**-0,77**
R17	Change management	**0,57**	**-0,97**
R18	Requirements definition features	0,45	-0,94
R19	Decision support capabilities	0,28	-0,28
R20	Requirements validation capabilities	0,33	**-1,00**
R21	Integration with other life-cycle tools	**1,00**	-0,23
R22	Security capabilities	**0,83**	-0,33
R23	Integration with web	0,33	-0,23
R24	Collaborative working	**0,67**	**-0,57**
R25	Ms Word support	0,23	-0,47

Notes: the highlighted better values and worse values cause user satisfaction or dissatisfaction

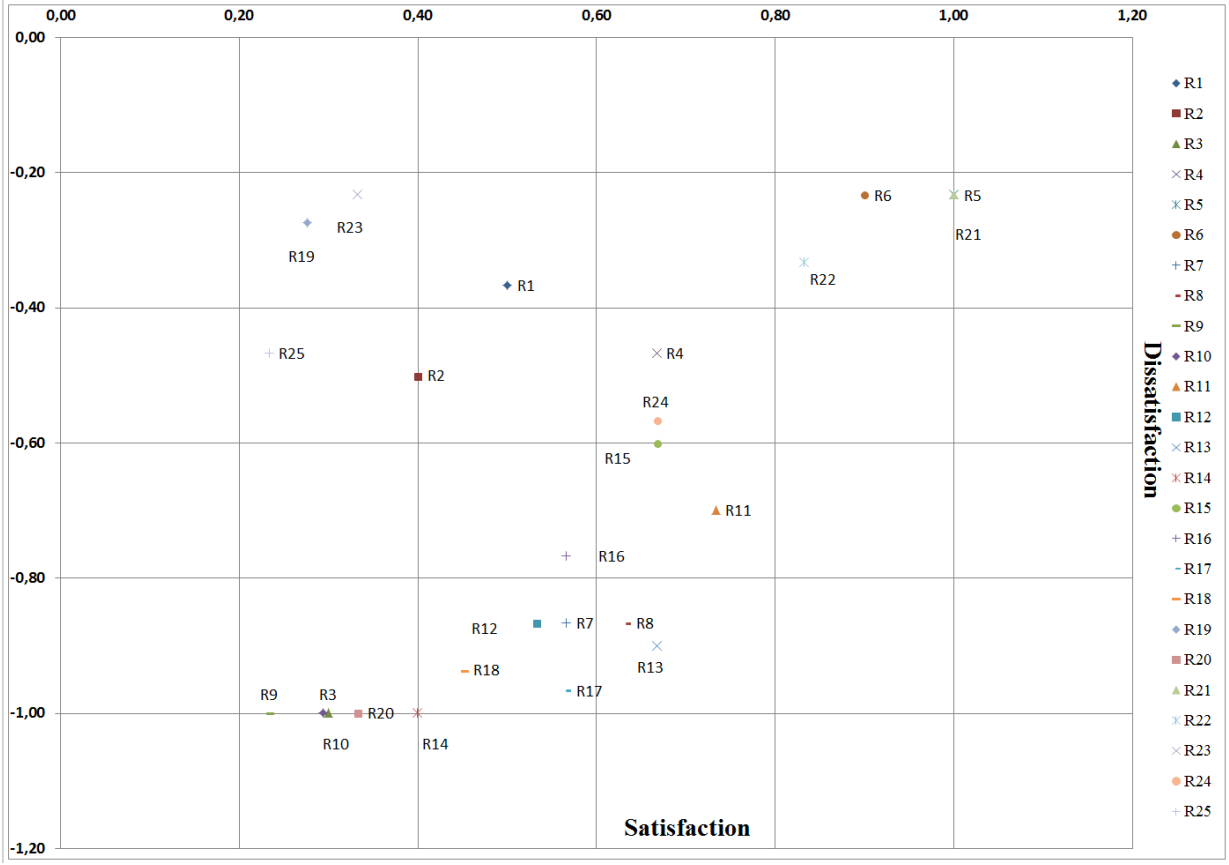

Figure 4 -Influence of product features on satisfaction or dissatisfaction.

This study has computed the user satisfaction (CS) index that states whether satisfaction can be increased by meeting a product feature or whether fulfilling this product feature merely prevents the user from being dissatisfied. For instance (see highlighted "Better" values of Table 6), the features of RM Tools which have the higher influence on user satisfaction because its CS coefficient is closer or equal to 1 are: ***R5*** *- Handle a large set of documents,* ***R6*** *- Being able to support complex set of documents,* ***R21*** *- Integration with other life-cycle tools;* ***R22*** *- Security capabilities.*

If improvements in the RM tools are provided in terms of ***R3****–Requirement identification & Capture,* ***R9*** *- Linking and tracing,* ***R10*** *- Requirement hierarchies,* ***R14*** *– Reusability of requirements,* ***R20*** *- Requirements validation capabilities* (which are the highlighted "Worse" values of Table 6), they can decrease user dissatisfaction markedly.

While improvements in the features of RM Tools such as ***R7****– Customizable,* ***R8****– Extensibility,* ***R11*** *- Identify inconsistence,* ***R12*** *– Storing,* ***R13*** *– Comparing,* ***R15*** *- Workflow capabilities,* ***R16*** *- Reporting and analysis,* ***R17*** *- Change management,* ***R24*** *- Collaborative working* (which are the highlighted "Worse" and "Better" values of Table 6) not only can increase users' satisfaction but also can decrease users' dissatisfaction.

Kano model and the Better/Worse value can help stakeholders to prioritize the features of RM tools. So it is possible to categorize user needs and provide appropriate actions or investments in user satisfaction improvement in the context of RM products. RM tools stakeholders must fulfil all must-be or high Worse value elements to prevent user dissatisfaction, improve on one-dimensional and high Better value of quality elements and provide attractive quality elements.

CONCLUSION

Customer satisfaction is related to the fulfilment of customer needs. The fulfilment of those needs depends on the existence and performance of certain customer requirements in the product or service. Hence, there is a need to study and develop procedures that can help a company or project team to gain knowledge of customer requirements and satisfaction, and then develop products with innovative features. The Kano model provides an effective approach to categorize user requirements and to understand their nature. It is important to highlight that Kano model does not include a cost dimension but in our study this factor is not relevant. Most of the traditional measures used in the decision making process tend to be financial measures and business decisions and they are often taken in an attempt to maximise or reduce the impact of these financial indicators [22]. Our proposed methodology ranks features only according to their value to customers and users and not according to their estimated cost of implementation. By including these customer-oriented performance measures in the re-engineered process, the new business process will then have a customer orientation.

The proposed characterization offers a *foundation* to allow stakeholders (including customers, users, developers, researchers) to identify users' needs and documenting these in a form that is amenable to subsequent implementation.

Drawing from the literature in the areas of Requirements Management tools, many works have gave good overview of the main tools features and then have published evaluations and benchmarking reports on RM tools present on the market at that time. In none of these works, the voice of the RM stakeholders is considered, so in order to fill this gap this paper presents an evaluation methodology of RM tools features. The authors have identified a core set of features that a tool for requirements management must possess taking into account the previous catalogues, surveys, researches conducted in this context. Then a Kano model has been designed to categorize the RM tools attributes relying on how they are perceived by the user and their effect on user satisfaction.

The results of this methodology are helpful not only for RM tool developers but also for enterprises that are looking for such a tool for their core business. For SW developers, it is important to have the most clear picture on RM scenario needs in order to provide the appropriate correction action on their solution for increasing their market quote. To deliver a high-quality application, it is imperative that user play an active part in the development process. The described methodology can be an effective means of communication essential to get the tool right the first time and avoid expensive re-work later in the development cycle. The analysis result is also useful for enterprises because they support the management in identifying which could be the real advantage of introducing in their organisation an integrated RM tool.

The analysis results reflect the influence of the automotive, aircraft, defense background, as the considered voice of customer comes from these sectors. However, the authors think that these outcomes could be useful to support the improvement of RM tools for other domains. In fact, according to Schwaber, C. and P. Sterpe [10], as business conditions change, requirements management tools are essential to understand the impact of requirements change and enforce the processes that surrounds it not only for the automotive, aircraft, defense sectors (where the analysis was conducted) but also for other industrial contexts. So, considering that our study was conducted using responses from segments portrayed by high complexity products, many components, different disciplines, and configurations, in which requirement management is a key element for the product development success, this result could be employed also for other segments as naval, train,… where the same product complexity exists.

The results obtained show that if features such as ***R5***– *Handle a large set of documents*, ***R6*** - *Being able to support complex set of documents*, ***R21*** – *Integration with other life-cycle tools*, ***R22***- *Security capabilities* are included in enhanced RM tools, the user satisfaction increases. Other features such as ***R3*** – *Requirement identification& Capture*, ***R9*** - *Linking and tracing*, ***R10*** -

Requirement hierarchies, ***R14*** *– Reusability of requirements*, ***R20*** *- Requirements validation capabilities* decrease the user dissatisfaction. These results allow to know how to rank user needs; this can help developers plan releases by indicating which functions are critical and which can be added (and in what order) over successive releases. Unambiguous knowledge about requirement priorities help stakeholders to more effectively and efficiently manage projects and allocate resources based on the requirement importance to the project as a whole.

REFERENCES

1. Matzler, K. and H.H. Hinterhuber, *How to make product development projects more successful by integrating Kano's model of customer satisfaction into quality function deployment.* Technovation, 1998. **18**(1): p. 25-38.
2. Hanan, M. and P. Karp, *Customer Satisfaction: How to Maximize, Measure, and Market Your Company's" ultimate Product"*. 1989: American Management Association.
3. Nilsson, P. and B. Fagerström, *Managing stakeholder requirements in a product modelling system.* Computers in Industry, 2006. **57**(2): p. 167-177.
4. Schuh, G., et al., *Process oriented framework to support PLM implementation.* Computers in Industry, 2008. **59**(2): p. 210-218.
5. Hoffmann, M., et al. *Requirements for requirements management tools*. in *Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International*. 2004. IEEE.
6. Gotel, O. and P. Mader. *How to select a requirements management tool: Initial steps*. in *Requirements Engineering Conference, 2009. RE'09. 17th IEEE International*. 2009. IEEE.
7. Beuche, D., et al. *Using requirements management tools in software product line engineering: the state of the practice*. in *Software Product Line Conference, 2007. SPLC 2007. 11th International*. 2007. IEEE.
8. Ballejos, L.C. and J.M. Montagna, *Modeling stakeholders for information systems design processes.* Requirements Engineering, 2011. **16**(4): p. 281-296.
9. Génova, G., et al., *A framework to measure and improve the quality of textual requirements.* Requirements Engineering, 2013. **18**(1): p. 25-41.
10. Schwaber, C. and P. Sterpe, *Selecting The Right Requirements Management Tool–Or Maybe None Whatsoever. Forrester Research.* Inc.,(2007, 28th September), 2007.
11. Sommerville, I. and P. Sawyer, *Requirements engineering: a good practice guide. 1997.* England: John Willey & Son Ltd.
12. James, L., *What's wrong with requirements management tools?* Requirements Engineering, 1996. **1**(3): p. 190-194.
13. Xu, Q., et al., *An analytical Kano model for customer need analysis.* Design Studies, 2009. **30**(1): p. 87-110.
14. Kano, N., et al., *Attractive quality and must-be quality.* The Journal of the Japanese Society for Quality Control, 1984. **14**(2): p. 39-48.
15. Berger, C., et al., *Kano's methods for understanding customer-defined quality.* Center for Quality Management Journal, 1993. **2**(4): p. 3-36.
16. Grady, J. *An open letter to the tool companies*. in *Proceedings of 6th Intl Symposium of the National Council on Systems Engineering*. 1996.
17. Gabb, A.P., N. Maheswaran, and A.M. Allwright, *Requirements Management Tools Evaluation User Needs and Evaluation Criteria*. 1997, DTIC Document.
18. Jones, D.A., et al. *Factors influencing requirement management toolset selection*. in *Proceedings of the Fifth Annual Symposium of the National Council on Systems Engineering*. 1995.
19. Regnell, B., R.B. Svensson, and K. Wnuk, *Can we beat the complexity of very large-scale requirements engineering?*, in *Requirements Engineering: Foundation for Software Quality*. 2008, Springer. p. 123-128.

20. Punj, G. and D.W. Stewart, *Cluster analysis in marketing research: review and suggestions for application.* Journal of marketing research, 1983: p. 134-148.
21. Romesburg, C., *Cluster analysis for researchers*. 2004: Lulu. com.
22. Jagdev, H., P. Bradley, and O. Molloy, *A QFD based performance measurement tool.* Computers in industry, 1997. **33**(2): p. 357-366.

WEB REFERENCES

23. DAIMLERCHRYSLER *swt.cs.tu-berlin.de* RM-Tool Evaluation, Requirements to RM Tools, 2004 (accessed April 2012)
24. INCOSE, www.incose.org (accessed April 2012)
25. ISIXSIGMA, http://www.isixsigma.com (accessed April 2012)
26. Vrinat, M. (2006). Requirements Management: Solutions Review," Stamford, Connecticut http://www.nohau.se/file/cpda-analyst-report-report.pdf (accessed April 2012)
27. CPDA, www.cpd-associates.com

APPENDIX A

QUESTIONNAIRE

PART 1

Please answer the following questions by placing a check mark in the appropriate box corresponding your preference.

	HOW DO YOU FEEL…..?	**I like it**	**I expected it**	**I'm neutral**	**I can tolerate it**	**I dislike it**
R1	If the RM Tool is easy to use and requires a minimal training?					
	If the RM Tool isn't easy to use and doesn't require a minimal training?					
R2	If the RM Tools has a simple framework?					
	If the RM Tools hasn't a simple framework?					
R3	If the RM Tool is able to identify and capture requirement?					
	If the RM Tool isn't able identify and capture requirement?					
R4	If the RM Tool is able to capture system element structure?					
	If the RM Tool isn't able to capture system element structure?					
R5	If the RM Tool is able to handle a large set of Documents?					
	If the RM Tool isn't able to handle a large set of documents?					
R6	If the RM Tool is able to support complex set of documents?					
	If the RM Tool isn't able to support complex set of documents?					
R7	If the RM Tool is customizable?					
	If the RM Tool isn't					

	customizable?					
R8	If the RM Tool supports extensibility?					
	If the RM Tool doesn't support extensibility?					
R9	If linking and tracing are supported by the RM Tool?					
	If linking and tracing aren't supported by the RM Tool?					
R10	If the RM Tool is able to do a requirement hierarchies?					
	If the RM Tool isn't able to do a requirement hierarchies?					
R11	If the RM Tool is able to identify inconsistence?					
	If the RM Tool isn't able to identify inconsistence?					
R12	If the RM Tool has a storing capabilities?					
	If the RM Tool hasn't a storing capabilities?					
R13	If the RM Tool has a comparing capabilities?					
	If the RM Tool hasn't a comparing capabilities?					
R14	If the RM Tool supports the reuse of requirements across projects?					
	If the RM Tool doesn't support the reuse of requirements across projects?					
R15	If the RM Tool includes workflow capabilities?					
	If the RM Tool doesn't include workflow capabilities?					
R16	If the RM Tool supports reporting and analysis?					
	If the RM Tool doesn't support reporting and analysis?					
R17	If the RM Tool supports					

	change management?					
	If the RM Tool doesn't support change management?					
R18	If the RM Tool has requirements definition features?					
	If the RM Tool hasn't requirements definition features?					
R19	If the RM Tool has decision support capabilities?					
	If the RM Tool hasn't decision support capabilities?					
R20	If the RM Tool has requirements validation capabilities ?					
	If the RM Tool hasn't requirements validation capabilities ?					
R21	If the RM Tool supports integration with other life-cycle tools?					
	If the RM Tool doesn't support integration with other life-cycle tools?					
R22	If the RM Tool has security capabilities?					
	If the RM Tool hasn't security capabilities?					
R23	If the RM Tool has integration with web?					
	If the RM Tool hasn't integration with web?					
R24	If the RM Tool supports collaborative working?					
	If the RM Tool doesn't support collaborative working?					
R25	If Ms Word is supported by the RM Tool?					
	If Ms Word isn't supported by the RM Tool?					

PART 2