An integrated approach to support the Requirement Management (RM) tool customization for a collaborative scenario

An integrated approach to support the Requirement Management (RM) toolcustomization for a collaborative scenario

Abstract

Requirement management represents one of the key process in the complex product life cycle because it is involved not only at the beginning, but also in the further phases where the definition of the technical specificationssometimes implicates requirements tradeoff due to conflicts. For this reason the role of RM tools and methodologies, that normally represents a stand-alone solution, has to change and to be more integrated in the Product Lifecycle Management platform.At present a real shared integrated RM solution doesn't exist and for this reason it is necessary to provide a framework for supporting the customization of the available RM solutions for catching the real and specific company needs in this new collaborative scenario.For this reason this paper presents a methodical approach that incorporates user-centered design principles into the customization process of the tool. It permits to be adopted in each possible company scenario thanks to its ability to catch the company specific needs and further identifying the right features for the company. The proposed methodology puts the user, rather than the system, at the center of the process because the RM solution could be considered effective only if it is able to save time and money in the data management by users. Moreover, this tool assessment method can help organizations efficiently determine candidate tools, to understand what is important in that organization and to make a tool selection customized for their needs. The case study on Requirement Management tools as Part of Product Lifecycle Management (PLM)Solution is presented.

Keywords

Requirement Management (RM), Product Lifecycle Management (PLM),User-Centered Design, Quality Function Deployment (QFD), Kano model

1. Introduction

Requirement Management (RM) is a processwhich helps designers to structure and manage requirements not only in the product design phase but in all phases of New Product Development (NPD). It is one of the most important activities in an industrial company. It is a structured process, usually represented as a defined number of stages, that proceed from idea generation to product commercialization. When NPD is applied to simple product, it seems understandable that product development can be organized as a fluent stage sequence, but when dealing with complex product, the string is not standardized. Obviously, also, in major project there are some steps that cannot be overstepped but frequently some stages have to be revised.

One of the major problems that cause a series of the rearrangement and revision, from concept to detailed design, is the complexity of requirements involved in product development. The product

design process engages many stakeholders that would like to see their needs fulfilled but their needs might be in contrast with each other; this leads to necessary revision of product project in order to create something that fulfill everyone expectation.For this reason the product design process should be dealt with aninteractiveapproach since it should be heavily focused on satisfying the needs and desires of the majority of people who will use the new product. So designers have to deal with new constraints coming not onlyfrom the increasing customer requirements, but also from the new environmental constraints (fuel consumption, emission of dioxide of carbon…),from the constant mutation of the product and from the continuous needed of specialist employees to drive and realize such products and processes(Fischer, Fadel, & Ledoux, 2011).

Moreover, complex project means working with complex information that could lead to misunderstanding and necessary revision. The set of requirements specified with market analysis and interviews might be changed over project proceeding because of external forces and trade-offs. In the light of the acknowledgement of these issues, it goes without saying that it is necessary to compromise between stakeholder's requirements and to manage them.

Requirement Management (RM) follows a process itself that starts with the identification and elicitation of stakeholder's (involved in the product development) requirements, and that finishes after the validation and verification of the proposed set of requirement. Once the set is established, Requirement Management proposal is to continue to revise it in order to keep it under control for possible inconsistency and bias. In order to achieve this,Requirements Management Tools (RMt) must be developed and embedded in Product Lifecycle Management (PLM) solutions.In the last years, Product lifecycle management (PLM) has become essential in companies since it provides technologies and methodologies to manage data, information, and knowledge along the whole product lifecycle(Marco Alemanni, Alessia, Tornincasa, & Vezzetti, 2008; M. Alemanni, Destefanis, & Vezzetti, 2011; Guerra, Gidel, Kendira, Vezzetti, & Jones, 2013; Vezzetti, Violante, & Marcolin, 2014). The tendency is to use a PLM strategy to integrate people, processes, business systems, and information in order to manage the product development and support its lifecycle(Vezzetti, 2009, 2012; Vezzetti, Moos, & Kretli, 2011).In this context, the use of RM toolsas Part of PLM Solutions seem an obvious answer in order to manage product changing, relationship and requests from stakeholders, prototyping and testing and in the end the maintenance of a database of requirements and documents developed.Nowadays this combination is not yet well mature and established. Many organizations just use a general-purpose tool to manage their requirements rather than dedicated requirements management tools. This is a bad practice that has caused many project failures or has placed many projects at risk. Requirements development is an iterative and incremental process that involves elicitation, analysis, development, elaboration, and refinement, all of which need to be performed in a collaborative and transparent environment(blog.enfocussolutions.com).RM and Product Lifecycle Management (PLM) bring unique value to the enterprise, and when well combined, provide a wholly collaborative environment that has a major impact on successful product development performance and the ability to maintain a competitive advantage (http://www.eurostarconferences.com).At present a comprehensive and integrated system that

supports a reliable and integrated requirement management process in the entire product lifecycle is still unavailable (Lau, Mak, & Lu, 2003). This is an open issue in the interactive design studies because every decision in this earlier stages on the product engages the majority of the future costs of design, production, assembly, maintenance, and disassembly (Fischer et al., 2011).Since at present a real integrated RM solution doesn't exist, for to the success of thissynergyit is important to truly understand whichcapabilities best support theactivities of the RM process within Product Lifecycle Management (PLM) platform.

Evaluating and deciding which requirements tools best fit an organization can cost much time, money and resources. There are several requirements tools on the market. Tools can be specialized or fairly generic, expensive or cheap, simple or complex, and so on.For an organization it is important to understand the potentiality of a tool in the right way so that the tool's functionality and its structure are optimized for the best use. Only then the tool can provide reliability, predictability and save time by making it possible to find the right information at the right time (http://pmblog.accompa.com).Over the years, there have been different studies about RM tool, concerning the features to be incorporated in and trying to establish which of the proposed solution can be classified as the best one (www.incose.org, www.seilevel.com ,(Wiegers, 1999)). But for the time being, it is still a little nebulous to determine which of the proposed and existing solutions best incorporate required RM tool features as part of a PLM solution.

As a consequence of this scenario, an user-centered strategy has been developed in this paper. This choice could be justified by the necessity to capture the real company need picture in term of requirements management processes. Only by involving an user interactive methodology, that focalize its attention on those actors that daily work and interact with RM tools, thatformalize, manage and traceproduct requirements along the entire product lifecycle, and share data interactively in a real -timeit is possible to get enough data for starting with an innovative and reliable RM solution design strategy. The proposed methodology has the aim to identify which technical features,disconnected from vendors' offers, an 'synergical' RM tool as Part of PLM Solution should present to meet expectations and latent needs of RM users (employees used to manage requirements involved in the product development) capturing as many as possible users views points. Unlike otherstudies, such asINCOSE (www.incose.org), where only the vendors provide the evaluations, in this evaluation RM users, not affiliated with any vendor, have determined RM tool criteria, priority and validated results. The technical features,identified by users, are defined not only as a list of attributes that characterize Requirement Management tool as Part of PLM Solution, but also for the weight that is connected to every attribute and that identify its importance in thesoftware tool. Moreover, the paper analyzeshow theseidentified key features, which arepart of a requirements management tool in a PLM solution,are supported by specific commercial products.

Moreover, this evaluation process could be useful for the developers to explore which key features are most desirable,plentiful or attractive for the users to be performed and embedded in the future release of the RM tool. Moreover, this tool assessment method can help organizations efficiently determine candidate tools, to understand what is important in that organization, to

define how RM tools handles various key features and use these results to make a tool selection customized for their needs.

In general, this paper presents amethodical approach that incorporates user-centered design principles into the development process of a toolpermitting to select the right features in the context of creating software and helping organizations to make a tool selection customized for their needs.

The paper is organized as follows: in section 2 a brief description of the user-centered approach is presented, in section 3we describe the used methodology and in section 4 we present the case study on the RM tools.

2. Literature review

The user-centered framework of the paper is based on the use of methodologies such as Quality Function Deployment (QDF), Kano's model and Tontini's method.

The philosophy, called user-centered design, incorporates user concerns and support from the beginning of the design process and dictates which user needs should be foremost in any design decisions. In the context of creating software, this approach puts the user, rather than the system, at the center of the process: users work and interact with the product interface and share their views and concerns with the designers and developers.A software development process with this model will enhance the quality of the software. Software quality can be viewed asconformance to software requirementsfrom customers. For effective new-software-development, a systematic approach to understand customer requirements and further embed them into the future software is desirable(Beyer, 1997; Lamont, 2003).

Designers should recognize that they are not typical users. They have more intimate knowledge and understanding of the system they are developing than the average user ever will. Aspects of the interface that are unclear or confusing to most users might therefore be perfectly clear to someone who has worked on the project. So, by focusing on typical users' needs early and revising the design based on user testing often, user-focused software designers produce better designs and, as a result, better products and better acceptance from users(Gould, Boies, & Lewis, 1991; Gould, Boies, & Ukelson, 1988).

Quality must be designed into the product and can be defined as meeting customer needs (where in this paper customer is conceived as company user of the product or service). In literature it is emerged that the use of Quality Function Deployment (QFD) has been beneficial indeveloping new software products andupgrading or enhancing existing softwareproducts. It helps to enhancecommunication between customersand software developers and testers(Haag, Raja, & Schkade, 1996; Karlsson, 1997).

Quality Function Deployment (QFD) is a structured approach to define customer needs or requirements and translate them into specific plans to produce products to meet those needs and to provide superior value. QFD is primarily a people system. Nothing happens without people. Its

point of departure is the ``voice of the customers'' (VoC). The "voice of the customer" is the term to describe these stated and unstated customer needs or requirements. consumer and then translating the consumer's demand into design targets and major quality. In Akao's words, QFD "*is a method for developing a design quality aimed at satisfying the consumer and then translating the consumer's demand into design targets and major quality assurance points to be used throughout the production phase. ... [QFD] is a way to assure the design quality while the product is still in the design stage.*" As a very important side benefit he points out that, when appropriately applied, QFD has demonstrated the reduction of development time by one-half to one-third (Akao, 1997; Akao & Mazur, 2003).

The voice of the customer could be captured in a variety of ways: direct discussion or interviews, surveys, focus groups, customer specifications, observation, warranty data, field reports, etc. According to Griffin and Hauser, at least 20-30 customers should be interviewed to obtain 90-95 per cent of all possible customer needs(Griffin & Hauser, 1993).

The voice of customers has to be transformed into real customer requirements, which means know what and how the customer wants to use the product. Once the needs are identified they have to be classified, a valid method to do this classification is the Kano model(Kano, Seraku, Takahashi, & Tsuji, 1984). The classification is based on the satisfaction that these attributes bring to customers. As customers tend to rate basic requirements with high importance, the traditional QFD method tends to give higher priority to these requirements to the detriment of innovative ones. The Kano model allows the identification of exciting requirements, usually associated with innovations(Tontini, 2007). So this model brings a different perspective for the analysis of improvement opportunities in products and services because it takes into consideration the asymmetry and non-linear relationship between performance and satisfaction (Kano et al., 1984; Matzler & Hinterhuber, 1998). Kano's model classify requirements into different categories, which define different types of perceived quality:

- *Must-be*. This type of requirements has to be fulfilled or the customer will be disappointed. On the other hand, these attributes are taken for granted, so their fulfillment will not increase customer satisfaction. Must-be requirements are prerequisites of a product so customer does not explicit them, but since these are basic criteria, customer expects the product to meet these attributes.
- *One-dimensional*. Quality attributes result in satisfaction when fulfilled and result in dissatisfaction when not fulfilled. Customer satisfaction is proportional to the level of fulfillment of these requirements, and so the dissatisfaction is when requirements are not fulfilled.
- *Attractive*. These are those having the greatest influence on customer satisfaction. They can be described as surprise and delight attributes, and provide satisfaction when achieved fully but do not cause dissatisfaction when not fulfilled. These requirements are not usually expressed by customers, so it is more difficult to explicit and understand them; it goes without saying that their fulfillment is difficult, too.

- *Indifferent.* Their fulfillment or not fulfillment does not influence customer satisfaction or dissatisfaction. Customers are completely uninterested in these requirements.
- *Reverse*. The last category affects those requirements which eventual fulfillment will decrease customer satisfaction. The fulfillment of these requirements should be avoided in order to maintain a high level of customer satisfaction.

The classification of customer's requirements is achieved through the development of a questionnaire. For each requirement a pair of questions is formulated. Each question intends to investigate satisfaction and dissatisfaction on product features, the first question concerns the reaction if the product fulfill that requirement; the second concerns the reaction if the product does not fulfill the same requirement. In order to analyze questionnaire answers, for each requirement functional and dysfunctional answer are combined in a evaluation table, through the combination each requirement is classified. The results from every questionnaire are collected in the table of results and final requirement classification is provided by the analysis of this table; higher frequency is used as discernment criteria.

Kano's classification enables developers to classify requirements into five different classes having different characteristics, however this doesn't provide a value of customer satisfaction. Regarding this, Berger (Berger et al., 1993)proposed to calculate two coefficients; the satisfaction index (SI) and dissatisfaction index (DI) are calculated as follow:

$$SI = \frac{A + O}{A + O + M + I}$$

$$DI = \frac{O + M}{(A + O + M + I)x\ (-1)}$$

where A, O, M and I stands for attractive, one-dimensional, must-be and indifferent columns.

The customer satisfaction coefficients (CS-coefficient) state whether meeting a product requirement can increase satisfaction, or whether fulfilling this product requirements merely prevents the customer from being dissatisfied. The dissatisfaction index (DI) has negative sign cause it emphasize the negative influence that it has on customer satisfaction if product attribute is not fulfilled.The dissatisfaction index (DI) ranges from -1 to 0; the closer the value is to -1, the higher the influence on customer dissatisfaction.The satisfaction index (SI) ranges from 0 to 1; if the value is closer to 1 it means that the influence on customer satisfaction is high. These index are calculated for each requirement in order to define the satisfaction improvement if the requirement is fulfilled, and the dissatisfaction that will be cause by not fulfilling it.

The integration of the Kano model in the QFD allows innovative requirements to receivethe necessary attention in the product's development process. The literature shows that QFD and Kano model can be integrated effectively to identify customer needs more specifically and to yield

maximum customer satisfaction (Chaudha, Jain, Singh, & Mishra, 2011; Matzler & Hinterhuber, 1998; Tan & Shen, 2000; Tontini, 2007).

3. Methodology

The proposed methodology permits to be adopted in each possible company scenario thanks to its ability to catch the company specific needs and further translate in the tool right technical features. Moreover, it can help organizations efficiently to determine candidate toolsmaking a tool selection customized for their needs.

The first phase of the methodology evaluates which most important features a tool should present to meet company expectations (fig.1).

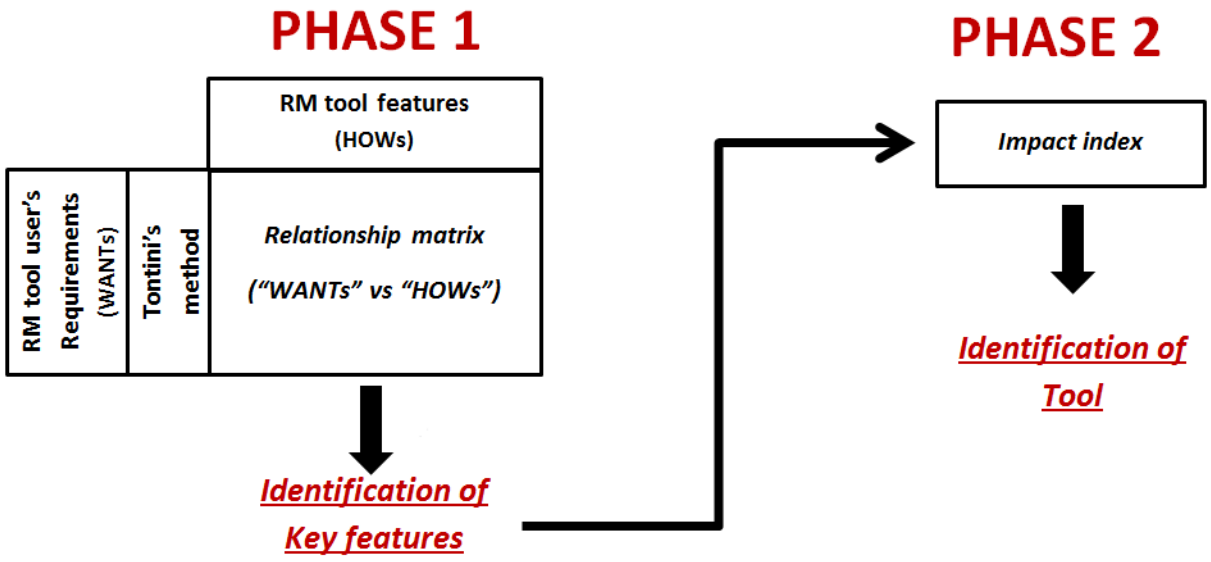

Figure 1 – Methodology scheme

To identify the tool user's requirements (defined as the voice of the customers-VoC) and to list them in order of importance to users the Tontini'sintegration method has been used. In this method, the importance column in the QFD matrix is replaced bythe result of the following equation:

$$Tontini's\ Adj.Factor = Max\ (|SI|; |DI|)$$

where SI and DI are the satisfaction and dissatisfaction indexes (Berger et al., 1993). Theadjustment factor is the higher absolute value of SI or DI, putting more weight on therequirements that bring more satisfaction when present or that bring more dissatisfactionwhen absent(Tontini, 2007). In this case, excitement, performance and basic requirements will be takeninto consideration depending on the degree of satisfaction or dissatisfaction that they couldbring to customers(Tontini, 2007).

After having prioritized the tool user's requirements, the tool technical features have to be determined. This is implies to translate the market model as expressed in subjective terms by the user's words into objective factors of a technical nature that is into a description of the RM tool expressed in the technician's own language (called voice of the engineers VoE). Finally, the key technical characteristics of the tool (those that best affect the satisfaction of each users requirements) have to be educed by mean of *relationship matrix*.

In order to evaluate thekey tool technical features, it is important to determine the level of importance $w_j$of each technical features. It is obtained by summing the products of *relative importance* of each users requirement (obtained by Tontini's method) multiplied by the quantified value of the relationship existing between that *j*-th characteristic and each of the requirements related to it.

The *absolute importance* of the tool features is obtained by:

$$w_j = \sum_{i=1}^{n} d_i \cdot r_{ij}$$

where:

w_j= absoluteimportance of the *j*thtool feature;
d_i = relative importance of the ithtool user requirement obtained by Tontini's method;
r_{ij} = cardinal relationship value between ithtool user requirement the jthtool feature;*i= 1,2, …n* and *j= 1,2,…m*
n = number of tool user requirements;
m = number of tool features.

The measure of the (absolute) level of technical importance has transformed into a measure of relative technical importance w'_j expressed as a percentage:

$$w'_j = \frac{w_j}{\sum_{j=1}^{m} w_j}$$

where:

w'_j= relativeimportance of the *jth* tool feature;
w_j= absoluteimportance of the *j*thtool feature;
j=1,…,m

This measure represents the importance that the user indirectly assigns to each tool features and used to define a ranking order of the tool technical features identifying the key features. If some of these key features will be selected from the tool designers for the development of future release of the tool, it will be ensured that the new product will immediately give the users a greater degree of satisfaction.

Thesecond phase of the methodology measuresthe *impact index*whose intentis to compare existing commercial tools against the number of features that they possess. This index permits to analyze how commercial requirements tools handlethe various key technical features identified in the previous step. This phase is interesting because it could help organizations efficiently determine candidate tools able to supporttheir goals, contexts and mostvalued scenarios.

The introduction of the "*Impact index*" underlines the importance of the discretization according to aborderline value and it is defined as:

$$Impact\ index = \frac{\#\ key\ features}{\#\ n - key\ features}$$

where *key features*($n - key\ features$) are the features with*high*(*low*) values of normalized importance.

4.Experimental Validation on Requirement Management tools

4.1 Definition of RM tool User's Requirements

The first step in the analysis of RM tool is the definition of *user's requirements*; this is usually achieved through focus group, individual interviews and other techniques. Since the aim of this work is to characterize those features that most fulfill user's requirements, inquire tried to distinguish the greatest number of it.

The requirements identified are listed in Table 1; they have been divided into two levels. The first level is more generic and identifies the main features that an RM tool should possess; the second level is more detailed and explains the specific features that users require. For the analysis, the requirements that will be used are those of second level, if not present those of first level will be used as well.

Table 1-List of RM tool requirements identified by users

RM TOOL USER'S REQUIREMENTS			
1st level		**2nd level**	
1	User friendly	**1.1**	Easy to use and minimal training
		1.2	Simple framework
2	Capturing requirement/identification		
3	Capturing system element structure		
4	Managing documents	**4.1**	Handle a large set of documents
		4.2	Being able to support complex set of documents
5	Customizable		
6	Extensibility		
7	Software support	**7.1**	Ms Office tools support
		7.2	Ms Word support
8	Traceability	**8.1**	Linking and tracing

		8.2	Requirement hierarchies
		8.3	Support sharing of common requirements across project
		8.4	Identify inconsistence
9	Linking requirement to the component and to the system elements		
10	Baselining	**10.1**	Storing
		10.2	Comparing
11	Reusability	**11.1**	Reusability of requirements
		11.2	Parent/Child relationships
12	Workflow capabilities	**12.1**	Graphical utilities
		12.2	Tree view
13	Reporting and analysis	**13.1**	Generate freely configurable change reports
		13.2	Documentation output in standard format
14	Change management		
15	Requirements definition features		
16	Decision support capabilities		
17	Requirements validation capabilities		
18	Integration with other life-cycle tools	**18.1**	Inter-tool communications
		18.2	External Applications Program Interface available
		18.3	Support Open database system
		18.4	Integrates with other tools, such as testing, design or project management
19	Security capabilities	**19.1**	Access control
		19.2	Defines users and groups and their access privileges
20	Collaborative working	**20.1**	Support the concurrent view and co-working
		20.2	Multi-level assignment/access control
21	Integration with web		

Once the phase of requirements retrieve has been completed, Kano's method has been used to evaluate them. The role played by the method is to classify requirements in order to recognize those that improve customer satisfaction and those that reduce it. Kano's method finds its basis in the creation of a questionnaire that investigates requirements. The questionnaire has been submitted to users that are familiar with Requirement Management tools; interviewers knowledge on this type of tools is necessary because they already comprehend their utility and principal characteristics.Since not all user's requirements are equally important, with the use of the Kano's model it is possible to verify how the presence or absence of a requirement requested by the user for the RM tool will result in quite a different degree of satisfaction or dissatisfaction. The Kano questionnaire inquires the presence and absence of every requirement with functional and dysfunctional questions. For every requirement, answers of the individual interviewed are combined to obtain the class where each requirement should fall. The further step is to collect these data and assign a category for every requirement according to frequency(Violante & Vezzetti, 2014). After evaluating every compiled questionnaire, the requirement is classified in the category where it reached the higher frequency; if the category can't be assigned, the rule is M>O>A>I.

The pie chart (Figure 2) provides a visual insight of the perception that RM tool users have on the listed requirements. The high majority of this pie is occupied by *Must-be* requirements; this is the signal that there are many requirement that users expect an RM tool to meet, which absence would provoke dissatisfaction.

An interesting data is that concerning *Attractive* requirements, these occupy another great slice of the requirements set; their presence means that there are some demands that RM tool usually do not respond to. It is useful to highlight attractive requirements because of their characteristic of improving customer satisfaction if present, but not decreasing it if absent.

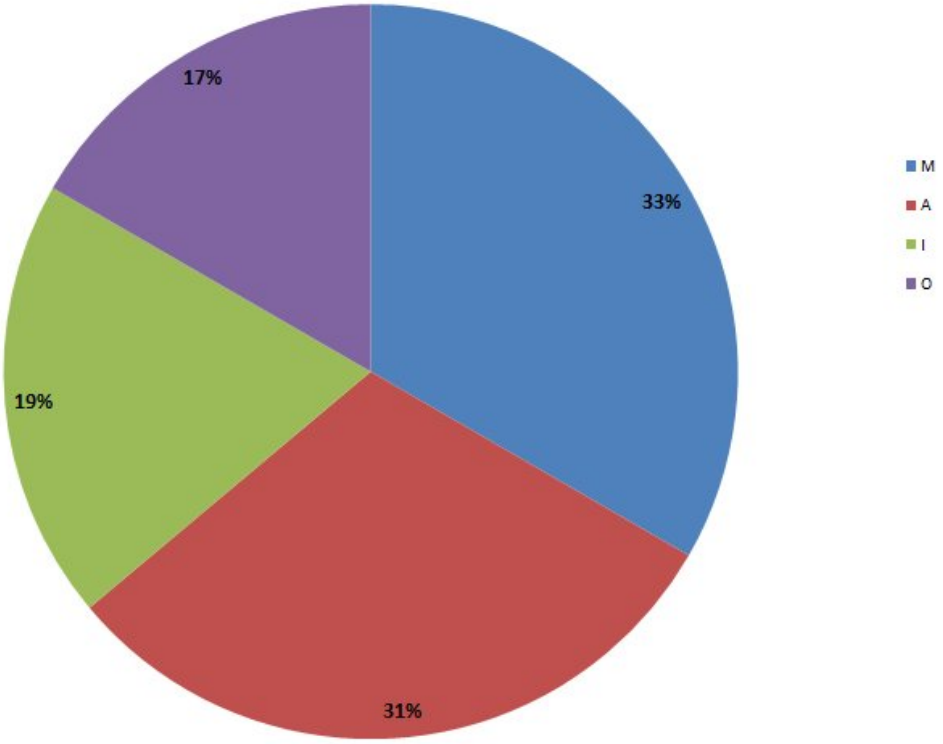

Figure2 - Requirements pie chart

4.2Relative weight

Once requirements have been assignedin the Kano's categories, it is possible to calculate the satisfactory and dissatisfactory index (*SI* and *DI*) and then the Tontini'sadjustment factorused to calculate *relative weight* of every requirement.(Table 2).

Table 2- Berger's indexes, Tontini's adjusted factor and Relative weights

		Kano category	A+O / A+O+M+I	O+M / (A+O+M+I)*(-1)	Tontini's factor	Relative weight
1	Easy to use and minimal training	I	0,333	-0,333	0,333	1,1765%
2	Simple framework	I	0,333	-0,333	0,333	1,1765%
3	Capturing Requirement/identification	M	0,333	-1,000	1,000	3,5294%
4	Capturing system element	A	0,667	-0,333	0,667	2,3529%

	structure					
5	Handle a large set of documents	A	1,000	0,000	1,000	3,5294%
6	Being able to support complex set of documents	A	1,000	0,000	1,000	3,5294%
7	Customizable	O	1,000	-0,667	1,000	3,5294%
8	Extensibility	O	1,000	-1,000	1,000	3,5294%
9	Ms Office tools support	M	0,000	-1,000	1,000	3,5294%
10	Ms Word support	I	0,333	0,000	0,333	1,1765%
11	Linking and tracing	M	0,333	-1,000	1,000	3,5294%
12	Requirement hierarchies	M	0,000	-1,000	1,000	3,5294%
13	Support sharing of common requirements across project	I	0,333	0,000	0,333	1,1765%
14	Identify inconsistence	O	0,667	-1,000	1,000	3,5294%
15	Linking requirement to the component and to the system elements	A	0,667	-0,333	0,667	2,3529%
16	Storing	M	0,667	-0,667	0,667	2,3529%
17	Comparing	O	0,667	-1,000	1,000	3,5294%
18	Reuse of requirements across projects	M	0,000	-1,000	1,000	3,5294%
19	Parent/Child relationships	M	0,000	-1,000	1,000	3,5294%
20	Graphical utilities	A	0,667	0,000	0,667	2,3529%
21	Tree view	M	0,000	-1,000	1,000	3,5294%
22	Generate freely configurable change reports	O	0,667	-1,000	1,000	3,5294%
	Documentation output in standard format	M	0,000	-1,000	1,000	3,5294%
24	Change management	O	0,667	-1,000	1,000	3,5294%
25	Requirements definition features	M	0,333	-0,333	0,333	1,1765%
26	Decision support capabilities	I	0,000	-0,333	0,333	1,1765%
27	Requirements validation capabilities	M	0,333	-0,333	0,333	1,1765%
28	Inter-tool communications	A	1,000	-0,333	1,000	3,5294%
29	External Applications Program Interface available	A	1,000	0,000	1,000	3,5294%
30	Support Open database system	I	0,333	0,000	0,333	1,1765%
31	Integrates with other tools, such as testing, design, and project management	A	1,000	-0,333	1,000	3,5294%
32	Access control	A	0,667	-0,333	0,667	2,3529%
33	Defines users and groups and their access privileges	M	0,000	-1,000	1,000	3,5294%
34	Support the concurrent view and co-working	A	1,000	-0,333	1,000	3,5294%
35	Multi-level assignment/access control	A	1,000	0,000	1,000	3,5294%
36	Integration with web	I	0,000	-0,333	0,333	1,1765%
					28,333	100%

4.3Definition of the "RM Tool features"

To complete the House of Quality, RM tool technical specifications corresponding to*voice of engineers* need to be delineated; these are the translation of RM tool user's requirements into characteristics that compose the RM tool. These have been outlined by experts (Tool vendors and developers) and they are reported in Table 3.

Table 3- List of RM tool technical specification

TECHNICAL SPECIFICATIONS		
Characteristic		**Explanation**
1	Open architecture	Software architecture that is designed to make adding, upgrading and easily changing components
2	API	Protocol intended to be used as an interface by software components to communicate with each other
3	Role-based access control	Approach to restricting system access to authorized users
4	Multi-user secure access	Provide access by multiple user of a computer
5	Web based user interface	Interface that accept input and provide output by generating web pages which are transmitted via the Internet and viewed by the user using a web browser program
6	Historical database	Organized collection of data
7	Project data search and retrieve	Creation of a safe storage and backup for data and projects
8	Project data definition	Title, description, context, and usage of a reporting element
9	Guided procedures	Simple structure to define the project, waterfall life-cycle or agile development
10	Modular projects	Interchangeable modules that each contains everything necessary to execute only one aspect of the entire project
11	Visual prototyping	Integration with software to validate a design before making a physical prototype (CAE, CAD,...)
12	Integrated test solution	Evaluation of the system's compliance with specified requirements, status reporting of project compliance

The technical features can be unified into 5 main categories:

- Integration with other tools:
  - Open architecture
  - API
- User interface:
  - Role-based access control
  - Multi-user secure access
  - Web based user interface
- Data management:
  - Historical database
  - Project data search and retrieve
- Project support tools:
  - Project data definition
  - Guided procedures
  - Modular projects
  - Visual prototyping

- Validation and verification:
  - Integrated test solution

Technical attributes mainly refer to requirement management as the definition of data and input and their managing, storing and tracking. Instead other attributer are linked to users, for example the possibility to access to the project by more than one user, or the guided procedures that can be applied when working with a new product. Another group of characteristics deals with the integration of the tool with other tool in order to use different software during product development.

4.4Construction of the relationship matrix

To estimate the relationship between RM tool users requirements and RM tool specifications, another questionnaire has been proposed and submitted to users. The values proposed by users are shown in Table 4.The correlation value (or cardinal relationship) indicates the influence strength that a technical specification has on user's requirements: the values 1, 3, 9 represent respectively low, medium and strong correlation.

Table 4 - User's requirements-technical specification correlation values

		TECHNICAL SPECIFICATION											
		1	2	3	4	5	6	7	8	9	10	11	12
RM TOOL USER'S REQUIREMENTS		Open architecture	API	Role-based access control	Multi-user secure access	Web based user interface	Historical database	Project data search and retrieve	Project data definition	Guided procedures	Modular projects	Visual prototyping	Integrated test solution
1	Easy to use and minimal training			3		9				9			
2	Simple framework					3							
3	Capturing Requirement/identification			3				3	9	3	9		
4	Capturing system element structure							3	3	3			
5	Handle a large set of documents							3	3				
6	Being able to support complex set of documents								9			9	
7	Customizable	9	9										
8	Extensibility	9	9			3			3				
9	Ms Office tools support	3	3						3				
10	Ms Word support	3	3						3				
11	Linking and tracing							9	9				
12	Requirement hierarchies							9	9		3		
13	Support sharing of common requirements across project						3		9		9		

14	Identify inconsistence							3	9	3	3		3
15	Linking requirement to the component and to the system elements							3	9	3			
16	Storing						3	9	3				
17	Comparing						3	9	9				
18	Reuse of requirements across projects						9	9	9	3	9		
19	Parent/Child relationships							3	9	3			
20	Graphical utilities	3	3			3						9	
21	Tree view					3	1		3				
22	Generate freely configurable change reports							3		3			
23	Documentation output in standard format		3										
24	Change management						9	3	3			3	
25	Requirements definition features						3	3	9				
26	Decision support capabilities			9		1	9	9	3				
27	Requirements validation capabilities							3			1	3	9
28	Inter-tool communications	3	3			3			3				
29	External Applications Program Interface available	9	9										
30	Support Open database system	3	3										
31	Integrates with other tools, such as testing, design, and project management	9	9						3				9
32	Access control			9	9								
33	Defines users and groups and their access privileges			9	9				3				
34	Support the concurrent view and co-working			9	9				3				
35	Multi-level assignment/access control			9	9								
36	Integration with web					9							

A first analysis of these correlation values highlights that there are some requirements, such as *Project data search and retrieve* and*Project data definition*, that influence an higher number of requirements. These RM tool features will probably have a higher value of absolute and relative importance in the House of Quality; it leads to the consideration that developer should put extra effort in enhance these attributes.

On the other hand, other attributes influence a lower number of needs; it might be interpreted as these attributes are not worthy to be developed, but from a general point of view they collaborate to the creation of a superior product.

4.5Definition of the "RM tool Key features"

In order to evaluate the key RM tool technical features,it is important to integrate the relationship matrix with the requirements relative weights obtained by Tontini's method.In Table 5 the final values of absolute and relative importance for each RM tool featureare summarized.

Table 5 Absolute and Relative importance of the technical features

	1	2	3	4	5	6	7	8	9	10	11	12	
	Open architecture	API	Role-based access control	Multi-user secure access	Web based user interface	Historical database	Project data search and retrieve	Project data definition	Guided procedures	Modular projects	Visual prototyping	Integrated test solution	TOTAL
Absolute Importance	1,6235	1,7294	1,4118	1,1647	0,6471	1,0235	2,4353	4,1294	0,7765	0,9647	0,6706	0,5294	17,106
Relative Importance	9,491%	10,110%	8,253%	6,809%	3,783%	5,983%	14,237%	24,140%	4,539%	5,640%	3,920%	3,095%	100%

Theclassification of technical features according to the relative importance is reported in Table 6.

Table 6 -Classification of technical features

#	Technical specification	Relative Importance
1	Project data definition	24,140%
2	Project data search and retrieve	14,237%
3	API	10,110%
4	Open architecture	9,491%
5	Role-based access control	8,253%
6	Multi-user secure access	6,809%
7	Historical database	5,983%
8	Modular projects	5,640%
9	Guided procedures	4,539%
10	Visual prototyping	3,920%
11	Web based user interface	3,783%
12	Integrated test solution	3,095%

RM tool features such as *Project data search and retrieve* and *Project data definition*occupy the first two positions; their importance is easily explained because these features are connected with the definition and storage of project data.

Focusing the attention to technical attributes that have a lower importance, such as *Visual prototyping* and *Integrated test solution*, these features are as well indispensable for an RM tool but their influence on user's requirements is not as strong as that of other features. As said, if the attention is addressed to the whole project, these attributes acquire importance because cover final activities in product/project development, but if the attention is directed only to requirement

management, their implication do not emerge. Relative importance means to indicate the technical features that developers should focus on to meet user's needs.

To conclude, it is possible to state that users are obviously more interested in technical attributes that deal with the creation and management of requirements, this intend that users expect the tool to be able to characterize requirements, create link between them and mostly to keep tracing of their evolution in order to be easily found.

An important value of Relative Importance is hold by attributes that refers to the integration and communication with other software. Their value is explained by the importance that the ability to communicate and use other tool beside of RM tool has when developing a product. Since working on a new project concern many people, RM tool is required so support multiple accesses on the same project at the same time.

The importance values of technical characteristics reflect the influence that each attribute has on user's needs. These technical features represent the ideal characteristics that an RM tool should possess and that generally cover the entire set of requirement and the activities that concern the use of this type of tool.

Obviously all these characteristics might not be present in a RM tool because of choices that have to be made when developing software. Since this product is time and money consuming, developers have to deal with costs and quality trade-off.

As said some are more important than others and relative values prove it, so it is possible to discretize technical features by their relative importance. The relative importance value of 6% has been taken as *breaking point*, so that features with relative importance equal or higher than 6% are considered "*key features*", the other ones are defined not key feature*("n-key feature")*. The word *"Key"*does not refer to the real quality of technical features, since every feature is important and should implemented; but ,from customer's point of view, some technical arrangements are more valuable than others and when a decision has to be taken it is preferable to have a propensity for those features that increase customer satisfaction.

In order to define the "key features", RM tool technical specifications have beensort out according to the borderline value that is 6% (Table 7).

Table7 - Technical featuresclassification

KEY features (Relative Importance ≥ 6%)	**N-KEY features** (Relative Importance < 6%)
Project data definition	Historical database
Project data search and retrieve	Modular projects
API	Guided procedures
Open architecture	Visual prototyping
Role-base access control	Web based user interface

Multi-user secure access	Integrated test solution

4.6Impact index

The introduction of the "*Impact index*"underlines the importance of the discretizationaccording to the borderline value of 6%. In order to apply the Impact index in RM tool study and analyze the impact of the key features inexistingRM tools, INCOSE and the Atlantic System Guild's (www.volere.co.uk/tools.htm)databases have been used. The table 8 shows which technical features the three RM tools possess while the Table 9 shows the Impact index of the three RM tools.

Table 8 –RM tool features

RM tool	Project data definition	Project data search and retrieve	API	Role-base access control	Open architecture	Multi-user secure access	Historical database	Modular projects	Guided procedures	Web based user interface	Visual prototyping	Integrated test solution
RM tool 1	x	x	x	x	x	x		x	x	x	x	x
RM tool 2	x	x		x	x	x	x	x	x	x	x	x
RM tool 3	x	x		x	x	x		x	x	x	x	x

Table 9 -Impact Index

RM tool	# Key features	# N-key features	Impact Index
RM tool 1	6	5	1,2
RM tool 2	5	6	0,83
RM tool 3	5	5	1

Theseresults how that the RM tool with the highest number of "not key" characteristics and the lowest value of impactindex doesn't meet user expectation.

It is possible to evaluate the three RM tools by comparing every main category of the technical features and place them in a radar chart (Figure 3). Their comparison is not on the presence or absence of technical specification but on how these have been embeddedby each RM tool. The analysis is made by looking if every technical category has been integrated and developed in complete, discrete, inadequate way or hasn't been developed at all. Technical specifications categories have been placed on a 1 to 4 scale, where 1 correspond to absence of attribute

development, 2 to inadequate development, 3 to discrete development, 4 to complete development.

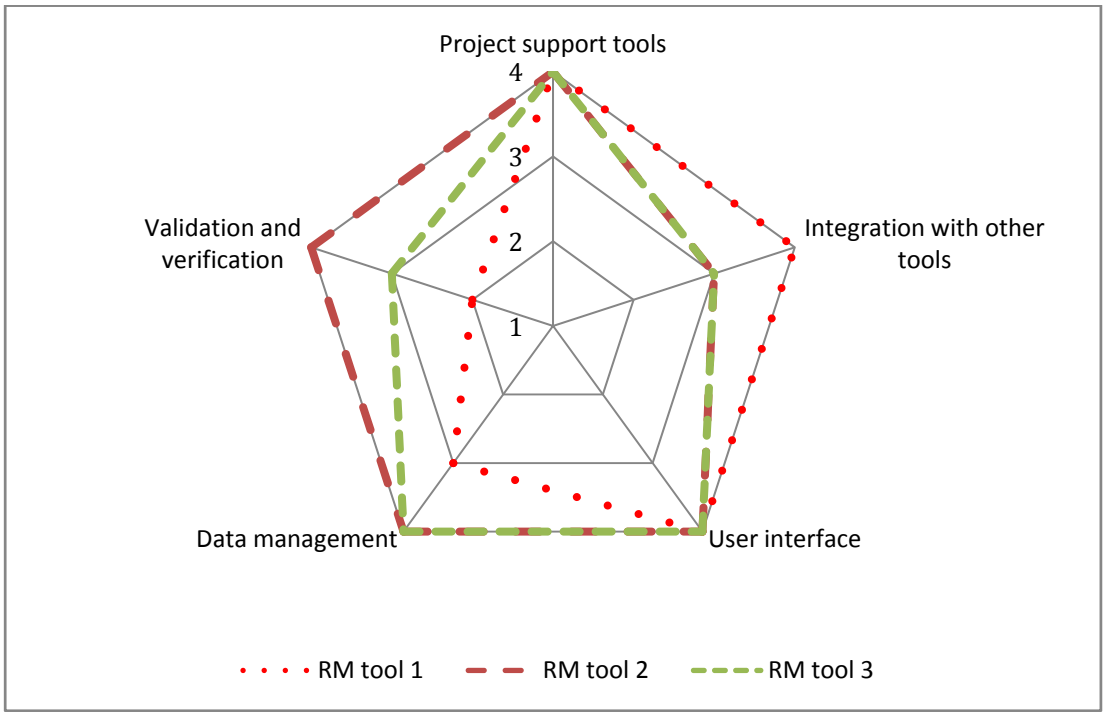

Figure 3. Existing RM tool comparison

As it is possible to see, *User interface* and *Project support tools* have been developed in a complete way in all three RM tools; the other three major categories, instead, have been implemented in different ways by every RM tool, which determine a different positioning in radar chart.

Integration with other tools have been fully integrated by RM tool 1, this means that the attributes *Open architecture* and *API* have been greatly developed in that tool. Since these technical attributes both have a relative importance value higher than 6%, so have been considered as "key characteristics", their development is a great result cause it will guarantee an high customer satisfaction.

On the other hand, *Verification and validation* is well implemented by RM tool 2 and not fully developed by the other tools. Since this category includes *Integrated test solution* its fully integration in the RM tool will not assure customer satisfaction because relative importance is the lower among all technical specification. Moreover, the same line of reasoning can be followed for *Data management*; this category covers two technical attributes, *Project data search and retrieve* (key feature) and *Historical database* (n-key feature).

This analysis agrees with that made by Impact Index. The tool having the lower value of Impactindex is that that has improved technical features with lower value of relative importance. Instead, RM tool 1 has high Impact index and technical specification less developed are those with lower relative value, instead *Project support tools*, *Integration with other tools* and *User interface* have been fully developed since these are those consisting of high relative importance attributes.

In the end, it seems obvious to state that when financial resources are limited, it is better to develop fewer technical attributes and to focus on those that really matter to customers that guarantee to gain high customer satisfaction, instead of developing those of lower importance.

6. Conclusion

For effective new-software-development, a systematic approach to understand customer requirements and further embed them into the tool is desirable. This paper presents an user-centered framework based on the use of Quality Function Deployment (QDF) and a method of integration of Quality Function Deployment (QDF) into Kano model (Tontini's method) usable in the development ofany tools. Evaluating and deciding which tools best fit an organization can cost much time, money and resources. Thisintegrated approach can help organizations efficiently determine candidate tools and to make a tool selection customized for their needs.

The case of study on Requirements Management tools integrated in a PLM scenario is presented. Thekey technical features studied in the paper represent ideal characteristics that an RM tool should possess and that generally cover the entire set of requirement and the activities that concern the use of this type of tool within PLM platform. Obviously all these characteristics might not be present in a RM tool because of choices that have to be made when developing software. Since this product is time and money consuming, developers have to deal with costs and quality trade-off. The results presented provide a suggestion for future RM tool development; however nowadays there are already many tools that can be adopted. Usually the technical specification that is the most absent in tools is that related to the creation of reporting about project status. As has been described by Relative Importance, the attribute Integrated test solution is the one that characterize the lower number of user's requirements and that have a lower weight on the set of technical specification.Other features that are recurrently absent are those concerning the integration with other tools. Usually Ms Office is supported, but it is not possible to make some addition and to interface the tool with other software. This deficiency is a negative reinforcement in RM tools because from the analysis before emerged that the possibility of integrating and using other software has a great impact for users.

In addition, the results of thisintegrated approach can help organizations to understand which tool is appropriated in that organization since the approach defines how RM tool handles various key features identifiedmaking a tool selection customized for the organization needs.

Web References

INCOSE http://www.incose.org/INCOSE Tool Database Working Group (TDWG), INCOSE Requirements Management Tools Survey, Accessed 25/09/2013

John Parkerhttp://enfocussolutions.comConsiderations When Choosing a Requirements Management Tool, CEO of Enfocus Solutions Inc, Accessed 25/09/2013

Ulf Eriksson http://www.eurostarconferences.com Choosing the Right Tool for your Testing and Requirements Management,Accessed 25/09/2013

Michael Shrivathsanhttp://pmblog.accompa.com Requirements Management Tools – Overview Accessed 25/09/2013

Beatty, J., Ferrari, R. www.seilevel.com/ba-resources/requirements-toolsreviews/(2011), "How to Evaluate and Select a Requirements Management Tool", *SeilevelWhitepaper*, Accessed 25/09/2013

http://www.volere.co.uk/tools.htm, Accessed 25/09/2013

References

Akao, Y. (1997). *QFD: Past, present, and future.* Paper presented at the International Symposium on QFD.

Akao, Y., & Mazur, G. H. (2003). The leading edge in QFD: past, present and future. *International Journal of Quality & Reliability Management, 20*(1), 20-35.

Alemanni, M., Alessia, G., Tornincasa, S., & Vezzetti, E. (2008). Key performance indicators for PLM benefits evaluation: The Alcatel Alenia Space case study. *Computers in industry, 59*(8), 833-841. doi: http://dx.doi.org/10.1016/j.compind.2008.06.003

Alemanni, M., Destefanis, F., & Vezzetti, E. (2011). Model-based definition design in the product lifecycle management scenario. *The International Journal of Advanced Manufacturing Technology, 52*(1-4), 1-14. doi: 10.1007/s00170-010-2699-y

Berger, C., Blauth, R., Boger, D., Bolster, C., Burchill, G., DuMouchel, W., . . . Shen, D. (1993). Kano's methods for understanding customer-defined quality. *Center for Quality Management Journal, 2*(4), 3-36.

Beyer, H. (1997). *Contextual Design: Defining Customer-Centered Systems*: Morgan Kaufmann Publishers Inc.

Chaudha, A., Jain, R., Singh, A. R., & Mishra, P. K. (2011). Integration of Kano's Model into quality function deployment (QFD). *The International Journal of Advanced Manufacturing Technology, 53*(5-8), 689-698. doi: 10.1007/s00170-010-2867-0

Fischer, X., Fadel, G., & Ledoux, Y. (2011). Interactive Product Design *Research in Interactive Design Vol. 3* (pp. 45-84): Springer Paris.

Gould, J. D., Boies, S. J., & Lewis, C. (1991). Making usable, useful, productivity-enhancing computer applications. *Communications of the ACM, 34*(1), 74-85.

Gould, J. D., Boies, S. J., & Ukelson, J. (1988). How to design usable systems. *Handbook of human-computer interaction, 2*, 231-254.

Griffin, A., & Hauser, J. R. (1993). The voice of the customer. *Marketing science, 12*(1), 1-27.

Guerra, A. L., Gidel, T., Kendira, A., Vezzetti, E., & Jones, A. (2013). *Co-evolution of design tactics and CSCWD systems: Methodological circulation and the TATIN-PIC platform.* Paper presented at the DS 75-9: Proceedings of the 19th International Conference on Engineering Design (ICED13), Design for Harmonies, Vol. 9: Design Methods and Tools, Seoul, Korea, 19-22.08. 2013.

Haag, S., Raja, M. K., & Schkade, L. L. (1996). Quality function deployment usage in software development. *Commun. ACM, 39*(1), 41-49. doi: 10.1145/234173.234178

Kano, N., Seraku, N., Takahashi, F., & Tsuji, S. (1984). Attractive quality and must-be quality. *The Journal of the Japanese Society for Quality Control, 14*(2), 39-48.

Karlsson, J. (1997). Managing software requirements using quality function deployment. *Software Quality Journal, 6*(4), 311-326. doi: 10.1023/A:1018580522999

Lamont, S. (2003). *Case study: Successful adoption of a user-centered design approach during the development of an interactive television application.* Paper presented at the Anais do 1st European Interactive Television Conference. Brighton.

Lau, H. Y. K., Mak, K. L., & Lu, M. T. H. (2003). A virtual design platform for interactive product design and visualization. *Journal of Materials Processing Technology, 139*(1–3), 402-407. doi: http://dx.doi.org/10.1016/S0924-0136(03)00510-7

Matzler, K., & Hinterhuber, H. H. (1998). How to make product development projects more successful by integrating Kano's model of customer satisfaction into quality function deployment. *Technovation, 18*(1), 25-38. doi: http://dx.doi.org/10.1016/S0166-4972(97)00072-2

Tan, K. C., & Shen, X. X. (2000). Integrating Kano's model in the planning matrix of quality function deployment. *Total Quality Management, 11*(8), 1141-1151. doi: 10.1080/095441200440395

Tontini, G. (2007). Integrating the Kano Model and QFD for Designing New Products. *Total Quality Management & Business Excellence, 18*(6), 599-612. doi: 10.1080/14783360701349351

Vezzetti, E. (2009). Product lifecycle data sharing and visualisation: Web-based approaches. *The International Journal of Advanced Manufacturing Technology, 41*(5-6), 613-630.

Vezzetti, E. (2012). A knowledge reusing methodology in the product's lifecycle scenario: a semantic approach. *INTERNATIONAL JOURNAL OF MANUFACTURING TECHNOLOGY AND MANAGEMENT, 26*(1), 149-160.

Vezzetti, E., Moos, S., & Kretli, S. (2011). A product lifecycle management methodology for supporting knowledge reuse in the consumer packaged goods domain. *Computer-Aided Design, 43*(12), 1902-1911.

Vezzetti, E., Violante, M. G., & Marcolin, F. (2014). A benchmarking framework for product lifecycle management (PLM) maturity models. *The International Journal of Advanced Manufacturing Technology, 71*(5-8), 899-918.

Violante, M. G., & Vezzetti, E. (2014). A methodology for supporting requirement management tools (RMt) design in the PLM scenario: An user-based strategy. *Computers in industry, 65*(7), 1065-1075. doi: http://dx.doi.org/10.1016/j.compind.2014.05.001

Wiegers, K. E. (1999). Automating requirements management. *Software Development, 7*(7), 1-5.