María Florencia Santillán

CRITERIA FOR THE EVALUATION OF REQUIREMENTS MANAGEMENT TOOLS SUPPORTING DISTRIBUTED SOFTWARE PRODUCT LINE ENGINEERING AND MANAGEMENT

UNIVERSITY OF JYVÄSKYLÄ
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS
2015

ABSTRACT

Santillán, María Florencia
Criteria for the evaluation of requirements management tools supporting distributed software product line engineering and management
Jyväskylä: University of Jyväskylä, 2015, 108 p.
Information Systems, Master's Thesis
Supervisors: Heimbürger, Anneli
Käkölä, Timo

The development of software product lines (SPL) by geographically distributed teams increases the complexity of requirements management (RM) related tasks, making the support of a RM tool necessary. However, the selection of the most appropriate RM tool can be challenging, therefore the use of a comprehensive criteria list guiding the selection becomes essential. The goal of this research is to determine whether available criteria lists include comprehensive criteria for the evaluation of RM tools to be adopted in distributed SPL requirements management.

This research answers the following research questions: (1) Do currently available criteria lists for RM tool evaluation allow the assessment of tools supporting distributed SPL requirements management? (2) To what extents do criteria lists support the evaluation of RM tool general features, as well as SPL requirements and distributed collaboration related features? and (3) How would a comprehensive criteria list for the evaluation of tools used in distributed SPL requirements management look like? To answer these questions a literature review and a constructive research were conducted. The review focused on fully available criteria lists for the evaluation of RM tools, as well as research in the fields of distributed collaboration and SPL to identify tool features necessary in these activities. Based on the results obtained, the constructive research was conducted to create a framework to evaluate the available criteria lists and guide the creation of the new criteria list.

The analysis of the available criteria lists left in evidence the lacking of a list with comprehensive criteria for the evaluation of tools features supporting distributed SPL requirement management. Therefore, a new list was created taking as a base the criteria list found most comprehensive and extending it with new requirements. The evaluation framework, the comparison of available criteria lists and the newly created criteria list, present a contribution to current research, however, further research is required to validate these results.

Keywords: requirements management, requirements management tools, tool evaluation criteria, criteria lists, software product line, distributed collaboration

FIGURES

TABLES

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Background and motivation

Requirements engineering (RE) is a vital part of the systems engineering process (Hull, Jackson & Dick, 2011). Laplante (2014), in the introduction to his book, defines it as "the key to improved, on-time, and on-budget delivery of software and systems projects (2014, p. xvii)". Failing in performing other project activities can lead to software defects or overruns in costs or schedule, but failing to deliver good requirements can mean the development of the wrong product (Chemuturi, 2013). Requirements constitute the basis of every project (Hull et al., 2011), especially important because they are the starting point for all further work (Young, 2004).

RE can be divided into two main sub-disciplines, Requirements development (RD) and Requirements management (RM) (Wiegers & Beatty, 2013). The activities performed during RD give as a result a set of requirements that are reviewed and agreed on. However, even after an agreement on the set of requirements has been reached, further changes are often unavoidable. These changes can be caused by a modification in the priorities of the business, errors or omissions found in the requirements, the discovery of new requirements, or due to changing external circumstances. The process in charge of managing these changes constitutes the second RE main activity, namely RM. (Kotonya et al., 1998.) The objective of RM is to keep track of changes and ensure that modifications to the requirements document are done in a controlled fashion (Kotonya et al., 1998), maintaining the integrity, accuracy and currency of the agreed requirements for the duration of the project (Wiegers et al., 2013).

Until today, one of the most used approaches for documenting and managing the project's requirements is by structured specifications written in natural-language. However, this approach presents a wide range of limitations as projects grow in size and complexity (Wiegers et al., 2013.), as happens in software product line (SPL) development and distributed companies (Bosch & Bosch-Sijtsema, 2010). In these scenarios, the support of a RM tool becomes

fundamental. (Violante & Vezzetti; 2014; Derakhshanmanesh, Fox & Ebert, 2014; Beuche et al., 2007; Sinha, Sengupta & Chandra, 2006; Lang & Duggan, 2001) Nowadays, an extensive amount of RM tools are available in the market (Volere, 2014; INCOSE, 2010; Birk & Heller, 2014; Alexander, 2014). These tools range from simple web-based applications storing requirements' information, to potent multiuser products with a variety of feature sets and the ability to handle vast projects (Wiegers, 2006).

RM tools make requirement-related activities more sophisticated and capable (Wiegers et al., 2013), yet the selection of the most appropriate tool for a specific company or project can be challenging (Carvallo, Franch & Quer, 2005). According to Young (2004), choosing the wrong tool, or one that is too complex, can hinder the project. Therefore, the company or project team selecting a RM tool should use a criteria list that allows the evaluation of available RM tools according to their specific needs (Carvallo et al., 2005). A criteria list allows its user not only to know what types of functionalities can be expected in a RM tool, but also helps improve the tool selection process by reducing the amount of time required and by making it less prone to errors (Carvallo et al., 2005).

The construction of a criteria list is "a time-consuming and costly activity" (Carvallo et al., 2005, p. 136). Therefore, the usage of an already available criteria list can save these valuable company resources. At present, several criteria lists for the evaluation of RM tools are available to be used directly, or be taken as a reference, when selecting a RM tool (INCOSE, 2010; Gabb, Maheswaran & Allwright, 1997; Bokhorst, 2001; Hoffmann, Kühn, Weber & Bittner, 2004a; Beuche et al., 2007; Seilevel, 2011a). These lists present different levels of detail and focus, therefore being appropriate for diverse situations. In their research, Beuche et al. (2006) identified the need to incorporate SPL related requirements to current criteria lists to support the selection and development of RM tools used during SPL development. This research group also recognized the importance of tool features supporting the collaboration of team members involved in the development of SPLs, who are often distributed. However, a preliminary inspection of criteria lists available at present suggests that detail requirements, for the evaluation of SPL features as well as distributed collaboration features, are still missing.

The goal of this Master thesis is to study the degree of support provided by available criteria lists for the evaluation of RM tools used by companies or teams managing SPLs requirements in a distributed setting. To achieve this, criteria lists will be analyzed and compared based on their inclusion of requirements for the evaluation of core RM features, as well as requirements for the evaluation of distributed SPL requirements related features. Requirements to evaluate distributed SPL requirements related features will be identified from practical cases of developed tools and solutions presented in published research. The results of the analysis and comparison will indicate whether a comprehensive criteria list is needed supporting the evaluation of those RM tool features required by the mentioned user group. These results, in

combination with the features observed from the literature, will serve as a base for the construction of such criteria list.

1.2 Key concepts

This section introduces the key concepts used throughout this research.

Requirements – "A specification of what should be implemented. They are descriptions of how the system should behave, application domain information, constraints on the system's operation, or specifications of a system property or attribute. Sometimes they are constraints on the development process of the system." (Kotonya et al., 1998, p. 6)

Requirements engineering – "Subset of systems engineering concerned with discovering, developing, tracing, analyzing, qualifying, communicating and managing requirements that define the system at successive levels of abstraction" (Hull, Jackson & Dick, 2011, p. 8)

Requirements management tool – Software system supporting RM activities

Software product line – "Group of products having a common set of features that are varied to satisfy the requirements of two or more applications" (Sutherland, 2002, p. 1)

Distributed collaboration – Cross-site communication and coordination between of distributed teams members (Damian, Marczak & Kwan, 2012)

1.3 Research questions

The selection of a RM tool is a critical activity requiring special attention. This task becomes even more complex when evaluating tools meant to support the RM activities of teams developing SPLs in a distributed environment. In this context, the organization or project selecting the tool needs to take additional features into consideration during tool evaluation. The utilization of a comprehensive criteria list is, in these cases, a key factor for the thoroughly evaluation of available RM tools.

The first step of this research consists of the identification of available criteria lists and the analysis of their suitability for the evaluation of RM tools by companies or teams developing SPLs in a distributed setting. This leads to the first research sub-question:

RQ1: Do currently available criteria lists for RM tool evaluation allow the assessment of tools supporting distributed SPL requirements management?

Once available criteria lists have been identified and analyzed to get to know their general support for RM activities in distributed SPL development environments, a more detailed analysis will be necessary. During this step, the level of detail in which these criteria lists help in the assessment of critical features in the specified context will be evaluated. This leads to the second research sub-question:

RQ2: To what extents do criteria lists support the evaluation of RM tool general features, as well as SPL requirements and distributed collaboration related features?

The results obtained from the previous analysis would indicate if a more comprehensive criteria list for the evaluation of RM tools supporting distributed SPL requirements management is required. The aspects to be taken into consideration for the construction of such a criteria list lead to the third and final sub-question of this research:

RQ3: How would a comprehensive criteria list for the evaluation of tools used in distributed SPL requirements management look like?

The questions stated above will be answered in the remaining of this document.

1.4 Research limitations

A set of aspects were determined in order to delimit the scope of the present research. Bearing in mind that the main focus of this study is on criteria lists for the evaluation of RM tools, only criteria lists conceived for RM tool assessment shall be taken into account. Lists containing only requirements for the evaluation of tools supporting other requirements engineering activities, such as tools for requirements' writing, modelling, or elicitation, shall not be considered.

Only criteria lists that have been published entirely and are available to the general public shall be taken into account for detailed analysis and comparison. This will allow the research to concentrate on those lists that are actually fully available to practitioners in the requirements field. Partially available lists shall only be used as a reference for the development of the evaluation framework used to compare fully available criteria lists and for the subsequently creation of the new criteria list.

The level of support provided by available criteria lists shall be evaluated by conducting a quantitative analysis of the amount of requirements presented by the lists under a unified set of categories. This approach was selected, over a

qualitative analysis of the requirements descriptions, due to its transparency and the analysis possibilities it offers.

The new criteria list for RM tool evaluation constructed as a result of this research shall not be employed in a practical environment for the actual evaluation of RM tools available in the market. However, the requirements included in the new criteria list shall originate either from criteria lists that have been successfully applied in practice or from validated research results in relevant fields. This decision is based on time restraints originating from the extent of the research.

1.5 Research method

To answer the questions proposed in this thesis two research methods are used, systematic literature review and design science research (DSR). According to Kitchenham (2004, p. 1), a systematic literature review is ‘‘a means of evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest”. Systematic reviews are conducted for a variety of reasons, such as the compilation of evidence, the identification of gaps in current research, or to serve as a background for further research activities (Kitchenham, 2004), as is the case in this research. The results obtained from the literature review will be the input of the DSR. “Design science (...) creates and evaluates IT artifacts intended to solve identified organizational problems” (Hevner, March, Park & Ram, 2004, p. 77). The problem identified in this study consists in the restricted support provided by criteria lists for RM tool evaluation to teams or companies managing SPL requirements in a distributed context.

The guideline proposed by Kitchenham (2004), which was adapted for its utilization by software engineering researchers, is followed in this study. This guideline is divided in three main phases, namely planning the review, conducting the review, and reporting the review. During the first phase the research questions are specified, and the review protocol is developed and validated. During the second phase, relevant research is identified, primary studies are selected and their quality is assessed, and data is extracted and synthesized. During the third and last phase, the review is written and validated.

The primary sources of this research consist of available criteria lists for the evaluation of RM tools, and features supporting RM activities in distributed SPL development.

The search for currently available criteria lists is performed on Google scholar. This source selection is founded in the need of finding lists that are not only used in academia, but that are available to the public in general. For the search of available criteria lists, the following search expression is used: ("requirements management tools" OR "RM tools")+(evaluation OR selection OR assessment OR survey)+(requirements OR criteria).

The features needed to support RM activities in distributed SPL engineering and management, are extracted only from revised research published in electronic databases. This search is carried out on the databases ieeeXplore, ScienceDirect and SpringerLink using different combinations of the strings "requirements management" or "RM tools", with "software product lines" or "distributed collaboration", or closely related terms. Related articles published before and after the ones obtained from the search, are also analyzed as recommended by Webster and Watson (2002).

The criteria lists and complementary features or requirements obtained from the literature review in the fields of SPL development and distributed collaboration of teams, are the basis for the second step of this research. This step consist of the creation of an evaluation framework for lists comparison, the actual comparison of the available lists using the framework, and the construction of a comprehensive criteria list for the evaluation of RM tools supporting SPL development in a distributed setting.

The DSR methodology process, presented by Peffers, Tuunanen, Rothemberger and Chatterjee (2007), is followed. This process includes six activities, namely problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication. The first two activities of this process were presented earlier in this introduction. The third activity consists in the creation of the artefact, in this case the evaluation framework to compare available criteria lists and the new criteria list for RM tool evaluation. The fourth activity demonstrates the use of the artefact to solve the presented problem. The fifth activity consists of the observation and measurement of how well the artefact solves the stated problem. (Peffers, etc., 2007.) To conduct this evaluation, a descriptive method will be applied to the developed artefact, using information from relevant research to justify its utility (Hevner etc., 2004). The final activity in this process consists of the communication of the research's resulting knowledge (Peffers, etc., 2007).

1.6 Structure of the thesis

The present research is organized as follows. Chapter 1 presents the background and motivation for the study, key concepts, research questions, limitations of the research, research method and the thesis' structure.

Chapter 2 reviews the literature concerning the RE discipline and its main activities, RD and RM. An overview of RE in the context of distributed collaboration and SPL development is presented as well.

Chapter 3 focuses on RM tools. Main concepts regarding RM tools will be introduced, their structure, classification and examples. Common features of RM tool will be described, followed by features supporting distributed collaboration and SPL development environments. A generic RM tool selection process will be presented in this chapter as well.

Chapter 4 describes the main criteria lists for RM tool evaluation taken into consideration in this research. Additionally, relevant published research and other criteria lists are described.

Chapter 5 presents the criteria defined for the analysis and comparison of currently available criteria lists for RM tool evaluation. These criteria consist of a set of attributes, used to evaluate the lists' presentation and organization aspects, as well as a set of categories and a scoring system, used to evaluate the lists' content.

Chapter 6 focuses on the analysis and evaluation of the criteria lists' attributes and content, using the criteria previously defined.

Chapter 7 presents the aspects taken into consideration for the construction of the new criteria list for the evaluation of tools supporting distributed SPL requirements management.

Chapter 8 presents the results obtained in this research by answering the research questions proposed.

Chapter 9 concludes the study, discussing the results obtained and presenting conclusions. A critical point of view on the results of this study and suggestions for future research will be presented in this chapter as well.

2 REQUIREMENTS ENGINEERING

RE is involved with the recognition of the goals and objectives to be accomplished by a system (Sadiq & Jain, 2012). It consists of a systematic approach in which requirements, in the form of functional requirements, quality attributes and constraints, are developed, specified, analyzed, verified, allocated, traced, and managed. During RE, an agreement is established and maintained between the customer and the project team on the changing requirements. (Ebert, 2012.)

According to Wiegers and Beatty (2013), requirement related activities need to be carried out regardless of the life cycle followed in the project. The authors divide RE into two main disciplines, namely RD and RM (see Figure 1).

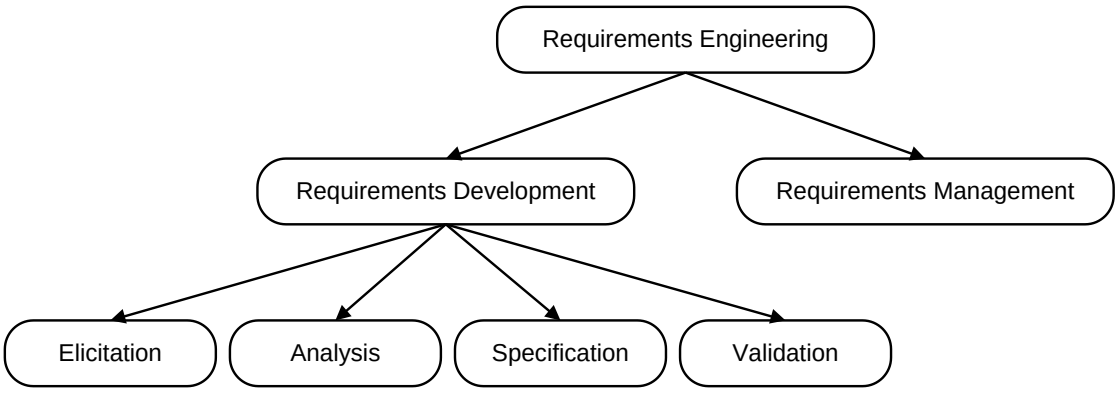

FIGURE 1 Sub-disciplines of requirements engineering (adapted from Wiegers et al., 2013)

The following sections, Section 2.1 and Section 2.2, present the main RE disciplines in more detail. Section 2.3 and Section 2.4 present RE in the context of global software development and SPL development respectively.

2.1 Requirements development

The objective of the RD process is the identification, agreement, and recording of requirements to satisfy defined business objectives (Wiegers, 2006). It can be divided into elicitation, analysis, specification and validation (see Figure 2). These are not one-time-only activities, but consist of multiple cycles to progressively refine the requirements until a sufficient level of detail and quality is reached. The RD activities can also be performed concurrently on different sets of project requirements (Wiegers, 2006.)

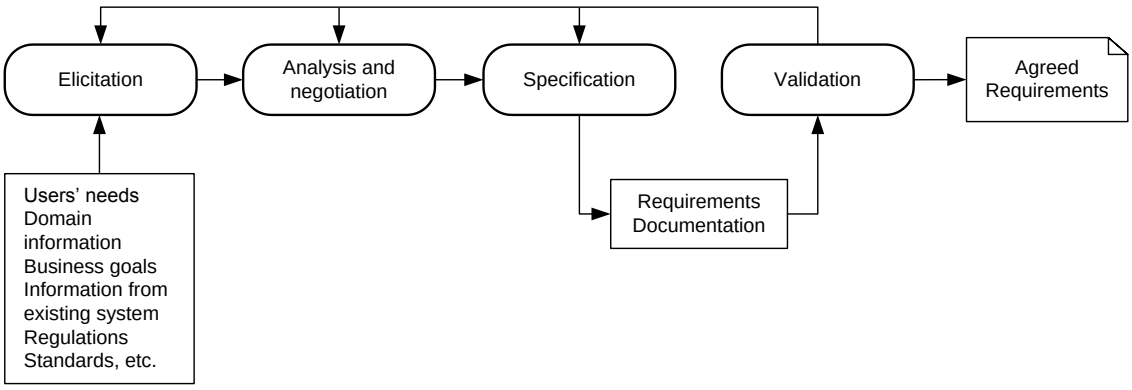

FIGURE 2 Requirements development activities (adapted from Kotonya et al., 1998)

Elicitation consists of the activities performed to understand the users and discover what their needs are (Wiegers, 2006). These requirements can be discovered from different sources, such as consultation with stakeholders, system documents, domain knowledge, and market studies. (Kotonya et al., 1998) In the case of requirements' elicitation from stakeholders, there are different techniques that can be used, e.g. interviews, brainstorming, requirements workshops, storyboards, etc. (Young, 2004).

The activities performed during analysis and negotiation aim at solving conflicts between the requirements discovered from different sources. Here, information that might be incomplete, or requirements that are incompatible with the project's budget, is identified. The requirements elicited in the previous RD activity are analyzed in detail during this task and stakeholders negotiate on which requirements will be accepted. (Kotonya et al., 1998.)

Specification consists of documenting the requirements information gathered in the previous activities in a way that enables communication between all project stakeholders. Traditionally, the resulting specification consists of documents including descriptions in natural language and diagrams, which can be stored in a database or in a RM Tool. (Wiegers, 2006.)

Validation ensures that the requirements are correct, satisfy the needs of the customer and are of high-quality (Wiegers, 2006). The goal of this activity is

to detect problems in the requirements before they are used for the system's development (Kotonya et al., 1998).

The result of the RD process is a set of agreed requirements that, after the necessary iterations through the RD activities, is believed to be good enough to allow the development of the next portion of the project. This set of requirements constitutes the requirements' baseline, which can be seen as a snapshot of the requirements at a certain point in time serving as a basis for additional work. (Wiegers et al., 2013.)

2.2 Requirements management

"A solid requirements management process is the key to a successful project" (Laplante, 2014, pp. 205). The goal of RM is to increase the probability that an application will meet its intended functionality and that its predicted value will be delivered to the business (Schwaber & Sterpe, 2007).

Managing requirements is an essential activity in every project, regardless of the methodology used. Effective RM keeps the project stakeholders informed about the status of the requirements during the whole development process, reducing the expectation gap. (Wiegers et al., 2013.) "It lets you know where you're headed, how the trip is going, and when you've arrived at your destination" (Wiegers et al., 2013, pp. 470).

RM activities start with the definition of a baseline including a set of agreed requirements obtained as a result of the RD process. Taking this baseline as a basis, schedule and cost commitments are made; therefore, it is necessary to establish a formal change process to control modifications to the requirements that could affect the already made commitments. (Wiegers, 2006.) According to Wiegers et al. (2013) the RM process can be synthesized in four core activities, namely Version control, Change control, Requirements status tracking, and Requirements tracing. These activities are depicted in Figure 3.

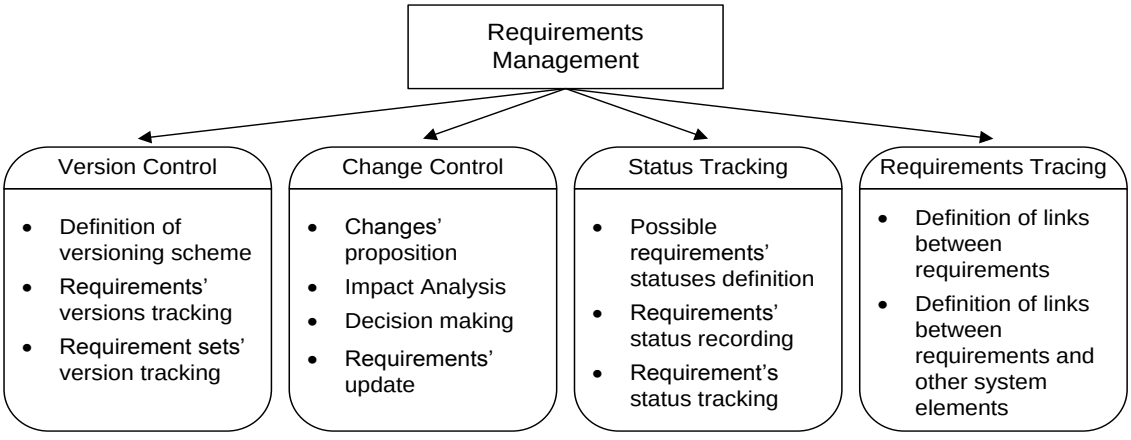

FIGURE 3 Major requirements management activities (based on Wiegers et al., 2013)

As presented in the figure, Version control consists of identifying in a unique fashion the different versions of an item, which can be a requirement or a set of requirements. This activity should start the moment a requirement is created, to save the complete history of changes made. Requirements' documents, as well as requirements in a RM Tool, should have a revision history including changes made, date of each change, who made the change and the reason why. (Wiegers, 2006.)

Change control consists of a series of activities to document, report, analyze, assess cost and implement changes. The main objective of this activity is to establish a formal process for collecting, verifying and assessing changes; as well as analyzing the impact of changes on the system as a whole. A general change management process can be divided in three stages. First, a problem in the requirement(s) is identified and analyzed, and changes to solve the problem are proposed. Then, the proposed changes are analyzed using traceability information to know how many requirements, and perhaps system components, would be affected by the change. Once the modifications that are actually needed are discovered, the customer is consulted about the changes, and the cost of the modifications is calculated. Finally, the changes to the requirements are implemented, if they were not rejected in the process. (Kotonya et al., 1998.)

Status tracking consists of monitoring the status attribute of the each requirement to have a clearer notion of what the progress of the project is. To achieve this, a set of meaningful requirements statuses must be defined, recorded and tracked. The classification of requirements into an appropriate set of statuses, allows a more meaningful analysis than just a percentage of completion. (Wiegers et al., 2013.)

Requirements tracing consists of creating logical links between requirements and other items in the project. Traceability links can connect a requirement with its origins, such as use cases, business rules, etc., or to other products that were created to satisfy the requirement, e.g. design elements and source code. Traceability information guarantees that there is no orphan code or neglected requirements in the system; ensuring that requirements relate to design elements, and these to source code, that test cases were defined for all requirements, etc. (Wiegers, 2006.) Using traceability presents several advantages, such as allowing the verification of requirements' implementation, the identification of gold-plated solutions in the system and in the requirements, the analysis of the impact of changes, the reuse of requirements in other projects, the assignment of development effort to requirements and the simplification of system maintenance (Pohl & Rupp, 2011).

2.3 RE in global software development

Nowadays, big and small companies develop software in a globally distributed setting (Schmid, 2014). While this trend continues to grow (Bosch et al., 2010),

RE activities need to be performed in a way that deal with distribution and internationalization aspects inherent to global software development (GSD). (Schmid, 2014). In this distributed scenario, RE becomes a challenge because of its need of intense collaboration and communication (Damian et al., 2012).

Several researchers have directed their attention to difficulties arising when people, working on RE, need to interact with a globally distributed development team. In this case, the goal is to guarantee that every person involved has equal understanding on the system to be developed and that this understanding correlates with the aims of the customer. (Schmid, 2014.)

According to Ebert (2012), requirement-related risks are augmented in global development projects due to delayed visibility of open issues, uncertainties, or misunderstandings. They state that these risks can be mitigated by specifying requirements and the understanding of requirements, storing requirements, guaranteeing adequate collaboration and communication, managing changes and designing change processes.

In their research, Damian and Zowghi (2003), identified four major RE challenges originating from geographical distribution of involved stakeholders, namely inadequate communication, knowledge management, cultural diversity and time difference. The first challenge, inadequate communication, refers to the obstacles introduced by distance to informal and face-to-face communications between stakeholders, making them dependent on the quality of synchronous or asynchronous electronic communication tools. Knowledge management, the second identified challenge, refers to inappropriate level of requirements' information sharing among distributed stakeholders. They observed that, when strategic requirements' information was transmitted through one key stakeholder to distant team members, distance was exploited to strengthen power positions in the organization. The third challenge, cultural diversity, refers not only to differences on the stakeholder's language and national culture, but also to differences regarding organisational and functional aspects. This diversity affects the obtainment of a common understanding on requirements and the promotion of negotiations around requirements. The last challenge, time difference, originates from the distribution of stakeholders across continents, affecting their synchronous collaboration. (Damian & Zowghi, 2003.)

Furthermore, the introduction of more flexible development methodologies, allowing requirements' changes to be introduced later in the development process, presents additional challenges when working in a distributed setting. These challenges are mainly related to collaboration and communication among sites, and can be faced by collaborative project management, collaborative RM, cultural awareness, informal and formal communication, and information sharing. (Heimbürger, Kiyoki & Ylikotila, 2011.)

Damian et al. (2012) focused their investigation on the effect a team's distribution has on three different types of awareness, namely general awareness, current awareness and availability. General awareness consists of

the knowledge a team member has about who has expertise in the project. Current awareness refers to knowledge about work assignation to other team members. Availability consists of knowing other team members' availability for information seeking purposes. They found out that even when distance does not seem to affect the awareness of team members working on the same requirements, it does create challenges in maintaining awareness of remote team members' accessibility for collaboration.

According to Sinha, Sengupta and Chandra (2006), the challenges presented due to geographic separation lead to increased requirement defects that go undetected until late stages of the project, when their fixing cost is too high. They affirm that this is due to the deficient social contact between the distributed teams, which could be improved by the utilization of tools with support for collaboration. According to Lang and Duggan (2001), collaboration between developers and users during the development of a system can improve the exchange of knowledge, help discover hidden requirements, enhance the communication of expectations, help with the early detection of errors, and improve planning and decision making. They also observed that the application of a collaborative tool supporting the RM process can improve communication and understanding among multidisciplinary users, and better project management and knowledge management activities.

2.4 RE in SPL development

A SPL allows a common platform to be shared by a product family, being product-specific functionalities built on top of that platform. Three concepts are important in SPL; namely variability, commonality and configuration. Commonality indicates what is common to all application in the SPL, variability shows where the differences between products lye, while configuration consists of the selection of variants for a specific application. (Bosch, 2013.)

"Software Product Line Engineering is a paradigm to develop software applications (…) using platforms and mass customization" (Pohl, Böckle & van der Linden, 2005, p. 14). The usage of platforms in applications development involves proactively planning reuse, building reusable parts, and actually reusing what was built to be reused. A software platform can be defined as "a set of software subsystems and interfaces that form a common structure for which a set of derivate products can be efficiently developed and produced" (Pohl et al., 2005, pp. 15). These software subsystems consist not only of code, but also of requirements, architecture, and other artefacts of the development process (Pohl et al., 2005).

The SPL paradigm involves two main processes, domain engineering and application engineering. During domain engineering the reusable platform is established, and the SPL's commonality and variability are defined and realised. During application engineering, applications are derived from the platform established during domain engineering, by reusing domain artefacts

and exploiting the variability of the SPL. This division of the paradigm in two processes ensures the building of a robust platform and the building of customer-specific application in shorter times. (Pohl et al., 2005.)

Domain engineering is divided in five sub-processes, namely product management, domain RE, domain design, domain realisation, and domain testing. The goal of these sub-processes is to detail and refine the variability established by the previous sub-process, as well as provide feedback to the previous sub-process about the feasibility of realising the required variability. (Pohl et al., 2005.)

Domain RE, the domain sub-process of interest to this research, comprises all activities related to the elicitation and documentation of common and variable SPL requirements. In other words, it consist of the elicitation of requirements from different sources as well as the identification of requirements that are common to all applications, and requirements that vary between the applications. The input of this process is the product roadmap, while its outputs are reusable requirements and the SPL variability model. (Pohl et al., 2005.)

Domain RE presents some differences with respect to RE for single systems. One key difference is that the domain RE analyses requirements to find those being common to all application and those being specific to particular applications. Another key difference lays on the explicit documentation of possible choices regarding requirements, in the form of a variability model. (Pohl et al., 2005.)

The second SPL main process, application engineering, is divided in four sub-processes, namely application RE, application design, application realisation, and application testing. Application RE, the application sub-process of interest to this research, comprises those activities in charge of developing the requirements specification of the application. A main concern of this sub-process is the detection of differences between application requirements and available capabilities of the platform. The inputs to this sub-process consist of the domain requirements, the product roadmap with the application major features, and sometimes additional specific requirements that were not captured during domain RE. The output of this sub-process is the requirements specification for the particular application. (Pohl et al., 2005.)

Application RE presents also differences with respect to RE for single systems. One of these differences is that the requirements are not elicited anew, but are derived from the domain requirements, taking in consideration the available commonality and variability. Another difference is the detection, evaluation and documentation of deltas between domain requirements and application requirements during elicitation activities. This is performed in order to know required adaptation costs and be able to make trade-off decision while reducing cost and increasing reusability of domain artefacts. (Pohl et al., 2005.)

SPLs are being used in a variety of domains, including automotive, finance, and telecommunication; serving as a proof of the actual usefulness of the concept (Poth, 2009). Bosch (2013) presents several reasons why a company

might adopt a SPL approach. One of the most common ones is the diversification of the product portfolio of the company, which allows the offering of a wider and richer set of products for the same R&D investment. Other less frequent reason originates in the need for more customizable products to satisfy customer needs, or the production of high-quality products using a reliable product core. (Bosch, 2013.) According to Pohl, Böckle and van der Linden (2005), the main motivations are the reduction of development costs, the enhancement of quality and the reduction of time to market. They identify additional motivations, such as the reduction of maintenance effort, improving cost estimation, organizing the evolution of products, managing the increasing complexity, and providing customers with products adapted to their needs.

In this chapter, main concepts related to RE, RD and RM, were introduced. Additionally, relevant aspects of RE in the context of GSD and SPL were presented.

3 REQUIREMENTS MANAGEMENT TOOLS

According to Kotonya et al. (1998), RM tools were created to provide a solution to the problems of managing the vast amounts of data collected during the RE process and the requirements' volatility. In today's constantly changing business environment, RM tools have an essential role in understanding the impact of requirements' change and in enforcing the processes around it (Schwaber et al., 2007).

As development effort widens and requirements become more detailed, the support of RM tools becomes more and more important as a mean to make proper RM activities cost-effective. (Schwaber et al., 2007.) Generally, the need for RM tools increases as projects expand, become more critical to the business, include more than one product, or involve other companies (Alexander & Beus-Dukic, 2009).

The usage of an automated requirements tool was included by Young (2004) in his list of best practices for requirements development and management. Here, he remarks the importance of a requirements tool supporting the company's process and that of the training imparted to those people most using the tool.

Gotel and Mäder (2009b) present a list of situations where the support of a RM tool becomes necessary. They identified the needs to distribute, share and align the project's work between more than one person, localization or organization; the need to scale, due to the big amount of requirements and/or multiple releases; the need to diversify and reuse, as in the development of product families; the need to know, to improve quality and to collect metrics; the need to associate different engineering steps to obtain a product from the requirements; the need to alleviate staff in charge of repetitive and administrative tasks; the need to demonstrate, due to legal or contractual reasons; and the need to maintain, when projects are expected to have a long life and many customers.

Nowadays, a variety of tools is available providing distinct levels of support to RM activities. These tools can be classified in General Purpose, Special Purpose and Life-cycle Tools (see Figure 4).

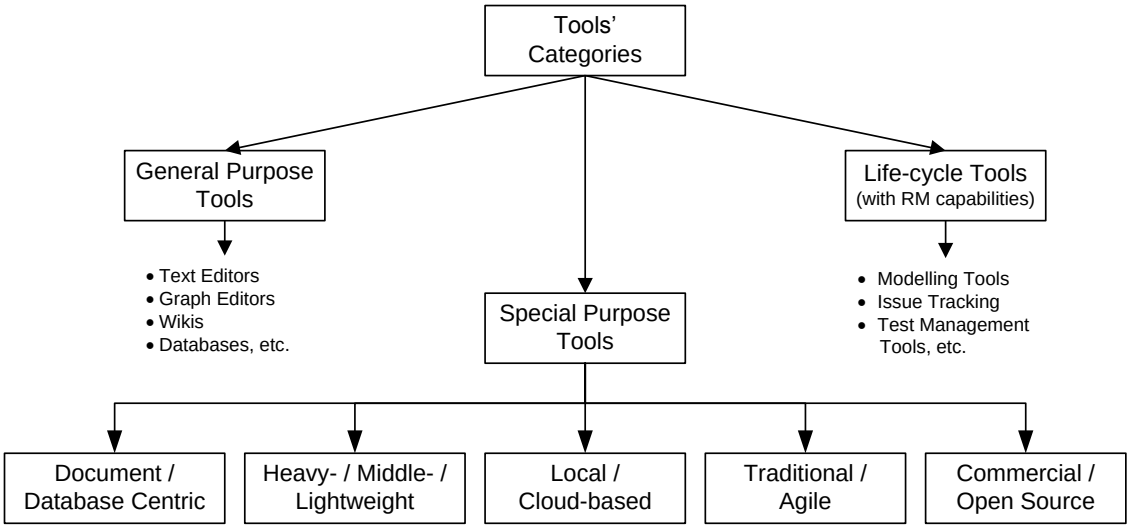

FIGURE 4 Categorization of tools supporting RM activities

General purpose tools, such as text editors or spreadsheets, can be configured to support RM activities previously carried out manually, or in a paper-based way. This kind of tools are flexible, widely available and already familiar to people involved in the requirements' process. However, they are difficult to maintain and do not provide some basic RM functionality, making them suitable only for small projects. (Gotel & Mäder, 2009b.) According to Pohl and Rupp (2011), such tools do not allow the team to obtain the levels of productivity and performance that can be achieved by specialized tools.

Special purpose tools, on the other hand, are those created specially to support the RM activities, fully or partly (Gotel & Mäder, 2009b). These RM tools can be further differentiated into those being mainly database centric or document centric; heavyweight, middleweight or lightweight; local or cloud-based; supporting traditional or agile methodology; and commercial or open source. These categories are not exclusive, there are several tools aiming at improving their offer by providing a combined option. Such is the case of RequisitePro being both document- and database-centric; and TraceCloud providing support for the more traditional oriented iterative-waterfall methodology, as well as for scrum.

Finally, life-cycle tools are those offering support for all or some phases of a system's development. Generally, these tools are not specialized in RM, but provide means to perform RM activities. The RM functionality covered by these tools varies greatly; while some nearly include the functionality of a special purpose tool, others do not actually support RM main tasks. (Gotel & Mäder, 2009b.)

RM tools available in the market have a similar architecture. This architecture is presented in Figure 5, and consists of a user interface, a database, and capabilities for import, export, and report generation. The user interface of the tool allows the user to perform all RM activities. The information managed

by the tool is stored in the database and can be modified with the help of an integrated editor. Import capabilities allow the introduction of data generated elsewhere into the tool, while exporting capabilities guarantee the generation of data in formats that are usable to other tools. Finally, reporting capabilities support the generation of reports and views of the information stored in the tool. (Pohl & Rupp, 2011).

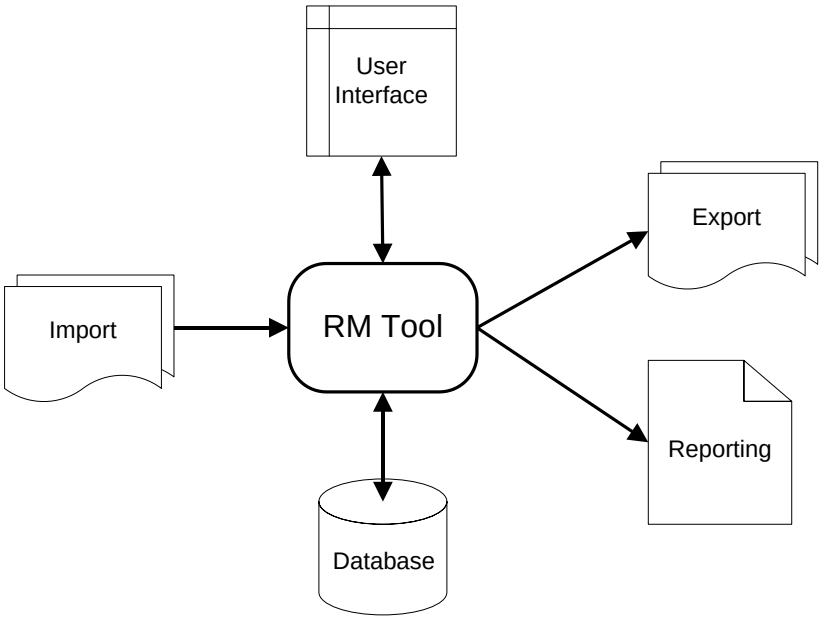

FIGURE 5 Basic RM tool's architecture (adapted from Gotel & Mäder, 2009b)

RM tools are utilized by a variety of users, including requirements engineer, requirements engineer head, software engineer and project manager. The requirements engineer is in charge of defining the projects' requirements and their relationships. The requirements engineer head determines the rationale of the project's requirements. The software engineer develops a system taking as a reference the requirements defined for the project. Finally, the project manager is the user in charge of defining and managing project and users. (Carvallo et al., 2005.)

Between the commercial RM tools available in the market, DOORS (Dynamic Object Oriented Requirements System) is one of the most popular, having thousands of users around the globe (Hull et al., 2011). This is a heavyweight, local, database centric RM tool supporting traditional oriented RM activities.

Nowadays, the offering of RM tools is substantial, varied and under permanent development. New RM tools enter the RM tool market constantly, while others leave it quietly. However, some trends can be identified from this changing market place, e.g. the availability of free trials, the lowering of prices

for web-based solutions, and the inclusion of visualization and collaboration capabilities. (Alexander, 2014). As a result of this fast evolving marketplace, the practitioner in need of a tool has a wide range to choose from, as well as a big challenge to select the appropriate tool to answer his own needs.

A number of listings including RM tools are available online (see Table 1). The amount of information on the tools varies from list to list, as well as the frequency of revision of the list.

TABLE 1 Listings of RM tools

Name	Source	Year	Tools	RM Tools
The Making of Software	www.makingofsoftware.com/resources/list-of-rm-tools	2014	112	83
Seilevel	www.seilevel.com/wp-content/uploads/Requirements-Gathering-Management-Tool-Evaluation-Worksheet.xlsx	2011	108	60
Ian Alexander	www.scenarioplus.org.uk/vendors.htm	2014	83	59
Volere	www.volere.co.uk/tools.htm	2014	79	51
INCOSE	www.incose.org/productspubs/products/rmsurvey.aspx	2010	34	34

As it can be observed in the table above, the website "The Making of Software" presents the most extensive list of RM tools. The first version from this list dates from 2011 and has been fully revised and updated in February 2014, making it the most updated between the available lists. Tools in the list are assigned different RM-related categories to indicate the tool's scope, such as RM, requirements definition, visual modelling, etc. The details displayed for each tool include its name, consisting of a link to the tool's webpage, its last version, the tool vendor and the scope of the tool. (Birk & Heller, 2014.)

Seilevel's list of tools is part of Seilevel's RM Tool evaluation; described in Chapter 4. The tools included in this list are classified under requirements definition, mock-ups, RM and agile specific tools. The information provided by this list includes the tool's name, vendor, type and the address of the tool's website. A list of the tools that were not considered, and the reason why, is provided as well. (Seilevel, 2011a.)

The list of tools created by Ian Alexander includes information about the name of the tool and its provider, followed by a description of the tool. These descriptions are provided by the tool vendors themselves or are written by third parties. Information on the creation or update dates is not displayed in this site. (Alexander, 2014.) This list was used as a starting point by Jones (2011) for their RM tool selection process.

The list of tools published by Volere consists of a series of tools' descriptions. The list does not apply categories to the tools and no explanations of the source of the tools' descriptions are provided. The information provided on the tools consists of the tool's name, the vendor's name, the tool's website and a description that, in most of the cases, lists the main features of the tool.

For some tools the date of creation or update of the entry is displayed. (Volere, 2014.)

The INCOSE Tools Database contains the results of the survey conducted by INCOSE Requirements Working Group. This database consist of a survey containing a list of requirements for RM tools, explained further in Chapter 4, and the answers to the survey provided by 34 tool vendors. Half of the answers date from 2008, while the other are from 2009 and 2010. The answers to the survey include detailed information on each tool. (INCOSE, 2010.) The INCOSE Survey consists of the most referenced list of RM tools in the consulted literature (Beatty, 2013; Violante & Vezzetti, 2014; Gotel & Mäder, 2009a).

The previously presented listings of RM tools contain valuable information for a fast identification of currently available tools. However, most of them include only limited information about the tools, making their usage for a detailed evaluation of the tools, and later selection, unfeasible.

Another source of information regarding RM tools and their capabilities, in the form of comparisons of available tools, can be found in the literature (Alghazzawi, Siddiqui, Bokhari & Hamatta, 2014; Zainol & Mansoor, 2011; Seilevel, 2011b; INCOSE, 2010; Vrinat, 2010; Beuche et al., 2007; Bokhorst, 2001; Wiegers, 1999). These comparisons are carried out under different sets of criteria, and provide a way to quickly identify certain differences between the considered tools. Between these comparison documents, the most detailed are those developed by Seilevel (2011b) and INCOSE (2010). However, while the information in Seilevel's document originates from the results of an actual evaluation, the answers to INCOSE's survey were provided by the tool providers themselves (Hoffmann et al., 2004a; Beatty, 2013).

3.1 Tool capabilities

In this section, features commonly provided by RM tools are introduced, together with features supporting SPL development and distributed collaboration.

3.1.1 Common features

Tools developed specifically for RM include common features, such as version and change management, storage of requirement's attributes, impact analysis, identification of missing and irrelevant requirements, requirement's status tracking, access control, facilitation of stakeholders communication, requirements reuse, tracking of issue's status and report generation. Wiegers and Beatty (2013) group these features under the categories Capture requirements, Manage requirements, Trace requirements, Output requirements and Access tool help (see Figure 6).

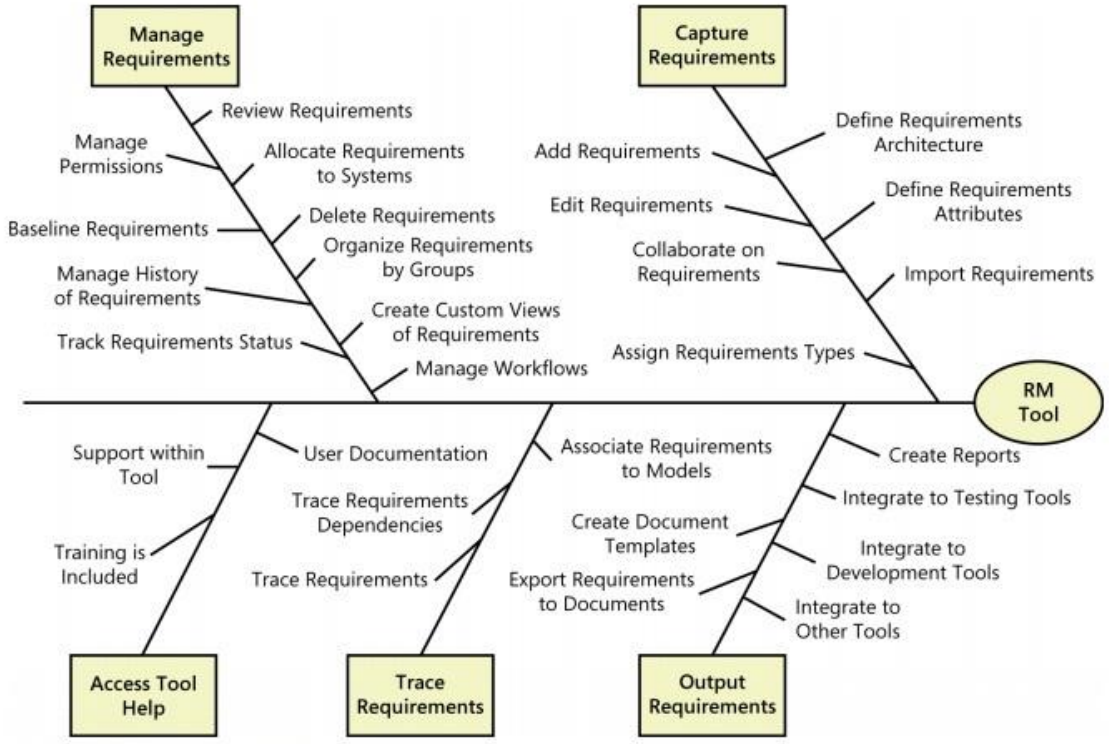

FIGURE 6 Common RM tools features (Wiegers et al., 2013)

As it can be observed in the figure, the feature set named "Capture requirements" groups those characteristics related to the introduction of requirements into the tool, as well as the definition of requirement' attributes, types and architecture. The feature set "Manage requirements" includes characteristics supporting the actual management of the requirements. "Trace requirements" consists of the creation and maintenance of links between requirements and other objects in the tool. "Output requirements" covers the integration of the RM Tool to other tools, such as development tools, testing tools, modelling tools, etc, as well as the generation of reports and the exporting of requirements. Finally, "Access tool help" groups the support provided by the RM tool vendor to the customer, in the form of user manuals, training, etc. (Wiegers & Beatty, 2013.)

3.1.2 Features supporting distributed collaboration

According to Sinha et al. (2006), available RM tools do not offer sufficient support to stakeholders, nor allow effective collaboration on the project's requirements. In their research, they identified challenges of distributed RM, defined requirements for a RM tool capable of addressing these challenges, and constructed a RM tool based on the defined requirements. These requirements

were grouped into four areas, namely "Informal collaboration support", "Change management", "Awareness promotion", and "Knowledge management". "Informal collaboration support" enables ad-hoc discussions around requirements. "Change management", a crucial part of a RM solution, includes the ability to raise changes on requirements, identify and notify the impacted requirement's stakeholders, and record the following actions performed on the requirement. Thanks to these two capabilities, all information related to the requirements is stored in the tool and can be accessed by new team members, reducing the learning time needed. "Awareness promotion" provides information and cues about team members and the activities performed by them in order to facilitate collaboration. Examples of these awareness mechanisms are cues indicating requirements that were recently focus of a discussion, recent assignation of tasks to a team member, or the status of a team member indicating his availability. "Knowledge management" allows the navigation, management and understanding of the project's content. These two capabilities agree with the different types of awareness considered crucial in distributed teams by Damian et al. (2012), which were presented earlier.

Portillo-Rodríguez, Vizcaíno, Piattini and Beecham (2012) evaluated tools supporting Global Software Development, between which six requirement tools are found. As a result of their tools' evaluation they concluded that the most frequent feature in requirements tools consists of the "Issue tracking system", which keeps the tool users aware of changes and issues. They observed that visual awareness techniques are generally included by the tools to complement the "Issue tracking system", allowing the insertion of comments or by highlighting changes.

Smite (2006) identified main practices to successfully manage requirements in a distributed setting. Between these practices, there are some that can be supported by RM tools, such as the usage of a common requirements template, the development of a glossary, the implementation of version control, the establishment of clear responsibilities and priorities, and the support of continual communication.

3.1.3 Features supporting SPL development

Requirements for SPLs are, in general, more complex, longer lived, and possess more critical dependencies than requirements for single-products. RM tools are necessary to control and categorize requirements, and to track changes in them. Until today, almost all available RM tools focus on single-product development, leaving key aspects of product-line development, such as variability management and systematic reuse, unsupported. However, these tools are used successfully in SPL settings by the implementation of workarounds, such as customization and extensions to the tool or changes in the processes around the tool. Unfortunately, these workarounds demand big amounts of effort, resulting too expensive for smaller companies in the need of full support by one tool. (Beuche et al., 2007.)

A variety of tools and extensions, mainly to the RM tool DOORS, have been developed by academia and industry to support requirement related activities in SPL settings. Some examples of these include pure::variants (Beuche, 2008), metadoc (Thurimella & Janzen, 2011), PLUSS (Eriksson, Börstler & Borg, 2009), RequiLine (Von der Maßen & Lichter, 2004), and REMAP (Schmid, Krennrich, & Eisenbarth, 2006). However, many of the developed extensions have been created and used only in academia or in industrial case studies, and are not commercially available. (Lisboa et al., 2010.)

Requirements for RM tools supporting SPL development have been defined in the literature. These requirements can be found as a description of criteria against which developed tools are later evaluated (Beuche et al., 2007; Djebbi et al., 2007; Lisboa et al., 2011), as requirements to be fulfilled by a newly developed tool and the actual features included in the tool (Eriksson et al., 2009; von der Maßen et al., 2004; Schmid et al., 2006; Thurimella et al., 2011), or as a stand-alone list of requirements for RM tools used in SPL development (Salo et al., 2005).

In their article, Schmid, Krennrich, and Eisenbarth (2006), present a list of requirements to be satisfied by a RM tool used for SPLs and the implementation of these requirements by their tool extension, REMAP. They present a set of eight high level requirements, derived from SPL concepts and existing RM tools. The first requirement consists of the ability to divide the requirements base into a SPL infrastructure and products, allowing the definition of reusable requirements, requirements containing variability and the explicit management of individual products. The second requirement consists of the ability to handle commonalities, variabilities and product-specific requirements for each product; these three types of information must be managed individually. The third requirement consists of the ability to represent variability in the SPL infrastructure, allowing explicit descriptions of the products' variability and the management of relationships between variabilities. The fourth requirement refers to the support of a decision model, including the creation and evolution of the decision model, and the management of the relations between the decision model and other requirements. The fifth requirement refers to the instantiation support provided by the tool, which should allow the derivation of a product instance from the SPL infrastructure. The sixth requirement consists of the ability to modify instantiations, allowing their evolution in time. The seventh requirement refers to the traceability support provided by the tool, which should manage SPL specific traces along with traditional traceability capabilities. These SPL specific traces consist of the relation between requirements in instantiated products and the requirements in the SPL infrastructure, from which the products were derived. The eighth and last requirement refers to the versioning support provided by the tool, which should manage the SPL versions, taking into account the status of the decision model, traces, etc. (Schmid et al., 2006.)

Lisboa et al. (2010) conducted a review of available domain analysis tools in order to find out how they support the domain analysis process. In contrast

with other research, the features taken into account to compare the tools were first extracted from the tools and then compared, instead of being defined by the researchers before hand. The identified features were divided in the categories planning, modelling, validation and extra functionalities. Within the modelling functionalities, they defined a set of common functionality supported by almost all analyzed tools, namely "Domain representation", "Variability", "Mandatory features", and "Composition rules". The feature "Domain representation" consists of ways provided by the tool to represent the defined scope, e.g. by tables, models, trees and others, in order to formalize the variations in the domain. "Variability" provides means to represent the possible variations of a product feature. "Mandatory features" provides a way to indicate what features must be present in the product. Finally, "Composition rules" allow the creation of restrictions in the domain, regarding the representation and relation of features. (Lisboa et al., 2010.)

3.2 Tool selection process

This section presents the process to be followed when selecting a RM tool. The main inputs of this process are introduced as well, giving special attention to the criteria list used for the actual tool evaluation.

The selection and acquisition of a RM tool is not an easy task; according to Gotel and Mäder (2009a, p.1) "it can be one of the most costly decisions an organization can make". The initial costs of procuring the tool can be insignificant in comparison to the impact of an ill-suited RM tool on the organization's software development and maintenance efforts (Gotel et al., 2009a). However, as Jones (2011) states in his article, the selection and purchase of a tool can be accomplished in challenging timeframes, when the team acquiring it has a good knowledge of their own processes. In their experience, one of the tools with the lowest cost, satisfied their needs better than other more expensive tools, letting them also conclude that a higher cost does not mean a tool's superiority.

In order to choose the right tool, the individuals running the selection process must take control of it and drive it to realization, giving special consideration to the resources available for tool acquisition, use and administration (Jones, 2011). According to Pohl and Rupp (2011), the selection of a RM tool cannot take place before the necessary techniques and processes to achieve the goal of RE activities are defined, and responsibilities have been assigned to the people involved in the process.

As seen in the previous section, today's offering of RM tool is extensive. This makes the process of analysing the available tools and selecting the right tool to the given circumstances more difficult, therefore the usage of a criteria list is necessary for a complete evaluation of the tools. These lists can reduce the risk of leaving out of consideration features or tool characteristic that can be critical to carry out RM activities.

According to Wiegers and Beatty (2013), the process to select a RM tool should be led by business analysts, by defining the evaluation criteria and carrying out the assessment. The goal of this process is to select a tool that best fits the development environment and culture of the company, based on desired functionality, platform and pricing (Wiegers et al., 2013). Several approaches to the selection process of a RM tool can be found in the literature (Wiegers et al., 2013; Gotel & Mäder, 2009b; Jones, 2011; Beatty, 2013; Heindl, Reinisch, Biffl & Egyed, 2006; Carvallo et al., 2005). The process depicted in Figure 7 is based on the process presented by Wiegers and Beatty (2013).

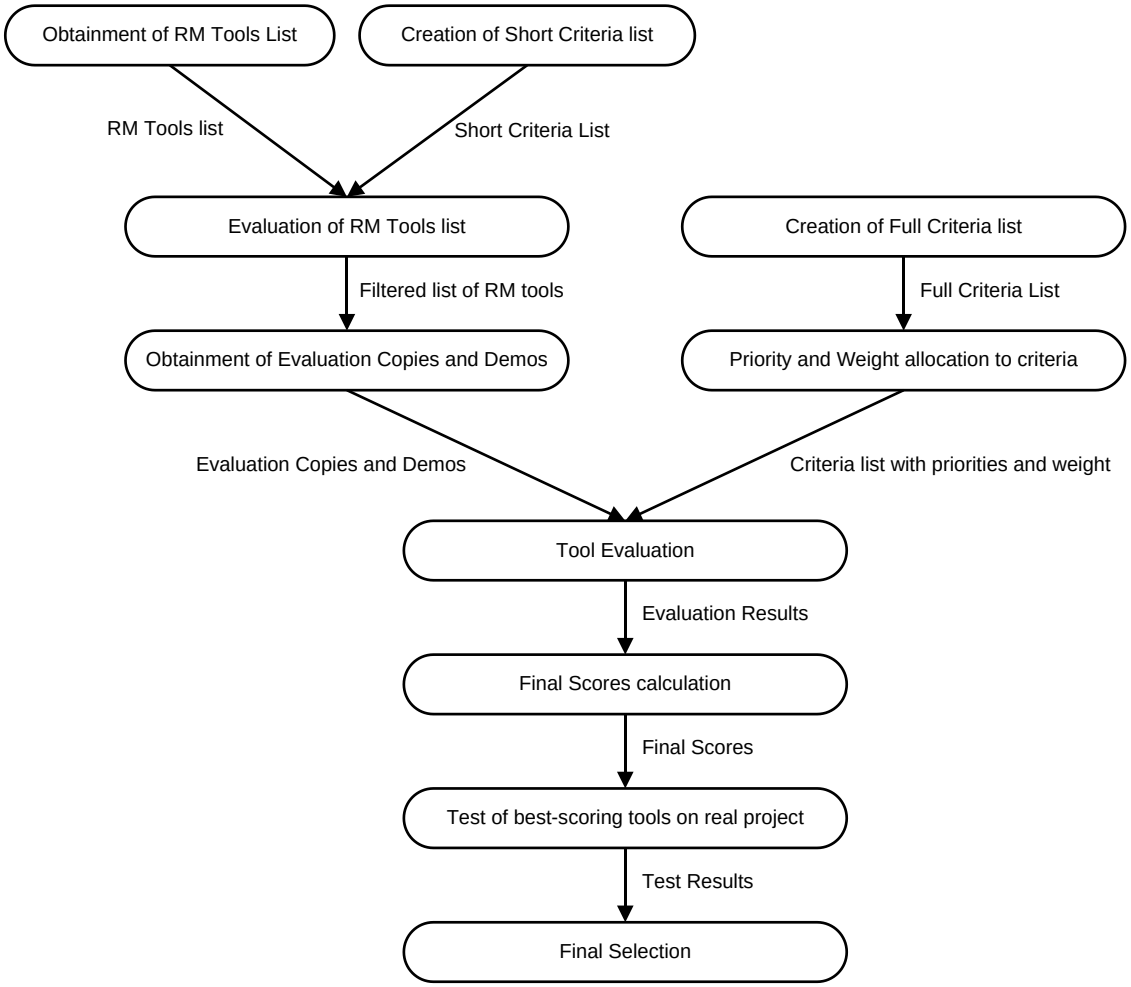

FIGURE 7 RM tool selection process

As observed in the figure, the activities preceding the actual evaluation of the RM tools can be divided into two main courses of action to be conducted concurrently. This division was introduced in the figure as a way to clearly show that the company or team selecting the RM tool should do so based on their specific needs (Carvallo et al., 2004; Young, 2004). However, these activities can be conducted consecutively as described by Wiegers et al. (2013).

The results of the two initial sets of activities consist of evaluation copies and demos of a reduced set of available RM tools, and a prioritized criteria list to fully evaluate the previous demos.

To produce the set of demos mentioned above, the team in charge of the selection needs to obtain a list of currently available RM tools, such as the listings presented earlier in this chapter. At the same time, a short criteria list should be created to make a first selection of the RM tools in the listing. The short criteria list should include one or more characteristics that would allow the team to filter the initial tools' list, into a more manage amount of tool for final evaluation. To produce a short list of tools Jones (2011)'s used a very simple approach that consisted in giving the tool's website five minutes to describe the tool's capabilities in some way. As a result, they reduced the list of tools from 71 tools to only 6. Once the list of tools has been filtered, the selection team should request evaluation copies or setup demos of the tools that passed the first evaluation.

To obtain a prioritized criteria list, a full criteria list should first be created. Then, priorities and weights should be assigned to each requirement according to the importance it has to the team or organization acquiring the tool. (Wiegers et al., 2013.)

The tool evaluation will take place once the evaluation copies or demos of the selected RM tools are available and the full criteria list is complete and prioritized. The person in charge of the evaluation should score each tool consistently against all requirements in the criteria list. Final scores for each tool will be calculated using the criteria score obtained and the weights assigned to the criteria. Tools that obtained the best scores should be then tested in real project environments to observe if they behave as anticipated from the scores. (Wiegers et al., 2013.)

Finally, a last selection will take place based on the combination of the tool's score, licensing cost, other ongoing costs, vendor support, current users input, and the subjective opinion of the team members that evaluated the tool (Wiegers et al., 2013).

As it can be observed in the figure presented before, a full criteria list is one of the main inputs of the actual RM tool selection phase. According to Young (2004), real project needs should be considered, avoiding arbitrary criteria, such as monetary constraints, personal preferences or preordained tools. The criteria list for RM tools should therefore consist of a catalogue of the needs of the group acquiring the tool. These needs are not only in the form of tool features, but also quality requirements, e.g. security, portability or scalability, support provided by the tool vendor, such as training and user manuals, costs involved in the tool's purchase, use and maintenance, etcetera.

The construction of a criteria list is a costly and time-consuming activity (Carvallo et al., 2005). A variety of criteria lists for RM tool selection, originating from industry experiences, are publicly available nowadays. They consist mainly of textual descriptions of requirements that guide the selection process through a series of features that need to be taken into account. The features

generally covered by the lists consist of the main RM activities, other support activities, such as import functions, reporting, users and permissions, etc., and general system specifications. Ready evaluation templates, derived from two available requirements' lists, are at disposal online and can be used in the selection process. It was however recommended to adjust the priorities assigned to the requirements according to the needs of the group or organization acquiring the tool (Seilevel, 2011a; Hoffmann et al., 2004b).

Furthermore, criteria lists are not only helpful for the selection and purchase requirements tools, they also allow individuals involved in requirement activities to better understand the relationship between requirements activities and tool capabilities, and to identify activities or processes that can be improved by the usage of a requirements tool (ISO/IEC TR 24766, 2009). Besides, criteria lists can guide requirements tool vendors in the development of future tools (Hoffmann et al., 2004a).

In conclusion, this chapter presented an overview of RM tools and their capabilities, in terms of common features and features supporting distributed collaboration and SPL development. Then, the process followed to evaluate and select RM tools was introduced. The concept of a criteria list was presented in this section as well, and its role and importance as an input of the selection process was described.

4 CRITERIA LISTS FOR REQUIREMENTS MANAGEMENT TOOLS

This chapter introduces an overview of available criteria lists for the selection of requirement tool obtained as a result of the literature review. These results were divided into criteria lists for the evaluation of RM tool published in its totality and others criteria lists.

A total of six full criteria list were obtained from the literature review. The main information of these criteria lists is summarized in Table 2. The following six subsections present these criteria lists in more detail, in their order of creation.

TABLE 2 Criteria lists for RM tool evaluation

	List 1	**List 2**	**List 3**	**List 4**	**List 5**	**List 6**
Publisher	INCOSE	DSTO	ITEA DESS	Daimler-Chrysler	RWTH Aachen	Seilevel
Creation year	1990'?	1997	2001	2004	2006	2011
Categories	13	11	18	21	24	23
Requirements	68	149	78	101	110	230
Publicly available	yes	yes	yes	yes	yes	yes
Evaluation template	no	no	no	yes	no	yes

The last subsection in this chapter describes shortly other criteria lists left out of consideration due to their partial availability, and research results, both of interest for the comparison of criteria lists in terms of their content.

4.1 List 1: INCOSE

The INCOSE Tools Database (INCOSE, 2010) was created by the INCOSE Requirements Working Group in the late 1990's and has been continually updated since then. It is an excel document composed by a survey and the answers given to this survey by different RM tool vendors. This database allows a direct comparison of tools against the defined set of features covered by the questions of the survey.

The database document includes a first page with the survey's last date of update, its background and related information, and the survey's categories. The second page shows the survey questions and the consolidated results for all covered tools, in the form of a table. Additional 34 pages show the detailed responses provided by the tool vendors. The survey does not include priorities or levels of importance assigned to the questions.

The results of this survey are normally at disposition in the official INCOSE website, but currently they are unavailable. However, the last available version, which dates from 2010, can be found in the internet. It includes a list of 68 requirements divided in 14 categories, and answers from 34 RM tool vendors. All requirements in the list are uniquely identified by a code, which consists of the category's number and the position of the requirement in the category. Some requirements have sub-requirements questioning the main requirement in more detail. Most requirements in the list consist of a main feature, followed by explanatory text, either in the form of a statement or a question. The answers provided by the tool vendors are composed by the level of compliance of the tool, which can be full, partial or none. The possibility of adding additional comments is also provided to the tool vendor.

This survey has been used as a reference for the creation of other criteria lists for RM tool selection (Hoffmann et al., 2004a; Heindl et al., 2006; Beatty, 2013; Violante et al., 2014).

4.2 List 2: DSTO

This RM tools evaluation guideline was created in 1997 by researchers in the Australian Defence Science and Technology Organisation (DSTO). The goal of this document is to serve as a basis for systematic assessment of possible tools to be adopted by the Australian Defence Organisation (ADO). (Gabb, Maheswaran & Allwright, 1997.)

This document is publicly available and is composed of two main parts. The first part consists of an introduction, a description of the reason why requirement tools are needed, an overview of the development of requirements specifications at ADO, the environment and users of the tool and other assumptions. This is followed by scenarios, cases, viewpoints and needs of the users of the tools in terms of functionality. The second part of the document

consists of an introduction to the section, a description of the nomenclature used throughout the document and the priorities' scheme assigned to the requirements. Finally, the list of requirements for RM tools are described, derived from the user needs identified in the previous part. (Gabb et al., 1997.)

The list of requirements presented in this document follows a hierarchical organization. A requirement in the list can be uniquely identified by the numbers assigned to the series of categories the requirement is included in, and the position of the requirement in the last category. Most requirements are formulated as questions, some of which are followed by possible options or an explanation. Requirements are prioritized by the values "[P1]", "[P2]", "[P3]", and "[P4]", indicating whether the requirement is critical, important, relevant or useful.

The list contains a total of 149 requirements organized in a three-level hierarchy of categories. This hierarchy consists of 11 meta-categories divided in 35 categories, four of which contain sub-categories. The meta-categories of this list are: Requirements data; Requirements input, Output, Generation and compatibility; Management of requirements; Specification generation; Reporting; Usability; Functions; Performance issues; Privileges and access; Tool administration; and Miscellaneous. (Gabb et al., 1997.)

This criteria list does not present similar requirements or references to the previous list, INCOSE (2010).

4.3 List 3: ITEA DESS

The ITEA DESS RM Tools Requirements was published at the end of 2001 by the Information Technology for European Advancement – Software Development Process for Real-Time Embedded Software Systems (ITEA DESS) project. The main objective of this project was the definition of a software development methodology for real-time and embedded systems. Therefore, this list of requirements describes RM tool features that support the ITEA DESS methodology. (Bokhorst, 2001.)

This document is publicly available and includes not only the list of requirements, but also a section named "Tool Overview" with a comparison of four commercial RM tools against 24 criteria. These criteria cover some of the requirements included in the list of requirements, but not all.

This criteria list includes approximately 78 requirements, divided in 18 main categories. These categories are: Requirements elicitation and identification, Batch-mode document/source-link update, Supports components based development, Traceability, Traceability analysis, Change control, Configuration management, User identification and access control, Documenting and Reporting ability, Multi-level access control, Multi-site multi-user support, Integration to other tools, System requirements, Support for testability, Compliance to standards, Customizing ability, and Quality. Some of these categories contain sub-categories. (Bokhorst, 2001.)

The presentation of this list of requirements is not uniform. Some categories or sub-categories have a numbered list of requirements, while others present a bulleted list, a simple paragraph or a table of requirements. The requirements are not uniquely identified.

This list references the one created by INCOSE (2010) and includes a couple of identical requirements.

4.4 List 4: DaimlerChrysler

The criteria list Requirements for RM Tools was developed by Hoffmann, Kühn, Weber and Bittner (2004a) at DaimlerChrysler AG. The requirements included in this list are based on extensive project experience in the areas of automotive, aircraft and defence systems (Weber et al., 2004a). No references to other criteria lists are included in this document, or requirements that are identical to those included in previously presented lists.

This list of requirements follows a hierarchical organization, with three levels of hierarchy. The first level is a grouping by stakeholder, the second by criteria and the third level contains the RM tool's requirements. All requirements in the list are uniquely identified by a code, which consists of the stakeholder's number, the number of the criteria and the position of the requirement in the criteria group. The requirements, as well as the criteria, are prioritized by the values "(++)", "(+)" and "(-)", representing high, medium and low priority respectively. For each criteria group, a description of the rationale is provided.

The list contains a total of 101 requirements divided in 21 criteria, which are assigned to three different stakeholders: developers, project administrators and tool administrators. The requirements from the developers' point of view include the core functionality of the RM tool, under the criteria: Information model, Views, Formatting, Multimedia and external files, Documentation of history, Change management and comments, Baselining, Traceability, Analysis functions, Tool integration, Import, Document generation, Collaborative work on same development task, Checking out for offline use and Web access. The requirements from the project administrators' point of view cover issues that essential for larger projects and are grouped under the criteria: Central installation and administration of projects, Users, roles and rights, Size restrictions, Workflow management and Extensibility. Finally the requirements from the system administrators' point of view are divided in the criteria Database and Encryption only, focusing on reliability and data security.

A tabular evaluation template, based on this list of requirements, is publicly available in the internet (Hoffmann et al., 2004b) and can be directly used for the evaluation of RM tools.

4.5 List 5: RWTH Aachen

The list of requirements for RM Tools in the context of SPLs engineering was developed by the GI Work Group of the Aachen University of Technology. This research included not only the creation of the requirements' list, but also the evaluation against these requirements of four available tools for the management of requirements in SPLs settings. (Beuche et al., 2006.)

This list of requirements was developed taking as a basis the requirements catalogue created by Hoffmann et al. (2004a) with the objective of including requirements for the evaluation of tool features required in SPL developments. The new requirements added to the catalogue were derived from real tool usage scenarios of members of the research group. The changes performed to the original list were not extensive; only specific changes were applied preserving the hierarchical organization of the original list and its general style. Requirements from the original catalogue were revised, reorganized and modified, and new requirements were added. Some of the existing requirements were reformulated in order to clarify their meaning, without changing their intention. (Beuche et al., 2007.) According to Beuche et al. (2007), no requirements were removed from the original list; however, few requirements are not present in this list. Additionally, the categories Configuration management, Collaborative work and Priorities were incorporated, containing existing and/or new requirements. Finally, the priorities assigned to requirements and categories in the original requirements' catalogue were preserved, but new requirements were not prioritized using the same scheme applied in the original criteria list. Instead, a system of points and priorities, showing the importance of the requirement in relation to SPLs, was assigned to all requirements in the catalogue. (Beuche et al., 2007.)

This catalogue of requirements for SPLs has a total of 110 requirements divided in 24 categories, presented in the form of a table. As in the original list, the requirements are uniquely identified by a code consisting of the stakeholder's number, the number of the category and the position of the requirement in the category group.

4.6 List 6: Seilevel

The Seilevel's RM Tools Evaluation Worksheet (Seilevel, 2011a) is one of the results of the RM Tool evaluation conducted by Seilevel in 2011. The documentation available about the process undertaken by Seilevel includes the document previously mentioned and the RM Tools Evaluation Results (Seilevel, 2011b). The list of requirements included in these documents is the most recent list available and also the most extensive one, in terms of amount and level of detail of the requirements.

The RM Tool evaluation worksheet contains a prioritized list of use cases, a prioritized list of requirements called "Tool Criteria", a list of initial tools that would be later subject to evaluation, and a list of features grouped by use case. This evaluation worksheet can be directly used for the evaluation of RM tools.

The list of use cases includes a set of 50 use cases of a RM tool identified by Seilevel. The actor in the majority of the use cases is the business analyst, being managers, developers, and business users the actors of a smaller portion of use cases. The priorities assigned to the use cases are 1, 2 and 3, representing low, medium and high priority respectively.

The "Tool Criteria" contains a list of 233 requirements that were derived from the previously identified use cases and are classified into 23 categories. The different categories are: Baselines, Collaboration, Custom views, Edits, Export, Extensibility, Help, History, Import, Integration, Models, Navigate, Offline/Online, Permissions, Reports, Requirements architecture, Reviews, Testing, Traceability, Usability, Workflow, Pricing and Technology.

Each item in the tool criteria list has a numeric ID which identifies it, a related use case, the use case's priority, the requirement's category, its first pass value, its feature description and the its priority. The different priority values are 1, 2 and 3, representing low, medium and high priority respectively. The "First Pass" value is used as a means of filter in the evaluation process; it indicates features that a tool must have in order to continue to the second phase of the evaluation.

The RM Tools Evaluation Results document includes the evaluation of all initial tools against the requirements list, a list of the best 17 tools and their scores, the scored obtained by these 17 tools differentiated by category, the list of initial tools and the list of the original use cases. As in the case of the INCOSE Survey (2010), this document is publicly available and allows a direct comparison of tools against defined features. The difference with the INCOSE Survey is that this evaluation was performed by one researcher in Seilevel, in order to avoid bias and obtain consistent results, and not by the RM Tool vendors themselves (Beatty, 2013).

According to Seilevel, the list of requirements included in these documents was constructed taking as a base Seilevel's previous work and the INCOSE Survey (2004). However, this list presents several requirements that are identical to the requirement in the criteria list created by DSTO.

A characteristic of Seilevel's list, that differentiates it from the previous lists, is its inclusion of specific software products, mainly from the Microsoft family, in some of the requirements. In these requirements, specific software products are mentioned without specifying the type of software it represents. Other requirements in the list mention the class of software and use the specific software products as examples.

The list of requirements included in the documents published by Seilevel contain the same requirements, but under different sets of categories. The first document, the RM Tools Evaluation Worksheet (Seilevel, 2011a), applies a more detailed categorization, dividing the requirements in relevant sections that

correlate to RM-related activities. Therefore, this document is the one used for the comparison of criteria lists in the next chapter.

4.7 Other criteria lists

In addition to the previously presented criteria lists, a variety of lists and relevant research was found as a result of the literature review. However, due to their partial availability or their different scope, they are excluded from the criteria list analysis and comparison. These criteria lists are taken into consideration for the definition of criteria to compare the main criteria lists' content and attributes.

A value-based RM Tool selection approach was developed at Siemens Program and System Engineering (PSE) and published by Heindl, Reinisch, Biffl and Egyed in 2006. This approach aims at helping find the optimal tool support for each particular project in Siemens PSE. Between the activities performed in this research, a features catalogue for RE tools was created and organized in the form of a tree. The leaves of this feature tree consist of the tool's requirements, which were derived from RE processes, the INCOSE Survey (INCOSE, 2010) and from the experience of tool users at Siemens PSE (Heindl et al., 2006.) Unfortunately, only a portion of this feature tree was included in the publication of this research.

Carvallo, Franch and Quer (2005) created a method named COSTUME dedicated to building ISO/IEC 9126-1-compliant quality models for the evaluation and selection of COTS components, and applied it for the construction of a quality model for RM tool selection. In their article, they presented the COSTUME process, a detailed description of its application for the construction of the quality model for RM tool selection, and the results of the utilization of the later quality model in a real RM tool selection scenario. This quality model for RM tools selection is also presented partially.

The ISO/IEC TR 24766 (2009), Information technology – Systems and software engineering – Guide for requirements engineering tool capabilities, is a technical report developed by a joint technical committee established by the ISO (International Organization for Standarization) and the IEC (the International Electrotechnical Commission). This report contains a total of 158 RE tool capabilities, which are divided into six sections, namely requirements elicitation, requirements analysis, requirements specification, requirements verification and validation, RM and other tool capabilities. The section RM covers main RM activities, such as version management, change management and traceability management. There are two extra categories included in this section, risk management and project management, which are not specifically part of the RM or RE process, but are related to it. Other important features in RM tools, such as Baselining, Requirements capture, Traceability analysis, Import and Export from and to other sources, Collaborative access and Graphical user interface, are included in other sections of the report. However,

due to the focus on RE requirements, instead of only RM tools, this report was taken into account for the creation of the evaluation criteria, and not for the analysis and comparison of criteria lists' contents.

Lang and Duggan (2001), Salo and Käkölä (2005), and Sinha, Sengupta and Chandra (2006), focused their investigation on collaborative RM tools, one of the two interest areas in this research. They point out that currently available RM tools do not provide appropriate support for multidisciplinary teams working on a distributed environment. To solve this issue, the research teams first identified tool requirements for collaborative RM, developed a software solution that satisfied those requirements, and validated their solution on a practical environment. Being the requirements identified in these researches limited to collaboration aspects of requirement tools, they are not taken into consideration for the comparison of full lists of requirements for RM tool. However, they will serve as input for the collaboration aspects of the new list of requirements created in this research.

The other interest area of this research, RM tools used in SPL settings, is approached by several studies that will serve as a reference for the creation of SPL related requirements later in this research. On one hand, the studies conducted by Von der Maßen and Lichter (2004), Schmid, Krennrich, and Eisenbarth (2006), Eriksson, Börstler and Borg (2009), Thurimella and Janzen (2011), analyze individual tools or tool extensions for the management of requirements in SPL developments, presenting the tool's main features and their application, for example by case studies. On the other hand, Lisboa et al. (2010) and Djebbi, Salinesi and Fanmuy (2007), compare available tools or tool extensions, evaluating their level of support for the activities involved in RM in SPL settings.

The article on RM tool selection, published by Forrester Research (Schwaber et al., 2007), presented a list of main features that should be expected in a RM tool together with a description of those features that are frequently expected, but should not be. This article would help in the identification of possible out of scope requirements included in the lists.

This chapter presented currently available criteria lists for the evaluation of RM tools, as well as partially available criteria lists and research published in the fields of collaborative RM tools and tools used in SPL settings to support RM activities. This content will serve as a basis for the definition of criteria used to evaluate the main criteria lists described here and to construct a new criteria list in Chapter 7.

5 EVALUATION FRAMEWORK FOR THE ANALYSIS AND COMPARISON OF CRITERIA LISTS

This section presents the evaluation framework defined to analyze and compare the main criteria lists presented in the previous chapter. This analysis and comparison is performed from two points of view, the attributes and the contents of the criteria lists; therefore, two different sets of evaluation criteria were defined.

Section 5.1 presents the attributes considered for the evaluation of the lists' general organization and presentation. Section 5.2 presents the process followed for the identification of categories used to evaluate the lists' contents, the obtained set of categories, and the scoring system defined to grade the criteria lists' content coverage level.

5.1 Attributes

To be able to analyze and compare the specific qualities of the available criteria lists, a set of distinctive attributes was defined. The goal of this set of attributes is to represent the main characteristics of the lists in terms of their presentation, the organization and structuring of the requirements included in the list, the system used to identify the requirements, the style used in the requirements description, and the means used to clarify or support the requirements in the document.

For the definition of the attributes, the main criteria lists presented in the previous chapter were examined and contrasted, looking for those aspects in which these criteria lists agree, as well as those aspects in which they disagree. As a result of this analysis, a set of nine attributes was obtained. These attributes and their descriptions are presented in Table 3.

TABLE 3 Attributes for lists' analysis and comparison

Attribute name	Description
Document format	Format in which requirements are presented in the document, such as tables, numbered lists, bulleted lists, or text
Requirements' structure levels	Amount of levels of hierarchy used to organize or group the requirements in the document
Requirements' position in structure	Level in the hierarchy at which requirements are located, showing the depth of the hierarchy used to organize the requirements and whether there is balance in the structure
Requirements' identification	System used throughout the document to identify the requirements, such as numbers, codes or mixed systems
Requirements' formulation	Form in which the requirements' descriptions are written, such as statements, questions, or a combination of both
Detail level	Level of detail, clarity and specificity of the requirements' descriptions, rated as low, medium and high
Prioritized requirements	Whether priorities are assigned to requirements included in the lists
Priority levels	Amount of priority levels used, when applicable
Additional information	Inclusion of additional information supporting or clarifying the requirements

5.2 Categories

To analyze and compare the content of the criteria lists, a set of requirements' categories was created. These categories were obtained from the analysis of RM related concepts in the literature, the categories found in available criteria lists, and from research conducted in the fields of collaboration of distributed teams and SPL engineering and management.

The goal at this stage was to produce a set of categories that would allow the evaluation of the entire content of the available criteria lists. As a way to avoid partiality, the set of categories is based in all criteria lists, not just in one of them; being complemented by categories obtained from the literature.

The category identification process is depicted in Figure 8. As a first step, the categories were obtained from the available criteria lists and literature. Different factors were taken into consideration for the selection of the categories, such as its relation to actual features of RM tools, the grouping of sets of closely related requirements, and the inclusion of a balanced amount of requirements in relation to other potential categories. Then, names for each of the identified categories were defined. This is due to the fact that current criteria lists or literature give different names to categories that are actually equivalent. Therefore, in some cases the most representative category name was selected from a current criteria list or the literature; while in other cases, the category was renamed as to better represent the equivalent categories observed among

the current criteria lists or literature. As a last step, the renamed categories were grouped in meta-categories, including sets of related categories.

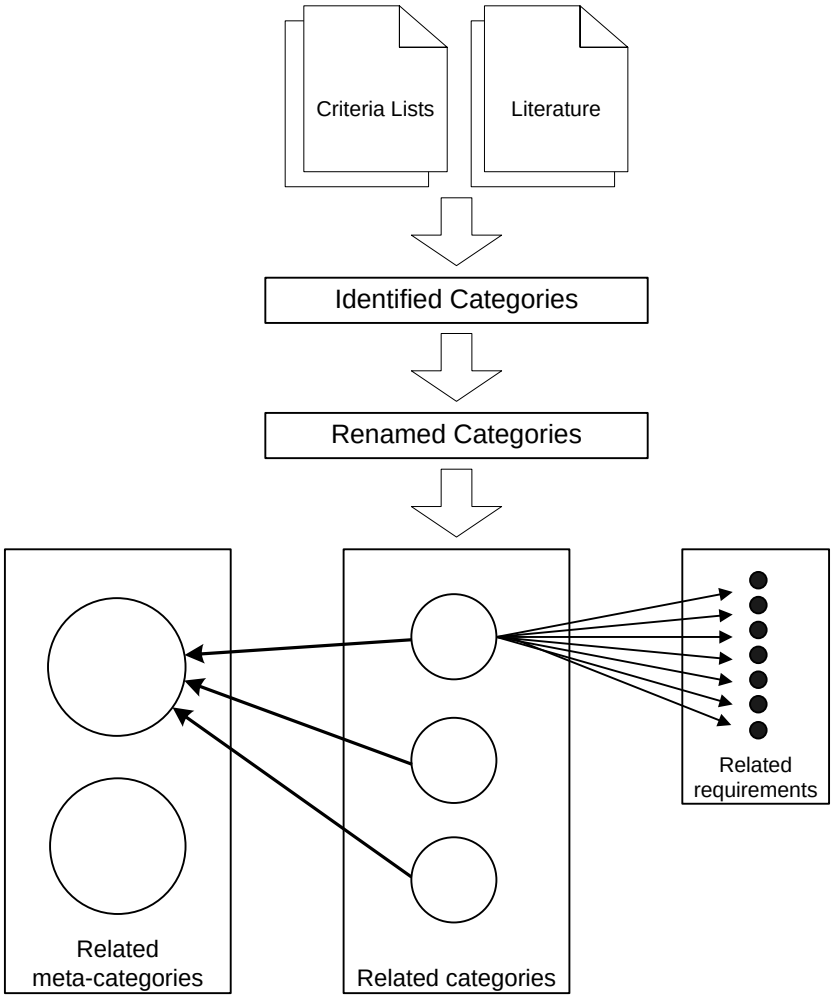

FIGURE 8 Category identification process

The category identification process produced as a result a set of 36 categories. The list of identified categories and the sources from which they were derived is presented in Appendix 1. These categories were grouped into six meta-categories of related categories, namely "RM main activities and baselining", "Information management", "Requirements' data management", "Technical specification, licensing and support", "Out of scope requirements", "Distributed collaboration" and "Software product line requirements". The meta-categories were further divided into meta-categories including general content and meta-categories including content for the evaluation of features supporting distributed SPL requirement related activities.

Meta-categories, the categories they include, and the division of the meta-categories into general content and content for the evaluation of features

supporting distributed SPL requirement related activities, are depicted in Figure 9.

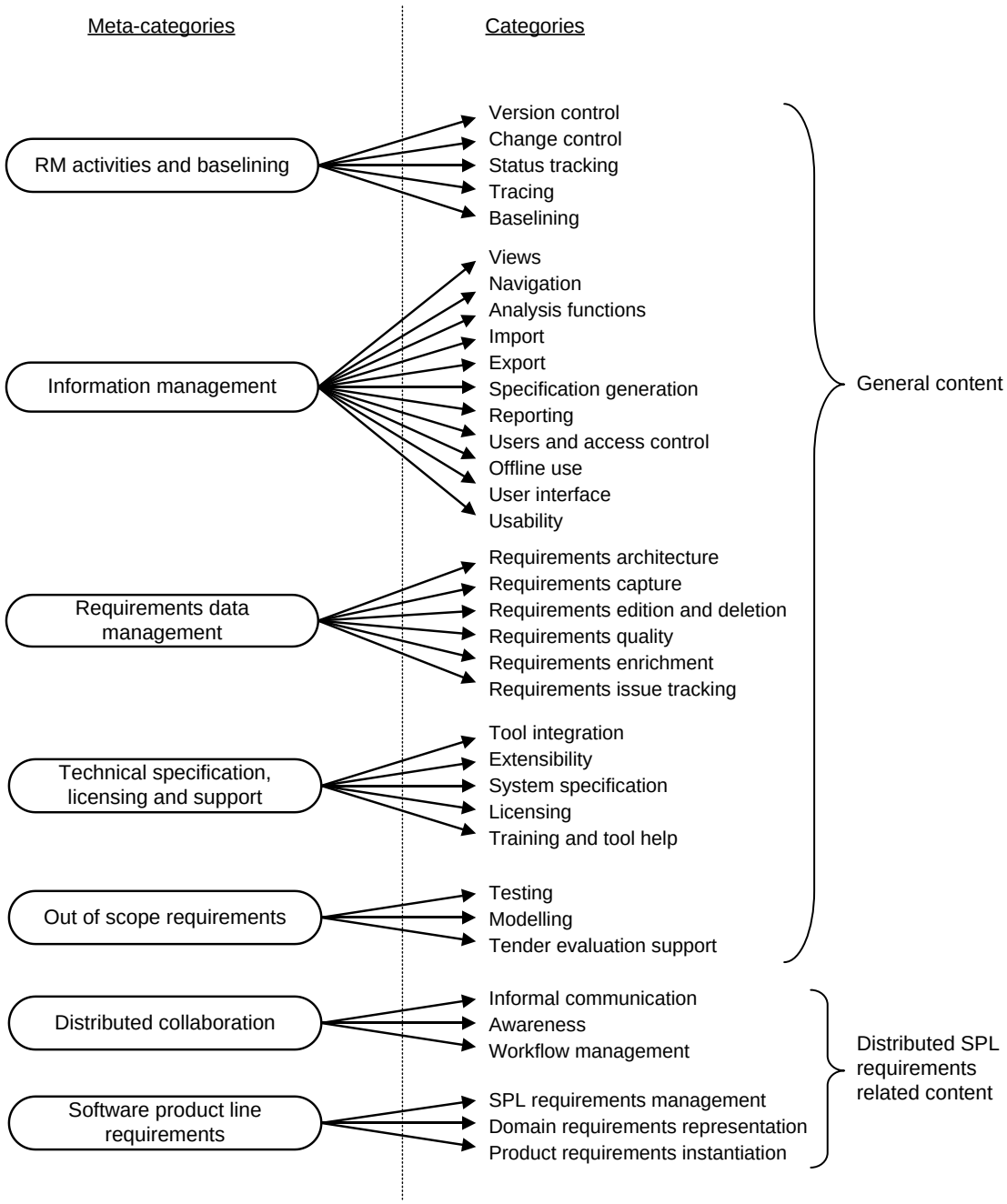

FIGURE 9 Categories for content analysis and comparison

The following subsections describe meta-categories and their categories in more detail. Section 5.2.1 presents general content meta-categories, while Section 5.2.2 presents meta-categories with distributed SPL requirements related content.

5.2.1 General content

This section presents the general content categories identified from the currently available criteria lists and the literature. The following subsections focus on a specific meta-category and describe the categories included in it.

5.2.1.1 RM main activities and baselining

This meta-category groups requirements evaluating the support of the tools of RM core activities, as well as baselining.

The RM categories were derived from the division presented by Wiegers et al. (2013) of the RM domain; namely version control, change control, status tracking and tracing. Baselining, however not a RM core activity, is included in this meta-category due to the importance it possess as a starting point for RM activities. These categories are described in Table 4.

TABLE 4 RM main activities and baselining meta-category

Category name	Description
Version control	Criteria evaluating the tool's ability to version single requirements and requirements' sets, as well as other objects stored in the tool; allowing comparison of current with older versions, restore older versions, etc.
Change control	Criteria evaluating the management of formal changes to the requirements, together with formal comment or discussion capabilities, and the documentation of changes history and its visualization
Status tracking	Criteria evaluating the tracking of requirements' status, as well as the definition of customized status values for specific projects
Tracing	Criteria evaluating the linking of objects having the same or different type, as well as objects stored inside the tool or outside the tool
Baselining	Criteria evaluating the tool's ability to support baselines, allowing their creation, comparison and management

5.2.1.2 Information management

This meta-category groups requirements evaluating tool features that, without being part of the RM main activities, provide key support for the management of requirements in the tool. These categories are related to the input and output of data to and from the tool, the visualization and analysis of the information stored in the tool, the generation of reports and specifications, as well as the control of users and their rights.

For this meta-category, eleven categories of requirements were identified from the criteria lists' analysis, including "Views", "Navigation", "Analysis

functions", "Import", "Export", "Specification generation", "Reporting", "Users and access control", "Offline use", "User interface" and "Usability". These categories are described in Table 5.

TABLE 5 Information management meta-category

Category name	Description
Views	Criteria evaluating the visualization of data stored in the tool and the ability of users to adapt these views to their specific needs, as well as the capability of filtering and sorting the information displayed in the views
Navigation	Criteria evaluating the user's ability to view and navigate the requirements stored in the tool, along with the capacity to modify the position of requirements in the hierarchy in a simple way, perform searches and add bookmarks
Analysis functions	Criteria evaluating the ability of the tool in terms of analysis of stored data, mainly regarding the status of the project and the consistency of the relations between requirements
Import	Criteria evaluating the import of requirements into the tool from different types of existing documents, as well as the ability to parse and identify requirements from text documents
Export	Criteria evaluating the export of requirements into different standard formats
Specification generation	Criteria evaluating the generation of requirement specifications, as well as the ability of the user to define the format and the content of the final generated document
Reporting	Criteria evaluating the ability of the tool to generate different types of reports based on the information stored in the tool
Users and access control	Criteria evaluating the administration of users, user groups, user roles, and their permissions to perform different tasks within the tool
Offline use	Criteria evaluating the ability of the user to view and edit requirements in a disconnected mode, as well as the synchronization of modified data on reconnection
Usability	Criteria evaluating features that simplify the usage of the tool

5.2.1.3 Requirements' data management

This meta-category groups categories of requirements that evaluate tool features supporting the creation, edition, deletion and consistency checking of requirements in the tool, as well as the definition of the requirements' structure and their content.

Six categories of requirements belonging to this meta-category were identified from the lists' analysis. These include "Requirements' architecture", "Requirements' capture", "Requirements' edition and deletion", "Requirements'

quality", "Requirements' enrichment" and "Requirements' issue tracking". These categories are described in Table 6.

TABLE 6 Requirements' data management meta-category

Category name	Description
Requirements' architecture	Criteria evaluating the definition of the structure of single requirements and their attributes, the organization of requirements in groups or hierarchies, and the definition of templates and glossaries
Requirements' capture	Criteria evaluating the creation of requirements directly in the RM tool
Requirements' edition and deletion	Criteria evaluating the modification of specific attributes in single requirements or requirement sets, as well as the removal of requirements from the tool
Requirements' quality	Criteria evaluating the quality of requirements in the tool, e.g. spelling and grammar; as well as the inclusion of other tools to improve the quality of requirements, such as dictionaries or acronym lists
Requirements' enrichment	Criteria evaluating the formatting of the requirements' text, such as the inclusion of symbols, mathematical expressions, or foreign characters; as well as the association of non-text objects, like videos or images, to the requirements' descriptions
Requirements' issue tracking	Criteria evaluating the creation issues related to requirements, associating them to requirements, and track the issues' status

5.2.1.4 Technical specification, licensing and support

This meta-category groups categories of requirements related to the integration of the RM tool with other tools used in the project, the ability of the user to adapt the tool to his specific needs, system specifications of the tool, licensing system and other support provided to the user by the tool vendor.

Five categories of requirements belonging to this meta-category were identified from the lists' analysis, including "Tool integration", "Extensibility", "System specification" "Licensing", and "Training and tool help". These categories are described in more detail in Table 7.

5.2.1.1 Out of scope requirements

This meta-category consists of categories of out of scope requirements found in currently available criteria lists. A requirement was classified as out of scope when it evaluated a feature that was not a RM main activity, a support activity of the RM process, a general support feature or technical specification. The identified out of scope requirements were grouped under the categories "Modelling", "Testing" and "Tender evaluation support", and are described in Table 8.

TABLE 7 Technical specification, licensing and support meta-category

Category name	Description
Tool integration	Criteria evaluating the ability of the tool to integrate with other tools used in the development project with the objective of sharing data stored in the RM tool in a transparent way, e.g. project management tools, configuration management tools and problem tracking tools
Extensibility	Criteria evaluating the ability of the user to adapt and extend the tool in order to satisfy the needs of the organization or project, e.g. by the usage of an external API or script language
System specification	Criteria evaluating technical aspects of the tool, such as supported platform, database specifications, amount of concurrent users, amount of requirements, stability and recoverability
Licensing	Criteria evaluating the tool's licence type, warranty and cost, as well as the maintenance and upgrade policy offered by the tool vendor
Training and tool help	Criteria evaluating the type of training provided by the tool vendor to the user, as well as the type of help and technical support available

TABLE 8 Out of scope requirements

Category name	Description
Modelling	Criteria evaluating the support of the tool in terms of the representation of requirements through models, such as data flow diagrams, process flow diagrams, context diagrams, between others
Testing	Criteria evaluating features that facilitate the testing the of developed software, such as the generation of regression test suite, generation of test scripts, etc.
Tender evaluation support	High level requirement evaluating the support of the tool for tender evaluation

According to Wiegers et al. (2013), requirements' modelling is performed during the requirements' specification, one of the RD main activities, as a way to complement and augment the requirements descriptions. By being part of the RD process, instead of RM, requirements evaluating modelling features are considered out of scope. In their article containing List 2, Gabb et al. (1997) omitted requirements' modelling and analysis features as well, considering that the main purpose of a RM tool is the management of requirements instead of their analysis. Modelling requirements were also included by ISO/IEC TR 224766 (2009) under the Requirements analysis category, instead of the RM category. Additionally, modelling related requirements were found in only one of the main criteria lists for RM tool evaluation (Seilevel, 2011a).

Testing is an activity of the development cycle based on the requirements (Wiegers et al., 2013), but not part of the actual RM discipline. Additionally, only two of the criteria lists for RM tools evaluation contain test related requirements, while such features have not been pointed out in the literature as

required in RM tools. Based on this, testing requirements are considered out of scope.

The category "Tender evaluation support" was created to allow the classification of a unique high level requirement found in two of the analyzed lists, namely the ones created by DSTO (Gabb et al., 1997) and Seilevel (2011a). This requirement, despite being included in Seilevel's evaluation worksheet (Seilevel, 2011a), is not present in the RM tools evaluation results' document (Seilevel, 2011b), leading to the conclusion that the requirement was not taken into consideration for the actual evaluation of tools. Additionally, this requirement represents a functionality that is not part of the RM discipline nor directly supports it, and therefore is considered out of scope.

5.2.2 Distributed SPL requirements related content

This section presents the categories, identified from the literature and the currently available criteria lists, including content for the evaluation of features supporting distributed SPL requirements related activities. The following subsections describe the categories included in the meta-categories "Distributed collaboration" and "Software product line requirements".

5.2.2.1 Distributed collaboration

This meta-category groups categories of requirements that evaluate features allowing the collaboration of distributed teams. The identified categories consist of "Informal communication", "Awareness" and "Workflow management", and are described in Table 9. These categories were obtained from the results of published research and from categories of requirements included in the analyzed criteria lists.

TABLE 9 Distributed collaboration meta-category

Category name	Description
Informal communication	Criteria evaluating features that allow synchronous or asynchronous communication between users around requirements, containing contextual information
Awareness	Criteria evaluating features that provide the user knowledge about the status or availability of other users of the tool, the assignation of requirements or tasks to himself or to other users, as well as changes or activity around requirements or objects stored in the tool
Workflow management	Criteria evaluating the establishment and management of workflows in the tool

The category "Informal communication" was included in this meta-category, not only due to its presence in the majority of the analyzed criteria lists, but also due to the importance given to it in published research. Synchronous and asynchronous communication is considered to be a key activity supporting collaboration in distributed contexts (Heimbürger et al., 2011; Smite, 2006; Harlea & Greenberg, 1998) and an important feature to be included in RM tools (Sinha et al., 2006).

The category "Awareness" was not included by any of the analyzed criteria lists as a category in itself. However, according to Damian et al. (2012) and Sinha et al. (2006), collaboration tools should be able to provide information to maintain the awareness of distributed team members. They affirm that being RE a complex task involving constant knowledge acquisition and sharing, the maintenance of users' awareness becomes critical.

The category "Workflow management" was included in two of the analyzed criteria lists (Seilevel, 2011a; Hoffmann et al., 2004a), and is classified in this research under the collaboration meta-category due to its support to the coordination of work between members of the team. Workflows help implement RE processes and guide the involved users, ensuring that all needed steps are completed, providing information and granting permissions based on the current step on the process (Hoffmann et al., 2004a; Harlea et al., 1998).

5.2.2.2 Software product line requirements

This meta-category groups categories of requirements that evaluate features supporting RM activities in SPL development. The identified categories, described in Table 10, consist of "Software product line requirements management", "Domain requirements representation" and "Product requirements instantiation".

TABLE 10 Software product line requirements meta-category

Category name	Description
Software product line requirements management	Criteria evaluating features for the explicit allocation of requirements to the SPL platform or to an individual product, indicating whether the requirement is shared by all members of the SPL or is a special product requirement, as well as the visualization of the SPL information and the possibility to adapt them based on the user needs
Domain requirements representation	Criteria evaluating the representation of the defined scope through tables, models or trees; as well as the inclusion of mechanisms to control the integrity of domain models are included as well in order to detect redundancies, anomalies and inconsistencies in the models
Product requirements instantiation	Criteria evaluating the derivation of single products from the SPL infrastructure, the analysis of the consistency of the created product configurations, and the possibility to modify these products in the future

The categories listed in the table above were not included in any of the analyzed criteria lists. These categories were obtained solely from published results of research describing tool requirements and features supporting RM in SPL (Beuche et al., 2007; Djebbi et al., 2007; Lisboa et al., 2011; Eriksson et al., 2009; von der Maßen et al., 2004; Schmid et al., 2006; Thurimella et al., 2011; Salo et al., 2005).

5.2.3 Scoring system of category content coverage level

This section presents the scoring system adopted to rate the level of coverage of the main criteria lists. Scores are assigned to the criteria lists according to the amount of requirements the lists contain for each of the previously identified categories. This scoring allows for a later quantitative analysis and comparison of the lists' level of detail per category, as well as a general content coverage analysis.

The objective of this score system is to make the ultimate analysis and comparison of the lists more clear and simple. By replacing the number of requirements each criteria list contains per category with a range value, the identification of those lists including no requirements, some requirements, or numerous requirements under a specific category, becomes easier. Furthermore, this score system simplifies the comparison of the lists' coverage level, presenting a manageable amount of score values for evaluating and contrasting.

The scoring scale is presented in Table 11 and consists of four different scores. The first score, the minus sign, indicates the absence of requirements in the analyzed criteria list under a specific category. The following three scores, the pluses, indicate the amount range under which the identified number of requirements falls for the analyzed criteria list.

TABLE 11 Category content coverage scoring values

Score	Description
-	No requirements identified for the category
+	One or two requirements identified for the category
++	Between three and five requirements identified for the category
+++	Six or more requirements identified for the category

Similar scoring strategies, using pluses, minuses and other symbols as well, have been applied in published research to evaluate available RE and RM tools (Abma, 2009; Aiello, Bulanov & Groefsema, 2010; Carrillo de Gea et al., 2010). However, in all these studies, the assignation of scores is based on a qualitative assessment of the tools' support instead of a quantitative estimation, as is the case in this research. Besides, these studies use their scoring for the evaluation of actual tools; none has applied it to the evaluation of criteria lists.

This chapter described the criteria to be used for the analysis and comparison of the main criteria lists presented in Chapter 4. These criteria were divided in attributes, to evaluate the lists' presentation and organization, and categories, to evaluate the lists' content. Besides, a scoring system to rate the level of content coverage of the lists was defined.

6 ANALYSIS AND COMPARISON OF CRITERIA LISTS

The goal of this chapter is to present the results obtained from the analysis and comparison of the available criteria lists for RM tool evaluation. The analysis is conducted taking as a basis the criteria defined in the previous chapter, namely the lists' attributes and the set of categories.

The following section, Section 6.1, presents the results of the analysis and comparison of the lists' presentation and organization attributes. Section 6.2 presents the results obtained from the evaluation of the lists' general content and distributed SPL requirements related content. In the last section of this chapter, Section 6.3, the results obtained from the analyses and comparisons are summarized, and conclusions are derived from them.

6.1 Analysis and comparison of the lists' attributes

In this chapter, the presentation, structure and other general characteristics of the criteria lists for RM Tool evaluation are analyzed and compared. The information collected from the criteria lists, based on the defined attributes, is summarized in Table 12. Then, the attributes of the analyzed criteria lists are compared.

As it can be observed in the table, each of the analyzed criteria lists differs from the other lists in several aspects. This is not the case of List 5, which, being created based on List 4, presents the same main characteristics of this list. Nevertheless, clear similarities can be distinguished among the criteria lists originating from different sources as well.

TABLE 12 Comparison of lists' attributes

	List 1: INCOSE	**List 2:** DSTO	**List 3:** ITEA DESS	**List 4:** Daimler Chrysler	**List 5:** RWTH Aachen	**List 6:** Seilevel
Document format	table	numbered list	mixed list	table	table	table
Req. structure levels	3	4	3	3	3	2
Req. position in structure	2nd and 3rd	2nd to 4th	2nd to 3rd	3rd	3rd	2nd
Reqs. identification	code	code	mixed	code	code	number
Reqs. formulation	mixed	question	mixed	statement	statement	statement
Detail level	medium	high	medium	medium	medium	high
Prioritized reqs.	no	yes	no	yes	yes	yes
Priority levels	-	4	-	3	3	3
Additional information	none	nomenclature priorities' description	none	stakeholders categories' rationale	stakeholders categories' rationale	use cases

Regarding the format of the document containing the criteria lists, the most frequent format used is a table. This format is presented by four of the six lists, namely List 1, List 4, List 5 and List 6. The structure of these tables, or the number of columns included, varies from list to list. While in List 1 the identification of the requirements, the requirements' name and the requirements' description are in one column; List 4, List 5 and List 6 present this information in a more structured way using separated columns. On the other hand, List 2 and List 3 present their requirements in a less structured format as the previous lists, in the form of a numbered list and a mixed list respectively. However, while the organization of List 2 is consistent and clear, allowing requirements to be easily identified; requirements in List 3 are presented using a mixed style, diminishing its clarity and making the document less suitable for direct tool evaluation. This mixed style consists of requirements presented as bulleted lists, numbered lists, a table, and in the form of blocks of text, which makes the identification of single requirements more difficult.

The requirements included in all criteria lists are divided into certain categories or groups of related requirements. However, the amount of levels of hierarchy used in the lists varies from two levels to four levels. The most common amount of levels in the requirements' structure is three, being presented by four lists, List 1, namely List 3, List 4 and List 5. List 2 presents four levels, while List 6 only uses two.

Regarding the position of the requirements in the defined structure, it was observed that in four of the lists the requirements have a fixed position, while in the other two the position of the requirements varies from category to category.

An example of this is List 2, where some requirements are high in the hierarchy, at level two right under a meta-category, while others are deep in the hierarchy, at level four under a sub-category.

The requirement's identification system present in most of the lists consists of a code deriving from the hierarchy of requirements. The criteria lists using this schema are List 1, List 2, List 4 and List 5. This system allows the user of the lists to identify the category of the requirement and the position of the requirement or sub-requirement in the category just by looking at the requirements' ID or the numbering assigned by position to the categories and requirements. In the case of List 4 and List 5, the system used allows the identification of the requirements' stakeholder as well. List 3 and List 6, on the other hand, use different identification systems as the previous lists. List 3 presents a mixed style, while List 6 uses numbers to uniquely identify the requirements. Requirements in List 6 have just one number assigned to them, without reference to the category the requirements belong to, which can be identified in the category column of the requirement row. This numeration system is also not continuous, which makes the identification of missing or overlooked evaluation criteria more difficult. List 6 presents no clear system for assigning IDs to requirements. When the list is sorted by the category field, the IDs of the requirements in each category are not consecutive. Same results appear when sorting by Use Cases.

The requirements in the analyzed criteria lists are described in two manners, as statements and as questions. Three of the analyzed lists present their requirements mainly in a statement form, namely List 4, List 5 and List 6. List 1 and List 3 contain a combination of the two forms, while List 2 presents mainly questions.

Differences can also be found regarding the level of detail of the requirements in the lists. List 2 and List 6 present requirements with a high level of detail, specifying the exact features to be evaluated in the RM tools. List 1, List 3, List 4 and List 5 describe requirements more generally.

In terms of prioritization of requirements, List 2, List 4, List 5 and List 6 assign priorities to the requirements using different priority values and levels. List 2 uses priority values that go from P1 to P4, being P1 the highest. List 4 uses a system of plusses and minus symbols. List 5 inherits the priorities used by List 4, and adds a PL related priority to all requirements. And finally, List 6 assigns numbers from 1 to 3, being 3 the highest priority. Regarding the levels of priority used, List 2 applies four levels, while List 4, List 5 and List 6 only three. List 1 and List 3, on the other hand, present no priority assignation.

The addition of information complementing the criteria list is another distinctive aspect. List 2 is the only list presenting a nomenclature section with descriptions of some of the terms used in the document, as well as a explanation of the priorities assigned to the requirements. List 4 includes the rationale of each category and the main stakeholders of the requirements, information maintained by List 5. List 6 includes related Use Cases from which

the requirements were initially derived from. List 1 and List 3 do not include relevant additional information supporting the criteria list.

6.2 Analysis and comparison of the lists' contents

In this chapter, criteria lists are analyzed and compared in terms of their content. The lists' content is previously divided into general content, including also out of scope requirements, and content related to features supporting distributed SPL requirements management.

In order to analyze and compare the content of the lists, a series of steps was followed. First, the requirements in each list were analyzed and classified. Identified requirements were then grouped in the categories of related requirements identified in Chapter 5. In specific cases, requirements were allocated to more than one category due to their broad scope. Once the requirements of all lists were classified in categories, a score was assigned per category to each criteria list.

The following sections present the scores allocated to the criteria lists based on the level of support of the identified categories. Section 6.2.1 presents the results of the lists' general content coverage. Section 6.2.2 presents the results of the lists' coverage of content supporting distributed SPL requirements management. The complete scores assigned to each criteria list are presented in Appendix 2.

6.2.1 General content

This section focuses on the analysis and comparison of the criteria lists based on their coverage of the general content meta-categories, namely "RM main activities and baselining", "Information management", "Requirements' data management", and "Technical specification, licensing and support". The analysis and comparison of the lists based on their inclusion of "Out of scope requirements" is presented in this section as well.

6.2.1.1 RM main activities and baselining

As mentioned in Chapter 2, the RM discipline can be synthesized in four core activities, namely Version control, Change control, Status tracking and Tracing. This meta-category includes these RM main activities together with baselining. The results of the comparison of the criteria lists based on their coverage of RM main activities and baselining are summarized in Table 13.

Despite the importance of a RM tool giving support for the RM main activities and Baselining, only List 2, List 3 and List 6 contain requirements to evaluate features supporting the totality of the mentioned activities. Of these lists, List 6 presents the highest level of detail. The remaining of the lists, List 1,

List 4 and List 5, include requirements for four categories of the five, being List 1 the one with the smallest amount of requirements of the three.

In this meta-category, categories covered by all lists include Version control, Change control, Tracing and Baselining; being Tracing the category with the most amounts of requirements. The less supported category of requirements, on the other hand, is Status tracking. This category is covered by three of the lists, with a limited amount of requirements in two of them.

TABLE 13 Comparison of RM main activities and baselining

	List 1: INCOSE	**List 2:** DSTO	**List 3:** ITEA DESS	**List 4:** Daimler Chrysler	**List 5:** RWTH Aachen	**List 6:** Seilevel
Version control	+	+	+	++	+++	+
Change control	+	++	++	+++	+++	+++
Status tracking	-	+	+	-	-	++
Tracing	++	+++	+++	+++	+++	+++
Baselining	+	+	++	+	+	++

(-) No related requirements, (+) between 1 and 2 related requirements, (++) between 3 and 5 related requirements, (+++) 6 or more related requirements

6.2.1.2 Information management

This meta-category includes categories related to the input and output of data to and from the tool, different way to access these data, as well as the control of users and their right to access and modify the information stored in the tool.

Eleven categories of requirements belonging to this meta-category were identified from the lists' analysis. The results of the comparison of the lists based on their coverage of these categories of requirements are summarized in Table 14.

For this meta-category, List 6 is the most comprehensive list, being the only list including requirements for all identified categories. List 2, List 3 and List 5, contain requirements under all categories except one; List 1 covers all categories except two; and List 4, includes requirements in eight out of the eleven categories.

Categories covered by all lists include Views, Navigation, Analysis functions, Import, Specification generation, Reporting, Users and access control, and User interface. Of these categories, the one including the most amounts of requirements is Views, followed by Specification generation. This shows the importance of visualizing the data stored in the tool and generating specification documents.

The categories Export, Offline use and Usability are not included by all lists. The less supported of these categories is Usability, covered only by two lists; while Offline use and Export are covered by four lists.

TABLE 14 Comparison of information management

	List 1: INCOSE	**List 2:** DSTO	**List 3:** ITEA DESS	**List 4:** Daimler Chrysler	**List 5:** RWTH Aachen	**List 6:** Seilevel
Views	+	+++	++	+++	+++	+++
Navigation	+	+++	+	+	+	+++
Analysis functions	++	++	++	++	+	+
Import	+++	++	+++	+	++	++
Export	-	+	+++	-	+	+++
Specification generation	++	+++	+	+++	+++	++
Reporting	++	+++	++	+	+	+++
Users and access control	++	+++	+++	+++	++	++
Offline use	-	-	+	+	+	++
User interface	++	+++	++	+	+	+++
Usability	-	++	-	-	-	+

(-) No related requirements, (+) between 1 and 2 related requirements, (++) between 3 and 5 related requirements, (+++) 6 or more related requirements

6.2.1.3 Requirements' data management

This meta-category includes requirements for the evaluation of features supporting activities closely related to the requirements. In contrast with the previous meta-categories, more accentuated differences were observed in terms of the lists' level of coverage of these categories. The results of this comparison are summarized in Table 15.

TABLE 15 Comparison of requirements' data management

	List 1: INCOSE	**List 2:** DSTO	**List 3:** ITEA DESS	**List 4:** Daimler Chrysler	**List 5:** RWTH Aachen	**List 6:** Seilevel
Requirements' architecture	+	+++	+++	+++	+++	+++
Requirements' capture	+	+	+	-	-	+++
Requirements' quality	+	+	-	+	-	++
Reqs. edition and deletion	-	++	++	+	-	+++
Requirements' enrichment	-	+++	-	++	++	+++
Requirements' issue tracking	-	-	-	-	-	++

(-) No related requirements, (+) between 1 and 2 related requirements, (++) between 3 and 5 related requirements, (+++) 6 or more related requirements

For this meta-category, List 6 is the most comprehensive list, being the only one including requirements for all six categories, followed by List 2, covering five of the six categories. List 1, List 3 and List 4 cover three categories each, while List

5 includes requirements under only two categories. In general, List 1 is the list with the smallest amount of requirements, while List 6 contains the biggest amount.

The only category covered by all lists is Requirements' architecture. The categories Requirements' capture, Requirements' quality, Requirements' edition and deletion, and Requirements' enrichment are included by four of the six lists. Finally, the category Requirements' issue tracking is the less supported in the meta-category, covered by only one of the six lists.

6.2.1.4 Technical specification, licensing and support

For the meta-category "Technical specification, licensing and support", five categories of requirements were identified from the analyzed lists.

The results of the comparison of the lists, based on their coverage of these categories of requirements, are summarized in Table 16.

TABLE 16 Comparison of technical specification, licensing and support

	List 1: INCOSE	**List 2:** DSTO	**List 3:** ITEA DESS	**List 4:** Daimler Chrysler	**List 5:** RWTH Aachen	**List 6:** Seilevel
Tool integration	++	-	+++	+++	+++	+++
Extensibility	++	-	++	++	+++	+
System specification	+++	+++	+++	+++	+++	+++
Licensing	++	-	-	-	-	++
Training and tool help	+++	+	-	-	-	+++

(-) No related requirements, (+) between 1 and 2 related requirements, (++) between 3 and 5 related requirements, (+++) 6 or more related requirements

List 1 and List 6 are the most comprehensive lists in this meta-category, including requirements for all identified categories. List 3, List 4 and List 5 include requirements for three categories out of five, while List 2 includes requirements under two categories only.

List 3, List 4 and List 5 contain requirements for the same subgroup of categories, while List 2 has only one category in common with these three lists.

Requirements related to System specification are included by all lists, followed by Tool integration and the Extensibility, covered by five of the six lists.

The less supported category in this group consists of Licensing, being covered only by two lists.

6.2.1.5 Out of scope requirements

During the analysis of the selected requirements lists, requirements for the evaluation of features out of the RM scope were identified. These features were grouped under the categories Testing, Modelling and Tender evaluation support.

The results of the comparison of the lists based on their coverage of requirements for features considered out of score are summarized in Table 17.

TABLE 17 Comparison of out of scope requirements

	List 1: INCOSE	**List 2:** DSTO	**List 3:** ITEA DESS	**List 4:** Daimler Chrysler	**List 5:** RWTH Aachen	**List 6:** Seilevel
Testing	-	-	+	-	-	+++
Modelling	-	-	-	-	-	+++
Tender evaluation support	-	+	-	-	-	+

(-) No related requirements, (+) between 1 and 2 related requirements, (++) between 3 and 5 related requirements, (+++) 6 or more related requirements

Requirements related to testing were included by List 3 and List 6, while requirements for Modelling were only included by List 6. List 2 and List 6 included both one requirement related to Tender evaluation support.

List 1, List 4 and List 5 do not include requirements that are considered out of scope for the evaluation of RM tools. List 6, on the other hand, includes requirements for all three categories. The requirements for two of these categories are detailed.

6.2.2 Distributed SPL requirements related content

In this section, criteria lists are compared based on their coverage of the meta-categories "Distributed collaboration" and "Software product line requirements". The first meta-category includes categories of requirements supporting the collaboration of teams working in a distributed environment. The second meta-category includes categories of requirements supporting to SPL requirements related activities.

6.2.2.1 Distributed collaboration

For the meta-category "Distributed collaboration", three categories of requirements were identified mainly from the literature review. The results of the comparison of the lists based on their coverage of these categories are summarized in Table 18.

TABLE 18 Comparison of distributed collaboration

	List 1: INCOSE	**List 2:** DSTO	**List 3:** ITEA DESS	**List 4:** Daimler Chrysler	**List 5:** RWTH Aachen	**List 6:** Seilevel
Informal communication	-	-	-	+	+	+
Awareness	-	+	-	-	-	-
Workflow management	-	-	-	+	+	++

(-) No related requirements, (+) between 1 and 2 related requirements, (++) between 3 and 5 related requirements, (+++) 6 or more related requirements

None of the criteria lists includes requirements under all three categories identified for this meta-category. The most comprehensive lists are List 4, List 5 and List 6, containing requirements for two categories. List 2 includes requirements in one category, while List 1 and List 3 do not present any requirements for these categories.

List 6 is the list presenting the most amounts of requirements for the meta-category, though limited.

For this meta-category, none of the identified categories of requirements is covered by all lists. The most supported category consists of Workflow management followed by Informal communication, being included by the same three lists with a limited amount of requirements. The less supported category consists of Awareness, being included by only one list, List 2.

6.2.2.2 Software product line requirements

This meta-category groups categories of requirements, identified from the literature, allowing the evaluation of features that support RM activities in SPL development. The results of the comparison of the lists based on their coverage of these categories of requirements are summarized in Table 19.

TABLE 19 Comparison of software product line requirements

	List 1: INCOSE	**List 2:** DSTO	**List 3:** ITEA DESS	**List 4:** Daimler Chrysler	**List 5:** RWTH Aachen	**List 6:** Seilevel
SPL reqs. management	-	-	+	-	+	-
Domain reqs. representation	-	-	+	-	+	-
Product reqs. instantiation	-	-	-	-	+	-

(-) No related requirements, (+) between 1 and 2 related requirements, (++) between 3 and 5 related requirements, (+++) 6 or more related requirements

For this meta-category, List 5 is the most comprehensive lists, including requirements for all three identified categories. However, the amount of

requirements included by this list is very limited. List 3 presents requirements for two categories, while List 1, List 2, List 4 and List 6 do not include requirements in the meta-category.

The most supported categories in this case are Software product line requirements management and Domain requirements representation, being covered only by two of the six lists with a limited amount of requirements each. Product requirements instantiation, on the other hand, is the less supported category in this group, covered only by one list.

6.3 Summary of lists' analysis and comparison

In this section, the results previously obtained from the analysis and comparison of the currently available criteria lists are summarized and evaluated with the objective to gain knowledge of the strengths and weaknesses of the these lists.

In terms of the attributes of the lists, several aspects were identified in which a majority of the criteria lists coincide. Regarding organization and presentation of the lists, it was observed that most of the lists present requirements as a table and that a three level structure is mostly used to organize the requirements. Regarding the requirements, it was observed that codes are used to identify the requirements, based on the position of the requirement in the structure, and that requirements are mainly formulated as statements, described using an intermediate level of detail. Furthermore, requirements are prioritized in the majority of the lists, using mainly three levels of priorities.

In terms of the content of the lists, a differentiation can be observed regarding the percentage of categories covered by the lists, as well as the level of coverage presented by the lists per category. The percentage of identified categories covered by the analyzed criteria lists is depicted in Figure 10. This figure shows the percentage of categories for which the criteria lists contain requirements, independent of the amount. To simplify the analysis, the meta-categories "Information management", "Requirements' data management", and "Technical specification, licensing and support" are grouped under Support categories.

As it can be observed in Figure 10, none of the analyzed criteria lists fully covers the totality of the identified category groups. While List 4 presents requirements under all identified in-scope categories, the percentage of coverage of the "RM activities and baselining" and "Distributed collaboration" meta-categories, as well as support categories, is not ideal. List 6, on the other hand, presents full coverage of the "RM activities and baselining" meta-category and support categories, but only a 67% coverage of the "Distributed collaboration" category and no requirements under the "Software product line requirements" meta-category.

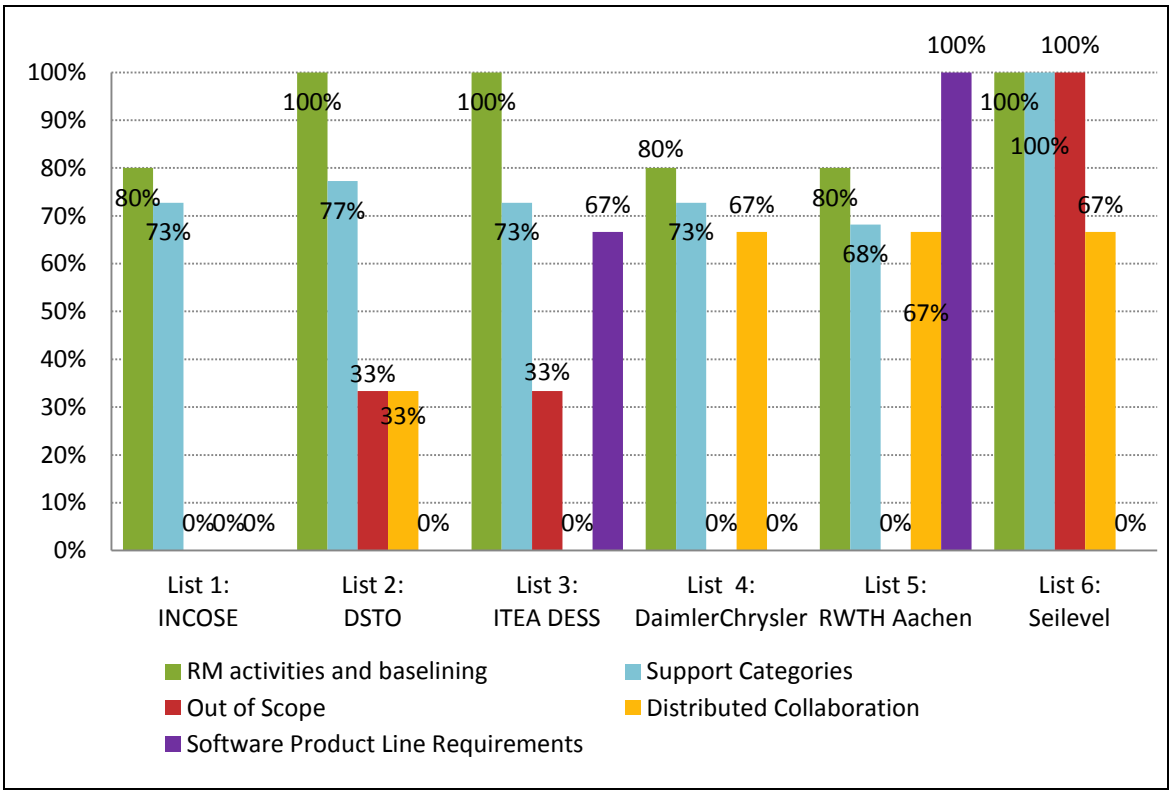

FIGURE 10 Percentage of covered categories per list

The amount of pluses and minuses assigned to the criteria lists, under in-scope categories, is presented in Table 20. As it can be observed in the table, List 6 presents the biggest amount of pluses, as well as the smallest amount of minuses, under general content categories. However, this is not the case under distributed SPL requirements caregories, where List 5 presents the best results.

TABLE 20 Amount of pluses and minuses assigned under in-scope categories

	General Content Categories		**Distributed SPL requirements**	
Criteria Lists	**# of pluses**	**# of minuses**	**# of pluses**	**# of minuses**
List 1: INCOSE	35	10	0	6
List 2: DSTO	47	5	1	5
List 3: ITEA DESS	43	6	2	4
List 4: DaimlerChrysler	40	7	2	4
List 5: RWTH Aachen	40	8	5	1
List 6: Seilevel	65	0	3	4
Maximum amount possible	81	27	18	6

The level of coverage, provided by the analyzed lists, of the identified categories is depicted in Figure 11. This figure is constructed based on the percentage of "+++", "++", "+" and "-" scores assigned to each criteria list for all identified categories, except the categories considered out of scope.

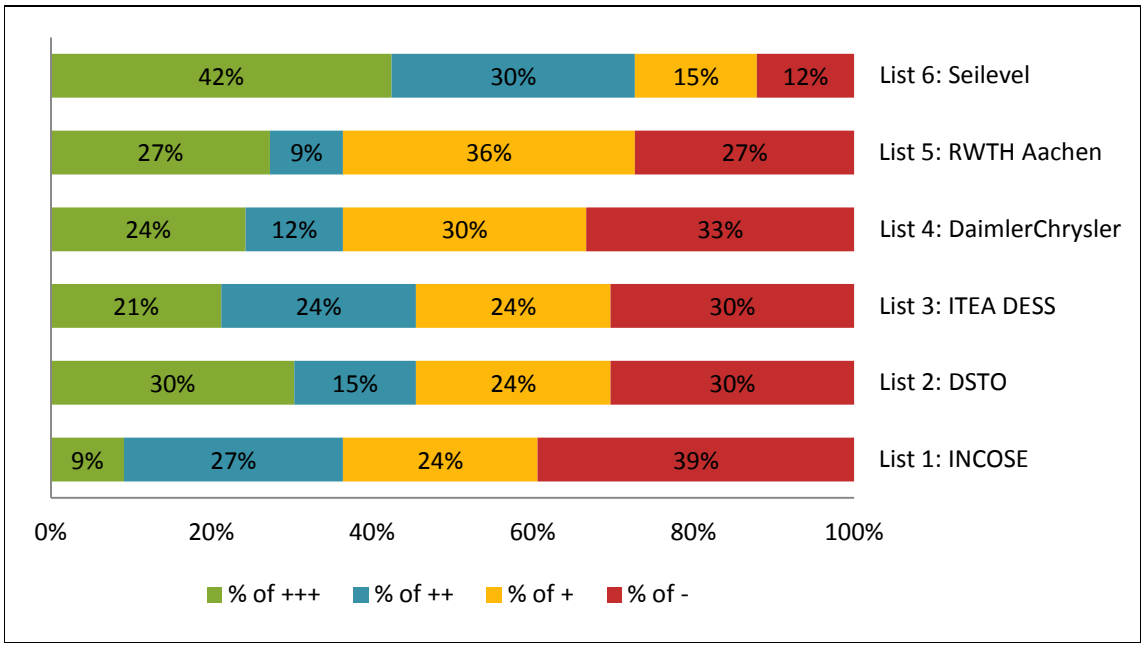

FIGURE 11 Percentage of coverage level per list

Figure 11 clearly shows the higher level of detail presented by List 6. In this image, it can be observed that this list obtained the highest score, a "+++", for a 42% of the categories and presents also the most amounts of "++" scores in comparison to the other lists. The amount of "+" scores assigned to this list constitutes only a small percentage of the list's scores, while the percentage of "-" score is less of the half in comparison to the other criteria lists. These results could be expected due to the bigger amount of requirements included in this list.

The lists analyzed and compared in this chapter originate in different fields and therefore answer to different needs. As a result of this, there are many areas in which these criteria lists present similarities and others in which they present differences, having each their strengths and weaknesses. The previous analyses of the lists' content coverage put in evidence that, while the general content of the lists is mostly detailed, this is not the case for distributed collaboration and software product line requirements related content. Under these categories, a small amount of requirements, and sometimes no requirements at all, were found. This weak support and guidance for the evaluation of features considered crucial in distributed SPL requirements management, leads to the conclusion that a new criteria list is required.

The results obtained in this section, in terms of content and attributes of currently available lists, will be taken into consideration for the creation of the new criteria list. The content of this list will count of two main sources. The first source consists of the requirements under RM and support categories presented by List 6, published by Seilevel (2011a). The selection of Seilevel's list as a foundation for the creation of the new criteria list is based not only on this list's higher level of detail, but also on the fact that it was developed by a company with knowledge and experience in the evaluation of RM tools. The second source will consist of results from relevant research in the field of distributed collaboration and SPL requirements. The selection of research results as a source originates from the limited content, in terms of requirements supporting the evaluation of distributed collaboration and SPL requirement features, observed in the analyzed criteria lists. The attributes of the new criteria list, on the other hand, will be mainly based on those aspects found in the majority of the analyzed criteria lists.

This section presented the results obtained from the analysis and comparison of available criteria lists using the criteria defined in Chapter 5. These results showed the dominant attributes of the criteria lists, as well as their strengths and weaknesses in terms of content. These weaknesses led to the conclusion that a new criteria list is needed; containing more detailed support for the evaluation of tool features used in distributed SPL requirements management. In the next section, the steps followed to construct this new criteria list are described.

7 CREATION OF NEW CRITERIA LIST FOR RM TOOL EVALUATION

This chapter presents the process followed to construct the new criteria list for the evaluation of RM tool. This process consists of three main phases, where the available criteria list selected as a source is transformed into the new criteria list (see Figure 12).

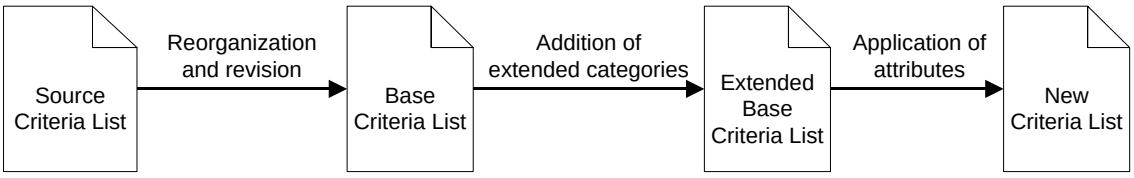

FIGURE 12 New criteria list construction process

In the first phase, the requirements of the criteria list selected as a source are reorganized, revised, and eventually reformulated, obtaining as a result a base criteria list. In the second phase, the categories "Distributed collaboration" and "Software product line requirements" are extended with requirements obtained from current research in these fields. These extended categories are added to the base criteria list. In the last phase, selected attributes are applied to the extended base criteria list, obtaining as a result the new criteria list for the evaluation of RM tools. This new list is presented in its totality in Appendix 3, in the form of a ready template to be used for full RM tool evaluation.

7.1 Reorganization and revision of source criteria list

The first phase in the construction of the new criteria list consists of a series of steps, conducted to obtain a base criteria list. These steps are depicted in Figure 13. As observed in the figure, the input of this phase is the list selected as a source, namely List 6 (Seilevel, 2011a). This selection was based on the results obtained from the analysis and comparison of the content of available criteria lists conducted in the previous chapter. The higher level of detail, better coverage of general content categories, as well as the origin of this list, contributed to its selection as the source criteria list.

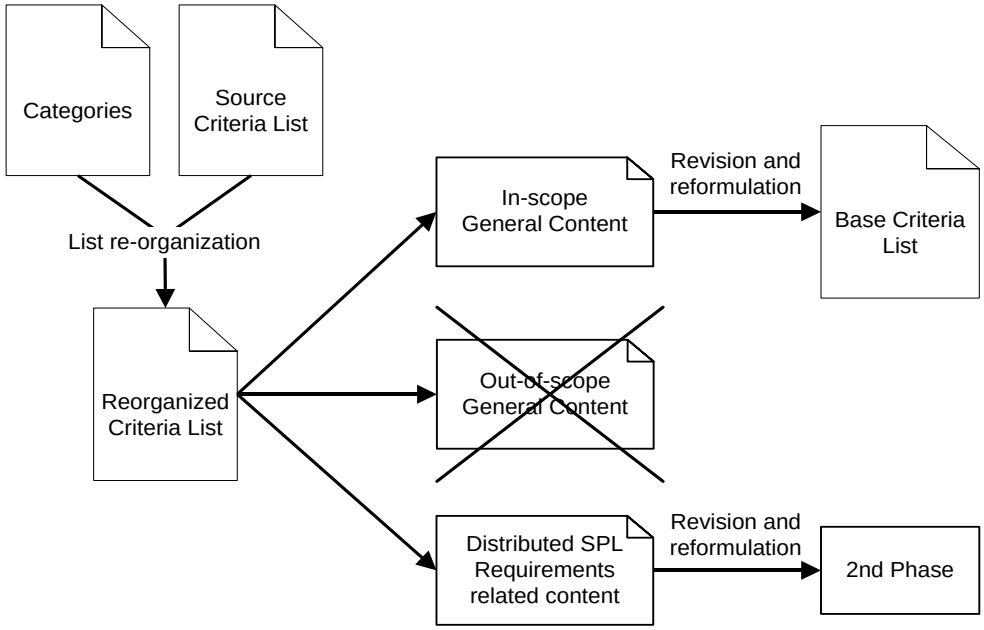

FIGURE 13 Base criteria list obtainment

As a first step to obtain a base criteria list, the requirements included in the source criteria list are reorganized into the 36 categories identified in Chapter 5. The second step consists of dividing the re-arranged requirements into in-scope general content, out-of-scope general content, and distributed SPL requirements related content. The out-of-scope requirements, namely those under the categories "Modelling", "Testing", and "Tender evaluation support", are removed. The descriptions of in-scope and distributed SPL requirements related content are revised and eventually reformulated. Finally, the revised in-scope general content constitutes the base criteria list, while the distributed SPL requirements related content is used as an input of the second phase.

As presented in Section 6.2.1.5, List 6 includes requirements under all categories considered out of scope. These out of scope requirements are displayed in Table 20, and were removed during the second step of this phase.

TABLE 21 Removed out of scope requirements

Category	Out of scope requirement
Modelling	1. Model process flows directly in the tool 2. Visually can model a use case and it automatically translates it into text requirements 3. Automatically generate diagrams from written requirements (e.g., context diagram, use case diagrams) 4. Incorporate UI requirements with a UI prototype (e.g., define a set of requirements or an image and associate it to the requirements. Let you click/hover to view those requirements) 5. Incorporate UI requirements with specific sections of a UI prototype (e.g., define areas of an image that have associated requirements. Let you click/hover to view those requirements) 6. Link models to sub-objects in other models (link decision tree to a step in a process flow) 7. Link requirements to sub-objects in models (link requirements to a step of a process flow, to a system in a context diagram, to a state object in a state diagram) 8. Model context diagrams directly in the tool 9. Model org charts directly in the tool 10. Can mockup low-fidelity screens in the tool directly 11. Can mockup high-fidelity screens in the tool directly
Testing	1. Create a regression test suite based on the requirements 2. Create a system test suite based on the requirements 3. Assign verification methods to requirements 4. Link requirement to status of product verification (from requirement can see support test plans, determining test plan and test procedure coverage, determination of successful test 5. Auto generate a draft UAT script for a set of requirements 6. Create UAT test scripts linked to requirements
Tender evaluation support	1. Can evaluate multiple software vendors against a set of requirements in the tool for review in a matrix format

As mentioned in Chapter 4, a fraction of the requirements in Seilevel's criteria list was found to be vendor specific, making reference to actual software products instead of their respective software types or classes. The purpose of the new criteria list is to serve as a reference to requirement practitioners in general, and not only those using a particular software instance. Therefore, the requirements from Seilevel's list used as a base for the construction of the new criteria list are revised and, when necessary, the requirement's description is reformulated, as to avoid the new list being vendor specific. The main changes applied to the requirements' descriptions in Seilevel's list consist of replacing specific software products by their respective software classes and adding software products as an example of the software class in the requirement description. Additionally, definitions of the referred software classes are added to the list, as well as links to the websites of the specific software products

provided as examples. The objective of these changes is to clarify the lists' requirements, without modifying their original meaning.

7.2 Addition of extended categories

In this section, the new content for the categories "Distributed collaboration" and "Software product line requirements" is presented. The need to broaden these categories was observed in the results obtained from the previous analysis and comparison of currently available criteria lists. The objective of this section is to include relevant requirements to the mentioned categories. These requirements aim at improving the support provided by the new criteria list in the evaluation of RM tools with features allowing distributed SPL requirements management. The extended categories obtained in this section, will be added to the base criteria list to conform the extended base criteria list.

The requirements here presented, are obtained from the literature review and, in a smaller degree, from the analyzed criteria lists presented in Chapter 4. The selection of research results as a main source originates from the limited content, in terms of requirements supporting the evaluation of distributed collaboration and SPL requirements features, observed in the results of the analysis and comparison of available criteria lists presented earlier.

A total of 21 requirements were identified for the "Distributed collaboration" meta-category; eleven of which proceed only from published research, five from analyzed criteria lists, and the last five from both sources. For the "Software product line requirements" meta-category, on the other hand, a total of 33 requirements were identified from published research only.

The following two sections present the requirements identified for the categories belonging to the mentioned meta-categories. These meta-categories, and their categories, were introduced in Section 5.2.2.1 and Section 5.2.2.2 respectively. As presented in those sections, the "Distributed collaboration" meta-category contains the categories "Informal communication", "Awareness" and "Workflow management"; while the meta-category "Software product line requirements" contains the categories "Software product line requirements management", "Domain requirements representation", and "Product requirements instantiation".

7.2.1 Distributed collaboration categories

This section presents the requirements identified for the evaluation of features supporting the distributed collaboration of team members involved in RM activities. These requirements originate from studies published in the field and some of the previously analyzed criteria lists. For each requirement, the description and the sources from which it was obtained are specified.

7.2.1.1 Informal communication

For this category, a total of seven requirements were identified from the literature, two of which are also present in some degree in three of the analyzed criteria lists. Informal communication criteria are listed in Table 21.

As observed in the table, the requirements related to synchronous and asynchronous communication capabilities between users of the tool are present in the majority of the consulted distributed collaboration literature. This indicates the importance of such features for the collaboration of distributed team members.

TABLE 22 Informal communication criteria

Description	Source
Ability to initiate new conversations with other stakeholders	8
Availability of synchronous contextual communication around requirements, such as chat, videoconference or VoIP	1, 3, 4, 5, 7, 8
Availability of asynchronous (email-like) contextual communication around requirements	1, 2, 3, 4, 5, 8
Integration of synchronous and asynchronous communication	8
Insertion of navigable links to relevant objects into messages, automatically when the context is obvious, or manually by users	8
Ability to access all the discussions a user has participated in	1, 8
Ability to edit a conversation's transcripts and save only relevant information	6, 8

[1] Harlea et al., 1998; [2] Lang et al., 2001; [3] List 4: DaimlerChrysler (p.37); [4] List 5: RWTH Aachen (p.38); [5] List 6: Seilevel (p.38); [6] ISO/IEC TR 24766, 2009; [7] Portillo-Rodríguez et al., 2012; [8] Sinha et al., 2006

7.2.1.2 Awareness

For this category, a total of eight requirements were identified from the literature, two of which were also present in some degree in two of the analyzed criteria lists. These requirements are listed in Table 22.

As observed in Table 22, features or requirements related to availability of users and e-mail notification about other users' actions were the most supported by the literature in terms of awareness promotion.

7.2.1.3 Workflow management

A total of six requirements were identified for the evaluation of features supporting the management of processes that guide RM activities. These requirements were obtained mainly from the list created by Seilevel (2011a), and are listed in Table 23.

TABLE 23 Awareness criteria

Description	Source
Information on which users are currently connected to the tool and their status is provided (Session Awareness or Availability)	1, 2, 4, 5
Information on the roles of the users in the project is provided	5
Information on where other team members are located geographically is provided (Global Awareness)	4
Information on other team members workload is provided, letting know ''who is doing what'' (Task Awareness or Change Awareness)	1, 4
Visual cues are provided, helping users be aware of the actions recently performed by other users, online information and pending change requests	4, 5
Ability to personalize the visual information provided	5
E-Mail notifications to affected team members about actions performed by other users	1, 3, 4
Ability to subscribe to requirements of interest to receive automatic notification of the requirement's changes	5

[1] Damian et al., 2012; [2] List 2: DSTO (p.35); [3] List 6: Seilevel (p.38); [4] Portillo-Rodríguez et al., 2012; [5] Sinha et al., 2006

TABLE 24 Workflow management criteria

Description	Source
Ability to define custom workflows for each type of requirement	3
Pre-defined workflows are available	3
Ability to define one custom workflow for all types of requirements	3
Track requirements approval/signoff	3
Request requirements approval/signoff via email	3
Ability to define tasks and assign rights to users for each workflow step	1, 2, 4

[1] List 4: DaimlerChrysler (p.37); [2] List 5: RWTH Aachen (p.38); [3] List 6: Seilevel (p.38); [4] Salo et al., 2005

7.2.2 Software product line requirements categories

This section presents the requirements identified for the evaluation of features supporting RM activities in SPL developments, obtained from studies published in the field.

As mentioned in Chapter 3, these publications can be divided into those presenting requirements and evaluating current tools (Beuche et al., 2007; Djebbi et al., 2007; Lisboa et al., 2011), those presenting new tools created to support RM activities in SPL development and the features of these tools (Eriksson et al., 2009; von der Maßen et al., 2004; Schmid et al., 2006; Thurimella et al., 2011), and those describing requirements for tools supporting

requirement related tasks in SPL development (Salo et al., 2005). In the case of Beuche et al. (2007), requirements were obtained from the report's text which offered a higher level of details than the the criteria list presented in this document, namely List 4. For each identified requirement, the description and the sources from which it was obtained are specified.

7.2.2.1 Software product line requirements management

This category presents a total of 14 requirements for the evaluation of features sopporting the management of product lines requirements. These criteria are listed in Table 24.

TABLE 25 Software product line requirements management criteria

Description	Source
Support for multiple projects and products requirements	1, 5
Ability to create new product line requirements projects	1, 5
Ability to link features to requirements that substantiate these features	3, 4, 5
Ability to easily relate features and requirements to the product lines platform or to a product	5, 7
Ability to control if the affiliation of features and requirements to platform and products is accurate	5
Ability to manage features information, such as name, unique identifier, description, rationale, status, source, priority, maturity and history information	4, 5
Ability to attach supporting material to features, e.g text documents, table sheets, diagrams, pictures, etc	5
Support for multi project and multi product requirement views	1, 5
Support for multi project/multi product requirements status and progress reporting	1
Ability to hide features and requirements that do not address a product of interest to the current user	1, 5
Categorization of findings into errors (severe problems) and warnings (light problems)	5
Ability to define whether errors and/or warning should be reported	5
Query interface retrieving information of features and requirements related to the product line	5
Support for stakeholder collaboration during domain modelling and product instantiation, e.g. using issue modelling	1, 6, 8

[1] Beuche et al., 2007; [2] Djebbi et al., 2007; [3] Eriksson et al., 2009; [4] Lisboa et al., 2010; [5] von der Maßen et al., 2004; [6] Salo et al., 2005; [7] Schmid et al., 2006; [8] Thurimella et al., 2011

As it can be observed in Table 24, a greater portion of the requirements comes from the research published from von der Maßen et al. (2004), alone or in combination with other publications.

7.2.2.2 Domain requirements representation

This category presents a total of 11 requirements for the evaluation of features sopporting the representation of the domain. Requirements and their source are listed in Table 25, where it can be observed that the majority originates from at least two of the consulted research results.

TABLE 26 Domain requirements representation criteria

Description	Source
Support for feature modelling, or other domain representation approaches, e.g. tables, decision models, etc	3, 4, 5, 7
Support for graphical modelling of features	1, 7
Relationship types: provides different types of relationships between the features, e.g. composition, generalization/specification and implementation	1, 2, 4, 5
Mandatory features: represent the features that will always be in the products	3, 4
Variability: represents the variabilities a feature can have (optional, alternative and or)	1, 2, 4, 6
Ability to handle commonalities, variabilities and product-specific requirements for individual products	6
Feature group identification: classifies the features according to the type of information they represent; e.g. capability, domain technology, implementation techniques and operation environment	4, 5
Composition rules: create restrictions in the domain for representing and relating the features, such as mutual exclusion and dependency, regular expressions, or artificial intelligence, among others	1, 3, 4, 5
Redundancy check: same concept modelled twice in the model	5, 7
Anomaly check: possible instantiations lost due to improper relationships between features	5, 7
Consistency check: verifies if the generated domain follows the composition rules created and no conflicting relationships were created in the models	2, 4, 5, 7

[1] Beuche et al., 2007; [2] Djebbi et al., 2007; [3] Eriksson et al., 2009; [4] Lisboa et al., 2010; [5] von der Maßen et al., 2004; [6] Schmid et al., 2006; [7] Thurimella et al. 2011

7.2.2.3 Product requirements instantiation

This category presents a total of seven requirements for the evaluation of features sopporting the instantiation of products. Requirements and their sources are listed in Table 26.

TABLE 27 Product requirements instantiation criteria

Description	Source
Ability to derive a product requirements instance from the product line requirements infrastructure	3, 4, 5, 6, 7
Ability to change a product requirements instantiation at a later point in time	2, 6
Resolving of variation points check: variations points are resolved correctly in the instantiated product	2, 5
Completeness check: no features are missing	5, 7
Correctness check: no contradicting dependencies present in the product and transitive dependencies are considered	2, 5, 7
Automatic tracing between requirements in instantiated products and the respective requirements in the product line infrastructure	1, 6
Documentation to every product is provided with information such as product description and domain version	4

[1] Beuche et al., 2007; [2] Djebbi et al., 2007; [3] Eriksson et al., 2009; [4] Lisboa et al., 2010; [5] von der Maßen et al., 2004; [6] Schmid et al., 2006; [7] Thurimella et al., 2011

As observed in the table above, the requirements related to the ability to derive products from the product line infrastructure, as well the consistency check of these products, are present in the majority of the consulted literature. This indicates the importance of such features in the tool.

7.3 Application of selected attributes

As a result of the analysis and comparison of presentation and organization aspects of the criteria lists carried out in the previous chapter, dominant attributes were observed among the lists. These observations were taken into consideration for the selection of attributes to be applied to the extended base criteria list obtained from the previous phase.

The complete list of attributes, as well as a summary of the results obtained from the analysis and comparison of the lists' attributes, is summarized in Table 27. This table shows, for all identified attributes, the value found in the majority of the criteria lists, as well as the amount of lists in which this value was encountered. The decision made, on whether the attribute is applied to the new criteria list or not, is included in the table as well. As it can be observed, five of the dominant attributes observed in the currently available criteria lists are applied to the new criteria list for RM tool evaluation, while four of them are not. The adopted attributes and the reason of the adoption are described below.

TABLE 28 Summary of common attributes

Attribute	**Dominant value**	**Amount of lists** (out of 6)	**Applied to new list**
Document format	table	4	yes
Requirement structure levels	3	4	yes
Requirement position in structure	3	4	yes
Requirements' identification	code	4	yes
Requirements' formulation	statement	3	yes
Detail level	medium	4	no
Prioritized requirements	yes	4	no
Priority levels	3	3	no
Additional information	mixed	4	no

The new criteria list is presented in the form of a table (see Figure 14). This format, besides being adopted by the majority of the available criteria lists, allows a clear organization of the list and a direct usage for tool evaluation. As depicted in the figure below, the table consists of five columns, namely ID, description, priority, weight and tool score.

ID	Description	Priority	Weight	Tool Score

FIGURE 14 New criteria list table format

The requirements included in the new criteria list are organized in a hierarchical structure, consisting of three levels. This type of structure, apart from being the most popular among current lists, is considered to improve the list's readability and maintainability (Hoffmann et al., 2004a).

The applied structure is depicted in Figure 15. The first level provides a grouping per meta-category. Meta-categories contain related categories, which are at the second level. Categories group related requirements, which are at the bottom of the hierarchy.

1 **(Meta-category)**
1.1 **(Category)**
1.1.1 (Requirement)
1.1.2 (Requirement)
1.1.3 (Requirement)
1.2 **(Category)**
1.2.1 (Requirement)
1.2.2 (Requirement)
2 **(Meta-category)**
2.1 **(Category)**
2.1.1 …

FIGURE 15 New criteria list structure and requirements' identification

As observed in the figure above, the identification of the requirements in the new criteria list derives from its position in the structure. This approach is also the one adopted by most of the current criteria lists.

The description of the requirements in the new criteria list is done using statements, in accordance with the majority of the analyzed criteria lists. However, an exception is made regarding the level of detail of the requirement's description. As observed from the precise analysis of the lists, a higher level of detail of the requirements' description is found to be more useful for the direct evaluation of features, and therefore is adopted in the new criteria list, instead of the less detailed approach presented by the majority of the lists.

As it can be observed in the results summarized in Table 27, most of the criteria lists apply priorities to the requirements. However, according to Wiegers et al. (2013), the assignment of priorities and weights to requirements in the list should be performed by the user of the criteria list. This is also the recommendation given by Beatty (2013) when presenting the results obtained by Seilevel in the evaluation of RM tools. Beatty affirms that the priorities in Seilevel's list (2011a) are default priorities applied based on the Seilevel's internal use of the RM tool. She suggests that, any organization using this list, should re-prioritize each requirement according the needs of the organization. Based on these recommendations, and considering that the objective of the new criteria list is to be available for use by any organization, no priorities are assigned to the requirements.

The last attribute consists of the inclusion of additional information to the criteria list. A majority of the available criteria lists includes additional information, such as nomenclature, priorities' description, stakeholders, categories' rationale and use cases. However, this information depends on other aspects of the criteria list, and therefore cannot be generalized. Considering the higher level of detail used to describe the requirements included in the list, as well as the clear structure in which requirements are organized, no additional data is included.

This chapter presented the process followed to construct the new criteria list for the evaluation of tools used in distributed SPL requirements

management, describing the sources of content and attributes of the new list. The content of the new list was created taking as a base the list created by Seilevel (2011a), as well as requirements obtained from the literature with the objective of completing the meta-categories “Distributed collaboration” and “Software product line requirements”. The presentation and organization aspects applied to the new criteria list were based on attributes observed in the majority of the analyzed criteria lists and other recommendations from the literature.

8 RESULTS

To accomplish the goals set in this study, three research questions were proposed. Research question one, "*Do currently available criteria lists for RM tool evaluation allow the assessment of tools supporting distributed SPL requirements management?*", required the obtainment of criteria lists for RM tool evaluation available in business and academia. Criteria lists were gathered by means of a literature review. From these criteria lists, only those fulfilling the following requirements were chosen to be included in this study: they must be publicly available in their totality and include requirements to evaluate RM tools. With the given limitation six criteria lists were identified. These lists were presented and discussed in detail in Chapter 4. Chapter 4 introduces also other criteria lists, which are only partially available or have a different scope than RM tool evaluation. These lists were not considered for the analysis and comparison of available criteria lists, but were taken into account during the construction of the evaluation framework in Chapter 5.

The preliminary analysis of the criteria lists aimed at determining whether these lists include requirements to assess tool features supporting distributed collaboration and SPL requirements activities. This analysis clearly showed that none of the six criteria lists includes requirements to assess distributed collaboration and SPL requirements related features. This lack of an integral support required further investigation and an in-depth analysis of the lists content, which led to research question two.

Research question two, "*To what extents do criteria lists support the evaluation of RM tool general features, as well as SPL requirements and distributed collaboration related features?*", required the detailed analysis of the six identified criteria lists. In order to conduct this analysis, an evaluation framework had to be first constructed. This framework consists of a set of attributes, to analyze the criteria lists' organization and presentation characteristics, and a set of categories and scoring system, to classify and rate the lists' content. The set of attributes consist of nine distinctive characteristics observed in current criteria lists; e.g., document format, requirements identification system, and usage of priorities. A set of 36 categories was defined taking as a base the categories of

available criteria lists, RM concepts from the literature, and the results of research conducted in the fields of distributed collaboration of teams and SPL requirements management. The 36 categories were grouped in six meta-categories. The meta-categories were further divided into those including general content, and those including distributed SPL requirements related content. As a final step, a scoring system was defined to rate the content coverage of the criteria lists in terms of amount of requirements per category. This scoring system consisted of a scoring scale ranging from the value "-", indicating the absence of requirements under a category, to the value "+++" indicating a high amount of requirements. The complete evaluation framework is presented in detail in Chapter 5.

The six criteria lists were then analyzed using the evaluation framework. The analysis of the lists' attributes showed that the majority of the lists are presented as a table where requirements are organized in a hierarchy of three levels. The identification of requirements is mainly done with codes resulting from the position of the requirement in the hierarchy. The requirements' descriptions are mainly formulated as statements with an intermediate level of detail. Additionally, requirements are prioritized in the majority of the lists, using three levels of priorities. The detailed analysis of the lists' content confirmed the lack of integral support for distributed collaboration and SPL requirements feature evaluation. The complete results of the analysis and comparison of the attributes and content of available criteria lists are presented in Chapter 6.

The lack of a criteria list supporting the evaluation of distributed SPL requirements management features led to the last research question. Research question 3, "*How would a comprehensive criteria list for the evaluation of tools used in distributed SPL requirements management look like?*", required the creation of a new criteria list for RM tool evaluation.

The starting point for the construction of this new criteria list was the selection of a source list. Based on the results of the lists' analysis, List 6 was chosen due to its higher level of detail and better support of the general content categories. Table 28 presents the total amount of pluses and minuses assigned to the criteria lists under these categories. The absence of minuses in the case of List 6 shows the inclusion of requirements under all general content categories. Additionally, the total of 65 plusses assigned to List 6, represents an 80 percent of the maximum possible amount of plusses. The remaining criteria lists, on the other hand, fail to cover all general categories, as indicated by the presence of minuses. These lists present also a smaller amount of plusses, in comparison to List 6, what translates into a lower level of coverage of the categories.

TABLE 29 Amount of pluses and minuses under general content categories

Criteria Lists	# of pluses	# of minuses
List 1: INCOSE	35	10
List 2: DSTO	47	5
List 3: ITEA DESS	43	6
List 4: DaimlerChrysler	40	7
List 5: RWTH Aachen	40	8
List 6: Seilevel	65	0
Maximum amount possible	81	27

The creation of the new criteria list consisted of three main steps, the reorganization and revision of the source list, the addition of extended categories and the application of selected attributed. These steps are depicted in Figure 16 and explained in more detail in Chapter 7.

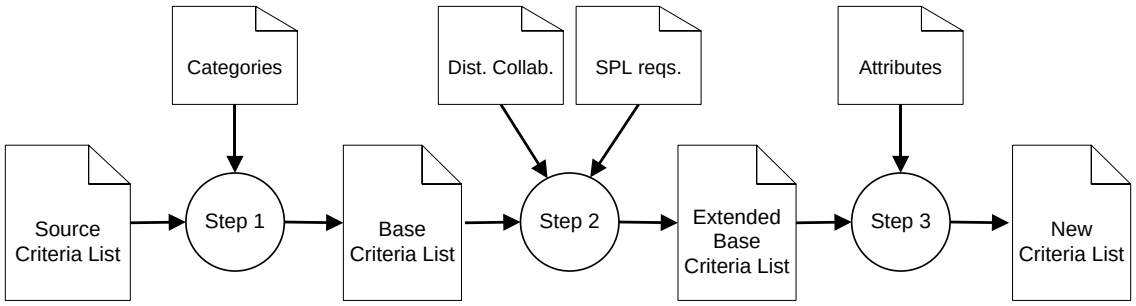

FIGURE 16 New criteria list construction steps

During the first step, requirements in the source list were reorganized according to the set of identified categories. Requirements falling under the out-of-scope categories were removed, while other requirements were revised and reformulated, if necessary, to ensure the lists' independence from specific commercial software products. This revision was considered necessary due to the vendor specificity observed in List 6 (Seilevel, 2011a) and the absence of definitions of the software classes included in the list. The revised general requirements from List 6 resulted in a base list, while distributed SPL requirements related content was considered in step 2.

During the second step, categories under the meta-categories Distributed collaboration and SPL requirements were extended with new requirements. The addition of requirements is based on the lack of comprehensive content observed under the majority of these categories in the source list. A total of 21 requirements were added to the distributed collaboration categories, 5 of which were obtained from the source list. SPL requirements categories were filled

with 32 new requirements, obtained only from published research. The resulting extended categories were added to the base list forming the extended base criteria list.

As a last step, selected attributes were applied to the extended base criteria list. The selection of the applied attributes was based on predominant characteristics observed in the available criteria lists, as well as recommendations found in the literature. The objective behind each selected attribute was the production of a clear list, with a suitable format for direct tool evaluation. As a result of this process a new criteria list for the evaluation of RM tool with support for distributed SPL requirements management was constructed. The first page of this new criteria list is displayed in Figure 17, which is presented fully in Appendix 3.

ID	Description	Priority Weight		Tool Score
1	**RM main activities and baselining**			
1.1	**Version control**			
1.1.1	Reproduce an earlier version of requirements exactly as they were			
1.1.2	Roll back to a prior version of a requirement			
1.2	**Change control**			
1.2.1	Tool can distinguish between formal and informal changes (for example, informal changes might be changes made between formal baselines)			
1.2.2	Filter for a list of new/changed requirements and review the changes			
1.2.3	Automatically maintain audit trail for requirement changes (user, time/date, annotation of change, and change detail)			
1.2.4	Notify affected project participants by e-mail about requirement changes			
1.2.5	Link email to a specific version of a requirement in the change history			
1.2.6	History of a requirements changes is easy to view			
1.3	**Status tracking**			
1.3.1	Filter a view of requirements by criteria (e.g., view all requirements in Draft status)			
1.3.2	Setup complex attributes with predefined value choices (e.g. custom status values)			
1.3.3	Edit requirements' status			
1.4	**Tracing**			
1.4.1	Trace requirements to existing modelling documents			
1.4.2	Link requirements to a full model			
1.4.3	Link client documentation to one or more requirements			
1.4.4	Link requirements to documents that are not stored on a local drive (e.g., store linked document in a database)			
1.4.5	Link requirements to actual documents stored in a content management software[1] (e.g. MS SharePoint[2])			
1.4.6	Link notes (created in another tool) from gathering sessions to individual requirements			
1.4.7	Link emails to specific requirements (Someone will send me a request via email and I will make the change. I would like to link the email to the requirements so that I can easily refer to			

FIGURE 17 New criteria list for RM tool evaluation

The criteria list is presented in the form of a table with five columns. The first two columns contain the requirements ID and description. The last three columns, left empty, allow the user the assignation of priorities and weights to each requirement, and to insert the tool score obtained. The calculation of the tool score, based on the requirement's priority and weight, allows the user of

the list to use it directly for the tool selection process presented by Wiegers et al. (2013) and described in Chapter 3.

Requirements in the list are organized in three levels of hierarchy. The first level consists of the meta-category, the second is the category and the third the requirement. In the figure, the fist meta-category of the list and four of its categories can be observed. Requirements are described as statements and are identified trough a code. The code is based in the position of the requirement in the meta-category and category. This code allows for a fast identification of the requirement's meta-category and category.

9 DISCUSSION AND CONCLUSIONS

The main goals of this thesis were to determine whether currently available criteria lists for RM tool evaluation allow the assessment of features supporting distributed SPL requirements activities; and, if this does not hold true, to construct a new list permitting such evaluation. To achieve these goals, an evaluation framework was designed, available criteria lists were analyzed and compared, and a new criteria list for the evaluation of RM supporting distributed SPL requirements was constructed. These results constitute the major contributions of this research and aim at helping requirements practitioners during RM tool selection with a consequent saving of resources.

During the last decades a series of criteria lists have been created rather independently from each other, using different approaches to present and organize their requirements. However, no tool has been created to support the selection of one of these lists. Therefore, in order to help the analysis and comparison of these lists, an evaluation framework was designed in this research. This framework not only guides the assessment of specific lists' attributes, but also provides a unified set of categories to classify and estimate the content of the list under meaningful sets of related categories. It was built based on the available criteria lists, published research and literature, with the objective to ensure its objectivity and its applicability to the analysis and comparison of other criteria lists created from now on. This framework serves also as a guideline for the creation of new criteria lists, by providing a unified set of interrelated categories and a set of attributes that can be used as a guide.

The results of the analysis and comparison of the available criteria lists provides useful information to requirements practitioners about the attributes and amount of content of these lists. The classification of the lists' requirements into a unified set of categories, allows the user of the list to identify those sets of features that can be better evaluated by the different criteria lists. Such a comparison of current criteria lists was not available in industry or academia until today. The comparison's results help team members in charge of the selection of RM tools that, having decided not to create a new criteria list, need to select between the criteria lists available at present.

Teams developing SPLs are often distributed (Beuche et al., 2006), therefore, the criteria list used to evaluate RM tools should contemplate not only SPL requirement related features, but also distributed collaboration features. However, available criteria lists do not allow an integral evaluation of both groups of features. The new criteria list for RM tool evaluation, most important contribution of this research, fills this gap observed in current research and available criteria lists by allowing such evaluation. Considering that the reduction of costs and of time to market are between the main motivations of SPL development (Pohl et al., 2005) and that the creation a new criteria list consumes these two valuable resources (Carvallo et al., 2005), the usage of this new criteria list becomes of great value to distributed teams managing SPL requirements.

9.1 Critical point of view

Taking into consideration the scope and limitations of this research, some critical aspects should be discussed. As presented earlier, the type of analysis selected to assess the content of the available criteria lists was quantitative, namely based on the amount of requirements each list contains for the set of identified categories. This approach was selected, instead of a qualitative analysis, because it allowed a clear overview of the amount of content of the lists and facilitated the comparison among lists. It could be argued that a qualitative analysis would allow a more in-depth examination and understanding of the lists' requirements, on the expense of more time and resources.

Another aspect worth of attention is the decision to consider only complete criteria lists available to the public as apt for analysis. The reason behind this was to focus on lists that are actually at hand and can be used by practitioners without further difficulties. However, the literature review revealed the existence of other well constructed lists, such as the ones created by Heindl et al. (2006) and Carvallo et al. (2004), which could have enriched this research. Unfortunately, these lists were not available to the public and were published only partially in academic outlets, making them unfit to the purpose of this research.

As presented in the results of this study, the creation of the new criteria list was based on a currently available criteria list, developed by a company with expertise in the field of RM tool evaluation. Besides, this list was extended with validated research published in the fields of collaboration of distributed team members and RM activities during the development of SPL. However, a last point of criticisms to the present research could be that the lack of validation of the new criteria list as a whole, for example by its application for the evaluation of actual RM tools. This decision originated on time restraints due to the extent of the research.

9.2 Suggestions for further study

Several future study directions can be suggested on the basis of the findings, conclusions, and also the limitations of the present study.

The first suggestion consists of the evaluation of the content of available criteria lists using a qualitative assessment of the requirements coverage, instead of a quantitative approach, as in this research. This qualitative assessment would require a more profound analysis of the requirements' level of detail and characteristics, and an extensive knowledge of actual RM tool capabilities. The results of such qualitative assessment could be used to validate the results of the lists' content evaluation done in this research. Additionally, it could show whether the bigger amount of requirements in a list translates into a better coverage of the category, as affirmed in this research.

The second suggestion is the validation of the new criteria list for RM tool evaluation. This validation could be conducted by using the new criteria list for the evaluation and later selection of actual RM tools. An alternative would be the comparison of this list's evaluation capabilities against other criteria lists, as a way to validate its comprehensiveness.

Taking into account the extensive content of the created list, another suggestion would be the study of potential ways of making the usage of the list easier. This could be attained by applying a more graphical representation of the list, or by developing a software tool that, by using the criteria list as a parameter, can guide users during tool evaluation.

The last recommendation consists of the addition of further information to the list. A possibility would be to follow the approach presented by Hoffmann et al. (2004a), where the role or stakeholder of the requirements is specified. Another interesting addition could be a multiple prioritization approach, providing different sets of priorities to the requirements depending on the potential area or field where the tool is used.

REFERENCES

Abma, B. J. M. (2009). *Evaluation of requirements management tools with support for traceability-based change impact analysis*. Master's thesis in Software Engineering. University of Twente

Aiello, M., Bulanov, P. & Groefsema, H. (2010). Requirements and Tools for Variability Management. In *Proceedings of the 34th Annual IEEE Computer Software and Applications Conference Workshops* (pp.245-250). Los Alamitos: IEEE Computer Society

Alexander, I. (2014, June 30). *Requirements Tools*, Scenario Plus. Accessed July 10, 2014 http://www.scenarioplus.org.uk/index.htm

Alexander, I. & Beus-Dukic, L. (2009). *Discovering Requirements, How to specify products and services*. Chichester: John Wiley & Sons, Ltd.

Alghazzawi, D. M., Siddiqui, S. T., Bokhari, M. U. & Hamatta, H. S. A. (2014). Selecting Appropriate Requirements Management Tool for Developing Secure Enterprises Software. *International Journal of Information Technology and Computer Science*, 4, pp. 49-55

Beatty, J. (2013). Winning the Hidden Battle: Requirements Tool Selection and Adoption. In *Proceedings of the 21st IEEE International Requirements Engineering Conference* (pp. 364-365). Piscataway: IEEE Computer Society

Beuche, D., Birk, A., Dreier, H., Fleischmann, A., Galle, H., Heller, G., Janzen, D., John, I., Kolagari, R.T., von der Massen, T. & Wolfram, A. (2007). Using Requirements Management Tools in Software Product Line Engineering: The State of the Practice. In *Proceedings of the 11th International Software Product Line Conference* (pp. 84-96). Los Alamitos: IEEE Computer Society

Beuche, D., Birk, A., Dreier, H., Fleischmann, A., Galle, H., Heller, G., Janzen, D., John, I., Kolagari, R.T., von der Massen, T. & Wolfram, A. (2006). *Using Requirements Management Tools for Product Line Engineering*, Report of the GI Work Group. Aachen University of Technology, Department of Computer Science

Beuche, D. (2008). Modelling and Building Software Product Lines with pure::variants. In *Proceedings of the 12th International Software Product Line Conference* (pp. 358). Los Alamitos: IEEE Computer Society

Birk, A. & Heller, G. (2014, February). *List of Requirements Management Tools*, The Making of Software. Accessed July 10, 2014 http://makingofsoftware.com/resources/list-of-rm-tools

Bokhorst, L. (2001). *Task 1.8 Requirements Management Method D1.8.3 – RM Tools proposal*, Information Technology for European Advancement – Software Development Process for Real-Time Embedded Software Systems (ITEA DESS) Documentation. Available: http://www.dess-itea.org/deliverables/ITEA-DESS-D183-V02P.pdf

Bosch, J. (2013). Software Product Line Engineering. In R. Capilla, J. Bosch & K. C. Kang (Eds.), *Systems and Software Variability Management: Concepts, Tools and Experiences* (pp. 3-24). Berlin: Springer

Bosch, J. & Bosch-Sijtsema, P. (2010). From integration to composition: On the impact of software product lines, global development and ecosystems, *The Journal of Systems and Software,* 83, pp. 67–76

Carrillo de Gea, J. M., Nicolás, J., Fernández Alemán, J. L., Toval, A., Ebert, C. & Aurora Vizcaíno, A. (2010). Requirements Engineering Tools, *IEEE Software, 28 (4),* pp. 86-91

Carvallo, J., Franch, X. & Quer, C. (2005). A Quality Model for Requirements Management Tools. In J. L. Maté & A. Silva (Eds.), *Requirements Engineering for Sociotechnical Systems* (pp. 119-138). Hershey: Information Science Publishing

Chemuturi, M. (2013). *Requirements Engineering and Management for Software Development Projects* (1st ed.). London: Springer

Damian, D. E. & Zowghi, D. (2003). RE challenges in multi-site software development organizations, *Requirements Engineering, 8*, pp. 149–160

Damian, D., Marczak, S. & Kwan, I. (2012). Practice: Requirements Engineering in Global Teams. In Ebert, C., *Global Software and IT: A Guide to Distributed Development, Projects, and Outsourcing* (pp. 257-267). Hoboken: John Wiley & Sons, Inc.

Derakhshanmanesh, M., Fox, J. & Ebert, J. (2014). Requirements-driven incremental adoption of variability management techniques and tools: an industrial experience report, *Requirements Engineering, 19 (4)*, pp. 333-354

Djebbi, O., Salinesi, C. & Fanmuy, G. (2007). Industry Survey of Product Lines Management Tools: Requirements, Qualities and Open Issues. In *Proceedings of the 15th IEEE International Requirements Engineering Conference* (pp. 301-306). Los Alamitos: IEEE Computer Society

Ebert, C. (2012). Requirements Engineering. In *Global Software and IT: A Guide to Distributed Development, Projects, and Outsourcing* (pp. 39-44). Hoboken: John Wiley & Sons, Inc.

Eriksson, M., Börstler, J. & Borg, K. (2009). Managing requirements specifications for product lines – An approach and industry case study, *The Journal of Systems and Software, 82*, pp. 435–447

Gabb, A. P., Maheswaran, N. & Allwright, A. M. (1997). *Requirements Management Tools Evaluation User Needs and Evaluation Criteria* (DSTO-GD-0139). Information Technology Division Electronics and Surveillance Research Laboratory, Australian Department of Defence. Available: http://www.dsto.defence.gov.au/publications/1974/DSTO-GD-0139.pdf

Gotel, O. & Mäder, P. (2009a). How to Select a Requirements Management Tool: Initial Steps. In *Proceedings of the 17th IEEE International Requirements Engineering Conference* (pp. 365-367). Los Alamitos: IEEE Computer Society

Gotel, O. & Mäder, P. (2009b). How to Select a Requirements Management Tool: Initial Steps. Available: http://ollygotel.com/downloads/tutorial-slides-how-to-select-a-requirements-management-tool.pdf

Harlea, D. & Greenberg, S. (1998). Using a Groupware Space for Distributed Requirements Engineering. In *Proceedings of the 7th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* (pp.57-62). Los Alamitos: IEEE Computer Society

Heimbürger, A., Kiyoki, Y. & Ylikotila, T. (2011). *Communication Across Cultures in the Context of Multicultural Software Development*. University of Jyväskylä, Department of Mathematical Information Technology

Heindl, M., Reinisch, F., Biffl, S. & Egyed, A. (2006). Value-Based Selection of Requirements Engineering Tool Support. In *Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications* (pp. 266-273). Los Alamitos: IEEE Computer Society

Hevner, A. R., March, S. T., Park, J. & Ram, S. (2004). Design Science in Information Systems Research, *MIS Quarterly, 28,* pp. 75-105

Hoffmann, M., Kühn, N., Weber, M. & Bittner, M. (2004a). Requirements for Requirements Management Tools. In *Proceedings of the 12th IEEE International Requirements Engineering Conference* (pp. 301-308). Los Alamitos: IEEE Computer Society

Hoffmann, M., Kühn, N., Weber, M. & Bittner, M. (2004b). Evaluation Template for Requirements Management Tools. Available: http://swt.cs.tu-berlin.de/lehre/eks/RM-Tool%20EvaluationTemplate%20v1.3.1.xls

Hull, E., Jackson, K. & Dick, J. (2011). *Requirements Engineering* (3rd ed.). London: Springer

INCOSE Tools Database Working Group (2010). INCOSE Requirements Management Tools Survey

ISO/IEC TR 24766, (2009). *Information Technology – Systems and Software Engineering – Guide for Requirements Engineering Tool Capabilities* (1st ed.). Geneva: ISO

Jones, P. (2011). Can Requirements Tool Vendors Tell Us About User Needs? In *Proceedings of the 4th International Workshop on Managing Requirements Knowledge* (pp. 31-34), August 30

Kelanti, M., Hyysalo, J., Kuvaja, P., Oivo, M. & Välimäki, A. (2013). A Case Study of Requirements Management: Toward Transparency in Requirements Management Tools. In *Proceedings of the 8th International Conference on Software Engineering Advances* (pp.597-604). IARIA XPS Press

Kitchenham, B. (2004). *Procedures for Undertaking Systematic Reviews*. Joint Technical Report, Keele University, Computer Science Department, (TR/SE-0401) and National ICT Australia Ltd (0400011T.1)

Kotonya, G. & Sommerville, I. (1998). *Requirements Engineering, Processes and Techniques*. Chichester: John Wiley & Sons, Ltd.

Laplante, P. A. (2014). *Requirements Engineering for Software and Systems* (2nd ed.). Boca Raton: CRC Press

Lisboa, L. B., Garcia, V. C., Lucrédio, D., de Almeida, E. S., de Lemos Meira, S. R. & de Mattos Fortes, R. P. (2010). A systematic review of domain analysis tools, *Information and Software Technology, 52,* pp. 1–13

Peffers, K., Tuunanen, T., Rothemberger, M. A. and Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research, *Journal of Management Information Systems, 24(3)*, pp. 45-77

Pohl, K., Böckle, G. & van der Linden, F. (2005). *Software Product Line Engineering: Foundations, Principles, and Techniques*. Heidelberg: Springer

Pohl, K. & Rupp, C. (2011). *Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam - Foundation Level - IREB compliant*. Santa Barbara: Rocky Nook

Portillo-Rodríguez, J., Vizcaíno, A., Piattini, M. & Beecham, S. (2012). Tools used in Global Software Engineering: A systematic mapping review, *Information and Software Technology, 54,* pp. 663–685

Poth, A. (2009). Product Line Requirements Engineering in the Context of Process Aspects in Organizations with Various Domains, *Software Process: Improvement and Practice*, *14*, pp. 315–323

Salo, A. & Käkölä, T. (2005). Groupware Support for Requirements Management in New Product Development, *Journal of Organizational Computing and Electronic Commerce, 15(4)*, pp. 253-284

Sadiq, M. & Jain, S. K. (2012). An Insight into Requirements Engineering Processes. In V. V. Das and J. Stephen (Eds.), *Advances in Communication, Network, and Computing* (pp. 313-318). Heidelberg: Springer

Schmid, K. (2014). Challenges and Solutions in Global Requirements Engineering – A Literature Survey. In D. Winkler, S. Biffl, and J. Bergsmann (Eds.), *Software Quality - Model-Based Approaches for Advanced Software and Systems Engineering* (pp. 85-99). Berlin: Springer

Schmid, K., Krennrich, K. & Eisenbarth, M. (2006). Requirements Management for Product Lines: Extending Professional Tools. In *Proceedings of the 10th International Software Product Line Conference (SPLC'06)* (pp. 113-122). Los Alamitos: IEEE Computer Society

Schwaber, C. & Sterpe, P. (2007). Selecting the Right Requirements Management Tool — or Maybe None Whatsoever. *Forrester Research, Inc.*, September 28

Seilevel (2011a). Requirements Gathering Management Tool Evaluation Worksheet. Available: www.seilevel.com/wp-content/uploads/Requirements-Gathering-Management-Tool-Evaluation-Worksheet.xlsx

Seilevel (2011b). Seilevel Report - Requirements Management Software Evaluation Summary. Available: www.seilevel.com/wp-content/uploads/Seilevel-RequirementsManagementToolEvalResults2.xls

Sinha, V., Sengupta, B. & Chandra, S. (2006). Enabling Collaboration in Distributed Requirements Management. *IEEE Software, 23(5),* pp. 52-61

Smite, D. (2006). Requirements Management in Distributed Projects, *Journal of Universal Knowledge Management*, 1(2), pp. 69-76

Sutherland, M. (2002). Product Line Requirements Management. In *Proceedings of the 12th Annual International Symposium of the International Council On Systems Engineering (INCOSE)*

Thurimella, A. K. & Janzen, D. (2011) metadoc Feature Modeler: A Plug-in for IBM Rational DOORS. In *Proceedings of the 15th International Software Product Line Conference.* Los Alamitos: IEEE Computer Society

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, pp. 1065-1075

Volere (2014, May). *Requirements Tools*, Volere – Requirements Resources. Accessed July 10, 2014 http://www.volere.co.uk/tools.htm

Von der Maßen, T. & Lichter, H. (2004). RequiLine: A Requirements Engineering Tool for Software Product Lines. In F. van der Linden (Ed.), *Software Product-Family Engineering, Lecture Notes in Computer Science* (pp. 168–180). Berlin: Springer

Vrinat, M. (2010). *Integrating Requirements Management Across Product Development: Scorecard on Requirements Management Tools*. Collaborative Product Development Associates (CPDA). Available: www.nohau.se/file/cpda-analyst-report-report.pdf

Weber, M. & Weisbrod, J. (2003). Requirements Engineering in Automotive Development: Experiences and Challenges, *IEEE Software*, *20(1)*, pp. 26-24

Webster, J. & Watson, R.T. (2002). Analyzing the past to prepare for the future: Writing a literature review, *MIS Quarterly, 26 (2),* pp. R13-R23

Wiegers, K. E. (2006). *More about Software Requirements: Thorny Issues and Practical Advice*. Redmond, Washington, USA: Microsoft Press

Wiegers, K. (1999). Automating Requirements Management, *Software Development, 7*(7), pp. 1-5

Wiegers, K. & Beatty, J. (2013). *Software Requirements*, *third edition.* Redmond, Washington, USA: Microsoft Press

Young, R. R. (2004). *The Requirements Engineering Handbook*. Norwood: Artech House

Zainol, A. & Mansoor, S. (2011). An Investigation of a Requirements Management Tool elements. In *Proceedings of the 2011 IEEE Conference on Open Systems*

APPENDIX 1 CRITERIA CATEGORIES IDENTIFIED IN THE LITERATURE AND CATEGORY NAMES ADOPTED

Category name adopted	Criteria categories identified in the literature	Source
RM main activities and baselining		
Version control	Version control	Wiegers et al., 2013
	Version control	Zainol et al., 2011
Change control	Change management and comments	Beuche et al., 2006
	Change control	Bokhorst, 2001
	Change control	Gabb et al., 1997
	Change management and comments	Hoffmann et al., 2004a
	Change control	Wiegers et al., 2013
	Change control	Zainol et al., 2011
Status tracking	Status tracking	Wiegers et al., 2013
	Status tracking	Zainol et al., 2011
Tracing	Traceability	Beuche et al., 2006
	Traceability	Bokhorst, 2001
	Support for linking and decomposition activities	Gabb et al., 1997
	Links (traceability)	Gabb et al., 1997
	Traceability	Hoffmann et al., 2004a
	Requirements flowdown	INCOSE, 2010
	Link management	INCOSE, 2010
	Traceability	Seilevel, 2011a
	Tracing	Wiegers et al., 2013
	Requirements tracing	Zainol et al., 2011
Baselining	Baselining	Beuche et al., 2006
	Managing multiple baselines	Gabb et al., 1997
	Working with separate databases	Gabb et al., 1997
	Baselining	Hoffmann et al., 2004a
	Baselines	Seilevel, 2011a
	Baselining	Zainol et al., 2011
Information management		
Views	Views	Beuche et al., 2006
	Views	Gabb et al., 1997
	Views	Hoffmann et al., 2004a
	Custom views	Seilevel, 2011a
Navigation	Navigation	Gabb et al., 1997
	Navigate	Seilevel, 2011a
Analysis functions	Analysis functions	Beuche et al., 2006
	Traceability analysis	Bokhorst, 2001
	Analysis functions	Hoffmann et al., 2004a
	Traceability analysis	INCOSE, 2010
Import	Parse and import requirements from other documents	Gabb et al., 1997
	Import	Seilevel, 2011a

Category name adopted	Criteria categories identified in the literature	Source
Export	Export	Seilevel, 2011a
Specification generation	Document generation Documenting and reporting ability Specification generation Document generation Documents and other output media	Beuche et al., 2006 Bokhorst, 2001 Gabb et al., 1997 Hoffmann et al., 2004a INCOSE, 2010
Reporting	Documenting and reporting ability Reporting Documents and other output media Reports	Bokhorst, 2001 Gabb et al., 1997 INCOSE, 2010 Seilevel, 2011a
Users and access control	Users, roles and rights User identification and access control User access control Users, roles and rights Permissions	Beuche et al., 2006 Bokhorst, 2001 Gabb et al., 1997 Hoffmann et al., 2004a Seilevel, 2011a
Offline use	Checking out for offline use Checking out for offline use Offline/Online	Beuche et al., 2006 Hoffmann et al., 2004a Seilevel, 2011a
User Interface	User interfaces	INCOSE, 2010
Usability	Usability Usability	Gabb et al., 1997 Seilevel, 2011a
Requirements' data management		
Requirements architecture	Information model ID fields and numbering Attributes Information model Requirements architecture	Beuche et al., 2006 Gabb et al., 1997 Gabb et al., 1997 Hoffmann et al., 2004a Seilevel, 2011a
Requirements capture	Creation, deletion and modification of requirements Capturing requirements/identification	Gabb et al., 1997 INCOSE, 2010
Req. edition and deletion	Creation, deletion and modification of requirements Reviews Edits	Gabb et al., 1997 Seilevel, 2011a Seilevel, 2011a
Requirements quality	Word proofing	Gabb et al., 1997
Requirements enrichment	Formatting, multimedia and external files Special data formats Formatting, multimedia and external files	Beuche et al., 2006 Gabb et al., 1997 Hoffmann et al., 2004a
Requirements issue tracking	Issues tacking system	Portillo-Rodríguez et al., 2012
Technical specification, licensing and support		
Tool integration	Tool integration Integration to other tools Tool integration Interfaces to other tools Integration	Beuche et al., 2006 Bokhorst, 2001 Hoffmann et al., 2004a INCOSE, 2010 Seilevel, 2011a

Category name adopted	Criteria categories identified in the literature	Source
Extensibility	Customizing ability Extensibility Extensibility Extensibility	Bokhorst, 2001 Beuche et al., 2006 Hoffmann et al., 2004a Seilevel, 2011a
System specification	System requirements Platform Computer environment System environment Technology	Bokhorst, 2001 Gabb et al., 1997 Gabb et al., 1997 INCOSE, 2010 Seilevel, 2011a
Licensing	Support and maintenance Pricing	INCOSE, 2010 Seilevel, 2011a
Training and tool help	Technical support Support and maintenance Training Help	Gabb et al., 1997 INCOSE, 2010 INCOSE, 2010 Seilevel, 2011a
Out of scope requirements		
Testing	Support for testability Testing	Bokhorst, 2001 Seilevel, 2011a
Modelling	Models	Seilevel, 2011a
Tender evaluation support	Support for tender evaluation	Gabb et al., 1997
Distributed collaboration		
Informal communication	Synchronous and asynchronous collaboration (Informal) communication Informal collaboration	Harlea et al., 1998 Portillo-Rodríguez et al., 2012 Sinha et al., 2006
Awareness	Awareness Awareness Awareness	Damian et al., 2012 Portillo-Rodríguez et al., 2012 Sinha et al., 2006
Workflow management	Workflow management Process support Workflow management Process support Workflow Workflow	Beuche et al., 2006 Harlea et al., 1998 Hoffmann et al., 2004a Kelanti et al. (2013) Schwaber et al., 2007 Seilevel, 2011a
Software product line requirements		
SPL requirements management	Relating features and requirements Partitioning of requirements base Managing PLs and their members	Eriksson et al., 2009 Schmid et al., 2006 von der Maßen et al., 2004
Domain requirements representation	Feature meta-modelling Feature modelling Domain analysis Representation of variability in PL Variability modelling (by decision models) Support for feature modelling	Djebbi et al., 2007 Eriksson et al., 2009 Lisboa et al., 2010 Schmid et al., 2006 Schmid et al., 2006 von der Maßen et al., 2004

Category name adopted	Criteria categories identified in the literature	Source
Product requirements instantiation	Application modelling	Djebbi et al., 2007
	Product instantiation	Eriksson et al., 2009
	Product derivation	Lisboa et al., 2010
	Instantiation support	Schmid et al., 2006
	Evolution of instantiation	Schmid et al., 2006
	Product configurator	von der Maßen et al., 2004

APPENDIX 2 COMPARISON OF CRITERIA LISTS

	List 1: INCOSE	List 2: DSTO	List 3: ITEA DESS	List 4: Daimler Chrysler	List 5: RWTH Aachen	List 6: Seilevel
RM main activities and baselining						
Version control	+	+	+	++	+++	+
Change control	+	++	++	+++	+++	+++
Status tracking	-	+	+	-	-	++
Tracing	++	+++	+++	+++	+++	+++
Baselining	+	+	++	+	+	++
Information management						
Views	+	+++	++	+++	+++	+++
Navigation	+	+++	+	+	+	+++
Analysis functions	++	++	++	++	+	+
Import	+++	++	+++	+	++	++
Export	-	+	+++	-	+	+++
Specification generation	++	+++	+	+++	+++	++
Reporting	++	+++	++	+	+	+++
Users and access control	++	+++	+++	+++	++	++
Offline use	-	-	+	+	+	++
User interface	++	+++	++	+	+	+++
Usability	-	++	-	-	-	+
Requirements' data management						
Requirements' architecture	+	+++	+++	+++	+++	+++
Requirements' capture	+	+	+	-	-	+++
Reqs. edition and deletion	-	++	++	+	-	+++
Requirements' quality	+	+	-	+	-	++
Requirements' enrichment	-	+++	-	++	++	+++
Requirements' issue tracking	-	-	-	-	-	++
Technical specification, licensing and support						
Tool integration	++	-	+++	+++	+++	+++
Extensibility	++	-	++	++	+++	+
System specification	+++	+++	+++	+++	+++	+++
Licensing	++	-	-	-	-	++
Training and tool help	+++	+	-	-	-	+++
Out of scope requirements						
Testing	-	-	+	-	-	+++
Modelling	-	-	-	-	-	+++
Tender evaluation support	-	+	-	-	-	+

	List 1: INCOSE	List 2: DSTO	List 3: ITEA DESS	List 4: Daimler Chrysler	List 5: RWTH Aachen	List 6: Seilevel
Distributed collaboration						
Informal communication	-	-	-	+	+	+
Awareness	-	+	-	-	-	-
Workflow management	-	-	-	+	+	++
Software product line requirements						
SPL reqs. management	-	-	-	-	+	-
Domain reqs. representation	-	-	+	-	+	-
Product reqs. instantiation	-	-	+	-	+	-

APPENDIX 3 NEW CRITERIA LIST FOR RM TOOL EVALUATION

ID	Description	Priority	Weight	Tool Score
1	**RM main activities and baselining**			
1.1	**Version control**			
1.1.1	Reproduce an earlier version of requirements exactly as they were			
1.1.2	Roll back to a prior version of a requirement			
1.2	**Change control**			
1.2.1	Tool can distinguish between formal and informal changes (for example, informal changes might be changes made between formal baselines)			
1.2.2	Filter for a list of new/changed requirements and review the changes			
1.2.3	Automatically maintain audit trail for requirement changes (user, time/date, annotation of change, and change detail)			
1.2.4	Notify affected project participants by e-mail about requirement changes			
1.2.5	Link email to a specific version of a requirement in the change history			
1.2.6	History of a requirements changes is easy to view			
1.3	**Status tracking**			
1.3.1	Filter a view of requirements by criteria (e.g., view all requirements in Draft status)			
1.3.2	Setup complex attributes with predefined value choices (e.g. custom status values)			
1.3.3	Edit requirements' status			
1.4	**Tracing**			
1.4.1	Trace requirements to existing modelling documents			
1.4.2	Link requirements to a full model			
1.4.3	Link client documentation to one or more requirements			
1.4.4	Link requirements to documents that are not stored on a local drive (e.g., store linked document in a database)			
1.4.5	Link requirements to actual documents stored in a content management software[1] (e.g. MS SharePoint[2])			
1.4.6	Link notes (created in another tool) from gathering sessions to individual requirements			
1.4.7	Link emails to specific requirements (Someone will send me a request via email and I will make the change. I would like to link the email to the requirements so that I can easily refer to			

[1] Content Management Software – "computer application that allows publishing, editing and modifying content, organizing, deleting as well as maintenance from a central interface" (Wikipedia, 2015)

[2] https://products.office.com/en/SharePoint/sharepoint-2013-overview-collaboration-software-features

ID	Description	Priority	Weight	Tool Score
	the request the person sent.)			
1.4.8	Create links between requirements of the same type			
1.4.9	Create links between requirements of different types			
1.4.10	Display any kind of traceability results in a table			
1.4.11	Display traceability results in a diagram (e.g., a tree)			
1.4.12	Describe the nature of the relationship link between requirements (whether it is a "traces from", "dependency", etc.)			
1.4.13	Create links between defined groups of requirements			
1.4.14	Link requirements to those of another requirement type such as business objectives			
1.4.15	Create links between many requirements all at once (e.g. preselect a set and link them all individually to another requirement in a mass update)			
1.4.16	Can create a custom tracing model to link requirements to user-defined types of artefacts (e.g. link requirements to compliance documents and enforce a trace model)			
1.4.17	The tool must be able to enforce the creation and change of certain types of traceability links (links can be setup so no requirement is left stranded)			
1.4.18	Tool supports M:N bidirectional linking			
1.5	**Baselining**			
1.5.1	Create baselines of the requirements			
1.5.2	Compare baselines of the requirements			
1.5.3	Work with specific baseline of the requirements			
1.5.4	Need to be able to baseline so we can show the difference, and what is new between the last version and this version			
2	**Information management**			
2.1	**Views**			
2.1.1	Sort view of requirements by multiple criteria at once (e.g., view all requirements in ascending order by creation date)			
2.1.2	Save private custom views			
2.1.3	Save public custom views			
2.1.4	Filtered view applies instantly			
2.1.5	Apply filters to custom attributes of the requirements objects in addition to out of the box attributes (I have a set of requirements that can be functionally divided (i.e. tax/shipping/etc) but can also be divided by region (US, EMEA, APAC) and want to be able to view the requirements in the way that is convenient to me.)			
2.1.6	View requirements as a list to make it easy to scan many requirements at once and change fields directly in that list			
2.1.7	Display customized view of traceability data in a table where user can select which types of objects show up in the columns			
2.1.8	Show the requirements in a hierarchy			
2.2	**Navigation**			
2.2.1	Search requirements by keyword			
2.2.2	Navigate the requirements hierarchy by going up and down and across branches			
2.2.3	For an individual requirement, can visually emphasize where in the requirements hierarchy it sits			

ID	Description	Priority	Weight	Tool Score
2.2.4	In lists of requirements, can scan the list and see old versions of the requirement in the list			
2.2.5	Provide drag-and-drop ability to move requirements within the hierarchy			
2.2.6	Navigate traceability hierarchy easily			
2.3	**Analysis functions**			
2.3.1	Traceability analysis to identify missing links within the requirements (e.g. functional requirement orphans not linked to a use case)			
2.3.2	Tool detects inconsistencies in links (e.g., circular linking or linking in the "wrong" direction in the hierarchy)			
2.4	**Import**			
2.4.1	Bulk enter requirements in the tool directly			
2.4.2	Automatically identify requirements from external text document by key words, structure, etc. (e.g. specify what keywords to search for in the doc and import based on those)			
2.4.3	Batch import structured data as new requirements from spreadsheet document (e.g., import spreadsheet file of previously developed requirements into the new managed system)			
2.4.4	Batch import data as new requirements from word processing document (e.g., import word file of previously developed requirements into the new managed system)			
2.5	**Export**			
2.5.1	Export requirements to word processing document (e.g., doc or rtf)			
2.5.2	Export requirements to markup language document (e.g., HTML or XML)			
2.5.3	Export requirements to delimited format (e.g., CSV)			
2.5.4	Highlight changes on exported document against a previous baseline			
2.5.5	Export a subset of requirements based on a filter view applied			
2.5.6	Export data from any report to spreadsheet document (e.g., export metrics data to MS Excel[3] for further analysis)			
2.6	**Specification generation**			
2.6.1	Produce requirements documentation in various pre-defined military and commercial standard formats			
2.6.2	Define custom format for documentation output (e.g., provide an SRS[4] template that the system populates)			
2.6.3	Preview of requirements document generation			
2.7	**Reporting**			
2.7.1	Metrics reporting (e.g., progress/status, requirements velocity, requirement changes, tracing metrics, etc.)			
2.7.2	Metrics charting (e.g., colour graph of volatility data over time)			
2.7.3	Report to show any problems with requirement links (e.g.,			

[3] http://products.office.com/EN/excel
[4] Software Requirements Specification

ID	Description	Priority	Weight	Tool Score
	parent requirement is invalid so children are marked as invalid)			
2.7.4	Report on the maturity of requirements by nature of requirements status and number of reviews per requirement			
2.7.5	Report on whether a set of requirements have been reviewed (looked at)			
2.7.6	Report on all current open issues about requirements			
2.7.7	Report on who is working on what requirements			
2.7.8	View dashboard reports			
2.7.9	Customize dashboard reports			
2.7.10	Customize reports to include any attributes of the requirements objects			
2.7.11	Create burndown reports on requirements status			
2.7.12	Generate traceability reports (downward relationship from a given set of requirements, upward relationship from a given requirement			
2.7.13	Report on the number of reviews of requirements or sets of requirements			
2.7.14	Report on number of issues open, closed, in progress, etc. about requirements by requirement or some other attribute			
2.8	**Users and access control**			
2.8.1	Manage access permissions for individual users (e.g., some users can edit requirements, others can only read them)			
2.8.2	Manage access permissions for groups of users			
2.8.3	Restrict requirements from clients based on some attribute of the requirements (e.g. only let them see requirements in a non-draft status)			
2.8.4	Restrict sets of requirements from clients based on project they belong to (e.g. only let them see the requirements for their project)			
2.9	**Offline use**			
2.9.1	Provide ability to work disconnected (i.e., no connection to requirements repository) and merge changes upon reconnecting (may require user to trigger)			
2.9.2	Create an output from the tool that you can review it and make changes in the output that get put back into the tool			
2.9.3	Automatically synchronizes when coming back online			
2.9.4	Periodically reminds user to synchronize offline work			
2.9.5	Provide ability to work disconnected in read-only view			
2.10	**User interface**			
2.10.1	Edit requirements with undo functionality			
2.10.2	Drag and drop to link requirements to one another (e.g. want to map my requirements to use cases very quickly. I believe that drag and drop to link requirements would be fast)			
2.10.3	Supports cut and paste to and from other applications			
2.10.4	Supports cut and paste between fields within the tool			
2.10.5	Client platform is web interface			
2.10.6	Can the user interface be easily modified to assign functions to buttons, hot keys and menus			
2.10.7	Can macros be easily defined and assigned to buttons, keys and menus			

ID	Description	Priority	Weight	Tool Score
2.10.8	Where user defined enumeration types are used for attributes, can the user "pick" the value desired from a list			
2.10.9	Are short cuts provided for "find again" and "find/replace again			
2.10.10	Edit individual requirements quickly without having to go through multiple clicks to get to an editable view			
2.10.11	Requires only a few clicks to view detailed information about requirements			
2.10.12	Quickly read through many requirements with minimal user interaction (e.g., mouse clicks, keyboard presses, etc.)			
2.11	**Usability**			
2.11.1	Instinctually easy to use			
2.11.2	Easy to learn			
3	**Requirements' data management**			
3.1	**Requirements' architecture**			
3.1.1	Define what data should be captured for each type of requirement (custom data fields)			
3.1.2	Group requirements by project (i.e., system supports multiple projects)			
3.1.3	Define hierarchical relationships between requirements			
3.1.4	Organize requirements by groups (e.g., group by sub-system)			
3.1.5	Define custom format for requirements IDs (e.g., UCXX for use cases, BRXX for business rules, etc.)			
3.1.6	Provide a template for new projects to minimize setup			
3.1.7	Define dependencies between requirements separate from hierarchy of requirements			
3.1.8	Define a glossary for a set of requirements			
3.1.9	Can store fluffy text around the requirements to give people context as to what the purpose of a given set of requirements are or other types of explanation			
3.1.10	Track who originally requested a specific requirement			
3.1.11	Setup complex attributes with predefined value choices (e.g. custom status values)			
3.1.12	Captures what release requirement is slated for			
3.1.13	Capture statements of rationale associated with requirements statements and collections			
3.1.14	Set whether requirement attributes are required or not			
3.1.15	Store a url (or other network path) to a document located in a content management software (e.g., MS SharePoint)			
3.1.16	Defined glossary for a set of requirements allows you to hover or click on keywords to get the definition			
3.1.17	Voting on requirements (priorities)			
3.1.18	Set priority on requirements			
3.1.19	Define and capture different types of requirements (e.g., use cases, functional requirements, etc.)			
3.2	**Requirements' capture**			
3.2.1	Add new requirement			
3.2.2	Enter individual requirements with minimal user interaction (e.g., mouse clicks, keyboard presses, etc. - in a word processor, you can press 'Enter' to enter another requirement)			
3.2.3	Allows a unique ID assignment to each requirement or object			

ID	Description	Priority	Weight	Tool Score
	(could be a manual entry of an ID)			
3.2.4	Can add different types of requirements			
3.2.5	Automatically create unique ID for each requirement			
3.2.6	Assign different requirements groups to different analysts for completion (e.g. owners on requirements or sets of requirements)			
3.2.7	New requirements entered are marked as new for review			
3.2.8	Assign requirements to owners for work or review			
3.2.9	Requirements can be copied from another project			
3.2.10	Define and capture different types of requirements (e.g., use cases, functional requirements, etc.)			
3.2.11	Define a stakeholder for each requirement			
3.2.12	Capture source of requirements			
3.2.13	Requirements can be re-used between projects (without modifications to them)			
3.3	**Requirements edition and deletion**			
3.3.1	Edit requirements			
3.3.2	Delete requirements			
3.3.3	Bulk edit properties, meaning you can select multiple requirements and edit a common property and all items would receive that value			
3.3.4	Edits are updated to the master copy of the data in real-time			
3.3.5	Allow a delete option on requirements that doesn't forever lose the requirement			
3.3.6	Restrict two people from concurrently editing a requirement in online mode			
3.4	**Requirements' quality**			
3.4.1	Automatically check requirements for spelling errors			
3.4.2	Automatically check requirements for grammar usage			
3.4.3	Automatically check requirements for use of ambiguous words			
3.4.4	Mark a requirement as a duplicate of another and merge them together			
3.4.5	Can define validation checks for user defined attributes			
3.5	**Requirements' enrichment**			
3.5.1	Document a requirement with rich text formatting			
3.5.2	Support entering tables within requirements			
3.5.3	When copy/paste into fields from word processing or spreadsheet documents, Tool maintains existing field formatting			
3.5.4	Can add mathematical equations of varying complexity			
3.5.5	Can add symbols and formatting (Greek symbols, subscripts, superscripts)			
3.5.6	Describe a requirement with an image directly in the database (e.g., a bitmap context diagram or MS Visio[5] file image)			
3.5.7	Describe a requirement with an embedded document in the database (e.g., a MS Visio diagram)			

[5] http://products.office.com/en/Visio/flowchart-software

ID	Description	Priority	Weight	Tool Score
3.5.8	Can upload and store audio files about requirements in the tool			
3.5.9	Can link video files related to requirements to one or more requirements			
3.5.10	Can upload and store video files about requirements in the tool			
3.5.11	Can link audio files related to requirements to one or more requirements			
3.6	**Requirements' issue tracking**			
3.6.1	Allow issue creation from the requirement itself and automatically link the issue to the requirement			
3.6.2	Requirement can have a link to an issue in another tool as a URL (e.g. a link to the issue stored in a content management software)			
3.6.3	Tool has issue tracking as part of its functionality			
3.6.4	Tracks resolutions to open issues on requirements			
3.6.5	Tracks open issues by requirement			
4	**Technical specification, licensing and support**			
4.1	**Tool integration**			
4.1.1	Automated integration to issue tracking system[6] (e.g., MS SharePoint)			
4.1.2	Automated integration with design tools (e.g., IBM Rational Rose Modeler[7], Enterprise Architect[8], etc.)			
4.1.3	Automated integration with development environments (e.g., Eclipse[9], MS Visual Studio[10])			
4.1.4	Automated integration with revision control system[11] for any linked documents (e.g., MS SharePoint, TortoiseCVS[12], MS Visual SourceSafe[13], etc.)			
4.1.5	Tool allows integration with note-taking software (e.g., MS OneNote[14])			
4.1.6	Automated integration with test management system[15] (e.g. HP Quality Center[16])			
4.2	**Extensibility**			

[6] Issue Tracking System - "computer software package that manages and maintains lists of issues, as needed by an organization. Issue tracking systems are commonly used in an organization's customer support call center to create, update, and resolve reported customer issues, or even issues reported by that organization's other employees." (Wikipedia, 2015)

[7] http://www-03.ibm.com/software/products/en/rosemod

[8] http://www.sparxsystems.com/products/ea/

[9] https://eclipse.org/ide/

[10] https://www.visualstudio.com/

[11] Revision Control System – "software implementation of revision control that automates the storing, retrieval, logging, identification, and merging of revisions" (Wikipedia, 2015)

[12] http://www.tortoisecvs.org/

[13] https://msdn.microsoft.com/en-us/library/ms950420.aspx

[14] http://www.onenote.com/

[15] Test Management System **–** "tools used to store information on how testing is to be done, plan testing activities and report the status of quality assurance activities" (Wikipedia, 2015)

[16] http://www8.hp.com/us/en/software-solutions/quality-center-quality-management/

ID	Description	Priority	Weight	Tool Score
4.2.1	External API[17] available			
4.2.2	Perform ad-hoc queries of requirement data			
4.3	**System specification**			
4.3.1	Installed central repository (not-hosted)			
4.3.2	Installed client only			
4.3.3	Client platform is web interface (yes or no)			
4.3.4	Client platform support for non-web interface on Windows			
4.3.5	Client platform support for non-web interface on Unix/Linux			
4.3.6	Platform resource requirements (memory, CPU, and disk space, etc.)			
4.3.7	Allow multiple concurrent users			
4.3.8	Server platform support for Windows (e.g., Windows 2000 or XP)			
4.3.9	Server platform support for Unix/Linux			
4.3.10	Database platform support for Windows (e.g., Windows 2000 or XP)			
4.3.11	Database platform support for Unix/Linux			
4.3.12	Database platform support for MS Access			
4.3.13	Database platform support for SQL Server			
4.3.14	Database platform support for Oracle			
4.3.15	Database platform support for MySQL (open-source)			
4.3.16	Database is commercial (not proprietary)			
4.3.17	Supports merging of requirements from different databases			
4.3.18	Ease of installation			
4.3.19	Ease of administration (e.g., database maintenance, installing upgrades, etc.)			
4.3.20	Open-source			
4.3.21	How many customers use the tool today			
4.3.22	Does vendor think it supports agile and waterfall approaches			
4.3.23	The tool scales with increasing number of input requirements (i.e., how does the tool compare with 1000 vs. 15000 vs. 30000 requirements?)			
4.3.24	The number of user defined attributes is not limited			
4.4	**Licensing**			
4.4.1	License policy (site, node locked, floating, etc.)			
4.4.2	License cost			
4.4.3	Warranty options			
4.4.4	Maintenance & upgrade policy			
4.4.5	Maintenance cost			
4.5	**Training and tool help**			
4.5.1	Availability of learning aids (e.g., sample requirements, workflow tutorial, etc.)			
4.5.2	Availability of training classes			
4.5.3	Availability of documentation (either online, soft-copy, or hard-copy)			
4.5.4	Availability of context-sensitive help within the system			
4.5.5	Availability of phone support			
4.5.6	Availability of online support (e.g., message board,			

[17] Application Programming Interface

ID	Description	Priority	Weight	Tool Score
	knowledge base, users group, wiki, etc.)			
5	**Distributed collaboration**			
5.1	**Informal communication**			
5.1.1	Ability to initiate new conversations with other stakeholders			
5.1.2	Availability of synchronous contextual communication around requirements, such as chat, videoconference or VoIP			
5.1.3	Availability of asynchronous (email-like) contextual communication around requirements			
5.1.4	Integration of synchronous and asynchronous communication			
5.1.5	Insertion of navigable links to relevant objects into messages, automatically when the context is obvious, or manually by users			
5.1.6	Ability to access all the discussions a user has participated in			
5.1.7	Ability to edit a conversation's transcripts and save only relevant information			
5.2	**Awareness**			
5.2.1	Information on which users are currently connected to the tool and their status is provided			
5.2.2	Information on the roles of the users in the project is provided			
5.2.3	Information on where other team members are located geographically is provided			
5.2.4	Information on other team members workload is provided, letting know ''who is doing what''			
5.2.5	Visual cues are provided, helping users be aware of the actions recently performed by other users, online information and pending change requests			
5.2.6	Ability to personalize the visual information provided			
5.2.7	E-Mail notifications to affected team members about actions performed by other users			
5.2.8	Ability to subscribe to requirements of interest to receive automatic notification of the requirement's changes			
5.3	**Workflow management**			
5.3.1	Ability to define custom workflows for each type of requirement			
5.3.2	Pre-defined workflows are available			
5.3.3	Ability to define one custom workflow for all types of requirements			
5.3.4	Track requirements approval/signoff			
5.3.5	Request requirements approval/signoff via email			
5.3.6	Ability to define tasks and assign rights to users for each workflow step			
6	**Software product line requirements**			
6.1	**Software product line requirements management**			
6.1.1	Support for multiple projects and products			
6.1.2	Ability to create new product line projects			
6.1.3	Ability to link features to requirements that substantiate these features			
6.1.4	Ability to easily relate features and requirements to the product lines platform or to a product			

ID	Description	Priority	Weight	Tool Score
6.1.5	Ability to control if the affiliation of features and requirements to platform and products is accurate			
6.1.6	Ability to manage features information, such as name, unique identifier, description, rationale, status, source, priority, maturity and history information			
6.1.7	Ability to attach supporting material to features, e.g text documents, table sheets, diagrams, pictures, etc			
6.1.8	Support for multi project and multi product requirement views			
6.1.9	Support for multi project/multi product status and progress reporting			
6.1.10	Ability to hide features and requirements that do not address a product of interest to the current user			
6.1.11	Categorization of findings into errors (severe problems) and warnings (light problems)			
6.1.12	Ability to define whether errors and/or warning should be reported			
6.1.13	Query interface retrieving information of features and requirements related to the product line			
6.1.14	Support for stakeholder collaboration during domain modelling and product instantiation, e.g. using issue modelling			
6.2	**Domain requirements representation**			
6.2.1	Support for feature modelling, or other domain representation approaches, e.g. tables, decision models, etc			
6.2.2	Support for graphical modelling of features			
6.2.3	Relationship types: provides different types of relationships between the features, e.g. composition, generalization/specification and implementation			
6.2.4	Mandatory features: represent the features that will always be in the products			
6.2.5	Variability: represents the variabilities a feature can have (optional, alternative and or)			
6.2.6	Ability to handle commonalities, variabilities and product-specific requirements for individual products			
6.2.7	Feature group identification: classifies the features according to the type of information they represent; e.g. capability, domain technology, implementation techniques and operation environment			
6.2.8	Composition rules: create restrictions in the domain for representing and relating the features, such as mutual exclusion and dependency, regular expressions, or artificial intelligence, among others			
6.2.9	Redundancy check: same concept modelled twice in the model			
6.2.10	Anomaly check: possible instantiations lost due to improper relationships between features			
6.2.11	Consistency check: verifies if the generated domain follows the composition rules created and no conflicting relationships were created in the models			
6.3	**Product requirements instantiation**			
6.3.1	Ability to derive a product instance from the product line			

ID	Description	Priority	Weight	Tool Score
	infrastructure			
6.3.2	Ability to change a product instantiation at a later point in time			
6.3.3	Resolving of variation points check: variations points are resolved correctly in the instantiated product			
6.3.4	Completeness check: no features are missing			
6.3.5	Correctness check: no contradicting dependencies present in the product and transitive dependencies are considered			
6.3.6	Automatic tracing between requirements in instantiated products and the respective requirements in the product line infrastructure			
6.3.7	Documentation to every product is provided with information such as product description and domain version			