Project Scope Management: A Strategy Oriented to the Requirements Engineering

Igor Luiz Lampa, Allan de Godoi Contessoto, Anderson Rici Amorim, Geraldo Francisco Donegá Zafalon, Carlos Roberto Valêncio and Rogéria Cristiane Gratão de Souza

Department of Computer Science and Statistics, São Paulo State University, São José do Rio Preto, Brazil

Keywords: Project Management, Scope Management, Requirements Engineering.

Abstract: Scope management is an area of project management defined by PMBoK, which has processes to register and control everything that belongs to the project boundaries. Although the relevance of this area to the success of projects, its application is still a challenge, which is potentialized by the lack of computational tools to support project management that integrate the scope management totality. In addition, the lack of understanding of project requirements is another factor that hinders the execution of this area, because often the stakeholders do not have full knowledge of their needs at the beginning of the project, resulting in changes throughout the project lifecycle, which increases the costs and deadlines. In this sense, the objective of this work is to propose the integration of the scope management with the requirements engineering, in order to better identify the requirements of a project and to understand what needs to be done, contributing to the success of projects. For the evaluation of the results obtained, a requirements engineering module was developed and integrated with a previously developed computational tool for project management and it aims to assist the application of the project management following the guidelines and good practices proposed on PMBoK.

1 INTRODUCTION

Project management is gaining increasing importance in organizations, as it has become critical to the control and elaboration of business decisions. The Project Management Body of Knowledge – PMBoK, developed by the Project Management Institute – PMI, addresses the best practices that support project management activities and can be used in most projects (PMBoK, 2013).

The PMBoK guide presents project management as a set of ten knowledge areas, being the scope management responsible for delimiting what will be done in the project, defining a group of processes responsible for ensuring that all the work needed, and only what is necessary, to complete the project successfully, is carried out (PMBoK, 2013).

In this context, the objective of the scope management is to control product and project boundaries, which can be a complex activity because the boundaries are not always clear and well defined and may involve political, social, technological, organizational and economic forces (Alexander et al, 2009). It is worth mentioning that small variations in scope can cause costly impacts in different areas, as time, cost and quality (PMBoK, 2013; Badariah *et al*, 2009).

According to Smith (2002) one of the main problems in information technology projects is related to the system requirements. Errors in the requirements are costly and can lead to loss of time, revenue and reputation of the responsible organization. Furthermore, when considering the correction of these requirements when they have already been implemented, the cost associated with correcting errors could generate even greater impacts under the project budget (Badariah *et al*, 2009).

The elaboration of requirements involves several stakeholders that are directly or indirectly affected by the project. These stakeholders have different experiences and expectations with the project. Thus, the requirements analysis process must be performed completely, because these stakeholders may not be able to define exactly what they actually need. So, they can express their needs incompletely, which

Lampa, I., Contessoto, A., Amorim, A., Zafalon, G., Valêncio, C. and Souza, R.
Project Scope Management: A Strategy Oriented to the Requirements Engineering.
DOI: 10.5220/0006318603700378
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 370-378
ISBN: 978-989-758-248-6

increases the probability of failures on project scope (Sommerville, 2011).

In this context, the present work aims to promote the integration of requirements engineering techniques to help organizations better control the scope of their projects. Thus, a new module was developed as an expansion of the System to Aid Project Management – SAPM (Souza *et al*, 2011), previously developed with the objective of supporting the execution of managerial activities according to the good practices presented on PMBoK guide. Therefore, requirements engineering techniques were incorporated into the "Collect Requirements" process of the scope management, in order to allow the evaluation of the adequacy of the obtained results, as well as the identification of possible deficiencies.

2 SCOPE ENGINEERING AND THE REQUIREMENT ENGINEERING

The scope can be divided into product scope and project scope. According to Alexander *et al* (2009), the product scope is related to everything that is within the boundaries of the product, as features, restrictions and details that allow to define the results or product desired in a project. On the other hand, the project scope is related to description of all necessary work to achieve what was defined on the product scope, attending all features and functions previously specified (PMBoK, 2013).

The area of scope management has a group of six processes that, together, aim to register, monitor and control which belongs or do not belongs to the project boundaries, besides managing the product requirements and ensure that the project include all necessary work and just the necessary to the project conclusion (PMBOK, 2013). These processes are: Plan Scope Management, Collect Requirements, Define Scope, Create WBS, Validate Scope and Control Scope.

The scope management is related to other areas of project management, as cost, time and risk management. Thus, when the scope is not well defined, it causes inconsistencies and many failures may happen throughout the project (Bjamason *et* al, 2012; Kumari *et al* 2014). Specifically in software development, Hall *et al* (2003) show that 48% of failures noticed on the project are related to poorly defined requirements.

The requirements represents services, functions, restrictions, features and behaviors that the final project must attend (Kotonya and Sommerville, 1998). The requirements must reflect the necessities and the demand imposed by the stakeholders (Sommerville, 2011). Thus, properly specified requirements describe clearly and accurately the real need of the stakeholders (Pandey *et al*, 2010). This way, it is important that the requirements of the product scope to be consistent with the necessities exposed by the stakeholders.

As mentioned by Ballejos *et al* (2008), for the correct identification of the requirements is necessary that all participators of this process may be committed with the project. Thus, it must be defined a group of stakeholders that can identify the requirement broadly, through their different visions, experiences and expectative over the project (Damian, 2007).

On the scope management, the "Collect Requirements" process assists the stages of requirements identification, management and analysis, and preparation of the final documentation, which will describe the real necessities of the stakeholders over the project. The PMBoK guide shows what activities and documents must be prepared to the conclusion of this process, however it can be noticed the lack of a procedure to assist and show how this stage should be conducted and managed.

This way, the Requirement Engineering (RE) is used in this work as a technique to support the activities of the project scope management. The RE is an area of Software Engineering (SE) that contains sub processes whose final goal is to obtain, uncover and manage the software requirements, besides creating and maintaining a complete documentation of these requirements (Bjamason *et al*, 2012; Ballejos *et al* 2008; Di Thommazo *et al*, 2015; Sommerville, 2011). Thus, the RE is composed by six activities: Feasibility Study, Requirement Elicitation, Requirement Analysis and Negotiation, Requirements Documentation, Requirements Validation and Requirements Management (Sommerville, 2011).

Although the fact of the scope management is essential to conduct the project activities, the lack of mechanisms to support the requirements gathering can result in failures, because when a requirement is documented, it becomes part of the project planning (Di Thommazo *et al*, 2015). Thus, we can notice the importance of the RE to support the scope management, since it helps on reduction of requirement uncertainties and, consequently, failures on the final product.

In this context, as showed by Svahnberg *et al* (2013), the use of RE has presented good impacts in prevention of possible problems on the initial stages of the project, instead of wait they appear when the project is in progress or finished. Thus, the cost of corrections of the requirements is considerably less than when they are corrected on the stage of requirements gathering.

3 DISCUSSION OF RESULTS

The methodology for the development of this work was divided into three stages, for which the corresponding results have been identified, as described in the following subsections.

3.1 Empirical Study

In this stage, the objective was understand the current shortcomings presented by the computational tools available in the market to support the scope management. So, the main tools for the application of this knowledge area were analyzed and evaluated. For the selection of the tools were considered those previously mentioned and evaluated in related works (Souza *et al*, 2011; Contessoto *et al*, 2016), besides tools widely used today: Microsoft Project 2016 - MS (MSProject, 2016), NetProject 2016 - Net (NetProject, 2016), ProjectLibre 2016 – Lib (ProjectLibre, 2016) and Project Planner - Pla (ProjectPlanner, 2016).

The criteria used in the evaluation correspond to the scope management processes presented in PMBoK guide. The results of the analyzes and respective evaluations are presented in Table 1, where the service coverage of the tools with respect to the criteria is represented by percentage.

Through the analysis of the results obtained from the evaluation of the tools it is possible to verify that several processes of the scope management are not contemplated, or are only partially contemplated in the analyzed tools, standing out "Collect Requirements" and "Plan Scope Management" processes. As described by Souza et al (2011) there are several tools to support project management, but most of them have limited focus and scope in relation to the processes of specific areas of PMBoK. Therefore, project managers need different tools for the complete management of a project.

In relation to the scope management, based on the result of the evaluation of the tools presented in Table 1, it is also necessary the integrated use of different support resources to cover the largest number of scope management processes. Consequently, inconsistencies may occur between information from the same project that is managed by different tools.

Table 1: Analyzed tools.

Tools / Processes	Lib (%)	Pla (%)	MS (%)	Net (%)	SAPM (%)
Plan Scope	0	0	0	0	0
Collect Requirements	0	0	0	0	0
Define Scope	50	12,5	25	50	75
Create WBS	50	37,5	75	100	100
Validate Scope	0	0	25	50	50
Control Scope	50	50	62,5	75	75

Considering specifically the analysis involving the “Collect Requirements” process, focus of this work, it can be notice that it is not contemplated by the analyzed tools, although it is a key process for the scope management. Inadequate management of project requirements, aggravated by inefficient collection of requirements, can compromise the real needs of stakeholders. The consequences of developing a project that does not meet the requirements are several, for example, expansion of the planned schedule, increase of costs stipulated in the initial budget due to possible corrections and especially customer dissatisfaction with the requested product (Kotonya and Sommerville, 1998).

From the analysis of computational tools, essential requirements engineering techniques that can be integrated into the scope management were identified, in terms of the addition of functions (F) in a computational tool, like SAPM, aiming its automation:

F1. Project Feasibility Study: This function allows, from basic project information, as the description of the project objectives, of the preliminary scope, of the project contributions, and of the time and cost constraints, verify if it is really valid to proceed with the development of the project;

F2. Requirements specification and documentation: This function enables the identification of product scope requirements, ensuring that they are properly specified and documented. During specification of requirement, information such as the dependency between requirements, as well as the identification of the stakeholder who requested it and the person in charge of the development team, should be stored, as they are crucial information for the automated generation of traceability matrices;

F3. Automation of analysis and negotiation: This function aims to enable the analysis and negotiation of the requirements, after their identification. Thus, the requirements should be analyzed in detail, in order to find conflicts, inconsistencies, lack of information and compatibility with the budget constraints and schedule. As soon as the problems are identified, the negotiation with the stakeholders must take place, with the objective of finding the solution that satisfies the constraints and needs (Simão and Varela, 2009). To support this activity, the used strategy an analysis checklist, which allows a standard check of all the requirements, according to the same criteria;

F4. Automation of requirements validation: This function allows the final validation of a requirement, considering aspects such as compliance with required quality standards, poorly formulated or ambiguous requirements, conflicting requirements, among other aspects that have not been corrected or have not been observed at the moment (Ito *et al* 2011). Therefore, this activity adopts a validation checklist, which allows all requirements to be validated from the same criteria;

F5. Requirements history management: This feature allows manage all the changes that requirements undergo throughout the project lifecycle, once they become available for stakeholder analysis. Managing the requirements is a key function, since the requirements evolve according to changes in the application domain environment and according to understand of the project objectives by the stakeholders. Kotonya and Sommerville (1998) estimated that about 50% of the requirements will be changed until they are effectively used in project development;

F6. Management of exclusions: This function allows the management of the historical basis of the excluded requirements, being possible to consult its specifications, as well as the reason for the requirement to have been excluded from the project. This function is important for the development of similar projects, since the excluded requirements can be consulted as lessons learned;

F7. Automated generation of traceability matrices: This function allows the creation of traceability matrices that present the dependencies between the requirements. According to Kotonya and Sommerville (1998), a critical part of the requirements changes management process and, consequently, the management of the requirements history, is the evaluation of the impact of the change under the rest of the requirements and parts that make up the project. Consequently, it is essential to ensure the traceability of requirements and to provide practical ways of understanding the relationship between them, and therefore, facilitate the verification of impacts that changes may cause on other requirements. Thus, three types of matrices are considered: “requirement X requirement”, “requirement X stakeholder” e “requirement X responsible team member”.

The traceability matrix "requirement X requirement" allows check which requirements are related to other requirements. In this way, it is possible to evaluate the impact that the change or exclusion of some requirement will cause in the requirements with dependence. The traceability matrix "requirement X stakeholder" allows verify what are the requirements requested by a particular stakeholder, providing a global view of the stakeholders who requested the most requirements and, therefore, need to follow the development of the project in a more active way. Finally, the traceability matrix "requirement X responsible team member" allows verify those responsible for managing each of the project requirements, allowing the overall view of allocation of responsibility to the members of the development team;

F8. Automated generation of control charts: This function allows the creation of control charts with project informations, necessary to control requirements, through a simpler and easier way to understand. In this sense, three possible charts can be generated: “requirement X amount of dependencies”, “quantity of requirements excluded X stakeholder requestor” and “quantity of requirements excluded X team member responsible”.

The "requirement X amount of dependencies" chart allows viewing the project requirements according to the number of dependencies associated with them. The purpose of this chart is to enable the identification of requirements that are isolated and also of those that have many dependencies, which may imply greater risk and impact on the project if they are excluded or altered. The “quantity of requirements excluded X stakeholder requestor” chart allows investigate into why the requirements for a given stakeholder are being excluded. This chart leads to the creation of hypothesis that need to be verified because poorly identified requirements can lead to project failure. Finally, the “quantity of requirements excluded X team member responsible" chart allows investigate why the requirements under the responsibility of a particular team member are excluded.

From the functions identified, the requirements engineering module was implemented, which was later integrated with SAPM, contributing to the refinement of the scope management.

3.2 Implementation

In this stage, the requirements engineering module was developed, in order to find the functional requirements previously identified in order to promote the effective application of the "Collect Requirements" process of scope management. In terms of development, we have used open computational resources, as PHP, HTML and JavaScript languages, MySQL server to the database system management and Apache web server. This module was integrated with SAPM, allowing the evaluation by professionals about its suitability.

Every application of requirements engineering in scope management starts from the principle of correctly recording and analyzing the requirements based on their features, and validating the requirements in order to confirm their relevance. This way, the proposed implementation follows an execution flow that is presented in Figure 1.

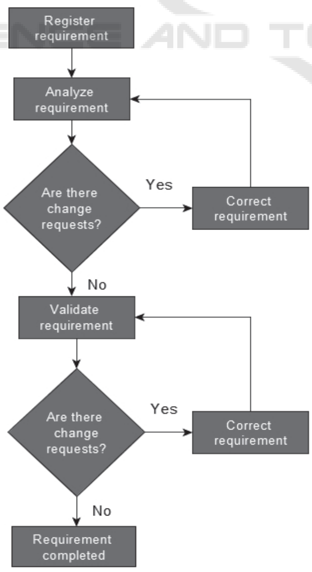

Figure 1: Main flow of requirements engineering.

The first action of the main flow is to "Register requirement", in which all specifications of the requirement are documented, as well as the dependent requirements, stakeholder requestor and member of the team responsible for managing the requirement. Subsequently, we have the "Analyze requirement", in order to verify if there are inconsistencies or failures in its specification, according to specific criteria. When the "Analyze requirement" action encounters some inconsistency, the requirement is directed to the "Correct requirement" action and, after correction, the requirement becomes available for the analysis, so the process is repeated until no inconsistency is found.

If there is not any inconsistency, the requirement is available for the "Validate requirement" action, which is performed in order to confirm the quality of the requirement specification, according to complementary criteria to those that were followed in the analysis action and also directing the requirement for correction when some inconsistency is detected. If the requirement is correct, as expected by the stakeholders, the flow is completed by making available the requirement for consultation of the entire development team.

The "Analyze requirement" and "Validate requirement" actions are derived, respectively, from the "F3. Automation of analysis and negotiation" and "F4. Automation of requirements validation", presented in section 4.1. In both functions, checklists are applied to help in the analysis and validation of the requirements in a coherent way, through previously established criteria. The checklist questions of the "F3. Automation of analysis and negotiation" and their respective descriptions are:

- "Are there requirements combined?": must be checked whether there are internal requirements, ie whether the requirement can be subdivided into other;
- "Is the requirement necessary?": must be checked whether is really essential for the project scope;
- "Is the requirement in line with business objectives?": must be checked whether the requirement is consistent with the objectives of the organization requesting the project;
- "Does the requirement have ambiguity?": must be checked whether there is a double interpretation associated with the requirement;
- "Is the requirement realistic?": must be checked whether the requirement is possible to be implemented, given the relevant project constraints.

In turn, the checklist questions of the "F4. Automation of requirements validation" and their respective descriptions are:

- “Is the requirement complete?”: must be checked if there is lack of information in the description of the requirement;
- “Is the requirement understandable?”: must be checked whether the information describing the requirement allows it to be understood;
- “Is the requirement redundant?”: must be checked if there is repeated information, ie unnecessary for the definition of the requirement;
- “Is the requirement consistent?”: must be checked whether there are contradictions in the description of the requirement and also if it does not oppose another requirement;
- “Is the requirement measurable?”: must be checked whether the requirement can be tested or measured in any way.

As mentioned previously, the use of the two checklists provides a standard way to analyze and validate the requirements, based on common criteria. Thus, every completed requirement presents a standard of understanding common to stakeholders and, therefore, it can be developed.

Figure 2 shows the registration screen of a requirement, in order to stand out the information requested from users so that the requirement is documented correctly. It is worth mentioning that the information registered is used in the automated generation of traceability matrices and control charts.

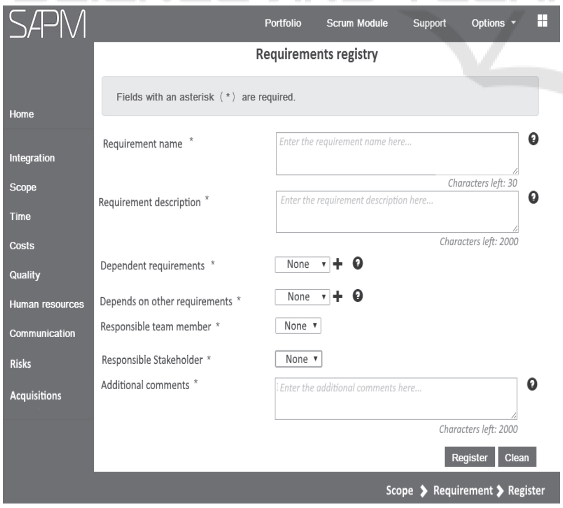

Figure 2: Requirements registry screen.

The traceability matrices, demonstrating the dependencies between the requirements, are a way of understanding the possible impacts caused by changes in requirements. This way, it can be seen in Figure 3 the traceability matrix "requirement X requirement" implemented as SAPM expansion.

As can be noticed in Figure 3 the requirements are arranged in the horizontal and vertical axes of the traceability matrix and, when there is a relationship between the requirements, a "X" is marked at the intersection of the axes. Thus, the more markings a requirement has, the more relationships it has with other requirements and, consequently, the greater the impact of any change.

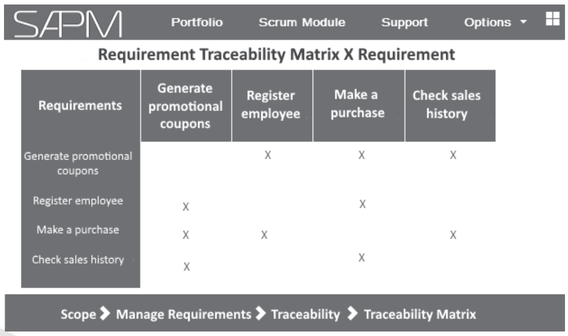

Figure 3: Traceability matrix “requirement X requirement”.

In addition to traceability matrices, control charts are also generated automatically from the requirements registry. Thus, the "requirement X amount of dependencies” information is shown in Figure 4.

Observing Figure 4, the horizontal axis has all the project requirements, while the vertical axis contains the number of requirements dependencies. From this chart, it is possible to verify in a clear and intuitive way the number of dependencies of the requirements.

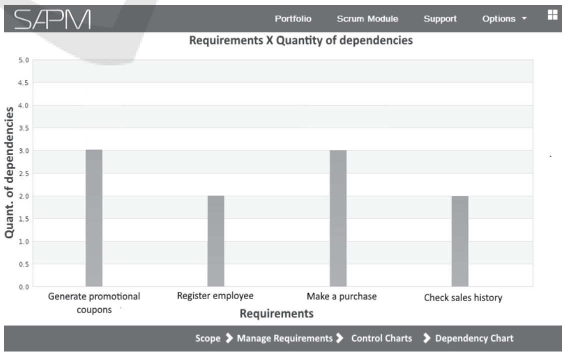

Figure 4: Control chart “requirement X amount of dependencies”.

Thus, the functions were implemented in order to allow the application of requirements engineering as an aid to the "Collect Requirements" process of the scope management, given its relevance to the execution of this knowledge area and, consequently, to the management of a project. It is worth to point that the automated creation of traceability matrices and control charts represents a simple and quick way for the development team to visualize project information, supporting the decision-making.

3.3 Evaluation Process

In this stage, the evaluation of the benefits that the integration of requirements engineering provided to the project scope management was carried out, as well as the identification of improvement points. This process relied on the collaboration of 16 professionals who apply project management in their daily activities and, consequently, the scope management.

All participants in the evaluation process were trained on the functions of SAPM, mainly on the functions related to the scope management and the new functions of the requirements engineering module, with the purpose of contributing to the ease of use and, consequently, to the evaluators maintain the focus in the evaluation of the desired functionalities. After the training, participants had free access to SAPM for a period of 14 days, in which they could evaluate the integration of the requirements engineering with the scope management.

After the evaluation period, the participants were invited to answer an evaluation form, where they express their opinion about the real benefits provided by the application of requirements engineering to “Collect Requirements” process, provided in scope management. In the form responses the participants assigned scores from 0 to 10 points to each of the analyzed criteria, besides to comment on the strengths and weaknesses observed.

The evaluation process allowed the confirmation of the relevance of integration of requirements engineering to scope management, in order to assist the “Collect Requirements” process.

In the histogram of Figure 5, it can be seen the results obtained by the functions analyzed in the evaluation. It is important to notice that the function "Project feasibility study" was not considered in the evaluation process, since this function presents to the user pertinent information for the feasibility study, however, no activity is properly automated.

From the histogram, it is possible to verify that the grades vary from 7 to 10 points. In this way, it is possible to conclude that the proposed integration brings a real benefit to the effectiveness of the scope management in projects, supplying the current needs of the area.

The strengths of requirements history (F5) and of exclusions (F6), as well as the generation of control chart (F8), were stood out. In addition, on the open questions for feedback from participants in the evaluation process, it was stood out the importance of the generation of traceability matrices (F7). The criticisms collected focused on aspects of the tool interface that were adopted because the requirements engineering module was linked to the previously defined layout for SAPM.

4 CONCLUSIONS

Scope management is a knowledge area of great relevance to project management as it defines all the work required for the project to be successfully completed. Despite this, it was verified by the empirical study that the main computational tools of available project management do not approach the scope management totality, failing the application of some processes, as the “Collect Requirements” process. Thus process is fundamental to the application of the scope management, since poorly collected requirements can lead to changes in deadlines, costs, and even cancellation of the project. Thus, it is important to adopt techniques of the Requirements Engineering that can help in the accomplishment of this process, contributing to the scope management.

To prove this, a requirements engineering module, contemplating specific techniques of RE focused on "Collect Requirements" process, was developed and integrated into the scope management of SAPM, contributing to the refinement of this computational tool and enabling the evaluation of the module by professionals. The implemented functions allow the project scope requirements to be correctly specified and documented, using a flow of actions that reduce problems as inconsistency, ambiguity, redundancy and, thus reduce the probability that these errors will be propagated to other parts of the projects and cause losses. Another relevant contribution is the automated generation of traceability matrices and control charts.

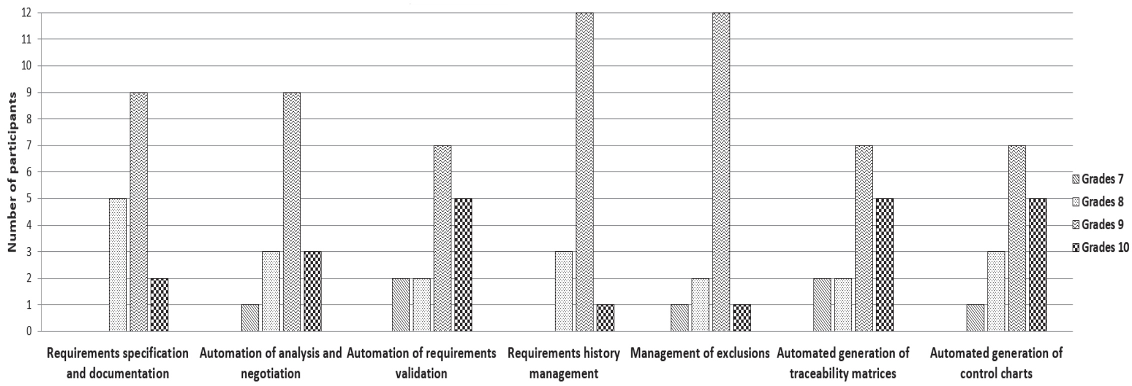

Figure 5: Assessment of requirements engineering functions.

As a perspective to future work, it is possible to point improvements in the developed module, with the intention of making it even more comprehensive. Among the possible functions is the automated generation of a formal document, which allows collecting all the information of the requirements of the project scope and organizing them in a standardized way, establishing templates to be adopted, generating new charts. In the case of specific software projects, the system could offer the option of specifying and documenting requirements through area-specific graphical notations such as Unified Modeling Language Use Cases.

REFERENCES

Alexander, I., Beus-Dukic, L., 2009. *Discovering requirements: how to specify products and services*. John Wiley & Sons. West Sussex, 1st edition.

Badariah, S., Sahibuddin, S., Ghani, A. A. A., 2009. Re-defining requirements engineering process improvement model. In *16th Asia-Pacifc Software Engineering Conference*. Penang, vol.16, pages 87-92.

Ballejos, L. C., Montagna, J. M., 2008. Method for stakeholder identification in interorganizational environments. *Requirements engineering*, vol.13, n. 4, pages 281-197.

Bjamason, E., Wnuk, K., Regnell, B., 2012. Are you bitting off more than you can chew? A case study on causes ans effects of overscoping in large-scale software engineering. *Information and Software Technology*, vol. 54, issue 10, pages 1107-1124.

Contessoto, A. G., Sant'Ana, L. A., Souza, R. C. G., Valêncio, C. R., Zafalon, G. F. D., Amorim, A. R., Esteca, A. M. N., 2016. Improving risk identification process in project management. *In Proceedings of the 28th International Conference on Software Engineering and Knowledge Engineering (SEKE),* San Francisco Bay, USA, pages 555-558.

Damian, D., 2007. Stakeholders in global requirements engineering: Lessons learned from practice. *IEEE software*, vol. 24, n. 2, pages 21-27.

Di Thommazo, A., Camargo, K., Hernandes, E., Goncalves, G., Pedro, J., Belgamo, A., Fabbri, A., 2015. Using the dependence level among requirements to priorize the regression testing set and characterize the complexity of requirements change. *In Proceedings of the 17th International Conference on Enterprise Information Systems (ICEIS)*, vol. 2, pages 231-241.

Hall, T. S., Beecham, S., Rainer, A., 2003. Requirements problems in twelve software companies: an empirical analysis. *Empirical software engineering*, vol. 8, n. 1, pages 7-42.

Ito, M. L., Fuzii, R. Y. M., Souza, R. C. G., Valêncio, C. R., Tronco, M. L., 2011. Support tool the validation processo f functional requirements. *Latin America Transactions IEEE*, vol. 9, n. 5, pages 889-894.

Kotonya, G., Sommerville, I., 1998. *Requirements Engineering: Process and techniques*. John Wiley and Sons.

Kumari, N., Pillai, A. S., 2014. A study on project scope as a requirements elicitation issue. *Computing for Sustainable Global Development (INDIACom) International Conference on IEEE*, pages 510-514.

Microsoft Project, https://products.office.com/en-us/project/project-and-portfolio-management-software.

NetProject, http://www.netproject.com.br/.

Project Management Institute – PMI, 2013. *A Guide to the Project Management Body of Knowledge*. 5th edition.

Pandey, D., Suman, U., Ramani, A. K., 2010. An effective requirement engineering process model for software development and requirements management. *In International conference on advances in recent technologies in communication and computing,* Kottayam, pages 287-291.

Project Planner, 2016 http://www.smartworks.us.

ProjectLibre, http://www.projectlibre.org/.

Simão, I., Varela, R. A., 2009. Requirements Engineering as an innovative process in organizations. *IET working papers series no. WPS08/2009*, Lisboa, Portugal.

Smith, J., 2002. The 40 root causes of troubled IT projects. *Computing & Control Engineering Journal*, vol. 13, pages 109-112.

Sommerville, I., 2011. *Software Engineering*, Pearson, Boston, 9th edition.

Souza, R. C. G., Esteca, A. M. N., Santos, A. B., Valêncio, C. R., and Honda, M. T., 2011. Web System to Aid Project Management. In *Proceedings of the 23th International Conference on Software Engineering and Knowledge Engineering (SEKE)*, Miami, USA, pages 325-330.