Argumentation Schemes for the Collaborative Debate of Requirement Risks in Software Projects

Denise da Luz Siqueira[1], Lisandra M. Fontoura[1], Rafael H. Bordini[2], Luis A. L. Silva[1]
[1]Programa de Pós-Graduação em Ciência da Computação, UFSM. Santa Maria, RS, Brazil
{denise.siqueira, lisandramf, silva.luisalvaro}@gmail.com
[2]Programa de Pós-Graduação em Ciência da Computação, PUCRS. Porto Alegre, RS, Brazil
rafael.bordini@pucrs.br

Abstract – **Managing risks in real-world software projects is of paramount importance. A significant class of such risks is related to the engineering of requirements, commonly involving the presentation and analysis of risk management arguments from both software engineers and clients involved in collaborative debates. In this work, drawing inspiration from argumentation theory in Artificial Intelligence, we introduce a number of "argumentation schemes" and associated "critical questions" to support such discussions. In doing so, we propose schemes related to risks due to excessive numbers of requirements; inadequate client representatives and poor understanding of client needs; incorrect, incomplete and conflicting requirements, and complex and non-traceable requirements. We also present a case study where the developed schemes were used to support the discussion of requirement risks in the context of a research and prototyping software project for the Brazilian Army.**

Keywords-component: Argumentation Schemes; Requirement Risks; Risk Management; Argumentation.

I. INTRODUCTION

Understanding and documenting key clients' needs is the foundation for success of software projects. As part of *requirement engineering* (RE) processes, these needs can be exploited through the development of RE tasks [1]. Despite these tasks, IT projects have changed in recent years because of increasingly complex requirements as described in [2]. Ubiquitous computing, inter-organizational systems, and an increment on consumer-targeted systems have created new challenges, making it difficult to manage requirements. This can be shown when major failures in software projects are analyzed [3]. It indicates that project stakeholders really ought to collaborate to the systematic identification of these problems in the initial phases of a software project, which is the time when the requirements are captured and analyzed as defined in software standards such as CMMI [4]. As explored in our research project [5-6], early consideration and treatment of software project risks amounts to the development of collaborative *risk management* (RM) tasks.

In collaborative scenarios, RM debates where "argumentation" techniques from Artificial Intelligence (AI) [7] can be used by project stakeholders to identify and analyze risks and their causes along with the proposition of risk response plans. As described in [8], argumentation processes can be used as a form of acquisition and construction of knowledge, which is a task based on presentation and computation of arguments and counter-arguments as well as the deliberations on particular issues. As described in [9], "argumentation schemes" allow the construction of such a dialogue-based system, where these schemes allow software engineers to describe argument structures commonly used in daily discourse. Such a scheme model presents a template for an argument type in an application problem, where this template can be used by stakeholders to elaborate and analyze argument instances to be advanced in their debates. As exploited in different applications [10-13], argumentation schemes can also be proposed so that they are used in the collaborative management of requirement risks in software projects.

As presented in [3], an open-ended questionnaire answered by IT professionals assessed specific factors contributing to either the success or the failure of software projects. Among the success factors identified, 2 of them refer to the management of requirements: the (high) level of abstraction for client participation and the thorough definition of requirements from the beginning. Among the failures identified, the study highlights the lack of requirement definition. From such results, this work proposes a new set of argumentation schemes from requirement risks. As the high-level of client participation impacts the success of a project, we specified schemes to capture arguments about requirement risks due to: the utilization of inadequate client representatives and requirements that do not fulfil clients' needs. The scheme for risks of inadequate client representatives assesses the experience of clients in the business domain of a project, as well as the interest and involvement of these clients in the elicitation of requirements. Moreover, the lack of involvement of clients can end up causing problems in the identification of their true needs. In effect, the argumentation scheme for requirements that not fulfil clients' needs is helpful in the assessment of the clients' commitment towards the elicitation and validation of requirements. A thorough initial definition of requirements can only be achieved when risks due to incorrect, incomplete and conflicting requirements are avoided. In this paper we introduce argumentation schemes to deal with such problems. In particular, the scheme for risks of incorrect requirements refers to whether the project scope is defined properly or not. The scheme for incomplete requirements aims to assess if all the requirement elicitation work has been done adequately so that the project could have better chances of achieving its goals [14]. In this context, issues regarding the specification of an excessive number of requirements can also be approached via

10.18293/SEKE2017-065

one of the schemes we put forward here. Conflicting requirements can occur when stakeholders have different views about project requirements or conflicting sources of information on the requirements. In general, the lack of requirement definition occurs when risks referring to the complexity of requirements are not prioritized. Among other reasons, complex requirements make it difficult for the project team to understand and share these requirements, reflecting how requirements are conceptualized and structured. According to [2], this kind of modelling issue indicates that it is difficult to understand, specify, and communicate the requirements. Finally, risks regarding the traceability of requirements ought to be examined when requirements are subject to many changes in the project, where this argumentation topic is considered in the specification of another scheme. To assess the usefulness of the schemes we created, we developed a collaborative debate on the management of requirement risks for a software project being carried out at UFSM. This project involves the research and prototyping of a virtual tactical simulator for the Brazilian Army – the SIS-ASTROS project. In this paper, a small fragment of this debate is used to illustrate how our requirement risk schemes were used by members of this project.

The remainder of this paper is structure as follows. Section II reviews background information about RM, RE and argumentation. Section III presents the new schemes developed. Section IV shows a fragment of a collaborative debate in which such schemes were used. Section V shortly discusses related works. Section VI summarizes the contribution of this paper and discusses future works.

II. Requirement Risks and Argumentation Schemes

The engineering of requirements in software projects is a process involving tasks for requirement elicitation, analysis, specification, validation and management [1]. In the requirement elicitation, clients' needs regarding their expected software products are described in a user language. In the requirement analysis, information captured during elicitation is gradually refined. This refinement leads to the construction of a model for the project requirements, where aspects of the function and behaviour of the software are identified. In the requirement specification, a conceptual model of the software is developed, which is used in the validation of the specified requirements and the planning of software development processes. In the requirement validation, the quality of the specifications of the requirement documents is examined by stakeholders. These specifications are reviewed to guarantee that the requirements were not captured as ambiguous descriptions, in addition to the detection of inconsistencies, omissions and errors. The team conducting such revision tasks should involve stakeholders interested in the examination of the specifications in the search for content and interpretation errors.

Despite all such efforts in RE, problems in those requirement specifications can be approached as risks in software projects. In simple terms, a risk is an event that can affect the goals of a project [15]. So, RM in software projects can be taken as the application of a set of principles and practices to the identification, analysis and treatment of risk factors aiming to prevent a project from failing, despite the fact that a risk can alternatively be treated as an opportunity in many software projects. This indicates that "risks that are due to software requirements" ought to be identified and their occurrence probability and project impact analyzed. Based on such risk evaluation tasks, risk treatment plans are established to risks with high priority in a project. In general, these RM processes demand a more intense communication process inside and outside the limits of the project. The need for collaboration is even higher when software projects involve stakeholders that are distributed geographically, in a scenario where the web is the primary communication medium. As described in [4], RM frameworks ought to be founded on significant steps of collaborative communication among stakeholders. Computational support for these activities is approached by web-based tools for recording and querying risk statements, their analysis and response plans. In effect, the collection of such RM information is naturally done in a dialogue where multiple stakeholders are involved, where collaborative steps support the discussion of different points of view and experiences. So, processes of collaborative decision making are what tend to make the RM most effective [4].

In argumentation scenarios involving project stakeholders, RM debates can ground the identification and analysis of risks and their response plans. As modelled by argumentation schemes, a set of speech acts is defined in the form of rules, which are used to give support to a conclusion statement advanced in such debates. In this case, the premises of an argumentation scheme allow users to capture common assumptions made in arguments. If such assumptions are accepted as true, this guarantees that the conclusion defined in the body of a scheme will also be accepted. In addition to a set of structural rules represented as premises and conclusion, each scheme is fundamentally defined by a set of *critical questions* (CQs). In case a user does not answer any one of these questions with relevant information in a debate situation, arguments that are based on the reasoning template of a scheme can be rejected. In general, the indication of variables helps the formulation of more generalized specification of argumentation templates, showing places where debate participants can shape scheme-based arguments with concrete pieces of information specific to the issue being discussed.

Argumentation schemes can be investigated to promote the identification of rather generalized patterns of argumentation. Described in an informal logic notation, the catalogue of such schemes [9] contains generalized templates for arguments from cause to effect, arguments from expert opinion, arguments from sign, etc. In fact, such kinds of templates could be exploited in the construction of tools to support collaborative debates. To approach such catalogues, it is relevant to notice that the analysis of risk causes and causal dependencies related to software failures is a significant RM characteristic (e.g. [16]). As defined in the "argumentation scheme from cause to effect" [9], argument templates can express relationships between causal factors and certain situations, where such situations are taken as effects from these causes. The generalized description of this scheme is: *Major premise*: Generally, if A occurs, then B will (might) occur. *Minor premise*: In this case, A occurred or will (might) occur. *Conclusion*: Therefore, in this case, B will (might) occur. *Critical questions*: How strong is the causal relation between (X) and (Y) (if this causal generalization is true at all)? Is the evidence (X) mentioned (if there is any) strong enough to warrant the cause-effect generalization as stated? Are there other factors (F) that would or will interfere with the production of the effect (E) in this case? Is (X) the main (or single) cause for the occurrence of (Y)? Questions like these can be used to attack scheme-based instances of arguments when the debate participants present counter-arguments. In many situations, instances of questions created according to a scheme can present doubts about the structural connection between the premises and the conclusion of the arguments. From these questions, a scheme-based argument can be attacked in various ways, where the proposition of a counter-argument contains a conclusion opposing the conclusion of the original argument. Importantly, the assessment of such questions can also promote the presentation of supporting information to back up such arguments (Fig. 1 shows examples of these scheme-based speech moves).

Argumentation schemes are exploited in the solution of different application problems. In [10], an argumentation model proposed the use of schemes in the formalization of a dialogue protocol for agent communication. The protocol focuses on the presentation and computation of scheme-based arguments in favor of and against decisions, aiming to help users in the deliberation about the viability of organ transplants. In [11], schemes are used in the conciliation of results from similar biological experiments. Due to complexity of these experiments, results obtained when they are conducted may be conflicting or contradictory. In [12], a scheme-based approach to the representation and analysis of arguments regarding the security of computational systems is discussed. Those schemes were constructed from the generalization of safety-based issues commonly found in this domain. In [13], schemes are used in e-government and e-participation scenarios, where a web-based platform to promote democratic participation and debate of legal projects is proposed. Through a web-based tool, legal proposals can be input to the system, and citizens participate in the evaluation of those proposals through a website, either supporting or not the proposals. Schemes are used in the structuring of such proposals so as to promote their evaluation by members of the public. Despite the existence of these applications of schemes, and even though a preliminary set of schemes for RM [6] (as part of the research project where this work was conducted), the available schemes in the literature do not cover argument templates directed to the analysis of risks related to requirements in software projects.

III. Argumentation Schemes for the Collaborative Management of Requirement Risks

Following catalogues of requirement risks [17-20], argumentation schemes addressing the analysis of requirement risks can be proposed. To formulate these templates, requirement risks were analyzed according to the key RE tasks [1]. Taking advantage of a knowledge engineering process for the specification of argumentation schemes as proposed in [21], a set of 8 new schemes was structured via interpretations, premises, conclusion and CQs.

A. Argumentation scheme for risks related to excessive number of requirements

Risk interpretations: large numbers of requirement information sources (sometime different sources) available for query; large number of unnecessary requirements; too large and ambitious project scope; lack of consensus among project stakeholders;

Major premise: If there is a large number of information sources (S) available for consultation, there will (might) have an excessive number of requirements (R)

Minor premise: In the project (P) there is a large number of information sources (S) available for consultation

Conclusion: There is an excessive number of requirements (R)

Critical questions: Is the large number of available information sources (S) being questioned properly by project stakeholders (K)? Is there a too large and ambitious scope in the software project (P)? Is there a lack of consensus among project stakeholders (K) about the large number of requirements (R)?

For this scheme, an additional set of CQs that is typical in argumentation schemes from cause to effect are: How strong is the causal relation between the large number of information sources (S) and the excessive number of requirements (R)? Is the large number of information sources (S) the main (or single) cause for the occurrence of an excessive number of requirements (R)? Is there evidence (X) that there is a large number of information sources (S) available for query? Is there an excessive number of different information sources (S) available? As these cause-and-effect types of questions were instantiated in the context of the argumentation scheme from risks of excessive number of requirements, they can also be instantiated in the context of the other schemes presented later in this paper. Looking at RE tasks, it is also possible to ask if RE techniques are in place, if there is a proper exploitation of these techniques, if the people involved in the development of RE tasks have the skills to adequately execute these tasks in the project. Having a general applicability in all schemes proposed here, the overall templates for these RE kinds of CQs are: Are there requirement (elicitation, analysis, specification, validation, and management) techniques (T) to support (X) so that risk (R) does not apply to project (P)? Are there requirement (elicitation, analysis, specification, validation, management) techniques (T) being exploited properly by project stakeholders (K) so that risk (R) is not in the project (P)? Are the knowledge and experience of requirement engineers (E) adequate to do (T) so that risk (R) does not apply to project (P)? In order words, it amounts to question whether requirement engineers (E) have the right set of skills to develop the RE task (T). For instance, these RE questions in the context of the argumentation schemes from risks of excessive number of requirements can be written as: Are there requirement elicitation tasks (T) to support the exploitation of a large number of requirement information sources (S) available for query so that an excessive number of requirements (R) does not occur in project (P)? Are there requirement elicitation techniques (T) being exploited properly by project stakeholders (K) so that an excessive number of requirements (R) does not occur in project (P)? Are the knowledge and experience of requirement engineers (A) adequate to exploit a large number of requirement information sources (S) available for consultation so that an excessive number of requirements (R) does not occur in project (P)?

The set of CQs defined here is not exhaustive. If users find it relevant to include new questions in this list, they should do so. A possible example for this is the formulation of questions related to the kinds of projects that are most commonly executed in a software development organization. In the SIS-ASTROS project, CQs related to distributed software development issues and related requirement risks can be formulated. This is a context where the company acts on the design and implementation of simulation systems, which have particular risks associated.

B. Argumentation scheme for risks related to inadequacy of client representatives

Risk interpretations: client representatives not having experience in the target application domain; client representatives having opposing interests towards the project; client representatives are not truly committed to support the RE tasks;

Major premise: If client representatives (C) are not able to offer support to the RE tasks (T), there will (might) be client representatives (C) that are not adequate

Minor premise: In the project, the client representatives (C) are not able to offer support to the RE tasks (T)

Conclusion: The client representatives (C) are not adequate

Critical questions: Are client representatives (C) appearing to have just a vague knowledge about the target application domain (D)? Is there any reason for client representatives (C) to omit relevant requirement information (R)? Are client representatives (C) not really available to support the RE tasks (T)?

C. Argumentation scheme for risks related to incorrect requirements

Risk interpretations: misunderstanding the client needs; project scope not well defined; incorrect communication of project information by the client; lack of requirement validation tasks;

Major premise: If the client needs (N) are not being analyzed properly by requirement engineers (E), there will (might) be incorrect requirements (R)

Minor premise: In the project, the client needs (N) are not being analyzed properly by requirement engineers (E)

Conclusion: There are incorrect requirements (R)

Critical questions: Are the business rules (B) of the target application domain (D) too complex? Is there evidence (X) that the client needs (N) were not understood properly by requirement engineers (E), or that project information (I) was not properly communicated by clients? Are clients (C) really sure about what the significant requirements (R) are?

D. Argumentation scheme for risks related to complex requirements

Risk interpretations: large amount of information regarding project inputs and outputs; complex interaction with too many external interfaces (e.g. other systems, web services, etc); target application domain contains terms and concepts that are not well defined;

Major premise: If the target application domain (D) is complex for project stakeholders (K), there will (might) be complex requirements (R)

Minor premise: In the project, the target application domain (D) is complex for project stakeholders (K)

Conclusion: There are complex requirements (R)

Critical questions: Is there really a large amount of information related to inputs (I) and outputs (O) and interfaces (S) in project (P)? Is the complexity related to requirements (R) that are not being understood correctly? Are software artefacts (T) that help reduce the complexity of the requirement representations (R) being used?

E. Argumentation scheme for risks related to incomplete requirements

Risk interpretations: clients do not know what they want; clients forgetting to state important requirements; missing significant requirements as far as the project scope is concerned;

Major premise: If the requirements (R) have a large number of missing requirement specifications (S), there will (might) be incomplete requirements (R)

Minor premise: In project (P), requirements (R) have a large number of missing requirement specifications (S)

Conclusion: There are incomplete requirements (R)

Critical questions: Are clients (C) really sure about their needs in project (P)? Are there missing significant requirements (R) as far as the project scope (P) is concerned? Is there evidence (X) that clients are forgetting to state important requirements (R)?

F. Argumentation scheme for risks related to requirements that do not fulfil client needs

Risk interpretations: specified functionalities do not meet the clients' needs; requirement engineers not understanding the clients' needs; non-compliance issues between the specified requirements and the clients' needs for the project;

Major premise: If there are non-compliance issues (I) between the specified requirements (R) and the clients' needs (N), there will (might) be requirements (R) that do not fulfil clients needs (N)

Minor premise: In project (P), there are non-compliance issues (I) between the specified requirements (R) and the clients' needs (N)

Conclusion: There are requirements (R) that do not fulfil clients' needs (N)

Critical questions: Are clients (C) making their needs (N) clear (not vague or poorly visible) to requirement engineers (E)? Are client representatives (C) adequately supporting the development of the RE tasks (T)? Is there evidence (X) that client representatives (C) are committed to the development of the RE tasks (T)?

G. Argumentation scheme for risks related to conflicting requirements

Risk interpretations: conflicting information among specified requirements; different stakeholders proposing conflicting requirements; lack of a proper definition of scope for the project;

Major premise: If there are inconsistencies (I) among specified requirements (R), there will (might) be conflicting requirements (R)

Minor premise: In project (P), there are inconsistencies (I) among specified requirements (R)

Conclusion: There are conflicting requirements (R)

Critical questions: Are the conflicting requirements (R) related to divergences of opinion from different stakeholders (K)? Is there evidence (X) that there are conflicting requirements from different sources of information (S)? Are the conflicting requirements (R) due to incorrect understanding of clients needs (N)?

H. Argumentation scheme for risks related to non-traceable requirements

Risk interpretations: requirement sources are not tracked in the project; requirement changes are not managed in the project; lack of requirement management tasks;

Major premise: If the requirements (R) are not linked to their sources (S), there will (might) be non-traceable requirements (R)

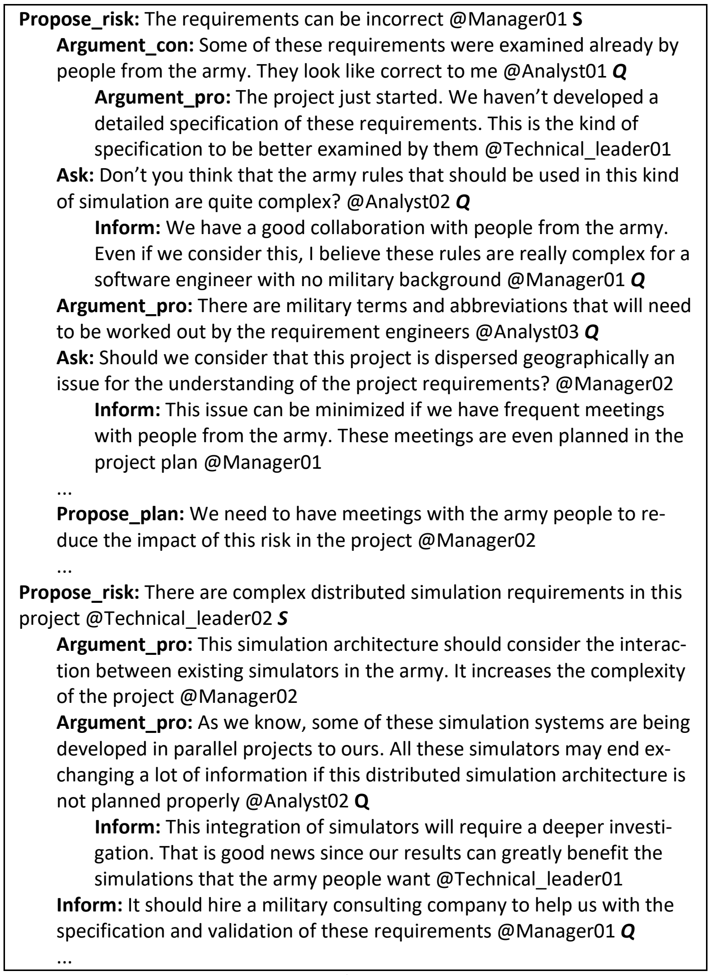
Minor premise: In the project (P), the requirements (R) are not linked to their sources (S)

Conclusion: There are non-traceable requirements (R)

Critical questions: Are there requirement management techniques (T) being used by stakeholders (K) in the definition, capture and recording of links between requirement specifications (R) and their sources (S)? Are there requirement management techniques (T) being used by stakeholders (K) in the definition, capture and recording of links between requirement specifications (R) and requests for changes? Are the requirement changes (R) not being managed by stakeholders (K)?

IV. A Case for Argumentation Schemes in the Collaborative Management of Requirement Risks

A fragment of a collaborative debate of requirement risks in the SIS-ASTROS project (Fig. 1) can be presented to expose alternative forms that our schemes were used by participants of this project. First of all, this debate shows that a risk proposal can be grounded on a selected argumentation scheme (S next to a Propose_risk act of speech). The debate also shows that participants can use CQs when they want to refute a previous risk proposal argument. This kind of argument against the point being made can be advanced with the help of a selected CQ (Q next to an Argument_con). There, questions can also be submitted by participants, where a question is linked to a CQ of a selected scheme in the debate (Q next to an Ask). Alternatively, project information can be advanced as an answer to a CQ (Q linked to an Inform).



- **Propose_risk:** The requirements can be incorrect @Manager01 **S**
  - **Argument_con:** Some of these requirements were examined already by people from the army. They look like correct to me @Analyst01 ***Q***
    - **Argument_pro:** The project just started. We haven't developed a detailed specification of these requirements. This is the kind of specification to be better examined by them @Technical_leader01
  - **Ask:** Don't you think that the army rules that should be used in this kind of simulation are quite complex? @Analyst02 ***Q***
    - **Inform:** We have a good collaboration with people from the army. Even if we consider this, I believe these rules are really complex for a software engineer with no military background @Manager01 ***Q***
  - **Argument_pro:** There are military terms and abbreviations that will need to be worked out by the requirement engineers @Analyst03 ***Q***
  - **Ask:** Should we consider that this project is dispersed geographically an issue for the understanding of the project requirements? @Manager02
    - **Inform:** This issue can be minimized if we have frequent meetings with people from the army. These meetings are even planned in the project plan @Manager01
  - ...
  - **Propose_plan:** We need to have meetings with the army people to reduce the impact of this risk in the project @Manager02
  - ...
- **Propose_risk:** There are complex distributed simulation requirements in this project @Technical_leader02 ***S***
  - **Argument_pro:** This simulation architecture should consider the interaction between existing simulators in the army. It increases the complexity of the project @Manager02
  - **Argument_pro:** As we know, some of these simulation systems are being developed in parallel projects to ours. All these simulators may end exchanging a lot of information if this distributed simulation architecture is not planned properly @Analyst02 **Q**
    - **Inform:** This integration of simulators will require a deeper investigation. That is good news since our results can greatly benefit the simulations that the army people want @Technical_leader01
  - **Inform:** It should hire a military consulting company to help us with the specification and validation of these requirements @Manager01 ***Q***
  - ...

Figure 1. An example of a requirement RM debate

Answers to a CQ can also be presented via an argument pro (Q next to Argument_pro), making either stronger or weaker the risk proposal analyzed by pros and cons. There, the overall debate that is promoted via argumentation schemes can help the gathering of project information from stakeholders. That is relevant in the determination of the probability and impact of a risk proposed, and their consequent prioritization. In the end, the answers presented to CQs used by debate participants can also contain pieces of explanation for the determination of different risk response plans to be used in the project. In the web-based system we have developed [5-6] that organizes these collaborative debates, the labels Q and S along arguments presented can be used by users. As a result, the debate system shows the schemes used in the construction of those arguments allowing users to reuse these templates in the construction of new arguments.

V. Discussion

Communication between project stakeholders is fundamental in the engineering of requirements [4]. Argumentation [7-8] focuses on the study and development of intelligent systems that model several kinds of communication aspects, such as how to mediate argumentation-based debates, for instance. Argumentation techniques are mostly directed to problems in the RE context such as the resolution of inconsistencies in the consolidation of multiple requirement specifications [22] and the identification of security requirements in making mitigation decisions [23], for instance. In our work argumentation is used as a framework for the assessment of project risks related to requirement problems. In RM, these approaches for argumentation are being exploited as essential components of risk response plans, so as to assess the requirement specifications of a project in order to reduce the probability and impact of requirement risks. As possible forms of extending the approaches mentioned above, the argumentation schemes presented here focus on the capture of stakeholders' arguments in collaborative discussions of requirement risks. Through these schemes, the overall idea is to engage these stakeholders in the critical argumentation-based identification and analysis of these issues and consequent planning of risk responses.

Similar to [10-12], a domain-specific specification of argumentation schemes is exploited in the schemes proposed here. In [10], although the dialectical process modeled as the analysis of scheme-based pros and cons has a connection with how humans make qualitative decisions, the deliberation of solutions to RM problems is typically grounded on the additional utilization of more informal argumentation acts, where the dialogue used by project stakeholders is not limited to pros and cons kinds of scheme-based arguments. In [11], schemes are designed to support users in the analysis of the nature of inconsistencies in experimental results in Biology. In contrast with our work, a kind of cause-and-effect examination is not explicitly presented there as it is in our set of schemes. The schemes used in [13] rely on the generalized argumentation templates presented in [9]. Our work also exploits the reuse of such generalized scheme formulations, adapting a selected scheme from such catalogue to the specification of argumentation templates for the analysis of requirement risks.

VI. Concluding Remarks

One of the aims of this research is to reduce the communication gaps among project stakeholders particularly in relation to the collaborative development of RE tasks. In this context, the contributions of this paper are twofold. First, inspired by work on argumentation theory in the field of AI, we developed a number of new argumentation schemes to support collaborative discussions of requirement risks. Associated with these schemes (i.e., reasoning patterns) there is a large number of CQs which can be very useful to both software engineers and clients engage in such discussions. Second, we showed a case study conducted in the context of a real software project; the debate excerpt demonstrates how our schemes and particularly the CQs can be useful in practice. Future work includes the development of a larger set of requirement risks schemes, besides further experimentation with the tools we have developed [5-6] to support discussion of requirement risks based on argumentation schemes.

Acknowledgment

We thank the Brazilian Army for the financial support through the SIS-ASTROS Project (813782/2014), developed in the context of the PEE-ASTROS 2020.

References

[1] K. E. Wiegers, and J. Beatty, *Software Requirements*, 3ª ed.: Microsoft Press, 2013.

[2] L. Mathiassen, and T. Tuunanen, "Managing Requirements Risks in IT Projects," *IT Professional,* vol. 13, pp. 40-47, 2011.

[3] M. Warkentin *et al.*, "Analysis of Systems Development Project Risks: An Integrative Framework," *ACM SIGMIS Database: the DATABASE for Advances in Information Systems,* vol. 40, no. 2, pp. 8-27, 2009.

[4] SEI, "CMMI® for Development, Version 1.3." p. 482.

[5] F. Severo, L. M. Fontoura, and L. A. L. Silva, "A Dialogue Game Approach to Collaborative Risk Management " in The 25th Int. Conf. on Software Engineering and Knowledge Engineering, Boston, MA, 2013, pp. 548-551.

[6] R. C. B. Pozzebon *et al.*, "Argumentation Schemes for the Reuse of Argumentation Information in Collaborative Risk Management," in Proc. of the 15th IEEE Int. Conf. on Information Reuse and Integration, Redwood City, CA, 2014, pp. 179-186.

[7] C. Chesñevar, A. Maguitman, and R. P. Loui, "Logical Models of Argument," *ACM Computing Surveys,* vol. 32, no. 4, pp. 337-383, 2000.

[8] B. Moulin, H. Irandoust, and M. Bélanger, "Explanation and Argumentation Capabilities : Towards the Creation of More Persuasive Agents," *Artificial Intelligence Review,* pp. 169-222, 2002.

[9] D. Walton, C. Reed, and F. Macagno, *Argumentation Schemes*: Cambridge University Press, 2008.

[10] P. Tolchinsky *et al.*, "Deliberation Dialogues for Reasoning about Safety Critical Actions," *Autonomous Agents and Multi-Agent Systems,* vol. 25, no. 2, pp. 209-259, 2012.

[11] K. McLeod, G. Ferguson, and A. Burger, "Using Argumentation to Resolve Conflict in Biological Databases," *Proc. of Computational Models of Natural Argument (CMNA),* vol. 9, pp. 15-23, 2009.

[12] T. Yuan, and T. Kelly, "Argument Schemes in Computer System Safety Engineering," *Informal Logic,* vol. 31, no. 2, pp. 89-109, 2011.

[13] D. Cartwright, and K. Atkinson, "Using Computational Argumentation to Support E-participation," *IEEE Intelligent Systems,* vol. 24, no. 5, pp. 42-52, 2009.

[14] PMI, "PMBOK Guide: A guide to the Project Management Body of Knowledge," 2013.

[15] B. W. Boehm, "Software Risk Management: Principles and Practices," *IEEE Software,* vol. 8, pp. 32-41, 1991.

[16] T. W. Kwan, and H. K. N. Leung, "A Risk Management Methodology for Project Risk Dependencies," *IEEE Transactions on Software Engineering,* vol. 37, no. 5, pp. 635 - 648, 2011.

[17] S. Amber, N. Shawoo, and S. Begum, "Determination of Risk During Requirement Engineering Process," *Journal of Emerging Trends in Computing and Information Sciences,* vol. 3, no. 3, pp. 358-364, 2012.

[18] S.-J. Huang, and W.-M. Han, "Exploring the Relationship between Software Project Duration and Risk Exposure: A Cluster Analysis," *Information & Management,* vol. 45, no. 3, pp. 175-182, 2008.

[19] L. Wallace, M. Keil, and A. Rai, "How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model," *Decision Sciences,* vol. 35, pp. 289-321, 2004.

[20] L. Wallace, M. Keil, and A. Rai, "Understanding Software Project Risk: a Cluster Analysis," *Information and Management,* vol. 42, no. 1, pp. 115-125, 2004.

[21] D. L. Siqueira *et al.*, "A Knowledge Engineering Process for the Development of Argumentation Schemes for Risk Management in Software Projects," To Appear at The 29th Int. Conf. on Software Engineering & Knowledge Engineering, Pittsburgh, USA, 2017.

[22] E. Bagheri, and F. Ensan, "Consolidating Multiple Requirement Specifications through Argumentation," in ACM Symposium on Applied Computing (SAC '11), TaiChung, Taiwan, 2011, pp. 659-666.

[23] Y. Yu *et al.*, "Automated Analysis of Security Requirements through Risk-based Argumentation," *Journal of Systems and Software,* vol. 106, pp. 102–116, 2015.