

A Collaborative Platform for Better Managing Technical Documentation: An analysis from a Requirements Engineering Perspective

Pedro Cecilio Lopes, Alberto Rodrigues da Silva
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
Lisbon, Portugal
{pedrocecilio, alberto.silva}@tecnico.ulisboa.pt

Abstract—Requirements Engineering (RE) involves important activities of the system development process that requires a good communication among the stakeholders to achieve a shared vision of the problem to be addressed. Therefore a collaborative tool to manage system requirements specifications (SRSs) is crucial to manage the interactions and expectations of these parties. This paper introduces ITBox, a collaborative platform for better managing projects' technical documentation. Although this platform supports multiple types of technical documentation (such as project plans, requirements or tests specifications) specified according different languages and templates, this paper discusses particularly its features focused on a requirements engineering perspective. In particular, this paper discusses how to manage SRS documents with an emphasis on ITBox most innovative aspects such as on social, reusability and productivity features.

Keywords—Collaborative Platforms; Requirements Engineering; Reusability

I. INTRODUCTION

Requirements Engineering (RE) is about reaching a shared understanding between business and technical stakeholders. This bridge can only be established through an effective communication between them [1]. To build a Requirements Specification or a System Requirements Specification (SRS) document, RE considers a set of tasks, which demand a high level of collaboration between these two types of stakeholders, to achieve a common understanding of the software system to be developed. The adverse consequences of disregarding the importance of early RE activities are well-known and reported in the literature [2,10,11]. To avoid those consequences, interpretation problems shall be reduced and, therefore, an effective communication shall be achieved, so that everyone can communicate by means of a common language. The need for requirements and business analysis tools has become ever-more prevalent in organizations today [9].

Therefore, to mitigate these quality problems, users must adopt the most suitable approach for their projects. ITBox is a collaborative web platform which allows the management of projects, and supports multiple types of technical documentation (such as project plans, requirements or tests specifications) specified according different languages and

templates. ITBox is a collaborative platform for managing technical documentation. It combines common and relevant features of other available tools (e.g., TopTeam Analyst, Modern Requirements, Visure Requirements, iRise), while offering innovative reusability features, like model variability aspects in the field of requirements engineering. Furthermore, it is based on an extensible technology – the Google Drive and its APIs –, making it easy to develop and integrate new features regarding the automatic extraction and analysis of data from technical documents.

The structure of this paper is as follows. Section 2 introduces the background namely the RSLingo approach and the RSL language. Section 3 introduces the ITBox main concepts. Sections 4 and 5 detail the platform's collaborative and reusability features, respectively. Finally, Section 6 concludes the paper and lays down some ideas for applying and extending the current work.

II. BACKGROUND

ITBox is a web-based collaborative platform for managing requirements specification documents and other technical documentation. Although ITBox can support multiple types of documents, in this paper, it is analyzed and explained according a RE perspective, namely focused on Software Requirements Specifications (SRSs). SRS is a document that describes technical concerns of a software system, and serves to share the system view amongst its stakeholders throughout different stages of the project life-cycle. It usually follows a previously defined template prescribing the use of multiple modular artifacts corresponding to different views.

We found in the literature a diversity of SRS templates that can be integrated and supported in ITBox, such as IEEE Std 830-1998 (IEEE 830 template) [14], RUP Software Requirements Specification Template (RUP template) [13], Withall template [12], or RSLingo RSL [6]. The majority of these templates are structured in chapters and some complementary appendixes; and they recommend the use of different constructs and models structured at different abstraction levels. For the sake of simplicity we discuss in this paper the RSL language and its respective template.

RSLingo is a RE research initiative that recognizes that natural language, although being the most common and

preferred form of representation used within requirements documents, is prone to produce such ambiguous and inconsistent documents that are hard to automatically validate or transform. Originally RSLingo proposed an approach to use simplified natural language processing techniques as well as human-driven techniques for capturing relevant information from ad-hoc natural language requirements specifications and then applying lightweight parsing techniques to extract domain knowledge encoded within them [3]. More recently, RSLingo's RSL (or just "RSL" for brevity) was proposed based on former languages and experiments [5,6]. RSL is a controlled natural language to help the production of SRSs in a systematic, rigorous and consistent way. RSL includes a rich set of constructs logically arranged into views according to multiple RE-specific concerns that exist at business and system abstraction levels. These constructs are defined as linguistic patterns and represented textually by mandatory and optional fragments [6]. For example, the people and organizations that can influence or will be affected by the system are represented in the Stakeholders view. Other constructs for requirements in RSL are business and system goals, functional requirements, quality requirements, constraints, user stories, and use cases [5,6]. The RSL Excel Template¹ is based on a multi-view architecture and serves to document and model system requirements and other concerns in a set of different views, adapted to an Excel format.

III. ITBOX OVERVIEW

The objective of this research was to design and develop a collaborative platform for managing technical documents and so, we established the following key goals:

Support different types of languages, formats and templates. ITBox shall provide a web-based interface able to manage multiple projects with several documents and able to author multiple document formats and templates. As a result of this goal, it shall be necessary to integrate multiple editors, able to support such variety of formats.

Collaborative and multi-project platform. ITBox shall support multiple users. ITBox users shall be able to perform different tasks depending on the roles they may have, such as invite new users, create a new project, add documents to a project, manage libraries of requirements. Therefore, user roles were defined, each one having different allowed actions, both at the platform-level and at the project-level and their respective artifacts. Concerning the collaborative aspect in terms of synchronization between all the platform's existing documents, cloud storage had to be adopted, namely Google Drive. Making use of various Google Drive API services, it is possible to ensure that all users can access and edit concurrently the same documents' versions, simultaneously.

Reusability and variability mechanisms. To increase the efficiency of the documents manipulation, ITBox provides three different mechanisms which contribute for the automation, and consequent time-saving, of some user actions. These mechanisms are discussed in Section 5.

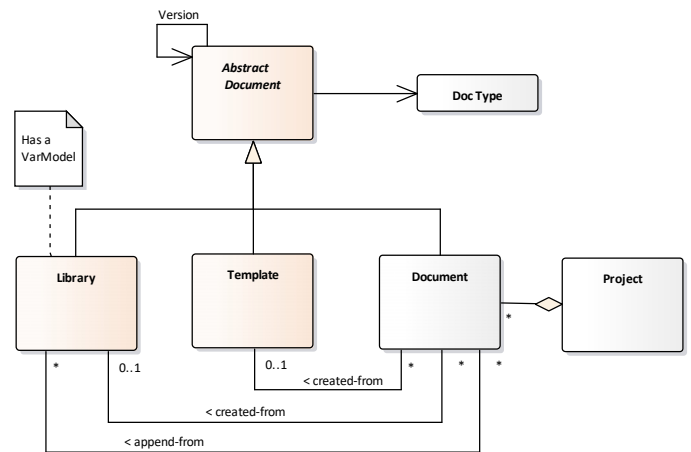


Fig. 1. The ITBox Domain Model (Documents)

Figure 1 presents a partial view of the ITBox domain model, showing its key concepts. (For the sake of simplicity it does not show User, Roles etc.) For a clear understanding of the remaining paper, the comprehension of this figure is relevant; therefore, a brief clarification follows.

Project represents the main social object supported by ITBox. Projects can contain different documents, of different *DocTypes*, and can be associated to multiple users with different roles (e.g., project owner, member or consultant; see details below). Any user can create a project and then can invite other users to be assigned to it.

Abstract Document is an abstract generalization for the other concepts like *Library*, *Template* and *Document*. Internally, ITBox manages Abstract Documents equally, as they are all stored into the same Google Drive repository, once they are uploaded into the platform. Due to the fact ITBox uses Google Drive and involved Google editors any Abstract Document can be accessed and edited concurrently.

DocType supports the management of different types of document, such as RSLingo RSL, RSLingo RSL Excel template, RUP use cases template, IEEE 830 template, etc.

Document contains in a consistent way a set of specifications defined according its respective *DocType* and is defined within a *Project*. For example, a SRS document of type "RSL Excel template" shall include several RE specifications according the sheets defined in such Excel.

Library is a set of modular and coarse-grained requirements organized by type, and can be later on associated with a *Document* by "created-from" or "append-from" its requirements to the ones contained in a specific *Document*. Furthermore, it is possible to define variability models (i.e., VarModels) for a *Library*, still increasing its reusability capabilities.

Template is a reusable document that means it can be used to bootstrap the creation of new documents in ITBox (showed in Figure 1 as the "created-from" relation). A template has pre-filled specifications that can then be reused by a newly created *Document*, saving the time of manually retyping the information and thus increasing the productivity of the *Document* setup process.

¹ <https://github.com/RSLingo/RSL-Excel-Template>

IV. ITBOX – COLLABORATIVE FEATURES

After analyzing similar collaborative platforms, it came clear that to provide a strong collaborative environment, a set of features would have to be offered, namely users management features with role-based access control method. In general, each user can have multiple roles assigned, and this user-role assignment shall be managed dynamically.

A. User Roles

ITBox provides a role-based access control (RBAC) method defined at two levels: platform- and project-level.

Platform-level roles. Depending on the ITBox roles, users may have different permissions to access the platform’s pages and widgets, determining the control that they may have over the platform: **ITBox Admin** has access to all ITBox pages (e.g., Home, Projects, Templates, Libraries, Settings) and can perform every existing action in the platform. Has full control over the remaining users and their associated roles. **ITBox Manager** has access to the Home, Templates and Libraries pages. The users which have this role are responsible for assuring the quality of the documents they upload, as those documents will be accessible to the remaining users for serving as templates for new project documents or to append to existing documents, depending if they are Templates or Libraries, correspondingly. **ITBox User** has access to the Home and Projects pages. While the ITBox Admin can access and manage all the existing projects in the platform, users which have the ITBox User role can only access the projects they belong to, assuring the projects’ confidentiality. **ITBox Guest** can sign up, sign in or recover the password of an existing account, and when sign in becomes a authenticated ITBox user.

Project-level roles. Any ITBox user can create a project, and then she becomes the Project Owner of that created project and may associate documents, etc. Therefore, when defining the Project Roles, it is reasonable to detail not only the actions that the different roles can perform in projects, but also in the project documents. **Project Owner** has full control over the projects that he/she owns. A project owner can view and manage (add and delete) project documents; manage the project members (invite and remove) and their project roles. Additionally, she can modify some of the project’s properties, like the name, description and status. **Project Member** can view and edit (including append libraries), the project documents. She can also invite other users to join the project, and to decide the role they will have. Project member can still modify some of the document’s properties, like the name and status. **Project Consultant** can only consult (view) the project documents. Although, project consultants can ask the project owner or a project member to have editing permissions, becoming project members.

B. Use Cases

Platform-level Actions. Table 1 summarizes the use cases commonly supported by ITBox at the platform-level management. In addition, this table shows for each of these use cases (or actions) the involved granted roles.

Similarly, Table 2 summarizes the use cases supported by ITBox at the project-level, as well as, for these use cases, the granted roles.

Table 1. Platform’s allowed actions by the different ITBox roles

Actions vs. Platform-level Roles	Admin	Manager	User	Guest
Access Home page	√	√	√	√
Sign up				√
Sign in				√
Recover password				√
Edit profile	√	√	√	
Access Projects			√	
Manage users	√			
Manage ITBox roles	√			
Manage DocTypes	√			
Manage Libraries		√		
Manage Template		√		

Table 2. Projects and documents’ allowed key actions per Project-level roles

Actions vs. Project-level Roles	Owner	Member	Consultant	
Project	View docs	√	√	√
	Manage docs	√	√	
	Manage members	√	√	
	Modify properties	√		
	Create a project	√	√	√
	Change status	√		
	Delete	√		
Project Documents	View	√	√	√
	Edit	√	√	
	Append library	√	√	
	Modify properties	√	√	
	Change status	√	√	
	Delete	√		

V. ITBOX – REUSABILITY FEATURES

ITBox offers some reusability features, including (i) features for uploading and managing templates; (ii) features for creating and reusing modular sets of reusable requirements (libraries); and (iii) a variability modeling framework that leverages the concepts of the CVL language to model variability aspects in the RE context.

A. Templates

The template management system offers the possibility to manage reusable templates that can be later used to bootstrap the creation of documents. The process of creating a template in the platform encompasses two steps. The first consists on the manual upload of a file containing the template into the platform. This document doesn’t need to hold the finalized template (it can even be empty), as ITBox allows it to be further edited once it is uploaded. Once the file is uploaded, it is saved into the local filesystem and the second step begins. This time, the file is automatically re-uploaded to the Drive cloud storage server. This template is then shared with any users allowed to access to it. Then it can be accessed and edited concurrently, and can be used as the base structure for new document created in ITBox.

B. Libraries

The reusability based on libraries offers the possibility to create libraries of reusable specifications that can then be added to any project's document. Requirements reusability across multiple projects is already one of the main ideas being explored by some of the platforms found in the related work regarding the efficiency increase of RE activities. To achieve this, each ITBox library serves as an open set of modular, agnostic and coarse-grained specifications that are classified depending on the library type. These libraries shall be created by a certain kind of users (of course with more mature experience and knowledge of some specific domain or expertise), which ensure the quality of the requirements specifications. Furthermore, these users can still add and remove specifications to these modules, allowing a continuous process of refinement and improvement. On the other hand, the process of adding a specification library to a project document is simple, since the new specifications only need to be "appended" to the existing ones. To conclude, libraries can also be used as basis for new project documents.

C. Variability Modeling

The reusability based on variability features is aligned with the Common Variability Language (CVL) [7,8], the OMG proposal for a domain independent variability modeling standard, which allows the creation of variability models based on requirements specification documents. In practice, this allows the representation the system's features variability to be independent from the Domain Specific Language (DSL) used to model those features. This domain-independence makes CVL an ideal language to develop a framework for modeling variability at the requirements level, incentivizing experimentation and the proposal of new approaches in the field.

As suggested in the Figure 1 it is possible to still increase the extensibility of ITBox libraries with variability models [4]. Currently ITBox supports the creation of Variability Models from the requirements views of the RSL Excel Model. Therefore, Goals, Functional Requirements, Quality Requirements and Constraint Requirements allow a wide scope of variability points within the spectrum of requirements engineering concerns. ITBox allows the creation of new documents from libraries with variability models previously defined, originating in resolved models.

VI. CONCLUSION

Requirements specification is a fundamental RE activity which purpose is to produce a shared and consistent vision among the stakeholders of the problem to be addressed. Therefore, an effective collaborative tool to manage requirements specification documents can not only facilitate the way how these documents are managed but shall also help to guarantee an alignment of expectations between all the involved parties.

This paper proposed ITBox, a collaborative web-based platform for better support the specification and the management of such types of technical documentation. In particular, ITBox allows the management of SRSs

privileging the quality and rigorousness of the requirements, based on the RSLingo approach. ITBox offers a set of new collaborative and reusability features, namely a variability modeling framework that leverages the concepts of the CVL language to model variability aspects in the context of Requirements Engineering.

Future work will research and evaluate the usability aspects of the ITBox, as well as to disseminate and promote its use as an open platform. In addition, we intend to develop RE reusable libraries for multiple domains and concerns, such as for privacy, personal data protection, usability, information security, etc.

To conclude, in spite of we have presented ITBox according to a RE perspective, however, ITBox already supports other documentation types such as tests specifications or project plans. As future work we also intend to research and develop methods to support traceability and automatic transformations between specifications defined in different types of documents.

ACKNOWLEDGMENT

This work was partially supported by national funds under FCT project UID/CEC/50021/2013.

REFERENCES

- [1] K. Pohl, *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer, 2010.
- [2] K. El Emam, G. A. Koru, "A replicated survey of IT software project failures", *IEEE Software*, vol. 25(5), pp. 84–90., 2008.
- [3] D. Ferreira, A. R. Silva, "RSLingo: An information extraction approach toward formal requirements specifications", *Proceedings of MoDRE'2012*, IEEE CS, 2012.
- [4] A. R. Silva, J. C. Fernandes, S. Azevedo, "Variability Aspects at a Textual Requirements Specification Level", *RE Workshops 2017*, IEEE Computer Society.
- [5] A. R. Silva, "A Rigorous Requirement Specification Language for Data-Intensive Enterprise Information Systems: Focus on RSL's Data Entities, Uses Cases and State Machines", *INESC-ID Technical Report*, 2017.
- [6] A. R. Silva, "Linguistic Patterns and Linguistic Styles for Requirements Specification (I): An Application Case with the Rigorous RSL/Business-Level Language", *Proceedings of EuroPLOP'2017*, ACM, 2017.
- [7] OMG, *Common Variability Language (CVL) OMG Revised Submission*, 2012.
- [8] H. Øystein, B. Møøller-Pedersen, J. Oldevik, G. K. Olsen, A. Svendsen, "Adding standardized variability to domain specific languages", *Proceedings - 12th International Software Product Line Conference, SPLC 2008*, 2008.
- [9] J. Beatty, A. Cardenas, J. Bartlett, M. Stowe, D. Reinhardt, *Requirements Management Tool Evaluation Report*, 2016.
- [10] L. Eveleens, C. Verhoef, "The Rise and Fall of the Chaos Report Figures", *IEEE Software*, 2010.
- [11] Standish Group, *Chaos Summary 2009 Report, The 10 Laws of Caos*, 2009.
- [12] S. Withall, *Software Requirements Patterns*, Microsoft Press, 2007.
- [13] IBM: *Rational Method Composer and RUP on IBM Rational developerWorks*, 2013.
- [14] IEEE: *IEEE Std 830-1998 (Revision of IEEE Std 830-1993)*. IEEE Recommended Practice for Software Requirements Specifications, 1998.