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

Pedro Cecílio Lopes[1] and Alberto Rodrigues da Silva[1]

[1] INESC-ID, Instituto Superior Técnico, Universidade de Lisboa (IST-UL),
Lisbon, Portugal
{pedrocecilio, alberto.silva}@tecnico.ulisboa.pt

Abstract. Requirements engineering 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 requirements specifications is crucial to support 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 requirements documents with an emphasis on ITBox most innovative aspects such as on social, reusability and productivity features. In addition, this paper shows the evaluation preliminary results from some user sessions conducted recently. Also, this paper compares and discusses the ITBox with other related tools.

Keywords: Collaborative Platforms • Requirements Engineering • Reusability

1 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 define 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 types of stakeholders, to achieve a common understanding of the system to be developed. The adverse consequences of disregarding the importance of early RE activities are well-known and extensively reported in the literature, such as in the Chaos reports [10, 11]. To avoid those consequences, interpretation problems shall be reduced and, therefore, an effective communication shall be supported, 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. Small businesses and large enterprises alike seek to deliver more business value, reduce rework, and eliminate budget overruns that happen too frequently due to poor requirements management [9]. To mitigate these problems, users must adopt the most suitable approach methods and tools for their projects.

ITBox is a collaborative web platform which that 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. It combines some common and relevant features of other tools available, while offering innovative reusability features, like model variability aspects in the field of requirements engineering using the CVL language [7]. 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.

This paper is organized in 8 sections. Section 2 introduces the research background in which ITBox has been developed, namely the RSLingo approach and the RSL language. Section 3 introduces the ITBox main goals and its main concepts. Sections 4 and 5 detail ITBox's collaborative and reusability features, respectively. Section 6 discusses the preliminary results obtained from the conducted user sessions. Section 7 briefly analyses and compares requirements management tools with the ITBox proposal. Finally, Section 8 concludes the paper and lays down some ideas for extending the current work.

2 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 just following a requirements engineering perspective.

System requirements specification, software requirements specification or just requirements specification (SRS) is a document that describes multiple technical concerns of a software system. An SRS is used throughout different stages of the project life-cycle to help sharing the system vision among the main stakeholders, as well as to facilitate the communication and the overall project management and system development processes [1, 15].

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], Volere Requirements Specification Template [16] and RSLingo RSL [6]. The majority of these templates are structured in chapters and some complementary appendixes; and they explicitly or implicitly recommend the use of different constructs and models structured at different abstraction levels such as at business and system levels. For the sake of simplicity, we consider in this paper the RSLingo RSL language and respective templates; however, other templates could be considered.

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 ambiguous and inconsistent documents that are hard to automatically validate or transform.

Originally RSLingo proposed an approach to use simplified Natural Language Processing (NLP) 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 the design of former languages and experiments [6]. RSL is a control natural language to help the production of SRSs in a more systematic, rigorous and consistent way. RSL is a language that 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 entities that can influence or will be affected by the system are represented in the Stakeholders view. Likewise, the objectives of business stakeholders regarding the value that the system shall bring are expressed in the *BusinessGoals* view. Other constructs for requirements found in RSL are system goals, functional requirements, quality requirements, constraints, user stories, and use cases [6].

The RSL Excel Template[1] is based on the multi-view architecture defined in the RSL language and serves to document and model system requirements and other RE concerns in a set of different views using the RSL language, adapted to an Excel format.

3 ITBox Overview

The purpose of our research was to design and develop a collaborative platform for better managing technical documents. In particular, and given the needs of the RE process for a strong cooperation among the involved stakeholders, we define the following research goals (RG).

RG1. Collaborative and multi-project platform. ITBox shall support multiple users and multiple projects. 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 created, each one having different granted actions, both at the platform-level and at the project-level. Still concerning the collaborative aspect in terms of synchronization between all the platform’s existing documents, cloud storage had to be adopted, namely Google Drive[2]. 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.

RG2. Support different types of formats and templates. ITBox shall provide a web-based interface able to manage multiple projects with many documents and able to author multiple document formats and

[1] https://github.com/RSLingo/RSL-Excel-Template

[2] https://developers.google.com/drive/

templates. For example, adopting the RSLingo approach, as referred previously in section 2, it should be possible to use not only RSL Excel templates, but also the RSL language directly as well as other templates. As a result of this goal, it shall be necessary to integrate multiple editors, able to support such variety of formats.

RG3. Extensibility to support multiple languages and approaches. RSLingo proposes a rigorous language to define requirements specifications, i.e., the RSL language. In the future, similar rigorous languages might appear, but applied to different types of technical documents. Therefore, it shall be possible to add the support of new languages to the ITBox platform. To achieve this, ITBox shall provide the management of types of documents (and languages), accessible to the platform's administrator.

RG4. Reusability and variability mechanisms. To increase the efficiency of the documents manipulation, ITBox has three different mechanisms which contribute for the automation, and consequent time-saving, of some user actions. These mechanisms are detailed in section 5.

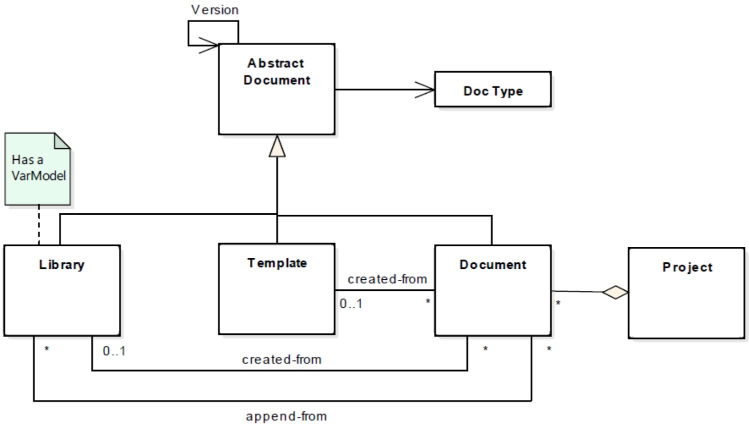

Fig. 1. The ITBox Domain Model (Documents)

Figure 1 presents part of the ITBox domain model, showing some of its key concepts. For a clear understanding of the remaining paper, the comprehension of this figure is relevant; therefore, a brief clarification about its concepts follows.

Project represents the main concept 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). 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 concrete concepts such as *Library, Template* and *Document*. Internally, ITBox manages Abstract Documents equally, as they are all stored as files 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 (e.g. Google Docs and Google Sheets), 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 or IEEE 830 template.

Document contains in a consistent way a set of specifications defined according its respective *DocType* and is defined within a *Project* (also multiple *Documents* can be associated to the same *Project).* For example, a SRS document of type "RSL Excel template" shall include several RE specifications according the sheets defined in such Excel template.

Library is a set of modular and coarse-grained requirements, usually organized by type, and can be later on associated with a *Document* by appending its requirements to the ones already contained in this *Document*. Furthermore, it is possible to define variability models (*VarModel)* for a *Library*, still increasing its reusability.

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

Some ITBox objects have an attribute to identify their state. Depending on the state an object is in, the allowed actions over that object change. Below, there is a description of the existing platform's objects with state and their corresponding state machines:

ITBox **project objects** have a "ProjState" attribute.

After a project is created, it becomes in the “Not Started” state. While the project is in this state, no documents or users can be added to it. Basically, this state only defines the existence of the project, as it does not allow to perform any changes on the projects (beyond their properties, like name and description).

From the “Not Started” state, the Project Owner can change the project state to “On Execution”, in case he wants to start the project. This is the only state in which is possible to add new documents and users to the project.

From both the “Not Started” and “On Execution” states, it is possible to cancel the project. This action might occur when a project is canceled for some reason. When a project is canceled, it has the same behavior as a “Not Started” or “Concluded” project, i.e., it is not allowed to modify the projects’ artifacts (documents and users). Both these states are terminal states, which means that after reaching one of these states, projects can no longer have other state.

Finally, the “Concluded” state is only possible to achieve if a project is in the “On Execution” state. When the Project Owner decides to finish the project, he changes its state to “Concluded”, not being possible to add new documents and users, and edit the project documents.

Figure 2 illustrates the transitions described above, through a state machine.

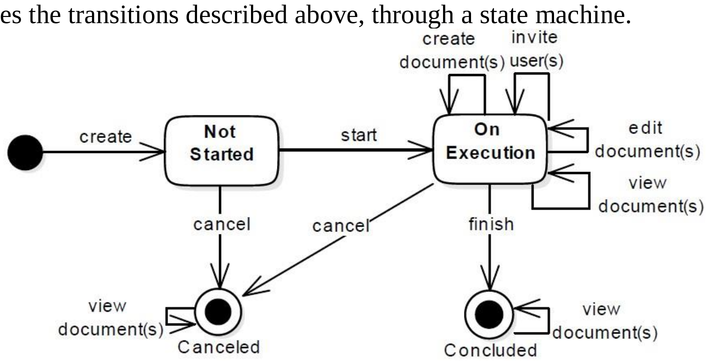

Fig. 2. Project State Machine

ITBox **project documents objects** have a “DocState” attribute.

After a project document is added to a project, the “On Edit” state is attributed to it. While the document is in this state, it can be edited by users who have this permission. From this state, documents might need to be approved, transiting to the “On Approve” state, or can be closed, going to the “Closed” state.

When a document is in the “On Approve” state, only the Project Owner can change its state. If the Project Owner decides that the document needs to be updated, he changes the state back to “On Edit”. If the Project Owner decides that the document can be approved, he changes the state to “Approved”. Finally, the Project Owner can also consider that the document does not need any more changes, changing its state to “Closed”.

From the “Approved” state, a document state has two possible transitions: a new version of the document might be required, making the document transit to the “On Edit” state, with a new version; or, the document has reached it final version, transiting to the “Closed” state.

Finally, when a document is in the “Closed” state, it can no longer be edited, but can still be viewed. This is a terminal state, which means that a document cannot transit to a different state after reaching this one. Figure 3 illustrates the transitions described above, through a state machine.

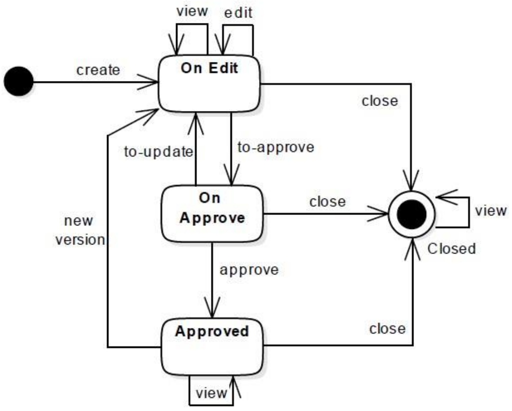

Fig. 3. Document State Machine

4 ITBox – Collaborative Features

ITBox is a project-centric collaborative environment providing a set of collaborative features with a role-based access control method. In general, each user can have multiple roles assigned, and this user-role assignment is managed dynamically. ITBox provides a role-based access control (RBAC) method [17] managed at two levels: at the platform-level and at the project-level.

Table 1 summarizes the common actions supported by ITBox at the platform-level. In addition, this table shows for each of these actions the granted roles.
Similarly, Table 2 summarizes the actions supported by ITBox at the project-level, as well as, for each of these actions, the granted roles.

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.

ITBox Admin role grants access to all the ITBox pages related with the management of the existing users, roles and *DocTypes*.

ITBox Manager role grants access to the Home, Templates and Libraries pages. The users with this role are responsible for assuring the quality of the documents they upload, as those documents shall 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 role grants has access to the Home and Projects pages. Users with this role can only access the projects they belong to, assuring the projects' confidentiality.

ITBox Guest role corresponds to current or future users of the ITBox platform. Once they reach the Home page, they can sign up, sign in or recover the password of an existing account. After signing in, users become at least ITBox user.

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

	Platform-level Role Action	**ITBox Admin**	**ITBox Manager**	**ITBox User**	**ITBox Guest**
ITBox Platform	Access Home page	√	√	√	√
	Sign up				√
	Sign in				√
	Recover password				√
	Edit profile	√	√	√	
	Access Projects			√	
	Manage Users	√			
	Manage ITBox Roles	√			
	Manage *DocTypes*	√			
	Manage Libraries		√		
	Manage Templates		√		

Project-Level Roles. Documents are associated and managed at a project level. Therefore, when defining the Project Roles, it is reasonable to detail not only the actions that these roles can perform in a project level, but also at the level of the project documents.

Project Owner role grants full control over the projects that he owns. This means that a project owner can: view and manage (add and delete) the project documents; manage the project members (invite and remove) and their project roles (Project Member or Project Consultant). Additionally, the project owner can modify the project's properties, like the name, description and state.

Project Member role grants members to view and edit (including append libraries) project documents. They can also invite other users to join the project, and to decide the Project Role they shall have. Project members can still modify some of the document's properties, like the name and state.

Project Consultant role grants users to only consult (view) the project documents. Although, later on, project consultants can ask a project owner or a project member to have editing permissions, becoming project members.

Table 2. Projects and documents allowed actions by the different project roles

	Project-level Role Action	Project Owner	Project Member	Project Consultant
Project	Manage Project Members	√	√	
	Modify Project Properties	√		
	Create a new Project	√	√	√
	Change Project State	√		
	Delete	√		
Project Documents	View Documents	√	√	√
	Edit Documents	√	√	
	Append Library to Document	√	√	
	Modify Document's Properties	√	√	
	Change Document's State	√	√	
	Delete Document	√		

The action of creating a new project is independent from the Project Roles, because once a ITBox user creates a new project, he becomes the Project Owner of that newly created project.

5 ITBox – Reusability Features

ITBox offers several features for managing requirements specifications, in particular reusability features including: (i) a template management system, for uploading/managing SRS templates; (ii) a library management system, for creating and reusing modular sets of reusable requirements; and (iii) a variability modeling framework that leverages the concepts of the CVL language to model variability in the context of requirements engineering.

5.1 Templates of Specifications

The template management allows to manage reusable templates that can be later used to bootstrap the creation of new documents. The process of creating a new template 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 to the ITBox filesystem, it is automatically re-uploaded to the Google Drive cloud storage server, being available to the users who can access it. From this point on, it can be accessed and edited concurrently like any other document, and it can be used as the base structure for new document created in ITBox.

5.2 Libraries of Specifications

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 one of the main ideas being explored in this research. To achieve this, each ITBox library serves as an open "database" of modular, coarse-grained specifications, that are classified depending on the library type. These groups of specifications must be created manually, by a certain kind of advanced users, which ensure the quality of the requirements specifications. Furthermore, after created, these advanced users can still add and remove specifications. The process of adding a specification library to a project document is simple, since these specifications only need to be "appended" to the existing ones. To conclude, specification libraries can be used as a basis for new project documents.

5.3 Variability Modeling

The reusability based on variability features is aligned with the Common Variability Language (CVL) [7], 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 languages used to model those features. This domain-independence makes CVL an ideal language to develop a framework for modeling variability at the requirements level, fostering 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. ITBox supports the creation of variability models from the existing libraries, allowing the variability models to be created 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 resolved models.

6 Evaluation

To better evaluate the ITBox platform, and to receive feedback from people not directly involved in this research, we decided to conduct a pilot user session. This session also allowed to detect potential bugs and user limitations. It involved a group of 10 participants in total with ages ranging from 22 to 50 years old and with at least a Bachelor of Science degree. All participant had previous knowledge and experience within the field of RE and 60% had professional experience. The user session was conducted under the following conditions: session took place in the laboratory (controlled environment); the assigned tasks were performed without previous use and learning (for the first time); the user must have had a computer with a web browser and Internet access; Direct Observation, i.e., while users performed the assigned tasks, their behavior and performance could be logged; users were free to think out loud and share ideas if they wanted.

Based on these conditions, participants received a 20 minutes presentation and explanation about the fundamental concepts of the ITBox platform and its features, particularly the ones tested during this session. Following that, they were given a script describing the steps they should follow. The work consisted in uploading a template, creating a project, adding the previously uploaded template to the newly created project, appending a quality requirements library to the project's document, and inviting a new user (with the Project Consultant role) to be part of the project. To complete these tasks, the users had a time limit of 30 minutes. In the end, participants were asked to fill in a questionnaire to rate the ITBox platform, and suggest improvements.

The questionnaire used in the user session focused on analyzing the quality of three aspects about ITBox: (1) the overall usability and the quality of its features, (2) the collaborative environment's quality and (3) the usefulness of the platform. The answers regarding the first two questions were classified in a scale of: 0 (N/A – Do not know), 1 (Very Low), 2 (Low), 3 (Medium), 4 (High) and 5 (Very High). The possible answers for the questions regarding the third aspect are presented below.

Regarding the ITBox web platform usability and its features' quality, the questionnaire contained four questions:

Q1. How do you rate the overall usability of the Web platform?

Q2. How do you rate the usefulness of the main productivity features? (Projects Management, Doc Editor, Template Manager, Libraries)?

Q3. How easy to learn (or how familiar) was the main document editing tool (Google Sheets)?

Q4. How suitable is the platform for managing Technical Documentation?

Table 3 summarizes the average score for the answers regarding the questions above, broken down by question.

In general, all the questions had very positive score, implying that the platform was successful in accomplishing its goals. To note that one of the answers given to question Q2 was 0 (N/A – Do not know). This was also the question with the lowest score, which suggests that the productivity features were not clear enough for the pilot-users. Regarding the other three questions, the responses were very positive, showing that the users considered the platform's features to be useful and easy to learn. Furthermore, the platform itself was considered very suitable for managing technical documentation.

Table 3. Survey average score (in a scale of 0-5) by question for the ITBox usability and its features' quality.

	Q1	Q2	Q3	Q4
Average	4.5	4.4	4.6	4.5

The collaborative environment's quality aspect also included four questions:

Q1. How do you rate management of user roles in projects?

Q2. How do you rate the suitability of the existing Project Roles (Owner, Member, Consultant)?

Q3. How do you rate the usefulness of having different user roles in the same project?

Q4. How do you rate the simplicity of adding a new user to a project?

Table 4 summarizes the average score for the answers regarding the questions above, broken down by question.

Similarly to the previous aspect, all the answers had a very positive score. The question regarding the suitability of the existing Project Roles (Q2) was the one with the lowest average score, suggesting that some improvement can be made regarding this aspect. However, users considered very useful and simple the act of adding new users to projects.

Table 4. Survey average score (in a scale of 0-5) by question for the ITBox collaborative aspect.

	Q1	Q2	Q3	Q4
Average	4.5	4.3	4.6	4.6

Finally, to evaluate the usefulness of the platform, two questions were made. The first question had as possible answers: (1) Yes, (2) No, different technical documents should be managed in different platforms and (3) Other, in which users could justify their answers as they wish. Regarding the second question, two answers were possible: (1) Yes and (2) No.

Q1. Do you think is useful to have just one platform to manage multiple projects of different technical documentation?

Q2. Would you use this platform on your own Engineering projects?

In both questions, all the users responded positively, i.e., “Yes”. This means that they think the platform is useful for managing multiple projects of different technical documentation, and furthermore, they would use ITBox on their future projects.

As can be seen in Table 5, the results were encouraging with very positive scores in all the three analyzed aspects. Nevertheless, it was observed that the platform’s usability, and its features’ quality, namely the collaborative environment can be refined. These average results were converted to percentage, so that all the results are in an equal scale.

Table 5. Survey average score (in percentage) for each of the questionnaire aspects.

	Usability and Features	Collaborative Environment	Platform’s usefulness
Average	90%	90%	100%

Regarding the number of participants, it can be stated that the sample was adequate to extract meaningful conclusions from the results. Some studies have noted that a group of 10 participants will find over 90% of the usability problems [18] and so the user session represented a considerable example of what could be expected from a bigger number of participants.

7 Related Word

There are several requirements management (RM) tools available on the market, which show and provide important features such as requirements traceability and project life-cycle integration. Seilevel[3] developed an evaluation report which summarizes the main existing RM tools available [9], highlighting the pros and cons of each tool. Through this report, we could not only get a perspective of the features that these tools provide, but also to understand what the ITBox could offer better.

TopTeam Analyst [4] is an end-to-end solution for requirements definition and requirements management, which is heavily focused on the user experience around managing requirements. TopTeam offers features like: an intuitive administration module for requirements architecture; managing baselines and document versions; and a comprehensive document management that allows teams to design custom templates and generate requirements documents very quickly. Besides that, it allows users to have roles like Business User, Product Manager, Customer, QA Manager or Team Lead, and each has different views and dashboards unique to that role. Besides providing some of these features, ITBox also offers requirements reuse based on libraries with variability models, and real-time collaboration.

[3] https://www.seilevel.com/

[4] http://www.technosolutions.com/topteam_requirements_management.html

Modern Requirements[5] offers a suite of requirements management products. Instead of having one comprehensive tool with a host of features offered to every user, eDev Technologies offers a few different tools that sit on top of MTFS (Microsoft Team Foundation Server) with modules that can be selected and used based on the customer needs. The tool behind the Modern Requirements suite is InteGREAT Studio[6]. It offers an intuitive requirements traceability feature, and contains full requirements modeling capabilities, allowing the creation of high-fidelity prototypes. Due to its architecture, not every feature is available in every module, so users may have to switch between the web application, Microsoft Office applications and the desktop application to fully manage their end-to-end process. ITBox is a web application and is fully usable through a browser. It makes use of online applications like Google Docs and Google Sheets, overcoming some of the Modern Requirements limitations.

Visure Requirements[7] is a flexible and complete Requirements Engineering lifecycle solution capable of streamlining requirements processes, allowing more effective collaboration and increasing quality. Visure strongest RE features are: the support of user-defined views, full traceability, requirements re-use and requirements baselining. Discussion forum is a feature that facilitates discussion between the teams, allowing them to share their questions and comments with other stakeholders. Visure supports user-defined views, based on different roles assigned to the users. Concerning the communication between the stakeholders, ITBox allows users to comment on the documents that they have access to. This feature is provided through the usage of the Google editors (Google Docs and Google Sheets).

IBM Rational DOORS[8] is a requirement management tool that makes it easy to capture, trace, analyze, and manage changes to information. It is a client–server application, with a Windows-only client and servers for Linux, Windows, and Solaris. There is also a web client, DOORS Web Access. Rational DOORS has its own programming language called DOORS eXtension Language (DXL). ITBox supports different technical documentation, which might be in different languages.

IBM Rational RequisitePro[9] is a requirements management tool that teams can use to manage project requirements comprehensively in order to promote communication and collaboration among team members and reduce project risk. It integrates with Microsoft Word to provide a familiar environment for activities such as requirements definition and organization. RequisitePro also includes views of requirements that display attributes, parent/child relationships, traceability, and the impact of requirement change. ITBox makes use of Google Docs, providing a familiar editor similar to Microsoft Word, but with the advantages of offering real time collaboration and not requiring additional installed programs.

ITBox[10] is a collaborative platform for managing projects' technical documentation. It supports multiple types of technical documentation (such as project plans, requirements or tests specifications) specified according different languages and templates. ITBox's focused on innovating social, reusability and productivity features. Therefore, ITBox main features comprehend custom templates, document versions, requirements reuse based on libraries with variability models, user-defined views, real-time collaboration and projects and documents states (through respective workflows).

8 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 supports the management of requirements specification documents, offering 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 RE.

[5] https://www.modernrequirements.com/

[6] https://www.modernrequirements.com/integreat-studio/

[7] https://visuresolutions.com/requirements-engineering-tool/

[8] https://www.ibm.com/us-en/marketplace/rational-doors

[9]https://www.ibm.com/support/knowledgecenter/en/SSSHCT_7.1.0/com.ibm.reqpro.help/get_start/c_product_overview.html

[10] http://itbox.inesc-id.pt/

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 or information security.

In spite ITBox has been presented in this paper according to the RE perspective, it already supports other documentation types such as tests specifications or project plans. So, as future work, we intend to research and develop methods to better support traceability and automatic transformations between specifications defined in different types of documents.

References

1. Pohl, K. Requirements Engineering: Fundamentals, Principles, and Techniques. Springer. (2010)
2. El Emam, K., & Koru, G. A. A replicated survey of IT software project failures. IEEE Software, 25(5), 84–90. https://doi.org/10.1109/MS.2008.107 (2008)
3. Ferreira, D., Silva, A. R., 2012. RSLingo: An information extraction approach toward formal requirements specifications, Proceedings of MoDRE'2012, IEEE CS.
4. Foster, H., Kroklnik, A., & Lacey, D. Assertion-Based Design (2nd ed.). Springer. (2005)
5. Firesmith, D. Specifying good requirements. Journal of Object Technology, vol2 (no. 4), 2(4), 77–87. https://doi.org/10.5381/jot.2003.2.4.c7 (2003)
6. Silva, A. R., 2017. Linguistic Patterns and Styles for Requirements Specification: The RSL/Business-Level Language, Proceedings of EuroPLOP'2017, Hillside Europe, 2017.
7. OMG. Common Variability Language (CVL) OMG Revised Submission, 2012.
8. Øystein Haugen, Birger Møoller-Pedersen, Jon Oldevik, Gøran K. Olsen, and Andreas Svendsen. Adding standardized variability to domain specific languages. In Proceedings - 12th International Software Product Line Conference, SPLC 2008, pages 139–148. (2008)
9. Beatty, J., Cardenas, A., Bartlett, J., Stowe, M., & Reinhardt, D. Requirements Management Tool Evaluation Report. (2016)
10. Eveleens, L., Verhoef, C., 2010. The Rise and Fall of the Chaos Report Figures, IEEE Software.
11. Standish Group, 2009. Chaos Summary 2009 Report, The 10 Laws of Caos.
12. Withall, S.: Software Requirements Patterns, Microsoft Press, 2007.
13. IBM: Rational Method Composer and RUP on IBM Rational developerWorks, 2013. http://www.ibm.com/developerworks/rational/products/rup/
14. IEEE: IEEE Std 830-1998 (Revision of IEEE Std 830-1993). IEEE Recommended Practice for Software Requirements Specifications, 1998.
15. Sommerville, I. and Sawyer, P. Requirements Engineering: A Good Practice Guide. Wiley, 1997.
16. Robertson, J., & Robertson, S. Volere. Requirements Specification Templates.
17. Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. Computer, 29(2), 38-47. (2000)
18. Jakob Nielsen and Thomas K. Landauer. A Mathematical Model of the Finding of Usability Problems. Transport, pages 206–213, 1993.