ITBox: A Collaborative Platform for Better Managing Technical Documentation

Pedro Miguel Cecílio Lopes

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Alberto Manuel Rodrigues da Silva

Examination Committee
Chairperson: Prof. João Emílio Segurado Pavão Martins
Supervisor: Prof. Alberto Manuel Rodrigues da Silva
Member of the Committee: Prof. Paula Cristina Negrão Ventura Martins

October 2017
Acknowledgments

First, I would like to thank my supervisor, Professor Alberto Silva, for his guidance during the whole process. His constant availability for answering any question, permanent guiding through this dissertation and his words of encouragement were always very important. Furthermore, his expertise in the field of Requirements Engineering and his suggestion on the topics to investigate were fundamental to steer the thesis in the right direction.

I also would like to thank all the colleagues at INESC-ID who helped me to develop and test the platform. A special “thank you” to João Fernandes, who helped me during the initial development phase, and was always available for clarifying my doubts.

I would especially like to thank my family for all the support through the time it took to reach this point. To my mother, my father, my sister and my grandmother, an enormous “thank you”. Their encouragement and support were fundamental for me to achieve this point.
Abstract

Requirements specification is a fundamental requirements engineering activity due to its purpose: to achieve a shared vision between stakeholders of the problems to be addressed. RSLingo is an approach which goal is to improve the quality of requirements specification, through the automation of some verification tasks, allowing the use of natural language, a popular choice for its familiarity, to formulate those requirements. After being formulated, it is a common practice to aggregate them in requirements specification documents, crucial to guarantee an alignment of expectations between all involved parties. Therefore, a collaborative tool to manage system requirements specifications is crucial to manage the interactions and expectations of these parties.

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. 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.

This dissertation introduces ITBox, a collaborative platform for managing technical documentation. Although this platform supports multiple types of technical documentation (such as project plans, requirements or tests specifications) specified according to different languages and templates, this dissertation discusses its features according to requirements engineering projects and project plans perspectives. This dissertation emphasis on ITBox most innovative aspects such as on social, reusability and productivity features. In addition, this dissertation shows the evaluation preliminary results from some user sessions conducted. Also, this dissertation compares and discusses the ITBox with other related tools.

Keywords: Requirements Engineering, Project Plans, Collaborative Platforms, Reusability
Resumo

A especificação de requisitos é uma atividade fundamental de engenharia de requisitos devido ao seu objetivo: alcançar uma visão partilhada entre as partes interessadas sobre os problemas a serem abordados. RSLingo é uma abordagem cujo objetivo é melhorar a qualidade da especificação de requisitos, através da automação de algumas tarefas de verificação, permitindo o uso de linguagem natural, uma escolha popular pela sua familiaridade, para formular esses requisitos. Depois de serem formulados, é uma prática comum agregá-los em documentos de especificação de requisitos, cruciais para garantir o alinhamento das expectativas entre todas as partes envolvidas. Portanto, uma ferramenta colaborativa para gerir os documentos de especificação de requisitos é crucial para gerir as interações e expectativas dessas partes.

O ITBox é uma plataforma web colaborativa que permite a gestão de projetos e que oferece suporte a vários tipos de documentos técnicos (como planos de projetos, especificações de testes e requisitos) especificados de acordo com diferentes idiomas e modelos. A plataforma combina algumas funcionalidades relevantes de outras ferramentas disponíveis, oferecendo também funcionalidades inovadoras de reutilização, como aspetos do modelo de variabilidade no campo da engenharia de requisitos através da linguagem CVL.

Esta dissertação apresenta o ITBox, uma plataforma colaborativa para gestão de documentação técnica de projetos. Embora esta plataforma ofereça suporte a vários tipos de documentação técnica (como planos de projeto, especificações de requisitos ou testes) especificados de acordo com diferentes idiomas e modelos, esta dissertação discute as suas características seguindo uma perspetiva de projetos de engenharia de requisitos e planos de projetos. Esta dissertação enfatiza os aspetos mais inovadores do ITBox, como as funcionalidades sociais, de reutilização e de produtividade. Para além disso, esta dissertação mostra os resultados preliminares da avaliação de algumas sessões de utilizadores realizadas. Por fim, esta dissertação compara e discute o ITBox com outras ferramentas relacionadas.

Palavras-Chave: Engenharia de Requisitos, Planos de Projetos, Plataformas Colaborativas, Reutilização
# Contents

List of Tables ................................................................................................................................. ix

List of Figures ................................................................................................................................. xi

Acronyms ........................................................................................................................................ xiii

1 Introduction .................................................................................................................................. 15
  1.1 Motivation and Context ........................................................................................................ 15
  1.2 Problem Definition and Research Goals ........................................................................... 16
  1.3 Proposed Solution ................................................................................................................ 16
  1.4 Thesis Statement .................................................................................................................. 17
  1.5 Research Methodology ........................................................................................................ 17
    1.5.1 First Iteration .................................................................................................................. 18
    1.5.2 Second Iteration ............................................................................................................. 18
    1.5.3 Publications ................................................................................................................... 19
  1.6 Outline ..................................................................................................................................... 19

2 Background and Related Work .................................................................................................... 20
  2.1 Background – The REBox Platform ..................................................................................... 20
    2.1.1 Overview .......................................................................................................................... 20
    2.1.2 REBox Features ................................................................................................................ 21
    2.1.3 Used Technologies .......................................................................................................... 23
  2.2 Similar Collaborative Platforms .......................................................................................... 24
    2.2.1 Requirements Management tools .................................................................................. 24
    2.2.2 Project Management/Planning tools .............................................................................. 25

3 Research Context ....................................................................................................................... 27
  3.1 RSLingo .................................................................................................................................... 27
    3.1.1 The RSL Language ......................................................................................................... 27
    3.1.2 The RSL Excel Template ............................................................................................... 28
    3.1.3 The RSLingo Studio Tool ............................................................................................... 29
  3.2 ProjectLingo ........................................................................................................................... 30
    3.2.1 The PSL Language ......................................................................................................... 30
    3.2.2 The PSL Excel Template ............................................................................................... 30
    3.2.3 The ProjectLingo Studio Tool ....................................................................................... 31

4 ITBox – Requirements Issues ...................................................................................................... 32
  4.1 Vision and Main Goals .......................................................................................................... 32
  4.2 ITBox Overview ..................................................................................................................... 33
    4.2.1 Domain Model .................................................................................................................. 34
List of Tables

Table 1 - Platform’s allowed actions by the different ITBox roles.......................................................... 40
Table 2 - Projects and documents allowed actions by the different project roles .................................. 41
Table 3 - Survey average score (in a scale of 0-5) by question for the ITBox Web Platform aspect 54
Table 4 - Survey average score (in a scale of 0-5) by question for the ITBox collaborative aspect. 54
Table 5 - Survey average score (in percentage) for each of the questionnaire aspects ......................... 55
Table 6 – Requirements Management Tools comparison with ITBox..................................................... 56
Table 7 - Project Management Tools comparison with ITBox ................................................................. 57
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>The REBox Domain Model [9]</td>
<td>21</td>
</tr>
<tr>
<td>Figure 2</td>
<td>The REBox Web Platform Overview [9]</td>
<td>23</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Overview of the DNN CMS architecture</td>
<td>24</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Example of a Goals Definition sheet of the RSL Excel Template</td>
<td>29</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Example of a Project Definition sheet of the PSL Excel Template</td>
<td>31</td>
</tr>
<tr>
<td>Figure 6</td>
<td>The ITBox Web Platform Overview</td>
<td>34</td>
</tr>
<tr>
<td>Figure 7</td>
<td>The REBox v2 Domain Model (Roles)</td>
<td>34</td>
</tr>
<tr>
<td>Figure 8</td>
<td>The ITBox Domain Model (Documents)</td>
<td>35</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Project State Machine</td>
<td>37</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Document State Machine</td>
<td>38</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Library State Machine</td>
<td>38</td>
</tr>
<tr>
<td>Figure 12</td>
<td>ITBox Home page</td>
<td>39</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Multiple sites on top of the DNN Web App Framework</td>
<td>44</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Home page main features</td>
<td>46</td>
</tr>
<tr>
<td>Figure 15</td>
<td>ITBox's Templates page</td>
<td>47</td>
</tr>
<tr>
<td>Figure 16</td>
<td>ITBox Templates Management page</td>
<td>49</td>
</tr>
<tr>
<td>Figure 17</td>
<td>ITBox Projects Management page</td>
<td>50</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Adding a new document to a project</td>
<td>50</td>
</tr>
<tr>
<td>Figure 19</td>
<td>RSL Excel Template edit page</td>
<td>51</td>
</tr>
<tr>
<td>Figure 20</td>
<td>– Append a Library to a Project Document page</td>
<td>52</td>
</tr>
<tr>
<td>Figure 21</td>
<td>– Management page of project’s participants</td>
<td>52</td>
</tr>
</tbody>
</table>
Acronyms

CMS Content Management System
CVL Common Variability Language
DNN DotNetNuke
DSL Domain Specific Language
RE Requirements Engineering
SRS Software Requirements Specification
RM Requirements Management
PM Project Management
1 Introduction

This chapter introduces some important topics to understand this document. Section 1.1 presents the motivation and context behind the project, introducing the RSLingo approach, for requirements specification and the ProjectLingo, for project plans. Section 1.2 details the problem that this dissertation aims to solve and the research goals it aims to achieve. Section 1.3 introduces the proposed solution, ITBox, specifying its main features and used technologies. Section 1.4 presents the Thesis statement. Section 1.5 presents the followed research methodology to conduct this work. To conclude, section 1.6 presents an outline of the remaining chapters of this dissertation.

1.1 Motivation and Context

Requirements Engineering (RE) is about reaching a shared understanding between two different domains: the business and the technical stakeholders. This bridge can only be established through an effective communication between these two [1]. To build a Requirements Specification Document, RE defines a set of tasks, which demand a high level of collaboration between these two domains, to achieve a mutual understanding of the software system to be developed. The adverse consequences of disregarding the importance of early RE activities are well-known [2]. The intrinsically abstract nature that characterizes the software conceptualization phase can often lead to a misalignment of the views amongst the various stakeholders, which ultimately leads to a system that doesn't match the initial vision [3]. Therefore, it is crucial to adopt proper RE techniques to mitigate all the interpretation problems regarding the business needs that must be addressed.

To prevent such interpretation problems, an effective communication should be achieved, so that everyone can communicate by means of a common language. Natural language is the preferred choice in most projects for allowing elements without knowledge in formal modeling languages to easily express their needs and views of the system [4]. However, although natural language is the most common and preferred form of requirements representation [5], it also exhibits some intrinsic characteristics that often present themselves as the root cause of many requirements defects, such as ambiguity, inconsistency, incompleteness, and incorrectness [1].

The RSLingo approach (section 2.1) seeks to overcome those problems by using simplified Natural Language Processing (NLP) techniques to extract relevant information from informal textual descriptions, such as natural language-written RE specifications, and using it to generate a formal representation in a specific RE language [6].

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 all too frequently due to poor requirements management [7].

Following the RSLingo approach, within the scope of Project Management (PM), Model-Driven Engineering (MDE), and Domain-Specific Languages (DSLs) is ProjectLingo (section 3.2). This project is currently being developed and it focuses the planning phase in Project Management (PM), a crucial phase since the effort spent in planning activities can save countless hours of rework, and costs
in subsequent phases that could lead to project failure [8]. Thus, project plans are fundamental artifacts for the success of any project, which might change several times during the project lifecycle.

Therefore, having a collaborative web platform which allows the management of projects, namely projects containing requirements specification documents and project plans, supported by the mentioned approaches, would represent not only a very important step for the mitigation of the already referred quality problems, but it would also encourage the use of those approaches for future projects.

REBox was a collaborative platform for RE supported by a language designed towards rigorous requirements specifications, RSL (section 2.2). The platform was conceived to provide a familiar but powerful editor, as well as a set of reusability features. It was developed during 2016 by João Fernandes, in the scope of his MSc Thesis [9].

Having the REBox platform available represented a crucial starting point for this dissertation definition. It allowed not only to realize the potential and need of having a web platform supporting the RSLingo approach, but it also triggered the idea of developing a new collaborative platform, more powerful, flexible and adaptable to the future approaches that might appear, including the ProjectLingo and its language, PSL (subsection 3.2.1).

1.2 Problem Definition and Research Goals

As introduced previously, the existence of a collaborative web platform supporting the RSL, PSL and any other future language that might appear, would contribute to the enhancement of the projects’ requirement specification documents and project plans quality, through the usage of the RSLingo and ProjectLingo approaches.

It is a fact that the REBox platform already offered a set of features related to the RSLingo approach. Although, what this dissertation aims to solve is much more: to offer a web platform capable of providing a collaborative environment with reusability and adaptability features, supported by two rigorous languages, RSL and PSL. Therefore, the aspects that this dissertation focused on are described on the following research goals:

RG1: Survey available platforms for managing both software requirements specification documents and project plans.

RG2: To identify concerns that should be addressed when designing a collaborative tool for both RE and Project Plans.

RG3: To design and develop a web platform with some novel features such as usability and usefulness in comparison with solutions available on the market.

RG4: Evaluate the results of the final solution through user session tests and the comparison with other related tools.

1.3 Proposed Solution

This research proposes ITBox, a web collaborative platform for managing projects, namely projects which follow the RSLingo and ProjectLingo approaches (sections 3.1 and 3.2). The platform was developed to offer reusability, variability modeling and cooperative features.

ITBox is a management system for technical documentation, adapted to requirements specifications documents and projects specifications documents. Other kind of documents can also be managed using the platform, but it was especially built to support the Requirements Specification Language (RSL) and Projects Specification Language (PSL). Its reusability features include a Document Template System for uploading/managing templates (which can be used as basis for projects’ documents) and
Libraries for creating and reusing modular sets of reusable specifications. It also offers an innovative variability modeling framework that leverages the concepts of the CVL language [7] to model variability aspects in the context of Requirements Engineering. The framework defines a process for applying the language’s central variability notions - the Variation Points - to the views of a structured requirements specification document, modeled using a rigorous requirements specification language: RSL [10].

This platform has been built using the DotNetNuke® (DNN) .NET web Content Management System (CMS) and employs the Google Drive® cloud storage service. The DNN technology facilitated not only the creation of pages, but also the management of users and their accesses within the platform. The Google technology provides both the Documents, Templates and Libraries editors though Google Sheets® and Google Documents®, as well as the collaborative aspects though its synchronization system and distributed architecture. In addition, ITBox uses a series of Google provided APIs for automating processes and managing data, which helped the development of its reusability features and variability framework.

This collaborative platform allows a decentralized managing and editing of RSLingo and ProjectLingo documents, though editors, that use the RSL and PSL languages, providing a rigorous model for these types of specifications.

1.4 Thesis Statement

This dissertation thesis states that it is possible to develop a collaborative environment for projects based on Requirements Engineering (RE) and Project Management (PM), on a simple but extensible technology, that allows an efficient management of the projects’ documents, their associated members, and a more rigorous definition of the concerns encoded in those documents.

In addition, this can be achieved by using the Google Drive cloud technology that provides accessible and familiar collaborative document editor, allied with the rigorous RSL and PSL structured representations. Furthermore, the set of powerful APIs provided by Google to manage the data encoded in those sheets, allowed the development of several reusability features, including a variability modeling framework that is based in the proposed domain-independent CVL variability language.

1.5 Research Methodology

This work has followed an iterative and gradual way based on the Action Research methodology [11]. This methodology suggests a cyclical process composed of five steps executed in a certain scope, known as client-system infrastructure. The five steps are the following: (1) Diagnosing - identification of the problem domain, (2) Action Planning - planning and definition of the proposed solution, (3) Action Taking - implementation of the solution, (4) Evaluating - evaluation/assessment of the solution developed and (5) Specifying Learning - lessons learned during the cycle and preparation of the next one.

This work had two key iterations, being that each iteration represented a different type of developed work, as it is described below:

---

2. https://www.google.com/drive/
3. https://www.google.com/sheets/about/
4. https://www.google.com/docs/about/
1.5.1 First Iteration

This iteration took the first semester of the year (September 2016 – January 2017) and covered the background investigation and planning for this dissertation main proposal: the ITBox collaborative platform. The Action Research steps that comprised this phase are described below.

**Diagnosing** – This step consisted in the learning of The RSLingo Approach (section 2.1), The RSL Language (section 2.2) and The REBox Platform (section 2.3). Similarly, an investigation of the existing collaborative tools in Requirements Engineering was conducted, resulting in a comparative study which is presented in section 2.5.

**Action Planning** – The planning step consisted in the definition of the ITBox architecture, namely its domain model (section 4.2) and the technologies that would be used. It was also important to define a tentative work scheduling, detailing the dates when each task should be completed.

**Action Taking** – The action taking step consisted on the experimentation of the technologies that would be used. First, the RSL Excel Template served as a basis for an initial practical contact with RSLingo Approach and the RSL language. Then, a simple web platform based on the DNN .NET CMS technology was developed. This platform was not related with the REBox platform, it was just an experimentation with the technologies that would be used. Finally, the REBox platform was extensively tested and the code reviewed, to get the full understanding of how it worked and how it was developed.

**Evaluating** – The evaluation of the work was done while writing, presenting and defending the dissertation project report. This allowed the gathering of important and different opinions and recommendations, not only about the work developed during this first iteration, but also for the work related to the second iteration.

**Specifying Learning** – After the experimentation with the approaches and technologies related to this dissertation, and having the recommendations from the evaluation step, it came clear that the proposed solution would be a valuable and meritorious progress for the research occurring in the RSLingo approach.

1.5.2 Second Iteration

The second iteration took the second semester of the year (February to October 2017) and it mainly corresponded to the implementation, testing and evaluation of the ITBox solution. It was also during this iteration that the dissertation was written. Again, the Action Research steps that comprised this phase are described below.

**Diagnosing** – This step consisted in investigating the best solutions for some of the problems pointed during the dissertation project presentation (e.g. user roles). The periodical review of the developed work is also part of this step, as it required a permanent diagnosis and evaluation of the choices that were being made.

**Action Planning** – The planning step consisted in the making of choices related to the design and operation of the web platform. Since the disposal of the pages and tabs, to the deciding of the data to show and allowed actions in each page, everything was carefully handled during this step.

**Action Taking** – This step consisted in the implementation of the platform itself. Both frontend and backend were developed during this step, and it was essentially to execute what was already planned and decided.
Evaluating – The evaluation of the work that was being developed happened regularly, through the weekly meetings with the supervisor Prof. Alberto Silva. It was possible to get a permanent feedback about the functionality and usability of the platform. Therefore, the recommendations and advices were highly appreciated and useful for the enhanced result. Still, a user evaluation session was performed with multiple pilot-users, allowing them to test and evaluate the developed system.

Specifying Learning – The tests effectuated on the development were positive, helped to identify possible enhancements and left some ideas for future work, like the extension of the platform support for other ITLingo languages (e.g. Tests Specification Language – TSL) of the libraries and variability features.

1.5.3 Publications

A paper titled “A Collaborative Platform for Better Managing Technical Documentation: An analysis from a Requirements Engineering Perspective” was prepared to a conference. This described the ITBox platform, according to a RE perspective.

1.6 Outline

This dissertation is organized in 7 chapters:

Chapter 2 presents the background related with this project’s work.

Chapter 3 presents the context of the work proposed in this dissertation.

Chapter 4 presents the ITBox web platform architecture and its conceptual choices, as well its main features.

Chapter 5 explains the used technologies to develop the ITBox system and presents some of its main pages.

Chapter 6 presents and discusses the evaluation performed on the ITBox system.

Chapter 7 presents the main conclusions of this work along with the future development perspectives.

In addition, there are the following appendixes:

Appendix A describes the user session guide used during the work evaluation.

Appendix B details the user session questionnaire results.
2 Background and Related Work

This chapter presents an overview of the most relevant related work for this dissertation. The main technologies and concepts are presented below, in a set of sections which cover: a background of the related work (section 2.1); a detailed presentation of the REBox platform (section 2.2), which served as basis for this dissertation work; a comparison between different similar collaborative platforms (section 2.3).

2.1 Background – The REBox Platform

To build a Collaborative Platform for RE Projects and Project Plans, it is important to understand what tools are currently being used to create and manage these kinds of projects. One of the tools available for managing Requirements Specification Documents is REBox, which was also developed in the RSLingo context by a former MSc student.

REBox served as a starting point for ITBox, but to develop the best possible solution, it is crucial to do a deep analysis of the REBox architecture choices and to rearrange the aspects which misfit the ITBox platform. Therefore, the following section contains a presentation and analysis of the REBox platform.

The REBox platform is a collaborative tool that was developed with the main goal of providing a toolset for Requirement Engineering projects developed using the RSLingo approach. To this end, there were some concerns when conceptualizing the system architecture and choosing what technologies to use to provide a seamless transition for all the projects currently in development [9].

The following subsections present an overview of the existing REBox platform, including a description of its domain model. The existing features, the way they are developed and the used technologies are also presented and analyzed below.

2.1.1 Overview

Figure 1 shows the REBox platform domain model. There is a brief description of its components below.

REBox is a system which was developed specifically for Requirement Engineering projects. Thus, RSProject is its main class and for each RE project developed in REBox, there is one corresponding RSProject object, which is related with one User that acts as his owner. There is also one associated RS Document and any number of RS Library modules to each RS Project.
The **RS Doc** class contains the information about a software requirements document that is represented in the context of REBox by a Google Sheets spreadsheet. The **RS Library** class aggregates the Requirements Modules represented in the **RS Module**, which contains a set of modular and coarse-grained requirements and can be associated with a **RS Document** by appending its requirements to the ones contained in the document.

Concerning the **User** objects, at creation-time, each one is associated with a set of **Permissions** that will determine which parts of the system are accessible to it. Essentially, these permissions are related with CMS Roles, and they dictate if a user can **view** or **edit** a page of the platform.

The **Requirement** class has 3 types: **ReqGoal**, **ReqFunctional** and **ReqQuality**, each containing the corresponding RSL view’s attributes. These requirements are part of the Requirements Specification Language (RSL).

The **RS Doc Template** class represents the Excel document with pre-filed fields that can be imported by a newly created RS Doc. The existence of this class saves the time of manually retyping the information.

Lastly, the **Variability Model** class represents a Variability Model document created in the platform and aggregates a series of Variation Points Constraints objects. A Variability Model is always liked to one **RS Document**, which represents its Base Model in the CVL notation [9].

These set of classes and their relations are clearly developed for Requirement Engineering projects. Reusability is present through the **RS Doc Template** and **RS Library**, but in terms of collaboration, this architecture has many lacks, as it does not differentiate user roles inside the platform. For example, users can view, edit and delete all the RS Projects, RS Library module and the RS Doc Templates.

### 2.1.2 REBox Features

This subsection describes the three most relevant features of the REBox system. These features allow the integration with the existing technologies related with the RSLingo approach (section 2.1).

The first feature is the **template management system** and it allows the creation and management of reusable requirements templates that can be later used to bootstrap new projects. This system makes full use of the APIs provided by Google. To create a template in the platform, there are two steps to complete: (1) manually upload a spreadsheet file containing the template into the platform, (2) which
is then re-uploaded to the Drive cloud storage server and, finally, shared with the users of the platform. After this, the template can be accessed and edited concurrently like any other document, and it can be used as the base structure for new projects created in REBox. Note that this uploaded document doesn’t need to hold the finalized template (it can even be empty), as REBox allows it to be further edited once its uploaded [9].

This feature is useful, as it allows the saving of time when creating new RS Projects. The use of the APIs provided by Google is a smart choice, as the Drive cloud provides powerful collaborative features, which are a must when accessing and editing documents concurrently. Although, there were some concerns which were not built following the best practices, as there is no distinction, for example, between templates to create new RS Projects and templates to create new RS Lib Modules.

Another restriction related to this approach is the fact that it is only possible to upload new files to the platform through the template management system. This means that if a user wants to upload a new file, for example, to start a new RS Project, he has to upload it to the template management system first.

Another feature is the **RS Library** which offers the possibility to create modules of reusable requirements that can then be added to any RS Project. To be able to reuse requirements across multiple projects is an idea which is being used in some other RE activities. Thus, REBox RS Library serves as an open “database” of modular, agnostic and coarse-grained requirements, which can be grouped by areas of interest. Users can create these groups, add and remove requirements to each module, allowing for an iterative process of refinement. Furthermore, they might be extracted from existing projects or uploaded directly into the platform.

Regarding the **RS Library**, its idea and motivation are valuable and meritorious, as the existence and use of requirements grouped by different areas of interest represents a relevant reusability feature of the platform. Although, and once again, this feature is focused on RE projects and there were aspects which could have been enhanced, like the users’ access to these modules and the way their attributes are manipulated (some of the requirements attributes are not shown, e.g. Type, SubType, Metric, Value, etc.). Therefore, users do not have full control over the requirements present in the RSLib Modules.

Lastly, a very relevant feature, **RSLingo Studio Integration**. This represents a very pertinent point to the requirements quality validation, because, as seen in subsection 2.2.2, the RSLingo Studio tool provides an editor for creating and editing RSL requirement specification documents, like REBox. The big advantage of this integration is that the RSLingo Studio tool has features allied, like syntax highlighting, error checking, auto-completion and source-code navigation, which enhance the correctness of a RSL document that might come from / go to the REBox system.

Figure 2 provides an overview of the REBox platform, its internal features and the way it integrates with the Google Drive Service and the RSLingo Studio tool.
This feature can ensure the correctness and completeness of the documents present in the REBox platform, enhancing the overall quality of them. Although, this solution could be improved in terms of automation and integration, as it requires users to manually upload the documents between the systems.

2.1.3 Used Technologies

REBox was implemented using the DotNetNuke web content management system (CMS) framework, with C# and JavaScript. DNN is an open source technology which allows users to add very easily third-party extensions into the system through an installation process. Thus, REBox was developed as an extension, which contains all the necessary data to be installed in any DNN-running server, facilitating the deployment of the platform.

Figure 3 provides a view of the DNN architecture. Multiple sites can be created on top of a basic web application framework. Each site consists of multiple pages, each containing multiple mini-applications called modules. Concerning REBox, it was developed as a single module which is installed over the DNN Web Application Framework.
Having the platform running, there were then concerns related to the technologies to use to achieve the required features. Google Drive system and its Google Sheets editor were used in the REBox RSDoc editor, providing a familiar and collaborative environment to its users. To enable users to upload excel spreadsheets into the platform, it was built a page using the Telerik ASP.NET File Explorer control. To host this uploaded document, the Drive API is called by the platform, uploading it to Drive.

Some other smaller technologies were also used to achieve the final version of the REBox platform, like jQuery, Json.NET and its plugin JSGrid.

2.2 Similar Collaborative Platforms

The main goal of this dissertation is to develop a collaborative environment for projects based on Requirements Engineering (RE) and Project Management (PM). Therefore, it is crucial to investigate and compare the existing tools which offer similar features. The following subsections contain brief descriptions and comparisons of the most popular tools.

2.2.1 Requirements Management tools

There are many requirements management (RM) tools available on the market, which aim to provide strong RM capabilities, like requirement traceability and project life-cycle integration. Although, the collaborative functionalities are often overlooked. Therefore, this subsection focuses on describing and comparing the different functionalities, from a variety of the most robust RM tools in the collaboration aspect. Some of the used information was extracted from a requirements management tool evaluation report [7].

TopTeam Analyst 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;

---

5 http://www.dnnsoftware.com/platform/start/architecture
6 http://demos.telerik.com/aspnet-ajax/fileexplorer/examples/overview/defaultcs.aspx
7 http://js-grid.com/
8 http://www.technosolutions.com/topteam_requirements_management.html
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.

Modern Requirements\(^9\) 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 TFS (Team Foundation Server) with modules that can be selected and used based on the customer needs. The power tool behind the Modern Requirements suite is InteGREAT Studio\(^10\). It offers a strong and intuitive requirements traceability feature, and it contains full requirements modeling capabilities, allowing the creation of high-fidelity prototypes. The Power Panel includes a collaboration pane where team members can engage in back and forth discussion on a single object, but the overall review and approval capability in the Studio itself is limited.

Visure Requirements\(^11\) 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 attributed to the users.

iRise\(^12\) is a collaborative prototyping tool that allows users to create, review, and update code-free prototypes and mock-ups, while defining requirements within the same tool. iRise is a powerful solution for teams that rely heavily on design-centric elicitation, as you can create requirements directly in iRise while collaborating on visual models or mock-ups. Stakeholders can comment directly on requirements, screens, and individual UI elements. Teams can work on the same screens at the same time, and see each other’s changes and feedback as it is made. Although, iRise lacks some core RM capabilities, it can integrate with almost any major requirements management tool, like JIRA (subsection 2.3.2).

IBM Rational DOORS\(^13\) 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).

IBM Rational RequisitePro\(^14\) 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.

### 2.2.2 Project Management/Planning tools

PM tools have the job, among other things, of helping project managers to develop project plans during project planning. This process requires utmost care and attention since commitments to

---

\(^9\) https://www.modernrequirements.com/
\(^10\) https://www.modernrequirements.com/integreat-studio/
\(^11\) https://visuresolutions.com/requirements-engineering-tool/
\(^12\) https://www.irise.com/
unrealistic time and resource estimates can result in delays, customer dissatisfaction, and, ultimately, project failure. Thus, it is important to understand the existing tools on the market which aim to facilitate the project managers task. This subsection presents an overview and comparison of the most relevant tools.

**Asana** is a web based SaaS (Software-as-a-Service) and mobile application designed to improve team collaboration. Workspaces can be created by each team, and contain projects that contain tasks. Notes, comments, attachments, or tags are some of the artifacts that can be added to these tasks. Users can follow projects and tasks they are interested in and, when the state of a project or task changes, get updates about the changes in their inboxes (not email inbox). Asana is integrated with productivity tools including Dropbox and Google Drive (file storage), Evernote (notes), Instagantt (Gantt chart creator), and others.

**Basecamp** is a web-based project management and collaboration tool. Its core features are: (1) To-dos for tracking work, (2) Message Board for posting announcements and updates, (3) Campfire chat room for quick casual chats with the team, (4) Schedule for posting deadlines and milestones, (5) Docs & Files for organizing all the assets and notes everyone needs to do their work, and (6) Automatic Checkins to get insights from the team on a regular basis. Basecamp is also integrated with many 3rd party tools, including some for Gantt chart creation.

**JIRA** is a project and issue tracking platform. JIRA allows the tracking of any kind of unit of work (be it an issue, bug, story, project, task, etc.) through a predefined (and customizable) workflow. Besides tracking and workflow customization, it also combines agile project management (JIRA Agile), service desk ticket tracking (JIRA Service Desk), integration with other platforms (e.g. Github, Freshdesk, or competitors such as Asana), and even Kandan boards. Tracking is JIRA’s focus, and they are one of the best on delivering it (via detailed custom reports and dashboards) allowing, for example, the user to retrieve the issues assigned to him, created in the last 7 days.

**Microsoft Project** is the most used PC-based project management software, designed to assist the project manager in developing project plans, assigning resources to tasks, tracking project progress, managing the budget, and analyzing workloads. Microsoft Project (or ‘Project’) creates budgets based on assignment work and resource rates. Resources (both human and nonhuman) can be shared between projects using a shared resource pool, and have their own calendar with their availability. Then, the application schedules task work based on this availability. However, Project is unsuitable for solving problems of limited materials and cannot determine how many products can be produced with a given amount of raw materials.

**Wrike** is an online tool and application for project management and work collaboration. Users can manage and track projects, deadlines, schedules, and other workflow processes. Private collaboration between users is also allowed. Wrike’s primary goal is to help streamline workflow and allow companies (of varying sizes) to focus on core tasks. It has an activity stream feature that updates users on any tasks performed by other users of their work groups. Wrike is integrated with multiple other software products including Microsoft Excel, Microsoft Project, Microsoft Outlook, Dropbox and Google Drive, and others.

---

15 https://asana.com  
16 https://basecamp.com/  
17 https://www.atlassian.com/software/jira  
19 https://www.wrike.com
3 Research Context

This chapter provides an overview of the context behind this dissertation. Section 3.1 introduces the RSLingo approach, which defines a series of processes and languages towards the development of more formal requirements specifications. The work developed in this initiative over the last years led to the origin of the RSL language, the RSL Excel template and the RSLingo Studio Tool, which are all presented below.

Like the RSLingo approach, but with the goal of improving the project plans’ quality by mitigating ambiguity, completeness, and consistency problems, there is the ProjectLingo, which is currently being developed, along with the PSL language, the PSL Excel template and the ProjectLingo Studio Tool (section 3.2).

3.1 RSLingo

The RSLingo approach supports the use of natural language for writing requirements specifications. It is often the preferred choice in most projects for allowing users without knowledge in formal modeling languages to easily express their needs and views of the system [4]. However, natural language possesses intrinsic characteristics that often lead to incorrect, inconsistent, incomplete and ambiguous requirements [12]. To prevent this, RSLingo proposes an Information Extraction approach based on linguistic patterns that are frequently used to express RE-specific concerns to improve the quality of requirements specifications written in ad-hoc natural language [6].

Despite being the preferred language to write requirements specifications, it still requires a considerable human effort to produce this type of documents. Thus, the automation of such a task could be used for the following purposes: (1) domain analysis, such as identifying relevant concepts and relations; (2) verification and consistency checking and (3) producing transformations, like the automatic generation of representations of diagrams or reports [6].

The full processing of natural language text is still too complex to be automatically performed by computers in a general-purpose context. However, despite its complexity, there are some simplifications that can be applied to Natural Language Processing (NLP) techniques [13] when the goal is to extract relevant information from the text.

3.1.1 The RSL Language

RSLingo's RSL (Requirements Specification Language) is a rigorous language with a fixed set of constructs for representing and conveying RE-specific concerns. RSL is designed to address RE-specific concerns to better support RE-related tasks, according to processes and techniques [6]. The language defines several constructs that are logically arranged into viewpoints according to the
specific RE concerns they address. For example, the people and organizations that can influence or will be affected by the system are represented in the Stakeholders viewpoint. Likewise, the objectives of business stakeholders regarding the value that the system will bring are expressed in the Goals viewpoint.

This grammar is defined based on the RSL Excel Template (subsection 3.1.2), using Xtext and Xtend for code generation and validation processes. The following subsections detail two tools which were developed for structuring the data according to the RSLingo approach (The RSL Excel Template), and for facilitating and improving the use of the RSL language (The RSLingo Studio Tool).

3.1.2 The RSL Excel Template

In the field of Requirements Engineering, a Software Requirements Specification (SRS) is a document that describes multiple 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 (that should be customized to the needs of the organization involved) prescribing the use of multiple modular artifacts corresponding to different views. The RSL Excel SRS Template is based on the multi-view architecture defined in the RSL language and is currently being used by other RSLingo-related research projects to document and model system requirements and other RE concerns in a set of different views using the RSL language [9].

The RSL language defines several constructs that are logically arranged into viewpoints according to the specific RE concerns they address. For example, the people and organizations that can influence or will be affected by the system are represented in the Stakeholders viewpoint. Likewise, the objectives of business stakeholders regarding the value that the system will bring are expressed in the Goals viewpoint. This multi-view architecture defined in the RSL language has since been adapted to an Excel format: the “RSL Excel Template” [14]. The template is illustrated in Figure 4.

Regarding the REBox platform, even though the whole template can be uploaded and used to model requirement specifications documents, only the main requirements views - Goals, Functional Requirements and Quality Requirements - are supported [9]. One of this dissertation goals is precisely, to extend the supported views, both in the creation of variability models and libraries.

---

The RSLingo Studio Tool

The RSLingo Studio is a software tool that is currently being developed as an adaptation of the homonymous RSLingo4Privacy Studio. RSLingo4Privacy\textsuperscript{22} is a multi-language approach that intends to improve the specification and analysis of privacy policies, supported by several processes and respective tools. This approach is backed by its own requirements specification language - the RSL-IL4Privacy (Rodrigues et al., 2016) that, despite sharing the same background and technologies as RSL, was recently defined independently and with the only purpose of supporting the rigorous specification of privacy policies with multi-representations.

Due to the nature of this dissertation, and to be as explicit as possible, only the RSLingo Studio tool will be referred, assuming that it already implements all the same features as its counterpart.

The most relevant feature that RSLingo Studio provides is a textual editor that allows creating and editing RSL files. This editor was developed using the Xtext framework\textsuperscript{23}, which allows the development of textual domain specific languages (DSLs) through their definition using Xtext’s grammar. From that grammar definition, it is possible to automatically generate the language infrastructure (e.g. parser and type checker) and a fully customizable Eclipse\textsuperscript{24} plugin containing the DSL editor with helpful features like syntax highlighting, error checking, auto-completion or source-code navigation. Furthermore, the RSLingo Studio tool supports multiple representations of a requirements document. It relies on several transformations to create a mapping between these representations by using RSL as a common language. This tool currently supports the following document types: ad-hoc and controlled natural language text, Excel, Word and JSON.

The REBox platform allows a direct integration with the RSL files and consequently with the RSLingo Studio Tool, through the import/export capabilities using the RSL Excel Template.

\textsuperscript{21} https://github.com/RSLingo/RSL-Excel-Template/blob/master/RSLingo-RSL-BillingSystem-v3.3.xlsx
\textsuperscript{22} https://github.com/RSLingo/RSLingo4Privacy
\textsuperscript{23} https://eclipse.org/Xtext/index.html
\textsuperscript{24} https://eclipse.org
3.2 ProjectLingo

As mentioned before, ProjectLingo is currently being developed and aims to improve the project plans’ quality.

The focus of this project is in the first two phases of project management: Initiating and Planning, but mainly on the latter. This phase is very important in PM, since the effort spent in planning activities can save countless hours of rework, and costs in subsequent phases that could lead to project failure. In fact, although planning does not guarantee project success, lack of planning will probably guarantee failure [15]. Most importantly, it is during this phase that the project plan, particularly relevant for this dissertation, is produced.

Project Plan, according to PMBOK [16], can be defined as a “Formal, approved document used to guide both project execution and project control. The primary uses of the project plan are to document planning assumptions and decisions, facilitate communication among project stakeholders, and document approved scope, cost, and schedule baselines”.

This artifact will eventually change several times during the execution of project iterations, being the improvement of its quality with the purpose of mitigating consistency, completeness, and ambiguity problems, the main goal of the project described in this document. By guaranteeing this, it is most likely that less changes will be needed to execute on the initial project plan and that budget and schedule do not suffer from deviations [8].

The fact of integrating ProjectLingo with ITBox makes perfect sense, because it would not only represent the availability of a collaborative platform for Project Plans, but it would also extend the domain of projects where the ITBox can be used (not just for RE projects). Another valid reason for this integration is the proximity between the RSLingo approach and ProjectLingo. In fact, they are both being developed under the supervision of Prof. Alberto Silva.

3.2.1 The PSL Language

The Project Specification Language is a rigorous and intermediate textual DSL (Domain Specific Language), for specification of multiple project related aspects, such as stakeholders, scope, costs, time, quality, and risks definition.

It is based on the PSL Excel Template (subsection 3.2.2) and its implementation is performed on top of the ProjectLingo Studio, using the eclipse integrated framework, Xtext, and Xtend programming language (as it happens with RSL).

3.2.2 The PSL Excel Template

The PSL Excel Template is like the RSL Excel Template, but applies to projects’ planning, providing a familiar representation of the projects’ details. It makes use of the multi-view architecture and is inspired on PM standards and frameworks such as ISO 21500, PMI PMBoK, IAPP, IPMA. This template allows the project manager to specify the plan using the RSL Language, through a simple and familiar editor (Excel). Finally, this project plan can be validated with the ProjectLingo Studio (subsection 3.2.3).

Figure 5 illustrates one of the template’s views.
3.2.3 The ProjectLingo Studio Tool

ProjectLingo Studio is an IDE based on Eclipse with features for DSL development and interoperability with other tools. ProjectLingo Studio allows the specification of project plans defined in PSL and provides import and export features, interoperability mechanisms, for Excel, Word, and Project. This also allows project managers to specify the project plan using Excel or Project, and validate it using ProjectLingo Studio.

It is currently being developed and the main idea is to have a tool with a similar behavior as the RSLingo Studio Tool, but applied to the ProjectLingo approach.

---

25 https://github.com/ProjectLingo/Project-Excel_template/blob/master/ProjectLingo-Project-Excel-Template-v1.0.xlsx
4 ITBox – Requirements Issues

This chapter aims to describe the main proposal of this dissertation: the ITBox collaborative platform for managing technical documentation, namely documents belonging to RE and PM projects, supported by the two rigorous languages presented previously, RSL and PSL. Section 4.1 presents the ITBox’s vision, including its main goals based on this thesis statement. Section 4.2 overviews the platform, presenting its domain model and the associated state machines. Section 4.3 describes in detail the platform’s main features. Lastly, section 4.4 extensively details the followed approach concerning the ITBox’s user roles and the corresponding allowed actions.

During this chapter, some references and comparisons with the REBox platform will be made. This will allow a better understanding of some of the work developed during this dissertation.

4.1 Vision and Main Goals

The ITBox platform aims to provide a collaborative environment for managing technical documents and, in particular, documents of projects based on Requirements Engineering and Project Management, which follow the RS Lingo (section 3.1) and ProjectLingo (section 3.2) approaches. This platform intends to allow an efficient management of these projects and their artifacts, contributing for the increase of these approaches’ languages adoption.

Thus, ITBox was developed considering the existing collaborative solutions available on the market, which, despite of not being developed focused on the RS Lingo and ProjectLingo approaches, offer a set of useful features that were adopted and adapted for this system.

Therefore, the ITBox system’s main conceptual goals can be summarized as follows:

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 templates. For example, adopting the RS Lingo approach, as presented previously, it should be possible to use not only RSL Excel templates, but also the RSL language directly. Because 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 created, each one having different allowed actions, both at the platform-level and at the project-level and their 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 APIs, ITBox can ensure the synchronization of documents and the timely sharing of information among users.
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 has three different mechanisms which contribute for the automation, and consequent time-saving, of some user actions.

The first one is a Templates system, which allow users to create new documents from templates, instead of having to create them from the ground. The next feature is the existence of a set of Libraries, which contain sets of reusable specifications (both requirements and projects specifications), that can be used between projects’ documents as long as they are of the same type. These platform’s templates and libraries are managed by a specific user role (see section 4.3), which must assure the documents’ quality. As ITBox supports documents of different types of languages, its Template and Libraries systems needed to match these multiple types. Thus, the reusability mechanisms were adapted and expanded, relatively to the ones existing in REBox, to support all the platform’s existing document types.

Finally, a variability modeling approach based on CVL (explained in section 4.4) also represents a relevant feature. This mechanism already existed in REBox and, once again was reformulated and adapted to the ITBox’s needs, being that one of the improves made corresponds to the extension of the RSL Excel Model supported views, namely “Constraint Requirements”, allowing a wider scope of variability points within the spectrum of requirements engineering concerns. Concerning this mechanism, the platform is ready to support variability points from the PSL Excel template, but this feature was not concluded due to the current development status of this template.

**Extensibility to support multiple languages and approaches:** RSLingo and ProjectLingo propose two different rigorous languages to use in software requirements specification documents and in project plans, RSL and PSL, respectively. As mentioned before, these two approaches are being developed under the supervisor of Prof. Alberto Silva. In the future, similar approaches might appear, but applied to different types of projects. Therefore, a simple way for adding the support of new languages to the ITBox platform, represents a relevant feature. To achieve this, ITBox has one management page for the supported types of documents (and languages), accessible to the platform’s administrator. Although, through this feature, users will only be able to upload and edit documents of the new types. To use mechanisms like the libraries or the variability models, manual coding will be required.

**4.2 ITBox Overview**

After gathering the lessons learned from analyzing the related work regarding collaborative platforms, having the REBox platform as the basis of the system, understanding the reasons why it was developed in such a way, and considering the main goals that defined the conception of ITBox, it came clear how its internal features and the way they integrate with the outside tools, like Google Drive Services, RSLingo Studio and ProjectLingo Studio tools, would have to become.

Figure 6 provides an overview of the ITBox platform, its internal features and the way it integrates with the external tools. The next sections describe in detail each of those components. The following subsections present the system’s domain model and the state machines associated to Projects, Documents and Libraries.
4.2.1 Domain Model

To clarify the ITBox concepts, two perspectives of the platform’s domain model will be presented. Figure 7 presents a domain model, from a collaborative perspective, focusing on the existing User Roles, and how they are related with Projects. For a clear understanding of the whole dissertation, the comprehension of the following figures is relevant; therefore, they are followed by a brief clarification about their concepts.

Figure 6 - The ITBox Web Platform Overview

Figure 7 - The REBox v2 Domain Model (Roles)
**User Account** is the most fundamental component of the system alongside the **Project**. Each **User Account** contains the information of the user of the platform. This component is associated with three different roles (**DNN Role**, **ITBox Role** and **Project Role**).

**DNN Roles** are related with CMS Roles, granting customizable access restrictions within the platform. They fall into two main categories: *view* and *edit*, and can be customized in a page by page basis.

**User Accounts** associated to the **DNN Admin Role** can add, edit and delete the existing pages, other **User Accounts** and their associated **DNN Roles**.

**DNN Registered Users** can view the pages they are have access to, depending on their corresponding **ITBox Role**.

The **DNN Guest Role** is assigned to current or future users of the ITBox platform, and, therefore, of the **DNN CMS**. Once they reach the Home page, they can sign up, sign in or recover the password of an existing account. After signing in, users are no longer **DNN Guests**.

**ITBox Role** is the role that each **User Account** plays in the ITBox platform. These roles are not related with the **DNN Roles** or **Project Roles**. Further information about the **ITBox Roles** is presented in subsection 4.3.1.

**Project Role** is the role that each **User Account** plays in each **Project**. Once again, the **Project Roles** are not related with the **DNN** or **ITBox Roles**. Further information about the **Project Roles** is presented in subsection 4.3.2.

**Project** is the main class of the system. Each new project developed in ITBox has one corresponding **Project object**. Each **Project** can contain several **Documents** and can be linked with multiple **User Accounts**.

The **Document** class is better described according to the following Domain Model perspective.

Figure 8 presents the ITBox domain model, from a Documents perspective, showing the types of document supported by this platform.

![Figure 8 - The ITBox Domain Model (Documents)](image-url)
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). 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 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. When adding a new DocType to the platform, users must specify the name, description, file extension and the desired editor for editing documents of that DocType.

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.

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

4.2.2 State Machines

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:

Project State

ITBox project objects have a “ProjState” attribute. As mentioned in subsection 4.3.2, only the Project Owners can modify this attribute.

After a project is created, the “Not Started” state is attributed to it. 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 projects.

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 9 illustrates the transitions described above, through a state machine.

---

**Document State**

ITBox project documents objects have a “DocState” attribute. As mentioned in subsection 4.3.2, Project Owners and Project Members can modify this 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, transitng 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, transitng to the “Closed” state.

Finally, when a document is in the “Closed” state, it can no longer be edited, it can only be viewed. This is a terminal state, which means that a document cannot transit to a different state after reaching this one.

Figure 10 illustrates the transitions described above, through a state machine.
Library State

ITBox Library objects have a “LibState” attribute. Only users allowed to manage Libraries (see subsection 4.3.1) can modify this attribute.

Libraries have only two different possible states, “Available” and “Unavailable”. When a Library is created, it has the “Available” state. This means that the Library will be available to be used by the ITBox’s users to start new project documents, to append it to project documents or to create new variability models. From this state, Libraries can only transit to “Unavailable”.

The “Unavailable” state does not represent a terminal state, which means that a Library can transit to the “Available” state again. The “Unavailable” state hides the Library to the platform’s users, not allowing the execution of the actions described before.

Figure 11 illustrates the transitions described above, through a state machine.
4.3 User Roles and Use Cases

After analyzing the similar collaborative platforms in section 2.3, it came clear that to provide a collaborative environment, a set of features would have to be offered, namely (1) user-defined views, (2) different user roles and (3) user management features. The DotNetNuke CMS technology facilitated the process to achieve the mentioned features, as detailed in subsection 5.4.1.

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 [25] managed at two levels: at the platform-level and at the project-level.

4.3.1 Platform-Level Roles

The “ITBox Roles” were defined at the platform-level. ITBox has multiple pages, which provide access to different contents and actions. 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.

Figure 12 shows the existing ITBox pages, which are described in detail in section 5.3.

![ITBox Home page](image)

Table 1 contains a representation of the ITBox roles and the corresponding actions that they can perform.
As the Table 1 represents, ITBox has four different user roles:

**ITBox Admin** role grants access to all the ITBox pages related with the management of the existing users, roles and DocTypes (see section 5.1).

**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 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.

### 4.3.2 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. Thus, Table 2 presents the allowed actions both for projects and their associated documents, for each Project Role.

<table>
<thead>
<tr>
<th>Platform-level Role Action</th>
<th>ITBox Admin</th>
<th>ITBox Manager</th>
<th>ITBox User</th>
<th>ITBox Guest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Home page</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Sign up</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Sign in</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Recover password</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Edit profile</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Access Projects</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Manage Users</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage ITBox Roles</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage DocTypes</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage Libraries</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Manage Templates</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITBox Platform</th>
<th>Access Home page</th>
<th>Sign up</th>
<th>Sign in</th>
<th>Recover password</th>
<th>Edit profile</th>
<th>Access Projects</th>
<th>Manage Users</th>
<th>Manage ITBox Roles</th>
<th>Manage DocTypes</th>
<th>Manage Libraries</th>
<th>Manage Templates</th>
<th>ITBox Admin</th>
<th>ITBox Manager</th>
<th>ITBox User</th>
<th>ITBox Guest</th>
</tr>
</thead>
</table>
Table 2 - Projects and documents allowed actions by the different project roles

<table>
<thead>
<tr>
<th>Project-level Role Action</th>
<th>Project Owner</th>
<th>Project Member</th>
<th>Project Consultant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Project Members</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Modify Project Properties</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create a new Project</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Change Project State</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>View Documents</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Edit Documents</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Append Library to Document</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Modify Document’s Properties</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Document’s State</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Delete Document</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

As the Table 2 represents, there are three different Project Roles:

**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.

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.

### 4.4 Features

This section presents the main features offered by ITBox. Subsection 4.3.1 describes a set of systems which allow the management of all the platform’s documents. The following subsection explains the system which responsible for leading with the ITBox’s users. Subsection 4.3.2 explains the existing integration with the RSLingo and ProjectLingo Studio Tools. Finally, subsection 4.3.4 details a reusability mechanism, the variability modeling framework.
4.4.1 Users Management

This system allows an easy management of the ITBox platform’s users and their roles (see section 4.3). As the roles that users have in projects determine the permissions they have in the projects’ documents, a relation between ITBox and Google Drive user accounts must exist. Therefore, the user management system also grants an alignment between these user accounts. To conclude, this system is also responsible for the management of the platform’s users and their roles, which are detailed in section 4.3.

4.4.2 Documents Management

The document management system guarantees that all the documents present in the ITBox platform are permanently synchronized between all the platform’s accesses. To achieve this, ITBox makes full use of the APIs provided by Google. The document management system employees Google Sheets and Google Docs to provide two types of editors, depending on the document formats (Excel or RSL/PSL). It can be simplified in three subsystems, which are described below.

**Template Management System:** The template management system offers the possibility to create and manage reusable templates that can be later utilized to bootstrap new projects’ documents. The process of creating a template in the platform encompasses two steps. The first one consists on the manual upload of a file containing the template into the platform. To note that 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 the users allowed to access to it (section 4.3). 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 project’s document created in ITBox.

**Library Management System:** This system 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 presented in the Related Work regarding the efficiency increase of RE activities [9], and REBox already offered this mechanism. Therefore, this system was extended and adapted to the ITBox domain, being now possible to create libraries of reusable project specifications. To achieve this, each ITBox library serves as an open “database” of modular, agnostic and coarse-grained specifications, that are classified depending on the library type (RE or PM). These groups of specifications must be created manually, by a certain kind of users (section 4.3), which ensure the quality of the specifications. Furthermore, after created, this kind of users can still add and remove specifications to the module, allowing an iterative process of refinement. The process of adding a specification library to a project document is very simple, since the new specifications only need to be “appended” to the existing ones.

**Document Upload System:** As mentioned before, a certain kind of users (see section 4.3) can add documents to projects using existing templates in the ITBox platform. Beyond that, the document upload system allows these users to manually upload the documents they want to the projects they belong to. By doing this, the procedure of uploading the file to ITBox is identical to the one adopted in the template system, but this time a template will not be created into the platform.

4.4.3 Variability Modeling Framework

The reusability based on variability features is aligned with the Common Variability Language (CVL) [17,18], 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 experimentaton and the proposal of new approaches in the field.

This feature already existed in REBox and it did not suffer new changes. It was only adapted to the
new platform.

As suggested in the Figure 8 (ITBox Domain Model, Documents) it is possible to still increase the
extensibility of ITBox libraries with variability models. 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.

4.4.4 RSLingo/ProjectLingo Studios Integration

As described in subsections 3.1.3 and 3.2.3, the RSLingo and the ProjectLingo Studio tools are
systems currently being developed in the scope of the RSLingo and ProjectLingo initiatives. Like
ITBox, they also provide an editor for creating and editing RSL and PSL specification documents.
However, while ITBox focuses on a more user-friendly representation, Studios focus on the textual
representation of the languages. By defining DSLs containing the languages’ grammars, they make
use of the Xtext framework to generate a parser and type checker for RSL and PSL, respectively.
These tools, allied to features like syntax highlighting, error checking, auto-completion and source-
code navigation, provide very powerful means for automatically analyzing and validating the
correctness of RSL and PSL documents.

Therefore, the integration with the RSLingo and ProjectLingo Studio tools would provide a very
big improvement in terms of specifications quality validation. From the start of the project that the
integration with other RSLingo and ProjectLingo technologies was defined as one of the main focuses,
and indeed, the choice to use the Google Drive technology allows the possibility not only to upload,
but also to download the spreadsheet files. Those files can then be manually uploaded to the RSLingo
and ProjectLingo Studio tools and converted into the textual RSL and PSL representations [9]. The
user can then receive visual feedback about the correctness of the documents.
5 ITBox – Implementation Issues

This chapter provides an overview of the ITBox’s web platform implementation. Section 5.1 presents the solution architecture. The technologies that were used to develop the platform are presented in section 5.2. Lastly, section 5.3 describes the ITBox’s main pages.

5.1 System Architecture

In terms of the platform’s architecture, there are no significant changes comparatively to the REBox’s system. Basically, the ITBox is a system installed over the DNN Web Application Framework, as a Third-party extension.

As it is represented in Figure 13, the DNN Platform architecture allows the creation of multiple sites on top of a basic web application framework. Each site consists of multiple pages, each containing multiple mini-applications called modules. Like the REBox platform, ITBox is also developed as a single module. This way, the web application framework provides a solid foundation for all the ITBox system needs and is open for integration with external applications via a service framework.

![Figure 13 - Multiple sites on top of the DNN Web App Framework](http://www.dnnsoftware.com/platform/start/architecture)

---

26 http://www.dnnsoftware.com/platform/start/architecture
5.2 Technologies

As mentioned in chapter 2, the ITBox platform is a system that used as basis an already existing platform, REBox. Therefore, the technologies chosen to implement ITBox were DotNetNuke Web CMS framework, using C# and Javascript. The decision was based in two factors: The first was to take advantage of the existing REBox system, which used the same technologies. The second reason was based on the features offered by the DNN, which matched the ITBox needs.

From all the ITBox technological requirements, we highlight the following, offered by DotNetNuke CMS:

Create and manage User Accounts: User Accounts, located on the Admin > User Accounts page, enables the creation and management of registered user accounts, as well as assignment of security roles. The fields displayed on the module can be set, as well as the way user accounts are handled. New profile properties can be created. User Accounts can be deployed to any page by an authorized however because it forms part of the Users & Roles module package it is set as a premium module by default to reduce the instance of it being accidentally added to a page and revealing personal user information it will not be displayed in the add module area by default.27

Role Based Access: Access to view and manage the site content and settings is controlled using role based access. Associating a user account with one or more security roles (also called roles) allows Administrators to control which users can access which pages and modules28.

Manage Roles: The Admin > Security Roles page allows authorized users to create and manage security roles and security role groups and assign or remove users from those roles. Security Roles which forms part of the Users & Roles module package and can be added to any site page. Users must be granted Edit Module (DNN Platform) or Edit Content permissions to perform tasks.29

Restrict access to view and manage pages: Administrators and Page Editors (users who have been granted Edit Page permissions for the page) can set access to view, edit and manage pages and page content.

During the development phase, multiple tools were used. Two of those were fundamental for smooth process progress: Microsoft Visual Studio and WebMatrix. Follows a summary of these tools.

Microsoft Visual Studio30 is an integrated development environment (IDE) from Microsoft. Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. During this dissertation, this tool was used to develop C# and ASP.NET files.

---

30 [https://www.visualstudio.com/](https://www.visualstudio.com/)
WebMatrix\textsuperscript{31} was released to support the large number of open source content management systems and to provide a lightweight web development environment for PHP and the new, simplified ASP.NET web pages. It focused on a clean, simple user interface allowing web developers to build websites from scratch or by customizing open-source web content management systems, like DotNetNuke. This tool allowed an easy deployment of the ITBox Web Platform.

5.3 Main Pages

The ITBox main pages, also called as “Tabs”, are the pages which are always available to the users, as long as they have permissions to access them. They are displayed on a horizontal bar, which is always visible, independently on the page. As explained on subsection 4.3.1, the access to these pages is dependent on the user’s ITBox role. Below, there is a detailed explanation of these pages and what the actions they allow users to perform. All pages are designed similarly between each other, granting consistency along the platform.

**Home page** is the initial web page of the ITBox platform. This is the first page that all the visitors will see and serves as a landing page too. This page provides basic information about ITBox, including a brief description of its main features. From this page, guests can register on the platform or login with their accounts, if they already have them. Figure 14 ITBox main features which are contained in the home page.

![Figure 14 - Home page main features](https://www.microsoft.com/web/webmatrix/)

**Projects page** is the page where users can view and manage the projects they have access to. It contains a list of the existing projects in which the user is involved in. It allows the creation of new projects and to filter the existing projects by date, state, and name. Still on this page, users can sort the projects by name or creation date. They can also access each project’s documents, permissions and properties (e.g. name, description, project owner, etc.). Finally, project owners can delete their projects from this page too.

\textsuperscript{31} [https://www.microsoft.com/web/webmatrix/]
Templates page grants users who have access to this page (section 4.3) can view and manage the existing ITBox templates. They are organized in a table, which shows the name, type and submission date of each template. Users can filter the templates by date, name and type. Again, the templates can be sorted by name, type or submission date. From this page, users can create new templates, and access or delete the existing ones.

Libraries page lists the existing ITBox Libraries. Their name, type, uploader, submission date and state are displayed. Once again, users can filter and sort the libraries. From this page, users can create new libraries. For each library, it is possible to access it, create a variability model (subsection 4.2.4) based on the library, edit the library properties or delete it. Finally, it is possible to access the list of the existing variability models, from this page. Figure 15 illustrates the representation of this page.

Settings page is composed by three subpages, which allow users to manage the existing ITBox Roles (subsection 4.3.1), the existing users, and the existing ITBox DocTypes. These pages allow an easy management of the platform settings, through an intuitive interface, avoiding the need of coding.
6 Evaluation

During this research, a case study was defined to test and evaluate the suitability of the ITBox platform, based on a running example publicly available in the context of the RSLingo initiative: The Billing System case study. A brief introduction and description of this case study follows:

The Billing System. This system was originally defined for showing how to specify a simple business information system based on the RSL language. It provides a complete set of features for managing customers, products and invoices and can be configured to support the work of different business managers (operators, product operators, managers, and sys-admins) to allow them to collaboratively manage and participate in the processes related the creation, approval, issue of invoices, control their respective payments, and produce several reports. It also implements a simple workflow associated to the invoice’s lifecycle, namely considering the following states: created, waiting-for-approval, approved, issued, paid, and deleted.

The Billing System was also used during the user session, which assessment is presented in section 6.1. It contains the results of a test session of ITBox by a set of pilot users. This assessment focused on testing the overall usability and quality of the implemented features. Section 6.2 summarizes the ITBox evaluation, comparing it to the previously presented similar platforms. Lastly, section 6.3 demonstrates a working example, which covers the main steps of the user guide used during the testing session.

6.1 Working Example

This section will loosely follow the major steps defined in the User Session Guide (Appendix A) to provide a view of the platform’s implementation. To note that this section is not intended to provide a detailed explanation of the functionalities (which was done is chapter 4), but rather give a view of the design and implementation choices made when encoding the platform.

For this example, a project called “ITBox Working Example Project” was created. The main tasks described below include (1) upload a RSL Excel Template file for the ITBox platform, (2) create and manage a project, (3) add the previously uploaded file to the newly created project, (4) append a quality requirements library to the project’s document, and (5) invite a new user (with the Project Consultant role) to be part of the “ITBox Working Example Project”.

48
Template Upload

The process starts with the manual upload of a file into the platform. After the file is uploaded to the local filesystem, it must be re-uploaded to Google Drive, so it can be edited.

When uploading a new template to the platform, the user must specify the type of document he wants to add. This will determine which editor will be used to access or edit the template. Furthermore, when a user tries to add a new document to a project, and wishes to import that document from the templates list, it will only be possible to import templates which match the type of the document the user is trying to add to the corresponding project.

Figure 16 shows an example of the Templates page.

![ITBox Templates Management page](image)

**Figure 16 - ITBox Templates Management page**

This page displays the existing templates names (which correspond to the uploaded filenames), the types of the templates, and the submission dates. Furthermore, it is possible to access the existing templates, edit and delete them.

Project Creation and Management

The project creation is a simple step. When creating a new project, a user must specify the desired name and description. After saving the project, it will have the “Not Started” state, which can be modified anytime by the Project Owner.

Figure 17 illustrates the ITBox Projects page.
This page design is like the Templates one, containing a table which displays the existing projects, and their corresponding creation dates and status. From this page, users can access each project’s documents, manage its participants, edit its properties (name, description and state) and delete them, if they are authorized to do so.

**Manage Project Documents**

After having a project created, the project owner and project members can add multiple documents to it. When adding a new document, users must fill a set of fields, including the document type and subtype (Simple Document or Variability Derived Document). This example will focus on a RSL Excel simple document. The difference here is the following: Simple Documents are created based on a Templates or Libraries described above, or based on files uploaded at that moment, while Variability Derived Documents are created based on a previously defined Variability Model plus a set of resolution values. This step will only focus on the “Simple Document”.

Figure 18 illustrates the process of adding a new document to a project.

![Figure 18 - Adding a new document to a project](image-url)
When the user hits the “Save” button, the template is re-uploaded to Google Drive, creating a different reference from the one previously created for the template. The document will then have the “On Edit” state.

Finally, after having added the document to the project, the user can see the existing project’s documents, namely their names, types, creation dates and state. Furthermore, users can access and edit them, append libraries, edit their properties (name, state) and delete them. When a user accesses a document, he is forwarded to the editing page for the corresponding type of document. In this example, the editor consists in the Google Sheets Editor\(^{32}\) embedded into a ITBox page, as the Figure 19 demonstrates.

![Figure 19 - RSL Excel Template edit page](https://www.google.com/sheets/about/)

**Append a Library to a Project Document**

Through the previous screens, project owners or project members can append libraries to project documents. When a user starts that process, he is forwarded to a new page, which is displayed in Figure 20.

\(^{32}\) [https://www.google.com/sheets/about/](https://www.google.com/sheets/about/)
Append a Library to a Project Document

Depending on the document type, the available libraries are filtered. This ensures that the users can only append libraries which have the same type as the documents.

In this scenario, the user is adding a quality requirements library to the “BillingSystem” document (RSL Excel type). After selecting the desired library to append, the library’s requirements are displayed, and can then be edited or deleted, independently. To conclude this process, the user only needs to click on the “Append Library” button.

Manage Project Participants

Project owners and project members can add and remove users from projects. To add a new user to a project, it is necessary to specify its email address and the desired project role (member or consultant). If that email is associated to an existing account, the corresponding user instantly becomes a project participant. Although, if the typed email is not associated with an existing ITBox account, an email is sent for the user inbox, inviting him to register at ITBox and join the project.

Figure 21 illustrates the page which allows the management of project participants.
Similarly to the previously displayed pages, this one also contains a table, with the projects participants and their corresponding project roles. This table shows the participants’ names, usernames, emails and project roles (which can be modified), and allows the project owner to remove participants.

6.2 User Session Assessment

To better evaluate the ITBox platform, and to receive feedback from people not directly involved in this research work, 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 (Appendix A) describing the steps they should follow. As mentioned before, 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 analysis of the results gathered from these questionnaires is described below.

Questionnaire Analysis

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 Web Platform aspect

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average</strong></td>
<td>4.5</td>
<td>4.4</td>
<td>4.6</td>
<td>4.5</td>
</tr>
</tbody>
</table>

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 to 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

<table>
<thead>
<tr>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average</strong></td>
<td>4.5</td>
<td>4.3</td>
<td>4.6</td>
<td>4.6</td>
</tr>
</tbody>
</table>

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.

54
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

<table>
<thead>
<tr>
<th></th>
<th>Usability and Features</th>
<th>Collaborative Environment</th>
<th>Platform’s usefulness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>90%</td>
<td>90%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Regarding the number of participants, it can be stated that the sample was adequate to extract meaningful conclusions from the results. Studies have noted that a group of 10 participants will find over 90% of the usability problems [24] and so the user session represented a considerable example of what could be expected from a bigger number of participants. The complete results of the questionnaires can be consulted in Appendix B.

6.3 Related Work

This section provides a comparative analysis of the ITBox’s features to other related proposals. The following subsections contain comparison tables to requirements management tools and project management tools, 6.3.1 and 6.3.2 correspondingly. The presented tools were already analyzed in section 2.3 (Related Work). These tables allow to perform a final balance in terms of the features offered by this dissertation’s proposal and the similar already existing tools.

6.3.1 Requirements Management tools

Table 6 compares the features offered by the requirements management tools analyzed in subsection 2.3.1 with the features offered by ITBox. The compared features include not only the collaborative aspects, but also the RE specific concerns.
<table>
<thead>
<tr>
<th>Developed By</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>TopTeam Analyst</td>
<td>TechnoSolutions</td>
</tr>
<tr>
<td></td>
<td>Administration module for requirements architecture;</td>
</tr>
<tr>
<td></td>
<td>Manage baselines and document versions;</td>
</tr>
<tr>
<td></td>
<td>Custom templates which allow to generate requirements documents very quickly;</td>
</tr>
<tr>
<td></td>
<td>Different user roles;</td>
</tr>
<tr>
<td></td>
<td>Different views and dashboards for each role;</td>
</tr>
<tr>
<td>Modern Requirements</td>
<td>eDevTECH</td>
</tr>
<tr>
<td></td>
<td>Requirements traceability feature;</td>
</tr>
<tr>
<td></td>
<td>Requirements modeling capabilities;</td>
</tr>
<tr>
<td></td>
<td>Creation of high-fidelity prototypes;</td>
</tr>
<tr>
<td></td>
<td>Collaboration pane where team members can engage in back and forth discussion;</td>
</tr>
<tr>
<td>Visure Requirements</td>
<td>Visure Solutions</td>
</tr>
<tr>
<td></td>
<td>User-defined views;</td>
</tr>
<tr>
<td></td>
<td>Full traceability;</td>
</tr>
<tr>
<td></td>
<td>Requirements re-use and requirements baselining;</td>
</tr>
<tr>
<td></td>
<td>Discussion forum;</td>
</tr>
<tr>
<td></td>
<td>Different user roles;</td>
</tr>
<tr>
<td>iRise</td>
<td>iRise</td>
</tr>
<tr>
<td></td>
<td>Integrates very easily with major RM tools;</td>
</tr>
<tr>
<td></td>
<td>Requirements can be quickly documented and traced to objects within models;</td>
</tr>
<tr>
<td></td>
<td>Online reviews and discussions;</td>
</tr>
<tr>
<td></td>
<td>Real-time collaboration;</td>
</tr>
<tr>
<td>Rational DOORS</td>
<td>IBM</td>
</tr>
<tr>
<td></td>
<td>Client-server application;</td>
</tr>
<tr>
<td></td>
<td>Windows-only client;</td>
</tr>
<tr>
<td></td>
<td>Web client – DOORS Web Access;</td>
</tr>
<tr>
<td></td>
<td>Own programming language – DOORS eXtension language (DXL);</td>
</tr>
<tr>
<td>Rational RequisitePro</td>
<td>IBM</td>
</tr>
<tr>
<td></td>
<td>Integrates with Microsoft Word;</td>
</tr>
<tr>
<td></td>
<td>Views of requirements, displaying attributes, traceability, parent/child relationships and impact of requirement change;</td>
</tr>
<tr>
<td>ITBox</td>
<td>Pedro Lopes and Alberto Silva</td>
</tr>
<tr>
<td></td>
<td>Custom templates which allow to generate documents very quickly;</td>
</tr>
<tr>
<td></td>
<td>Document versions;</td>
</tr>
<tr>
<td></td>
<td>Requirements re-use;</td>
</tr>
<tr>
<td></td>
<td>Different user roles;</td>
</tr>
<tr>
<td></td>
<td>User-defined views;</td>
</tr>
<tr>
<td></td>
<td>Different actions for each role;</td>
</tr>
<tr>
<td></td>
<td>Real-time collaboration;</td>
</tr>
<tr>
<td></td>
<td>Management pages summarizing the status of projects and documents;</td>
</tr>
<tr>
<td></td>
<td>Textual validation through RSLingo Studio Integration;</td>
</tr>
</tbody>
</table>

As the Table 6 represents, ITBox aggregates some of the best features of the already existing tools, which may have contributed for the positive feedback given by the pilot-users during the user session (section 6.2).
6.3.2 Project Management/Planning tools

Table 7 compares the features offered by the project management tools analyzed in subsection 2.3.2 with the features offered by ITBox. The compared features include not only the collaborative aspects, but also the PM specific concerns.

Table 7 - Project Management Tools comparison with ITBox

<table>
<thead>
<tr>
<th>Developed By</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Asana</strong></td>
<td>Asana, Inc.</td>
</tr>
<tr>
<td></td>
<td>Web based and mobile application;</td>
</tr>
<tr>
<td></td>
<td>Workspaces can be created by each team;</td>
</tr>
<tr>
<td></td>
<td>Users can follow projects and tasks they are interested in;</td>
</tr>
<tr>
<td></td>
<td>When the state of a project or task changes, users get updates;</td>
</tr>
<tr>
<td></td>
<td>Integration with Dropbox and Google Drive (file storage), Evernote (notes) and Instaganttt (Gantt chart creator);</td>
</tr>
<tr>
<td><strong>Basecamp</strong></td>
<td>Basecamp</td>
</tr>
<tr>
<td></td>
<td>Web based;</td>
</tr>
<tr>
<td></td>
<td>To-dos for tracking work;</td>
</tr>
<tr>
<td></td>
<td>Message Board for posting announcements and updates;</td>
</tr>
<tr>
<td></td>
<td>Chat room for quick casual chats with the team;</td>
</tr>
<tr>
<td></td>
<td>Schedule for posting deadlines and milestones;</td>
</tr>
<tr>
<td></td>
<td>Docs &amp; Files for organizing all the assets and notes;</td>
</tr>
<tr>
<td><strong>JIRA</strong></td>
<td>Atlassian</td>
</tr>
<tr>
<td></td>
<td>Web based;</td>
</tr>
<tr>
<td></td>
<td>Tracking of issues, bugs, stories, projects, tasks, etc.;</td>
</tr>
<tr>
<td></td>
<td>Agile project management (JIRA Agile);</td>
</tr>
<tr>
<td></td>
<td>Integration with other platforms, like Github and Freshdesk;</td>
</tr>
<tr>
<td></td>
<td>Detailed custom reports and dashboards;</td>
</tr>
<tr>
<td><strong>Microsoft Project</strong></td>
<td>Microsoft</td>
</tr>
<tr>
<td></td>
<td>PC-based;</td>
</tr>
<tr>
<td></td>
<td>Creates budgets based on assignment work and resource rates;</td>
</tr>
<tr>
<td></td>
<td>Resources can be shared between projects, and have their own calendar with their availability;</td>
</tr>
<tr>
<td></td>
<td>Automatic scheduling of tasks based on resources availability;</td>
</tr>
<tr>
<td><strong>Wrike</strong></td>
<td>Wrike, Inc.</td>
</tr>
<tr>
<td></td>
<td>Web based;</td>
</tr>
<tr>
<td></td>
<td>Managing and tracking of projects, deadlines, schedules, and other workflow processes;</td>
</tr>
<tr>
<td></td>
<td>Private collaboration between users;</td>
</tr>
<tr>
<td></td>
<td>Integration with Microsoft Excel, Microsoft Project, Microsoft Outlook, Dropbox and Google Drive;</td>
</tr>
<tr>
<td><strong>ITBox</strong></td>
<td>Web based;</td>
</tr>
<tr>
<td></td>
<td>Integration with Google Drive (storage), Google Documents and Google Sheets (editors);</td>
</tr>
<tr>
<td></td>
<td>Users can invite other users to their projects;</td>
</tr>
<tr>
<td></td>
<td>Projects contains documents (artifacts);</td>
</tr>
<tr>
<td></td>
<td>Textual validation through ProjectLingo Studio Integration;</td>
</tr>
</tbody>
</table>

As the Table 7 represents, ITBox aggregates some of the best features of the already existing tools, which may have contributed for the positive feedback given by the pilot-users during the user session (section 6.2).
7 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 dissertation proposed ITBox, a collaborative web-based platform for better managing technical documentation, including such types of requirements specification documents.

In particular, ITBox allows the management of requirements specification documents, 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.

ITBox already supports multiple documentation types such as requirements specifications, tests specifications or project plans. So, 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.

The work presented in this dissertation was developed over the last 12 months using the Action Research methodology and evaluated to gage its overall usefulness and the quality of its processes. The results were obtained through a user session in which the participants were asked to follow a user guide containing a set of instructions that implied making use of some of the platform’s capabilities.

The remaining of this chapter is organized as follows: section 7.1 describes the main contributions of this research work and section 7.2 indicates possible future research directions.

7.1 Main Contributions

We believe that ITBox provides a well-rounded collaborative platform for better managing technical documentation. It combines some of the most useful points of other tools available, namely Requirements Management and Project Management tools.

Furthermore, it is based on an extensible technology, the Google Drive and its APIs, making it very easy to develop new features regarding the extraction and analysis of the data from the documents.
Further contributions of this work can be aligned with the research goals defined in section 1.2 and with the following research questions:

**RQ1:** Provides comparative information regarding the strongest features of the available collaborative platforms for RM and PM.

**RQ2:** Provides a new collaborative platform which combines some of the main benefits from each one of the analyzed before, like (1) document versions, (2) user-defined view and (3) real-time collaboration.

**RQ3:** Provide a platform capable of supporting multiple types of technical documentation, offering different editors.

### 7.2 Future Work

This section presents some of the main directions that can be followed in a future research related with this dissertation. Some of these proposals were conceptualized, but did not get the change to be implemented. Nevertheless, it is important to emphasize that none of them undermine the achievement of this dissertation goals.

**Provide editors with more powerful capabilities** – The current platform’s editor for language files is Google Docs. This editor is very limited in terms of features related with the documents quality. As described in subsections 3.1.3 and 3.2.3, the RSLingo and ProjectLingo Studio Tools provide features like syntax highlighting, error checking and auto-completion. Although ITBox can integrate with these tools, allowing the validation of the respective documents, an “Ad-hoc” integration is not ideal. Therefore, a more advanced integration could make use of an editor like the Xtext web editor[^33], which provides identical features to the Studio tools.

**Extend the Variability Model beyond the RSLingo approach** – One of the ITBox’s reusability features is the possibility of defining variability models (*VarModels*) from Libraries. Although, these Libraries must be of the “RSL Excel” *DocType*, otherwise it will not be possible to define those *VarModels*. The reason why the variability model is only adapted to the RSLingo approach is because this is the only approach which both Excel Template and the language itself are completely established. When the documents of approaches like the ProjectLingo are more refined, it will be possible to define more accurate variability models.

**Extend the Libraries supported modules** – ITBox supports the creation of Libraries, which are defined by Modules (or Views). Users must specify which modules they want to include in the library they are creating, at creation-time. As it happens with the Variability Models, these modules are also dependent on the Excel Templates and language files defined by the approaches. Therefore, to fully support new approaches, the ITBox must be updated, so that the supported library modules can be extended. A perfect solution would be to make this process completely autonomous, not requiring code manipulation.

**Enhance several usability aspects of the ITBox platform** – Regarding the web platform, the visual frontend and some usability features were overlooked when trying to guarantee that at least the whole project and its features worked as intended. With this said, there are clear ways to make the system more user-friendly. Usability implies many testing sessions with pilot users. For future enhancement of this aspect, more tests should be done, iterating the process of the interface design and functionality.

References


Appendix A
User Session Guide
The objective of this pilot-user test session is to perform an evaluation of ITBox platform by users that are not familiarized with it, to detect bugs or user limitations. The overall usability and utility of the ITBox platform will be tested by the users through the creation of projects, documents and libraries. The platform will be tested, not you, so any difficulties or suggestions that you might have will be useful to determine further improvements to the platform.

During the session, you will access the platform using three different accounts, with different roles. This means that you will view different pages and perform different actions, depending on the account that you are logged in.

**Test Conditions:**

This user test session will be conducted under the following conditions:

- Tests are conducted in the laboratory (controlled environment);
- The tasks must be performed without previous use and learning (for the first time);
- The user must have a computer with a running web browser (Chrome preferred);
- Direct Observation, i.e., while users perform the assigned task, their behavior and performance can be logged;
- Users can think out loud and share ideas if they want;
- The evaluator does not interact with the users until the tests are finished (except in case of blocking errors);
- The session will last 30 minutes (at most).
- The user must fill a survey in the end – available at: [https://goo.gl/forms/Hz3zU8sbe6gqalt13](https://goo.gl/forms/Hz3zU8sbe6gqalt13)
Instructions:

1. Download the Excel files named “RSL-BillingSystem.xlsx” and “RSL-Cultural_Module.xlsx” from [https://drive.google.com/drivefolders/0B9RfuqO9zXuNVjZKUIJLQFFRTUk](https://drive.google.com/drivefolders/0B9RfuqO9zXuNVjZKUIJLQFFRTUk);

2. Rename the file named “RSL-BillingSystem.xlsx”, by adding your name as a prefix (e.g. pedrolopes_RSL-BillingSystem.xlsx);


4. Login the platform using the credentials:
   
   4.1. **Username**: ITBox_Manager
   
   4.2. **Password**: manager

   4.3. During the next steps, you will execute tasks which are only available to users who have the **ITBox Manager role**. This kind of users have access to the “Home”, “Templates” and “Libraries” pages.

5. **RSL Excel Template Upload**:
   
   5.1. Enter the “Templates” tab;
   
   5.2. Select “+ New Template”;
   
   5.3. In the option “Type” select “RSL Excel”;
   
   5.4. In the “File to Upload”, choose the Excel file that you renamed previously and click “Save”;
   
   5.5. Wait for the upload to finish and confirm that the file uploaded correctly;

6. **RSL Excel Library Creation**:
   
   6.1. Access the “Libraries” tab;
   
   6.2. Click the button “+ Create Library”;
   
   6.3. As the name, type your own name followed by “Cultural Lib” (e.g. Pedro Cultural Lib);
   
   6.4. Write a small description of the library (or alternatively, just write “Test library”);
   
   6.5. For the **Type**, select “RSL Excel”;
   
   6.6. For the **Modules**, select “Quality Requirements”;
   
   6.7. In the document to upload, select the “RSL-Cultural_Module.xlsx” file you downloaded previously;
   
   6.8. Click “Save”;
7. **Logout** the platform and **create** a new account;
   7.1. Click “**Logout**”, located on the top right corner;
   7.2. Now, click “**Login**”, located on the same place;
   7.3. Click “**Register**” and complete the required fields as you wish.
   7.4. After completing the registration process, you will be logged in with the account you just created. This account has the **ITBox User role**. This means that now you can only see the “Home” and “Projects” pages.

8. **Project Creation**:
   8.1. Enter the “**Projects**” tab;
   8.2. Click “**+ Create Project**”; 
   8.3. As the name, type your own name followed by “ITProject” (e.g. Pedro ITProject);
   8.4. Write a small description of the project (or alternatively, just write “Test project”);
   8.5. Click “**Save**”;
   8.6. You will be redirected to the Projects Page. Confirm that a new project was created with the name that you provided;
   8.7. Your newly created project should have the “Not Started” status. Change it to “On Execution”; 
   8.8. You can now invite users and add documents to your project;

9. **Add Document to Project**:
   9.1. Still in the Projects Page, in the “Actions” column, click the icon 📃, to access your project’s documents;
   9.2. As your project does not have any documents yet, there will be no documents to show;
   9.3. Click the “**+ New Doc**” button, below the documents table;
   9.4. As the name, type “BillingSystem”;
   9.5. For the **Document Type**, select “RSL Excel”; 
   9.6. For the **Sub-Type**, select “Simple Document”;
   9.7. About the Add Document From, select the “Template”;
   9.8. In the “**Document to Add**” option, select the document you uploaded previously;
   9.9. Click “**Save**”. It should take some seconds, the file is being uploaded to the cloud;
9.10. Now your project has a document. Confirm that this document was added with the name you provided and click the Access Document icon; Explore the spreadsheet document, particularly the \texttt{reqs.quality} sheet. If you want to increase the size of the document for better reading, click the “Maximize” button;

10. \textbf{Append Library to RSL Excel Document}

10.1. Enter the “Projects” tab;

10.2. Locate the project you created in point 8 and access the project documents clicking the \texttt{\text{open}} icon in the “Actions” row;

10.3. Locate the previously created document, and click the icon \texttt{+}, to append a library to your document;

10.4. Select the library you created previously;

10.5. You can check the library’s quality requirements that will be appended to your document;

10.6. Click the button “Append Library”;

10.7. Access the document, clicking the \texttt{\text{open}} icon and check that the requirements from the cultural library have been added to the \texttt{reqs.quality} sheet. (Google often loses formatting on new cells, so do not worry if the cells are not formatted correctly);

11. \textbf{Project Collaboration}:

11.1. Access the “Projects” tab;

11.2. Locate the project you created in point 8 and click the \texttt{\text{open}} icon in the “Actions” row;

11.3. The presented table shows the users who are involved in your project and what their project roles. Currently, your project should have no more users (beyond you);

11.4. Click the “+ Invite User” button, located bellow the users table;

11.5. In the user to invite, type \texttt{consultant@itbox.com”};

11.6. In the user role, select the “Consultant”;

11.7. Click “Invite”;

11.8. You should now see in the table the user you just invited;

11.9. Click “Logout”, located on the top right corner;
12. Login as Project Consultant
   
   12.1. Now, click “Login”, located on the same place;
   12.2. Login the platform using the credentials:
   12.3. Username: ITBox_Project_Consultant
   12.4. Password: consultant
   12.5. Explore the different permissions this account has. (This account has the ITBox User role, that is why you see the same pages as with the account you previously created. This user is consultant in different projects, including the one you created during this session. For example, if you access the documents of the projects where this user is consultant, you will not be able to edit them or change their status). Additionally, you will not be able to invite new users for the projects in which you are a Consultant;

For more information, refer to the informative documents in the folder: https://drive.google.com/drive/folders/0B9RfuqO9zXuNVjZKUIJLOFFRTUk

- The presentation document for the user session (ITBox_Presentation.pdf);
- The paper titled “A Collaborative Platform for Better Managing Technical Documentation: An analysis from a Requirements Engineering Perspective”.

And please, don’t forget to fill in the “ITBox Pilot-User Test Session Survey” available at: https://goo.gl/forms/Hz3zU8sbe6qsalt13

Thank you for your help!
Appendix B
User Session Questionnaire Results
Summary

**Age**
10 respondentes

<table>
<thead>
<tr>
<th>Age</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>2</td>
<td>20%</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>48</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>10%</td>
</tr>
</tbody>
</table>

**Gender**
10 respondentes

- Male: 70%
- Female: 30%
**Academic Degree**

- BSc: 50%
- MSc: 20%
- PhD: 30%

**Number of years of professional experience**

- 0 years: 4 (40%)
- 1 year: 1 (10%)
- 2 years: 1 (10%)
- 3 years: 1 (10%)
- 4 years: 1 (10%)
- 10+ years: 2 (20%)

**Knowledge and previous experience with Requirements Engineering**

- Yes: 90%
- No: 10%
ITBox Web Platform [How do you rate the overall usability of the Web platform?]

0 - (N/A – Do not know)
1 - Very Low
2 - Low
3 - Medium
4 - High
5 - Very High

ITBox Web Platform [How do you rate the usefulness of the main productivity features? (Projects Management, Doc Editor, Template Manager, Libraries)?]

0 - (N/A – Do not know)
1 - Very Low
2 - Low
3 - Medium
4 - High
5 - Very High

ITBox Web Platform [How easy to learn (or how familiar) was the main document editing tool (Google Sheets)?]

0 - (N/A – Do not know)
1 - Very Low
2 - Low
3 - Medium
4 - High
5 - Very High

ITBox Web Platform [How suitable is the platform for managing Technical Documentation?]

0 - (N/A – Do not know)
1 - Very Low
2 - Low
3 - Medium
4 - High
5 - Very High
Multiple Users [How do you rate the suitability of the existing Project Roles (Owner, Member, Consultant)?]

0 - (N/A – Do not know)  
1 - Very Low  
2 - Low  
3 - Medium  
4 - High  
5 - Very High

Multiple Users [How do you rate the usefulness of having different user roles in the same project?]

0 - (N/A – Do not know)  
1 - Very Low  
2 - Low  
3 - Medium  
4 - High  
5 - Very High

Multiple Users [How do you rate the simplicity of adding a new user to a project?]

0 - (N/A – Do not know)  
1 - Very Low  
2 - Low  
3 - Medium  
4 - High  
5 - Very High

75
Do you think it is useful to have just one platform to manage multiple projects of different technical documentation?

10 responses

100%

Would you use this platform on your own Engineering projects?

10 responses

100%
### Additional Comments (Suggestions, Problems, Bugs)

<table>
<thead>
<tr>
<th>Number</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>When maximizing the sheet it goes to the center of the page and not show up at first in the top. I was able to change the status of a project, using the consultant account, by editing its properties.</td>
</tr>
<tr>
<td>2</td>
<td>I think that you should add a Graphical File Upload Progress Bar.</td>
</tr>
<tr>
<td>3</td>
<td>no register, para screens pequenos meter uma scroll bar, para convidar colaboradores, escrevendo o mail a mão não deu, só selecionando os que estavam na lista.</td>
</tr>
<tr>
<td>4</td>
<td>Os icons nas acções do consultor devem ser revistos para projeto, para documentos e para &quot;roles&quot;. Sugiro uma matriz para verificar as operações para cada tipo de utilizador. O botão de criação de utilizadores aparece na parte inferior, ao contrário do que acontecia para projeto e documento.</td>
</tr>
<tr>
<td>5</td>
<td>Os projetos e documentos deviam, além da data disponibilizar o utilizador de criação. Devia ser possível ver a descrição associada, nem sempre o nome é identificativo e já que está disponível a sua inserção no registo devia estar disponível na lista, mesmo que em detalhe ou via ícone para aceder à descrição.</td>
</tr>
</tbody>
</table>