Transformations and Alignments in Project Management Tools: The PSL-v.2020 Experiments

Pedro Filipe da Cruz Baptista
Instituto Superior Técnico
Lisbon, Portugal
pedro.cruz.baptista@ist.utl.pt

ABSTRACT

Project management (PM) is the application of knowledge and skills to meet project requirements. Currently, there are two general PM approaches: the agile approach and the traditional, or predictive approach. The PM traditional approach follows international frameworks and standards like PMBOK, ICB, or ISO 21500. There are plenty of PM tools that provide features focused on the support of one or the other approach. For the traditional approach, each of these features can be classified according to a process group and a knowledge area (referred in the PMBOK terminology), while for the agile approach they are classified according to a framework. This thesis analyzes and discusses transformations and alignments in PM tools and concludes that there are an extensive set of them.

Author Keywords

ITLingo, PSL Excel Template, Project Management, Agile Methodologies, Project Management Tools.

1. INTRODUCTION

1.1. Context

Project management (PM) is the application of knowledge, techniques and tools to a given project. PM can be divided in two groups of methodologies. Traditional methodologies are organized by sequential process groups. However, many projects require doing repetitive tasks with intermediate demonstrations of the work developed, instead of having just the demonstration of the final product delivery [3]. On the other hand, agile methodologies are organized by multiple iterations that produce deliverables in each concluded cycle. This approach is used in IT projects, since it allows the team to get frequent feedback and adjust the product, leading to a continuous improvement over time.

Many PM tools are available on the market and they tend to specialize in one of the two approaches: traditional or agile [3]. Tools for agile approaches are mostly known for their collaborative features and for supporting agile methodologies like Scrum [4] or Kanban [5]. Some of these tools are JIRA [17], Azure DevOps [18], or Asana [19]. On the other hand, tools for traditional approaches still tend to be collaborative and to follow international frameworks and standards but, in general, they do not support the majority of the domain or areas of PM, such as Knowledge areas (KAs) of PMBOK [24]. Some of the most popular tools are MS Project [11], ProWorkFlow [16] or Confluence [12]. Since these tools do not support all the KAs, project managers must use more than one and use Office tools like MS-Office, Google Docs and Sheets.

Nowadays, the need to develop or improve products and services is critical for the survival of any organization, and therefore, these achievements shall be conducted as projects, which should be planned, analyzed, and controlled throughout their lifecycle and should adopt the best approaches and software tools, regardless if these approaches are more agile or more predictive [28].

1.2. Motivation

The motivation behind this research is to allow project managers to elaborate their plans with a single tool, while providing extensive validation mechanisms and macros that automate several tasks. Considering this general objective, the platform chosen to develop the tool was MS Excel because it is a popular and the most used tool for data management [31]. MS Excel allows every user with basic technical knowledge to adapt the initial template to his own needs and use macros to do repetitive tasks.

1.3. Proposal

The PSL (Project Specification Language) Excel Template v.2020 is a tool developed on top of the former version, v.2019 [32], which tries to mitigate the problem of using multiple tools to facilitate PM. This tool considers international frameworks and guidelines and proposes a solution that integrates the information spread in such different KAs. Because it is an Excel template, it is simple to use, with a fast learning curve that takes advantage of the MS-Excel itself. The tool includes validation mechanisms for the inserted data and automation features concerning data alignments and transformations, which is the focus of this research. These features provide many options and promote productivity by removing the need for manual or repetitive tasks. Comparing this template to other available tools, despite its coverage, ease to use and adoption, it loses in terms of collaborative and reporting features but excels in terms of integration and productivity.

1.4. Research Methodology

The research and development that led to the PSL-v.2020 was done between October 2019 and October 2020. To produce this work, we used the Design Science Research Methodology (DSRM) [30]. DSRM always outputs a purposeful artifact which “can be a product or a process; it can be a technology, a tool, a methodology, a technique, a procedure, a combination of any of these, or any other means for achieving some purpose" [29]. The process was divided into 4 major cycles: research, acquiring technical skills, implementation, and evaluation.

1.5. Structure

This paper is structured in 8 sections. Section 1 introduces the context this research and the motivation behind the proposed PSL-v.2020. Section 2 introduces the background and discusses PM approaches and technologies used. Section 3 analyses in more depth the most popular PM tools and Excel Templates available. Section 4 presents the overall improvements considered and its separation into the Traditional and Agile templates. Section 5 analyzes in depth the improvements and modifications made on the traditional version. Section 6 presents and analyzes the agile template. Section **Error! Reference source not found.** shows the evaluation of the proposal with three case studies and discusses the tools analyzed in Section 2 with the PSL-v.2020. Section 8 is the conclusion.

2. BACKGROUND

This section introduces the research background regarding the topics of project management, agile methodologies, the IT Lingo initiative, and the used technologies.

2.1. Project Management

Project management (PM) is the application of practices, principles, processes, tools, and techniques to project activities to meet project requirements [24, 25]. PM can be also described as an organizational approach to the management of ongoing operations [24] and the managers of these projects shall satisfy their stakeholders, the people involved and affected by the result of the project. To improve and disseminate the best practices of PM some international frameworks and standards have been defined, such as PMI [25], PMBOK [24], IPMA ICB [26] or ISO 21500 [27]. For example, PMBOK describes a project as the combination of time, financial, and technical performance goals defined by the project stakeholders and, according to PMBOK, the analysis of the PM discipline is divided into 5 process groups: initiating, planning, executing, monitoring & controlling and closing; and 10 knowledge areas (KA) integration, scope, schedule, cost, quality, resource, communications, risk, procurement, and stakeholder. However, these processes often involve long planning phases with a big design up front and large documentation efforts and are not designed to adapt to changing requirements, unexpected events or even the interaction with customers [3].

Therefore, due to these limitations, agile methods, like Kanban [5] or Scrum [4], have increased their popularity over the last two decades, mainly because they have been able to mitigate some of these challenges by emphasizing collaboration within teams, active customer involvement, change tolerance, and iterative delivery of features in an uncertain context [3]. However, they are designed for small, self-organized, and collaborative teams that work closely with customers, which is not the case in many situations.

2.2. PM Agile Methodologies

Agile PM focuses on deliverables and respective customer feedback. To create those deliverables, it uses an iterative approach with small cycles because agile development is about continuous improvement over the best practice. The most relevant agile frameworks are Scrum and Kanban.

2.2.1. *Scrum*

Scrum is a framework for developing and sustaining complex products [4]. This development uses three pillars: transparency, inspection, and adaptation. This means the process is kept visible and understandable for every party involved, the artifacts are reviewed at any given time to detect flaws, and the process is adjusted as soon as possible to address problems or undesired outcomes. Scrum is categorized by Scrum's roles, events, artifacts, and the rules binding them together.

A Scrum team is a self-organized and cross-functional group of people responsible for the completion of the project. The team members can have one of the following roles: Product owner, Development team member and Scrum Master. The Product owner is responsible for defining the value of the product and the work of the development team while the team defines their work schedule. The team does not have any individual titles, so every member is accountable for the work developed. The Scrum Master is the bridge between the product owner and the development team. He or she is also responsible for providing help to the product owner during the identification phase and helping the development team solve any problem.

The events in scrum are specifically designed to enable critical transparency and inspection [4] and are organized within a Sprint to avoid any unnecessary meetings. The sprint is defined as the iterative concept present in agile methodologies and consists of a well-defined period where all the processes and activities are executed. Inside the Sprint, the daily scrum is a flash meeting of fifteen minutes at the beginning of each day where the development team does a quick overview of the work status. The Sprint review is held at the end of a sprint where the sprint is evaluated, and product backlog can be modified to improve the product's value. The Sprint retrospective is held after the team review and focuses on analyzing the team performance.

The Artifacts in scrum represent the work of the project and are divided in two main categories: Product Backlog and Sprint Backlog. The Product Backlog is defined by the product owner and contains a list of features, functionalities, and requirements representing all the work expected to be done during the project. The Sprint Backlog contains a list of items selected from the Product Backlog for a sprint.

2.2.2. *Kanban*

Kanban [5] is another agile framework focused on getting things done and fast by improving workflow. This is achieved using a Kanban board that provides a clear view of the work divided in small tasks.

The concept behind the board is to represent the process phases of a task through columns. The main concepts are: To Do also known as backlog or the first phase of a task; In Progress; and Done. The size of the board is not restricted to just three columns as a task can have more than just three phases.

Kanban has critical features that differentiate the framework from other agile methodologies. First, it uses a restriction to limit the quantity of Work in Progress (WIP), giving the team an idea of the number of tasks that should be present in the In-Progress Column, at every moment. And second, it has the capacity to measure lead time and cycle time allowing the team to do a correct assessment of a task's life cycle. Lead time represents how long it took a task to move from all the operations to the Done column. Cycle time starts when a task is moved to the In-Progress column and concludes when the task leaves this column.

Looking at the main qualities of Kanban compared to other agile approaches, Kanban is easier to implement and adopt, consequently it is a more gradual approach for teams that are looking to move from traditional to agile methodologies.

2.2.3. *PM^2*

PM^2 is the most recent PM methodology and it is a union of the traditional with agile methodologies [6]. This union follows the same phases as the ones defined by the PMBOK but gives the project manager the freedom to decide what processes are relevant or not. Also, the implementation phase can follow a sequential planning like a traditional approach or use an agile framework. However, unlike the two other methodologies, PM^2 teams have a business manager responsible for the business value of the product and the interaction with the product owner.

2.3. ITLingo PSL

The ITLingo initiative [1] defines rigorous domain-specific languages for IT specific artifacts such as project plans. By doing so, the initiative provides a centralization of a project in one place unlike any other PM tools. This initiative is composed of several languages, one of them is the Project Specification Language (PSL) [2]. With PSL, specification of project plans can become more rigorous and systematic. The PSL-v.2020 was used as a base for the PSL development but has been further improved with more details, correctness, features, automatisms, and validations.

2.4. Technologies

This section briefly introduces the technologies that were used during this research (See FigureFigure **1**. Relationship between PSL-v.2020 Tools).

Microsoft Excel. is commonly used because it provides easy access to stored data, input of new data, simple to complex math's calculations, built in operations for alignment purposes, various types of graphs for representation, creation of tables, and development of macros.

Visual basic for applications. (VBA) [7] is an internal programming language for Microsoft Office tools. This tool allows programmers to customize the graphical user interfaces and develop macros that translate a sequence of inputs into a desired output. These macros were implemented on the PSL-v.2020 to help users define the project specifications.

Office RibbonX [8] is an open-source tool to customize the ribbon of MS tools. The customization includes new separators, new groups, and new options. This tool is used to create the PSL-v.2020 ribbon options that trigger the associated VBA macros.

Software key [9] is a tool used to license software securely. The system uses a public-key cryptography with a trial-version. This means a PSL-v.2020 trial can be securely distributed to the public and then activated to a full version. This also means the template needs to be analyzed and evaluated to identify what should be available on the trial version.

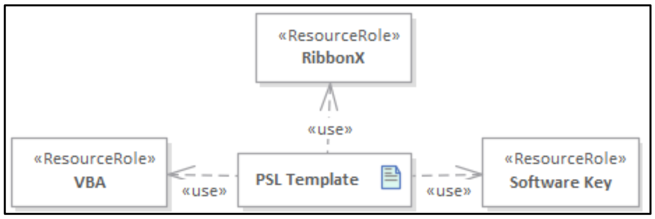


Figure 1. Relationship between PSL-v.2020 Tools

3. RELATED WORK

The PSL-v.2020 is a PM software tool and therefore we had to understand the current state of PM software tools. This research provides an insight of the available PM features and how they help to improve the PSL tool.

3.1. PM Tools

First, we analyze the most common and popular software PM tools supporting traditional and agile approaches.

Microsoft Project or MS Project. [11] is a PM tool specialized in two key aspects: the timetable based on a Gantt chart; and the expenses (containing information on the costs). The Gantt chart is fully customizable, consisting of work packages, activities, milestones, and dependencies between them. The resources can be thoroughly described in the expenses section and allocated to activities. Together, these aspects provide a complete view of the project schedule and costs. This tool can be used alongside with MS Share point.

ProWorkFlow. [16] is a complex tool that offers PM features focusing on 5 key aspects: time, tasks, staff, expenses, and reports. It allows the visualization of information in multiple projects simultaneously, providing a global schedule of the project manager's supervised activity. All the project's information is visible on the same page, with their dates, status, and a time graph. To create a project, the user gives the details, is able add a description, assign the staff involved and define the initial plan by choosing from a list of templates with various PM structures for various types of projects. These templates are well elaborated and specific to PM

Confluence [12] is distributed by Atlassian and is focused on the traditional PM approach. Confluence's main idea is that every web page is a blank page developed from scratch and can be customizable by any user. This tool does not provide PM templates but there are features for almost any type of content.

Focusing on Agile PM tools, **Azure DevOps** [18] is the Microsoft tool for agile teams. This tool provides a group of boards divided in work items, boards, sprints, backlogs, and queries. These groups are interconnected and customizable. However, Azure DevOps best features come from the other three groups: repos, pipeline, and test plans. They provide repository features, a development architecture and test features, making it an ideal tool for programmers.

JIRA Agile [17] has all the features required to apply the most common agile approaches but what distinguishes it, is the interoperability between Scrum and Kanban where the board on each project type can be adapted to the other as the items offer no constraints regarding their use on either one of the frameworks. This feature is an opportunity for teams to enjoy qualities of both approaches and to use a mixed methodology mostly known as Scrumban or Kanplan [21].

Asana [19] is a simpler and more customizable software tool than JIRA Agile. This tool offers 7 categories that are straight-forward with a fast learning curve. For this reason, Asana is ideal for teams that are looking to start using agile approaches before transitioning to JIRA Agile.

Quire [20] is the simplest tool to support a Kanban approach. The principle behind the software is unfolding ideas. These ideas are then turned into a customizable Kanban board with overview features. Quire is minimalist but guarantees success and user satisfaction their project visualization and management.

Smartsheet [14] is the last analyzed tool for PM because while it is not an Excel Template it uses the same base concepts. The tool uses sheets with data cells to store the information just like an Excel Template but how this data is presented depends on the desired purpose. It can be a grid, card, Gantt, or calendar view. Another difference is the available collaborative features for simultaneous development and live updates since the tool is built on a cloud service. This tool does not provide support for the 10 KAs but shows that the PSL-v.2020 can be improved in terms of flexibility and customization.

3.2. PM Excel Templates

Second, we analyze the most common and popular PM excel templates supporting traditional and agile approaches.

Starting with the **PSL-v.2019**, it provides a well-defined structure for the PSL and supports the application of both the traditional and agile methodologies [32]. The PSL-v.2019 is the last iteration of the tool with new features compared to older versions. These features are: a new organization of the configuration sheets, sheet protections to the entire workbook, a customized ribbon, and exporting features. This version also studied the use of a licensing software.

The PSL-v.2019 was evaluated using a fictional project. This evaluation showed it was possible to perform all the tasks required for a traditional approach using data tables where each table or group of tables focus on specific KAs.

The evaluation also detected some problems. First, only 8 KAs were supported. Second, some features were not properly working, generating bugs or incorrect outcomes. And third, while this template tried to provide support to Scrum it was not ready to do so.

Office Timeline [15] has a wide variety of PM excel templates. The most relevant features from these templates are the export for PowerPoint and options for report generation.

Gantt Excel [10] is a template focused on the Gantt Chart present in a traditional approach. This template is small with just three sheets but contains many configurations in its customizable ribbon. It is possible to create tasks, dependencies, manage resources and configure the view. Along with the ribbon the template uses forms facilitating the insertion of data and their modifications. These forms appear when a user clicks on specific cells and on some ribbon options.

Vertex42 [13] like Office Timeline, has many PM excel templates. These templates are more detailed, elaborated and focus on the project time, project budget, key performance indicators (KPIs), dashboards, and Kanban boards.

3.3. Tools Comparison

Lastly, we compare all the tools in terms of support for traditional and agile approaches.

Table 1. Comparison of tools based on traditional approaches.

		KAs Support	Custom-ization	User Friendli-ness	Report Generat-ion	Data Visualiza-tion
Tools	SmartSheet	++	+++	++	+	+++
	MS Project	+		++	++	+++
	ProWorkFlow	++	+	++	+++	++
	Confluence	++	+++	++	++	+++
	PSL-v.2019	+++	+	++	+	+++
Templates	Office Timeline	+	+++	+	+++	++
	Vertex42	++	++	+++	++	+++
	Gantt Excel	+	+++	+++	++	++

Table 2. Comparison of tools based on agile approaches.

		Scrum Support	Kanban Support	Customization	User Friendliness	Report Generation	Data Visualization
Tools	Smart Sheet		++	+++	+++	+	+++
	Azure DevOps	+++	+	++	+	+	++
	Jira Agile	+++	++	+	++	++	++
	Asana	+	++	++	+	+	++
	Quire		+	+	+++	+	+++
Templates	**PSL-v.2019**	+		+	++	+	+++
	Vertex 42		++	+	+++	+	+++

4. PROPOSAL

The PSL-v.2020 fixes the problems of the PSL template previous version, improves customization, and provides new features for the traditional and agile approaches. All these improvements are discussed and explained in this section and in section 5 and 6.

4.1. Tool new Structure

With the continuous addition of sheets, features, transformations, and alignments the template became a long workbook difficult to maintain.

Also, considering the Scrum and Kanban features are independent of the traditional sheets because they are agile frameworks, we decided to divide the template in two different versions: the PSL/Traditional-v.2020 and the PSL/Agile-v.2020.

The Traditional version has a new organization and 4 new sheets: Stakeholder.Organizations, Stakeholder.People, Resource.Resources and Procurement.Contracts. Figure Figure **2**. PSL Excel Traditional Methodologies Template Architectureprovides an overview of this new structure and its changes.

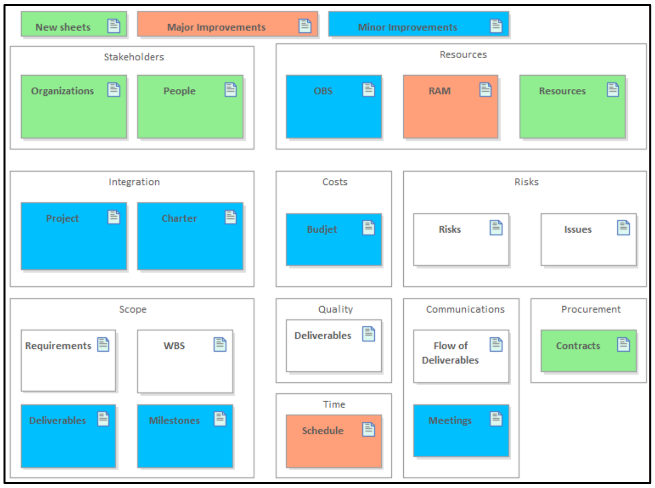


Figure 2. PSL Excel Traditional Methodologies Template Architecture

The PSL/Agile template keeps the older agile sheets focused on the Scrum framework, but its features were changed, improved, and replaced with new ones. This template is now organized in three groups as suggested in Figure 3. The Project group has ProjectIntegration, StakeholdersOrganizations, StakeholdersPeople, Team and Team Analyzer sheets since they are relevant for both Scrum and Kanban. Scrum group contains 4 sheets: ProductBacklog, Sprint Backlog, Sprint Analyzer and Sprint History. The last group is Kanban with two sheets: Kanban and Kanban Analyzer. All the groups are interconnected, and it is possible to use both Kanban and Scrum simultaneously.

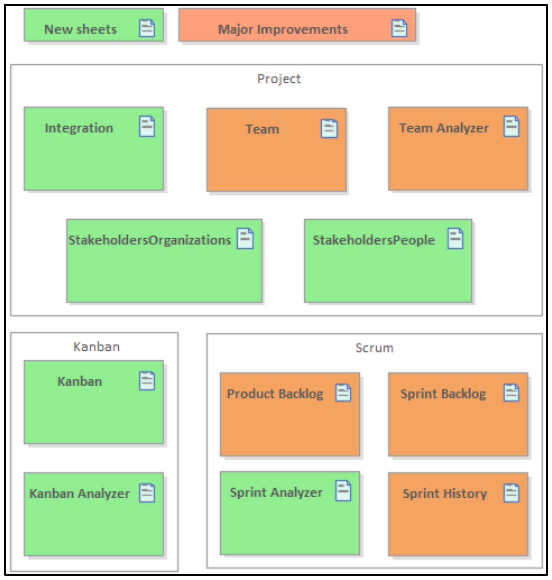


Figure 3. PSL Excel Agile Methodologies Template Architecture

This division of the tool did not remove any features from the previous version. This architecture only distributed data that was not logically or practically related and facilitates future specialization, development, and evolution.

5. PSL TRADITIONAL TEMPLATE

This PSL/Traditional-v.2020 supports all the 10 KAs as defined by the PMBOK, namely: integration, scope, schedule, cost, quality, resource, communications, risk, procurement, and stakeholder. These areas (KAs) are distributed among different sheets but maintain logical and structural dependencies between them. Many Excel formulas presented in various cells and tables assure those dependencies and validate the consistency of data. Some sheets with data configurations are also available to the users. On top of this, a specific Excel ribbon logically aggregates options that allow the alignment and transformation of data from logically dependent sheets, update graphs, and tables already present in the Excel. These extra options are implemented with VBA and RibbonX. This template allows a user to manage all the project information in one place and provide interoperability features such as data export into Word, PDF, JSON and other formats. Compared to the older version, PSL/Traditional-v.2020 supports the 2 missing

KAs, provides new graph configurations, control features for the project execution phase, improves user experience, and provides export features.

5.1. Alignments

This PSL/Traditional-v.2020 defines and maintains 22 dependencies between its sheets, as shown in Figure Figure 4. These dependencies are classified in 3 categories: weak, moderate, and strong. A **weak** dependency means the existence of a logical dependency, but with no representation on PMet. A **moderate** dependency means the existence of automatic mechanisms or suggestions of data to insert. A **strong** dependency means the existence of one or more macros options that guarantee that dependency. These dependencies are the following, represented as Di:

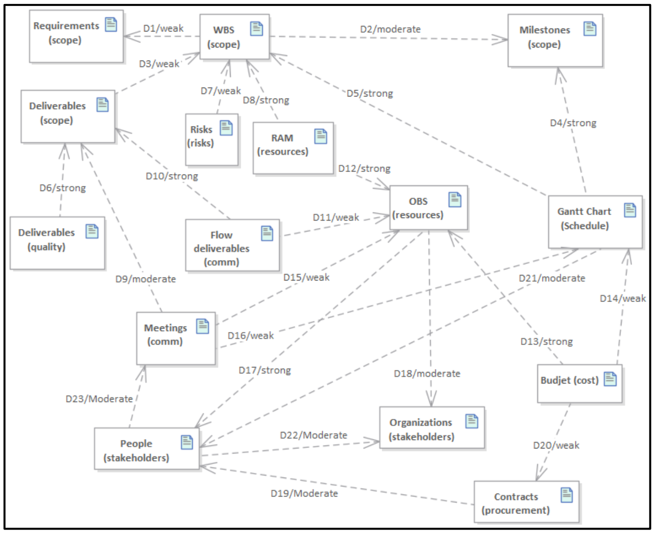


Figure 4. Dependencies in the scope of PSL/Traditional-v.2020 (in BPMN)

D1 represents the dependencies of the work packages (WPs), defined in the WBS, with the requirements of the project.

D2 represents the dependencies between each work package (WP) in the WBS and one or more milestones. These are defined in the milestone sheet and automatically fill the Milestone ID and Milestone Name columns on the WBS for the respective WP.

D3 represents the dependencies between each deliverable and a respective WP, in which work context that deliverable will be produced.

D4 represents the dependencies between the milestones and WPs of the Timetable (or Gantt Chart) with the respective milestone defined in the Milestones table.

D5 represents the dependencies between the activities defined in Timetable (or Gantt Chart) and the Phases and WPs of the WBS.

D6 represents the dependencies of the quality criteria of deliverables and the respective deliverables defined in the scope.

D7 represents the dependencies of risks that might be defined according to the structure of the WBS (e.g., phases and WPs).

D8 represents the dependencies between the rows of the RAM (i.e., phases and WPs) and the WPs defined in the WBS.

D9 represents the dependencies between each meeting and the deliverables that should be analyzed and/or approved in the context of each meeting.

D10 defines the deliverables that are used in the context of each communication flow.

D11 defines the people or resources that are involved in the context of each communication flow.

D12 represents the dependencies between the column's headers of the RAM and the top-level resources defined in the OBS.

D13 represents the dependencies between budget and each resource defined in the OBS.

D14 represents the dependencies between costs of people assigned to activities defined in the Timetable (or Gantt chart).

D15 represents the dependencies between the meeting participants with the resources defined in the OBS.

D16 represents the dependencies between the meetings with the respective activities defined in the Timetable (or Gantt chart).

D17 represents the dependencies between the OBS and the people defined in stakeholders.

D18 represents the dependencies between the OBS defined people and the organizations defined in the stakeholders.

D19 defines the people or resources involved in the context of each contract.

D20 represents the dependency between budget and contracts values.

D21 defines the people or resources that are assigned to activities defined in the Timetable (or Gantt chart)

D22 represents the dependency between each person and the respective organization.

5.2. Illustrative Example and Transformations

The best way to demonstrate some PSL/Traditional-v.2020 features, namely its distinctive aspects, related alignments, and transformations, is by illustrating it with an application example. FigureFigure 5, shows a recommend process for using the PSL/Traditional-v.2020. This example is of a fictional project, although it is representative of a real project.

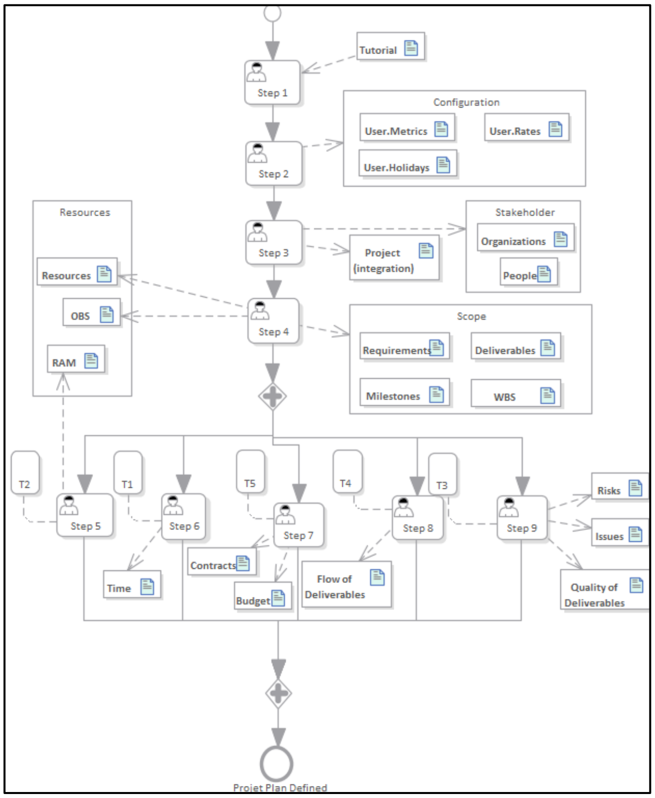
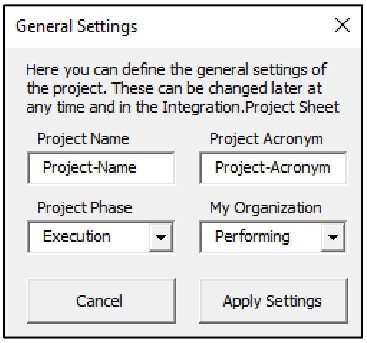


Figure 5. Traditional Planning Process (in BPMN)

Step 1: Read the Tutorial sheet (or other help documentation) to be aware, namely, that different color cells mean different things; for instance: red is for error messages; dark blue is for user data input; light blue is for automatically filled data; and grey cells means that they shall stay empty.

Step 2: Set the Configuration sheets by defining the holidays, user rates, user metrics and configure **Settings** (see Figure Figure 6).


General Settings

Here you can define the general settings of the project. These can be changed later at any time and in the Integration.Project Sheet

Project Name: Project-Name

Project Acronym: Project-Acronym

Project Phase: Execution

My Organization: Performing

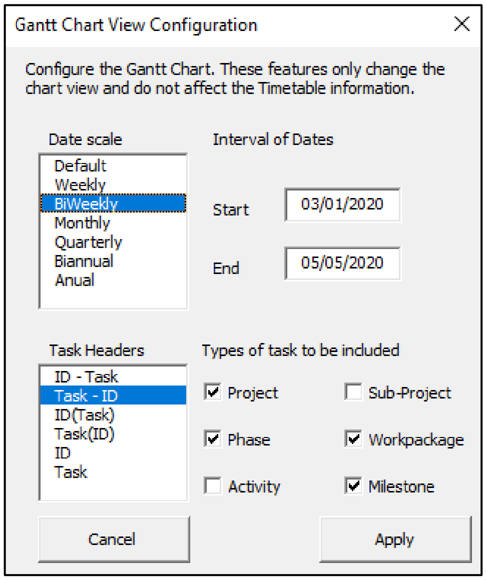
Figure 6. Settings form

Step 3: Identify the project Stakeholders by filling the Organizations and People sheets and **manage the Integration data** by filling the **Project sheet** with the general project information and then by accessing the **Charter sheet** to get the produced project charter report.

Step 4: Fill the scope group that consists of the **Requirements sheet, WBS sheet, Milestones sheet, and Deliverables sheet.** Once it is finished, the Resources sheet is the next to take into consideration. With the help of **"Import people"** the resources are now defined, and it is easy to view the OBS of the project.

Step 5: Define the relationship between the OBS and WBS, through the RAM. Press the **"Import from WBS and OBS" (T2)** that constructs the table. The legend box provides possible responsibilities and is movable.

Step 6: Define the schedule in the Time sheet. The first action is to **"Update from WBS and Milestone" (T1)** that inserts this information in the table. The second action is to add other activities, defined execution times, precedencies, allocate resources, and the rest is automatic. The third action is to **"Update Chart"** that updates the Gantt Chart with the new information and the fourth action is to customize the view (see Figure Figure 7).


Gantt Chart View Configuration

Configure the Gantt Chart. These features only change the chart view and do not affect the Timetable information.

Date scale: Default, Weekly, BiWeekly (selected), Monthly, Quarterly, Biannual, Anual

Interval of Dates

Start: 03/01/2020

End: 05/05/2020

Task Headers: ID - Task, Task - ID (selected), ID(Task), Task(ID), ID, Task

Types of task to be included

☑ Project ☐ Sub-Project

☑ Phase ☑ Workpackage

☐ Activity ☑ Milestone

Figure 7. View configuration form

Step 7: Define the project budget in the budget sheet. First **"Import from Resources" (T5)**, second define the resources rate, and third define the resources usage. Then, **define the project contracts** in the contracts sheet.

Step 8: Define the flow of deliverables and meetings, using in both cases the **"Import deliverables" (T4)** before starting to fill the table.

Step 9: Identify the issues, the quality of the deliverables with the help of **"Update deliverables" (T3)** option to import the deliverables, and **the risks**. At the end of this step the project plan can be complete with the majority of the KAs considered.

6. PSL AGILE TEMPLATE

The PSL/Agile-v.2020 is the new template that supports the application of agile best practices based on Scrum and

Kanban frameworks, utilizing Excel Formulas, VBA and RibbonX like the PSL/Traditional-v.2020.

6.1. Alignments

This PSL/Agile v.2020 is organized in 3 groups of sheets: Project, Scrum and Kanban (see Figure Figure 3). The tool defines 7 dependencies between its sheets, as shown in Figure 8. These dependencies are classified in 2 categories: moderate and strong. A **moderate** dependency means the existence of automatic mechanisms or suggestions of data to insert. A **strong** dependency means the existence of one or more macros options that guarantee that dependency. These dependencies are the following, represented as Di:

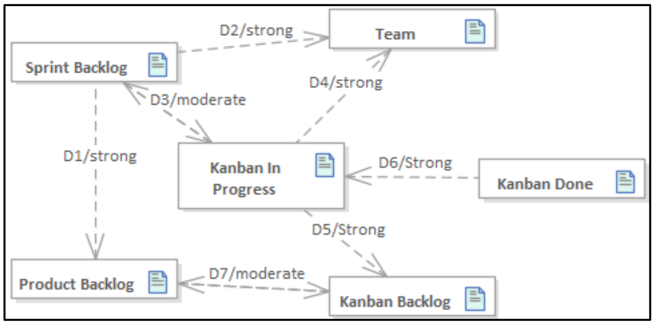


Figure 8. Dependencies in the scope of PSL/Agile-v.2020 (in BPMN)

D1 represents the dependencies of the Sprint Backlog tasks with the user stories defined in the Product Backlog.

D2 represents the dependency between a sprint task and the Team since only a team member can be allocated to a task.

D3 represents the dependency between the Sprint Backlog tasks and Kanban In Progress tasks. Since logically they represent the same, they can be transformed into one another.

D4 represents the same dependency as D2 but for Kanban tasks.

D5 represents the dependency between the Kanban In Progress tasks and Kanban Backlog tasks since all the Kanban In Progress were defined in the Kanban Backlog.

D6 represents the same dependency as D5 but between the Kanban Done tasks and Kanban In Progress tasks.

D7 represents the same dependency as D3 but between the Product Backlog and Kanban Backlog.

6.2. Illustrative Example and Transformations

The best way to demonstrate some PSL/Agile-v.2020 features, namely its distinctive aspects, related alignments, and transformations, is by illustrating it with an application example. Figure 9 shows a recommend process for using the PSL/Agile-v.2020. This example is of a fictional project, although it is representative of a real project.

Step 1: Read the Tutorial sheet (or other help documentation) to be aware, namely, that different color cells mean different things; for instance, red is for error messages; dark blue is for user data input; light blue is for automatically filled data; and grey cells means that they shall stay empty.

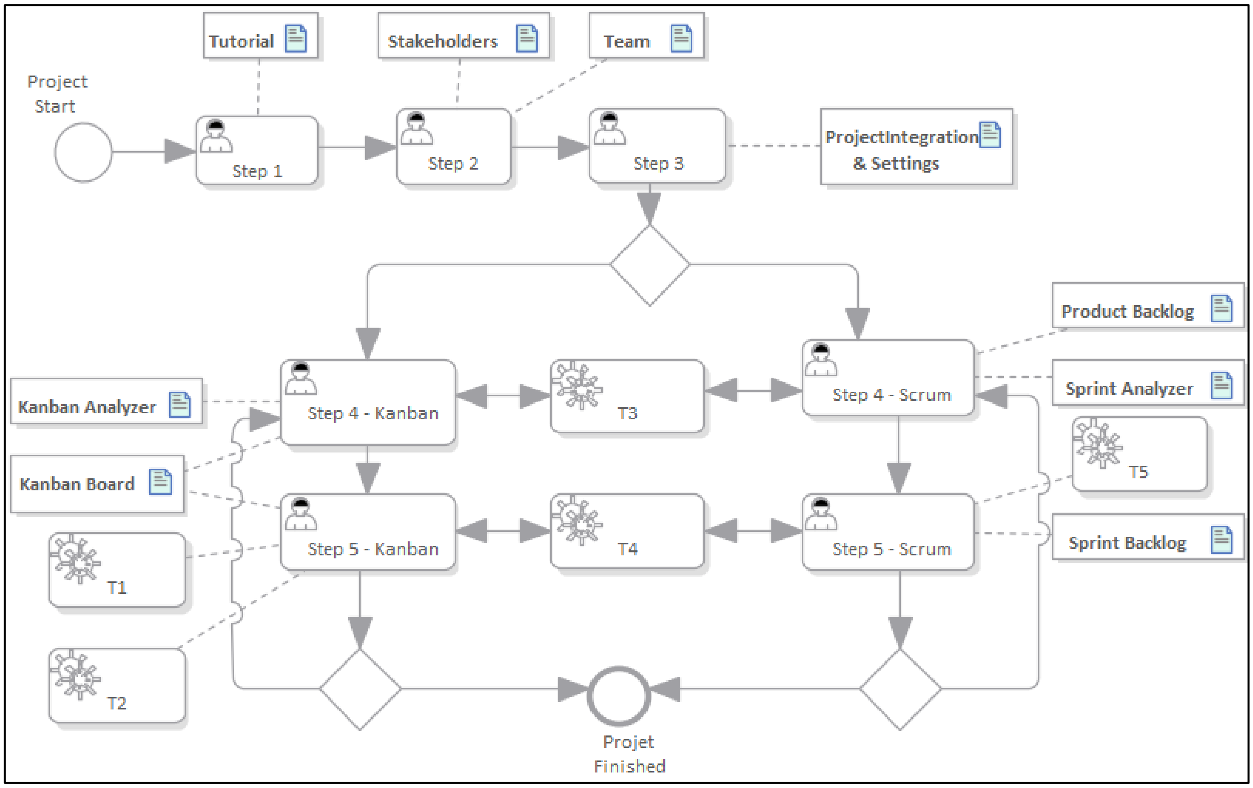


Figure 9. Agile project execution process (in BPMN)

Step 2: Identify the stakeholders and team by filling the StakeholdersOrganization, StakeholdersPeople and then Team sheets.

Step 3: Manage the Integration data by filling the ProjectIntegration sheet with the general project information and define what agile framework will be used by clicking the **Settings**.

Step 4 – Scrum: Set the Product Backlog in the Product Backlog by defining user stories, priorities, and effort. In case this is not the first cycle, check SprintAnalyzer sheet for KPIs. It is possible to change to the Kanban framework by going to the Step 4 – Kanban and using **“Update from the Product Backlog” (T3)** filling the Kanban Backlog table with this Product Backlog information.

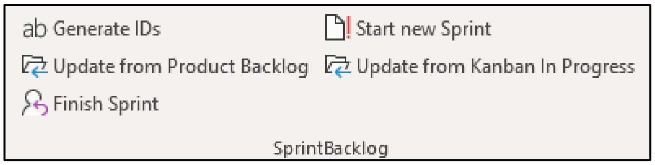


Figure 10. Sprint ribbon options

Step 5 – Scrum: Start a new Sprint by clicking on the **“Start new Sprint”** option and defining the sprint attributes. Then click the **“Update from Product Backlog” (T5)** to insert the user stories on the sprint and then **define tasks.** When the sprint is finished, use **“Finish Sprint”** to store the data in the SprintHistory sheet and update both SprintAnalyzer and Product Backlog sheets. Now it is possible to cycle back to Step 4 or finish the project. At any given time, it is possible to change to the Kanban framework by going to the Kanban sheet and using **“Update from Sprint Backlog” (T4)** filling the Kanban In Progress table with the sprint information.

Step 4 – Kanban: Define the project tasks in the Kanban Backlog table from the Kanban sheet. Double-clicking the “**description**” column helps doing the task. In case this is the not the first cycle, check Kanban Analyzer sheet for KPIs. It is possible to change to the Scrum framework by going to the Step 4 – Scrum and using **“Update from the Kanban Backlog” (T3)** filling the Product Backlog with this Kanban Backlog table information.

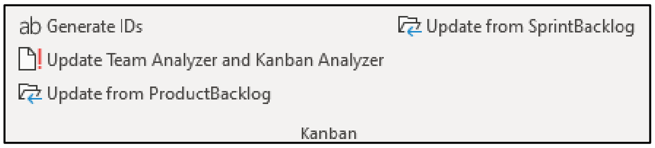

ab Generate IDs
Update from SprintBacklog
Update Team Analyzer and Kanban Analyzer
Update from ProductBacklog
Kanban

Figure 11. Kanban ribbon options

Step 5 – Kanban: Go to the Kanban sheet and start a task execution by clicking on the “**Move >**” **(T1)** column of the Kanban Backlog table. This moves the task to the Kanban In Progress table and records the start date. Then allocate people to each task. Once a task is done click the “**Move >**” **(T1)** column to move the task into the Kanban Done table. When all tasks are complete, and a cycle is finished “**Update Team Analyzer and Kanban Analyzer**” **(T2)** updates both analyzers with the relevant information from each task. Now it is possible to either finish the project or start a new cycle on Step 4. At any given time, it is possible to change to the Scrum framework by going to the SprintBacklog sheet and using **“Update from Kanban in Progress” (T4)** filling the sprint with this Kanban In Progress table information**.**

7. EVALUATION

During this work we evaluated the new and former versions of the PSL templates. The evaluation was done using two case studies: DOCIST and this MSc Thesis.

7.1.1. *Case study A: DOCIST on PSL-v.2019 & PSL-v.2020*

DOCIST is a fictional project designed by the Thesis adviser that was also used in the Information System Project Management course [**Error! Reference source not found.**]. The PSL-v.2019 was tested initially with this project. The tool was more appropriate to the task than using multiple tools, however there were few problems with macros, and was difficult to apply an agile framework. The same project was tested with the PSL-v.2020. This was a smoother experience because the template does not have any detectable bugs and contains an improved ribbon with more features that also provides better navigation features.

7.1.2. *Case study B: My MSc Thesis*

The second case study is My MSc Thesis project. The project uses a Scrum framework so naturally it was used to test the agile sheets of the PSL-v.2019 and test the new Agile Template. Testing the PSL-v.2019 lead to numerous problems while testing the new version was almost effortless. However, the template was tested once the project was almost finished and it should be tested during a project execution.

7.2. **Comparison with PM tools**

The evaluation phase provided the needed results to compare the new tool version to its older version and with other PM tools (see table Table **3** and Table **4**). This comparison is made in terms of the following aspects: KAs support for traditional tools, Scrum and Kanban support for agile tools, customization, report generation, user friendliness and collaboration. The subjective classification metric is in the following scale: - in the absence of any features, + for minimal applicability, ++ competent, +++ high-quality features.

For users looking for support to the majority of the PMBOK **KAs**, PSL/Traditional-v.2020 is the only one capable of doing so. PSL-v.2019 supports 8 KAs, while ProWorkFlow, Confluence, Vertex42 and Smartsheet support 5 to 7 KAs. MS Project and Office Timeline do not go beyond 2 to 3 KAs.

For users looking for **Scrum** support, Azure DevOps and JIRA Agile offer complex dependency features between sprint backlog and product backlog with multiple dashboard options. For Kanban support, PSL/Agile-v2020 provides the most intuitive Kanban Board. SmartSheet can be customized to some extent to replicate a Kanban Board while Azure DevOps and JIRA agile focus more on the board concept making the Kanban Board more complex than it should be.

Most of the tools provide **customizable** options for treating data and changes to its presentation, but confluence is the best solution for users looking for a clean start with no restrictions to the organization of the data. Smartsheet features to customize the view for different purposes with the same data are of high-quality.

Focusing on the **user friendliness**, PSL/Traditional-v.2020 and PSL/Agile-v.2020 are some of the best solutions because they provide a complete template to a very popular environment and an easy to learn and understand solution. Smartsheet, MS Project, ProWorkFlow and JIRA agile are tools with a more overwhelming first impression and require some adaptation and practice before being able to utilize the full potential of the tools. Confluence is the most complicated tool to learn because it starts from a blank state and Azure DevOps is a tool oriented towards IT making it difficult to use for other types of teams.

Office Timeline and ProWorkFlow are two good tools in what concerns **generate reports**. Office Timeline offers vast and quality options for presentations and ProWorkFlow provides a list of complex and detailed reports for any preference. For agile methodologies JIRA agile offers standard reports options that comprise a wide range of reporting applications.

Confluence is the best **collaborative** tool, it focuses on communication and making sure every member involved is part of the project and actively participates. It also has various interoperability features with many popular tools, like JIRA, Trello, and others. Smartsheet distinguishes itself by allowing to create workflows triggered by data changes, with custom notifications. Focusing on the agile tools, JIRA Agile offers similar features to Confluence making sure every member involved is part of the project and knows what is being developed. It also has interoperability features with MS Word, MS Excel, SQL, and others.

Finally, focusing on just the PMet, the tool has its quality features but also has a few shortcomings. Currently it does not have any collaborative mechanisms, besides the ones provides by Excel and Office 365, making it difficult for more than one person to use it simultaneously. Another aspect is the difficulty to use PMet in larger projects. While the tool is prepared for any size of information having very large data sheets could become difficult to manage.

Table 3. Comparison of tools based on Traditional Approaches

		KAs Support	**Custom-ization**	**User Friend-liness**	**Report Generat-ion**	**Collabor-ation**
Tools	SmartSheet	++	+++	++	+	+++
	MS Project	+		++	++	-
	ProWorkFlow	++	+	++	+++	+
	Confluence	++	+++	++	++	+++
	PSL/Tradition-al-v.2020	**+++**	**++**	**+++**	**++**	**-**
Templates	PSL-v.2019	+++	+	++	+	-
	Office Timeline	+	+++	+	+++	++
	Vertex42	++	++	+++	++	-
	Gantt Excel	+	+++	+++	++	-

Table 4. Comparison of tools based on Agile Approaches

		Scrum Support	**Kanban Support**	**Custom-ization**	**User Friend-liness**	**Report Genera-tion**	**Collaborat-ion**
Tools	Smart Sheet		++	+++	+++	+	+++
	Azure DevOps	+++	+	++	+	+	+++
	Jira Agile	+++	++	+	++	+++	+++
	Asana	+	++	++	+	+	++
	Quire		+	+	+++	+	++
	PSL/Agile-v.2020	**++**	**+++**	**++**	**+++**	**++**	**-**
Templates	PSL-v.2019	+		+	++	+	-
	Vertex 42		++	+	+++	+	-

8. CONCLUSION

The ongoing technological evolution has delivered several tools and techniques for project managers adopt and combine both traditional and agile approaches [28]. Most PM traditional tools provide high-quality features for some knowledge areas, but not for all of them while agile tools provide complex environments to inexperienced users. PSL-v.2020 distinguishes from these options by providing an effective, simple to use and to learn solution.

PSL-v.2020 is developed on top of the MS- Excel platform with VBA features providing a distinct solution for any interested user. The Traditional version comprises all KAs of a project according to a predictive approach as defined for example by PMBOK [24]. On the other hand, the agile version supports Scrum and Kanban frameworks while facilitating the transition between traditional and agile methodologies. This tool provides several mechanisms that remove repetitive work and guarantee consistency between the involved data.

This tool experiences have showed preliminary positive outlooks, but it is still needed to develop further evaluation studies regarding the relevance of this tool, to analyze the possibility of integration in a Web context (e.g., providing a smooth integration with Office 365 and Google Sheets platforms) to facilitate its continuous development and maintenance, and develop more import and export features particularly for Office tools like MS Excel, MS Project, and other PM tools.

9. REFERENCES

1. ITLingo, accessed October 2020, http://itbox.inesc-id.pt/ITLingo
2. PSLingo, accessed October 2020, http://itbox.inesc-id.pt/ITLingo/PSLingo
3. Daniel J. Fernandez & John D. Fernandez (2008) Agile Project Management — Agilism versus Traditional Approaches, Journal of Computer Information Systems, 49:2,10-17
4. Schwaber, K., Sutherland, J.: The Scrum Guide. Scrum. Org and ScrumInc, p. 17 (2017) http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf
5. Kniberg, H. & Skarin, M., Kanban and Scrum – Making the Best of Both, 2010
6. Q-Day 2019 Conference, PM2 - Uma Metodologia de Gestão de Projetos para a Europa: comum, livre e aberta, attended September 18,2019
7. Kenton, W., Visual Basic for Applications (VBA), accessed October 2020, https://www.investopedia.com/terms/v/visual-basic-for-applications-vba.asp
8. Office Ribbon X, accessed November 19, 2019, https://github.com/fernandreu/office-ribbonx-editor
9. SoftwareKey Licensing System, accessed October 2020, https://www.softwarekey.com
10. Gantt Excel – Free Gantt Chart Excel Template, October 2020, 2019, http://www.ganttexcel.com
11. MS Project, accessed October 2020, https://products.office.com/en/project/project-and-portfolio-management-software
12. Confluence, Atlassian, accessed October 2020, https://www.atlassian.com/software/confluence
13. Vertex42 – Project Management Templates, accessed October 2020, https://www.vertex42.com/ExcelTemplates/excel-project-management.html
14. Smartsheet, accessed October 2020, https://www.smartsheet.com
15. Office Timeline – Using Excel for Project Management, accessed October 2020, https://www.officetimeline.com/project-management/excel
16. ProWorkFlow, accessed October 2020, https://www.proworkflow.com/
17. JIRA Agile, Atlassian, accessed October 2020, 2019, https://www.atlassian.com/software/jira/agile
18. Azure DevOps, accessed October 2020, 2019, https://azure.microsoft.com/en-us/services/devops
19. Asana, accessed October 2020, 2018, https://asana.com/
20. Quire, accessed October 2020, 2018 https://quire.io
21. A Reddy, The Scrumban [r] evolution: getting the most out of Agile, Scrum, and lean Kanban, 2015
22. Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum (Vol. 1). Upper Saddle River: Prentice Hall.
23. A Holzinger “Usability engineering methods for software developers” – Communications of the ACM, 2005
24. Project Management Institute, “A guide to Project Management Body of Knowledge”, 6th edition, 2017
25. PMI | Project Management Institute. Retrieved March 2020, from https://www.pmi.org/
26. International Project Management Association, Individual Competence Baseline for Project, Programmer & Portfolio Management, Version 4.0, 2015
27. ISO | International Organization of Standardization, ISO21500: Guidance on project management, 2013
28. Project Management Institute, Agile Practice Guide, 2017.
29. J. Venable, J. Pries-Heje, and R. Baskerville, “A comprehensive framework for evaluation in design science research,” in Design Science Research in Information Systems. Advances in Theory and Practice, K. Peffers, M. Rothenberger, and B. Kuechler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 423–438.
30. K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” J. Manage. Inf. Syst., vol. 24, no. 3, pp. 45–77, Dec. 2007. Available: http://dx.doi.org/10.2753/MIS0742-1222240302
31. Walkenbach, J., Microsoft Excel Bible - The Comprehensive Tutorial Resource, 2016
32. Silva Rosário Legas, J. P., Exploring Project Management Tools based on the ITLingo PSL language, Msc Thesis, Instituto Superior Técnico, 2019

MSc program in Information Systems and Computer Engineering and Engineering, https://fenix.tecnico.ulisboa.pt/cursos/meic-a