POLITECNICO DI TORINO

Master's Degree in Engineering and Management

Master's Degree Thesis

Applying the PM approach to the academic context, building a thesis projects management framework. The back-end development perspective

Supervisor

Prof. Paolo De Magistris

Candidate

Giorgio Spadaro

December 2021

Abstract

Project management refers to the application of methods, tools, techniques and skills to a project carried out through processes aligned in a systemic vision. It includes the integration of the various stages of the project life cycle, each stage should have specific results that should be regularly reviewed to meet the requirements of the sponsors, customers and other interested parties. The topic of this thesis aims to apply this approach to the academic context, to facilitate the management of projects and elaborates, which see students as protagonists, through the use of management software, such as the PLM system. In the field of information systems, the Product Life-Cycle Management, commonly called PLM, allows you to manage the life of a product, in our case a project, keeping track of the components, tasks and resources useful for its realization. In fact, in the business environment, PLM software can be seen both as an IT strategy and as a corporate strategy. The software at the center of this study and which will be used first in its standard form and then customized to make the user experience easier, is Odoo. The core of this research focuses mainly on the nature of the project data from a “back-end” point of view.

Table of Contents

List of Figures

Acronyms

PLM

Product Life-Cycle Management

ERP

Enterprise Resource Planning

UML

Unified Modeling Language

ORM

Object Relational Mapping)

UX

User Experience

UI

User Interface

WBS

Work-Breakdown Structure

BOM

Bill Of Materials

SWOT

Strenghts Weaknesses Opportunities Threats

PM

Project Management

Chapter 1

Introduction

1.1 Background

Nowadays, the concept of project management turns out to be more and more flexible and applicable to an ever larger number of segments and contexts. The need, therefore, to fix the fundamental concepts and schematize their form, is increasingly present. In this regard, the standard that regulates this form is ISO 21500, which provides a high-level description of the processes and concepts considered to be good practice in the field of project management. A project, in fact, consists of a group of processes, divided into coordinated and controlled tasks, with precise start and end dates, carried out to achieve the objectives of the project. Achieving project objectives requires obtaining deliverable outputs that must conform to specified requirements, also presenting multiple implementation constraints. The need that nowadays is increasingly evident, once regulated and established as previously mentioned, a standard that is a guide for project management, is the digitization of this management, that is to bring the voices and actors present in ISO 21500, within a management software that is used to coordinate process deadlines, resources and tasks. We will use a PLM to digitize this process. PLM (Product Life-Cycle Management) systems are software focused on managing the life of a product, therefore they mainly take into consideration resources, materials, processes and deadlines.

1.2 Thesis Purpose

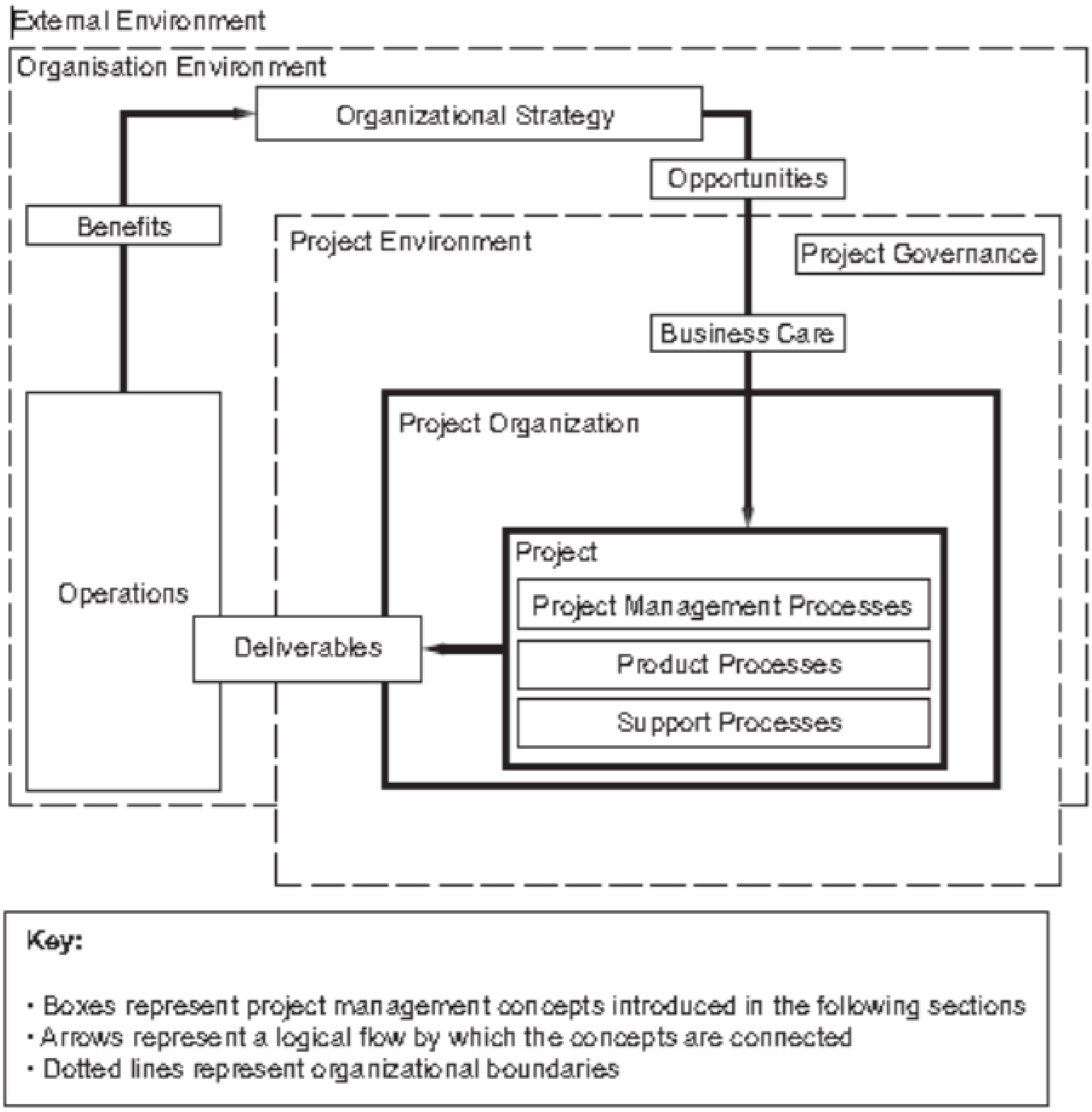

External Environment
Organisation Environment
Organizational Strategy
Opportunities
Benefits
Project Environment
Project Governance
Business Care
Project Organization
Project
Project Management Processes
Product Processes
Support Processes
Operations
Deliverables
Key:
• Boxes represent project management concepts introduced in the following sections
• Arrows represent a logical flow by which the concepts are connected
• Dotted lines represent organizational boundaries

Figure 1.1: Project Management schema

The goal of this paper is to use the logic of a PLM for project management. In particular, a project, in our use case the writing of a thesis or project paper, will be treated as a product, so resources, tasks, deadlines and delivery time estimates will always be present as attributes.

Figure 1.1 shows how project management concepts relate to each other: The organization's strategy identifies opportunities. The opportunities are evaluated, documented and a part of the selected opportunities are further developed in a specific 'case business' and then resolved into one or more projects that provide releasable outputs ('deliverables'). Some of these can be used to achieve appropriate benefits to the organization and these can be an input to implement and further develop the organization's strategy. Furthermore, following the model provided by the ISO 21500 standard, the focus of the work is shifted to the deliverables part, in particular we see how within a project we have three main items:

- Project Management Processes
- Product Processes
- Support Processes

Going to consider, as mentioned before, the development of a project such as the development of a product, the item that we will take into account and that will be the guide of this study will be the "Product Processes".

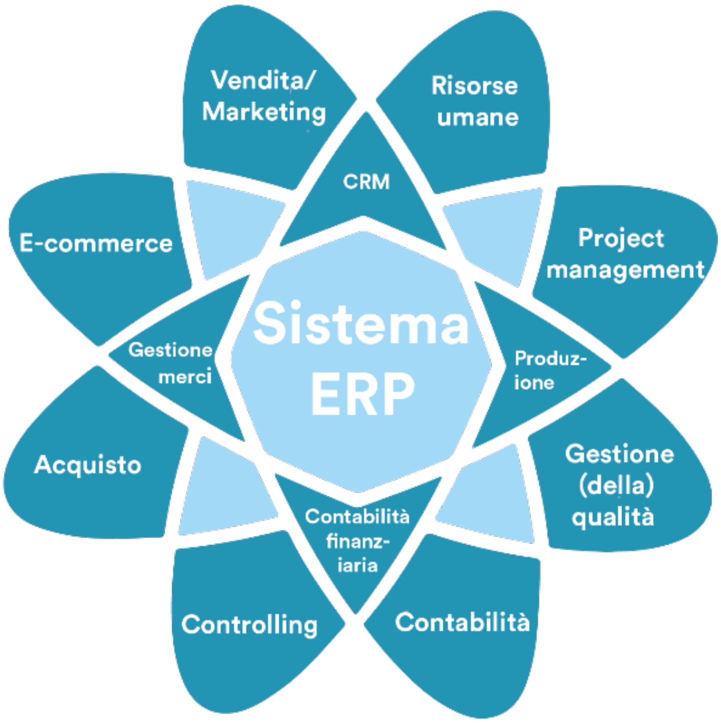


Figure 1.2: ERP System[1]

1.3 Context

The current context in which we are present presents a gap regarding a management software for academic use that encloses in itself effectiveness and simplicity of navigation. The state of the art in fact presents some competitors of large groups, too complex to be easily implemented and to be used as support tools for the management of an academic project, would in fact require a dedicated study. As for the ERP, the majority of the market, being oriented to the business use, previews services to payment and sure not open source, therefore with the code not visionabile. For this reason the need arises to research and develop a framework dedicated to project management for entry level users (beginner) and hence the purpose of this study.
Moreover, by structure, PLM is the software that is closest to the optimal management of a project. Dealing with the life cycle of a product in fact, we can replace features that are considered useful and maintain similar ones for both product and project, such as deliveries or resources that will deal with a single task.

1.4 Stakeholders

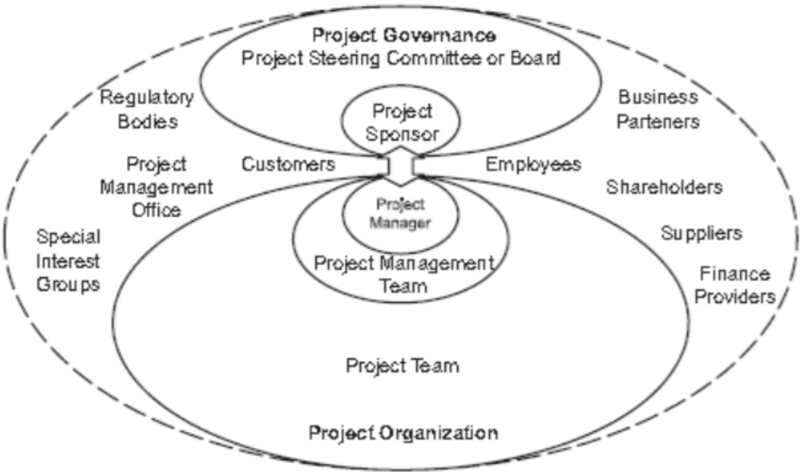


Figure 1.3: Project Stakeholders

By stakeholders we can mean people, groups or sectors of an organization that are in various ways impacted by a project or that may have some direct or indirect form of interest to the result or to some of the deliverable expected.
Not all are equal. Each has its own needs and expectations that must be met. It is the task of the project manager to define the list of parties involved and

the strategy he intends to adopt to manage them. It is important to identify the project stakeholders (internal and external) from the start-up stage so as not to run the risk that their late detection may lead to problems or even postponement of the project. When a person who has a role or a high power is involved too late, this results in serious trouble for the outcome of the project. Once identified, the project manager and the working group will have to collect the requirements that each of them is carrying along with their expectations. This step is an excellent opportunity to better understand their attitude towards the project (e.g. positive, supportive, resistant, indifferent or neutral) and identify a strategy to manage them[1].

In our specific case, the group of stakeholders is represented by the Professor, who requested for his students, the creation of a project management software to manage at the academic level the projects of the course and personal elaborations.

Chapter 2

Literature

2.1 ISO 21500

The ISO 21500 "Guidance on project management" standard. It is a project management guide that can be used by any type of organization, public, private or community, and for any type of project, regardless of complexity, size or duration. The standard presents concepts and processes considered to be good practices in project management. In some ISO 21500 countries (e.g. South Korea) it has already been chosen as the standard for Project Management to which all government tenders for projects should comply. Organizations generally set strategies based on their mission, vision, policies and factors external to the organization. Projects are often the means to achieve strategic objectives[2].

2.2 Project Management

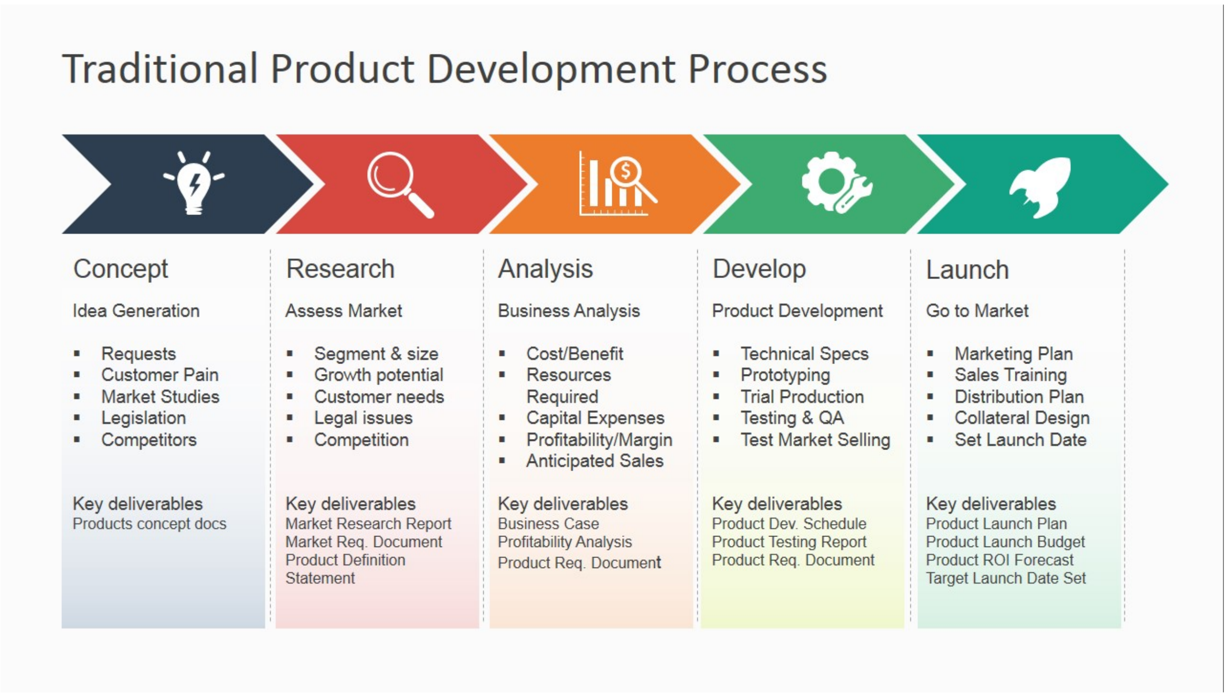
What project management is and what the work of project manager is in concrete is a question that has emerged more and more frequently in recent years. While its roots lie in business practices dating back more than a century, it is only in recent decades that project management has emerged as an essential component of any serious business development initiative. It has also become a key to success in a global economic environment where companies are constantly seeking a competitive advantage over their competitors. Unlike function or business unit managers, project managers do not only oversee individual areas of expertise. In fact, project managers must perform a more complex task: overseeing all aspects of a project, ensuring that it meets agreed objectives and time, budget and resource constraints. To be successful at work, project managers must become proficient in carrying out a whole range of activities and achieve certifications that qualify them for this challenging career. They must also demonstrate a mixture of analytical skills and management of organizational relationships. For those who can master its complexity, project management can prefigure a particularly interesting professional development[3] The main objective of Project Management is to anticipate dangers and possible problems that could hamper the success of a project. Planning, organizing and controlling activities, according to Project Management, are the main moves to achieve a project with the greatest possible success, minimizing the risk factor. In the management of a project, the organisation of a detailed plan leads to specific results. Formalising goals, without being vague with generic concepts such as "improving customer relationships", makes the plan work more concrete. " Reducing customer complaints by half" is - on the contrary - a good goal to formalize. Project Management teaches precisely this, a method to be more efficient and concrete. According to this discipline, the project team has the task of achieving a series of intermediate results in the pursuit of the final goal, because by proceeding gradually it will be easier to achieve the desired result. Contrary to what was established by Project Management, a work planning without intermediate stages is not able to develop skills or insights along the way, the same ones that allow the working group to progress in a productive way. [4]

Figure 2.1: Project Manager[5]

2.2.1 Product Processes

The product process is part following the outline of figure 1.1, of the concept of plan, that is that one that rigurada the management in an within to operating level. The product process analyzes all the variants that follow the development of

the product from a design point of view: the resources, milestones, activities and sub-activities and the performance of the processes are then analyzed. The role of the project manager in this area is certainly to ensure a linear development so as not to have problems with product quality and be punctual in delivery.
For this reason, in fact, risks are also estimated that can weigh not only on costs, but on delivery, and therefore on possible delays. From the manufacturing point of view, however, the product processes focus more on the detailed performance of the manufacturing processes and on the various Boms that follow the development in all its phases. Depending on the development maturity in which you are, in fact, you will have a type of BOM: from the engineering BOM (EBOM) to the integrated BOM (IBOM). You will then need to distinguish the tools used for product management and its development.



Traditional Product Development Process

Concept

Idea Generation

- Requests
- Customer Pain
- Market Studies
- Legislation
- Competitors

Key deliverables
Products concept docs

Research

Assess Market

- Segment & size
- Growth potential
- Customer needs
- Legal issues
- Competition

Key deliverables
Market Research Report
Market Req. Document
Product Definition Statement

Analysis

Business Analysis

- Cost/Benefit
- Resources Required
- Capital Expenses
- Profitability/Margin
- Anticipated Sales

Key deliverables
Business Case
Profitability Analysis
Product Req. Document

Develop

Product Development

- Technical Specs
- Prototyping
- Trial Production
- Testing & QA
- Test Market Selling

Key deliverables
Product Dev. Schedule
Product Testing Report
Product Req. Document

Launch

Go to Market

- Marketing Plan
- Sales Training
- Distribution Plan
- Collateral Design
- Set Launch Date

Key deliverables
Product Launch Plan
Product Launch Budget
Product ROI Forecast
Target Launch Date Set

Figure 2.2: Product Process[6]

2.3 BOM vs WBS

In this section, the main theoretical topic on which this study will focus and from which the to-be models that will be obtained will be taken out: the characteristics of the BOM and the WBS.

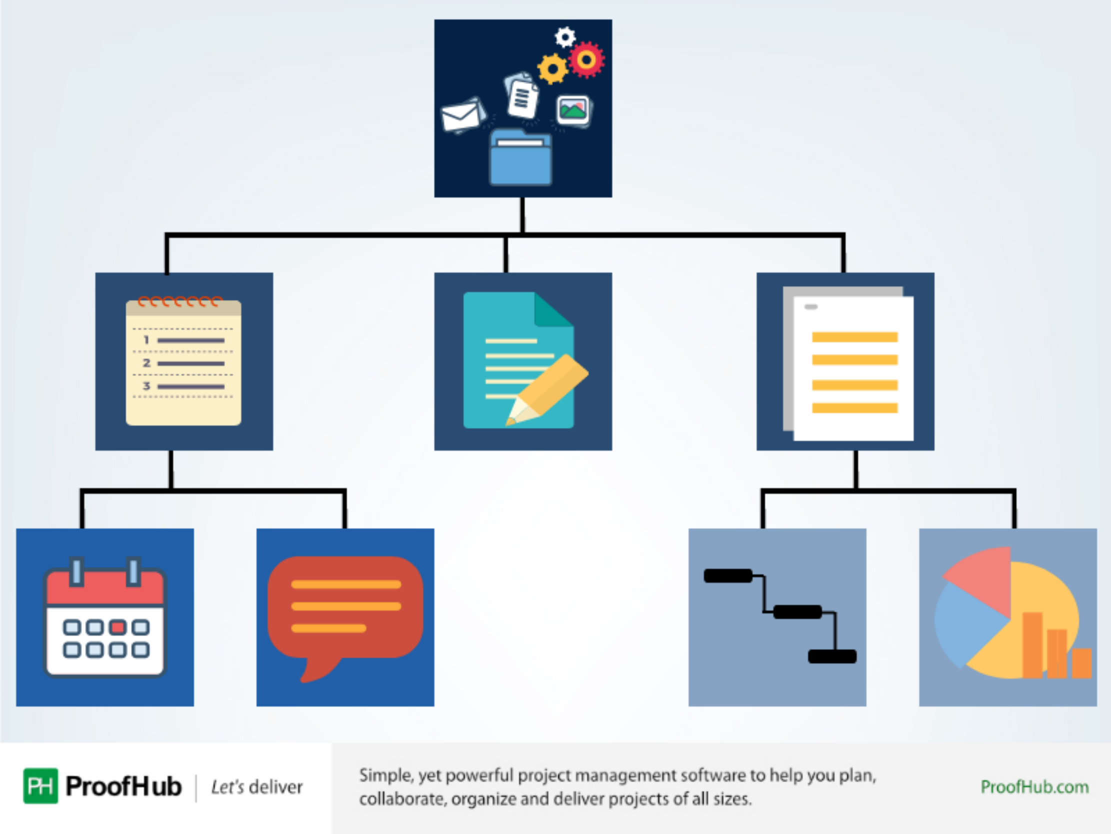

PH ProofHub
Let's deliver
Simple, yet powerful project management software to help you plan, collaborate, organize and deliver projects of all sizes.
ProofHub.com

Figure 2.3: Generic Structure[7]

2.3.1 BOM

The basic list is a document that highlights the list and the requirements of the elements necessary for the production of a product (raw materials, components and semi-finished products). Its purpose is primarily linked to the supply processes necessary to supply the production departments. It is also important to define a first measurement of the cost of a product.[8] The BOM is the hierarchical list of all assemblies, components, semi-finished products and raw materials needed to produce a product. This term is used in industrial sectors characterized by a "part" production, in which the various components that make up the finished product are made separately and (sometimes) at different times and then assembled.

In sectors where production is carried out "by process" (such as food, chemicals and pharmaceuticals), characterized by the fact that in the finished product it is no longer possible to distinguish or isolate the components used to make it, the separate basis is also called "recipe" or "formula".

A bill of materials, however, is not a simple list of components, but is organized in a hierarchical manner and is represented through a tree scheme that sees the final product at the head and gradually in the lower levels all the components used to derive it. By convention the finished product is at level zero, the direct components are at level one and so on. [9]

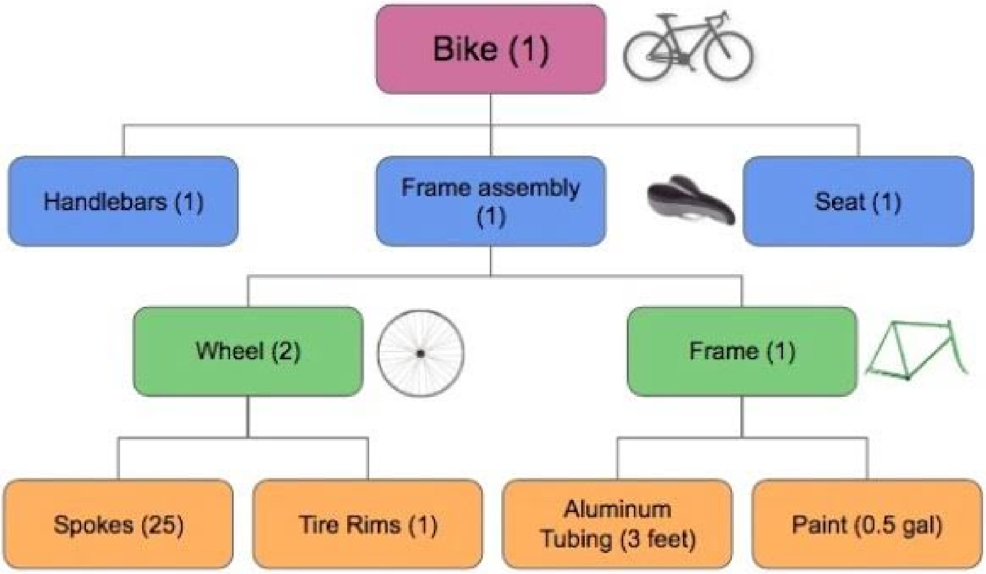


Figure 2.4: View BOM tree structure[10]

2.3.2 WBS

The WBS is a very important diagram for project managers that, as in any project that respects, orient their structure towards the achievement of a result and the achievement of objectives, a vital condition for an organization.

A correct definition that explains well what is a work breakdown structure speaks of it as a visual deconstruction, hierarchical and oriented to the final results of a project, which allows to subdivide the work into smaller activities, in order to make them more manageable and accessible to all those who are involved in the project management team.

The official definition of the Project Management Institute of work breakdown structure is that of hierarchical decomposition oriented to the final results of the work that must be performed by the project team.[11]

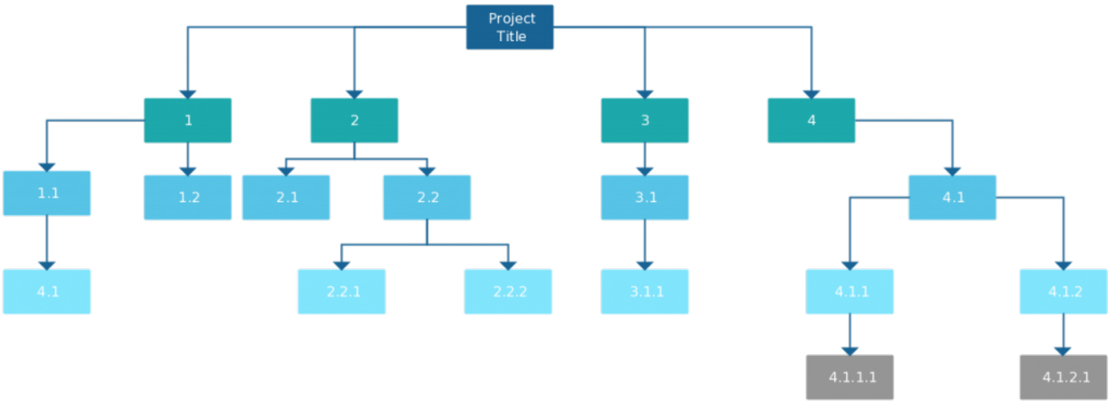


Figure 2.5: View BOM graph[12]

The work breakdown structure is developed to harmonize the parts, giving a common understanding of which is the aim of the plan, and establishing, as said, the hierarchy necessary to define the priority activities and the levels to reach.

Having a work breakdown structure within a project management is very important for all those who work in these mechanisms. Organizing the team's work in manageable sections is one of the priorities of this tool, which manages to simplify the planning, the actions to be implemented, the project to be delivered.[11] The difference between the work breakdown structure, project schedule, and project plan is a common confusion.

- Work breakdown structure describes the "what" of the project. It doesn't include timelines or resources.
- A project schedule is the "what", "when", and "who" of the project includes the project's deliverables as well as their deadlines and resource requirements.
- A project plan includes details on how the project will be executed, managed, and controlled, covering every aspect of the project.

The project plan is followed by the project schedule and then the work breakdown structure. [7]

2.4 Gaps Analysis

The differences that emerge are, in the light of what I said, basically two: BOM is process-oriented, that is, everything that concerns the internal process, so we go to detail production costs, materials, processing and resources.
The WBS is deliverable-oriented, therefore oriented to the sponsors and the delivery

of the work, oriented to the final results of the work that must be carried out by the teams. It is clear that, although they have a similar structure, they are logically different, because they are oriented to different objectives, and how the characteristics of both vary. Common features such as duration, resources, tasks and deadlines are the same that you will change and manage in this elaborate. The other features in the to-be model will not be considered.

Chapter 3

Project Management Framework Design

3.1 Research Question

As the main reasearch question is to understand the limitations of the software and what kind of requirements are to be implemented and which ones are offered by the basic version. Going to deepen the objectives, surely as the first goal you want to understand the model to-be that you want to get at the end of this study, what are its features and functionality. Project management now works a lot with agile projects, breaking down projects into smaller steps and detailing tasks and resources. Each step therefore has its own goal to achieve: in the first step that will be illustrated in Chapter 4, the analysis of the code includes the search for modules and the understanding of the structure. Once this is done, step 2 provides to identify, following the model that you want to cheat, the parts of interest code that deal with everything about the product. As a last step, the documentation will be produced to provide support for the customizations required, then for the processes of software development and deployment.
See how each step is connected to the previous one, a logical sequence is followed and each step is strictly dependent on the previous one. Also the structure of this work has a logical structure: as you can see, chapter 1 has the task of introducing the interlocutor to the subject and chapter 2 instead of fully immgergerlo in the subject through the paragraphs of a bibliographic nature. Chapter 3 illustrates the method and analysis tools used to establish the research points to be carried out in Chapter 4.
Finally, the last chapter has the task of drawing conclusions, then take-aways that will emerge from the analysis carried out.

3.2 Methodology

This elaboration concentrates the heart of its study and the output that it will produce in chapter 4. The chosen method to effetuare the analysis of a software that you do not know and of which you must look for the functionalities, is the reverse engineering.
The reverse engineering of the code allows programmers to reverse the processes of development and production of a software and then to get a valuable look behind the scenes of a program. The deconstruction and reverse engineering of a software allows you to extract the source code of an application. Once you get the code, the software can be read by experts as an open book. At this point, programmers and developers will be able to understand, rewrite or rebuild the architecture of a program, its operation and its internal structures. The understanding of the processes of the software, obtained thanks to reverse engineering, also allows to eliminate any bugs. In the software sector, reverse engineering is mainly used for the development of new products, troubleshooting or analysis of competing products.[13]
Reverse engineering is necessary both for the study of industrial production products and for the reconstruction of software. Software reconstruction involves one of the following three processes[13] :

- Recovery of Software Source Code
- Understanding the rules of a communication protocol
- Creating a template

In this case study, the recovery of the source code, being an open source software, is not expensive and having a model already well outlined, it is expected that the bottleneck of the analysis phase is the second point, that is to go to understand the methods and their workflow starting from the view of the software to the source of the code.

3.3 Unified Modeling Language - UML

Before talking about the framework from a more technical point of view, let's analyze , through a Use Case Diagram, the high-level workflow, that is, how the user will be able to "move" within the software. We will then use a SWOT Analysis to identify weaknesses and strengths, so we can minimize threats and maximize opportunities.

3.3.1 Use Case Diagram

The Use Case Diagrams are diagrams dedicated to the description of the functions or services offered by a system, as they are perceived and used by the actors who interact with the system itself. They are mainly used in the context of the Use Case View (view of use cases) of a model, and in this case they can be considered as a tool to represent the functional requirements of a system. However, the use of Ucds in other contexts can be assumed; during design, for example, they could be used to model the services offered by a given module or subsystem to other modules or subsystems. In many UML-based software development models, the Use Case View and the Use Case Diagram it contains represent the most important view, around which all other software life cycle activities develop

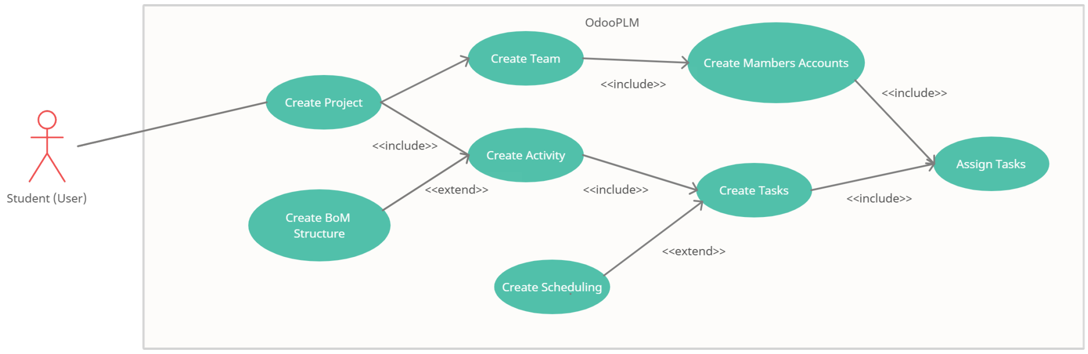


Figure 3.1: Use Case Diagram

3.3.2 SWOT Analysis

Swot analysis is one of the strategic planning tools through which companies can previously take into account the strengths, weaknesses, opportunities and threats of a given project. Its best known representation is the graphic one, in the form of a matrix as the figure below.

Figure 3.2: Use Case Diagram[14]

As for the strenghts, Odoo is presented as a complete software, with a form nature makes it easily accessible, as well as having a powerful and efficient database management being an ORM software, topic that will be discussed later.
The strenghts become opportunities, precisely analyzing them one by one we will have:

- Modulistic structure: gives us the possibility to interweave any type of module that does not provide the software in the basic version
- ORM: ORM software automatically manages the database, the point will be deepened in chapter 4
- ERP Complete: as ERP you don't need to integrate other modules

The weaknesses unfortunately are not few, in fact as first difficulty that has been found is the lack of documentation and reporting. In fact this weakness could threaten the system by driving away users who fail to have a sliding UX as a result, or programmers who fail to read the code or, To put it better, they would waste too much time deciphering and bringing out the functions of interest to them. The aim of this thesis is also to start providing and producing documents in order to guide the next steps regarding software development.

Another weakness is the instability of architecture. Perhaps due to the stratification of the continuous version updates, it is clear that Odoo has a complex architecture to outline and find. In fact, this weakness could become an important threat, that is, it could not be able to show all the features present or maybe go out of scope with respect to the target of the various modules. Simplifying the architecture would favor not only the UX, but also the UI, making the user access all the functions and showing them quickly and clearly.

Chapter 4

Project Management Framework Core

4.1 Odoo

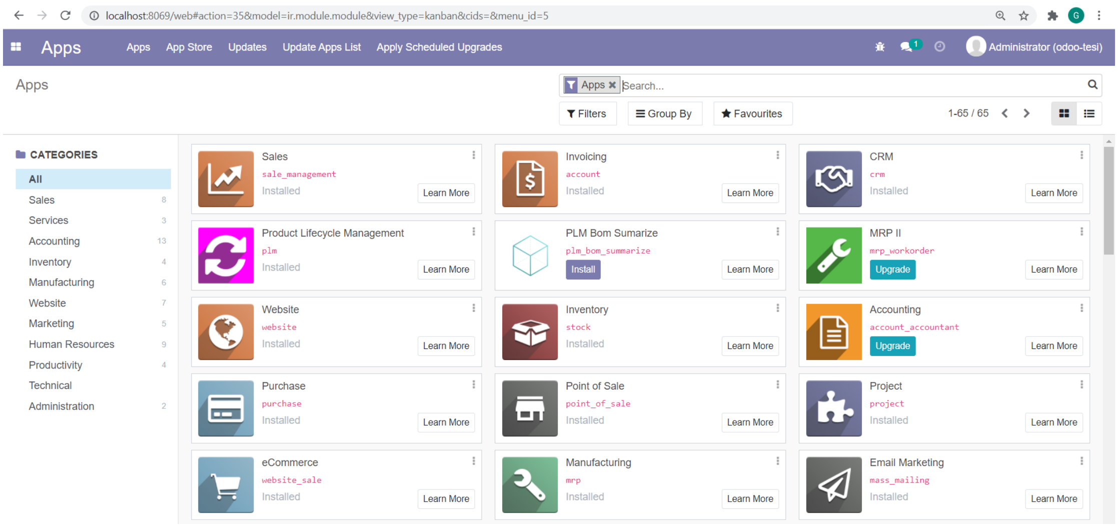
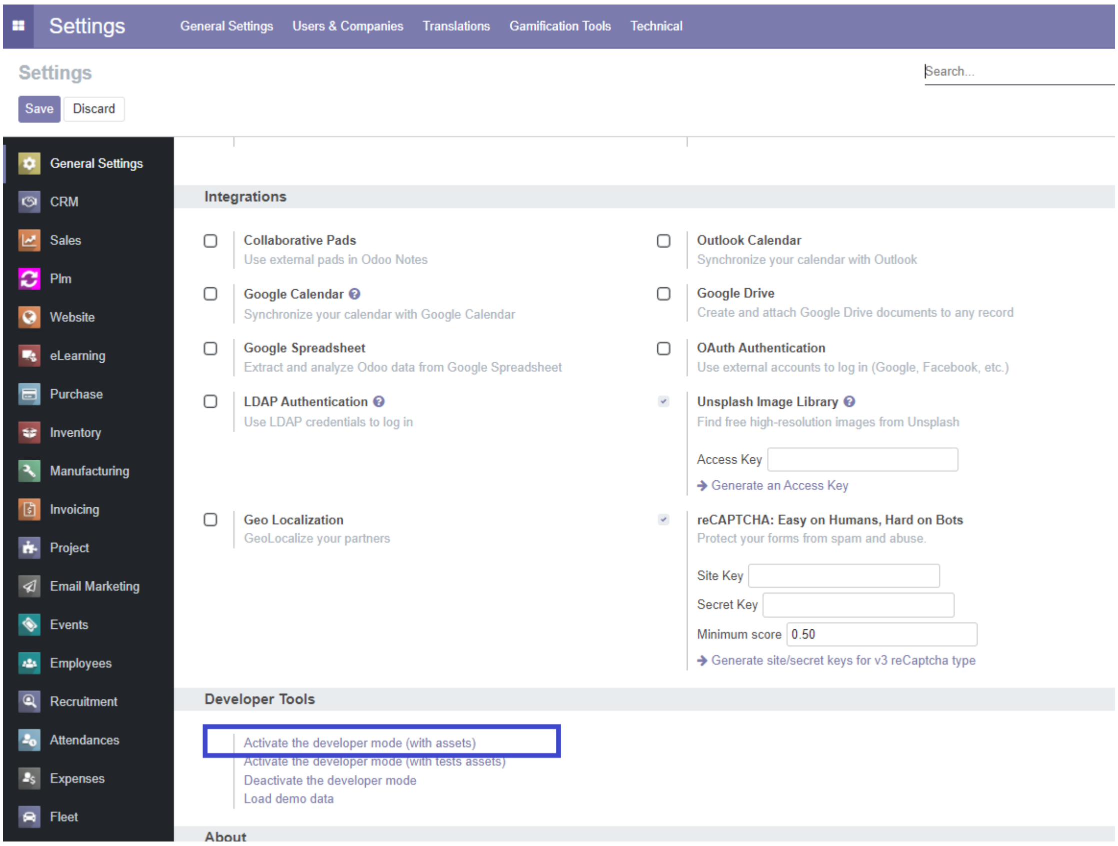


Figure 4.1: Odoo14 apps

Odoo is a modular business management software. Each module meets the digitisation needs of a functional area of the enterprise. The modules are selected and customized according to the needs of the company and then installed on a central computer called server, which is the only key computer whose management must be entrusted to specialized technicians. To work with Odoo does not need to install any software on your PC, but just a normal program for internet browsing on a computer connected to the server, logging in via username and password entry. In our case the server will be the "localhost" with port 8069. Although being mainly an ERP (Enterprise Resource Planning), its open source nature and its modular structure, as mentioned above, allows us to install new modules, such as OdooPLM that we will see later and that will be the core of our study, and be able to customize them by changing their code by entering into "developer" mode.[**odoo**]



Integrations

□ Collaborative Pads
Use external pads in Odoo Notes

□ Outlook Calendar
Synchronize your calendar with Outlook

□ Google Calendar
Synchronize your calendar with Google Calendar

□ Google Drive
Create and attach Google Drive documents to any record

□ Google Spreadsheet
Extract and analyze Odoo data from Google Spreadsheet

□ OAuth Authentication
Use external accounts to log in (Google, Facebook, etc.)

□ LDAP Authentication
Use LDAP credentials to log in

☑ Unsplash Image Library
Find free high-resolution images from Unsplash
Access Key
→ Generate an Access Key

□ Geo Localization
GeoLocalize your partners

☑ reCAPTCHA: Easy on Humans, Hard on Bots
Protect your forms from spam and abuse.
Site Key
Secret Key
Minimum score 0.50
→ Generate site/secret keys for v3 reCaptcha type

Developer Tools

Activate the developer mode (with assets)
Activate the developer mode (with tests assets)
Deactivate the developer mode
Load demo data

About

Figure 4.2: Odoo14 settings

4.1.1 Odoo Architecture

Odoo uses Postgresql as a database. Postgresql, also called Postgres, is an open source, free object relational database system (ORDBMS). Developed by Michael Stonebraker in 1986, it is the evolution of the research prototype POSTGRES DBMS.

The main features of Postgres are reliability, data integrity, functionality and extensibility, in addition to its open source community that manages, updates and develops performance and innovative solutions.[15]

Figure 4.3: PostgreSQL's Logo [15]

The software architecture is divided mainly into two main parts: client and server. As we can say in all web applications, the difference lies mainly in the management of the database. In fact, Odoo is an ORM (Object Relational Mapping) software. The ORM is a programming technique that serves to transform the data used in the object oriented programming language or program and the relational database (SQL type) as a persistent engine.

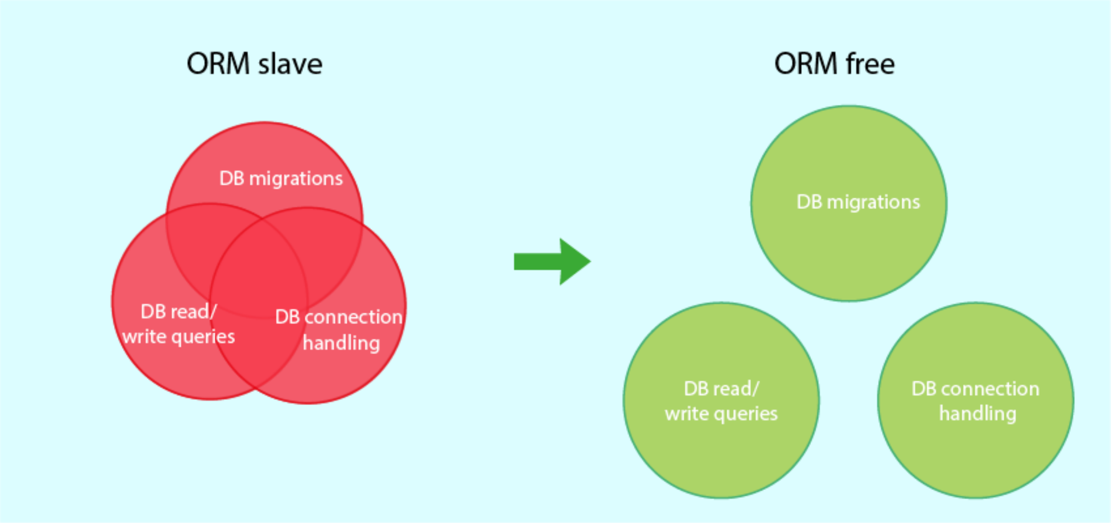


Figure 4.4: ORM Logic [16]

This will cause the program values to create an object-oriented virtual database to contain the necessary data. This creates a virtual database in which the values are located in the application you created in your code, and then they are linked to this database to give them persistence and register them in this simple way. This gives the recorded information persistence so that it can be stored, analysed, recorded or used afterwards[17].

ORMs are “easy” to use, but they come with a lot of incidental complexity that sooner or later will surface in our programs. In this case, easy is not the same as simple.

[16] Finally, ORMs are not a substitute for databases. If and when we need to use the full capabilities of the underlying database, the ORM falls short, resulting in the need to write raw database queries.[16]

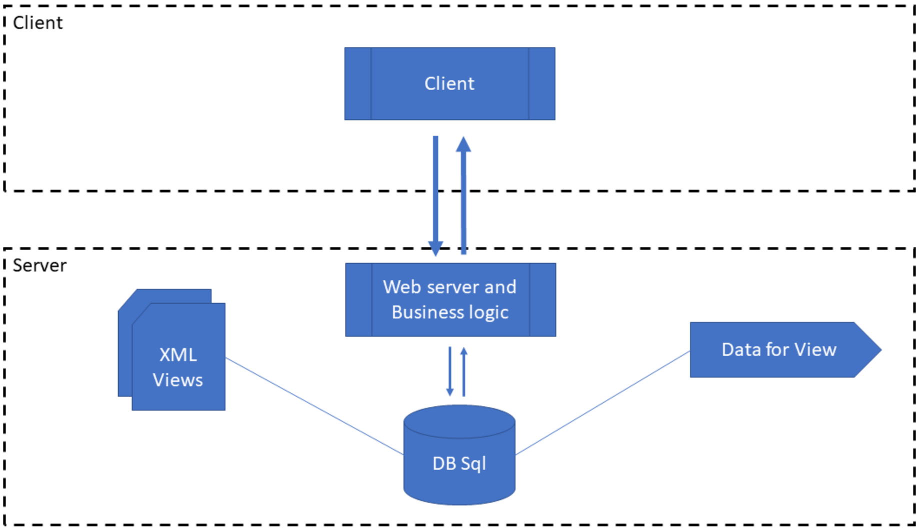
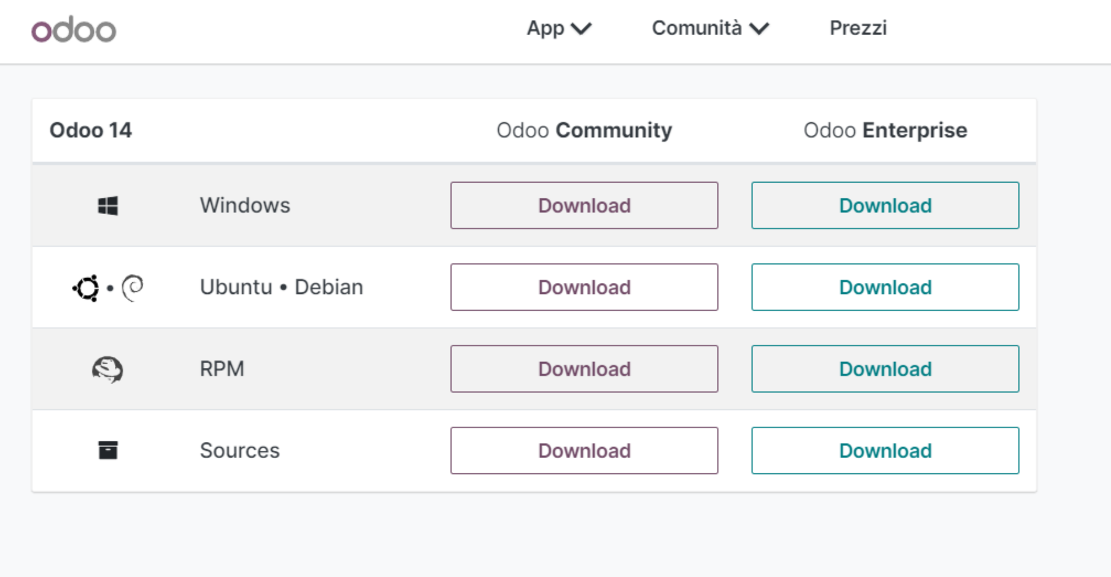


Figure 4.5: Odoo Architecture

4.1.2 Odoo set up

To get started, we simply download the software from the official website and follow the steps in the guide.

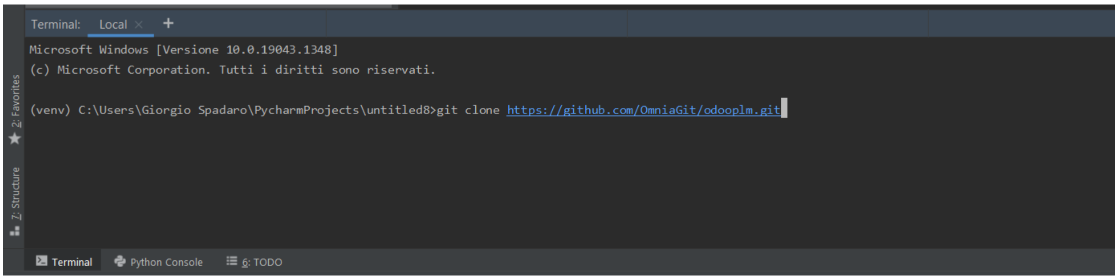


Odoo 14		Odoo **Community**	Odoo **Enterprise**
	Windows	Download	Download
	Ubuntu • Debian	Download	Download
	RPM	Download	Download
	Sources	Download	Download

Figure 4.6: Odoo14 Download

4.2 OdooPLM

As for Odooplm, let's clone the repository on the site Odoo Omnia Solution, partner of Odoo, and using the command "clone", we clone it in our IDE.



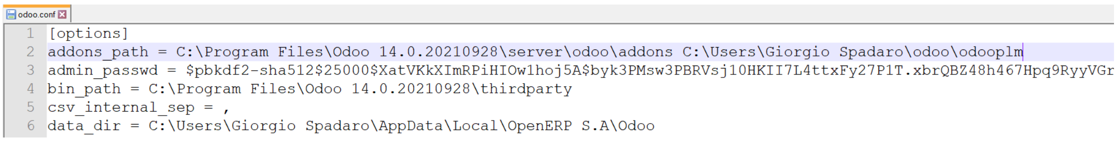
```
Microsoft Windows [Versione 10.0.19043.1348]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

(venv) C:\Users\Giorgio Spadaro\PycharmProjects\untitled8>git clone https://github.com/OmniaGit/odooplm.git
```

Figure 4.7: Odoo14 config file

4.2.1 OdooPLM integration

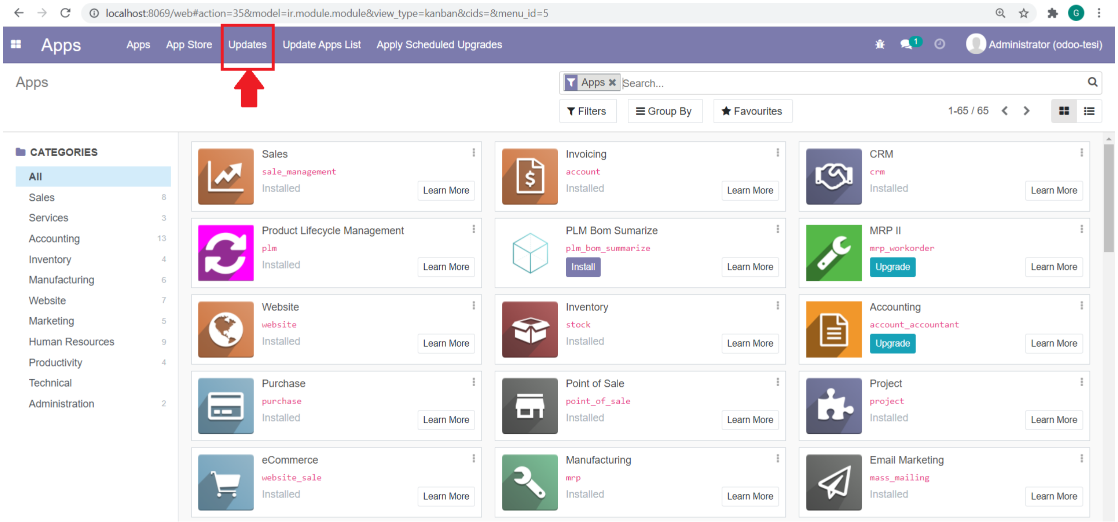
To integrate the PLM module with Odoo, you need to open the odoo.conf file inside the server folder and in the meantime stop the active service in the system. One flies found the file, copy the Odooplm installation path as in Figure 4.3 .



```
[options]
addons_path = C:\Program Files\Odoo 14.0.20210928\server\odoo\addons C:\Users\Giorgio Spadaro\odoo\odooplm
admin_passwd = $pbkdf2-sha512$25000$XatVKkXImRPiHIOw1hoj5A$byk3PMsw3PBRVsj10HKII7L4ttxFy27P1T.xbrQBZ48h467Hpq9RyyVGr
bin_path = C:\Program Files\Odoo 14.0.20210928\thirdparty
csv_internal_sep = ,
data_dir = C:\Users\Giorgio Spadaro\AppData\Local\OpenERP S.A\Odoo
```

Figure 4.8: Odoo14 config file

After that, we restart the service and launch Odoo. In the Homepage of the Apps we will not see our form yet, you need to click on updates.



localhost:8069/web#action=35&model=ir.module.module&view_type=kanban&cids=&menu_id=5

Figure 4.9: Odoo Homepage

The software will automatically install the modules, following the path added in the configuration file and will return in sight the PLM module, ready to be used.

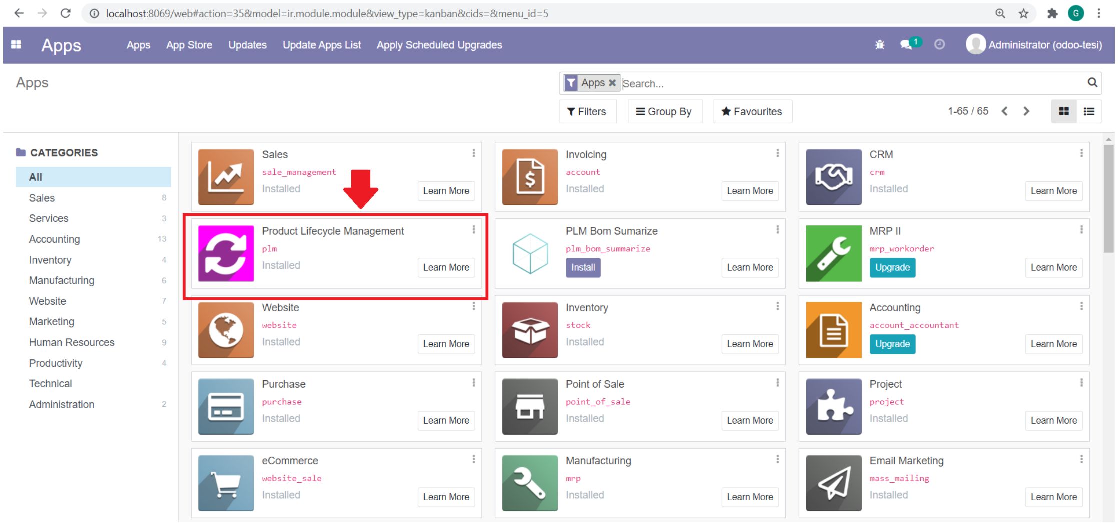


Figure 4.10: PLM module

4.3 Project Views

The structure of a project that we will manage within our software can be schematized as shown in the following figure:

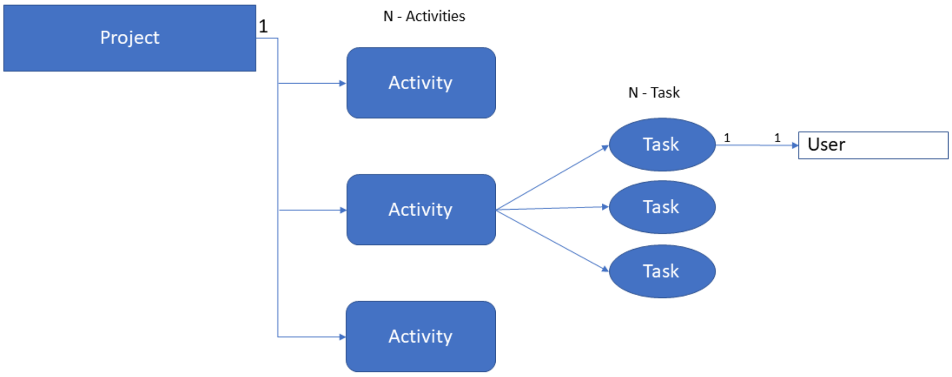


Figure 4.11: View Project

The relationship between project and activities is one-to-many: in fact a project is the combination of a set of activities and the latter are the result of the set of tasks and resources, for this reason also the relationship between Activity and Tasks is one-to-many. In our case the only exception is the relationship between task and user. Usually in more complex projects, a task can be carried out by more resources, thus fostering cooperation and teamwork. In Odoo, this does not happen and changing the logical structure of the database requires a very complex redesign work, so the relationship between task and resources is one-to-one, or each task can be assigned just one user.

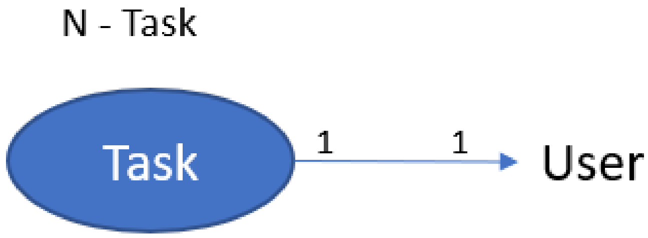


Figure 4.12: Task-User Relationship

4.4 Framework Development

4.4.1 Step 1: Reverse Engineering

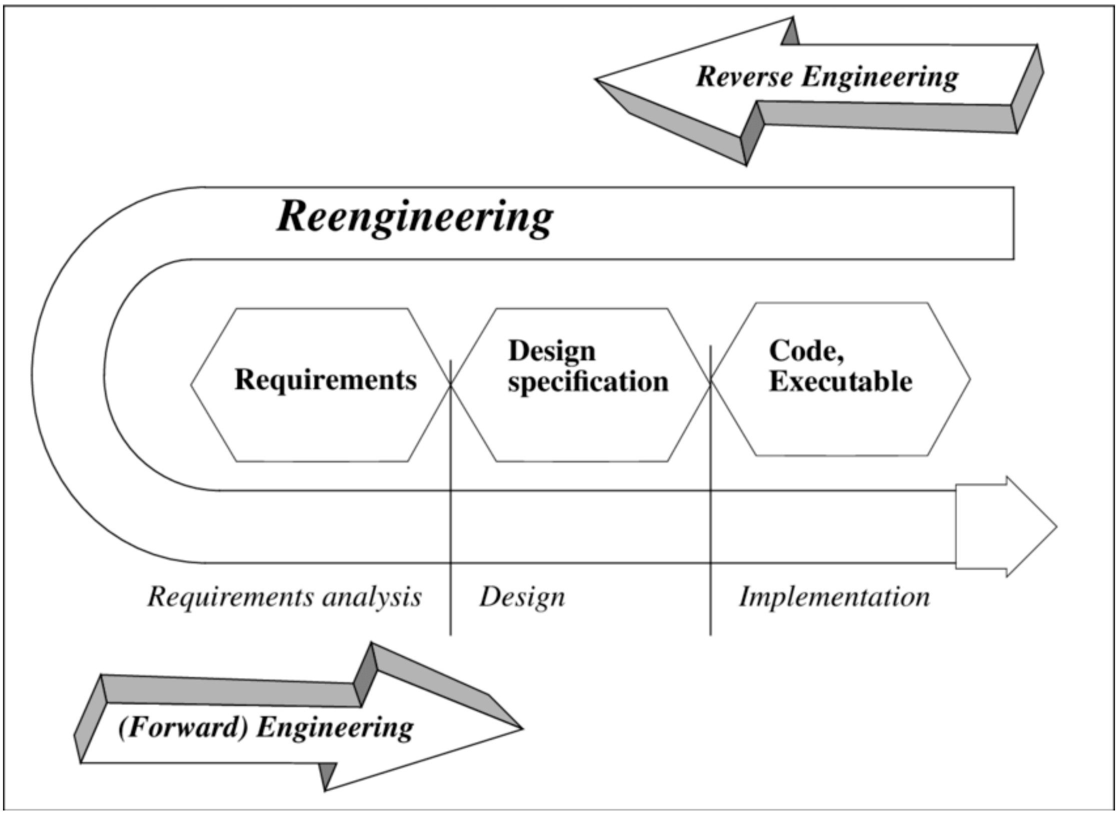
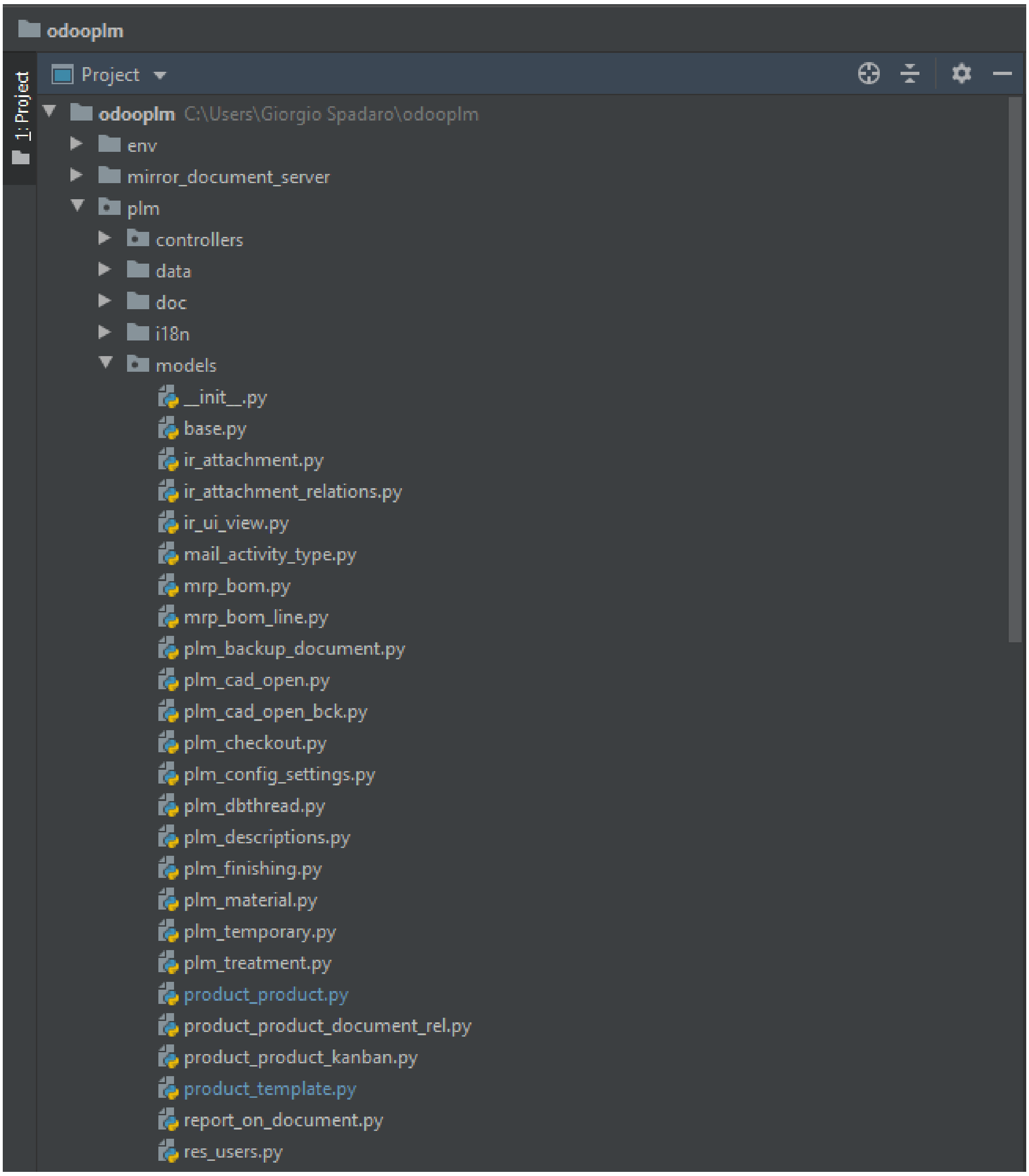


Figure 4.13: Software engineering, reverse engineering and reengineering

The first step of this study was dedicated to Reverse Engineering, a process to identify the properties of a physical object, in our case the software, through the complete analysis of structure, functions and operation of this element. With this process then we search within the structure of the software modules that we want to analyze and modify.



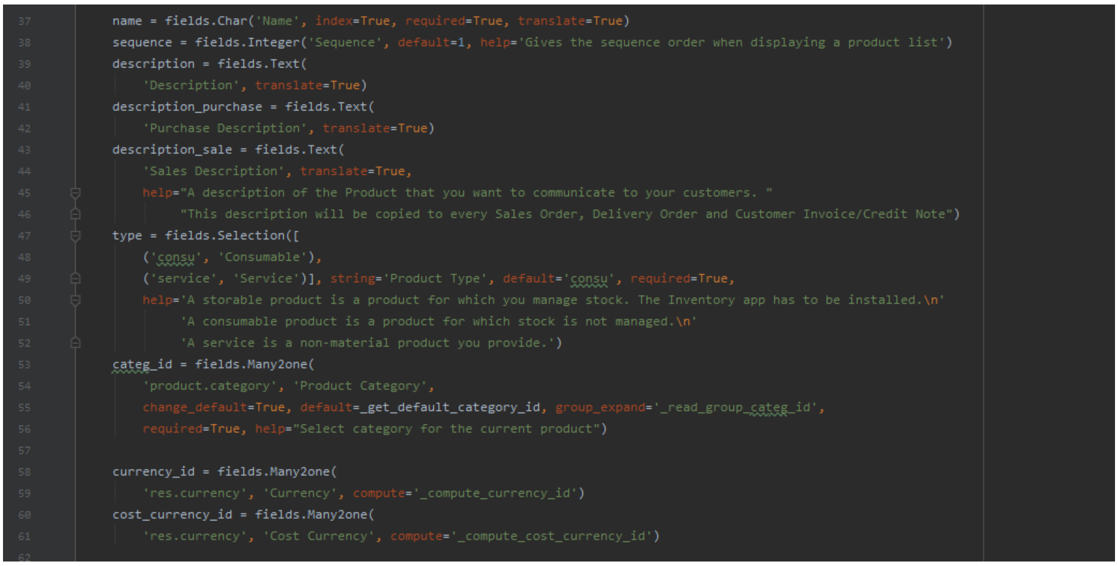
```
odooplm
Project
1: Project
▼ odooplm C:\Users\Giorgio Spadaro\odooplm
  ▶ env
  ▶ mirror_document_server
  ▼ plm
    ▶ controllers
    ▶ data
    ▶ doc
    ▶ i18n
    ▼ models
        __init__.py
        base.py
        ir_attachment.py
        ir_attachment_relations.py
        ir_ui_view.py
        mail_activity_type.py
        mrp_bom.py
        mrp_bom_line.py
        plm_backup_document.py
        plm_cad_open.py
        plm_cad_open_bck.py
        plm_checkout.py
        plm_config_settings.py
        plm_dbthread.py
        plm_descriptions.py
        plm_finishing.py
        plm_material.py
        plm_temporary.py
        plm_treatment.py
        product_product.py
        product_product_document_rel.py
        product_product_kanban.py
        product_template.py
        report_on_document.py
        res_users.py
```

Figure 4.14: OdooPLM folders structure

4.4.2 Step 2: Detecting meaningful code-parts

Opening the folders we see that the structure of each application maintains a standard, which facilitates its search.
Then we take the application of our interest, the folder "plm" and let's open the

sheet "prodcut.template".

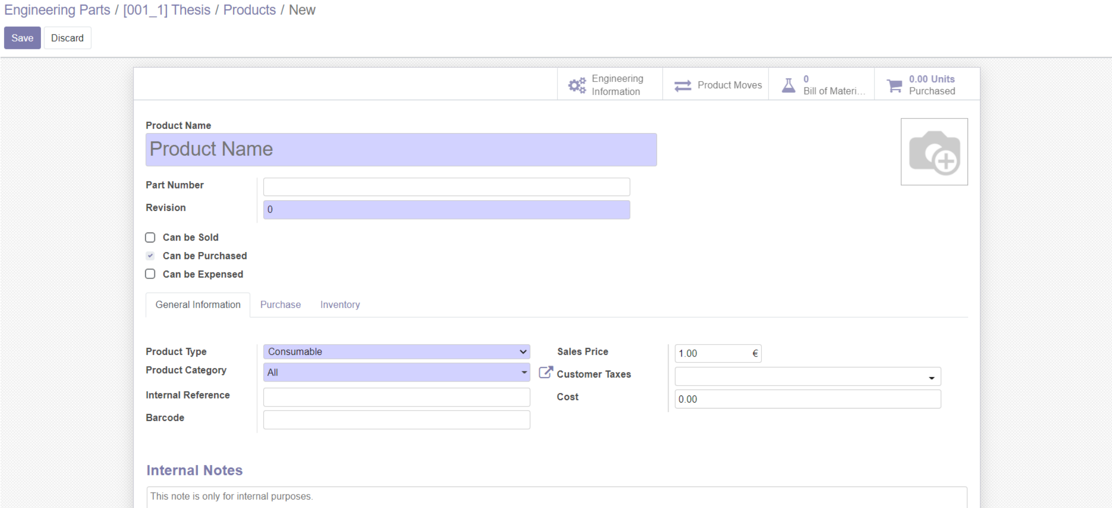


```python
    name = fields.Char('Name', index=True, required=True, translate=True)
    sequence = fields.Integer('Sequence', default=1, help='Gives the sequence order when displaying a product list')
    description = fields.Text(
        'Description', translate=True)
    description_purchase = fields.Text(
        'Purchase Description', translate=True)
    description_sale = fields.Text(
        'Sales Description', translate=True,
        help="A description of the Product that you want to communicate to your customers. "
             "This description will be copied to every Sales Order, Delivery Order and Customer Invoice/Credit Note")
    type = fields.Selection([
        ('consu', 'Consumable'),
        ('service', 'Service')], string='Product Type', default='consu', required=True,
        help='A storable product is a product for which you manage stock. The Inventory app has to be installed.\n'
             'A consumable product is a product for which stock is not managed.\n'
             'A service is a non-material product you provide.')
    categ_id = fields.Many2one(
        'product.category', 'Product Category',
        change_default=True, default=_get_default_category_id, group_expand='_read_group_categ_id',
        required=True, help="Select category for the current product")

    currency_id = fields.Many2one(
        'res.currency', 'Currency', compute='_compute_currency_id')
    cost_currency_id = fields.Many2one(
        'res.currency', 'Cost Currency', compute='_compute_cost_currency_id')
```

Figure 4.15: Product Template Code

Inside this python sheet, is contained the class "ProductTemplatEextension" that inherits the attributes from the table "product.template" present in ODOO Software. In reference to the code, the view that the user will have will be the following:



Product Name: Product Name

Part Number:

Revision: 0

☐ Can be Sold
☑ Can be Purchased
☐ Can be Expensed

Product Type: Consumable

Product Category: All

Internal Reference:

Barcode:

Sales Price: 1.00 €

Customer Taxes:

Cost: 0.00

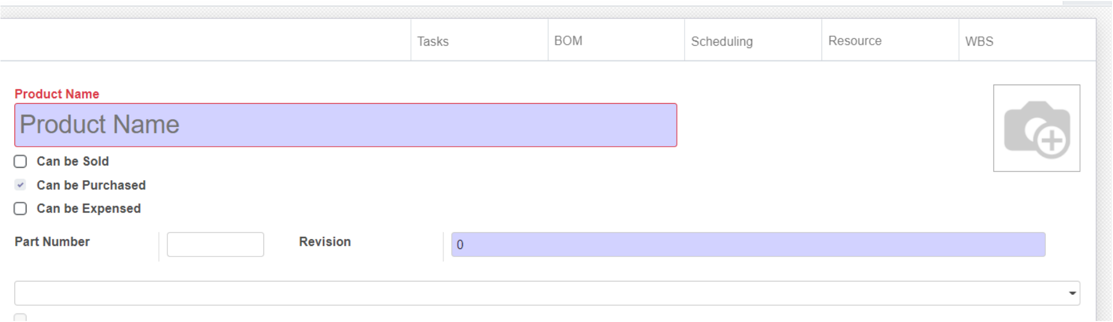
Internal Notes

This note is only for internal purposes.

Figure 4.16: View product creation

4.4.3 Step 3: Changing the code following the to-be model

Using the classes present in the module "project", we can combine the attributes in order to have a product, during his creation, with project features, returning the following view:



Product Name

Product Name

□ Can be Sold

☑ Can be Purchased

□ Can be Expensed

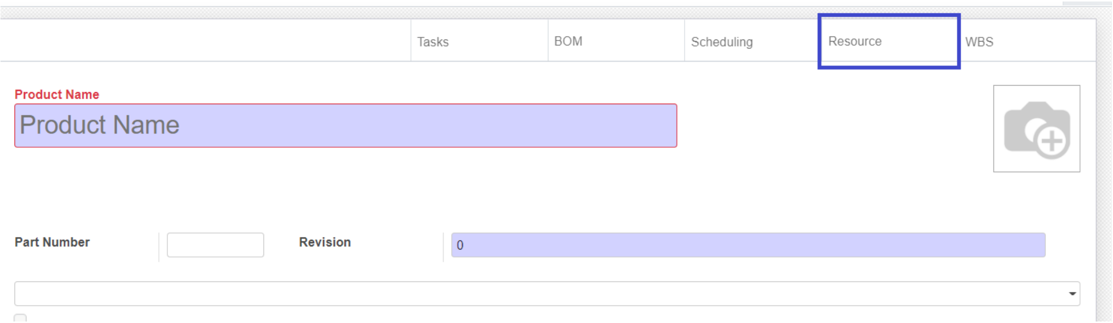
Part Number

Revision 0

Figure 4.17: View task creation to-be

Let's go into the individual tabs:

Resource First of all, we are going to create the account for the team members



Product Name

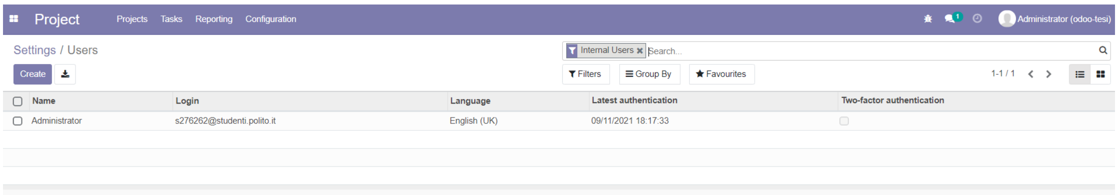
Product Name

Part Number

Revision 0

Figure 4.18: View Product Creation

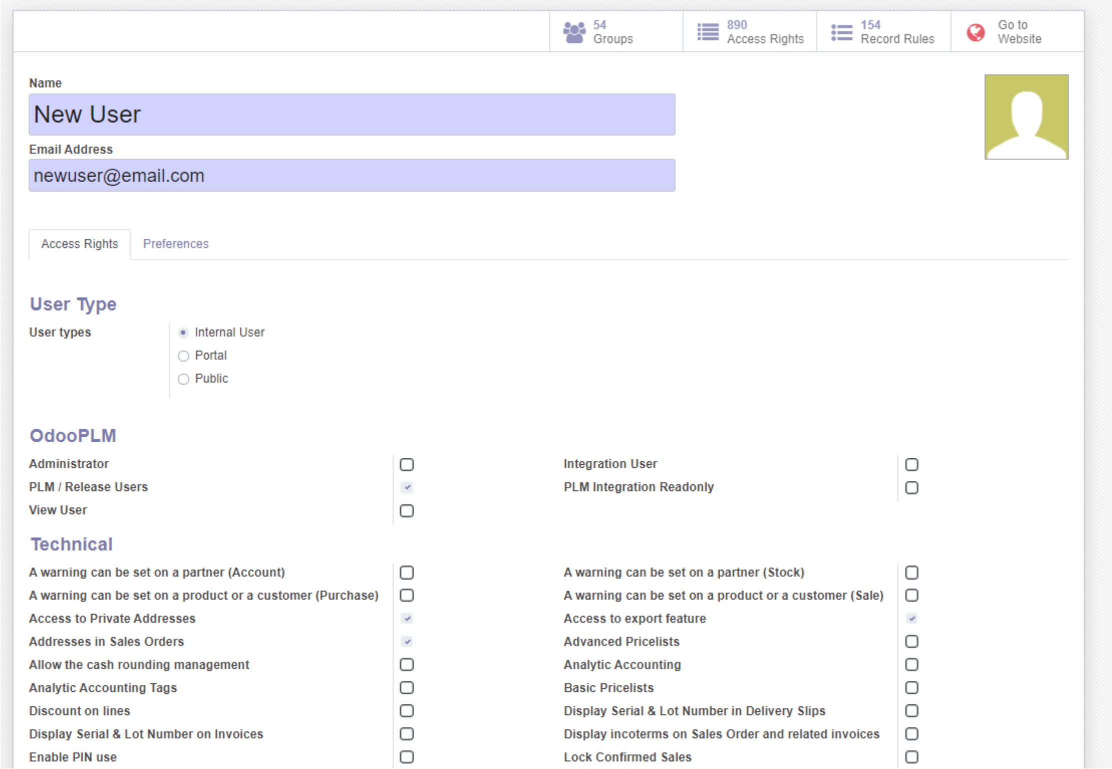
Then ideally you click the resource tab and the user will be sent back to the user creation page already implemented by Odoo. So here the work does not concern the modification of any class, but only put the reference, in the resource tab, of the method "create user".



☐	Name	Login	Language	Latest authentication	Two-factor authentication
☐	Administrator	s276262@studenti.polito.it	English (UK)	09/11/2021 18:17:33	☐

Figure 4.19: View User Creation

You select the "create" button and a new window will open as shown below:



Name: New User

Email Address: newuser@email.com

User Type

User types: ● Internal User ○ Portal ○ Public

OdooPLM

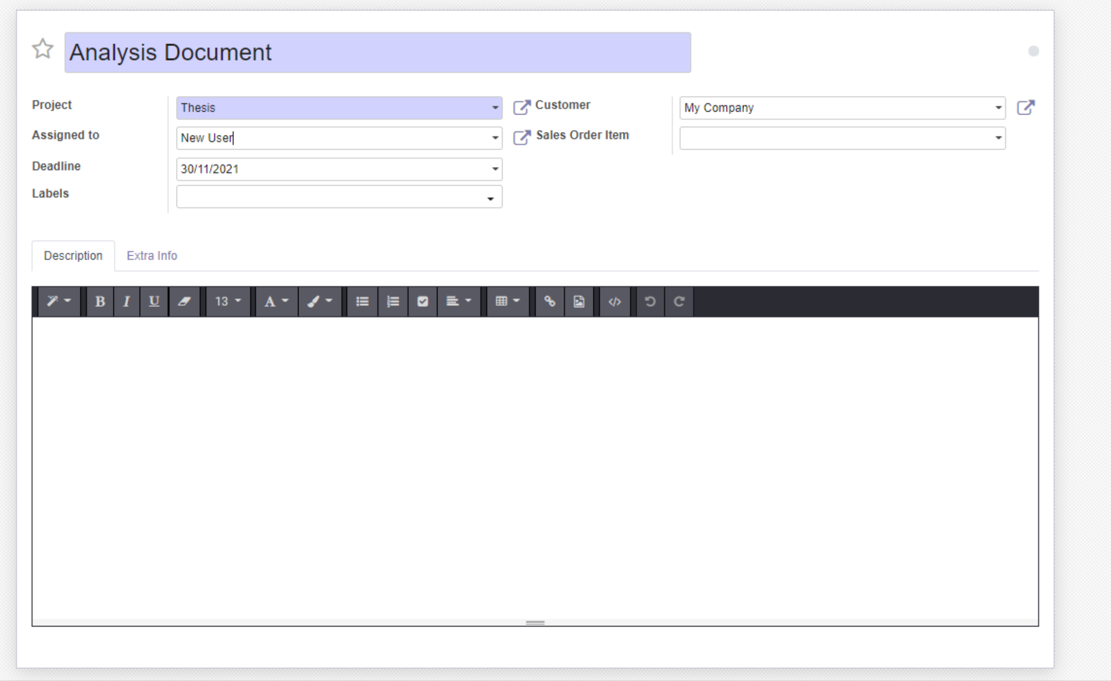
Administrator	☐	Integration User	☐
PLM / Release Users	☑	PLM Integration Readonly	☐
View User	☐		

Technical

A warning can be set on a partner (Account)	☐	A warning can be set on a partner (Stock)	☐
A warning can be set on a product or a customer (Purchase)	☐	A warning can be set on a product or a customer (Sale)	☐
Access to Private Addresses	☑	Access to export feature	☑
Addresses in Sales Orders	☑	Advanced Pricelists	☐
Allow the cash rounding management	☐	Analytic Accounting	☐
Analytic Accounting Tags	☐	Basic Pricelists	☐
Discount on lines	☐	Display Serial & Lot Number in Delivery Slips	☐
Display Serial & Lot Number on Invoices	☐	Display incoterms on Sales Order and related invoices	☐
Enable PIN use	☐	Lock Confirmed Sales	☐

Figure 4.20: View New User Creation

Odoo offers the ability to create a user very quickly, only asking for name and email. Already in creation, you can enable the user down the type, but above all you can enable the views, ie tell Odoo what can see this user and then where can operate, all this thanks to the list of boxes that you see in the figure.



Analysis Document

Project: Thesis

Assigned to: New User

Deadline: 30/11/2021

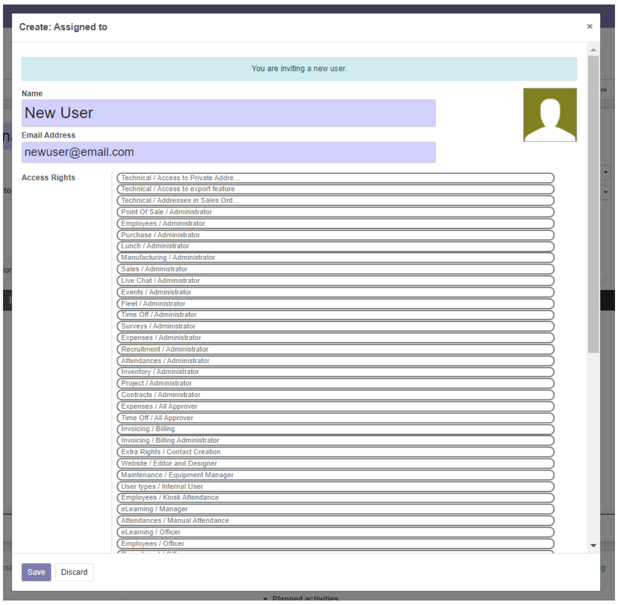
Labels:

Customer: My Company

Sales Order Item:

Figure 4.21: View User Assignment

Once you've created the task, you can assign it directly from a dropbox in the task creation view.



Create: Assigned to

You are inviting a new user.

Name: New User

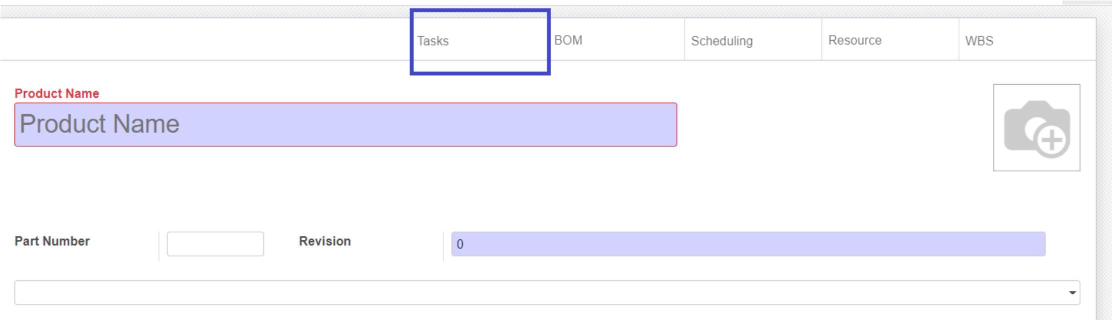
Email Address: newuser@email.com

Access Rights:

- Technical / Access to Private Addre...
- Technical / Access to export feature
- Technical / Addresses in Sales Ord...
- Point Of Sale / Administrator
- Employees / Administrator
- Purchase / Administrator
- Lunch / Administrator
- Manufacturing / Administrator
- Sales / Administrator
- Live Chat / Administrator
- Events / Administrator
- Fleet / Administrator
- Time Off / Administrator
- Surveys / Administrator
- Expenses / Administrator
- Recruitment / Administrator
- Attendances / Administrator
- Inventory / Administrator
- Project / Administrator
- Contracts / Administrator
- Expenses / All Approver
- Time Off / All Approver
- Invoicing / Billing
- Invoicing / Billing Administrator
- Extra Rights / Contact Creation
- Website / Editor and Designer
- Maintenance / Equipment Manager
- User types / Internal User
- Employees / Kiosk Attendance
- eLearning / Manager
- Attendances / Manual Attendance
- eLearning / Officer
- Employees / Officer

Figure 4.22: View Alternative User Creation Way

In the figure above we see an alternative and faster way to create a new account. We can easily access this screen by selecting the "add new user" option in the combobox we use to assign a resource.

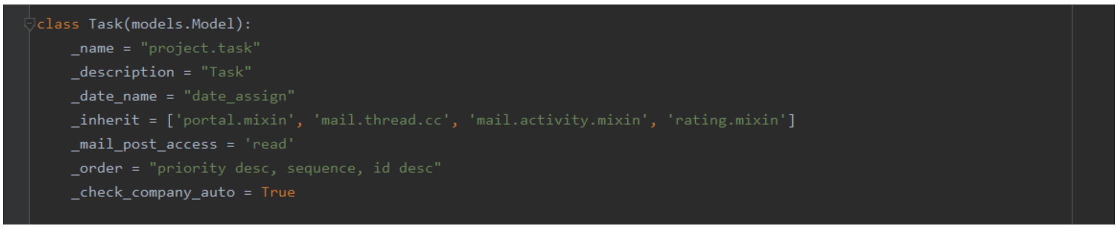


Product Name
Product Name

Part Number
Revision 0

Figure 4.23: Task class Odoo14

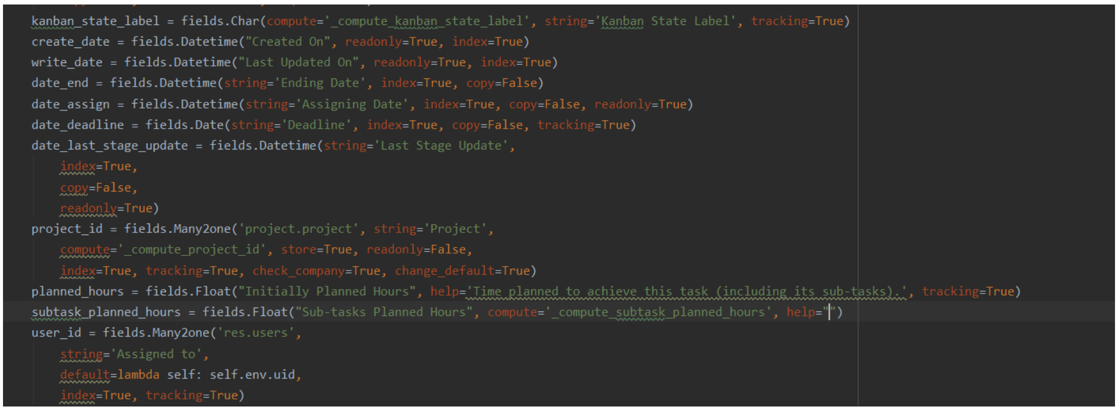
Tasks The aspects that characterize the tasks are mainly the duration, the type and to whom they are assigned. For this reason, we see the class "Task" present in the sheet "project.py" of the module "project" of Odoo, remembering that Odooplm works extends the tables already existing and inheriting the attributes.



```
class Task(models.Model):
    _name = "project.task"
    _description = "Task"
    _date_name = "date_assign"
    _inherit = ['portal.mixin', 'mail.thread.cc', 'mail.activity.mixin', 'rating.mixin']
    _mail_post_access = 'read'
    _order = "priority desc, sequence, id desc"
    _check_company_auto = True
```

Figure 4.24: Task class Odoo14

Scrolling down we find the description of the feutures



```
kanban_state_label = fields.Char(compute='_compute_kanban_state_label', string='Kanban State Label', tracking=True)
create_date = fields.Datetime("Created On", readonly=True, index=True)
write_date = fields.Datetime("Last Updated On", readonly=True, index=True)
date_end = fields.Datetime(string='Ending Date', index=True, copy=False)
date_assign = fields.Datetime(string='Assigning Date', index=True, copy=False, readonly=True)
date_deadline = fields.Date(string='Deadline', index=True, copy=False, tracking=True)
date_last_stage_update = fields.Datetime(string='Last Stage Update',
    index=True,
    copy=False,
    readonly=True)
project_id = fields.Many2one('project.project', string='Project',
    compute='_compute_project_id', store=True, readonly=False,
    index=True, tracking=True, check_company=True, change_default=True)
planned_hours = fields.Float("Initially Planned Hours", help='Time planned to achieve this task (including its sub-tasks).', tracking=True)
subtask_planned_hours = fields.Float("Sub-tasks Planned Hours", compute='_compute_subtask_planned_hours', help="")
user_id = fields.Many2one('res.users',
    string='Assigned to',
    default=lambda self: self.env.uid,
    index=True, tracking=True)
```

Figure 4.25: Task's table attributes

In figure 4.25, we see how Odoo manages the most important dates present in the task, through the following attributes:

- **create date:** which tells us when the task is created

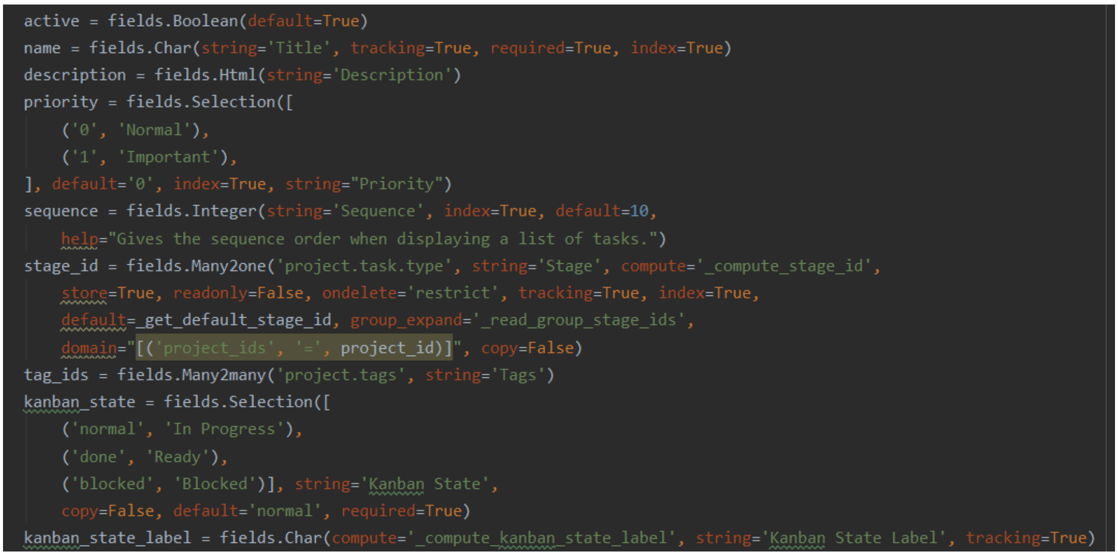
- **write date:** gives us information about when the task is updated

- **date end:** indicates the expected end date

- **date assign:** indicates when the task is assigned to a user

- **date deadline:** indicates the deadline within which the task must be completed

As for the assignment of the user, this takes place by filling the attribute "user id" with the id of one of the users who are allocated on the project. They have a many-to-one relationship, because a user can obviously have more tasks assigned.



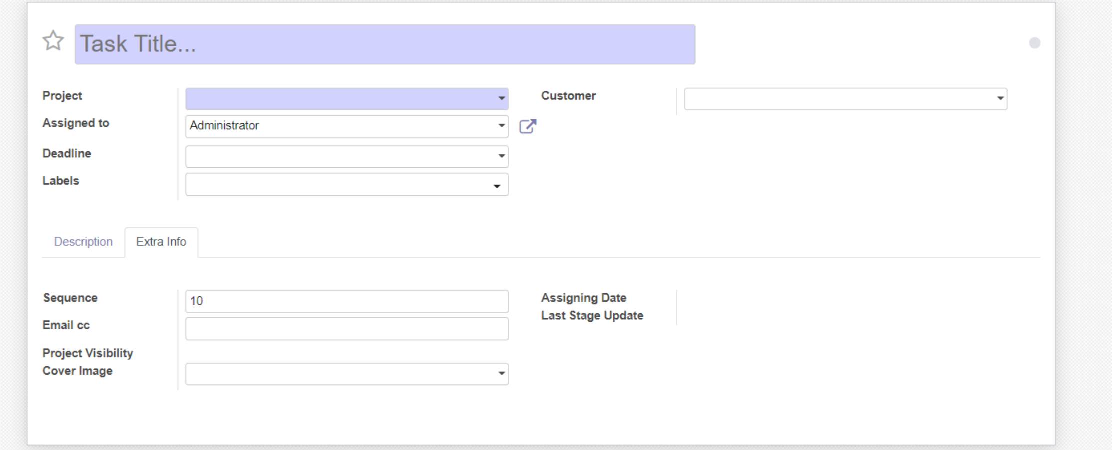
```
active = fields.Boolean(default=True)
name = fields.Char(string='Title', tracking=True, required=True, index=True)
description = fields.Html(string='Description')
priority = fields.Selection([
    ('0', 'Normal'),
    ('1', 'Important'),
], default='0', index=True, string="Priority")
sequence = fields.Integer(string='Sequence', index=True, default=10,
    help="Gives the sequence order when displaying a list of tasks.")
stage_id = fields.Many2one('project.task.type', string='Stage', compute='_compute_stage_id',
    store=True, readonly=False, ondelete='restrict', tracking=True, index=True,
    default=_get_default_stage_id, group_expand='_read_group_stage_ids',
    domain="[('project_ids', '=', project_id)]", copy=False)
tag_ids = fields.Many2many('project.tags', string='Tags')
kanban_state = fields.Selection([
    ('normal', 'In Progress'),
    ('done', 'Ready'),
    ('blocked', 'Blocked')], string='Kanban State',
    copy=False, default='normal', required=True)
kanban_state_label = fields.Char(compute='_compute_kanban_state_label', string='Kanban State Label', tracking=True)
```

Figure 4.26: Task's table main attributes

Beyond the name, the description and the priority, we find the attribute "kanban state", which indicates us the state where the task is located, which can be :

- In progress
- Ready
- Blocked

While the "sequence" shows us if the task is part of a larger sequence of tasks to be performed and in what order it is located. At the end of this analysis, clicking the tab present in Figure 4.7, we will have the following view:



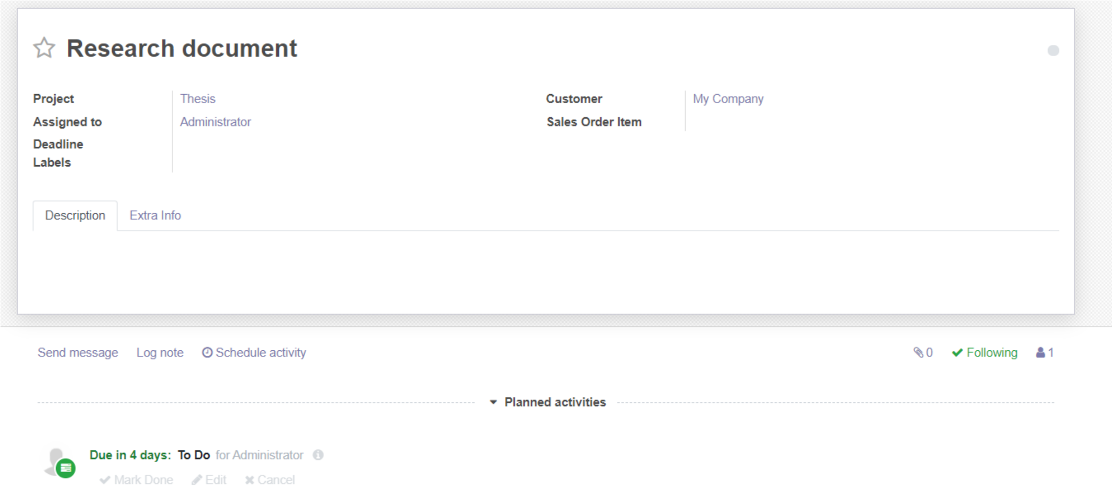
Task Title...

Project
Assigned to: Administrator
Deadline
Labels
Customer

Sequence: 10
Email cc
Project Visibility
Cover Image
Assigning Date
Last Stage Update

Figure 4.27: View Task creation

In which the project will be selected automatically and the user, that is the student, will only have to fill in the present forms and send in "production" the task. If all fields are filled in correctly, the task will be created and added to the project. It will always be possible to modify it anyway.



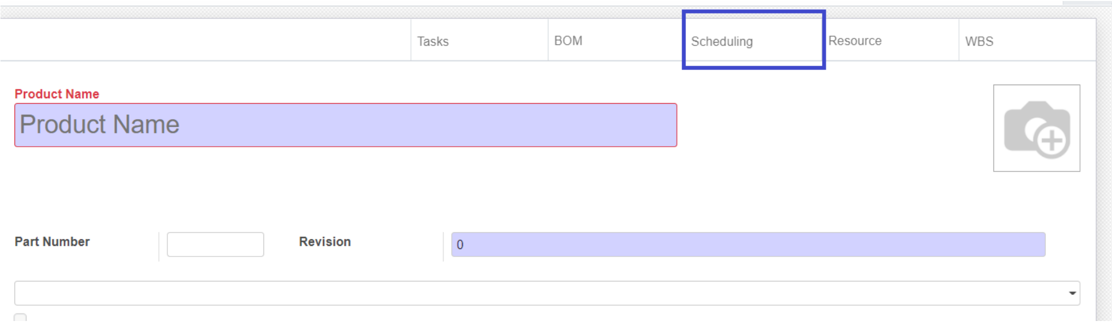
☆ Research document

Project	Thesis	Customer	My Company
Assigned to	Administrator	Sales Order Item	
Deadline			
Labels			

Due in 4 days: To Do for Administrator

Figure 4.28: View Task created

Scheduling The third point to be explored is scheduling: scheduling , as already seen in the use case diagram, will be a function that will not depend on the user, or rather, the user will only have to set the duration and sequences of all tasks, Once this is done, the system will give an overview of the durations and resources.



Product Name

Product Name

Part Number		Revision	0

Figure 4.29: View Product Creation

In Figure 4.30, you have the first view to schedule an activity: directly from the Kaban project, in which you have the backlog vision of the project.

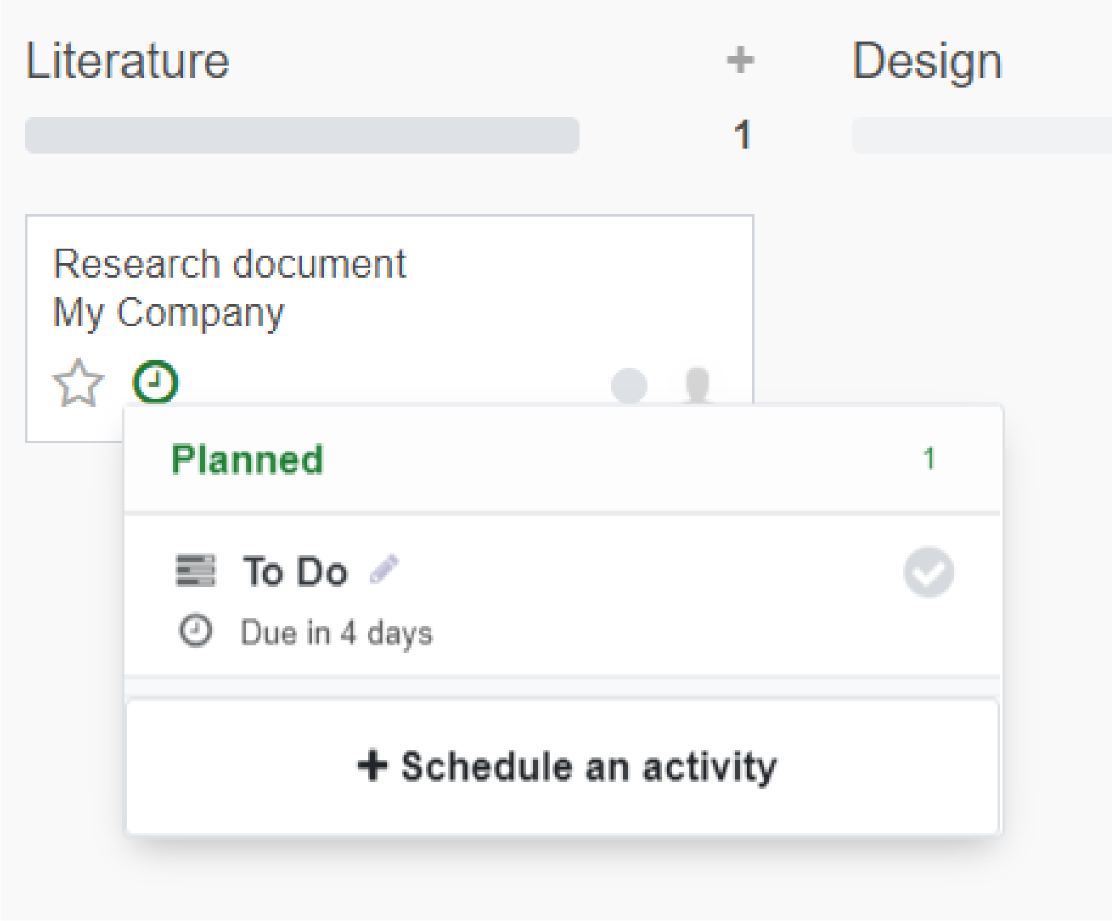
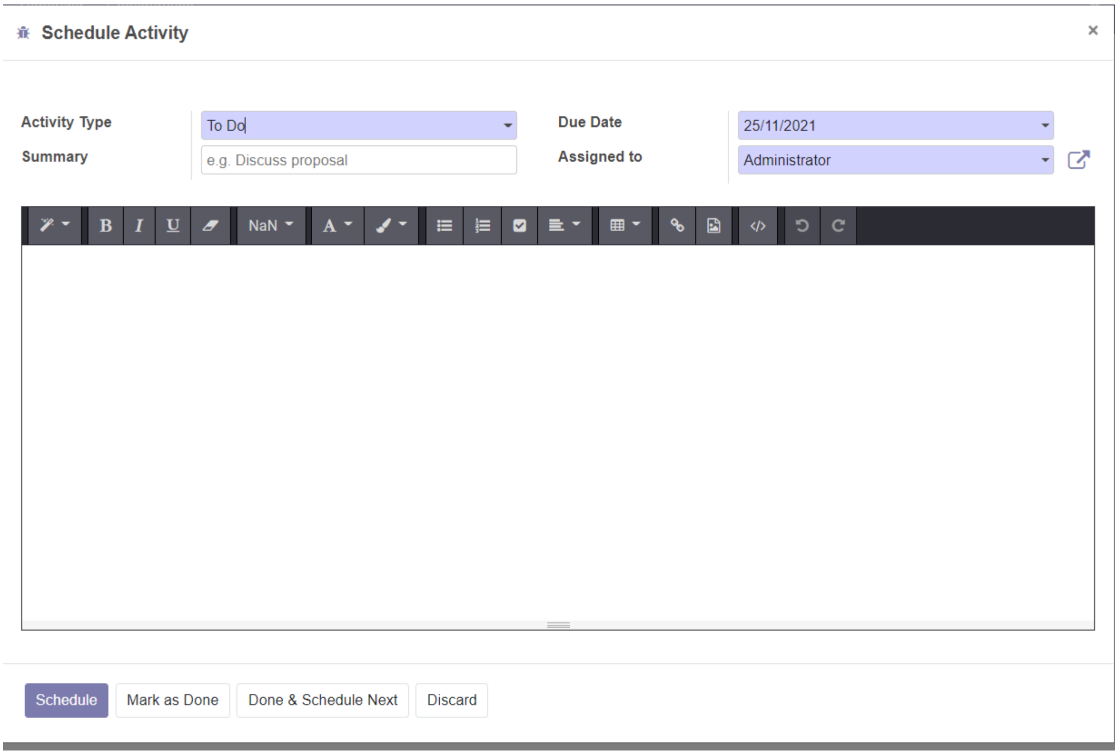


Figure 4.30: Scheduling a task

Once clicked on "schedule activity", the view that will appear will be similar to creating a task, with the difference that the fields and entries interested in the activity will be only:

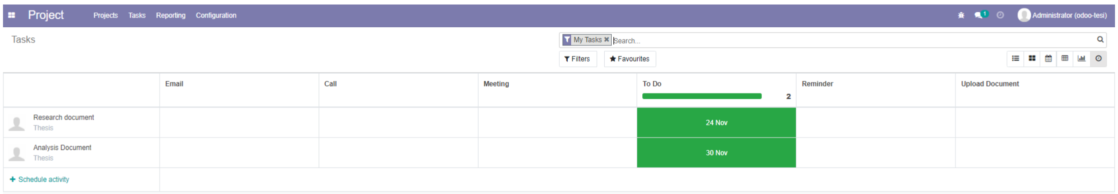
- Activity type
- Due Date
- Summary
- Assigned to



Activity Type	To Do	Due Date	25/11/2021
Summary	e.g. Discuss proposal	Assigned to	Administrator

Figure 4.31: Setting schedule

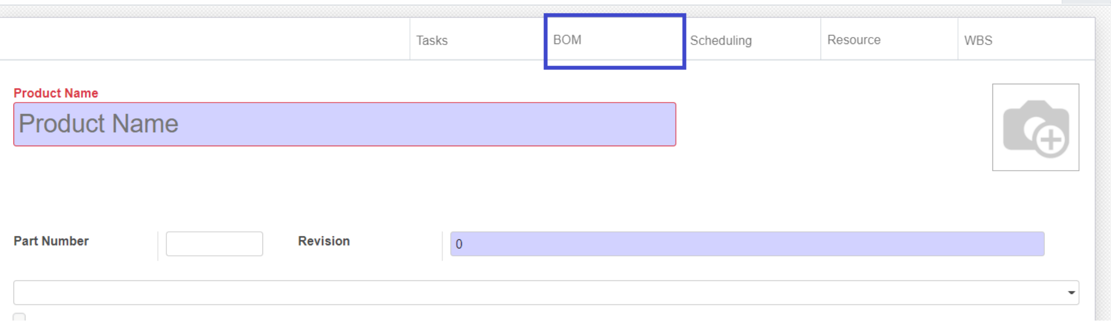
Once the fields mentioned above have been filled in, the system will assign the activity to the user concerned, will outline the various states of the activities present (To-Do, Ready, etc.) and will automatically account for the total duration.



	Email	Call	Meeting	To Do 2	Reminder	Upload Document
Research document Thesis				24 Nov		
Analysis Document Thesis				30 Nov		

Figure 4.32: View Scheduling

BOM As already mentioned in Chapter 3, the Bill of Materials is a view of the distinct base of the product, oriented to production, then internally to the company.



Product Name

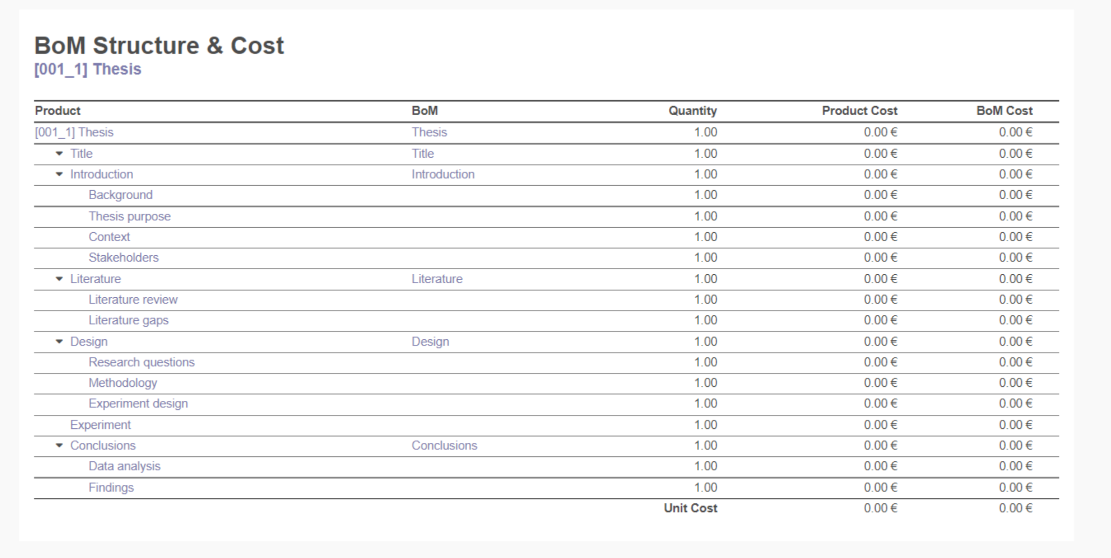
Product Name

Part Number

Revision 0

Figure 4.33: View Product Creation

As see in Figure 4.28, in the tree structure, are indicated the business characteristics of the product, that is the quantity and the cost of production.



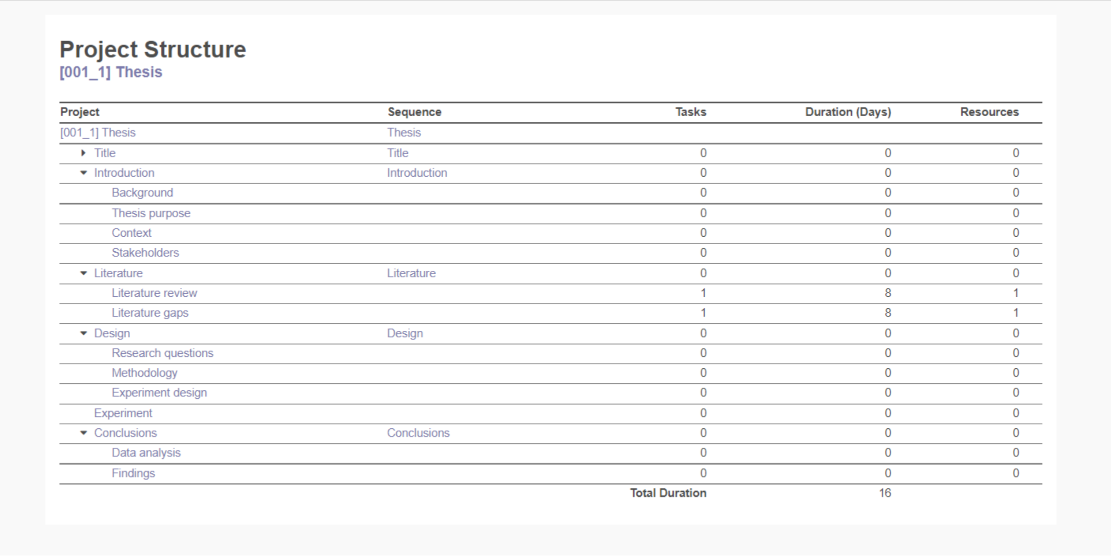
BoM Structure & Cost

[001_1] Thesis

Product	BoM	Quantity	Product Cost	BoM Cost
[001_1] Thesis	Thesis	1.00	0.00 €	0.00 €
Title	Title	1.00	0.00 €	0.00 €
Introduction	Introduction	1.00	0.00 €	0.00 €
Background		1.00	0.00 €	0.00 €
Thesis purpose		1.00	0.00 €	0.00 €
Context		1.00	0.00 €	0.00 €
Stakeholders		1.00	0.00 €	0.00 €
Literature	Literature	1.00	0.00 €	0.00 €
Literature review		1.00	0.00 €	0.00 €
Literature gaps		1.00	0.00 €	0.00 €
Design	Design	1.00	0.00 €	0.00 €
Research questions		1.00	0.00 €	0.00 €
Methodology		1.00	0.00 €	0.00 €
Experiment design		1.00	0.00 €	0.00 €
Experiment		1.00	0.00 €	0.00 €
Conclusions	Conclusions	1.00	0.00 €	0.00 €
Data analysis		1.00	0.00 €	0.00 €
Findings		1.00	0.00 €	0.00 €
		Unit Cost	0.00 €	0.00 €

Figure 4.34: BOM Structure As-Is

Once, then, the necessary modifications will be made to the product class, changing the process and production characteristics into design characteristics, the desired view will be as follows:



Project Structure

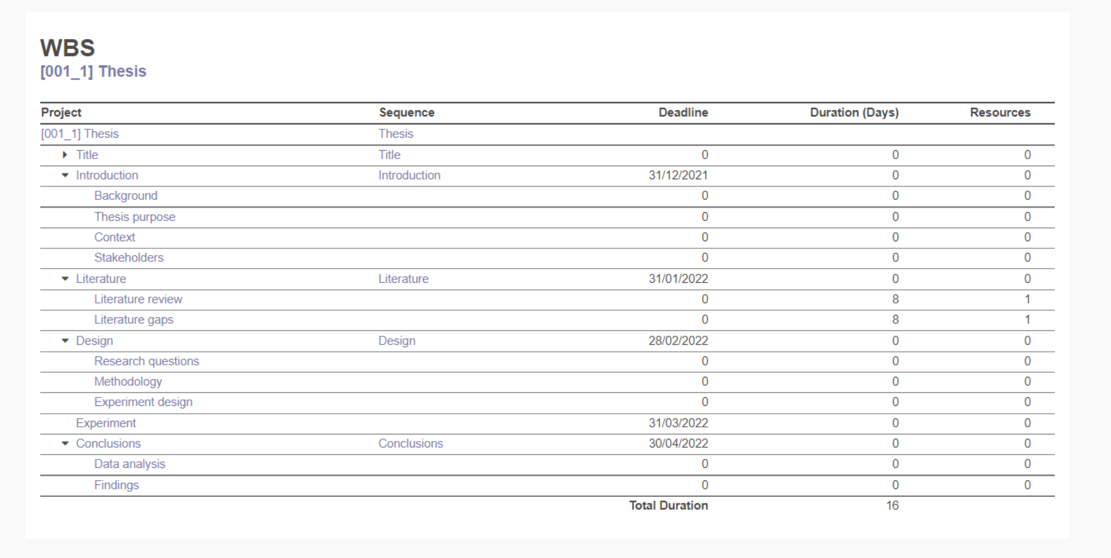
[001_1] Thesis

Project	Sequence	Tasks	Duration (Days)	Resources
[001_1] Thesis	Thesis			
▸ Title	Title	0	0	0
▾ Introduction	Introduction	0	0	0
Background		0	0	0
Thesis purpose		0	0	0
Context		0	0	0
Stakeholders		0	0	0
▾ Literature	Literature	0	0	0
Literature review		1	8	1
Literature gaps		1	8	1
▾ Design	Design	0	0	0
Research questions		0	0	0
Methodology		0	0	0
Experiment design		0	0	0
Experiment		0	0	0
▾ Conclusions	Conclusions	0	0	0
Data analysis		0	0	0
Findings		0	0	0
		Total Duration	16	

Figure 4.35: Project Structure To-Be

For every activity and subactivcity the sequence is shown, that is the father of the Bom, the number of tasks, the duration and the total resources employed.

WBS The WBS structure instead, being oriented to the delivery, like main attributes to visualize of the activities you have mainly the milestones, that is to project the attribute "deadline".



WBS

[001_1] Thesis

Project	Sequence	Deadline	Duration (Days)	Resources
[001_1] Thesis	Thesis			
▸ Title	Title	0	0	0
▾ Introduction	Introduction	31/12/2021	0	0
Background		0	0	0
Thesis purpose		0	0	0
Context		0	0	0
Stakeholders		0	0	0
▾ Literature	Literature	31/01/2022	0	0
Literature review		0	8	1
Literature gaps		0	8	1
▾ Design	Design	28/02/2022	0	0
Research questions		0	0	0
Methodology		0	0	0
Experiment design		0	0	0
Experiment		31/03/2022	0	0
▾ Conclusions	Conclusions	30/04/2022	0	0
Data analysis		0	0	0
Findings		0	0	0
		Total Duration	16	

Figure 4.36: WBS Structure To-Be

Chapter 5

Conclusion

5.1 Findings

At the end of this analysis we can say that Odoo retains the strengths mentioned above, namely a powerful database management and a considerable flexibility of customization. The weaknesses found unfortunately continue to exist, that is the poor documentation that, combined with a complex structure of the source code, makes it very difficult the reverse engineering process. In addition, over the years, the User Experience (UX) and the User Interface (UI) have stratified a considerable complexity becoming a point of weakness. This complexity is the direct cause of an instability in the architecture: in fact, over time Odoo has lost more and more its modulistic architecture, becoming more complex in the search for the functionality made available by the software. This causes an allocation of users who do not have the resources or abilities to be able to change the code and pull out the methods that interest.

In any case Odoo is presented as a complete ERP and it has been demonstrated in this elaboration that , thanks to its modulistic structure, can be adapted and used also like a PLM offering however of the performances highly satisfactory.

5.2 Proof of Concept

As proof of concept we can say that this elaboration is perfectly placed in the so called "0" phase that concerns the development of a software or its customization: the analysis.

In fact, in the analysis phase the bases are laid to operate in a clear and direct way in the next phases until deployment, such as software development and testing. In phase zero, in addition to dealing with the context in which you find the software and the set up of the latter, are drawn out the requirements that you want to

implement and that you want to view or simply the requests treated as nice to have. It is in fact used, as seen until now, a model to-be and compared to the model in which the system is currently located, called as-is, shedding light on the differences, then the changes that you want to make.
Once the documentation that summarizes the contents of the analysis has been created, this documentation is passed to guide the next steps.

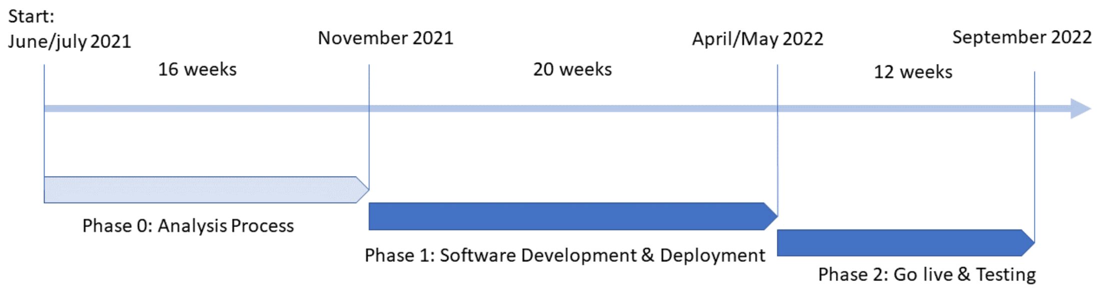

Start:
June/july 2021
November 2021
April/May 2022
September 2022
16 weeks
20 weeks
12 weeks
Phase 0: Analysis Process
Phase 1: Software Development & Deployment
Phase 2: Go live & Testing

Figure 5.1: Project's GANTT Diagram

Bibliography

[1] *Stakeholder di progetto.* 2021. URL: https://www.humanwareonline.com/project-management/center/stakeholder-di-progetto/ (cit. on pp. 3, 5).

[2] *ISO 21500, che cos'è?* 2021. URL: https://www.imlearning.it/iso21500/ (cit. on p. 6).

[3] *Cos'è il project management?* 2021. URL: https://www.humanwareonline.com/project-management/center/cose-il-project-management/ (cit. on p. 7).

[4] *Cos'è il project management? Facciamo chiarezza con 4 seminari.* 2021. URL: https://www.ordingbo.it/2020/11/10/cose-il-project-management-facciamo-chiarezza-con-4-seminari-scarica-le-slide/ (cit. on p. 7).

[5] *Cos'è e a cosa serve il project management.* 2021. URL: https://www.formalabsrl.it/news/cos-e-a-cosa-serve-il-project-management/56/ (cit. on p. 7).

[6] *Traditional Product Development Process.* 2021. URL: https://slidemodel.com/templates/traditional-product-development-process-for-powerpoint/6855-01-traditional-product-development-process-powerpoint-2/ (cit. on p. 8).

[7] *A Guide For Work Breakdown Structure in Project Management (For Successful Projects).* 2021. URL: https://www.proofhub.com/articles/work-breakdown-structure (cit. on pp. 9, 11).

[8] *DISTINTA BASE: COS'È? A COSA SERVE?* 2021. URL: https://farenumeri.it/distinta-base/ (cit. on p. 9).

[9] *DISTINTA BASE.* 2021. URL: https://www.logisticaefficiente.it/wiki-logistica/supply-chain/distinta-base.html (cit. on p. 10).

[10] *OpenBOM Part Catalogs and Bill of Material example.* 2021. URL: https://medium.com/@openbom/openbom-part-catalogs-and-bill-of-material-example-2ff29ad8b601 (cit. on p. 10).

[11] *Work breakdown structure, cos'è e a cosa serve nel project management.* 2021. URL: `https://www.almalaboris.com/organismo/blog-lavoro-alma-laboris/87-project-management/2753-work-breakdown-structure-cose-a-cosa-serve-project-management.html` (cit. on pp. 10, 11).

[12] *Work Breakdown Structure Templates by Creately.* 2021. URL: `https://creately.com/blog/examples/work-breakdown-structure-templates/` (cit. on p. 11).

[13] *Ingegneria inversa di un software.* 2021. URL: `https://www.ionos.it/digitalguide/siti-web/programmazione-del-sito-web/reverse-engineering-di-un-software/` (cit. on p. 14).

[14] *Free SWOT analysis templates.* 2021. URL: `https://www.aha.io/roadmapping/guide/templates/swot-analysis` (cit. on p. 16).

[15] *PostgreSQL.* 2021. URL: `https://www.geekandjob.com/wiki/postgresql` (cit. on pp. 19, 20).

[16] *Breaking Free From the ORM: Why Move On?* 2021. URL: `https://medium.com/building-the-system/dont-be-a-sucker-and-stop-using-orms-190add65add4` (cit. on pp. 20, 21).

[17] *ORM (Object Relational Mapping): automatizza la registrazione dei dati.* 2021. URL: `https://www.hwlibre.com/it/orm-object-relational-mapping/` (cit. on p. 21).