POLITECNICO DI TORINO

Collegio di Ingegneria Gestionale e della Produzione
Corso di Laurea Magistrale in Management Engineering

Exploiting collaborative product development in a PLM framework

Supervisor
Prof.ssa Giulia Bruno

Co- supervisor
Prof. Paolo Chiabert
Giovanni Marongiu

Candidate
Faezeh Bouloukifar

A.A 2024/2025

Contents

Abstract

The implementation of a PLM system represents a transformative shift in the product development environment, fostering a more structured and reliable approach to managing the product lifecycle. This underscores the importance of PLM education in training engineers, equipping them not only with theoretical knowledge of product development and lifecycle principles but also with practical, hands-on experience using PLM platforms. By integrating both conceptual understanding and applied learning, PLM education enhances students' skills in product data management. This thesis aims to implement a PLM platform as a supplementary tool for the teaching of product design principles, in particular to improve students' understanding and application of both product lifecycle management and collaborative product development principles. In the development of such a tool, Eppinger's product development model was utilized for implementing the PLM platform. This model allowed for breaking down the implementation process into clear, iterative stages, ensuring that each phase was aligned with the specific needs of the case study. The implementation process included: requirement analysis, a thorough analysis of the Aras PLM platform, design and selection of an appropriate solution, and performing the configurations. An evaluation was then conducted to assess and improve the usability and effectiveness of the system. The results of the evaluation indicated that the application had a positive effect on teaching quality, with the potential for improvement in various parts of the system. Further analysis is recommended to confirm the outcome of this project with a larger sample of students within a classroom environment.

1. Introduction

Product development is getting more and more complex and involves a dynamic collaboration across various fields, including design, engineering, and manufacturing. Therefore, adopting a structured product development process is crucial for projects' success [1]. In this context, Product lifecycle management (PLM) contributes to efficient design, technical competence, higher quality, cost management, customer satisfaction, and overall better engineering solutions to maintain competitive position in the market. PLM as a technology and as a management discipline supports an effective and efficient collaboration among all stakeholders. But integration of stakeholders is also about fostering a culture of trust, clear communication, and shared responsibility. This involves the structured sharing of product knowledge, technology, skills, and resources from different expertise in order to achieve project goals. Collaborative Product Development (CPD) provides such structure to drive collaboration between individuals and teams of different collective expertise and geographically dispersed such as internal departments, external partners, and suppliers. Given this complexity, it is essential for engineers to develop a deep understanding of PLM and CPD concepts. Modern engineering is no longer just about technical problem-solving; it also involves working within a shared product data environment, where cross-functional teamwork and efficient information exchange are critical. Engineers must be skilled in collaborating within digital ecosystems, understanding version control, change management, and data integrity in PLM systems. Additionally, they need to adopt a mindset that values interdisciplinary teamwork, agility, and adaptability in a rapidly evolving industry. To facilitate this learning process, developing a comprehensive glossary of PLM and CPD terminology can happen in the early stages of training to be an engineer, a designer, a consultant, a manager, etc. An analysis of PLM in engineering education shows that most of the universities especially in Italy and France are teaching PLM following different strategies, at different degree levels and presenting this approach from different perspectives [2].

At the Polytechnic of Turin these skills are taught as part of the Industrial Production Engineering and Automotive Engineering programs. In particular, students are taught how to adopt and perform product development principles using traditional teaching methods such as lectures and student projects, thus providing a standardized and practical opportunity for students to delve into these principles and practice it in projects. However, these teaching methods are currently being reassessed to identify more effective strategies for enhancing student performance. The Faculty of Management and Production Engineering is exploring the potential benefits of implementing a real-market PLM software to further improve the teaching approach to further motivate students by effectively engaging them with PLM and CPD principles in an educational environment. Studies on the effectiveness of utilizing such PLM platforms have reported success in increasing the motivation and participation of students and increase in their performance [3] [4] [5] [6]. However, the success of such approach is highly dependent on the design and implementation of the software.

The goal of this thesis is to implement a PLM platform in product design course to support teaching PLM and CPD principles and provide students with essential tools to perform their product development projects.
ARAS PLM Innovator was used in this project. ARAS is an open-source and web-based software allowing low level configurations to adapt to organization processes. The implementation was focused on supporting the teaching course as is without altering the course structure. Consequently, functionalities needed by the existing course structure were implemented. Finally, the platform was assessed by conducting a trial on proficient volunteers.

The work structure is divided as follows:

- Chapter 2 provides a review of the literature relevant to the research topic, establishing the theoretical foundation of the proposed framework.
- Chapter 3 focuses on the case study, offering a detailed analysis of the Aras PLM system, its architecture, and potential applications. This chapter further elaborates on the implementation process, explaining the methodology in alignment with Eppinger's product development approach.
- Chapter 4 offers a structured demonstration of the system's functionalities, providing a clear visualization of its structure, capabilities, and practical usability within the implemented framework.
- Chapter 5 analyzes the findings from the trial program, evaluating the effectiveness of the implemented solution and assessing its overall impact. In addition, it evaluates the application of Eppinger's product development methodology in the system's implementation, examining its influence on process efficiency, decision-making, and integration within the PLM framework.
- Chapter 6 synthesizes the key conclusions of the research, discussing the implications of the implementation. This chapter also outlines the main contributions of the study and proposes potential future developments.

2. Foundations and Contemporary Insights

Over the past century, there has been a dramatic increase in the importance of designing and delivering innovative products efficiently. Central to this challenge of today’s market, lies the product development (PD) process, a structured sequence of activities that transforms ideas into market-ready products. The environment in which products are developed has become increasingly complex due to several key factors, including technological advancements, market demands, regulatory pressures, and globalization. As a result, integration across various competencies and collaboration of diverse and geographically dispersed teams have become critical prerequisites for a successful PD process. The concept of Collaborative Product Development (CPD) ensures the seamless coordination of Different and geographically dispersed expertise areas in a PD process. On the other hand, Product Lifecycle Management (PLM) emerges as a driver in offering a comprehensive framework to manage data, streamline processes, and allocate resources throughout the entire product lifecycle. PLM and CPD intersect in meaningful ways to address the demands of evolved product development environment. While CPD focuses on collaboration methods and brings people and teams together, PLM provides the proper tools to implement those methods by centralizing information, facilitating decision-making and traceability. Together, these frameworks empower organizations to reduce time-to-market, enhance product quality, and foster innovation. A deep understanding of these concepts and their connections is crucial for contemporary engineers who have to adopt them in their daily work. However, to succeed in this challenging endeavor, it is necessary for universities to include PLM education in their engineering programs in order to familiarize students with modern PD practices.
The remaining of this section will illustrate the PD process, CPD, and PLM to describe how these concepts complement each other. A thorough literature review will further examine insights conducted on teaching PLM education in response to the evolving demands of PLM training.

2.1 Product Development

Product development is the process of bringing a new product to market or enhancing an existing product to meet evolving market demands. Every aspect of a product is defined during product development, including the product's components and their associated costs. Companies that can effectively manage product development activities will be well-positioned to succeed in today’s competitive marketplace.

2.1.1 Product Development Process

Product Development (PD) process is a chain of activities that an enterprise Uses to conceptualize, design, and deliver a product to market. In other words, it is a series of activities that start with identifying a market opportunity aligned with the company's competitive strategy and technical capabilities and conclude with the production, sale, and delivery of the product. All these steps take place while considering all aspects that will evolve and keep the product competitive in the market until its discontinuity. As technologies advance, customer preferences shift, competitors launch new products, and the macroeconomic landscape changes. product development must adapt to these dynamics. This process involves addressing challenges such as balancing trade-offs, time pressure, financing, the satisfaction of societal and individual needs, and team diversity [7, Ch. 1]. The Output of a PD process may be a physical product, a service, or any mix of products and services. According to Eppinger's generic PD process model [7] illustration, the PD process comprises six distinct phases (Figure 2.1). The characteristics of each phase are explained in the following:

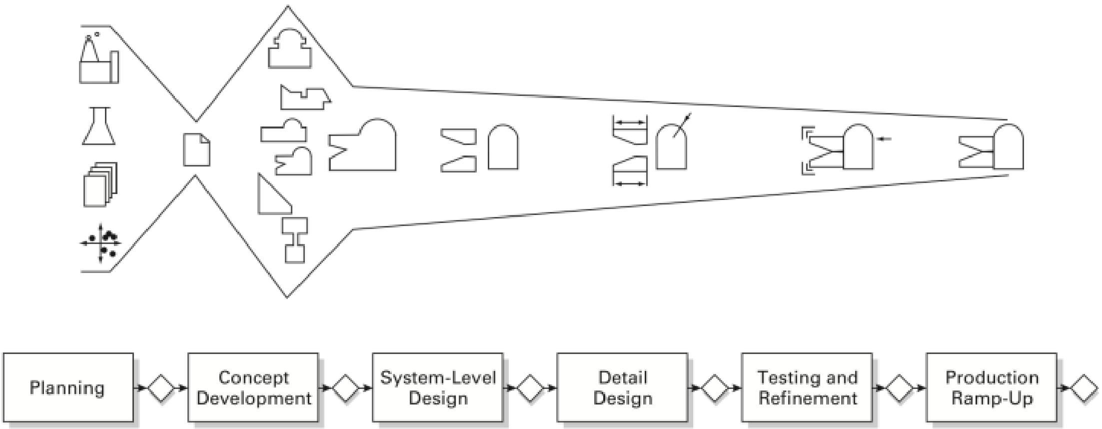


Figure 2.1 – The generic product development process [7, Fig. 2.2]

0. Planning:
1. Concept development
2. System-level design
3. Detail design
4. Testing and refinement
5. Production Ramp-UP

Phase zero begins with the project approval and launch of the product development process. This phase initiates with opportunity identification which includes an assessment of technology developments and market objectives [7, Ch. 2]. The opportunity identification explains a process of gathering, assessing, and selecting from a diverse array of product opportunities. [7, Ch. 3]. This process as shown in Figure 2.2 can be thought of as an innovation tournament, where only the most promising ideas succeed. In this tournament, the goal is to generate a large number of opportunities and efficiently eliminate those that are not worthy of further investigation. This process will feed the PD process with exceptional opportunities.

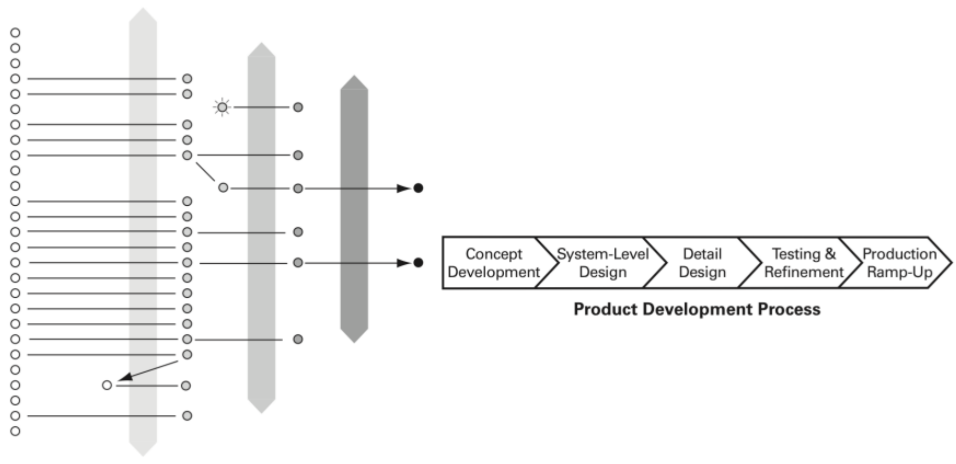


Figure 2.2 – Opportunity Identification Process [7, Fig. 3.4]

Once multiple promising opportunities are selected and prioritized, a portfolio of potential projects is created. Resources are allocated to these projects. The formulation of a product plan and the development of a mission statement therefore precede the actual product development process. Figure 2.3 illustrates the product planning process.

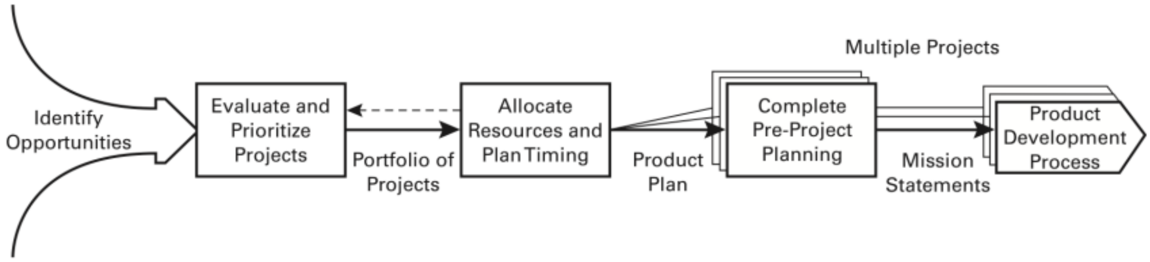


Figure 2.3 – The product planning process [7, Fig. 4.3]

However, the planning process is not essentially linear. The activities within this process are inherently iterative. The scheduling and budging often force a reassessment of priorities and further refinement of potential projects. Various departments carry out specific tasks to lay the foundation for the final project plan. The marketing team focuses on defining target market segments, while the design team evaluates the product architecture and assessing the potential of new technologies. In manufacturing, efforts are directed toward identifying production constraints and establishing a supply chain strategy. The finance team sets planning goals. The general management ensures effective resource allocation for the project. The project plan is reevaluated and modified frequently with latest information coming from these activities. Together, these activities create a cohesive strategy to launch the development process.
After project approval, a cross-functional team of people representing a wide range of expertise, conduct the mission statement document. This document summarizes the direction to be followed by the product development team. The document formulates a detailed definition of business goals, the target market, and the assumptions and constraints under which the development team will operate. Many more details are appended to this mission statement, Including the stakeholders, the environmental goals, service objectives, and specific technologies identified to be used in the project [7, Ch. 3].

0. Planning
1. Concept development:
2. System-level design
3. Detail design
4. Testing and refinement
5. Production Ramp-UP

In this phase, the target market's needs are identified, product specification defined, alternative product concepts are generated and evaluated, and one or more concepts are selected for further development [7, Ch. 5]. Figure 2.4 shows a process of activities in the concept development phase. Although this process is presented linearly, the activities often cycle back through each step multiple times. This process shows the distinction between customer needs and product specifications. Customer needs are qualitative desires, preferences, and requirements expressed by users, while product specifications are the measurable, technical parameters that define how the product will fulfill those needs. In other words, the specifications for a product which will be chosen to develop will depend on what is technically and economically feasible and what the competitors offer in the marketplace, as well as on customer needs.

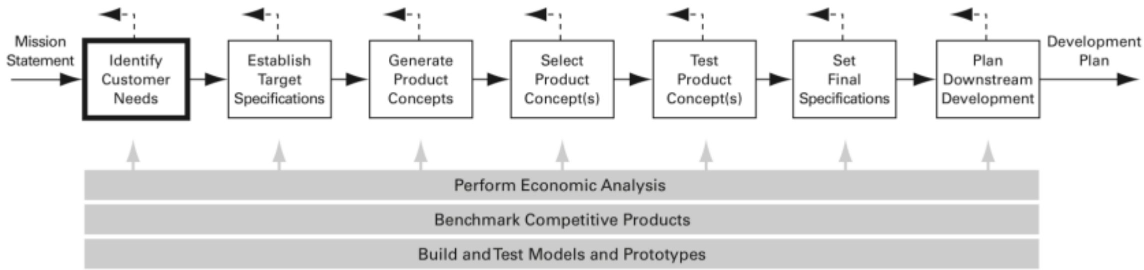


Figure 2.4 – The concept development process [7, Fig. 5.2]

The process starts with identifying customer needs. Eppinger [7] presents the process of identifying customer needs in five-step method:

1. Gather raw data
2. Interpret the raw data in terms of customer needs
3. Organize the needs into a hierarchy of primary, secondary, and (if necessary) tertiary needs
4. Establish the relative importance of needs
5. Reflect on the results and the process

Customer needs are generally expressed in the “language of customer”; however, while such expressions are helpful in developing a clear sense of the issue of interest to customers, they provide little specific guidance about how to design and engineer the product. To provide such information, development teams establish a set of specifications, which spell out in precise and measurable detail of what the product has to do to be commercially successful. The specifications must reflect the customer needs, differentiate the product from the competitive products, and be technically and economically realizable. Product specifications present what the team will attempt to achieve to satisfy the customer’s needs[7, Ch. 5].
Specifications are typically established at least twice:

- First, immediately after identifying the customer needs, the team sets target specifications. Target specifications represent the hopes and aspirations of the team, but they are established before the team knows the constraints the product technology will place on what can be achieved. The team may fail to meet some of these specifications and may exceed others, depending on the details of the product concept the team eventually selects. According to Eppinger [7, Ch. 5] the process of establishing the target specifications consists of four steps:
    1. Prepare the list of metrics
    2. Collect competitive benchmarking information
    3. Set ideal and marginally acceptable target values
    4. Reflect on the results and the process

- Second, after concept selection and testing the team develops the final specification. Final specifications are developed by assessing the actual technological constraints and the expected production costs using analytical and physical models. At this stage, the team must make difficult trade-offs among various desirable characteristics of the product. Specifications that originally were only targets expressed as broad ranges of values are now refined and made more precise. Eppinger [7, Ch. 5] presents the five-step process for refining the specifications is:

1. Develop technical models of the product
2. Develop a cost model of the product
3. Refine the specifications, making trade-offs where necessary
4. Flow down the specifications as appropriate
5. Reflect on the results and the process

At this stage, to leverage the best available knowledge of the market, customer needs, core product technology, and the cost implications of design alternatives, the specification process demands active collaboration from team members across marketing, design, and manufacturing functions within the enterprise.

Once the target specification has been set up, the concept generation process begins and results in a set of product concepts from which the team will make a final selection. A concept defines the form, function, and features of a product, typically accompanied by detailed specifications, a competitive product analysis, and an economic justification [7, Ch. 5]. A product concept is an approximate description of the technology, working principles, and form of the product. The degree to which a product satisfies customers and can be successfully commercialized depends to a large measure on the quality of the underlying concept. The development team will generate hundreds of concepts, a few number of which will merit serious consideration during the subsequent concept selection activity.

Concept selection involves evaluating potential solutions based on customer needs and other criteria, analyzing their strengths and weaknesses, and choosing one or more concepts for further investigation or development. A structured approach to concept selection significantly enhances the likelihood of a successful design. According to Eppinger [7, Ch. 8], this process can be divided into two stages:

- In the concept screening phase, a reference concept is used as a benchmark to evaluate various alternatives against predefined selection criteria. This stage employs a coarse comparison system to eliminate fewer promising options and focus on a narrower set of viable concepts.
- The concept scoring phase builds upon concept screening by using weighted selection criteria and a more refined rating scale to thoroughly assess the remaining options. However, this stage can be bypassed if concept screening identifies a clearly superior concept.

In addition to concept selection, concept testing plays a crucial role in the product development process by directly involving potential customers from the target market. Through this process, customers provide feedback on a description of the product concept, helping to validate its alignment with their needs and expectations[7, Ch. 9].
Concept testing serves multiple purposes. It verifies whether the product concept sufficiently addresses customer requirements, evaluates the sales potential of the concept, and collects valuable insights for refining the concept further. By integrating customer input at this stage, concept testing complements the structured approach of concept selection, ensuring that the chosen concepts not only meet technical and strategic criteria but also resonate with the target audience [7, Ch. 9].

0. Planning
1. Concept development
2. System-level design:
3. Detail design
4. Testing and refinement
5. Production Ramp-UP

The system-level design phase includes defining the product architecture, subsystems, components initial designs, detail design planning, and usually initial plans for the production system. [7, Ch. 10].
The product's architecture forms the base for future development steps. Decisions about product architecture have significant impacts on factors like product adaptability to change, the variety of versions, the use of standard parts, performance, manufacturability, and overall project management. Given the extensive implications of architectural choices, collaboration between marketing, manufacturing, and design teams is critical. A key characteristic of product architecture is the extent to which it is modular or integral:

- Modular architectures feature physical chunks where each chunk implements a distinct set of functional elements and interacts with other chunks in well-defined ways. Modular designs can be categorized into three types:
    1. slot-modular: Each module fits into a specific location.
    2. bus-modular: Modules connect to a common interface or "bus."
    3. sectional-modular: Modules connect in a flexible, combinable manner.
- Integral architectures spread the implementation of functional elements across chunks, resulting in less clearly defined interactions between them.

Modular and integral architectures require different project management approaches. Modular designs require meticulous planning at the system level, while detail design focuses on ensuring each module meets the target performance, cost, and goals for their components. In contrast, integral designs need less upfront planning but demand greater integration, coordination, and conflict resolution during detail design.
Architectural decisions are also closely tied to platform planning. The product platform is the set of assets shared across a set of products. Components and subassemblies are often the most important of these assets. An effective platform can allow a variety of derivative products to be created more rapidly and easily, with each product providing the features and functions desired by a particular market segment. Therefore, platform planning balances differentiation and commonality to address varying market segments with multiple product versions [7, Ch. 10]. However, the crucial strategic decision—whether a project will develop a derivative product from an existing platform or create an entirely new platform—occurs during phase zero (planning). Since platform development projects typically require 2 to 10 times more time and resources than derivative product development, it is not feasible for a company to approach every project as a new platform [7, Ch. 4].
The output of this phase usually includes a geometric layout of the product, a functional specification of each of the product’s subsystems, and a preliminary process flow diagram for the final assembly process [7, Ch. 2].

0. Planning
1. Concept development
2. System-level design:
3. Detail design:
4. Testing and refinement
5. Production Ramp-UP

Subsequent detailed design activities play a crucial role in the ongoing evolution and refinement of the product's architectural details. This phase involves creating comprehensive specifications for all aspects of the product, including geometry, materials, tolerances, tools, and process plans for every component.
This phase integrates key considerations such as:

1. Industrial design: Ensuring the product's aesthetics, ergonomics, and user experience align with customer expectations.
2. Design for manufacturing (DFM): Optimizing the design to simplify production processes, reduce costs, and improve manufacturability.
3. Design for the environment (DFE): Incorporating sustainable practices by minimizing environmental impact through material choices, production methods, and end-of-life considerations.

The detailed design phase ensures that the product transitions smoothly from concept to production, maintaining alignment with technical, customer, and environmental requirements.

0. Planning
1. Concept development
2. System-level design
3. Detail design
4. Testing and refinement:
5. Production Ramp-UP

This phase focuses on prototyping, involving the construction and evaluation of multiple preproduction versions of the product. A prototype is a representation of the product that approximates one or more dimensions of interest. Prototyping is an essential aspect of product development, as it provides opportunities for building and testing approximations of the final product.
Prototypes serve several critical purposes:

1. Learning: Gaining insights into the product's performance, usability, or functionality.
2. Communication: Conveying ideas and concepts to stakeholders.
3. Integration: Ensuring compatibility between various subsystems and components.
4. Milestones: Demonstrating progress and validating key design decisions.

Advancements in 3D CAD modeling and 3D printing technologies have significantly reduced the time and cost associated with creating and analyzing prototypes, accelerating the development process. Eppinger [7, Ch. 14] presents four-step method for planning prototypes:

1. Define the purpose of the prototype: Clearly outline what the prototype aims to achieve or test.
2. Establish the level of approximation: Determine how closely the prototype will resemble the final product in form, function, or performance.

3. Outline the experimental plan: Specify the tests or evaluations that will be conducted using the prototype.
4. Create a schedule: Develop a timeline for procurement, construction, and testing activities.

Prototyping is a critical phase that bridges the gap between conceptual design and production, ensuring the product is functional, manufacturable, and aligned with customer needs.

0. Planning
1. Concept development
2. System-level design
3. Detail design
4. Testing and refinement

5. Production Ramp-UP:

The production ramp-up phase is a critical stage in product development, marking the transition from prototype to full-scale production. During this phase, the product is manufactured using the intended production system, the workforce is trained, and any remaining issues in the production process are resolved.

The primary goal of this phase is to gradually increase production volumes to meet market demand while maintaining or improving product quality, process efficiency, and cost-effectiveness. Key activities in this phase include:

1. Testing and validating production systems: Ensuring equipment, processes, and workflows function as intended.
2. Refining manufacturing workflows: Addressing bottlenecks and inefficiencies to optimize production.
3. Training the workforce: Preparing employees to operate new systems or technologies.
4. Coordinating with suppliers: Ensuring a steady supply of materials and components to support production.
5. Incorporating customer feedback: Making last-minute refinements to the product based on real-world input.

This phase often presents challenges, such as maintaining stringent quality standards, managing uncertainties, and meeting tight time-to-market deadlines. Effective management of the production ramp-up phase is crucial to ensuring a seamless transition to mass production, minimizing costs, and achieving a competitive edge in the marketplace. Although standard approaches to product development are widely discussed in the literature, each organization implements the process differently, tailoring it to meet its specific needs and goals.

There are several perspectives on how to conceptualize the product development process:

- One approach views it as the progressive refinement of ideas, starting with the creation of a broad set of alternative product concepts. These alternatives are gradually narrowed down and specified in increasing detail until a product emerges that can be reliably and repeatably manufactured.
- Another perspective considers the product development process as an information processing system. This view emphasizes the systematic transformation of inputs—such as objectives, strategic opportunities, and available technologies—into outputs that meet customer needs.

Despite these differing frameworks, the overarching goal of any product development process remains consistent: to create value for the customer [7, Ch. 2].

Notwithstanding the structured nature of PD, one of the most critical success factors is the adoption of effective project management, which provides the necessary framework to handle the complexities of the process [7, Ch. 19]. The two disciplines are closely linked and show many similarities. Projects are composed of tasks interconnected by dependencies, which may be sequential, parallel, or coupled. The longest chain of dependent tasks defines the critical path, which determines the minimum time required to complete the project.
To manage these dependencies and timelines effectively, several tools and techniques are commonly employed:

- The Design Structure Matrix (DSM): Used to map and analyze task dependencies.
- Gantt charts: Visualize the timing and scheduling of tasks.
- PERT charts: Represent both dependencies and timing, often used to compute the critical path and identify potential bottlenecks.

Project planning is a cornerstone of product development, yielding essential outputs such as:

- A detailed task list.
- A project schedule.
- Staffing requirements.
- A project budget.
- A risk management plan.

These elements often form the foundation of a contract book, ensuring alignment and accountability among stakeholders. The planning phase is also where most opportunities to accelerate the project timeline emerge, whether by reorganizing tasks, reallocating resources, or streamlining workflows.
Project execution focuses on coordination, tracking progress, and addressing deviations from the plan. This stage ensures that the development stays on track and adapts to any unforeseen challenges.
Finally, evaluating project performance at the conclusion of the product development process fosters continuous improvement, enhancing both individual and organizational capabilities. This systematic approach to project management is vital for delivering successful products that meet customer needs while adhering to time and budget constraints.

2.1.2 Collaborative Product Development (CPD)

As a result of market globalization, companies now operate across both geographical and organizational boundaries. Many have expanded their global presence by establishing branches worldwide, franchising operations, and exporting products. Additionally, a surge of imports from low-cost regions has driven down the prices of goods in industrialized countries. In response, companies have sought to reduce costs by outsourcing not only production but also product development and service activities to these low-cost regions. This shift in outsourcing allowed businesses to concentrate on activities that offer the greatest strategic importance or competitive advantage [8, Ch. 3]. Consequently, product development has become not only more critical but also significantly more complex. To manage the challenges of shortening product lifecycles, rapid technological advancements, and compressed development timelines, organizations must harness a diverse range of skills and expertise while coordinating resources at a high level [9]. The flow of information and materials has intensified as companies increasingly collaborate and coordinate across globally distributed enterprises.
To meet these demands, businesses have adopted strategies that enhance both efficiency and innovation, one of which is Collaborative Product Development (CPD)—a strategic approach specifically designed to tackle these challenges. Bruce et al. [10] emphasize the

importance of adopting a flexible and adaptable approach to managing collaboration effectively. In line with this, CPD helps ensure that ideas are transformed into market-ready products in a way that is both efficient and adaptive to changing environment. CPD is the process of developing products through collaboration between individuals and teams of different collective expertise and geographically dispersed such as internal departments, external partners, and suppliers. It utilizes the structured sharing of product knowledge, technology, skills, and resources from different sites in order to achieve enterprise goals. It empowers organizations to drive innovation, shorten development timelines, and respond quickly to evolving customer needs.
As highlighted in [10], the nature of CPD is dynamic and evolving. The effectiveness of CPD depends on robust systems capable of managing the complexities of collaboration. While CPD emphasizes "how" collaboration happens and ensures that the right people and teams are working together, information systems provide the "what" and "where" by managing all the data, processes, and resources throughout the PD process. S. Mathrani et al. [11] identifies success factors within four contexts:

- Management
- cross-functional teams
- processes
- supporting tools

These factors interact with each other to achieve improved CPD performance and project outcome. For CPD to succeed, it requires effective management to facilitate information flow, cross-functional teams that integrate diverse expertise, and well-defined processes supported by the right tools to enhance productivity and innovation. These factors work together to ensure smooth coordination and efficient development.

In other words, CPD is the application of team-collaboration practices built upon the systems engineering, project/program management and concurrent engineering (CE) [12]. While (CE) focuses on structuring products, workflows, teams, and organizations, CPD emphasizes creating environments that facilitate effective, seamless information sharing and collaboration among peers involved in the PD process [12].

The study performed by R. Del Rosario et al. [13] describes CPD as an “extension” of Concurrent Engineering (CE), therefore it can be expected that many of the benefits attained by CE practitioners would be achieved with a CPD environment. The authors declared that CPD encompasses concurrency, attention to the life-cycle, suppliers, and information technology, all while maintaining a customer-focused environment. They investigated the application of CE and CPD in R&D section and confirm the significant contribution in improving R&D process through CE and CPD.

D. Litter et al. [9] identifies that benefits of CPD include faster development, cost reduction through shared expenses, and access to specialized skills and knowledge. On the other hand, the associated risks include information leakage, loss of control over the PD process, and increased costs and time required to manage the collaboration. The authors highlight several factors that play a crucial role in determining whether a collaboration will succeed or fail. These factors can significantly influence the outcome of collaborative product development, sometimes even undermining the anticipated benefits. Table 2.1. contains a summary of main factors, grouped by theme contributing to have a successful CPD project.

Factors	Description
Setting up the collaboration	Investment in selecting the right collaborative partner, setting clear goals, defining responsibilities and accountabilities, and establishing boundaries.
Process management	Frequent monitoring of progress and frequent consultation between partners, marketing, and technical personnel. It triggers development of trust between parties.
Allocation of resources	Financial and staff resources
Personnel involvement	Fosters accountability and effective engagement.
Ensuring equality	Prevents discontent and fosters fair contribution to build stronger partnerships
Past experience of collaboration management	Offers valuable knowledge to anticipate and handle challenges
Accessing external factors	Paying attention to monitoring environmental changes

Table 2.1 – Factors contributing to collaboration "success" identified [9]

A product development project may encounter challenges such as competing priorities, resource constraints, and other complexities. However, the establishment of clear communication channels and effective conflict resolution mechanisms can mitigate these challenges, facilitating smoother project execution and enhancing overall outcomes.
As defined by D. Litter et al. [9], the Champion, i.e., an individual or a team, advocates for the project, navigates challenges, and ensures its continued progress. In an internal setting, a champion could be a leader who bridges communication gaps between teams and, team members, secures resources, and maintains focus on the shared objectives.
The focus should be on clearly defining objectives, roles, and responsibilities to ensure successful collaboration. This clarity regarding each participant's contributions and expected outcomes is essential for preventing confusion and minimizing conflict.

2.2 Product lifecycle management (PLM)

"Product Lifecycle Management (PLM) is the business activity of managing, in the most effective way, a company's products all the way across their lifecycles; from the very first idea for a product all the way through until it is retired and disposed of." [8, Ch. 1]

PLM connects disparate data, processes, disciplines, functions, resources, and people, managing them in an integrated way. At the highest level, the objective of PLM is to increase product revenues, reduce product-related costs, optimize the value of the product portfolio, and maximize the value of current and future products for both customers and shareholders [8, Ch. 1].

Three important concepts in PLM are:

P: This letter represents the product. A product can be a physical good in various shapes and sizes, a service, a package of services, a bundle of products and services, or a solution containing multiple products or both products and services [8, Ch. 1]. A product also represents a key part of the company's value proposition and is designed to meet the needs of a target audience, standing as the central piece of what the business offers to the market [8, Ch. 1]. The product is the source of a company's revenue, generated through an ongoing stream of innovative new and upgraded products [14, Ch. 1].

L: This letter represents the lifecycle. There are five phases in the product lifecycle as represented in Figure 2.2. In each of these five phases, the product is in a different state and specific activities take place. According to [14, Ch. 1], in the ideation phase, is just an idea. In the definition phase, the ideas are being converted into a detailed description. By the end of the realization phase, the product exists in its final form. During the use/support phase, the product is being used by a user. Eventually, the product is retired by the company or disposed of by the customer because it is no longer useful. It may be recycled or get dumped.

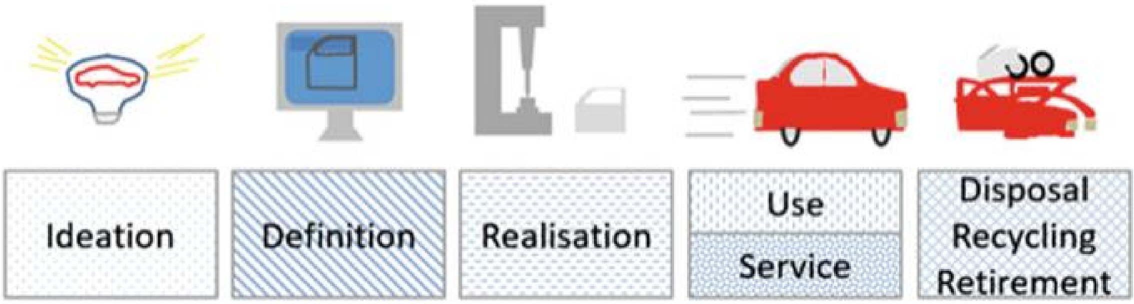


Figure 2.2 - The 5 phases of the product lifecycle [14, Fig. 1.3]

The specific activities that take place across the product lifecycle vary from one industry sector to another. However, whatever the specifics of a particular company or industry, its activities can be mapped, in some way, to the five phases of the product lifecycle shown in Figure 2.2 [14, Ch. 1].

M: This letter stands for management. The product needs to be managed at each phase of its life to create value for the company and its users. All phases of life of a product are affecting the product's value proposition. Product ideas need to be managed to make sure that they are not lost or misunderstood. When product id being defined, it has to be managed to be sure the resulting product meets customer requirements. When realizing the product, it is needed to make sure that correct version of the product definition is used during production. During use phase, product must be correctly maintained, taking account of its results when its being used. Also, its disposal must be managed to guarantee cost efficiency and least danger to environment. PLM manages the product "from dawn to dusk" [14, Ch. 1].

As discussed in [14, Ch. 1] PLM covers a wide range of activities in a product's lifecycle, including: project management, design of system architecture, management of release, change, versions, and impact analysis, bill of material (BOM), design of the manufacturing process, and any other section to contribute to reaching its objectives.
PLM provides strategic and operational benefits across a wide range of products and industries, with specific advantages varying for each individual company throughout the product lifecycle [14, Ch. 1]. These benefits generally include: revenue increase, cost reduction, time reduction, quality improvement, operational efficiencies, enabling opportunities [14, Ch. 1].

2.2.1 Emergence of PLM

The concept of Product Lifecycle Management (PLM) has evolved significantly over time. As J. Stark describes in [8, Ch. 5], the Market was primarily local in the past, with customers having limited choices. After the Second World War, social trends moved from rural life to city-dwellers. Demand exceeded production capability. Getting products to the market was the main focus and at the end of product life, products went to a tip, dump, field, or the sea.

J. Stark [8, Ch. 5] **presents the following timeline:**

1950s–1960s: Numerical Control (NC) and Computer-Aided Design (CAD) technologies were pioneering technologies that marked the beginning of automation in manufacturing and design. Each played a transformative role in its respective area, laying the groundwork for the highly automated processes we have today.

1960s-1970s: Computer power increased. CAD and, Computer Aided Manufacturing (CAM) developed and methodologies like Material Requirements Planning (MRP), Just-in-Time (JIT), ISO 9000 and TQM were adopted more widely in the industry.

1970s: System development methodologies were introduced to manage software development. Methodologies like Waterfall system development that divided projects into phases, and used deliverables and approvals to maintain control.

1980s: Methodologies like Concurrent Engineering to streamline manufacturing and development emerged to overcome problems in engineering and manufacturing serial activities. Also, MRP evolved to MRP2. CAD and CAM systems were no longer centralized. As companies started to be swamped by engineering data, the first Engineering Data Management (EDM) systems appeared. The development of ISO 10303, an ISO standard for the representation and exchange of product data, started.

1990s: A surge in globalization and greater competition in market happened. Companies started to outsource to low-cost countries to remain competitive. The link was made between CAD, management of product data, and business processes. Many companies started Business Process Reengineering initiatives, significantly streamlining many business activities. The World Wide Web, e-commerce, B2B and trading exchanges appeared. MRP2 evolved to ERP. CAD functionality became a commodity. EDM systems were relabeled as PDM systems and brought some order to all the product data. Web-based sales configurators allowed customers all over the world to purchase the exact products they wanted. With development outsourced to different locations, and developers needing to work closely together, Concurrent Engineering, the Web and CAD morphed into Collaborative Development.

1990s: Sustainability and environmental awareness increased and led to agreements like the Kyoto Protocol (1997) aiming to reduce greenhouse gas emissions and questions arising around the ecological impact of global production and transportation. As a result, companies were compelled to find new ways to manage and minimize the environmental impact of their products from the development stage onward. This pressure, coupled with the need to streamline processes, pushed firms to improve efficiency and speed in order to reduce costs while meeting environmental standards.

2000s: Stricter regulations and corporate accountability movements emerged to prevent corporate fraud and ensure transparency in business practices. The financial crisis and subsequent recession had a profound impact on global economies, reshaping corporate strategies and governance. In this context, companies not only faced increased pressure to reduce their environmental footprint but also had to navigate heightened scrutiny regarding financial practices. This combination of regulatory demands and economic challenges further fueled the need for more efficient, transparent, and sustainable business practices, pushing companies to innovate and adapt in order to remain competitive and compliant in an evolving global market.

2000s: The term "Web 2.0" appeared. The adoption of Software as a Service (SaaS) and Cloud computing started. Cloud-based CAD and PDM solutions appeared. The Internet of Things (IoT) emerged. "Develop Anywhere, Sell Anywhere, Manufacture Anywhere, Support Anywhere" (DASAMASA) raised describing modern global manufacturing. The environment in which companies managed products changed enormously between the middle of the twentieth century and the end of the twentieth century.

By the end of the 20th, there were computers everywhere, the Cold War between the capitalist West and the communist East was over, and globalization had happened .Quality and the Environment were major issues. However, these issues were not limited to a particular region or sector; they affected consumers, organizations, and manufacturers across the globe.

In the 21st Century, the global economy was asking companies to present desired products with high quality and low cost while maintaining competitiveness in the market. However, companies were facing intense competition, ever-changing consumer preferences, and financial fluctuations.

As demonstrated by the timeline, the product environment went through fundamental changes, as cost, quality, revenue, shareholder value, and market share became ongoing concerns for companies, making decision-making increasingly complex [8, Ch. 5]. This shift reflects a broader paradigm change, where existing strategies and practices are no longer aligned with the new realities of the market. As [14, Ch. 1] suggests, a paradigm shift occurs when the established framework is no longer suitable for the evolving domain, and while the core parameters may remain the same, their values change in response to new challenges and demands. In response to the pressures from the evolving domain, the PLM paradigm emerged in the early 21st century. Figure 2.3 shows key parameter values changed during this paradigm shift [14, Ch. 1].

#	*Parameter*	*PLM Paradigm*	*Previous Paradigm*
1	Organisation of Work	Business Processes	Departmental
2	Orientation	Business	Technical
3	Information Storage, Calculation and Communication	Digital	Analogue, paper
4	Span of Interest	Complete Product Lifecycle	Design to Factory Gate
5	Value of Product Data	High	Low
6	Management Approach	Holistic, joined-up	Piecemeal, separate
7	Focus	Product-focused	Unfocused

Figure 2.3 - Parameters and values of the PLM paradigm [14, Fig. 1.6]

2.2.2 Adoption of PLM

As companies recognized the limitations of the traditional management approach, the transition toward a Product Lifecycle Management (PLM) framework began to take shape. This shift was essential to keep up with ever evolving product environment.
As outlined in [14, Ch. 1] this shift includes:

- **Organization of Work:** Companies began to organize work around business processes rather than traditional functional departments. According to [14, p. 14]:
  "A business process is an organized set of activities, with clearly defined objectives, scope, roles, inputs and outputs, which creates business value. Usually, people from several functions are involved in the activities of a business process."
  Examples of these processes are New Product Development (NPD), Engineering Change Management (ECM), and Product Portfolio Management (PPM).
- **Orientation: From Technical to Business:** The focus of product management changed from a technical orientation to a business orientation aiming at business objectives. With technical orientation, companies had a mixture of many and often incompatible methods and approaches. Each of them had technical objectives (such as "design better"), not business objectives (such as "increase product revenues"). PLM is carried out to meet business objectives of increasing product revenues, reducing product-related costs, maximizing the value of the product portfolio, and maximizing the value of current and future products for both customers and shareholders. In addition to business objectives related to improved financial performance, PLM also includes objectives related to time reduction and quality improvement.
- **Information Calculation, Storage and Communication:** Products are managed digitally throughout their lifecycle. Calculations are performed by computers, information is stored in digital memory, and communication occurs over digital networks.
- **Span of Interest:** Span of interest changed from "Design to Factory Gate" to "Complete Product Lifecycle" to meet environmental requirements and Circular Economy concepts.
- **Value of Product Data:** The "product data" concept was introduced. It started to be seen as Intellectual property and a strategic corporate asset.
- **Management Approach:** The management approach changed from "divide and rule" to "joined-up" and "holistic". It addresses products, data, applications, business processes, people, work methods, and equipment together.
- **Focus:** The focus changed to the product and the customer.

Following the shift to the PLM framework, its application has expanded across a wide range of industries that develop, produce, and support products. PLM is now used in sectors such as discrete manufacturing, process manufacturing, distribution, and services, as well as in research, education, military, and government organizations. PLM is used in all sizes of companies ranging from large multinational corporations to small and medium enterprises.
While the particular PLM requirements of companies of different sizes may differ, the fundamental requirements do not. The core needs remain consistent: managing products, overseeing product data, controlling development and support processes, and facilitating the exchange of information.

2.3 The Intersection of CPD and PLM

As discussed in previous sections, the PLM environment supports a wide range of applications, one of which is Collaborative Product Development (CPD) management. Studies such as [15] and [16] explore the application of PLM in CPD, revealing that PLM facilitates CPD by offering a structured and integrated framework to manage the complexities of collaboration throughout the product lifecycle.

This application of CPD in PLM is achieved through several key features and functionalities:

- **Holistic Approach:** In this approach, PLM addresses many resources in the product environment such as products, data, applications, processes, people, work methods, and equipment [8, Ch. 1]. By providing a centralized platform for information management and collaboration, PLM ensures that everyone involved in product development has access to accurate and up-to-date information, regardless of their location or function. This structured environment for collaboration helps prevent misunderstandings, reduces the risk of conflicting information, and promotes better decision-making.

- **Joined-up Paradigm:** PLM joins up many previously separate and independent processes, disciplines, functions, and applications, each of which, though addressing the same product, previously had its own vocabulary, rules, culture, and language [8, Ch. 1]. It brings together activities like product development and product support, which were often handled by different teams with varying vocabularies, rules, cultures, and languages. This integration fosters a more cohesive and streamlined approach to product management.

- **Collaboration Management Tools:** These tools allow geographically dispersed teams and individuals to collaborate securely and efficiently in a structured virtual environment, ensuring access to the most up-to-date product data [8, Ch. 10]. They include a wide range of functionalities. Some examples are shared spaces, document collaboration, visualization tools, and social collaboration tools. Expanding on this, F. Stelian et al. [15] propose a method for PD using different collaborative tools in a PLM Platform. They explore how collaborative tools can improve product development by overcoming challenges such as geographical barriers, data management complexities, and time constraints. The authors conclude that their proposed method for integrating collaborative tools into a PLM platform offers a practical solution for overcoming the challenges of modern product development, ultimately contributing to a more efficient and innovative design process. In addition, Hai-yue LI et al. [16] established a conceptual PLM system architecture specifically designed to support collaborative product development. The focus of the research is the product information modeling architecture and the information system infrastructure.
- **Role-based Collaboration:** The activities of a process need to be clear, as do the participants in the process, the roles of the participants, the information they use and create, the tools they use, and the owner of the process. Anything that isn't clear will lead to hesitation and confusion. Time and money will be wasted [8, Ch. 9] . PLM provides functions to specify clear roles and responsibilities of participants all across the product life cycle [8, Ch. 30].
- **Project management Tools:** Project management frameworks provide the tools and processes for planning, monitoring, and controlling the development stages, ensuring that timelines, budgets, and quality standards are met. These frameworks enable teams to break down complex tasks into manageable increments, prioritize work, and quickly adapt to changes or challenges in the PD process. The project management framework plays a pivotal role in CPD. The use of project management frameworks within CPD helps maintain alignment across teams, track progress, and address issues proactively.

In other words, PLM acts as a central repository for product information, ensuring consistency, traceability, and accessibility for all stakeholders involved in CPD. CPD relies on the organizational and technological infrastructure that PLM offers, such as centralized data management, process standardization, and integrated workflows. On the other hand, PLM gains its full potential when applied in a collaborative setting, enabling real-time communication, reducing errors, and facilitating better decision-making across the product lifecycle. By integrating these frameworks, organizations can streamline their PD processes, enhance collaboration, and achieve greater efficiency, ensuring that products are developed in the most effective and cost-efficient manner possible.

2.4 PLM education

According to J. Sauza Bedolla et al. [3], PLM is known as the only stable channel of product information throughout its lifecycle and is an essential tool for coping with the challenges of increasing global competition. Business sectors are increasingly adopting PLM systems to optimize their PD processes. Consequently, the need for engineers and professionals able to work effectively in PLM is increasing as well. New engineers must be familiar with the PLM philosophy and use its tools to work effectively in collaboration in teams around the world. This has led to the emergence of PLM education as a critical field of study in academia and industry.

This section aims to explore the current state of the art in PLM education. The focus is on the integration of PLM software into academic curricula to provide hands-on training. We explore insights into how educational institutions are addressing the evolving demands of PLM training, ensuring that graduates and professionals are well-equipped to contribute to the digital transformation of product development processes.

A research conducted by the Polytechnic of Turin [2] highlights the current state of PLM education in universities, focusing on Italy and France. The research revealed a wide variety of approaches to teaching PLM across different universities and countries.

Furthermore, J. Sauza Bedolla et al. [3] implemented a Visualization Model (VM) that provides a graphical representation as an effective tool for teaching PLM principles and fostering collaborative design skills in the "Fundamentals of Machining Design and Drawing" (FMDD) course in Polytechnic of Turin. Positive feedback from students indicated that the VM was helpful in understanding PLM concepts and the collaborative nature of product design.

EAFIT University [4] also presented a didactic manufacturing plant where design and production are managed in a PLM environment. The didactic plant puts students in the difficulties and constraints that they will find in the industry. Moreover, collaboration skills are stimulated by working in teams and sharing information on the PDM platform.

Purdue University in [5] redesigned an already existing course named "Industrial Applications for Simulation" to a course emphasizing active learning and soft skills development. They used an industry-sponsored project to teach PLM concepts, emphasizing a practical, hands-on approach. The course prepared students with not only the knowledge of PLM but also the capability of problem-solving, communication, self-motivated teamwork, and leadership.

Oakland University has created a center for PLM education to develop an academic infrastructure supporting PLM, ERP, and MES [6]. They aim to provide IT infrastructure to host PLM applications on cloud for universities and educational institutions seeking to add

PLM training in their curricula. This infrastructure will remove the burden of building an IT infrastructure from institutions, which will just need to manage applications and data generated by PLM tools.

Universitat Politecnica de Catalunya [17] has designed and implemented a PLM strategy for engineering students. They aimed to give students practical experience with PLM concepts and software used in real-world product development scenarios by simulating an industrial product lifecycle within their coursework.

By exploring these insights, we notice that PLM has a flexible architecture that integrates all elements in the product environment and allows us to adapt and extend the system to support various purposes. As such, the PLM itself can be designed for students and can also serve as an appropriate tool for teaching other important concepts in product development as well, concepts as collaborative product development (CPD) and project management.

This thesis aims to provide a solution to utilize a PLM platform in educational settings, in which students gain hands-on experience in PD scenarios and experience CPD practices. This solution bridges theoretical concepts with practical applications, effectively preparing future engineers and designers for industry demands. Students will learn PLM philosophy and use its tools to work effectively in collaboration with teammates. The end goal is to prepare students with not only the knowledge of PLM but also the capability of problem solving, communication, self-motivated teamwork, and leadership.

3. Framework Development

The proposed framework addresses the need for skilled engineers by providing students with hands-on experience in collaborative product development (CPD) scenarios by means of a PLM platform, integrating theoretical concepts with practical applications. The project aims to develop a versatile solution applicable to any educational course focused on product development, regardless of the type of product. It also incorporates features for teachers to monitor progress and assess outcomes effectively. The solution will let students develop products and experience the complexities of product design and management. This framework helps to ensure that graduates are prepared for the challenges and opportunities of industry.

The system was designed to satisfy the following goals:

- Introducing the PLM concept to students.
- Providing hands-on experience with the PLM platform at every stage of product development.
- Emphasizing teamwork through various activities, throughout product development phases.
- Providing an easy-to-use platform for students’ team work.

Once the goals were set, Eppinger's product development model was employed as an approach to design the framework, which was divided in phases of Planning, Concept Development, System Level Design, Detail Design, Results to systematically address all relevant contexts and objectives.

The framework was validated by deploying it through Aras PLM Innovator in a Product Development course at Polytechnic of Turin. This course originally introduces the Product Development Process based on Eppinger's model [7] and Product Lifecycle Management. The integration of Aras PLM in this course supports teaching product development process in a CPD framework. It enhances the educational experience and prepares students to navigate the complexities of product development and lifecycle management.

The remaining of this chapter describes each phase in details.

3.1 Phase 0: Planning

The planning process takes place before substantial resources are employed and establishes the foundation for the implementation process. It involves defining the idea, goals, stakeholders, scope, constraints, key assumptions, and planning the implementation road map.
In terms of infrastructure, convenient access to necessary resources was granted to ensure a solid starting point for the project. A server with limited configurations was provided by the Polytechnic of Turin to host the community version of Aras Innovator. This initial setup allowed for an exploratory phase focusing on familiarizing with Aras applications and evaluating their capabilities. Based on the outcomes of this initial research phase, the potential acquisition of a more advanced version of Aras Innovator would be considered for further investigation and implementation in the Detail design phase.

3.1.1 Data Collection

The data collection process included both interviews and document analysis. Several qualitative, dialogue-based interviews were conducted with a diverse group of stakeholders, including professors and Ph.D. students from the Faculty of Management and Production Engineering. These interviews, along with course documents and sample student reports, provided multiple perspectives to construct a comprehensive overview of the case. Additionally, Aras documentation played a crucial role in understanding the Aras platform and its functionalities. The collected data at this stage helped define the project's scope effectively.

3.1.2 Overview of the case

The team project of the course of interest focuses on product development of Sliding Door Trolleys, an existing product in the market. This course explores the fundamental principles in product development based on Eppinger's Model [7], aiming to equip students with a profound understanding of the product development challenges in the market. Through a combination of theoretical instruction and teamwork-based projects students engage directly with real-world product development scenarios and build soft skills, critical thinking, and problem-solving skills.

The course focuses on the five phases of Eppinger's model: **Planning**, **Concept Development**, **System Level Design**, **System Design**, and **Test and Refinement**. The process of student project is depicted in the Business Process Model and Notation (BPMN) standard. The activities done by tutor and students within each phase are grouped and the expected outcome of each of them linked by association link to the document object as presented in the figure 3.1

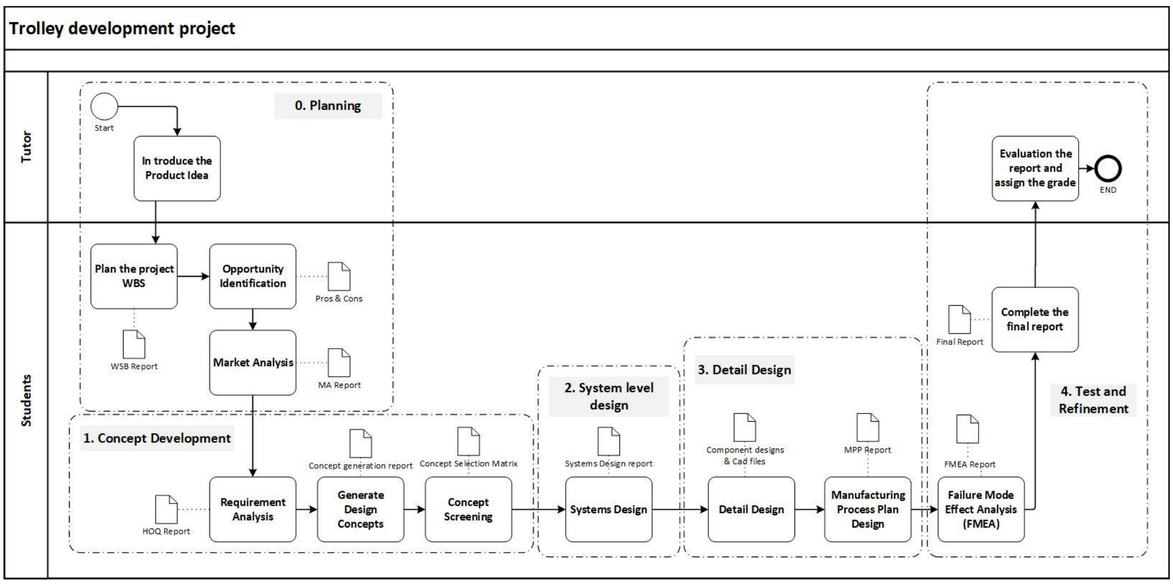


Figure 3.1 – Trolley development project process

0. Planning:

The project begins with the tutor introducing the product concept, followed by students creating a preliminary Work Breakdown Structure (WBS) to organize tasks and responsibilities. Students start the project researching sliding door mechanisms, comparing rotating and sliding models. Then, they perform an opportunity analysis to evaluate different models, leading to a pros and cons comparison and to the selection of a suitable design.
Next, a market analysis is conducted to assess the sliding door trolley market, including component costs and market size, aiming to identify potential customer demand. Using market penetration metrics and mathematical calculations, students determine the target market.
Based on this analysis, they develop a product policy following the Eppinger model, detailing the product description, goals, regulations, and target markets. After confirming profitability, if necessary, students modify WBS and organize tasks and responsibilities to ensure the project is run efficiently.

1. Concept development:

At this stage, students identify and prioritize customer needs using the House of Quality (HOQ) method. They collect, rank, and translate customer requirements into technical specifications. The HOQ analysis highlights the highest-priority requirements, providing a foundation for the system-level design phase by emphasizing the most critical features for product development.
With these technical requirements in hand, students begin designing trolley concepts. The design process balances "What we want to achieve" with "How we choose to satisfy the need." Multiple design concepts are generated, and the Concept Selection Matrix method is applied to evaluate them. The primary goal of this screening is to eliminate concepts unlikely to create value and focus on that worthy of further exploration. The most promising configurations are then analyzed for assembly time and cost, and the design offering the best combination of efficiency and affordability is ultimately selected.

2. System Level Design:

In this phase, students develop the overall product architecture and structure, translating technical requirements into a comprehensive system design. They define key subsystems and components, focusing on how each element interacts to fulfill the product's functionality. This phase involves creating rough geometric layouts, interface specifications, and mapping functions to physical components.
Students also establish the relationships between subsystems and assess potential trade-offs to optimize performance and manufacturability. They generate initial system configurations and evaluate different design alternatives to ensure alignment with customer needs and design goals. By the end of this phase, they finalize a system-level layout that serves as a blueprint for the subsequent detailed design work.

3. Detail design:

They define the shapes and dimensions for each component of their design. Based on the finalized product design, they generate a Bill of Materials (BOM), which represents a hierarchical structure detailing all the parts and sub-assemblies required for the assembly of the sliding door trolley.

In this phase, they also develop a Manufacturing Process Plan (MPP) and make "Make or Buy" decisions. A make-or-buy decision involves choosing between manufacturing a product in-house or purchasing it from an external supplier. This requires a detailed cost comparison to determine the most economical option. Additionally, they design a sequence of operations for manufacturing, specifying processes for parts produced in-house (make parts) and those sourced externally (buy parts).

4. Test and Refinement

Next, in order to identify potential failures of the designed product they perform Failure Mode and Effect Analysis (FMEA). FMEA is performed on both individual components of the "Sliding Door Trolley" and on the entire product to assess design vulnerabilities and identify potential failure modes during operation. The goal of FMEA is to list all possible failure modes, evaluate their effects, and assign a severity rank to each part and to the complete product. This process helps improve the design quality by reducing the risk of product failures and ensuring a more reliable final product.

3.1.3 Overview of platform: Aras innovator

An in-depth study of Aras innovator platform was conducted to increase the understanding of Aras PLM standard framework and its customization features. The study included theoretical research on PLM system methodology, hands-on training of system features and factors affecting the customizations.
Aras, an American-based software company, has established its product lifecycle management (PLM) solution with the ability to support multi-disciplinary requirements of the product lifecycle on a single platform. Aras Innovator, as an open-source PLM platform offers a high degree of flexibility and customization. It is renowned for its open architecture and ability to adapt to various industry business needs such as Aerospace and Defense, Automotive and Transportation, Industrial Equipment, High-Tech and Electronics, Life Sciences and Medical Devices. Aras Scales to support both small teams and global enterprises. It offers a model-based framework with no programming required for many configurations. The platform’s cohesive data and process management framework allows for accessibility and adaptability in different scenarios. Its Collaboration Tools Facilitate cross-departmental and cross-disciplinary collaboration. It automates workflows, improve efficiency and reduce errors. It integrates seamlessly with CAD, ERP, and other enterprise systems to fulfill the implementation of business processes.
The flexibility of Aras infrastructure allows this project to tailor the PLM environment to align with the specific needs of training processes and learning objectives of their courses. Additionally, choosing an open-source solution establishes a strong foundation for future academic and research developments, enabling further innovation and adaptability. The basic architecture, components, and system governance mechanisms are discussed in the following.

3.1.3.1 Aras innovator: Basic Architecture

Aras Innovator is an Internet application framework based on the Microsoft .NET platform that can be deployed On-premises, cloud-hosted, or hybrid setups. Also, it supports industry-standard protocols like RESTful APIs for advanced integrations with external applications and information systems. As shown in Figure 3.2 the basic components include:

- **Aras Innovator Client:** Aras uses a web-based browser interface which requires very few client resources. Aras Innovator client runs in Microsoft Edge, Mozilla Firefox, or Google Chrome browsers.
- **Aras Innovator Server:** Running as a Microsoft .NET application on Microsoft IIS 8 or higher requires the Microsoft Windows Server platform.
- **Database:** All configuration rules and code, as well as solution business objects are stored in the Microsoft SQL Server database.
- **Vault Server:** A separate server application maintains information about files that are linked to objects in the SQL Server database.
- **Vault(s):** A file directory location made known to the vault server to store physical files.

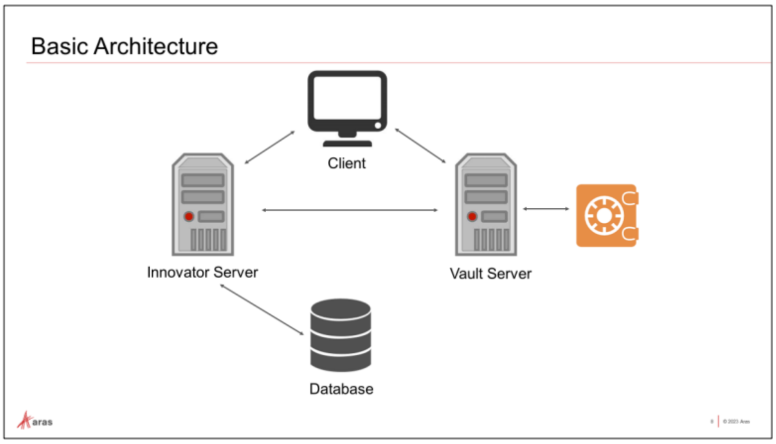


Figure 3.2 – Basic Architecture [18]

3.1.3.2 Aras innovator: Basic Solution Components

Aras Innovator is built on foundational components that each are playing a distinct role to enable the modeling and management of business processes within the system.

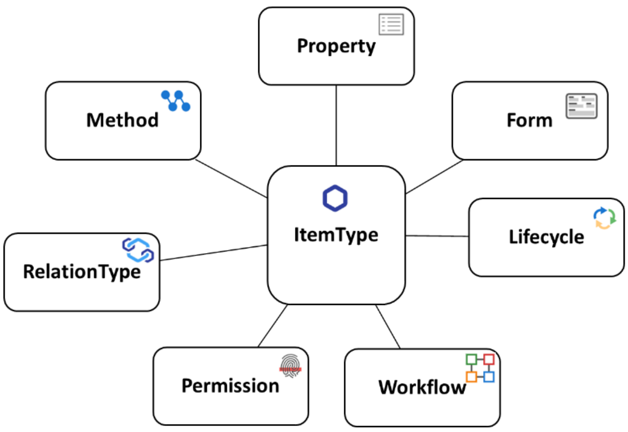


Figure 3.3 – Basic Components [18]

As shown in the figure 3.3 these components include:

- **ItemType:** An Itemtype is an analogous to a class in object-oriented systems, a template that defines the data structure and behavior of the item. An item is an instance of an Itemtype. The Itemtype determines how data related to that object are stored and governs the behavior of instances created from this type. Each Itemtype corresponds to a specific table in the database where its data is stored. Itemtypes serve as the foundational building blocks of the framework.
- **Property:** A Property represents a piece of information tracked for an Itemtype. Each Property has a name, a label, a data type, and several other settings to define its behavior. Each new Item that is created from an Itemtype in the system holds information relative to that instance of the Item. Itemtype Properties support a variety of Data Types, which are handled automatically by the Client and Server for presentation and storage purposes. (For example: created on holds the date and time an Item was created.) Properties are also referred to as attributes or meta data. Aras Innovator supports properties that are set by the server as well as custom properties created by administrators.
- **RelationshipType:** RelationshipTypes establish connections between Itemtypes, defining behavior and constraints of these relationships. Each RelationshipType includes a Source ItemType, an optional Related ItemType, and a Relationship ItemType, which acts as the link between the two.
- **Form:** A Form is a User Interface for an Itemtype to allow viewing and editing of an Item instance of that type. Forms can be configured as needed through a graphical user interface. The same Item can be displayed to different users in different ways by creating multiple Forms. One Form could be used for editing, while another Form could be used for printing.
- **Method:** A method is an Aras Innovator Itemtype containing executable .NET language code, C#, VB, JavaScript. Actions and Events trigger the execution of Methods. Methods define custom behaviors and logic for Itemtypes, enabling automation and enforcement of business rules. Methods determine the behavior of an Item; in this way, the software capabilities can be extended with low code methods
- **Permission:** A Permission defines what access a user or group has to an Itemtype. A Permission identifies one or more identities and the rights and privileges assigned to each identity, such as the ability to view, edit, etc. For example, it allows the user to search for items and view limited information about them in the search grid, but not view the detailed information that would be present on the Item form. Permissions can also be assigned to LifeCycle states to change the security of an Item over its lifetime. Permission can also be private to allow a user to change the access rights for the current Item. The configured Permission is unique to this single Item.

  Types of access:
    - Get Access: Enables identities to search and open Items.
    - Update Access: Enables identities to edit existing Items.
    - Delete Access: Enables identities to delete Items.
    - Can Discover: Allows the user to search for items and view limited information about them in the search grid, but not view the detailed information that would be present on the Item form.

- **Workflow:** The Workflow models a business process, i.e., a sequence of tasks. It determines who receives tasks and assignments and how voting decisions affect which path to take in a process, see figure 3.4.
- **Lifecycle:** A Lifecycle map is a series of states (i.e., stages, gates, or milestones) that an Item traverses during its existence, see figure 3.5. Most business processes define high-level stages to track the progress of an object in a Lifecycle map. An item can only have a single state at any given time. Every Item can be promoted through a defined Lifecycle map. As an object is promoted through various states, its status and other characteristics may change. Items can also be versioned so that each time an editing change is made to an object, a new generation ("version") of the changes is made in the database. Every Lifecycle map has a Starting state and Transitions that define the connection between two states, the promotion path. If desired, each state in a Lifecycle map can have a different permission set. Different permission sets guarantee that appropriate access to all items can be set as items progress through a lifecycle process or are promoted to a state that should have restricted access.

Figure 3.4 – Workflow example [18]

Figure 3.5 – Lifecycle example [18]

Lifecycle and workflows:

Lifecycle and workflow are two distinct but complementary concepts that help manage processes, tasks, and data throughout the product development phases. A Workflow can be configured to provide automatic promotion of an entities' Lifecycle, see figure 3.6. Using a combination of Workflow maps and Lifecycle maps provides the best of both technologies. The Workflow can create assignments, tasks, and use voting weight to decide when an object should be promoted to the next Lifecycle State.

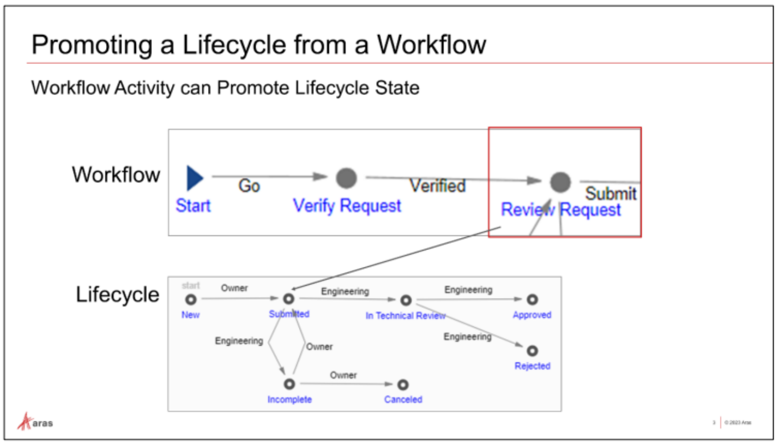


Figure 3.6 – Promoting a Lifecycle State from a Workflow [18]

ItemTypes and Items:

To configure the solution, Itemtypes will be defined as business objectives. Each Itemtype has Properties, Forms, Lifecycle Maps and Workflow Maps associated with the Item, Permissions, relationships, server and client Methods and Events that occur on the Item, and much more. Each Itemtype has distinct properties to carry pieces of information for instances of an Itemtype. An instance of an Itemtype is an Item (business object); each Item has distinct property values. For example, as shown in Figure 3.7, Aras Innovator contains a ItemType: Part that allows users to create unique Part instances in the database with unique property values.

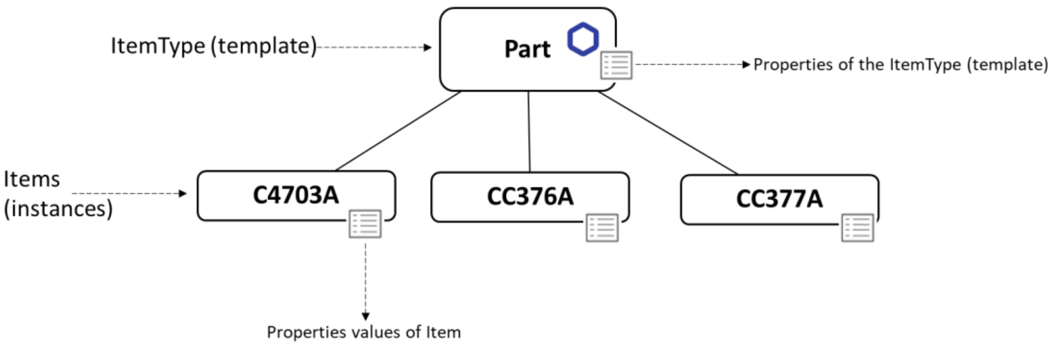


Figure 3.7 – Example of Itemtype and instance [18]

3.1.3.3 Aras innovator: System Governance

Aras Innovator includes a robust and highly configurable security model designed to provide precise control over system access and user permissions. This model allows administrators to define and enforce policies determining who can access the system and what actions they are authorized to perform once authenticated. By combining role-based access control, permissions at both the Itemtype and instance levels, and support for group memberships, Aras Innovator ensures that data and functionalities are protected while remaining accessible to authorized users.

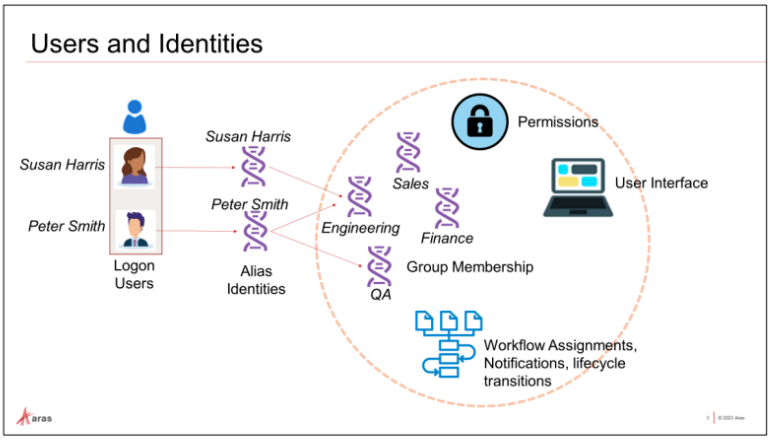


Figure 3.8 – Users and Identities [18]

As shown in figure 3.8, the security model basic components include:

- **Identity:** Identities are a core component of the security model and are used to manage user access and permissions. An identity item can represent a user alias, a group or a role within system operations.
- **User:** a user represents an individual person who interacts with the system. Users are linked to identities, which in turn determines their permissions, enabling controlled access to data, tools, and functionality based on organizational roles and responsibilities. A user must be defined in the system with unique credentials, including username and password. When a new user item is created, the Innovator Server automatically generates a corresponding system record known as an alias identity. Every user is assigned an alias identity, which represents them as an actor in system operations.
- **Group Identities:** Group identities define a list of user identities that share the same access. Alias identities can be part of several group identities to inherit accesses. Group identities are used to represent departments, roles, etc. All members of a group identity inherit the access privileges of the group identity they belong to.
- **Teams:** A Team, like group identities is defined as a group of one or more identities that are assigned as members but with difference that each identity that is included in a Team can be assigned a Team Role that indicates the member's responsibility for an Item. A Team is a defined logical group of users in Aras Innovator. A Team Role can then be assigned in a Permission to allow for dynamic security based on Team membership. Teams and Team Roles can also be referenced in LifeCycle Maps, Workflow Maps, and in most other situations where identities can be used. Team Roles can be assigned to a Permission to enhance access to Items. The Property team_id is an Item Property and it can be placed on a form like any other property. Once it is placed on a form, it can be used to select a Team as a responsible team for an Item. Teams are a vital component for organizing and managing groups of users who collaborate on specific tasks, projects, or workflows. Teams facilitate efficient collaboration, streamline assignment of responsibilities, and ensure proper access control within the system.

System Identities:

There are several system identities in Aras Innovator that are reserved for specific purposes, and cannot be deleted or changed (table 3.1). The role identities like Owner and Manager can grant specific privileges in a security policy; when their related property is placed on a Form, an individual or group can be chosen when the item is created or edited.

Title	Description
World	Represents all identities in the system
Administrators	Allows an identity to have administrative access to Aras.
Creator	This identity represents the User that created (first saved) an item.
Owner	This identity represents the value of the owned_by_id System Property on an item.
Manager	This identity represents the value of the managed_by_id System Property on an item.
Super User	The Super User is a special user identified as the "root" user of the system

Table 3.1 List of identities

As previously discussed, Aras Innovator is built on a comprehensive and interconnected framework of components that enable robust modeling, management, and governance of business processes. Through its flexible architecture, the system supports the creation, organization, and automation of data and workflows tailored to diverse organizational needs. Core components like Itemtypes, properties, forms, relationships, workflows, and lifecycles provide the tools for defining and managing business objects and their behaviors. Meanwhile, the security model, including permissions, identities, users, groups, and teams, ensures granular control over access and collaboration, promoting efficiency and compliance. The integration of workflows and lifecycles further enhances process governance by automating transitions and task assignments, optimizing operational efficiency.

3.1.3.4 Aras innovator: Applications

Aras offers applications standard designed to drive digital transformation across the organization. These applications have a foundation for managing complex product data, fostering collaboration, and streamlining workflows throughout the entire product lifecycle. Much of the functionality available within Aras Innovator is broken up between many applications. To better establish what is packaged with each application, some of those applications are introduced in the following:

- **Program Management (PM):** PM manages and view projects and processes with tools such as stage-gate/phase gate organizations, project trees, Gantt charts, tasks/activities, status roll-ups, and work breakdown structures. It allows Project planning by clear Structure in managing and tracking tasks, deadlines, and milestones.
- **Product Engineering (PE):** PE is the foundation for all of the applications across the platform, and ensure that every process implemented within Aras PLM has visibility and connections with parts, Bill of Materials, and Engineering Change

information. It enables a single platform for the end-to-end lifecycle processes involved in product design, development, manufacture, and maintenance. It allows for cataloguing parts and Bills of Materials, associating CAD documents with their identifying information like BOM structure, Parts numbers, manufacturers / suppliers, documentation, versioning, permissions, and more.

- **Requirement Engineering (RE):**RE facilitates the capture, tracing, and validation of requirements throughout the development process. It enables the expression of customer needs, ensures a clear understanding of product functionality, identifies key building blocks of products and systems, and verifies alignment between design and intent. Additionally, it supports the definition and validation of design and quality goals.
- **Change Management:** This application is a robust tool designed to streamline the process of managing changes of parts, assemblies, documents and CAD documents across an organization. It supports the entire change lifecycle, from identifying and evaluating change requests to implementing and verifying the changes. The application provides configurable workflows, enabling users to define approval processes that align with their business requirements. It integrates seamlessly with other Aras applications, ensuring that changes are effectively tracked and linked to related product data, such as Bills of Materials (BOMs), CAD files, and documents. With features like impact analysis, traceability, and audit trails, Aras Change Management ensures controlled and efficient handling of engineering changes while maintaining compliance with industry standards.
- **Simulation Management (SM):** SM application is designed to manage simulation data, processes, and results within the PLM environment. It enables traceability by linking simulations to product designs, requirements, and test cases. SM helps standardize simulation workflows, ensuring consistency and collaboration across teams. By integrating simulation with the product lifecycle, it enhances decision-making and supports validation of designs before physical prototyping.
- **Variant Management (VM):** VM allows to define and manage complex configurations of product variants across engineering disciplines and organizational boundaries. This enables to efficiently define and manage variability in product platforms, maximize module reuse, reduce cost of quality (COQ), lower costs, and shorten time to market. Establishing a consistent approach to authoring and managing variability is an essential element of product strategy.
- **Component Engineering (CM):** CE ensures users have full management and tracking of Manufacturer Parts, including the ease of selection and swapping of components. CE within Aras Innovator simplifies electronic component selection by allowing Engineers and Procurement Specialists instant access to current technical data statuses on hundreds of millions of parts. This includes components down to the board-level from leading manufacturers around the world. Using Information Handling Services (IHS) marketplace, CE allows the responsible engineer/specialist to ensure purchased components meet specifications, identify comparable alternate parts, and ensure obsoleted parts are replaced in timely manner without disruption.
- **Technical Documentation:** This application can structure the creation of many different kinds of product-related documents by connecting with data from all of the applications across the Aras Innovator platform. It automates data creation, pulling the latest product information into the customizable layout teams need to work in—whatever type of document that is. And, it automates changes: as each linked piece of content is updated, stakeholders are notified and can accept the change or continue

evolving the document. This quick and easy notification and synchronization accelerates their work and reduces the risk of inaccuracies.

- **Manufacturing Process Planning (MPP):** MPP application facilitates the creation, management, and utilization of the Manufacturing Bill of Materials (MBOM) and Process Plans, ensuring alignment between shop floors and engineering changes. It enables engineers to seamlessly transform the Engineering Bill of Materials (EBOM) into an MBOM while simultaneously defining the Bill of Process (BOP). The application supports the development of structured, work instructions using an intuitive document editor to design each step of the process plan.
- **Quality Management system (QMS):** QMS application introduces a couple of important documents that make it possible to create and maintain Failure Mode Effects Analyses (FMEAs) for both the design of products (DFMEA) and for the process of assembling products (PFMEA). It uses two primary Itemtypes responsible for planning and tracking the preventive actions that may be taken to address failures in products. These are the Design Quality Document (DQD) and Process Quality Document (PQD). The DQD is intended to track information on how the actual design of the product could result in failure. Perhaps in a stress test a particularly thin piece of product breaks, this would be something you'd want to track in DQDs. The PQD is intended to track information on the manufacturing process of products. In fact, PQDs can be linked directly to a Process Plan item if you're also using the Manufacturing Process Plan application. The PQD also comes with a few different views to help visualize processes. The first one you'll see is the Process Control Plan, and it is intended to just provide a general overview of each step in assembly. QMS ships with a Corrective Action Plan (CAP) Itemtype for tracking just this kind of information. The CAP comes with a few standard relationships for tracking information like what event caused the failure, how the failure can be contained, analysis of the failure, how the failure will be corrected, and how the failure can be prevented in the future.
- **Supplier Management (SM):** SM empowers organizations to collaborate more effectively with suppliers and OEMs via a unified secured environment, allowing all users access to a single set of processes and data that is up to date. By facilitating communication, data sharing, and connectivity across the full lifecycle of product development, organizations can achieve better visibility, data analytics, and efficiency. They can also reduce inventory and overhead costs, while improving quality control.

Aras applications work seamlessly together to support every stage of the product development lifecycle, enabling organizations to efficiently manage complex product data, processes, and workflows. The figure 3.9 presents how these applications integrate to create a cohesive product development environment:

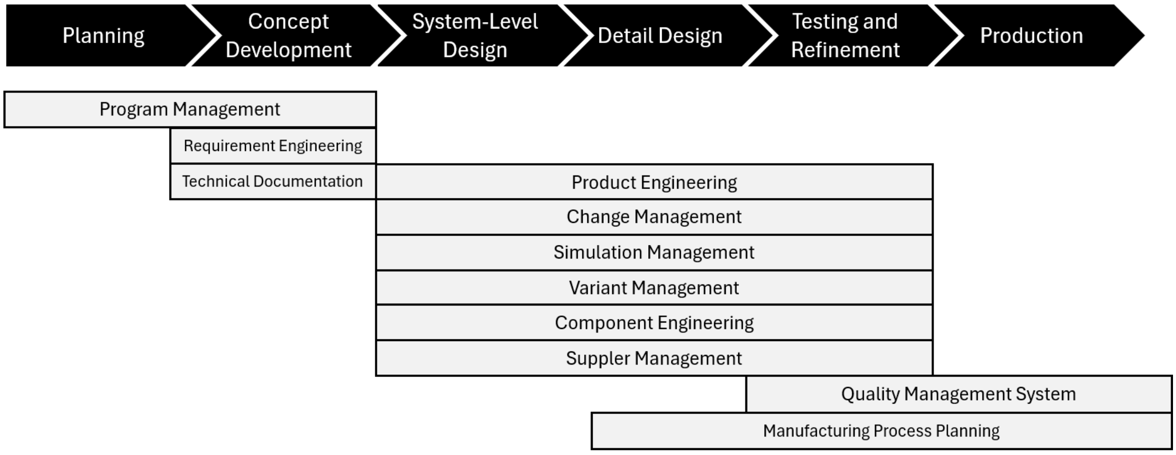


Figure 3.9 – Aras applications

3.1.4 Phase 0 Output: Mission Statement

The results of this phase outline the core purpose, the scope, key assumptions, constraints, environment and stakeholders of this project in the mission statement. The mission statement plays a crucial role in guiding the development of the optimal solution and stands for what the project aims to achieve. Table 3.2 presents the strategic direction for PLM framework development. This output is the input required to begin the concept development phase and which serves as a guide to the product development.

Mission Statement: Aras PLM solution for educational purposes	
Product Description	A PLM platform to provide users with a standard and real-world experience in collaborative product development using a PLM system.
Benefit Proposition	- Improving student product development project quality - Train new engineers to work with real world market tools - Enhancing collaboration skills
Key Business Goals	-Support Eppinger's Product development model -Be easy to use by not experienced users -Support facilitation tools for collaboration -Providing evaluation points
Environment	Educational courses in product development
Assumptions and Constraints	-Full license will be provided through negotiation between Polytechnic of Turin and Aras company -Proper Hardware resource will be provided by Polytechnic of Turin -The platform must be public accessible
Stakeholders	Polito administration/ Professors/ Students

Table 3.2 Mission statement

3.2 Phase 1: Concept Development

In the initial step of this stage, requirements were gathered through interviews with members of the Faculty of Management and Production Engineering. This process enriched data collection, enhancing both the depth and robustness of the analysis. The stakeholders' requirements were then identified and categorized into functional and non-functional requirements.
Next, insights and experiences from similar projects at Polito and other educational institutions were analyzed to guide the research and incorporate students' perspectives into the requirement-gathering process.
To clearly represent these requirements, a UML diagram was developed, providing a comprehensive view of user operations within the system. As part of the high-level requirements analysis, a use case diagram was created to illustrate key system functionalities and user interactions. For each operation, potential solutions were explored, and the most suitable ones were selected to serve as inputs for the system-level design phase.
Although this phase followed a structured approach, it was inherently iterative, involving continuous refinement and improvement as new insights emerged. The final results reflect these enhancements, ensuring a clear and comprehensive understanding of the required system features. This solid foundation facilitated the development of a conceptual solution aligned with stakeholders' needs and project goals.

3.2.1 Requirements Analysis

"A requirement is a condition/capability to be met/fulfilled by a system satisfying a contract, specification, standard, or formally imposed documents"[19]

Requirements are the foundation of software systems. Functional requirements indicate services, tasks or functions the system has to perform, while Non-Functional requirements define overall qualities/attributes of the system such as capacity, performance, and security [19]. Requirement analysis starts with requirements gathering. This involved conducting interviews with key stakeholders to gather their expectations and insights regarding the system. In order to structure the interviews some questions and prompts were prepared before the interviews:

- What are exact teaching steps in the course?
- What kind of information is shared with students?
- What are expectations of students at each step?
- What impact do you expect from the PLM system to have on teaching?

Handwritten notes method was used to document the interviews and results were analyzed to produce a list of requirements. To organize the raw data, a data template was implemented using a spreadsheet. The collected inputs were then categorized into functional and nonfunctional requirements. The Following table 3.3 presents the final requirements.

Type	Group	Requirement	User
Functional Requirements	Administration	User definition	Tutor
		Team structure definition	Tutor
		Team permission management	Tutor
	Project management	WBS creation	Student
		Task creation and assignment	Student
		Project Automatic scheduling	Student
		Gantt charts	Student
		Team communication (chat, comments, and notifications)	Student
		File sharing and document management	Student
		Shared workspaces	Student
	Meeting management	Meeting management tools	Student
	Requirement Analysis	Requirement Definition	Student
		House of Quality Tool	Student
	Product design	Product definition	Student
		Part definition	Student
		Cad File management	
		BOM definition	Student
	Change management	Part change order support	Student
	Process design	Operation definition	Student
		MPP definition	Student
	Quality management	FMEA Tool	Student
Non-Functional Requirements	Simplicity of use		-
	Each team's project and data should be isolated		-
	Optimized Performance		

Table 3.3 List of requirements

The assessment criteria of students' projects will be done the assessment of the expected outcomes to evaluate their performance, collaboration, and overall results of the project. At the end of the students' projects the following objectives will be assessed by the Tutor per each team:

- Having a completed project plan
- Having market analysis reports in the document
- Defined product, part, BOM, and cad documents
- Performed one specified change ordered by tutor using change management tool
- Having BOM and cost report
- Having a process plan

3.2.2 Similar Projects Insights

To research and incorporate requirements from the students' perspective, an analysis of student feedback from two similar projects is presented below:

2010, EAFIT University [20] **:** This project reports an investigative case study of PLM for educative purposes in EAFIT University. The university aimed to graduate autonomous, disciplined engineers with the ability to work effectively in groups, and PLM is seen as a tool to achieve. The outcomes of the pilot implementation showed an improvement in project organization, planning, and information management. The standard version of Aras Innovator was used during this investigation. This version comes with a lot of functionalities already included, that are in a way standard, like some user identities, permissions, item types, etc. Some of the add-ons that were downloaded: Meeting Manager, Multi-Level BOM Tools, Project Template Management. The main Items used by students are: “Document”, “Part” (CAD files), “Meeting” (a meeting log), “Project”, “Project Template”, and “Activity 2” (an activity assigned in a project). Most relationships were left as default, except for one in the Itemtype “Meeting”. This is an Itemtype that was added to Innovator through one of the community projects (add-ons) to generate reports of each of the counseling sessions the groups have with professors throughout the semester. r. In those meeting reports there should be expressed the highlights of each session, participants, commitments and person responsible for them for the following week. In order to facilitate software use, three of the default categories were left in the TOC: “My Innovator”, “Design” and “Documents”. They set automatically rules for naming created items. The case study project was performed in 4 phases. They used pre made project templates and roles for each of the 4 phases, to be used by students. The templates were standard and designed by professor but still they could be customized by students, except for some activities that were marked as “required”. The case study project calendar started from the first week of classes and by the fourth week they must have handed in the first two phases of the project. In this schedule, quickness to get students ready to work with ARAS Innovator, plan their project and upload documents was imperative. In order to achieve this, two introductory sessions were planned, both on the second week of classes. A third session, covering the topic of CAD Management in ARAS Innovator, was planned for the fifth week (when students start to work on CAD models). Table 3.4 presents student feedback grouped based on their similarities in the case study project.

Feedback	
Positive perception of the software	Easy to use
Technical Issues	Low connectivity
	Instability (frequent blocking)
	Low speed
	Poor compatibility with some browsers and software
Usability Concerns	Some students found the process of numbering and uploading files to be time-consuming and tedious
	Difficulty in solving problems with the software independently
Methodology Challenges	Some students acknowledged the need to change their work habits to align with the PLM methodology
	They often forgot to check the activities assigned to them in the system
Benefits of PLM	Students identified improved organization
	Easier information access
	Better project planning and control
	Greater punctuality in finishing tasks
	They appreciated having access to all the project information and documents

Table 3.4 student feedback

In addition to feedback, students’ recommendations on the PLM implementation are as the following:

- Solve the technical problems mentioned above

- Develop a printed manual of the software
- Create more virtual tutorials with graphics and troubleshooting tips
- Evaluate and grade user participation
- Introduce PLM methodology in prior to the project
- Offer more tutorial sessions
- Include email notifications

2014, a short PLM course [21]**:**This paper presents a short PLM course designed to be deployed all over the world in universities that do not have a PDM software. The course has been deployed several times in different universities (Ss. Cyril and Methodius University in Skopje, FYROM; University of Novi Sad, Serbia; Arts et Mètiers ParisTech, France) and technical institutes (ITIS OMAR Novara, Italy). The most complete experience has been reached in June 2013 at University Federico II of Naples. Results and conclusions refer to this last experience.18 professional master level students from different backgrounds participated in the 27 hours course Students were assigned to different teams according to their preferences: 2 Project Managers, 4 Quality Engineers, 4 Product Designers, 4 Process Designers and 2 Change Specialists. The course was given in 6 days in sessions of 3 and 6 hours. The project used a case study and Aras Innovator to teach PLM principles across various process areas, including requirements management, design, and change management. All modules used in the development of the exercise came along with the standard version of Aras Innovator 9.3. The only exception was the Requirements Management module that was developed by a third party and delivered as an add on to the software. The only customizations made to the software were Requirements classification, Requirements sequence number, Part sequence number, Document sequence number. Student teams collaboratively developed a product, gaining practical experience in information exchange and project management within a PLM environment.
The short PLM course experienced some technological and usability problems, as well as negative student perceptions. At the end of the course an anonymous questionnaire gathered student feedbacks. Overall, the course achieved its objectives satisfactorily and the contents of the course were considered original by the majority of students. The teaching method and material given were highly appreciated.
Table 3.5 presents student feedback.

Feedback	
Technical Issues	Aras worked exclusively on Internet Explorer (IE) using a .NET security framework which limited the use of computers with Windows operating systems. This was a problem because the software was not compatible with other browsers, such as Firefox
Usability Concerns	Frequent software bug occurrences

Table 3.5 Student feedback

3.2.3 Requirement Analysis Results

Therefore, following the requirement analysis from the interviews and the evaluation of similar past projects, the final set of requirements is presented below. The table 3.6 highlights the newly added requirements, providing a clear overview of the refinements made.

Type	Group	Requirement	User
Functional Requirements	Administration	User definition	Tutor
		Team structure definition	Tutor
		Team permission management	Tutor
	Project management	WBS creation	Student
		Task creation and assignment	Student
		Project Automatic scheduling	Student
		Gantt charts	Student
		Team communication (chat, comments, and notifications)	Student
		File sharing and document management	Student
		Shared workspaces	Student
	Meeting management	Meeting management tools	Student
	Requirement management	Requirement Definition	Student
		House of Quality Tool	Student
	Product design	Product definition	Student
		Part definition	Student
		Cad File management	
		BOM definition	Student
	Change management	Part change order support	Student
	Process design	Operation definition	Student
		MPP definition	Student
	Quality management	FMEA Tool	Student
	Notifications	Send notifications for task assignements and due dates	Student
Non-Functional Requirements	Simplicity of use		-
	Each team's project and data should be isolated		-
	Optimized Performance		-
	System technical reliability		-
	Process performance reliability		-

Table 3.6 Student feedback

3.2.2 System Modelling: Use Case Diagram

UML (Unified Modeling Language) is a standard language for modelling to visualize, specify, construct and document software system. Although UML is generally used for software development purposes, it is not limited within this boundary. It also can be used to model framework solutions [22]. In UML there are five diagrams available to model the system and the use case diagram is one of them.

The process of system modeling starts with capturing the dynamic behavior of the system in the environment when it is operating. The use case diagram is one way to model the behavior

of the system based on the high-level requirement analysis including internal and external influences. The use case diagrams are consisting of actors, use cases and their relationships. In this thesis, the functionalities of a desired PLM system are presented by utilizing these entities in use case diagram to provide the outside view of user's interactions with the system. In addition, each use case is explained trough a short non-detailed description.
The purposes of this section are to:

- Gather high level requirements of the system in an organized manner.
- Get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interactions among the requirements and actors.

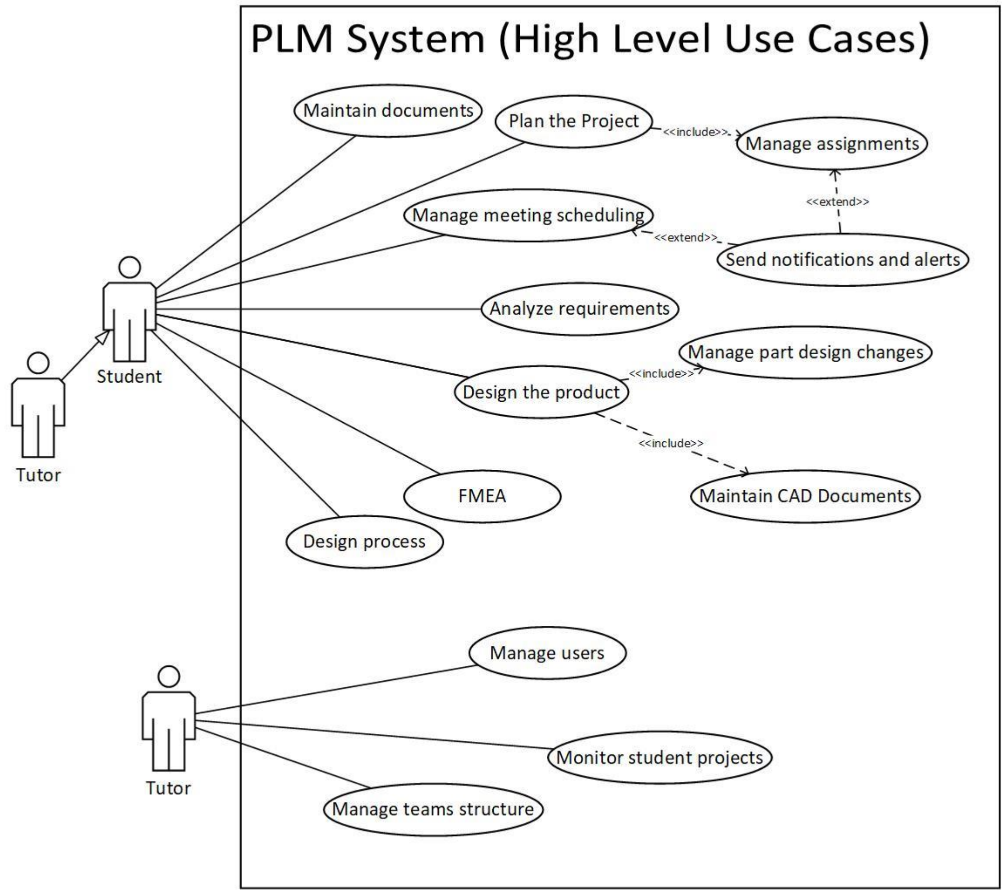


Figure 3.10 – Use Case Diagram

Use Case Briefs:

- **Plan the project**
  Actor: Student.
  Description: The student creates a project plan by defining WBS, setting milestones, and scheduling. The use case includes managing task assignments and dependencies.
  Includes: Manage Assignments, allowing the student to assign tasks to team members and track their progress effectively.
- **Manage Meeting Scheduling**
  Actor: Student.
  Description: The student schedules project-related meetings, sets agendas, and records meeting notes and decisions.
- **Analyze Requirements**
  Actor: Student.
  Description: The student defines and analyzes product requirements, utilizing tools such as the House of Quality for prioritizing and mapping requirements.
- **Design the Product**
  Actor: Student.
  Description: The student develops the product structure, defines parts, and designs the Bill of Materials (BOM).
  Includes: Manage part design changes and Maintain CAD documents
- **Design Process**
  Actor: Student.
  Description: The student defines manufacturing processes, including operation sequencing and resource allocation.
- **FMEA**
  Actor: Student.
  Description: The student uses Failure Mode and Effects Analysis (FMEA) tools to identify potential failure modes and assess their impact on the project.
- **Maintain Documents**
  Actor: Student.
  Description: The student manages project-related documents, including uploading, organizing, and sharing files with teammates.
- **Send notifications and alerts**
  Actor: Student.
  Description: The student receives notifications by task assignments and due date alerts.
- **Manage Users**
  Actor: Tutor.
  Description: The tutor creates, updates, and disable user accounts, setting permissions for access to project resources.
- **Manage Teams Structure**
  Actor: Tutor.
  Description: The tutor defines team structures, assigning users to teams and configuring their access to the system
- **Monitor Student Projects**
  Actor: Tutor.

Description: The tutor monitors the progress of student projects, reviewing task completion, deadlines, and project status.

3.2.3 Concept Generation & Selection

Once user requirements are clearly defined, the next critical step is mapping these requirements to system-level functional features. This mapping process ensures that the final solution fully aligns with the users' needs and expectations, providing a user-centered, functional solution that supports project goals.

This step involves a comprehensive exploration of the described functionalities of the Aras PLM core application and out of box applications (see section 3.1.3) to identify how the platform can meet each specified requirement. For each system-level user requirement, one or more potential concepts are proposed using the core system features or out of box applications. Following the identification of possible solutions, a detailed analysis is performed to evaluate their alignment with project objectives, functional and non-functional requirements. Factors considered in the analysis include:

- User experience and usability:
  - How effectively the solution supports users in performing their tasks.
  - Simplicity in performing operation.
- Cost and time efficiency: The effort required to implement and maintain the solution.
- Integration: How well the solutions of different functionalities integrate with each other.
- Scalability and future growth: The potential for future enhancement without extensive rework.

The following table 3.7 presents the results of the analysis at this stage, listing possible solutions for each requirement and highlighting the most suitable one. The finalized solution descriptions serve as a blueprint for the subsequent system-level design phase.

Requirements	**Actor**	**Possible Solutions**		**Selected Solution**
		Application	Desc.	
Plan the project	Student	Program Management	Project: Create a project, define WBS, assign tasks, schedule, and manage the project progress	☑
		Program Management	Program: Plan different phases in different projects Project: Create a project for each phase, define WBS, assign tasks, schedule, and manage the project progress	
Manage assignments	Student	Core	Inbasket: Send assignments to users, task management and update tasks progress	☑
Maintain Documents	Student	Core	Documents: Upload, maintain, organize, and share files with teammates	☑
Manage meeting Scheduling	Student	External extension		
Analyze requirements	Student	Requirement Engineering	Requirements: Define requirements	
		Requirement Engineering	Requirements: Define requirements	☑

			Req. Documents: Map and categorize requirements	
Design the product	Student	Product Engineering	Product: Define the product and generate multiple distinct versions Part: Define parts, design BOM, manage product Life cycles	☑
		Product Engineering	Part: Define parts, design BOM, manage product Life cycles	
Manage part design changes	Student	Change Management	Express DCO: Manage changes in documents, cad documents and their effects on other items Express ECO: Manage changes in parts and their effects on other parts, cad documents and documents	☑
		Change Management	Express DCO: Manage changes in documents, cad documents and their effects on other items Express ECO: Manage changes in parts and their effects on other parts, cad documents and documents. Rework Orders: To submit rework orders and manage rework reasons	
Maintain CAD Documents		Product Engineering	CAD Documents: Upload, maintain, organize, and share CAD files with teammates	☑
FMEA	Student	Quality Management	Design Quality Document: Perform FMEA and export the results	☑
Design Process	Student	Manufacturing Process Planning	Process: Develop the product development process and define operations in sequence Machines: Define machines in product development process Tools: Define tools in product development process Skills: Define skills in product development process	☑
Send notifications and alerts	Student	Core		
Manage users	Tutor	Core	Users: Define users and manage users access to the system	☑
Manage teams structure	Tutor	Core	Teams: Structure teams and assign students into teams	☑
Monitor student projects	Tutor	Program Management	Program: Define a program with all teams' projects, monitor projects of a course.	☑

Table 3.7 Solution analysis

The requirements "Manage meeting Scheduling" and "Send notifications" and alerts were excluded from the current implementation due to considerations of cost and time efficiency. Developing these functionalities was assessed as time-intensive and resource-demanding, given the current limitations in configuration capabilities and available resources. However, these features have been designed with scalability in mind, allowing for seamless integration and configuration in future phases of system enhancement.

In addition to selecting the most appropriate solution for each requirement, it is essential to address key non-functional requirements during the system-level design. These

requirements ensure that the overall system not only meets functional expectations but also delivers a robust and user-friendly experience.
The following non-functional requirements must be considered:

- Simplicity of Use: The system should provide an intuitive and user-friendly interface, enabling users to efficiently perform their tasks with minimal training or complexity.
- Project and Data Isolation: Each team's project data must be securely isolated to maintain privacy and prevent unauthorized access or data conflicts across different teams.
- Process Performance Reliability: Processes within the system should consistently perform as expected, delivering accurate results without failures or errors.
- Optimized Performance: The system should be designed for speed and responsiveness, minimizing delays during user interactions and data processing.
- System Technical Reliability: The system architecture should ensure high availability and resilience to minimize downtime and operational disruptions.

The results presented in this section offer a clear roadmap for the system's development. It serves as a strategic guide, outlining the key features, functionalities, and design priorities derived from user requirements and technical constraints. This roadmap ensures that development efforts remain aligned with the project's objectives. Additionally, it provides a work plan for tracking progress, measuring performance, and validating that the final system meets the defined user expectations and goals.

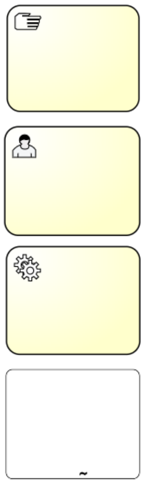
3.3 Phase 2 Systems Level Design

This section focuses on establishing the structure of the proposed solution. The purpose of the section is to define the core building blocks of the framework, detailing their functionality and how they interact to form a cohesive system. Structural decisions made at this stage provide the foundation for the detailed design of the solution in the next phase, ensuring a clear blue print of the implementation.

The structure includes multiple applications provided by Aras, each serving distinct purposes. These applications are composed of several Itemtypes, with each Itemtype supporting various operations that contribute to achieving the objectives of the application. While applications offer many benefits, they can also introduce challenges to the project. For example, if the underlying logic of an application diverges from the logic aligned with project requirements, it can impede achieving the desired outcomes. Additionally, applications may provide a broader range of features than necessary, potentially distracting focus from the primary purpose and leading to inefficiencies. The design presented aimed to maximize potential benefits while mitigating drawbacks by including and excluding some functionalities within each application, in order to optimize the interactions between applications and to ensure an efficient and well-integrated PLM solution that balances performance, usability, and scalability.

Based on the procedure identified through the requirement analysis, the sequence and usage of each activity is demonstrated using SAP Signavio Business Process Model and Notation (BPMN) (see figure 3.12) to illustrate how different activities are performed either by using the system or out of scope of the system. Therefore, the illustrated process aligns with the

course project workflow followed by students across different phases of product development.
This diagram represents a System-Level Design process using a swimlane workflow. It is consitituted by two pools, representing the external environment and PLM system respectively. Each lane in the PLM pool represents applications within the PLM system, while the Itemtypes used for each activity are mentioned using notation box. Different types of tasks used in the diagram are as the following figure 3.11:



A task that is performed manually by a person without any system involvement.

A task assigned to a human user but executed within a software system. The user interacts with an interface

A task that is executed automatically by a system without human interaction

An Ad Hoc Sub-Process is a flexible type of sub-process where tasks inside it do not follow a predefined sequence flow

Figure 3.11 – Different types of activities in BPMN

Multiple implicit ends in the diagram indicates activities and sub-processes which complete without terminating the process. This approach simplified modeling by avoiding unnecessary complicating the diagram.The process ends once all activities in the main flow from the external environment are done.

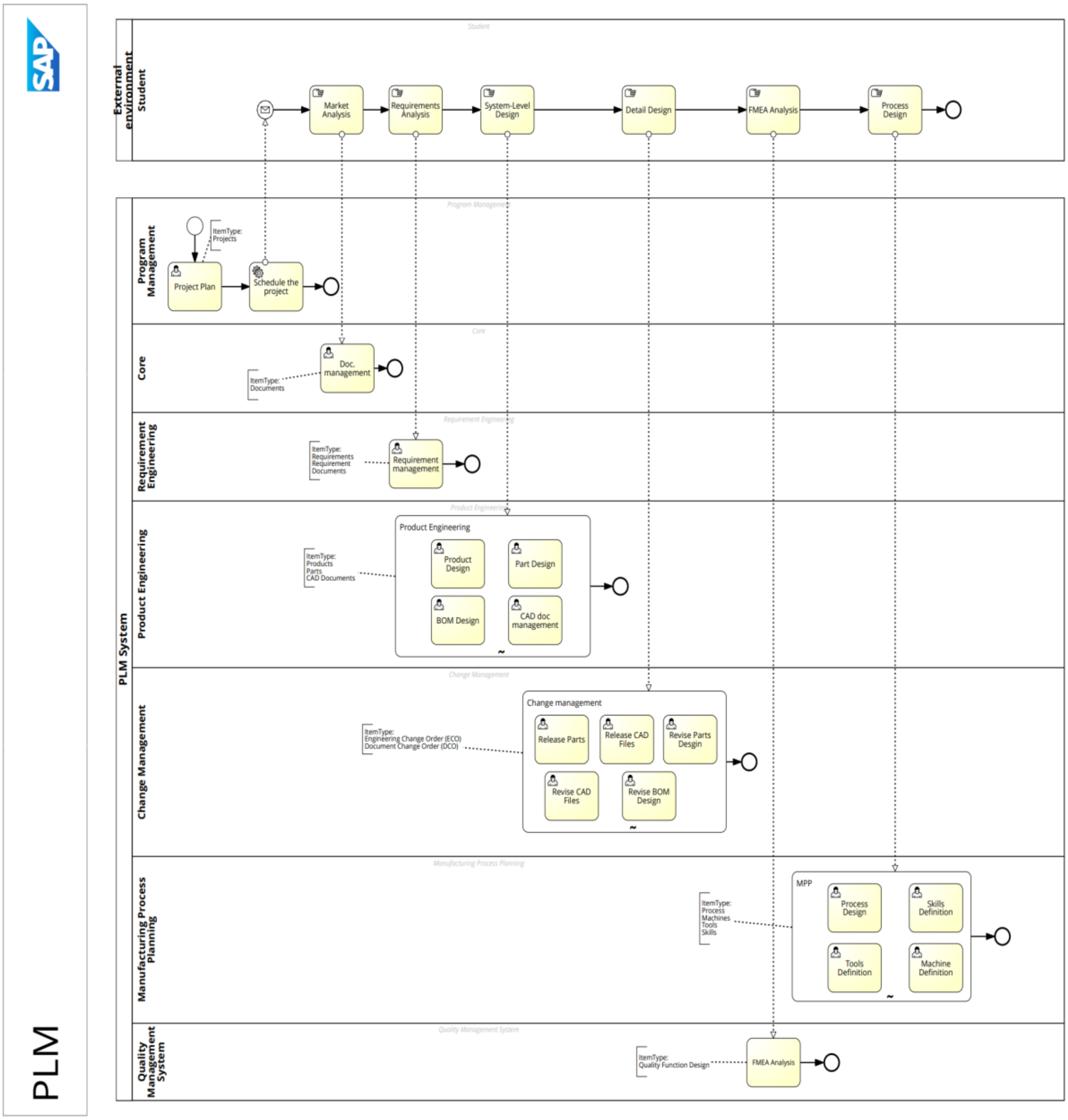


Figure 3.12 – PLM solution BPMN

The represented BPMN diagram illustrates the key stages from project initiation to FMEA analysis, and clarifies how the PLM system serves as the backbone of the process, ensuring structured collaboration and data generation and data sharing across disciplines such as project management, requirement engineering, product engineering, change management, manufacturing planning, and quality management system.

3.3.1 Project planning

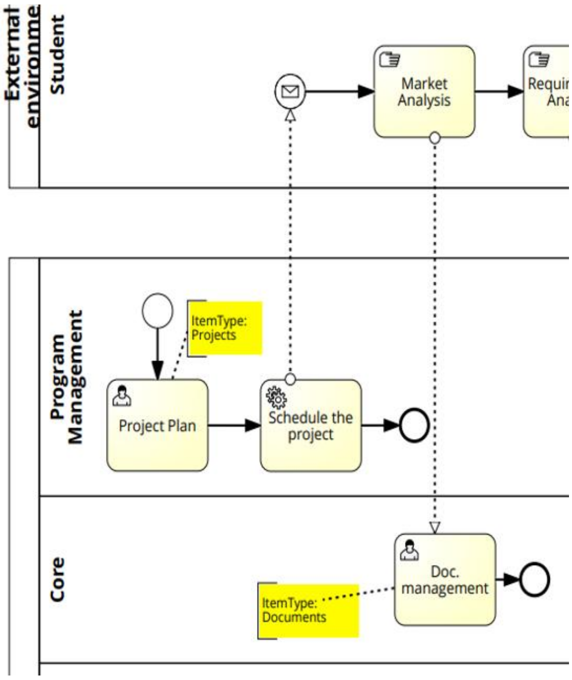


Figure 3.13 – Project planning

As the figure 3.13 presents, the process initiates with project planning in program management application. The Project Itemtype provides functionalities to break down a project into manageable phases or tasks, represented by the Work Breakdown Structure (WBS). This Itemtype allows students to define deliverables, create schedules, and assign tasks to responsible users across the team, ensuring effective project planning, tracking, and execution. Through the automatic scheduling service, the Project Itemtype can dynamically generate and adjust project plans based on predefined rules and dependencies. This automation ensures reduces manual effort and minimizing the risk of errors. By continuously synchronizing tasks and deadlines, the system enhances project efficiency, improves time management, and enables teams to focus on decision-making rather than routine scheduling tasks.

The Project Itemtype, when used in combination with other ItemTypes from the core application, excluding the standalone activities within the process can significantly enhance project management.

Post automatic scheduling service, Project Itemtype sends tasks to responsible users through the Inbasket Itemtype, to enable the users to receive the tasks and update their progress. Then after tasks updates by users, project progress will be updated automatically.
Inbasket Itemtype serves as a temporary workspace and personal storage area for all users. It allows users to manage their assignments efficiently, including both tasks from projects and workflows, and providing a centralized and accessible location for organizing and tracking work in progress.

In addition, Discussion Itemtype contains collaborative tools integrated with the design and development of Part and Document items. It allows to create secure text messages associated with an item that can be viewed and commented on by other users in the system. The students will organize comments by created Forums for their team and sharing with teammates. The comments can be managed through discussion panel and the associated items as well. Each comment can also be flagged by one or more users and tracked later to highlight important ideas from the discussion.

3.3.2 Market Analysis

Next activity is Market analysis which is performed out of the system; however users will use Document Itemtype to maintain and share generated files and data across the team. This Itemtype enables students to share and organize files efficiently, supporting comprehensive documentation and collaboration among team members. A single document can contain multiple files, offering a centralized structure for managing related resources. Furthermore, documents can be linked to other Itemtypes within the system, enhancing data connectivity and traceability. For instance, documents are ideal for recording and linking project deliverables, ensuring a clear and organized association between documentation and project outcomes.

3.3.3 Requirement analysis

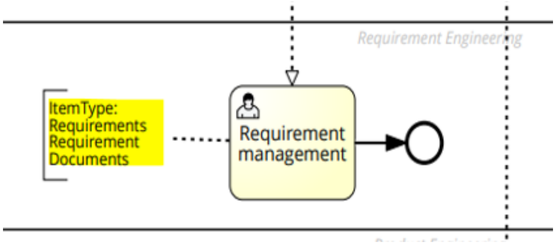


Figure 3.14 – Requirement analysis

The requirement analysis activity (see figure 3.14) employs Requirements and Requirement Documents Itemtypes to manage key building blocks of the product, allowing teams to define requirements, specifications, performance and validate design and quality goals. These entities are defined and organized into separate clusters for later connections to related parts, ensuring a cohesive and integrated approach to product development.

3.3.4 System Level Design

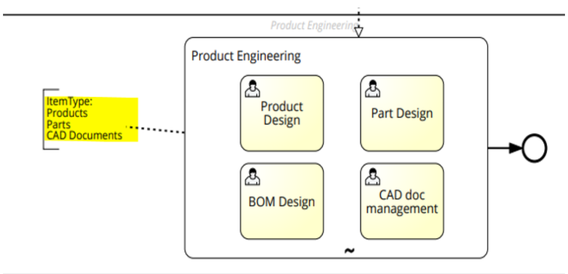


Figure 3.15 – System Level Design

System level design (see figure 3.15) is a crucial phase in product development, focusing on defining the overall architecture, interactions, and structure of a product before detailed design and manufacturing. Products, Parts, and CAD documents Itemtypes provide a structured framework to manage product data, configurations, and relationships, ensuring traceability, consistency, and efficiency throughout the product lifecycle.

The Part Itemtype represents individual components or subsystems within a product. It serves as the fundamental building block of system-level design and provides Bill of Materials (BOM) design tools by establishing hierarchical relationships between parts. Parts can be classified into categories such as Assembly, Component, Material, or Software and defined based on make-or-buy decisions. Furthermore, parts can be connected to requirements and associated requirement documents that they are designed to satisfy, ensuring alignment with specifications and standards.

Features like multi-level BOM reports and cost analysis reports can be useful for demonstrating project outcomes.

On the other hand, CAD Documents Itemtype provides defining and synchronizing the physical structure of part components.

The Product Itemtype represents an entity that will be released to market. A Product is related to one or more models, which are then related to a parent Part item that contains the top-level BOM for the Product Model.

3.3.5 Detail Design

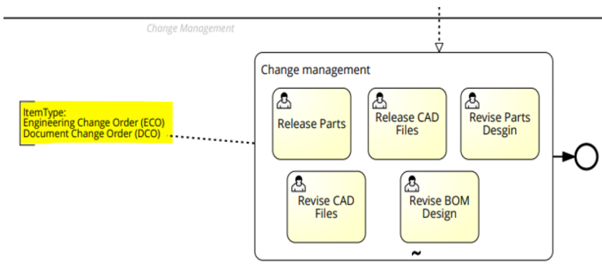


Figure 3.16 – Detail Design

As the design evolves (see figure 3.16), finalizing and revising of designs is made possible by the Change Management application, which allows to manage changes efficiently and prevent inconsistencies caused by those changes.

The ECO Itemtype implements a **change control process** for parts and CAD documents, enabling the release of parts after thorough reviews and approvals.

Throughout the change process, a part progresses through various Lifecycle states, depending on its position in the workflow. Users can use this function to add, modify, delete, release, and revise parts. Additionally, this Itemtype exploits the Impact Matrix, which allows users to quickly visualize and assess how changes to a part will affect other items within the hierarchy.

The DCO Itemtype provides the same functionalities as ECO, but for Documents instead. It ensures traceability by capturing the impact of any changes made to these entities, providing full visibility into modifications and their effects on the overall system.

This structured **change control process** prevents errors and ensures that only approved changes are implemented.

3.3.6 FMEA Analysis

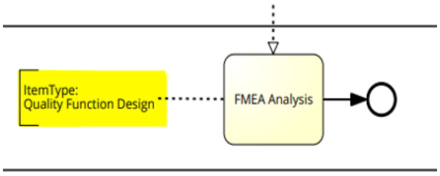


Figure 3.17 – FMEA Analysis

To ensure the product meets specifications and regulatory requirements, the Quality Function Design Itemtype provides features to perform FMEA Analysis (see figure 3.17). This Itemtype represents the Failure Mode and Effects Analysis used to identify and quantify risks associated with the design of the product and to identify any mitigation opportunities.

3.3.7 Process Design

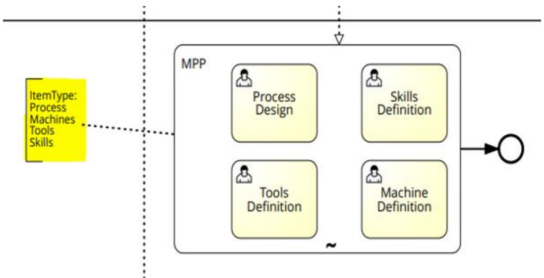


Figure 3.18 – Process Design

Once the design is finalized, the Process Itemtype supports design process plan for the final product by mapping operations, instruction steps, parts, machines, skills, and tools. Manufacturing Process Planning application provides Process, Machines, Tools and Skills Itemtypes to support this activity (see figure 3.18).

This interconnected structure fosters an efficient, well-organized, and highly collaborative product development process. By enabling seamless communication between different applications within Aras PLM, it ensures that teams work in a synchronized environment, minimizing errors, reducing redundancies, and optimizing resource utilization. This integrated approach not only accelerates project execution but also enhances overall productivity by streamlining workflows and improving data traceability. Furthermore, it strengthens collaboration across various stakeholders, ensuring that critical project information is always accessible, up-to-date, and aligned with organizational objectives. Ultimately, this system promotes transparency, efficiency, and innovation throughout the entire project lifecycle, laying a strong foundation for continuous improvement and long-term success.

3.4 Phase 3 Detail Design

This section will detail the configurations to reach the designed system in previous section. Configuring the system is a critical step in the implementation process that translates business process mappings into a functional software environment. This involves setting up system configuration, system administration, solution configuration and system interface design to align the system with the requirements. Below are the key subsections detailing the implementation process:

3.4.1 System Configuration

The installation of the Aras Innovator software was carried out following the step-by-step instructions provided in the official Aras Installation Guide [23] to ensure proper configuration and deployment. The process was completed on a dedicated server allocated specifically for this project, providing the necessary resources and environment to support the system's performance and scalability. A single server served as the Database, File Vault, and Web/Application servers at the same time. The server was deployed on a virtual machine hosted at Polytechnic of Turin. The software and hardware prerequisites for the installation of Aras Innovator were configured in accordance with the guidelines outlined in the official installation manual table 3.8 and table 3.9. After successfully installing the software, the academic subscription code provided by Aras was used to activate the software license. This activation enabled access to the features and capabilities of the Aras platform, to some extend not all features. This activation enabled access to the features and capabilities of the Aras platform, to some extend not all features. After installing the Aras core system then out of box applications were installed one by one.

Hardware Configuration	
Processor Speed	16 * 2295 Mhz cores
RAM	64 GB

Table 3.8 Hardware Configuration

Platform Configuration		
Component	Platform	Version
Operating System	Windows Server	2022 standard
Web Server	Internet Information Server (IIS)	8 • 10
.NET	.NET	
	ASP.NET Core and .NET Runtime and Hosting Bundle	
Relational Database	Microsoft SQL Server	2019
Microsoft Visual C++ Redistributable		2017
Report Server (Optional)	Microsoft SQL Server	2019

Table 3.9 Platform Configuration

Although the current scale of this project is relatively small, an important consideration is safeguarding data through regular backups to prevent data loss in the event of unexpected issues. The backup strategy for this project has been designed with the current scale and available hardware and software resources in mind. Naturally, as the system expands and is deployed on a larger scale, this backup plan will need to be revised and enhanced to accommodate increased data volumes and more robust protection requirements.
Aras [23] recommends three types of backups to ensure the security and integrity of the system:

1. **Backing Up Program Files:**
   It is important to back up program files before updating Aras to a new version. This backup is typically performed only during version updates rather than on a frequent basis.
2. **Backing Up Configuration Files:**
   Configuration files are essential for the proper functioning of Aras Innovator. These files generally remain unchanged unless configuration adjustments are made, such as adding a new database. Regular backups are recommended whenever configuration changes occur.
3. **Backing Up a SQL Server Database:**
   The database stores all user data and system metadata, making its backup a critical component of data protection. Regular and scheduled backups of the SQL Server database ensure data integrity and recovery in case of system failure or data loss.

For this project, the primary focus is on backing up the database. During the implementation phase, a backup is required only at key stages of implementations. However, once the system is in use for student testing, a comprehensive full backup is scheduled once per month, complemented by incremental backups performed weekly to ensure ongoing data protection and recovery capability with minimal storage overhead.

3.4.1 System Administration

The step encompasses a set of critical tasks to ensure the system is properly managed and secured. This step establishes the foundational governance logic and operational rules of the system for successful implementation and long-term management. Key activities include Setting up users, identities, group identities, teams and permissions.
The user configuration for this project was based on the standard settings provided by Aras, requiring no additional customization for authentication processes or password management. To organize users, this project deviates from traditional organizational structures with distinct departments and roles. Consequently, the predefined group identities provided in the default Aras setup do not meet the specific needs of this implementation. Instead, custom group identities were designed based on the requirements analysis outlined in the previous section. The list of group identities established to manage user roles is presented in table 3.10:

N.	Group Identity	Description
1	Students	Includes all student users
2	Tutors	Includes all tutor users
3	World	Includes all users in system
4	Course Admins	Include users authorized to have admin permissions

Table 3.10 List of identities

In addition to categorizing users into group identities, it is necessary to structure student users into separate teams. While all students must have access to the PLM applications, the data generated within each team must remain hidden from members of other teams but

accessible to the course tutor. Team definitions should be managed by the tutor; however, it is important to consider that the tutor may not have course administrator privileges.
To address this, the Itemtype - Team was used to define teams and assign student identities to each team. By adding a mandatory property, team_id, to each relevant ItemType, team-specific data can be securely restricted to the corresponding team members. This configuration ensures that only the team members and their tutor have access, while other teams remain isolated. Since the tutor does not grant specific roles to individual students within teams, all students are assigned with the Team Member role. In further steps, standard permissions are then applied to Itemtypes based on this role. This configuration balances flexibility, security, and simplicity, streamlining the team management process while adhering to access control policies. This approach offers several key benefits:

- **No need for administrative privileges for tutors**: Since modifying group identities requires admin permissions, this configuration allows tutors to define teams without needing course admin rights.
- **Simplified tutor user experience**: The team setup is straightforward, enhancing ease of use for tutors.
- **Efficient and reusable permission settings**: Permissions are configured once during setup, eliminating the need for modifications when new teams are created.
- **Controlled student interaction**: This setup limits student interaction to their teammates and tutor, ensuring data security and focused collaboration within each team.

Also, method “FillTeamFiledInTeamForm” was developed in Itemtype Team to restrict users to only see and choose their own team in the team search dialog.
Despite setting team permissions, users were still able to view other users in the search dialog, risking isolation of separate teams in using the platform. To address this issue, a custom server method named "FilterIdentitiesSearchDialog" was developed. This method was applied to all identity-sourced properties across relevant Itemtypes, effectively limiting the visibility of user data to only those identities were in the same team.
In summary list of methods developed are presented in table 3.11:

<table>
<tr><th>Title</th><th>Code</th></tr>
<tr><td>FillTeamFiledInTeamForm</td><td><pre>Innovator inn=this.getInnovator();
//fill the team_id property with the team's id
this.setProperty("team_id",this.getID());
return this;</pre></td></tr>
<tr><td>FilterIdentitiesSearchDialog</td><td><pre>var inn=new Innovator();

//check identities to understand if it's a student or not.
//retrieve student identity:
var stud=inn.newItem("Identity","get");
stud.setProperty("name","Students");
stud.setAttribute("select","id,name");
stud=stud.apply();
if (aras.getIdentityList().includes(stud.getID()))
    {
      var ids_ids=inn.newItem("Team Identity","get");
      ids_ids.setAttribute("select","id,related_id");
      ids_ids=ids_ids.apply();
      // Get an array of IDs that match our criteria
      var idArray = [];
      for (var i = 0; i < ids_ids.getItemCount(); i++)
      {
        idArray.push(ids_ids.getItemByIndex(i).getProperty("related_id"));</pre></td></tr>
</table>

	}
	 inArgs.QryItem.item.setAttribute("idlist", idArray.join(","));
	 }
	 return;

Table 3.11 List of methods

Table Of Content (TOC):

The TOC manages access to Itemtypes within the main tree view, with categories and labels configured using the TOC Editor. Designing an intuitive layout for the TOC improves navigation by providing more straightforward access to Itemtypes, thereby enhancing overall system usability. TOC layouts can be customized for different user groups. In this context, distinct TOC configurations are created for students to better align with their specific needs and responsibilities. See the figure 3.19:

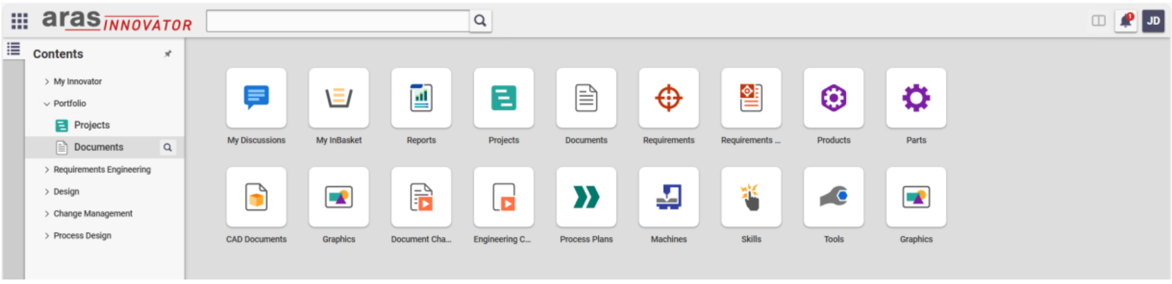


Figure 3.19 – Table of content

Permissions:

At this stage permissions of each Itemtype are set by granting different privileges to group identities based on their role in the system.

- Can add enables identities to create new items
- Get enables identities to search and open items.
- Update enables identities to edit existing items.
- Delete enables identities to delete items.
- Can Discover allow the user to search for items and view limited information about them in the search grid, but not view the detailed information that would be present on the Item form.

In order to grant students access, Team member identity used to limit their access to items inside their own team. Some Itemtypes in the list such WBS Elements or Activity2 are supportive Itemtypes for other functional Itemtypes such as Project and they be reached independently in the TOC. Table 3.12 presents the permission of each identity on Itemtypes:

ItemType	Can add	Get	Update	Delete	Can discover
Project	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
WBS Element & Sub WBS & Activity 2	World	World	World	World	World

Program	Tutors	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
Requirements	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member World
Requirements document	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
Product	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
Part	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
BOM Substitute	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
BOM Instance	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
Documents	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
CAD Documents	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
ECO	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
DCO	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
Process Plans	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
Machines	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
Skills	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
Tools	World	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member

Graphics	Tutors	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
Quality Design Document	Tutors Students	Tutors Team member	Tutors Team member	Tutors Team member	Tutors Team member
Teams	Tutors	Tutors Team member	Tutors	Tutors	Tutors Team member

Table 3.12 Permissions and identities

3.4.2 Solution Configuration

Since all applications were initially developed based on the basic needs of an organization in the standard solution provides by Aras, certain customizations were required to align them with the solution design of this project.
These modifications were implemented to adapt Itemtypes, enhance efficiency, and align the system with the course project's objectives.
The configurations included:

- Setting up appropriate functional tabs within each Itemtype to organize and display relevant information. Tabs such as Documents, Signoffs, and Changes organizes information within the system. The Signoffs tab tracks approval workflows, increasing visibility into review processes, while the Changes tab links related change requests and orders, enhancing lifecycle traceability.
- Defining lifecycle stages for some Itemtypes to manage their progression through various phases.
- Configuring workflows to automate processes and improve collaboration.
- Creating history templates to capture and store change histories, enhancing traceability and accountability.

The following table 3.13 provides a detailed overview of the configurations applied to each Itemtype.

Configuration	ItemType
Method Fill_Field	Team
Property team_id and added to form	Team
Set Tabs: Team Members	Team
Property team_id check manatory and added to form	Project
Update and scheduling mode default set: On activity completion	Project
Set variable: CorporateTimeZone=Central European Standard Time	Project
Lifecycle activated and role changed to team_member	Project
Set Tabs: Project Plan	Project
Method Filter Identities Search Dialog added to identity properties	Project
Method Filter Identities Search Dialog added to identity properties	Activity2

Set deliverables of Documents/ Parts/ Products	Project
Property customers got hidden in the form	Program
Set Tabs: Projects	Program
Set Tabs: Documents/ Related Parts	Requirement
Lifecycle disabled	Requirement
Property managed_by unchecked mandatory	Requirement
Property team_id check mandatory and added to form	Requirement
Method Filter Identities Search Dialog added to identity properties	Requirement
History template set to default	Requirement
Set Tabs: Contents/ Documents/ Related Parts	Requirements Doc
Lifecycle activated	Requirements Doc
Property team_id check manatory	Requirements Doc
Method Filter Identities Search Dialog added to identity properties	Requirements Doc
History template set to default	Requirements Doc
Set Tabs: Models	Product
Property team_id check manatory and added to form	Product
History template set to default	Product
Property team_id check manatory and added to form	Part
Property control_type removed from form	Part
Set Tabs: BOM/ BOM Structure/ Alternates/ Documents/ CAD Documents/ Goals/ Changes/ Requirements/ Requirements Documents	Part
Lifecycle activated	Part
History template set to default	Part
Method Filter Identities Search Dialog added to identity properties	Part
Property team_id check manatory and added to form	Document
Method Filter Identities Search Dialog added to identity properties	Document
Lifecycle activated	Document
Set Tabs: Files/ Changes	Document
History template set to default	Document
Property team_id check manatory and added to form	CAD Document
Method Filter Identities Search Dialog added to identity properties	CAD Document
Set Tabs: Structure/ Parents/ File/ Changes	CAD Document
Lifecycle activated	CAD Document
History template set to default	CAD Document
Property team_id check manatory and added to form	Express ECO
Set Tabs: Impact Matrix/ Attachments/ SignOffs	Express ECO
Method Filter Identities Search Dialog added to identity properties	Express ECO
Lifecycle activated	Express ECO

Workflow activated	Express ECO
Workflow assignments changed: Submit: Creator Planning: Team Member Plan Review: Team Member Draft Changes: Team Member Change Review: Team Member	Express ECO
Workflow steps tasks set (left original)	Express ECO
History template set to default	Express ECO
Property team_id check manatory and added to form	Express DCO
Set Tabs: Affected Items/ Attachments/ SignOffs	Express DCO
Method Filter Identities Search Dialog added to identity properties	Express DCO
Lifecycle activated	Express DCO
Workflow activated	Express DCO
Workflow assignments changed: Planning: Creator Draft Changes: Team Member Initial Review: Team Member Final Review: Team Member	Express DCO
History template set to default	Express DCO
Set Tabs: Produced Parts	Process Plans
Property managed_by unchecked mandatory	Process Plans
Method Filter Identities Search Dialog added to identity properties	Process Plans
Lifecycle activated and role changed to manager	Process Plans
History template set to default	Process Plans
Property team_id check manatory and added to form	Process Plans
Property team_id check manatory and added to form	Machines
Property team_id check manatory and added to form	Skills
Property team_id check manatory and added to form	Tools
Property team_id check manatory and added to form	Graphic

Table 3.13 List of configurations

Together, these configurations significantly enhance the efficiency, security, and usability of the PLM system, supporting better lifecycle management, improved collaboration, and streamlined operations aligned with best practices in PLM implementation.

3.4.3 System Interface Design

Optimizing form design is a straightforward yet powerful approach to enhancing data quality, simplifying navigation, and improving data entry efficiency. The selection of appropriate form fields not only contributes to a positive user experience (UX) but also encourages greater user participation. Conversely, poorly designed forms often result in incomplete submissions, reducing their effectiveness and not achieving the desired outcome.

This challenge was addressed by optimizing forms through the form design Graphical User Interface available in Aras.
Several best practices for form design were applied in this customization process:

- **Simplified Form Fields:**
  While it can be tempting to include a wide range of fields, it's crucial to keep forms as simple and intuitive as possible. By limiting fields to the essentials, the form became easier to complete, which encourages users to complete it. In addition, experience-enhancing custom properties, such as Description were added to provide extra convenience to users without overwhelming them.
- **Clear Differentiation Between Required and Optional Fields:**
  Clearly indicating which fields are mandatory and which are optional helps prevent confusion. This improves the user's understanding and guides them efficiently through the form.
- **Keyboard Navigation –Tab Functionality:**
  Many users prefer using keyboard shortcuts for faster form completion. The "tab" key functionality allows users to quickly navigate between form fields without needing to switch to their mouse. This small enhancement can significantly speed up the form-filling process, making it a seamless experience.
- **Smart Defaults:**
  Implementing smart defaults can streamline form completion by auto-filling fields with correct data. This not only accelerates the form-filling process but also reduces the chances of incorrect data entry, guiding users to make accurate selections.
- **Index Lists for Enhanced User Experience:**
  The index lists were optimized to display only the most relevant information, while unnecessary properties were hidden from list view. This approach allows users to focus on the key data more efficiently, enhancing both the clarity and usability of the interface. The streamlined index design provides valuable insights, improving the overall data retrieval process for users.
- **Unified Design Across All Item Types:**
  To ensure consistency and enhance data presentation, all forms and index layouts were standardized. For example, in the forms, fields such as item number, team, and status are now consistently positioned in the same locations across all forms. Similarly, in the index columns, fields like item number, name, creator, and creation date are arranged in the same order. This standardization minimizes cognitive load, allowing users to navigate forms and indexes more intuitively and efficiently.

3.4.4 Reports

Aras provides a variety of reporting tools that can be utilized in the course project, enabling students to effectively present their project outcomes. The key reports available include:

- **Multilevel BOM Report:** Shows all Parts of a BOM with their position in the hierarchy and their quantity.
- **BOM Quantity Rollup Report:** Shows total quantities of all Parts of a BOM.
- **BOM Costing Report:** Calculates the cost of a Parent from Actual or Estimated Costs of its Children.
- **Part Circular Reference Report:** Detects Circular References in a BOM

3.5 Evaluation of the Solution

The evaluation of the solution was conducted in two stages, with the possibility of a third stage in the future. The first stage took place after the system configuration was completed. In this phase, a demo of the system was presented to the teaching team of the course. The demonstration provided a detailed overview of the functionalities of the system to ensure that all desired features were correctly implemented and functioning as intended.

The second stage of the evaluation as a trial session aimed to gather deeper insights into both the usability and integration of the PLM system with its intended academic and project-based objectives. This evaluation was intended as a guide to refinements of system to ensure it is fully optimized for deployment in the upcoming course. The trial session followed the Goal Question Metric (GQM) methodology [24], a structured approach that ensures comprehensive and measurable evaluation. This process began by defining a clear evaluation goal, formulating specific questions that needed to be answered through the experiment outcomes, and finally measuring relevant metrics to address these questions.

The trial session also served as a preliminary assessment in preparation for a larger, more comprehensive experiment scheduled for the next semester within the product design course at Polytechnic of Turin. This upcoming third stage will focus on a broader evaluation of the system, involving participants with a more limited knowledge base and less familiarity with the platform. The objective is to assess how effectively the system can be adopted and utilized by users with minimal prior exposure, thereby providing valuable insights into its user-friendliness, learning curve, and overall accessibility.

3.5.1 Trial session Design

The trial program conducted for this project was a preliminary evaluation of the usability of the system and of the effectiveness of the system in supporting collaborative product development. According to GQM template, the goal can be expressed as following table 3.14:

Goal	Purpose: Evaluate usability and effectiveness
	Object: Proposed PLM Solution
	Context: Educational Environment
	Viewpoint: Students

Table 3.14 Evaluation goals

Following the GQM method two research questions were developed to pursue the goal of the evaluation. The evaluation session was designed to answers the following research questions:

One primary consideration in developing this solution was to provide a framework within Aras PLM that could be easily understood and interacted with by the students of the teaching course. However, it is essential that the usability evaluation encompasses the entire lifecycle of products, processes, and resources [24]. This holistic approach allows the results to be interpreted within the context of the environment’s characteristics and objectives, ensuring that both micro-level usability and macro-level system integration are thoroughly assessed.

Therefore, the question one addressing usability of the system is as following:

Q1: Is the system easy is to interact with by students?

Metrics:

To answer these questions two methods were used:

1. Counting the number of times the participant encountered an error while performing the tasks

2. Asking participants to complete the System Usability Scale (SUS) survey [25] upon completion of the trial. This survey consists of 10 standard statements designed to capture the user's experience with the system (see figure 3.20). Participants rated their level of agreement with each statement on a scale from 1 (strongly disagree) to 5 (strongly agree) as Likert questions. The responses were then analyzed to compute a usability score ranging from 0 to 100. A score above 68 indicates an above-average level of usability. The final score is determined using a specific calculation method that aggregates the participants' responses, providing a standardized measure of the system's usability performance.
   The score is calculated using the following equation 3.1:

   Equation 3.1 $$\text{Score} = 2.5\left(\sum(\text{Qodd} - 1) + \sum(5 - \text{Qeven})\right)$$

   Where Qodd is the value for odd numbered questions and Qeven is the value for even numbered questions.

System Usability Scale (SUS)

Please let us know how your experience has been by answering the following questions.

	Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1. I think that I would like to use this system frequently.	☐	☐	☐	☐	☐
2. I found the system unnecessarily complex.	☐	☐	☐	☐	☐
3. I thought the system was easy to use.	☐	☐	☐	☐	☐
4. I think that I would need the support of a technical person to be able to use this system.	☐	☐	☐	☐	☐
5. I found the various functions in this system were well integrated.	☐	☐	☐	☐	☐
6. I thought there was too much inconsistency in this system.	☐	☐	☐	☐	☐
7. I would imagine that most people would learn to use this system very quickly.	☐	☐	☐	☐	☐
8. I found the system very awkward to use.	☐	☐	☐	☐	☐
9. I felt very confident using the system.	☐	☐	☐	☐	☐
10. I needed to learn a lot of things before I could get going with this system.	☐	☐	☐	☐	☐

Please share your comments and suggestion for improvements regarding user experience:

Figure 3.20 – System usability survey

Q2: Does the system support the CPD process in the course project?

Metrics:

To answer these questions one method was used:

Assessing the post-trial questionnaire (see figure 3.21). This questionnaire was designed to evaluate how effectively the system addresses key CPD process objectives. Participants were asked to rank the importance and effectiveness of various system functionalities in fulfilling their needs. A ranking system was chosen for its ability to provide a clear understanding of the relative performance of different features. Additionally, an open-ended section was included to allow participants to provide detailed feedback, share their experiences, and suggest enhancements for the system.

System Effectiveness Survey

Please let us know how the system's different functionalities provide users with expected needs and support collaborative product development by answering the following questions.

	Very Poor	Poor	Fair	Good	Very Good
1. Project management	☐	☐	☐	☐	☐
2. Document Management	☐	☐	☐	☐	☐
3. Tasks Management	☐	☐	☐	☐	☐
4. Communications (Discussions)	☐	☐	☐	☐	☐
5. Requirement management	☐	☐	☐	☐	☐
6. Collaborative Product design	☐	☐	☐	☐	☐
7. Change management	☐	☐	☐	☐	☐
8. Process design	☐	☐	☐	☐	☐
9. Quality Management	☐	☐	☐	☐	☐
10. Reports	☐	☐	☐	☐	☐

Please share your comments and suggestion for improvements regarding system effectiveness in CPD projects:

Figure 3.21 – System effectiveness survey

3.5.2 Participants

To conduct the evaluation procedure two types of roles were performed. The first role was the guide, who was in charge of coordinating the session. The second was the evaluator, who was asked to interact with the system and assess its usability and functionality.
The experiment involved 5 evaluators representing the students expected to use the PLM platform in their course project. The participants were selected from Ph.D. students specializing in product engineering at the Polytechnic of Turin. They were specifically chosen for their prior knowledge of product development to ensure a comprehensive evaluation of the system's ability to support the CPD process.

3.5.3 Procedure

To conduct the evaluation session, the thesis project was first introduced, followed by an overview demo of the Aras PLM platform and its applications by the guide. This introduction helped evaluators understand the system's overall operation. Afterward, each participant was provided with the system's IP address and individual login credentials to access it independently.

The evaluation scenario focused on redesigning the Sliding Door Trolley. This product was selected for its simple structure, allowing the session to concentrate on PLM functionalities rather than product development knowledge. To facilitate this approach, scenario materials were prepared in advance and provided to evaluators during the session.

The evaluators were asked to assume the role of teammates within a single team, collaborating to redesign the final product by completing a set of assigned tasks within the system. The guide acted as the team manager, handling tasks that needed to be performed only once, such as project planning, to ensure the smooth execution of the evaluation session.

The trial involved a series of phases, each containing set of tasks designed to simulate the CPD process for the project. The tasks were carefully designed to replicate a complete project and to engage participants in critical activities of each phase.

Due to time constraints, training and task execution were conducted simultaneously. Evaluators first received a brief demonstrations of the necessary system's functionalities and task instructions in each phase. They then applied this knowledge immediately by performing the assigned tasks within the system. This approach allowed them to explore system features in real time while gaining a deeper understanding of its practical application. Each phase and tasks are presented as the following:

<table>
<tr><th colspan="3">0. Planning</th></tr>
<tr><th>ItemType</th><th>Guide Task</th><th>Evaluator Task</th></tr>
<tr><td>Discussion</td><td>- Create a group discussion
- Share the group with evaluators
- Follow evaluators to active direct messages</td><td>- Add the group discussion
- Follow other evaluators to active direct messages
- Send a confirmation message in the group discussion</td></tr>
<tr><td>Project</td><td>- Create a project
- Design WBS
- Schedule the project
- Set the team access on the project
- Assign the tasks to evaluators
- Activate the project</td><td>- Find the project and confirm having the access</td></tr>
<tr><td>Inbasket</td><td>- Confirm the assigned tasks in the basket</td><td>- Confirm the assigned tasks in the basket</td></tr>
</table>

Table 3.15 Phase planning evaluation tasks

<table>
<tr><th colspan="3">1. Concept Development</th></tr>
<tr><th>ItemType</th><th>Guide Task</th><th>Evaluator Task</th></tr>
<tr><td>Documents</td><td>- Upload the sample file of market analysis report
- Set the team access on the file
- Submit a message associated with the file</td><td>- Upload the sample file of market analysis report
- Set the team access on the file
- Submit a message associated with the file
- Complete the market analysis task in the inbasket</td></tr>
<tr><td>Documents</td><td>- Set the team access on the requirement file</td><td>- Confirm the access on the requirement file</td></tr>
<tr><td>Requirements</td><td>- Define requirements</td><td>- Define requirements</td></tr>
<tr><td>Requirement Documents</td><td>- Define requirement document</td><td>- Define requirement document</td></tr>
</table>

	- Add the requirements into the req. document	- Add the requirements into the req. document - Complete the requirement analysis task in the inbasket
Product	- Define the product - Define a Model - Link the model to the product - Set the access on the product	- Confirm the access on the product - Define a model - Complete the product design task in the inbasket

Table 3.16 Phase Concept Development evaluation tasks

2. System Level design		
ItemType	**Guide Task**	**Evaluator Task**
Documents	- Set the team access on the BOM and MPP file	- Confirm the access on the BOM and MPP file
Parts	- Define Parts - Design the BOM - Link the requirements to each part	- Define Parts - Design the BOM - Link the requirements to each part - Complete the BOM design task in the inbasket

Table 3.17 Phase System Level design evaluation tasks

3. Detail Design		
ItemType	**Guide Task**	**Evaluator Task**
ECO	- Release parts using the ECO workflow	- Release parts through the ECO workflow - Complete the release parts task in the inbasket
ECO	- Revis parts and BOM design using the ECO workflow	-
Machines	- Define machines	-
Tools	- Define tools	-
Process	- Define MPP process - Add machines to the process - Add tools to the process	- Define MPP process - Add machines to the process - Add tools to the process - Complete the MPP task in the inbasket

Table 3.18 Phase Detail Design evaluation tasks

Finally, Evaluators were asked to complete two post-trial questionnaires. These questionnaires aimed to capture their overall experience, feedback on system usability, and insights into its effectiveness in supporting real-world collaborative product development.

4. System Demonstration

As part of this research, a system demonstration of the developed solution on Aras Innovator was conducted to provide an overview of the system's functionalities and user interface. The demonstration highlights key functionalities of the system. By walking through these features, users can gain a clear understanding of how Aras Innovator facilitates collaborative product development, ensures data traceability, and enhances workflow efficiency.

The following sections will provide a detailed breakdown of the system's layout, covering the main interface components, navigation structure.

4.1 Logging On

The login page (see figure 4.1) is built on Windows Authentication Framework, allowing secure access to the Aras Innovator client directly from a web browser. Upon navigating to the system, users are prompted to select the intended database and enter their user credentials to authenticate and gain access to the platform.

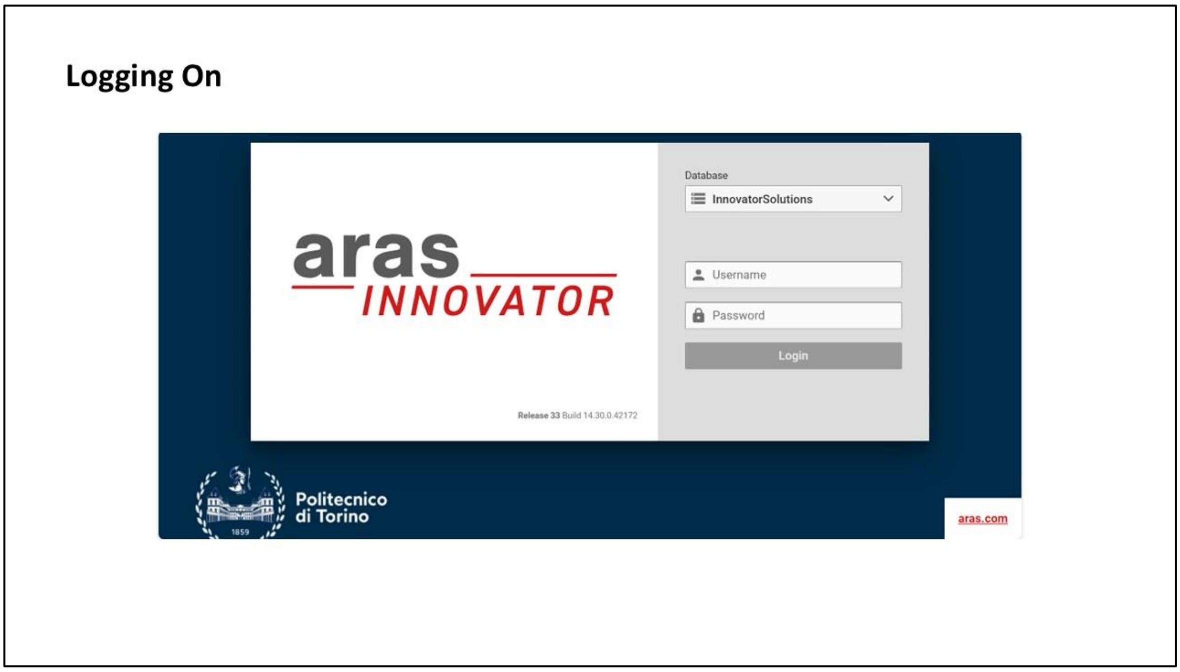


Figure 4.1 – Login Page

4.2 Main Window

The Main Window (see figure 4.2) in Aras Innovator serves as the central workspace where users interact with the system. It provides an intuitive interface for navigating through different functionalities.

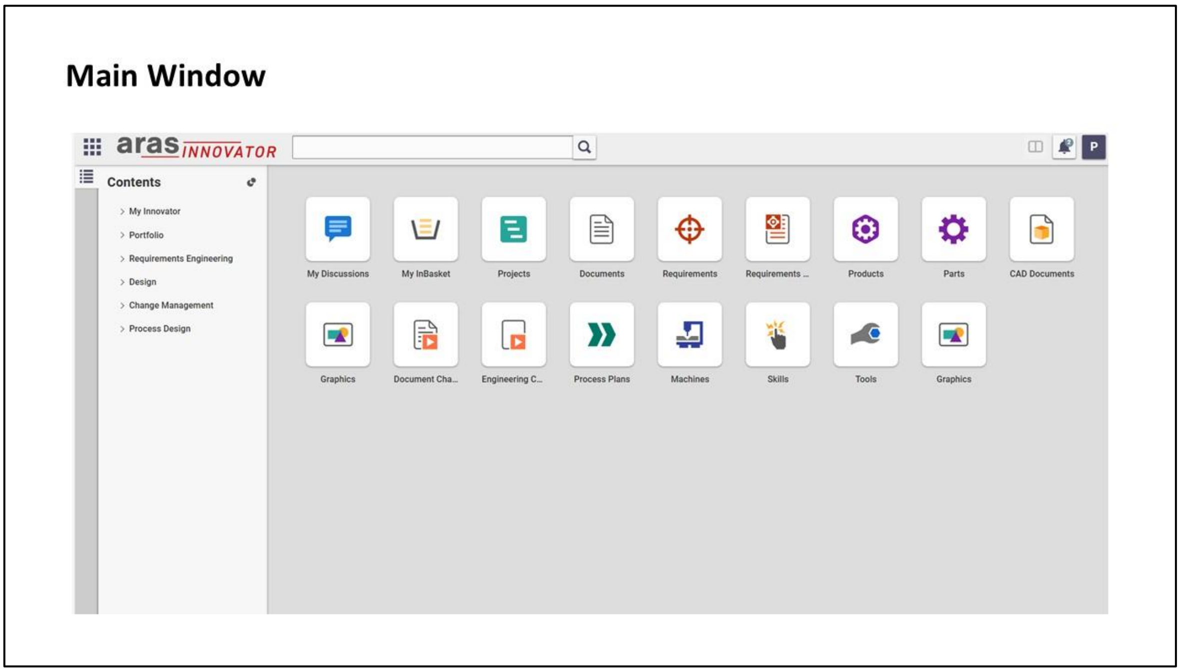


Figure 4.2 – Main Window

4.3 Main Window Toolbar and User Settings

The Main Window Toolbar (see figure 4.3) in Aras Innovator is a key interface element that provides users with quick access to essential sections. It is located at the top of the Main Window and contains various buttons and controls such as:

- The Navigation button which provides access to the Navigation panel.
- The Enterprise Search field enables search across items in the database. This function could be activated by purchasing subscription code.
- The Splitscreen enables to view item tabs and search grids side-by-side.
- The Notifications button displays alerts to users including Aras Innovator updates and system maintenance.
- The User Menu provides access to user preferences, reports, and custom actions.

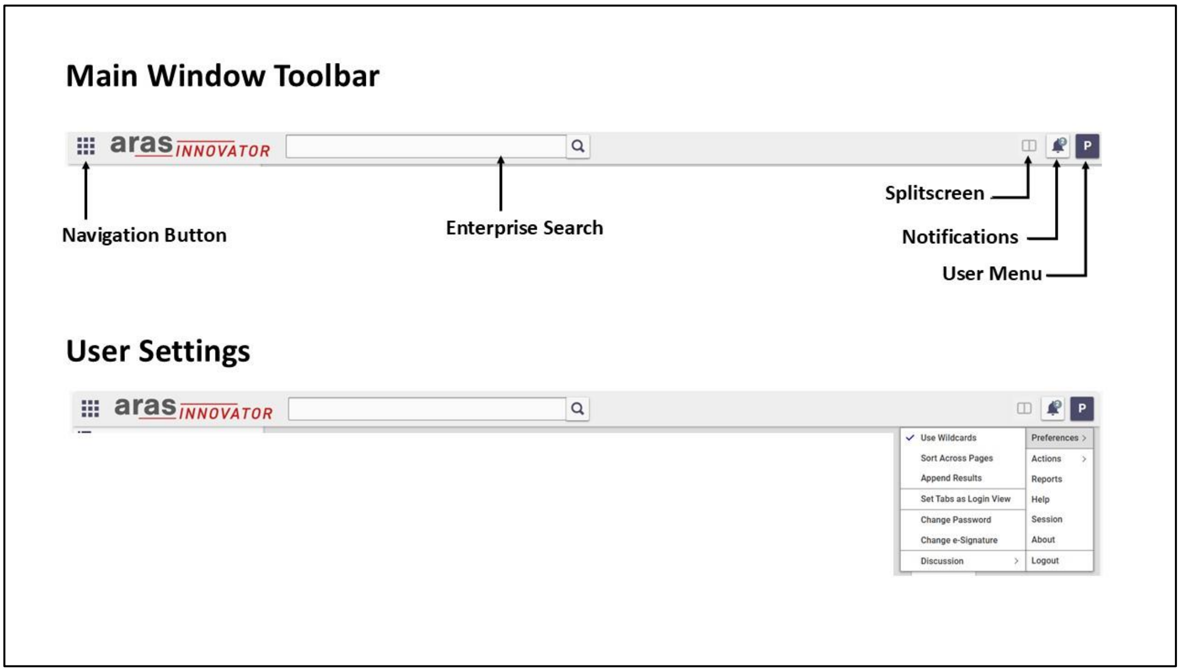


Figure 4.3 – Main Window Toolbar and User Settings

4.4 Navigation Panel

The Navigation panel (see figure 4.4) serves as the primary interface for accessing Itemtypes. Designed to provide a structured and intuitive navigation experience, it presents Itemtypes in expandable categories, allowing users to efficiently browse through different modules and data categories. The panel is highly flexible and can be pinned to remain visible at all times or collapsed to maximize screen space. Upon selecting a specific Itemtype, the panel dynamically updates to display a secondary menu containing functionalities unique to that ItemType. This ensures that users have quick access to relevant actions and data associated with the selected module. This menu contains search button to open a search grid for that Itemtype and create button to create a new instance of that Itemtype.

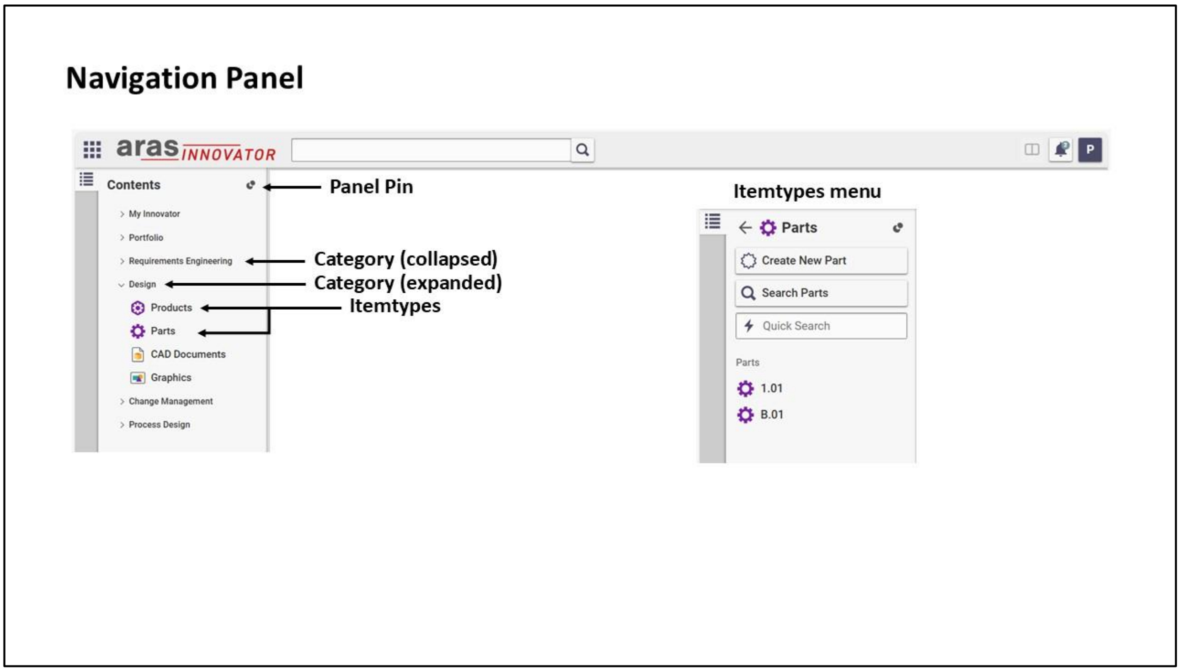


Figure 4.4 – Navigation Panel

4.5 Search Grids

Search Grids (see figure 4.5) are dynamic, interactive tables that allow users to quickly locate, filter, and manage Itemtype instances within the system. They provide a structured way to view and interact with large datasets, making it easier to retrieve relevant information from the database. Paging controls serves to explore through the results and control the number of viewable results. The command bar serves to execute searches, customize the properties displayed in the search grid columns, and export search results as Microsoft Word, Excel, or PDF documents. The search row to supports search criteria in one or more item properties displayed as columns in the grid.

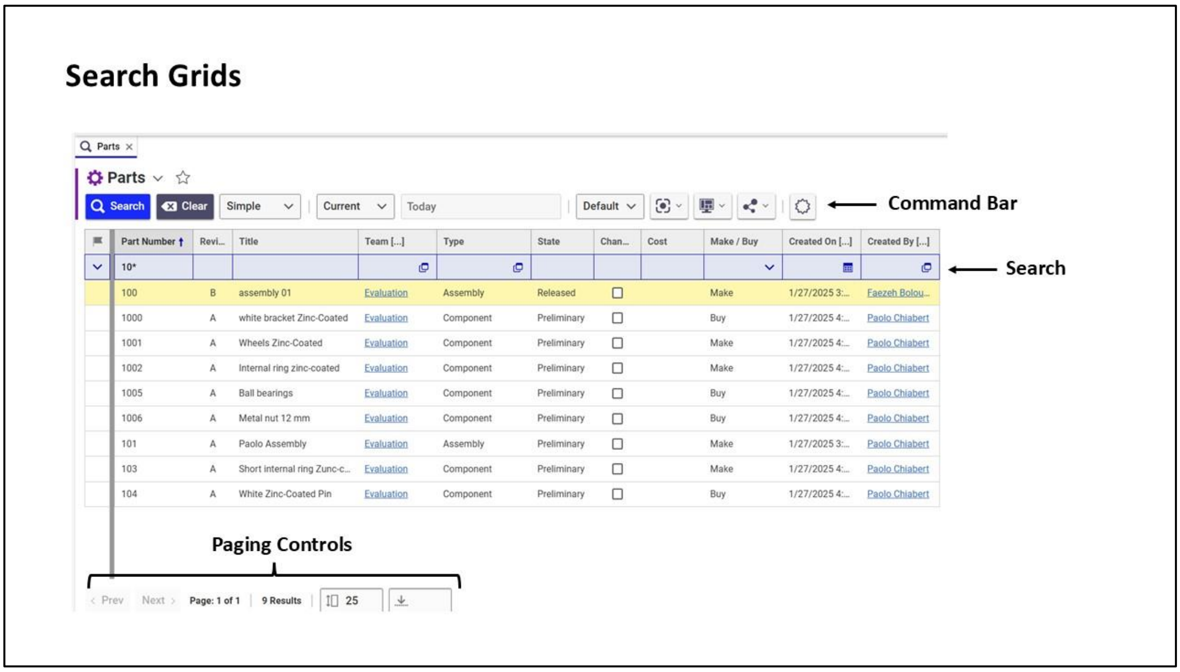


Figure 4.5 – Search Grids

4.6 Item View

The Item View (see figure 4.6) is the primary interface for displaying, editing, and managing the details of an individual Itemtype instance. It provides users with a structured and interactive form that presents all relevant data, relationships, workflows, and lifecycle states of an item. When a user selects a specific item from a search grid or creates a new instance, the Item View opens to provide a structured form where all relevant data, relationships, and lifecycle information are presented. Upon opening an item, the Item View initially appears in a read-only format, displaying key item properties in the top section and related items in the lower section. This structured layout allows users to quickly review an item's details while maintaining visibility over its dependencies and connections within the PLM system. Each section of the form can be collapsed or expanded using accordion-style arrow buttons at the top left, providing a customizable view to improve usability and focus on relevant information. **The Claim** function allows a user to reserve an item for future editing, ensuring exclusive control over modifications. Once an item is claimed, it prevents other users from making changes until the claim is released, maintaining data integrity and preventing conflicting edits in collaborative environments. **The Favorite** function enhances navigation efficiency by allowing users to bookmark frequently accessed items. When an item is marked as a favorite, it is added to the navigation panel under its respective category, enabling quick and easy access without searching through extensive data lists. **The Command Bar** at the top of the Item View provides access to essential actions and controls. Available actions may vary depending on the ItemType and user permissions, but commonly include Edit, Delete, Versioning, Lifecycle Transitions, and Custom Actions. **The Discussion Panel** enables collaborative communication directly within the Item View. Users can create new comments, respond to remarks, and engage in discussions about the item. This feature helps capture feedback, clarifications, and issue tracking within the PLM system, reducing the need for external communication. Discussions can be Filtered, Sorted, and Searched.

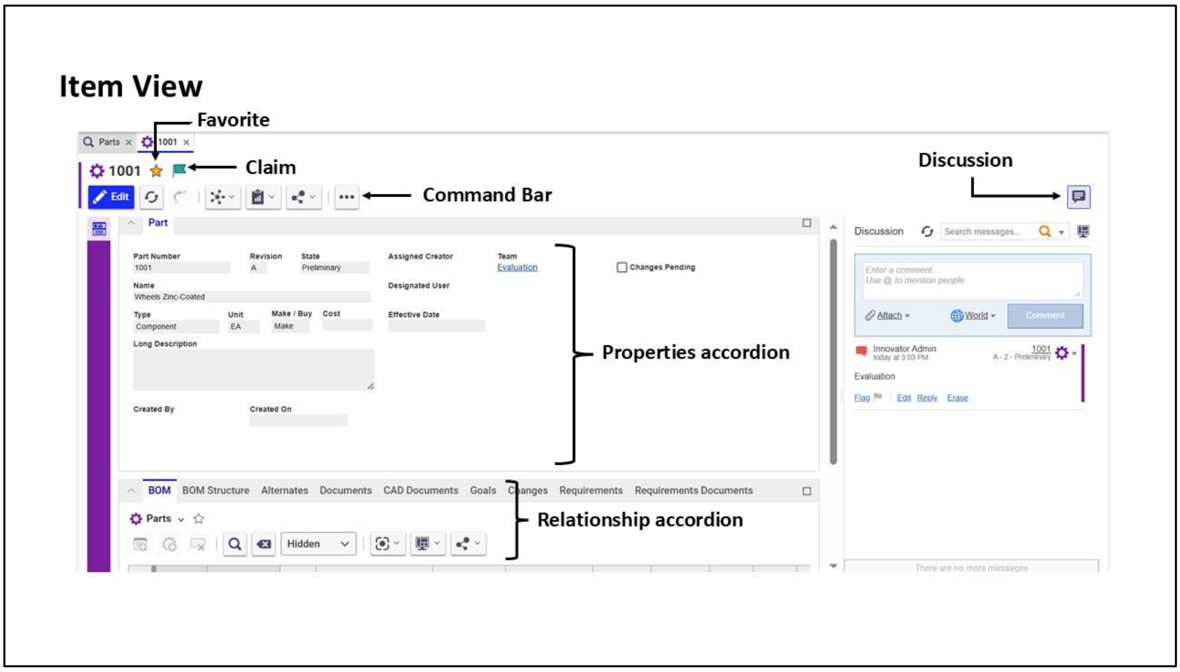


Figure 4.6 – Item View

4.7 Reports

Reports provide a structured way to present information, enabling users to generate insights for decision-making, compliance tracking, and performance monitoring across different business processes. When reports are configured on the Itemtype, they become accessible through the Reports button (see figure 4.7), located within the command bar, the report opens in a new tab, displaying the retrieved data in a well-structured format. This interactive report view allows users to review and analyze the output efficiently. Additionally, reports provide a built-in print functionality, enabling users to generate hard copies or export data for documentation, auditing, or further analysis.

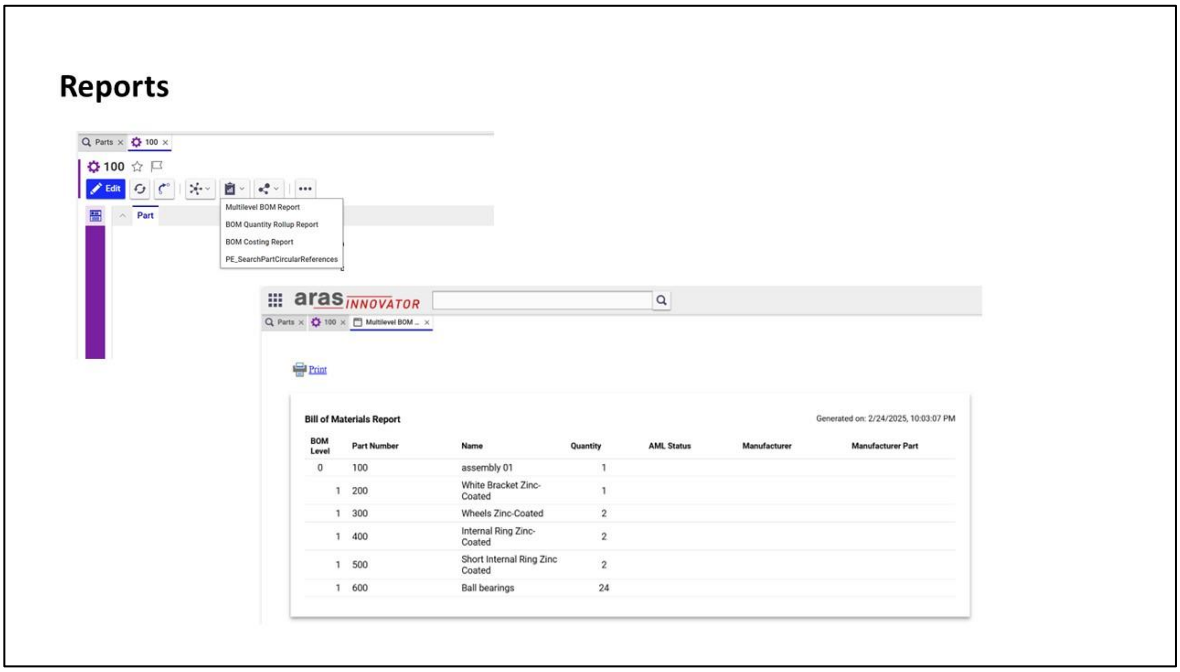


Figure 4.7 – Item View

4.8 Discussions

The Discussions Itemtype (see figure 4.8) provides a structured and interactive space for users to engage in collaborative discussions directly within the PLM system. This feature enhances communication by allowing conversations to be organized efficiently while maintaining contextual links to relevant data.

One of the key functionalities of the Discussions is its Bookmarking capability, which enables users to organize discussions based on authors, participants, or specific items. This ensures that important conversations are easily accessible and categorized for quick reference.

Additionally, users have the option to create named Forums, which serve as dedicated discussion spaces for specific topics, projects, or functional areas. This functionality ensures that critical conversations remain connected to business processes, reducing the need for external communication tools while preserving a traceable record of discussions for future reference.

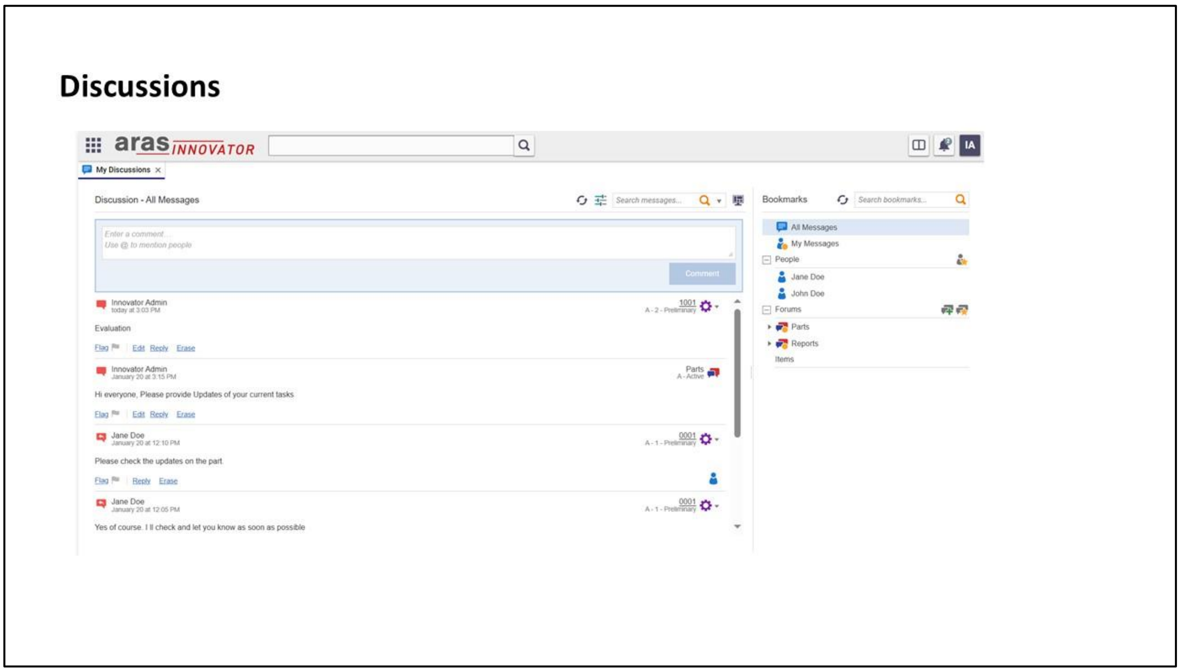


Figure 4.8 – Discussions

5. Results

This section presents the findings from the evaluation session, including both the recorded metrics corresponding to the two research questions analyzed and general observations from interactions with the evaluators.
The evaluation involved a group of experts with diverse backgrounds relevant to the study. The evaluators who actively participated and completed the questionnaires included:

- Three PhD researchers specializing in mechanical engineering.
- One professor with expertise in mechanical engineering and product development education.
- One PhD researcher specializing in information systems.

Additionally, other evaluators with similar backgrounds were present during the evaluation session but did not complete the questionnaires. Instead, their interactions and behaviors were observed qualitatively to complement the recorded results.

5.1 Usability (Q1)

The first evaluation criterion focused on the **usability** of the system.
Two key metrics were considered:

1. The number of times evaluators encountered errors while performing assigned tasks.
2. A usability survey completed by evaluators at the conclusion of the trial.

Notably, no issues were reported during the process, indicating that the system's configurations functioned correctly without errors. The system was functional and free of critical errors.

To assess usability quantitatively, the **System Usability Scale (SUS)** method was applied. Each evaluator's score was calculated using equation 3.1, and the final usability score was determined as the average across all evaluators' responses. The results are summarized in the table 5.1 below:

	Evaluators				
	1	2	3	4	5
Score	57.5	72.5	47.5	75	65
Average	63.5				

Table 5.1 Usability evaluation results

The obtained average SUS score was 63.5, which falls slightly below the commonly accepted usability benchmark of **68** for software applications. However, it is essential to consider the context in which the SUS was applied.
First, SUS is traditionally used to evaluate newly developed software, whereas this project focused on the implementation of an existing system rather than the creation of a new one. As a result, the interpretation of the score should account for the fact that users were adapting to an established system rather than assessing a product designed from scratch.
Second, this evaluation was conducted as a pilot study, primarily aimed at identifying areas for improvement rather than achieving an optimized usability score at this stage. The insights gathered from the evaluation will serve as a foundation for refining the system's usability and enhancing its alignment with user expectations in future iterations.

A detailed analysis of individual responses reveals that several potential factors have contributed to the lower SUS score.

1. The evaluators were unfamiliar with software user interface structure. Aras PLM employs a hierarchical, multi-panel layout that requires users to switch between different sections (for example ItemType, Itemtype Tabs). In the absence of familiarity with this structure, evaluators struggled to locate key functions or understand the relationship between different components.
2. Aras PLM specific terminology was not immediately intuitive for evaluators. (for example Requirements and Requirement Documents). The distinction between different component was not clear to evaluators, leading to confusion when linking product data.
3. Limited time of training led to absence of clear, step-by-step guidance, which in turn made it harder for users to fully grasp the capabilities of the system.

This suggests that while the core functionalities of the system are sound, usability articulated as ease of learning and intuitive interaction, can be improved.

5.2 Effectiveness (Q2)

The second research question focused on evaluating the **effectiveness of the system** by studying key CPD process objectives. The results of this evaluation determined which functionalities were perceived as useful, which needed improvement, and whether any functionalities were ineffective or created obstacles for users.

Participants were asked to rank the importance and effectiveness of various system functionalities in fulfilling their needs.

The score of post-trial questionnaire on system's effectiveness is considered key metric.

To quantify effectiveness, each of the functionalities was rated on a **five-point scale**, where evaluators assigned a score based on their experience:

Very Poor	Poor	Fair	Good	Very Good
0	1	2	3	4

Table 5.2 Effectiveness scale

Given that the questionnaire was completed by a total of five evaluators, the maximum possible score for each functionality was 20 (calculated as 5 evaluators × highest individual score of 4). This score serves as a benchmark for evaluating effectiveness, where higher scores indicate strong functionality alignment with user expectations, while lower scores suggest gaps in effectiveness that may require further investigation and refinement.
By comparing the obtained scores against this maximum value, we can **quantify the relative effectiveness of each functionality**, identify which features need improvement, and prioritize enhancements to optimize system performance in future iterations.

Since all functionalities included in the solution were expected to perform at a consistently high level of effectiveness, any low scores in the evaluation results indicate areas where the system did not meet user expectations, which could stem from insufficient performance, limitations in application capabilities, lack of alignment with user needs, inadequate training on the tool, or gaps in functionality coverage. Identifying these shortcomings is essential for refining the system, as it highlights key areas for enhancement, ensuring that future iterations better align with user needs and expectations.

The following sections will present a detailed analysis of the results (see table 5.3), outlining both the well-performing functionalities and those requiring further improvements.

	Evaluators					
	1	2	3	4	5	Total Score
Project Management	4	4	4	3	4	19
Tasks Management	4	4	4	2	4	18
Document Management	2	4	4	3	4	17
Process Design	3	4	3	3	4	17
Collaborative Product Design	3	4	3	2	4	16
Reports	2	4	3	3	4	16
Requirement Management	2	4	3	2	4	15
Change Management	2	4	2	4	3	15
Product Engineering	2	4	3	2	4	15
Discussions	2	4	3	1	4	14

Table 5.3 Effectiveness evaluation results

Project Management (Score: 19)

Among the functionalities assessed, Project Management received the highest score of 19, indicating that users found this feature to be highly effective. This score reflects that the tools and features provided for managing projects were well-aligned with user expectations. The project management functionality effectively supported various stages of project planning, assignments and milestone tracking. Users found the application intuitive and helpful in project management. The ability for team members to collaborate efficiently, share project updates, and communicate directly within the system likely enhanced team coordination and transparency of workload. Overall, the Project Management functionality demonstrated a high level of effectiveness, with users perceiving it as a valuable tool for driving project success within the PLM environment.

Tasks Management (Score: 18)

Task Management plays a crucial role as part of the core functionality of the system. For the purposes of this evaluation, the task management features were included in the questionnaire to assess how well these tools support the tracking and organization of tasks along with the project management framework. Task management received a solid score of 18, reflecting its significant contribution to the overall effectiveness of the system.
This score highlights the effectiveness of this functionality in structuring task and tracking their progress, thus improving productivity and collaboration. These features helped evaluators maintain a high level of organization and accountability by tracking the

completion status of each task and ensuring that responsibilities were clearly defined. Task management did not operate in isolation, but rather seamlessly integrated with the project management, making it an essential tool for keeping the project on track.

Document Management (Score: 17)

Document management also received a score of 17, reflecting its significant contribution to the overall effectiveness of the system. The evaluators found the documents well-suited to their needs, facilitating efficient storage, retrieval, and sharing of documents. They found it easy for team members to find the documents they need without searching through different systems or folders. Also, the ability to collaborate on documents and, open discussions on them positively affected the evaluation results. In addition to functionalities of Document management itself, its integration with other sections such as Project, Tasks, and Parts ensured that the correct documents are always available in the right context, improving the overall experience in the system.

Process Design (Score: 17)

Evaluators could easily insert production operations, machines, tools, and required skills within each phase of the process, ensuring a clear and organized production routing. Additionally, the integration with parts data played a crucial role in maintaining alignment with project information, promoting consistency, and ensuring traceability throughout the product lifecycle. This strong integration between process management and product data reinforced the system's effectiveness, making it a valuable tool for improving efficiency and standardization.

Collaborative Product Design (Score: 16)

This functionality received a score of 16, underscoring its significant role in facilitating teamwork and enhancing the product development process. By offering a centralized repository for all product-related data, system ensured evaluators with access to the latest generated data and effective contribution throughout the product lifecycle. However, the lower-than-expected score suggests that evaluators faced challenges in fully leveraging the available features. Limited familiarity with advanced features was the first factor lowering the score. Evaluators had limited exposure and training that led to challenging to utilize all available capabilities efficiently.

Reports (Score: 16)

The score of 16 reflects BOM multilevel and BOM cost report effectiveness, while also highlighting need for improvement and providing more intuitive reports.

Requirement Management (Score: 15)

The score 15 indicates requirement management capacity to satisfy the base and original needs of such system to handle diverse requirement types and maintain traceability throughout the product development process.

Change Management (Score: 15)

Evaluators recognized Change Management as a necessary and highly useful function within the system, particularly for ensuring structured and controlled modifications throughout the product lifecycle. However, the score of 15 suggests that while the feature is valuable, evaluators encountered challenges in fully leveraging its capabilities. One of the primary difficulties reported was the complexity of executing change workflows, which impacted overall effectiveness. Some evaluators found the process less intuitive, requiring additional time and effort to navigate through approval stages, impact assessments, and revision tracking

Product Engineering (Score: 15)

As this application serves as the foundation of core data within the PLM system, evaluators acknowledged its importance and benefits in managing product information. However, they expressed concerns that the module did not fully meet all their expectations in terms of usability and efficiency, which impacted its overall effectiveness score.
One of the key challenges identified was the breadth and depth of functionalities offered within the Product Engineering application. The assessment revealed that a complete understanding and effective use of these features demanded more time than was available within the evaluation timeframe. In fact, the complexity of the module, combined with its high level of integration with other applications, meant that users could not explore all functionalities in the limited time frame.
This suggests that a longer evaluation period and more training sessions could provide a clearer picture of the module's effectiveness.

Discussions (Score: 14)

The score of 14, indicates that while some evaluators recognized its importance in facilitating collaboration, others found that initial setting of discussions and locating discussion across different components required extra effort, reducing efficiency.

5.3 Observations and conclusion of results

Considering all evaluators who participated in the session, regardless of whether they completed the surveys, one of the key findings was that they needed more time to familiarize themselves with the system and effectively navigate its functionalities. Many evaluators encountered difficulties in locating specific features, indicating that the interface and overall navigation were not immediately intuitive. This suggests that an extended onboarding process, incorporating guided tutorials and multiple hands-on training sessions, could significantly improve user adaptation and confidence in using the system.

Furthermore, since the evaluators had prior knowledge of PLM and product development, they raised questions about specific process requirements that were not covered in the test scenario. This limitation may have influenced their evaluation, as certain expectations were left unaddressed. Therefore, it became evident that the test scenarios should be expanded to encompass all critical requirements within the system.

By integrating a broader range of test cases and conducting multiple evaluation sessions, users would have the opportunity to explore additional functionalities, leading to a more comprehensive and accurate assessment of the system. This iterative approach would not only provide valuable insights for refining usability and optimizing workflows but also ensure that the system meets user expectations and aligns with industry-specific requirements. Ultimately, a more structured and inclusive evaluation process would enhance the overall effectiveness and adoption of the system.

Expanding on this, the professor of the course suggested that the main training sessions course with students should be designed to align with the theoretical framework of New Product Development (NPD) while also emphasizing how Aras PLM functionalities support concurrent engineering (CE). A well-structured training program should not only introduce the fundamental concepts of NPD but also demonstrate how Aras enables real-time collaboration, seamless data integration, and efficient cross-functional teamwork.

Moving to PhD students' feedback, it was primarily focused on the system's functionalities, and in particularly on aspects related to the platform development and UI design. Their comments highlighted areas where usability and responsiveness could be improved to enhance the overall user experience.

One issue was the need to manually refresh the page to see updates or changes within the system, especially when following the provided guide. The absence of real-time updates disrupted the workflow, causing inefficiencies and frustration among users. Manually refreshing the page introduced unnecessary steps, making the interaction less seamless and increasing the time required to complete evaluations. This limitation was particularly problematic in scenarios where users needed to track modifications or collaborate with others in real-time. This recurring issue underscores a crucial area for technical improvement. Enhancing real-time synchronization and automatic updates would significantly improve both usability and efficiency. By addressing this limitation, the system can provide a more intuitive, streamlined, and user-friendly experience, ultimately improving adoption and productivity.

Another notable issue raised by users pertained to the UI design of various sections of the system, particularly the discussion feature. Users found the current interface less intuitive and suggested that it could be redesigned to resemble the user experience of common chat applications. By refining the UI to align with familiar chat application interfaces, the system could foster smoother collaboration, reduce learning curves, and improve user engagement. Implementing these enhancements would make discussions more accessible, ultimately leading to better information exchange and increased productivity within the platform.

The evaluation session revealed several important insights that can inform future improvements in both the system and its training program. A key finding was the need for additional time and a more structured onboarding process to help users familiarize themselves with the system. Many evaluators struggled with navigation, indicating that intuitive user interfaces and more guided training sessions would enhance the learning experience. Expanding the test scenarios to cover a broader range of use cases would also provide users with a more comprehensive understanding of the system's functionalities and ensure that all relevant requirements are addressed.
Furthermore, integrating theoretical frameworks like New Product Development (NPD) and concurrent engineering (CE) into the training program would better align the educational

experience with real-world practices, ensuring that users can leverage the full potential of Aras PLM in promoting cross-functional collaboration and efficient workflows.
User feedback, particularly from PhD students, highlighted technical and design issues that need to be addressed. The need for manual page refreshes and the lack of real-time updates was identified as a significant pain point, impacting workflow efficiency. Implementing automatic updates and real-time synchronization would address this challenge and improve the overall user experience. Additionally, the UI design, especially within the discussion features, could benefit from enhancements to make it more intuitive and aligned with familiar chat applications, thereby fostering better collaboration and reducing learning curves.
Overall, these findings suggest that with further refinement to both the system's functionalities and the training approach, Aras PLM can become a more effective and user-friendly platform for product development, ultimately supporting more efficient and innovative outcomes.

5.4 Evaluating Eppinger’s methodology in system design

This project was developed using Eppinger’s Product Development Process as a methodological framework for designing the PLM system. Therefore, part of the evaluation focuses on assessing the effectiveness of Eppinger’s methodology in guiding the design and deployment of the PLM system.

Eppinger’s product development model provides a structured and systematic approach to managing the product lifecycle, making it particularly valuable for the implementation of a PLM system. This methodology allowed to break down the implementation process into clear, iterative stages, ensuring that each phase was aligned with the specific needs of the case study. Following this structured approach, enabled to methodically analyze system requirements, map out data flows, and establish a logical sequence for integrating different PLM functionalities.
A key strength of Eppinger’s model is its emphasis on concurrent engineering and cross-functional collaboration, which are essential in streamlining data management and improving decision-making during software implementation. While this project was not a team effort, the principles of collaboration embedded in the model still proved invaluable. Treating different aspects of the implementation—such as data structuring, workflow automation, and system customization—as interconnected tasks rather than sequential steps, enabled to anticipate challenges, minimize rework, and ensure smoother integration.
Furthermore, the model’s iterative nature allowed for continuous refinement, enabling to test functionalities, address inefficiencies, and adjust configurations without disrupting the overall implementation flow. This adaptability was crucial in handling unforeseen complexities and ensuring that the final PLM setup was both efficient and scalable. Overall, applying Eppinger’s framework resulted in a more structured and methodical implementation, improving traceability, reducing redundancies, and fostering a more cohesive and well-organized product development environment.

6. Conclusion and Further Development

Given the evolving landscape of engineering education, there is a growing emphasis on providing students with hands-on experience in Product Lifecycle Management (PLM) and Collaborative Product Development (CPD) during their studies. PLM serves as a structured framework that helps students refine their design methodology, fostering a systematic approach to problem-solving. Before engaging in professional collaboration, future engineers must develop a deep understanding of how this approach functions to ensure effective teamwork in real-world settings. From an educational point of view, PLM can be regarded as a sophisticated tool for enabling students to enhance their problem-solving capabilities, strengthen their design skills, and develop a more comprehensive understanding of engineering system behavior. By integrating PLM into the curriculum, students gain valuable experience in managing complex product development processes, preparing them for the demands of modern engineering industries.

The designed platform in this thesis covers functionalities required in an educational setting and facilitates teaching PLM and CPD practices in product design course and acts as a teaching tool at Polytechnic of Turin. The system was designed to support students to perform the PD student projects on a platform managed by the course tutor.

The implementation process was structured according to Eppinger's product development approach, thus ensuring a methodical and industry-aligned workflow that met both user expectations and business requirements. The process began with the Planning phase, during which data collection and requirement gathering were conducted through stakeholder interactions to gain insights into business needs and operational challenges. Simultaneously, a study of the Aras PLM platform was carried out to understand its capabilities, available functionalities, and potential configurations that could support the project. A mission statement was developed as a key output of this phase, providing a clear direction for the next stage. In the Concept Development phase, a detailed requirement analysis was carried out to refine the system's scope, aided by use case diagrams that helped visualize functional interactions. Several alternative solutions were generated for each use case scenario, evaluated, and selected based on feasibility and alignment with project objectives. Following this, the System-Level Design phase defined the overall architecture, detailing the structural components and their interactions to meet the case study's specific needs. This phase ensured that different modules and system elements were properly structured to support seamless data management, workflow execution, and traceability. Once the system architecture was established, the Detailed Design phase focused on the precise configuration and implementation of all system functionalities, including workflows, data models, and user interface components. Every configuration was meticulously refined to optimize usability, efficiency, and system integration. By following Eppinger's structured development model, this phased approach ensured that the PLM system implementation was iterative, scalable, and effectively tailored to the case study requirements.

While the primary system design is crucial for defining core functionalities, the true success of implementation depends largely on thorough testing and a smooth go-live phase. However, before progressing to these stages, it is essential to assess the effectiveness and quality of the initial design to ensure a solid foundation for the system.

To ensure a fair evaluation of the designed system, it is important to consider that Aras Innovator, as an open-source platform, offers extensive flexibility and customization options. However, this flexibility comes with the challenge of requiring complex configurations to unlock its full potential. A deep understanding of Aras Innovator's configuration and functionalities is crucial for successful customization and for overcoming potential challenges during implementation. Without sufficient expertise in these areas, the process can become difficult and time-consuming. Access to direct ARAS support and configuration manuals at the early stages of the project can help streamline implementation and address technical challenges more efficiently. The web-based nature of the Aras platform eliminates the need for a client installation, making it more accessible for users. However, certain features, such as CAD and Office integrations, require additional installations for each client. Given the scope of this project, these installations were not feasible, and as a result, these features could not be utilized.

To evaluate the system, a trial program was conducted to evaluate the usability of the system and the effectiveness of its functionalities in applying PLM and CPD practices. The program involved participants representing the product engineering students, who were asked to complete a number of tasks using the system.

The usability of the application performed moderately well, scoring an average of 63.5 on the System Usability Scale (SUS). While this score is slightly below the industry standard of 68 for software applications, it is important to note that SUS is typically used for software development evaluation, whereas this project focused on implementation rather than development. Therefore, the usability score should not be interpreted as a direct indication of poor performance but rather as an opportunity for refinement. Participants were generally pleased with the aesthetics, layout, and intuitiveness of the system design. From feedback received during the trial, it was noted that some participants struggled with the complexity of certain PLM functionalities, requiring additional support and training materials. The effectiveness of individual functionalities was also assessed. The project Management and task management were the two top most effective elements among the all the functionalities included in this study. Elements related to progress tracking and guided assistance also performed well. The effectiveness of the tool at improving students' understanding of PLM adoption and increasing motivation of student engagements found to be convincing.
The observations suggested that the PLM training application had a positive effect on improving the trial participants' learning and engagement with system, though usability refinements could further enhance their experience

Despite the system's good performance on usability, several recommendations were proposed to improve it further. These recommendations included enhancing real-time synchronization to eliminate the need for manual page refreshes and adapting the instructional content to be more intuitive and accessible to students with different levels of familiarity with PLM concepts. It was also suggested to implement additional reference materials within the system, including a comprehensive tutorial to introduce the PLM components, their usage, and step-by-step instructions for completing key tasks.

This project demonstrated the potential of PLM-based digital learning tools and their components in motivating students to engage with a structured product engineering education. However, further analysis is recommended to better understand and quantitatively prove its effectiveness by conducting a larger experiment that integrates the PLM training tool within a real classroom environment.

6.1 Further developments

Based on the obtained results, user feedback, and the potential for achieving full PLM implementation, the following recommendations are proposed.

6.1.1 Extended experiment

For the next experiment, it is important to enhance both theoretical and practical aspects of PLM evaluation. The following recommendations are aimed at improving the evaluation quality and ensuring that the system is well-prepared to fully support the students throughout the case study course.

These recommendations include expanding training sessions, refining course structure, and enhancing student motivation and involvement.

- **Expanding theoretical material in training sessions:** Offering more in-depth theoretical training on PLM will provide evaluators with a solid foundation before they begin to evaluate the system. A deeper understanding of PLM concepts and methodologies will enable evaluators to perform a more thorough assessment.
- **Structure of training sessions:** It is better to perform training in multiple sessions. The training sessions should have clear expected outcomes outlined week by week, allowing evaluators to track their progress and develop a deeper understanding of the system capabilities.
- **Provide motivational support for evaluators:** To keep evaluators engaged and motivated throughout the evaluation process, it is essential to provide continuous motivation.

6.1.2 Extended functionalities

As Aras PLM offers a highly flexible and customizable platform, several enhancements can be considered to further extend its functionality and improve overall system efficiency. The proposed developments are as the following:

- **Report Service Subscription:** By subscribing to the Report Service, organizations can unlock the ability to create more intuitive and customizable reports tailored to specific needs. This will empower users to automatically receive updates on key project metrics, product performance, and other critical data.
- **Tutor Dashboard:** A Tutor Dashboard can be developed as a central control panel for educators or supervisors. This dashboard would offer a comprehensive overview of student activities, project statuses, and individual progress toward meeting deadlines. By integrating key metrics such as task completion rates, milestone achievements, and upcoming deadlines, the dashboard can help tutors quickly identify students who may need additional support or attention. The system could also include performance indicators, such as task accuracy, time spent, and engagement levels, providing a data-driven approach to student management.
- **Approved Manufacturers List (AML) Functionality:** The AML feature can be further enhanced to provide advanced supplier and part management capabilities. This functionality will be vital for managing the sourcing of specific components or

materials. By linking each part to an approved list of manufacturers, users can ensure compliance with quality standards and simplify procurement processes. For students or new users, this feature could provide a practical understanding of OEM (Original Equipment Manufacturer) relationships and the importance of supplier management systems.

- **Development of bulk user import:** An important enhancement for Aras PLM would be the development of a feature that allows for the bulk import of users into the system. This feature can be designed to allow tutors to upload a CSV or Excel file containing user information, such as names, student number, and other relevant data to create many users all at once.

Bibliography

[1] U. Srinivasan, D. Mittel, D. Lemberger, and A. Perzylo, “Structured Product Engineering: Ontology-based Advanced Product Quality Planning.” [Online]. Available: www.sciencedirect.comwww.elsevier.com/locate/procedia2212-8271

[2] J. Sauza Bedolla, G. D’Antonio, F. Segonds, and P. Chiabert, “PLM in engineering education: A pilot study for insights on actual and future trends,” in *IFIP Advances in Information and Communication Technology*, Springer New York LLC, 2017, pp. 277–284. doi: 10.1007/978-3-319-72905-3_25.

[3] J. S. Bedolla, F. Ricci, J. M. Gomez, and P. Chiabert, “A tool to support PLM teaching in universities,” in *IFIP Advances in Information and Communication Technology*, Springer New York LLC, 2013, pp. 510–519. doi: 10.1007/978-3-642-41501-2_51.

[4] J. Mora-Orozco *et al.*, “PLM in a didactic environment: the path to smart factory,” pp. 640–648, 2015, doi: 10.1007/978-3-319.

[5] Y. Isaac Chang and C. L. Miller, “PLM Curriculum Development: Using an Industry-Sponsored Project to Teach Manufacturing Simulation in a Multidisplinary Environment,” 2005.

[6] A. Khiste, B. Patrick Hillberg, R. Van Til, and M. Student Solutions Architect Professor, “Developing an IT Infrastructure for Educational Institutions Teaching Product Lifecycle Management,” 2014.

[7] Ulrich Karl and Steven Eppinger, “[6th] Product Design and Development,” 2015.

[8] J. Stark, “Product Lifecycle Management (Volume 2) The Devil is in the Details 3th Edition,” 2020.

[9] D. Litter, F. Leverick, and M. Bruce, “Factors affecting the process of collaborative product development: A study of UK manufacturers of information and communications technology products,” 1995.

[10] M. Bruce, F. Leverick, and D. Littler, “Complexities of collaborative product development,” 1995.

[11] S. Mathrani, A. Mathrani, and C. Liu, “Factors that drive success in collaborative product development,” 2011, pp. 454–461. doi: 10.1109/ITMC.2011.5996012.

[12] J. M. Davis, L. K. Keys, and I. J. Chen, “Nasa TM-2004-212965 Collaborative Engineering for Research and Development,” 2004.

[13] R. Rosario, J. Davis, and L. K. Keys, *Concurrent and collaborative Engineering Implementation in an R&D Organization*. IEEE, 2004.

[14] J. Stark, “Decision Engineering Product Lifecycle Management (Volume 1) 21st Century Paradigm for Product Realization 4th Edition,” 2020.

[15] S. C. Florica and G. Drăghici, “Integrated product development using different collaborative tools in a PLM multisite platform,” in *Applied Mechanics and Materials*, 2013, pp. 867–871.

[16] H. Y. Li, X. Liu, G. Q. Feng, and C. E. Wang, “A web-based PLM system research and implementation in a collaborative product development environment,” in *Proceedings - ICEBE 2005: IEEE International Conference on e-Business Engineering*, 2005, pp. 549–552. doi: 10.1109/ICEBE.2005.20.

[17] U. O. Yago, “PDM para Grados de Ingeniería,” 2011.

[18] “Aras_Innovator_ConfiguringSolutions_StudentGuide_October2024”.

[19] K. Mahalakshmi, B. K. Mahalakshmi, R. Prabhakar, K. Mahalakshmi α, and R. Prabhakar σ, “Performance Evaluation of Non Functional Requirements,” 2013. [Online]. Available: https://www.researchgate.net/publication/303922619

[20] P. S. Pérez, “A ‘PLM’ IMPLEMENTATION IN A DESIGN PROJECT COURSE OF PRODUCT DESIGN ENGINEERING PROGRAM,” 2010.

[21] J. S. Bedolla, J. M. Gomez, and P. Chiabert, “A short portable PLM course,” *IFIP Adv Inf Commun Technol*, vol. 442, pp. 111–120, 2014, doi: 10.1007/978-3-662-45937-9_12.

[22] S. S. Mule and M. Yashwant Waykar, “‘ROLE OF USE CASE DIAGRAM IN S/W DEVELOPMENT.’”

[23] “Aras Innovator 33 Installation Guide,” 2024.

[24] V. R. Basili, G. Caldiera, and H. D. Rombach, “THE GOAL QUESTION METRIC APPROACH.”

[25] J. R. Lewis, “The System Usability Scale: Past, Present, and Future,” *Int J Hum Comput Interact*, vol. 34, no. 7, pp. 577–590, Jul. 2018, doi: 10.1080/10447318.2018.1455307.