STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–18.
Published online in International Academic Press (www.IAPress.org)

Intelligent Decision Making and Knowledge Management System for Agile Project Management in Industry 4.0 context

Asmae ABADI [1,*], Chaimae ABADI [2], Mohammed ABADI [3]

[1] *Euromed University of Fes, UEMF, Morocco*
[2] *ENSAM, Moulay Ismail University, Meknes, Morocco*
[3] *Team Optimization of Production Systems and Energy, Laboratory of Advanced Research in Industrial and Logistic Engineering (LARILE), Hassan II University of Casablanca, Morocco*

Abstract This paper presents a smart, knowledge-driven system designed to optimize Agile Project Management (APM) processes, particularly for Industry 4.0 applications. By formalizing key concepts and integrating Rule Based Reasoning using Semantic Web Rule Language (SWRL) inference rules, the proposed APM ontology offers a robust framework for projects 4.0 knowledge management, interoperability, and decision support. The proposed ontology-based system lies in its capacity to integrate data from external systems, enabling holistic optimization and supporting intelligent decision-making. The system enhances task prioritization, resource allocation, and sprint planning by leveraging reasoning capabilities to streamline project workflows and reduce redundancy. Its implementation in a real-world cobot integration project demonstrated its ability to align tasks with project objectives, optimize resource utilization, and ensure efficient project execution.

Keywords Agile Project Management; Rule Based Reasoning, Information Computing, Knowledge Management; Inference Ontology; Expert system

DOI: 10.19139/2378-SOIC-2378

1. Introduction

The rapid advancements in technology, particularly within the context of Industry 4.0, have created a need for more dynamic, flexible, and efficient approaches to managing complex projects. Traditional project management methodologies often struggle to keep pace with the evolving demands of modern industries, especially those undergoing digital transformation. Agile project management, with its emphasis on iterative development, flexibility, and collaboration, has emerged as a promising solution to address these challenges.

Agile methodologies are designed to manage projects in an environment of uncertainty and change, allowing teams to adapt quickly to shifting requirements and new technological advancements [1, 2]. However, despite their success, Agile practices can still benefit from enhanced decision-making capabilities, improved knowledge management, and more streamlined processes [3]. This is where intelligent decision support systems, leveraging cutting-edge technologies such as ontologies and real-time data analytics, can play a critical role.

This paper proposes a smart decision-making and knowledge management system for Agile project management, specifically designed to support the implementation of Industry 4.0 technologies. The system uses an ontology-based framework to represent key Agile concepts such as roles, events, artefacts, and estimation techniques, providing a structured knowledge base that supports decision-making and optimizes project management processes. By integrating real-time data analytics and decision-support tools, the system aims to improve collaboration,

*Correspondence to: Asmae ABADI (Email: a.abadi@ueuromed.org). Euromed University of Fes, UEMF, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

enhance resource allocation, and ensure alignment with business objectives.
Through a detailed use case in the automotive sector, this paper demonstrates the practical application of the proposed system, highlighting its potential to transform Agile project management practices in Industry 4.0 environments. The findings underscore the importance of intelligent systems in accelerating the adoption of Industry 4.0 technologies, driving efficiency, and fostering innovation in complex manufacturing projects.The rest of the paper is structured as follows: Section 2 outlines the related work on Agile project management and associated knowledge-driven systems; Section 3 describes the methodology and the proposed system; Section 4 presents the use case and results; and Section 5 concludes with a discussion of the findings and future research directions.

2. RELATED WORK

This section outlines the related work. Initially, we provide a brief overview of agile project management practices, with a focus on Scrum and its advantages in our context. Then, we explore the use of ontologies in project management, highlighting their potential to address existing gaps in knowledge management and decision-making support.

2.1. Agile Project Management

Agile project management is defined by its core principles, including adaptability, collaboration, effective communication, iterative progress, and a focus on continuous improvement [1, 2, 3]. This approach empowers teams to operate autonomously and efficiently, promoting close customer engagement and enabling them to respond to evolving requirements with speed and flexibility. Agile teams deliver work incrementally and consistently assess their processes to enhance productivity and effectiveness [4, 5, 6]. Agile methodologies, encompassing a diverse range of frameworks, are designed to address the challenges posed by dynamic and unpredictable environments [7, 8]. These methodologies prioritize human factors, teamwork, and the ability to deliver incremental results frequently, adapting to new requirements even late in the project lifecycle [9].
Over the years, numerous frameworks have emerged to implement the values outlined in the Agile Project Management philosophy [2, 6, 11, 10, 13]. These frameworks share fundamental characteristics such as small, focused teams, fixed-duration iterations, frequent testing cycles, and accelerated delivery schedules [1, 13]. Notable agile methodologies include Scrum, extreme programming (XP), Kanban, lean software development, feature-driven development (FDD), the dynamic systems development method (DSDM), agile modeling, internet-speed development (ISD), and crystal methodologies [1, 6, 11, 10, 13, 14].. While some frameworks address specific phases of a project, others provide end-to-end guidance for managing the entire lifecycle [4].
Agile frameworks, such as Scrum, Extreme Programming (XP), and Lean, have transcended their origins in software development to find applications across various industries, including manufacturing, healthcare, and Industry 4.0. Among the various agile frameworks, Scrum has become the most widely adopted, both in academic research and industrial applications. Initially, extreme programming received significant attention [1, 4], but Scrum's approach to adaptability and incremental delivery gained prominence [4]. However, Annosi et al. [14] highlighted that the time pressure embedded in frameworks like Scrum may hinder learning and innovation by forcing teams to focus on immediate tasks, often at the expense of a broader understanding of the project. This time-related constraint can lead to challenges in balancing short-term demands with long-term strategic goals.
The ability to adapt to continuous change is at the heart of agile methodologies, making the role of knowledge exchange critical in ensuring project success [1]. Agile practices prioritize informal communication over extensive documentation, which enables teams to respond quickly to changing needs. Tacit knowledge, often developed through practice, is central to this adaptability, and it is shared within teams, whether co-located or distributed [9]. Though tacit knowledge is a key driver of innovation, its transfer can be slow and uncertain, requiring substantial interaction among team members to be effectively leveraged [9]. Agile methodologies, particularly Scrum, Kanban, and Lean principles, have been effectively applied beyond software development to improve outcomes in fields such as manufacturing, healthcare, and Industry 4.0. In manufacturing, agile practices enhance production flexibility and

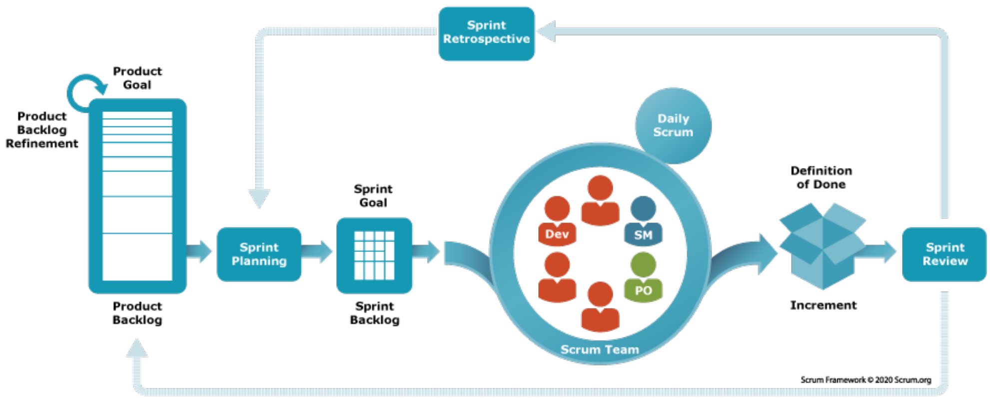

Figure 1. The SCRUM agile project management framework [22].

optimize supply chains, allowing organizations to respond quickly to fluctuations in demand [15]. In healthcare, these frameworks facilitate improved patient care coordination and more efficient resource management. The application of agile principles in Industry 4.0 is equally transformative, supporting the integration and management of smart systems and ensuring that organizations remain adaptable in the face of rapid technological advancements [10].

While agile project management has proven effective in promoting flexibility, collaboration, and adaptability in various industries, several gaps remain in its ability to support knowledge management and facilitate learning from previous projects. Specifically, while agile emphasizes rapid response to change and continuous delivery, it often lacks structured mechanisms for capturing and managing tacit knowledge, which is critical for informed decision-making and long-term project success. Furthermore, the agile process does not inherently support effective knowledge reuse or reflection on past projects, which are key components for improving future outcomes and avoiding repetitive mistakes. In addition, despite the widespread use of agile frameworks, there is a lack of decision-making support systems that integrate with agile practices to provide real-time insights and data-driven guidance during project execution. Our paper aims to address these gaps by proposing a decision support system that enhances knowledge management within agile project management, enabling better decision-making, learning from past projects, and ultimately improving the effectiveness and efficiency of agile methodologies.

2.2. Ontologies for Project Management

The use of ontologies in project management addresses key challenges such as the inconsistency and fragmentation of project-related knowledge. In traditional project management, much of the knowledge is tacit and distributed among individuals, making it difficult to retrieve and reuse effectively [16]. Ontologies help overcome this challenge by providing a structured, explicit knowledge representation that can be easily accessed and reused by team members, stakeholders, and decision-makers [17, 18]. Additionally, ontologies enable reasoning and inference capabilities that can support decision-making, offering insights based on past project data and current project status [16, 19].

In the realm of agile project management, ontologies hold particular promise for addressing issues related to the lack of knowledge management and the challenge of making informed decisions in dynamic, high-pressure environments. Agile practices prioritize flexibility and adaptability, but often at the expense of structured knowledge capture and reuse [16]. This aligns with the work of Kiv et al. [16] who proposed an ontology-based framework using UML to support agile practices adoption, aiming to help practitioners navigate the complexities of agile methodologies by offering structured, reusable knowledge. Moreover, the integration of ontologies with decision support systems (DSS) can further enhance their utility in APM. By incorporating real-time project data and knowledge from previous projects, DSS can leverage ontological structures to provide decision-makers with actionable insights. For instance, in the context of Scrum, an ontology could represent various practices, such as sprint planning, retrospectives, and daily stand-ups, alongside the relationships between team characteristics, project requirements, and external constraints. By reasoning over these ontologies, a DSS can recommend the most suitable practices, highlight potential risks, and identify opportunities for improvement, based on historical data and contextual analysis [20, 21].

The potential for ontologies to support knowledge management in agile project management has been explored in various studies [16] who highlighted the advantages of using ontologies to structure agile practices and improve decision support. However, despite their potential, ontologies in project management are still in the early stages of adoption, with challenges related to the complexity of knowledge representation and the need for domain-specific ontologies tailored to different project contexts. Furthermore, the lack of standardized ontological models for agile project management highlights the need for further research and development to create widely accepted ontologies that can be easily integrated into existing project management frameworks. In addition, only the expressiveness of ontologies was used and not in OWL standardized language, but their reasoning abilities for project management decision support was not at all explored in previous literature.

3. THE PROPOSED AGILE PROJECT MANAGEMENT SMART SYSTEM

In this paper, we propose leveraging the expressiveness and modeling capabilities of ontologies, along with their reasoning abilities, to support knowledge management and decision-making within the Scrum agile management framework. Scrum, as a comprehensive and widely adopted agile methodology, offers a structured yet flexible approach to managing projects, particularly in dynamic and complex environments like Industry 4.0 and manufacturing. Through the smart proposed system (Fig. 2), we aim to enhance communication, collaboration, and the reuse of past insights, ultimately supporting more informed decision-making. Additionally, ontologies enable reasoning over project data, helping Scrum teams adapt to changing requirements and improve overall effectiveness. This approach addresses a critical gap in Scrum's current application, where knowledge management and decision support mechanisms are often underdeveloped. Through this paper, we aim to demonstrate how the integration of ontologies can optimize Scrum's implementation, ensuring more efficient and adaptable project management practices across diverse industries.

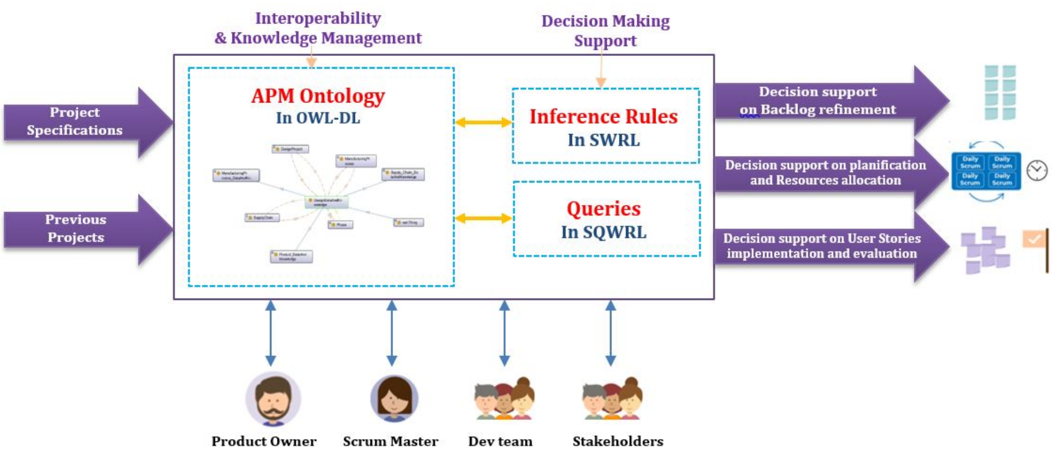

Figure 2. The proposed Smart system for Agile Project Management.

3.1. APM ontology :A proposed reference ontology for Agile Project Management

The core of the proposed smart system is the reference Agile Project Management ontology, denoted APM ontology. The Web Ontology Language (OWL), established as a standard by the World Wide Web Consortium (W3C) [28], has been adopted for the development of APM-Onto to ensure its standardization and broad reusability and interoperability.
The ontology was created using Protégé 5, a widely used, open-source ontology editor for developing OWL-based ontologies and systems [23]. The concepts of the Agile Project Management (APM) ontology are derived from a comprehensive analysis of Agile frameworks, with a primary focus on Scrum as the most widely adopted methodology in Agile practices. Sources include:

- *Agile Manifesto and Principles:* The foundational ideas of Agile, including flexibility, collaboration, and iterative progress, inform the high-level structure of the ontology [1, 25, 24].
- *Scrum Framework:* Concepts such as Sprints, Product Backlog, Daily Stand-ups, Roles (Scrum Master, Product Owner, Team Members), and Artifacts are directly incorporated to represent the practical workflows of Agile projects [3, 26, 27].
- *Industry Standards and Best Practices:* Industry reports and frameworks, provide insights into the operational nuances and challenges faced in real-world Agile implementations.

This approach ensures the ontology encompasses both theoretical underpinnings and practical considerations, making it a comprehensive representation of Agile Project Management. APM ontology is composed of three main elements:

- *Classes:* are a set of individuals that describe concepts in a specific domain. In this paper, the classes are related to the elements in the manufacturing domain;
- *Object properties:* They identify the links between the classes and the individuals;
- *Data properties:* They define modifiers for ontology classes or establish characteristics of the instances.

Fig. 3, Fig. 4 and Fig. 5 represent the main classes, Object Properties and Data Properties of the constructed APM ontology. The Agile Project Management (APM) ontology is structured to represent the key concepts, relationships, and practices of Agile methodologies, particularly SCRUM. The organization of the ontology reflects the hierarchy and interconnections between various elements, enabling a comprehensive semantic framework for Agile practices. The ontology begins with the top-level class, owl:Thing, which represents all entities within the model. The specific classes are organized into a logical hierarchy to reflect the structure and relationships in Agile project management. At the highest level, the classes are grouped into distinct categories, including Projects, Roles, Events, Artifacts, Backlog Components, Metrics, Estimation Techniques, Agile Tools/Practices, and Miscellaneous Concepts :

- The **Project** class, represented by Agile-Project, serves as the root for concepts related to Agile initiatives, including Product Vision and Release.

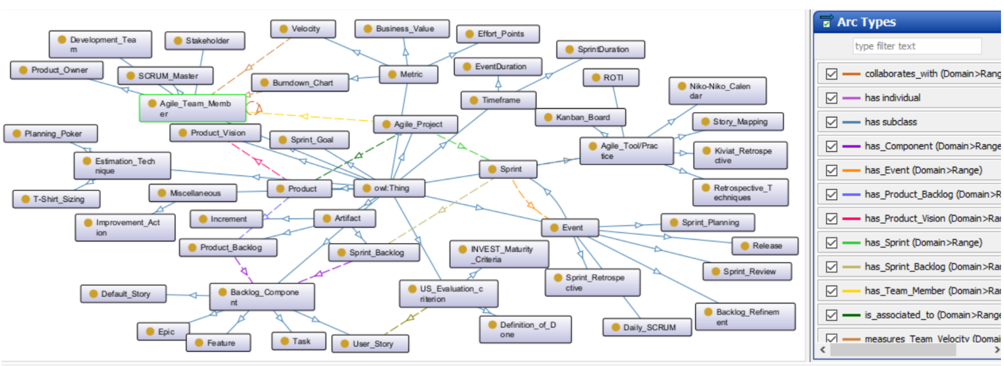

Figure 3. The proposed Agile Project Management Ontology developed in OWL using Protégé 5.

- **Roles** define the participants in the Agile process, such as Product Owner, SCRUM Master, Development Team, and Stakeholder. These roles interact with other classes through responsibilities and ownership of specific artifacts or events.
- **Events** capture the iterative processes in SCRUM, such as Sprint, Daily SCRUM, Sprint Planning, Sprint Review, and Sprint Retrospective. Subclasses like Sprint-Planning and Daily-SCRUM specify the unique characteristics of each event.
- **Artifacts** represent the deliverables and items used to manage Agile workflows, such as the Product Backlog, Sprint Backlog, and Increment. These are further detailed through subclasses like Backlog-Component, which includes User Stories, Tasks, and Default Stories.

The ontology also incorporates Agile-specific tools and practices under the AgileTool/Practice class. Examples include KanbanBoard, Niko-NikoCalendar, KiviatRetrospective, and estimation techniques like PlanningPoker and T-ShirtSizing. These tools are linked to other classes via object properties such as supports (e.g., a KanbanBoard supports task management) and usedIn (e.g., PlanningPoker is used in SprintPlanning). The APM ontology includes also additional classes to capture supporting concepts like Definition of Done (DoD) and INVEST Maturity Criteria for user stories. These guidelines are essential for ensuring quality and readiness. The INVEST criteria (Independent, Negotiable, Valuable, Estimable, Small, and Testable) are a cornerstone of Agile methodology, guiding the creation of high-quality User Stories.

The proposed Agile Project Management (APM) ontology delivers substantial value by leveraging its expressiveness to address knowledge management issues in Agile Project Management implementation. In fact, APM ontology ensures:

- **Comprehensive Knowledge Representation:** Since it captures complex Agile concepts, relationships, and workflows (e.g., roles, artifacts, events) while formalizing abstract elements like Definition of Done and Product Vision.
- **Improved Reusability:** It offers a modular design adaptable to various domains and methodological practices, enabling reuse of concepts such as User Stories and Metrics across projects.
- **Semantic Interoperability:** Ensures consistent communication across tools and platforms using shared vocabularies, supporting data integration and cross-domain collaboration (JIRA, Trello, 3DExperience, . . .).
- **Decision Support and Insights:** Incorporates metrics and reasoning (e.g., Velocity, Effort Points, SWRL rules) to automate insights, prioritize tasks, and monitor progress effectively.

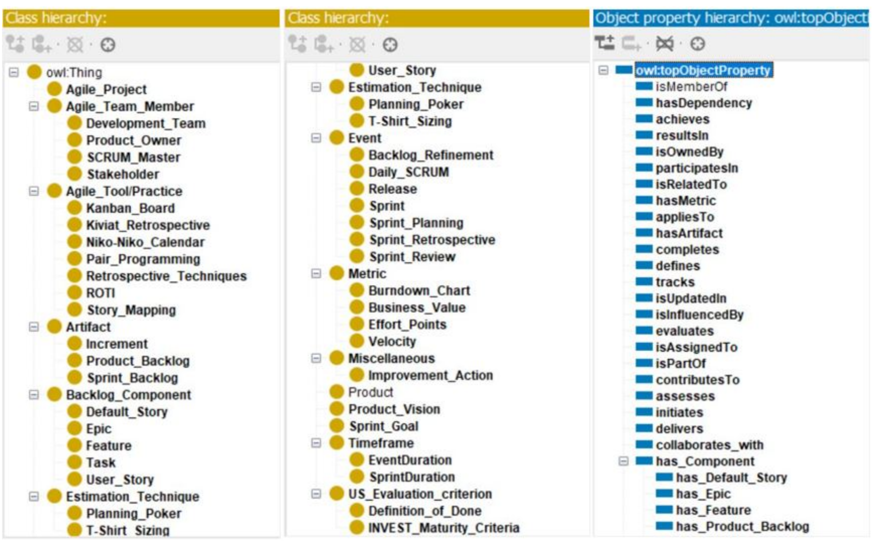

Figure 4. Main classes and object properties of the proposed Agile Project Management Ontology.

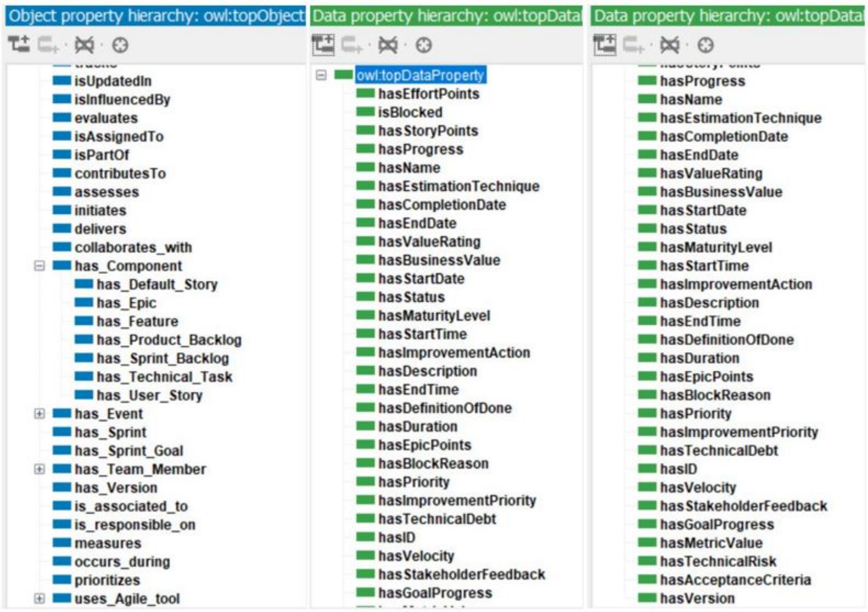

Figure 5. Main data properties of the proposed APM omtology.

- **Standardization and Scalability:** Aligns teams through standardized Agile principles (e.g., INVEST), while its hierarchical structure supports scalability for large projects and frameworks.
- **Knowledge Reusability and Experience Sharing:** Functions as a repository for best practices, retrospective insights, and historical data, promoting continuous improvement and efficient knowledge transfer.

By integrating these elements, the APM ontology serves as a robust tool for capturing project knowledge, fostering collaboration, and enhancing Agile project efficiency.

3.2. The proposed decision making smart system for Agile Project Management

The proposed decision-making smart system for Agile Project Management (APM) represents a significant advancement in integrating knowledge management and automated reasoning within a unified platform. This system harnesses the APM ontology to manage and reason about Agile processes, enabling smarter decision-making through real-time insights, continuous project adjustments, and optimized resource allocation. At the core of this system is the APM ontology, which encapsulates essential Agile concepts, such as roles, events, artifacts, backlog components, and estimation techniques. Through a combination of knowledge representation and inference capabilities enabled by SWRL (Semantic Web Rule Language), the system provides dynamic decision-making support while ensuring the alignment of project goals with ongoing progress. This approach integrates both the knowledge repository and the reasoning mechanisms within the same ontological model, offering a seamless, automated decision-making environment. The integration of SWRL (Semantic Web Rule Language) rules into the Agile Project Management (APM) ontology plays a vital role in enhancing its reasoning capabilities. These rules allow for automated inference, enabling the ontology to evaluate user stories, manage sprints, and optimize team performance based on Agile principles. By embedding these rules, the APM ontology ensures consistency, supports decision-making, and minimizes manual oversight in project workflows. The proposed SWRL rules are categorized into three distinct groups, each targeting a critical aspect of Agile Project Management:

a. Inference rules for Backlog definition and refinement

The first rules category focus on evaluating and refining user stories to ensure they are well-structured, prioritized, and ready for implementation. These rules validate agile criteria enabling a high-quality, actionable backlog.

The INVEST (Independent, Negotiable, Valuable, Estimable, Small, and Testable) principles are essential for high-quality user stories. This category of rules evaluates whether user stories meet these maturity criteria, enabling teams to ensure well-defined and actionable tasks (Table 1):

- Rules from R1 to R6 validate individual INVEST attributes (e.g., independence, business value, or estimability).
- Rule R7 combines these criteria to infer overall compliance with the INVEST framework, marking the user story as ready for implementation.

Then, backlog elements prioritization is key to Agile success, and this category assigns automatically business value to user stories using the MoSCoW framework according to the ranking given by the customer or product owner:

- Must-Have stories are deemed essential, receiving a maximum value of 100 (R8).
- Lower priority stories (e.g., Should-Have, Could-Have, and Won't-Have) are assigned decreasing values (R9 to R11) to ensure effective prioritization and resource allocation.

Table 1. Proposed SWRL Inference Rules for Backlog Definition and Refinement.

Rule ID	**SWRL Rule**	**Description**
R1	User_Story(?us) ˆ not(hasDependency(?us, ?other)) → isIndependent(?us, "true")	Identifies user stories without dependencies.
R2	User_Story(?us) ˆ hasNegotiableAttribute(?us, "true") → isNegotiable(?us, "true")	Marks user stories as negotiable if they have a negotiable attribute.
R3	User_Story(?us) ˆ hasBusinessValue(?us, ?bv) ˆ swrlb:greaterThan(?bv, 0) → isValuable(?us, "true")	Flags user stories with positive business value as valuable.
R4	User_Story(?us) ˆ hasStoryPoints(?us, ?sp) ˆ swrlb:greaterThan(?sp, 0) → isEstimable(?us, "true")	Declares user stories with estimated story points as estimable.
R5	User_Story(?us) ˆ hasStoryPoints(?us, ?sp) ˆ swrlb:lessThanOrEqual(?sp, 13) → isSmall(?us, "true")	Considers user stories with fewer than 13 story points as small.
R6	User_Story(?us) ˆ hasAcceptanceCriteria(?us, ?ac) ˆ swrlb:stringNotEqual(?ac, "") → isTestable(?us, "true")	Labels user stories with defined acceptance criteria as testable.
R7	User_Story(?us) ˆ isIndependent(?us, "true") ˆ isNegotiable(?us, "true") ˆ isValuable(?us, "true") ˆ isEstimable(?us, "true") ˆ isSmall(?us, "true") ˆ isTestable(?us, "true") → isINVEST(?us, "true")	Classifies user stories meeting all INVEST criteria.
R8	User_Story(?us) ˆ hasAcceptanceCriteria(?us, ?ac) ˆ swrlb:stringNotEqual(?ac, "") → isTestable(?us, "true")	Checks for acceptance criteria to mark as testable.
R9	User_Story(?us) ˆ hasPriority(?us, "Must_Have") → hasBusinessValue(?us, 100)	Assigns the highest business value to "Must Have" priority stories.
R10	User_Story(?us) ˆ hasPriority(?us, "Should_Have") → hasBusinessValue(?us, 50)	Assigns medium business value to "Should Have" priority stories.
R11	User_Story(?us) ˆ hasPriority(?us, "Could_Have") → hasBusinessValue(?us, 20)	Assigns low business value to "Could Have" priority stories.
R12	User_Story(?us) ˆ hasPriority(?us, "Won't_Have") → hasBusinessValue(?us, 0)	Assigns zero business value to "Won't Have" priority stories.

b. Inference rules to support planification and resources allocation

Effective sprint planning requires a balance between workload and team velocity. These rules help project managers assess team performance and adjust sprint goals to maintain productivity. This category focuses on (Table 2):

- Real Velocity Calculation (R12): Computing the actual progress of a sprint based on completed tasks.
- Workload Status Evaluation (R14, R15): Determining whether a sprint's workload is manageable (labeled as "Compatible") or excessive ("Overloaded").

In addition, to streamline resource utilization, these rules calculate team capacity and forecast sprint requirements:

Table 2. Proposed SWRL inference rules for project planification and resources allocation

Rule ID	SWRL Rule	Description
R12	Sprint(?s) ˆ User_Story(?us) ˆ hasSprint(?us, ?s) ˆ hasStatus(?us, "done") ˆ hasStoryPoints(?us, ?sp) → swrlb:add(?velocity, ?sp) ˆ hasRealVelocity(?s, ?velocity)	Calculates the actual velocity of a sprint considering the story points of completed user stories.
R13	Sprint(?s) ˆ User_Story(?us) ˆ hasSprint(?us, ?s) ˆ hasStoryPoints(?us, ?sp) ˆ swrlb:add(?totalPoints, ?sp) → hasTotalAssignedPoints(?s, ?totalPoints)	Computes the total story points assigned to a sprint by considering the points of all its user stories.
R14	Sprint(?s) ˆ hasTotalAssignedPoints(?s, ?totalPoints) ˆ hasEstimatedVelocity(?s, ?velocity) ˆ swrlb:greaterThan(?totalPoints, ?velocity) → hasWorkloadStatus(?s, "Overloaded")	Marks a sprint as "Overloaded" if the total assigned points exceed the estimated velocity.
R15	Sprint(?s) ˆ hasTotalAssignedPoints(?s, ?totalPoints) ˆ hasEstimatedVelocity(?s, ?velocity) ˆ swrlb:lessThanOrEqual(?totalPoints, ?velocity) → hasWorkloadStatus(?s, "Compatible")	Marks a sprint as "Compatible" if the total assigned points are within the estimated velocity.
R16	Sprint(?s) ˆ hasDuration(?s, ?duration) ˆ hasNonWorkingDays(?s, ?nonWorkingDays) ˆ hasTeamSize(?s, ?teamSize) ˆ TeamMember(?tm) ˆ isMemberOf(?tm, ?s) ˆ hasVacationDays(?tm, ?vacationDays) ˆ swrlb:multiply(?totalPotentialDays, ?duration, ?teamSize) ˆ swrlb:multiply(?totalNonWorkingDays, ?nonWorkingDays, ?teamSize) ˆ swrlb:add(?totalVacationDays, ?vacationDays) ˆ swrlb:subtract(?workingDays, ?totalPotentialDays, ?totalNonWorkingDays) ˆ swrlb:subtract(?capacity, ?workingDays, ?totalVacationDays) → hasTeamCapacity(?s, ?capacity)	Calculates a sprint's team capacity by accounting for non-working days and team member vacations.
R17	Backlog(?pb) ˆ hasUserStory(?pb, ?us) ˆ hasStoryPoints(?us, ?storyPoints) ˆ swrlb:add(?totalStoryPoints, ?storyPoints) ˆ Sprint(?s1) ˆ hasID(?s1, 1) ˆ hasRealVelocity(?s1, ?realVelocity) ˆ swrlb:divide(?rawSprintCount, ?totalStoryPoints, ?realVelocity) ˆ swrlb:floor(?flooredSprintCount, ?rawSprintCount) ˆ swrlb:subtract(?numberOfSprintsToPlan, ?flooredSprintCount, 1) → hasNumberOfPlannedSprints(?pb, ?numberSprintsToPlan)	Determines the number of planned sprints required to complete the backlog based on velocity and story points.

- Team Capacity Calculation (R16): Accounts for factors such as sprint duration, team size, non-working days, and individual vacations to compute available team effort in day-man units.
- Backlog Sprint Planning (R17): Determines the number of sprints required to complete the project backlog based on real sprint velocity, ensuring realistic project timelines.

c. Inference rules and SQWRL Queries to support User stories implementation and evaluation

This third category of Inference Rules and SQWRL Queries (Semantic Query Web Rule Language) supports informed decision-making by questioning, ranking and querying project elements for optimized User Story implementation and evaluation. Some examples of these queries are (Table 3):

- WSJF Ranking (Q1): Orders User Stories by their Weighted Shortest Job First (WSJF) scores, prioritizing the most valuable and least time-intensive stories for maximum business impact.
- Replanning Query (Q2): Identifies incomplete User Stories from previous sprints, highlighting areas for replanning and enabling iterative improvements in the project's execution.

These rules facilitate better reuse of experience and continuous adjustment of the Agile process. More concrete examples of the rules and queries in this category will be developed in the industrial use case later.
The implementation of these SWRL rules in the APM ontology transforms abstract Agile concepts into actionable, data-driven insights. By formalizing critical evaluation criteria, prioritization heuristics, and resource management processes, these rules ensure that the APM ontology functions as a dynamic, intelligent assistant for Agile teams. This fosters more consistent adherence to Agile principles, reduces bottlenecks, and supports scalable project execution.

Table 3. Proposed SQWRL queries for project planification and analysis

Rule ID	**SQWRL Query**	**Description**
R18	User_Story(?us) ˆ hasBusinessValue(?us, ?bv) ˆ hasEffortPoints(?us, ?ep) ˆ swrlb:divide(?wsjf, ?bv, ?ep) → hasWSJF_Index(?us, ?wsjf)	Calculates the Weighted Shortest Job First (WSJF) index for user stories in function of their business value and effort points.
Q1	User_Story(?us) ˆ hasBusinessValue(?us, ?bv) ˆ hasEffortPoints(?us, ?ep) ˆ hasWSJF_Index(?us, ?wsjf) → sqwrl:select(?us) ˆ sqwrl:orderBy(?wsjf) ˆ sqwrl:select(?bv, ?ep) ˆ sqwrl:columnNames("User Story", "Business Value", "Effort Points")	Selects and orders user stories by WSJF index, displaying their business value and effort points.
Q2	Sprint(?s) ˆ hasEstimatedVelocity(?s, ?estimatedVelocity) ˆ hasRealVelocity(?s, ?realVelocity) ˆ swrlb:subtract(?difference, ?estimatedVelocity, ?realVelocity) ˆ User_Story(?us) ˆ hasSprint(?us, ?s) ˆ hasStatus(?us, ?status) ˆ swrlb:notEqual(?status, "done") → sqwrl:select(?s, ?estimatedVelocity, ?realVelocity, ?difference, ?us) ˆ sqwrl:columnName ("Sprint", "Estimated Velocity", "Real Velocity", "Difference", "US to be replanned")	Identifies incomplete user stories and calculates the velocity difference between estimated and real velocity for all sprints.

4. USE CASE ON INDUSTRY 4.0 PROJECT IMPLEMENTATION

In this section, we will implement the proposed smart ontology-based system for Agile Project Management, in a concrete industrial case study, embedded in the context of Industry 4.0.

4.1. Use case specification

The use case focuses on deploying a collaborative robot (cobot) in an automotive assembly line, aiming to enhance productivity, safety, and operational flexibility (Fig. 6). The cobot will be deployed to handle the precision assembly of critical automotive components and delicate parts (for instance sensors, wiring harnesses and fasteners) onto the vehicles in the main assembly line. This application is selected due to its repetitive nature, demand for high accuracy, and ergonomic challenges for human operators. The use case demonstrates that Agile Project Management is a promising approach for the implementation of industry 4.0 projects due to its ability to manage complexity and uncertainty in dynamic environments. For instance, Scrum's iterative cycles allow for incremental progress and continuous refinement, which is essential when integrating a cobot into an existing assembly line with evolving requirements. Furthermore, the emphasis on collaboration and regular feedback aligns well with the need for frequent adjustments based on real-world testing, safety validation, and performance optimization, ensuring a smooth and adaptive implementation process. Throught four sprints of 3 weeks each (Table 4), different requirements are to be considered in the smart Cobot implementation :

- *Calibration:* The cobot should be calibrated to detect and handle components with high precision and should be integrated with vision systems to locate parts and align them accurately.
- *Task Execution:* The cobot picks, places, and secures components based on predefined assembly specifications. Adjustments are made in real time using input from IoT-enabled sensors and cameras.
- *Quality Control:* After assembly, the cobot uses built-in sensors or external systems to validate the positioning and fastening of components.
- *Operator Collaboration*: In semi-automated tasks, the cobot assists operators by holding parts or tools in place, improving efficiency and reducing physical strain.

4.2. APM Ontology for interoperability and project knowledge management

The APM ontology serves as a comprehensive framework for managing and formalizing all data and information related to the project. All project-related data, including requirements, stakeholders, and deliverables, were introduced into the ontology, ensuring consistency and traceability throughout the implementation process. One of the major advantages of the APM ontology is that manual data entry is minimized due to its design using the OWL (Web Ontology Language) standard. OWL ensures compatibility with XML, enabling seamless integration with external data sources and tools. This interoperability allows the ontology to communicate with a wide range of Industry 4.0 and project management software, reducing redundant effort and ensuring accuracy. For instance, collaborative tools like Jira, used for agile project management, provided data on team members and assigned tasks, which were automatically imported into the ontology. Similarly, CAD models and design specifications from 3DExperience software were directly integrated, supporting the cobot's configuration and workspace layout. The ERP system contributed information on existing suppliers, facilitating vendor selection and procurement processes.

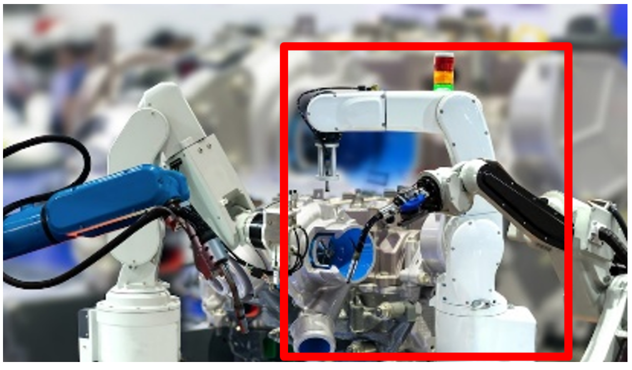

Figure 6. The studied Cobot integrated within the assembly line.

Table 4. Sprint Plan for Cobot Integration in Assembly Line.

Sprint	**Sprint Goal and Objectives**	**Deliverables**
Sprint 1. Requirements Analysis and Supplier Selection	Define system specifications aligned with Industry 4.0 principles and select an optimal cobot solution.	- Detailed technical specification document. - Supplier assessment report with the selected cobot model. - Cost-benefit analysis incorporating Total Cost of Ownership (TCO).
Sprint 2. Preparation for Installation	Prepare the assembly line for cobot integration using Industry 4.0 best practices.	- Assembly line reconfiguration plan incorporating digital twins for layout optimization. - FMEA report and implemented safety measures. - SOPs and operator training modules.
Sprint 3. Installation and Initial Testing	Commission the cobot on-site and validate its operational parameters under real-world conditions.	- Commissioned and fully operational cobot system. - FAT and SAT reports. - Updated task parameters and backlog based on test results.
Sprint 4. Full Deployment and Optimization	Deploy the cobot into full-scale production and establish continuous improvement protocols.	- Fully deployed cobotic system integrated into the assembly line. - Performance metrics report (cycle time, defect rate, OEE). - Retrospective report and roadmap for scaling cobot applications. - Continuous optimization plan based on real-time data.

By importing these datasets into the APM ontology, the project team ensured that all relevant information was readily accessible and standardized. This integration highlights the APM ontology's role as an interoperability enabler, connecting diverse tools and systems into a unified framework. It eliminates silos, supports real-time updates, and enhances collaboration by providing a single source of truth for the project. The ontology's compatibility with widely used standards and tools further emphasizes its utility in Industry 4.0 applications, ensuring that knowledge is effectively captured, shared, and utilized across all project phases.

The APM ontology was first utilized through SQWRL queries to explore insights from previous projects with similar product visions. A set of queries was introduced, and a sample of results is illustrated in Figure 7. For instance, in a prior project (Ref_Project135), the team worked on implementing a robot for the precise installation of door panels on the assembly line AL-23 and so is the case for the other previous similar projects. By querying the ontology, as in Table 5, on all previous product visions including "Robot" or "Cobot" or "robotic arm" or similar keywords, similar projects and their essential information such as suppliers, requirement specifications, CAD models, and other relevant data from these projects was efficiently retrieved in competitive time.

This capability demonstrates the APM ontology's powerful information model, enabling the seamless reuse of knowledge and assets, significantly reducing redundant effort and accelerating decision-making in the current project.

4.3. APM Ontology for agile decision making support

In the cobot implementation project, the APM ontology was enriched with the various inference rules already presented to provide actionable insights and decision support all throughout the agile project management process. The SWRL rules, categorized into three main types, were executed using the Pellet reasoner [28].

These rules covered critical aspects such as requirements specification, backlog definition, sprints planning, team

Table 5. Proposed SQWRL Queries for Project Analysis.

Rule ID	SQWRL Query	Description
Q1.1	Project(?pj) ˆ hasSupplier(?pj, ?sup) ˆ hasCAD(?pj, ?cad) ˆ hasProductVision(?pj, ?pv) ˆ swrlb:contains(?us, ”Robot”) → sqwrl:select(?pj, ?pv, ?sup, ?cad) ˆ sqwrl:columnNames(”Project”, ”Product Vision”, ”Main Supplier”, ”CAD Model”)	Identifies similar robotic projects according to their user stories, and display them with their product vision, supplier, and CAD details.
Q1.2	Project(?pj) ˆ hasSupplier(?pj, ?sup) ˆ hasCAD(?pj, ?cad) ˆ hasProductVision(?pj, ?pv) ˆ swrlb:contains(?us, ”Cobot”) → sqwrl:select(?pj, ?pv, ?sup, ?cad) ˆ sqwrl:columnNames(”Project”, ”Product Vision”, ”Main Supplier”, ”CAD Model”)	Identifies similar Cobotic projects according to their user stories, and display them with their product vision, supplier, and CAD details.
Q1.3	Project(?pj) ˆ hasSupplier(?pj, ?sup) ˆ hasCAD(?pj, ?cad) ˆ hasProductVision(?pj, ?pv) ˆ swrlb:contains(?us, ”Robotic arm”) → sqwrl:select(?pj, ?pv, ?sup, ?cad) ˆ sqwrl:columnNames(”Project”, ”Product Vision”, ”Main Supplier”, ”CAD Model”)	Identifies similar robotic arms implementation projects according to their user stories, and display them with their product vision, supplier, and CAD details.

velocity, resource allocation and prioritization of tasks. For example, rules R8 to R11 related to requirements prioritization according to the customer (Product Owner in Scrum and APM ontology) view automated User Stories Business values determination. Figures 7 and 8 show the corresponding results obtained by Protégé 5 using the reasoner Pellet.

Figures 9 and 10 present other results of SWRL rules and SQWRL queries addressing Sprint Velocity that profigrvided forecasts on team performance based on historical data (user stories done).

The execution of these rules enabled informed decision-making, ensuring alignment with project goals and efficient use of resources throughout the implementation process. This capability underscores the APM ontology's role in enhancing project management through knowledge-driven reasoning and demonstrate its efficiency for various domains of application including industry 4.0 complex projects.

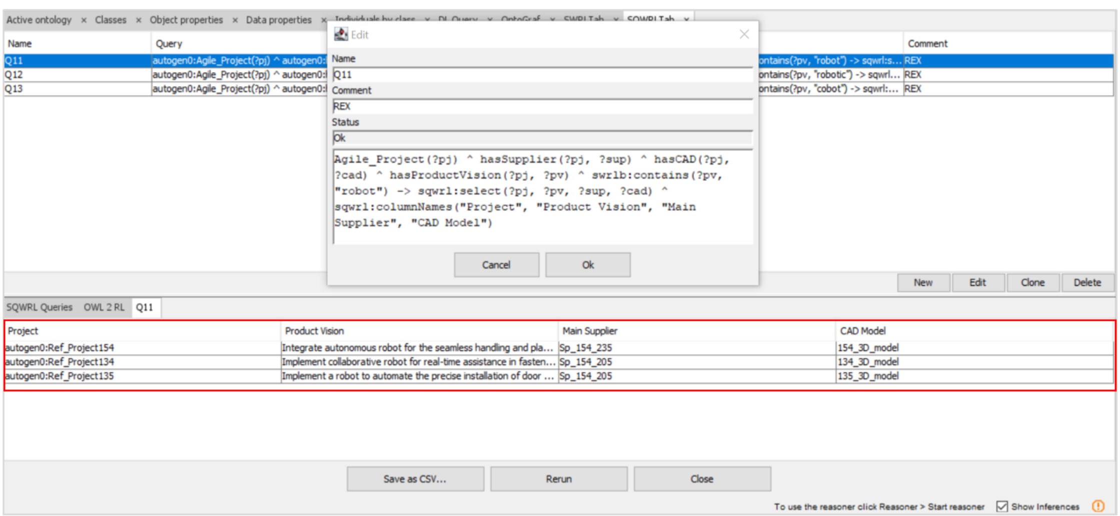

Figure 7. The results of previous projects having similar product vision to be reexploited.

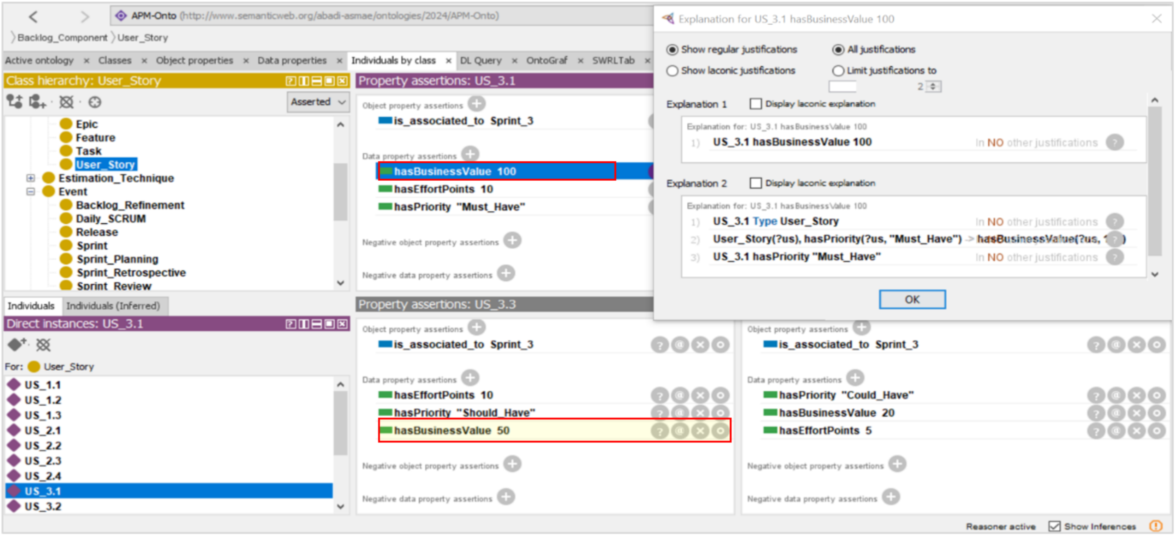

Figure 8. The results of Business Values determination for each user story.

4.4. *The proposed system evaluation*

To evaluate the effectiveness of the knowledge-driven system designed to enhance Agile Project Management (APM) and the developed APM ontology, we conducted a quantitative assessment. While the use case demonstrates the system's capabilities in knowledge management and reasoning, the inclusion of measurable metrics provides a deeper understanding of its impact. The evaluation compared two real-world Agile projects involving cobot (collaborative robot) implementations on highly similar automotive assembly lines: one using conventional Agile practices (pre-ontology project) and the other integrating the knowledge-driven system and APM ontology (post-ontology project). This comparative analysis revealed significant improvements across three main categories of quantitative metrics: Resource Utilization Efficiency, Project Completion Time, and User Satisfaction.

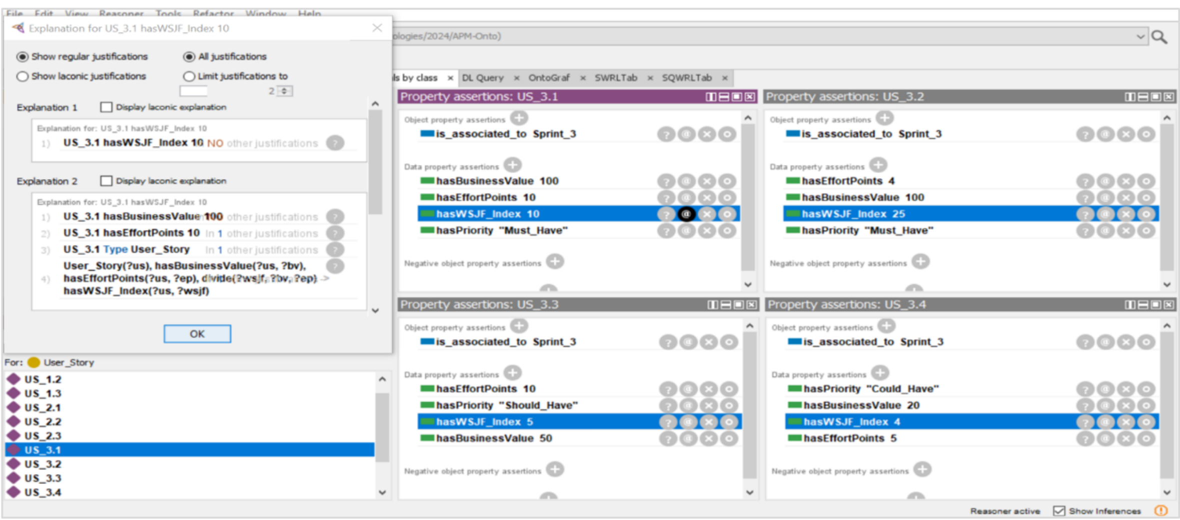

Figure 9. The results of WSJF index determination for each user story for sprint 3.

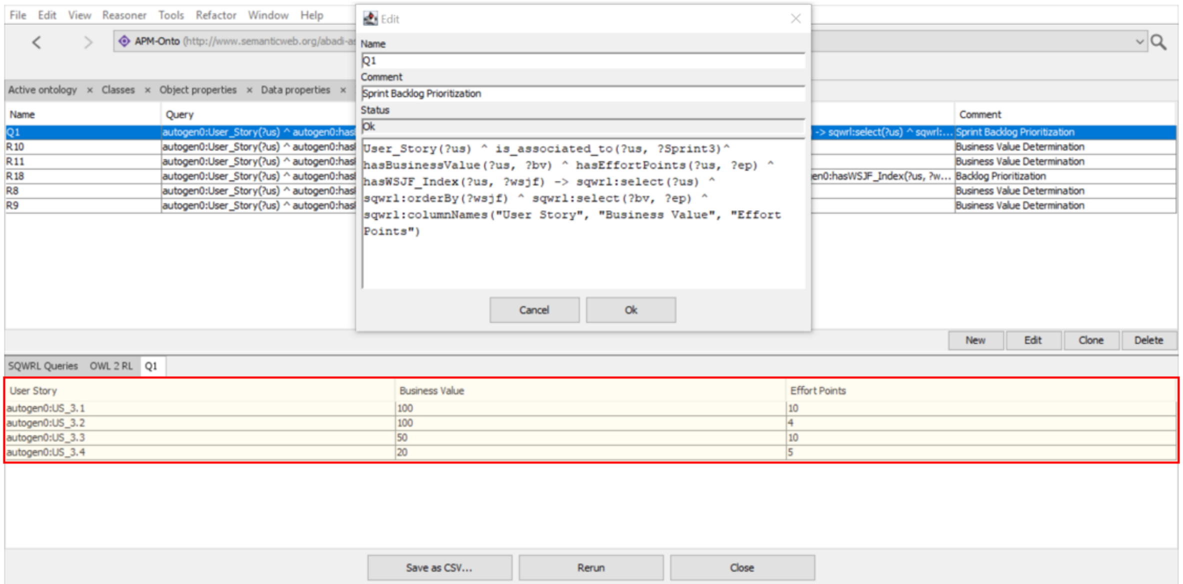

Figure 10. Example of the results obtained on Sprint Backlogs prioritization.

a. Velocity and resource utilization efficiency analysis

Through the comparison between the pre- and post-implementation of the proposed APM ontology, as presented in figure 11, the effective use of resources in the projects was analyzed as a primary focus. In the post-ontology project, team velocity (measuring the number of story points completed per sprint) increased by 22% compared to the pre-ontology project. This improvement can be attributed to the APM ontology's advanced knowledge management capabilities, which facilitated better task prioritization and reduced time spent on resolving ambiguities. Similarly, capacity utilization (the ratio of actual work hours logged to available hours) improved by 19% in the post-ontology project. This increase resulted from the ontology's reasoning engine, which enabled dynamic resource allocation, minimized idle time, and optimized task dependencies.

b. Project Completion metrics analysis

The second category, Project Completion Time, highlights the efficiency of task execution and delivery.

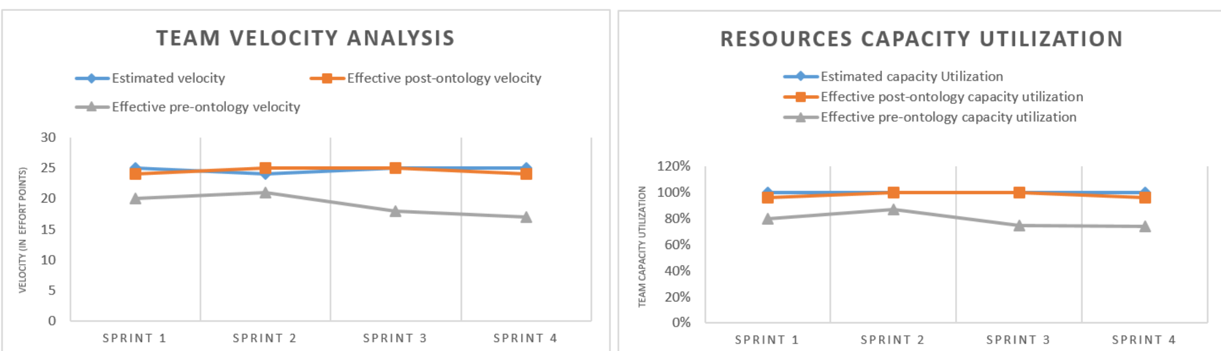

Figure 11. Team Velocity and Resources utilization comparison in pre and post use of the APM ontology.

Through the analysis performed, significant improvements were identified by comparing key metrics such as mean cycle time and lead time between the pre- and post-implementation scenarios, as presented in figure 12. These metrics were evaluated to determine how effectively tasks progressed through their lifecycle, revealing the impact of the proposed APM ontology on reducing delays and improving workflow efficiency.In the post-ontology project, the mean cycle time (the duration a task spends from initiation to completion) was reduced by 8.9% compared to the pre-ontology project. Lead time (which measures the total time from task creation to completion, including both waiting and active phases) decreased by 14.9% in the post-ontology project. These reductions were driven by the system's ability to minimize delays caused by information gaps, ensure smoother task progression, and prioritize high-impact tasks effectively. The reasoning capabilities of the APM ontology also facilitated early identification of bottlenecks and accelerated decision-making.

c. User Satisfaction metrics analysis

The third category, User Satisfaction, assesses the quality and impact of the delivered solution on stakeholders. The Customer Satisfaction Score (CSAT), derived from post-release surveys, increased by 12.2% in the post-ontology project compared to the pre-ontology project as presented in figure 12. This improvement reflects the system's ability to align project outcomes more closely with customer expectations and deliver higher-quality solutions. Additionally, defect density (the number of defects per unit of functionality) was reduced by 55% in the post-ontology project. This significant reduction stems from the ontology's proactive identification of potential quality issues and the facilitation of preventive measures during development through enhanced knowledge sharing and reasoning. The quantitative evaluation comparing the pre-ontology project and the post-ontology project demonstrates the substantial impact of the proposed knowledge-driven smart system and its underlying APM ontology. Improvements in Resource Utilization Efficiency, Project Completion Time, and User Satisfaction validate the system's effectiveness in optimizing workflows, improving resource allocation, and delivering superior project outcomes. These findings highlight the transformative potential of knowledge-driven systems in enhancing efficiency, quality, and satisfaction in Agile project management.

4.5. Discussion of system scalability and performance implications

The proposed system ensures scalability and robust performance through the use of OWL DL (Web Ontology Language Description Logic), a framework that balances expressivity and computational efficiency. OWL DL provides precise modeling of complex relationships and dependencies in Agile project management, while its compatibility with XML-based standards ensures interoperability with widely used project management tools such as Jira, Trello, and MS Project. This interoperability allows seamless integration into diverse project ecosystems,

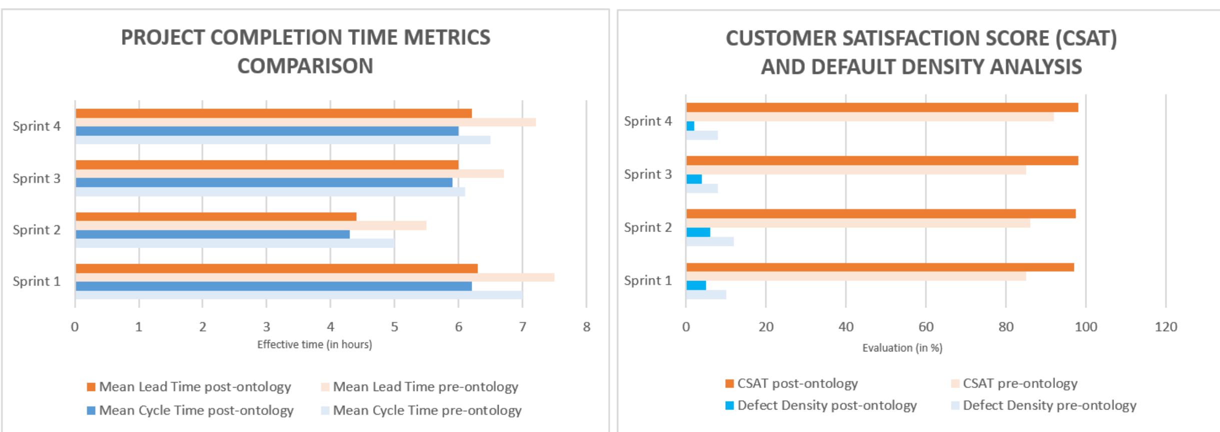

Figure 12. Completion Time, CSAT and Default Density analysis in pre and post use of the APM ontology.

enabling real-time data exchange and decision-making.
Compared to other reasoning engines like Fact++ and HermiT, Pellet was selected for its comprehensive support of OWL DL reasoning tasks, including classification, consistency checking, and SWRL query answering. While Fact++ is known for high-speed reasoning, it is less suited for complex ontology structures that require advanced reasoning capabilities. HermiT, on the other hand, offers optimized reasoning for expressive ontologies but can encounter performance bottlenecks with frequent incremental updates. Pellet provides a balance between these capabilities, supporting incremental reasoning strategies and modular task partitioning, making it well-suited for dynamic Agile environments with rapidly evolving data and dependencies.
The scalability of the system is further enhanced by the extendibility of the APM ontology. Its modular design enables incremental updates, allowing new concepts, rules, and relationships to be integrated without disrupting existing functionality. This ensures that the system remains adaptable to the evolving demands of large-scale projects involving numerous stakeholders and complex workflows.
To highlight the unique contributions of the proposed OWL-based APM ontology, a comparison with existing ontology-based systems and decision-support tools in Agile project management was conducted. The findings underscore the distinct advantages of the proposed approach.
In fact, the proposed ontology is formalized using the standardized OWL language, unlike existing systems that rely on UML-based models [29][30][31] or completely software domain specific [16] as shown in table 6. OWL offers superior expressivity, reasoning capabilities, and interoperability, making it more suitable for complex project scenarios requiring dynamic decision-making and knowledge management.
In addition, the proposed system incorporates advanced reasoning capabilities via the Pellet reasoner. This enables dynamic consistency checking, classification, and decision-making based on Agile workflows. In contrast, existing systems lack reasoning mechanisms, limiting their ability to support dynamic and adaptive project management.
The domain applicability of the proposed ontology extends beyond software development to include industrial projects, aligning with the demands of Industry 4.0. While existing systems are tailored to specific domains, such as requirements engineering [30] or Scrum-based workflows [31], the proposed ontology is versatile and supports both industrial and computer science projects, making it applicable across a broader range of use cases.

Table 6. Comparison of the proposed ontology based system with Existing Systems.

Ref	**Focus**	**Ontology Language**	**Domain Focus**	**Support for Industry 4.0**	**Decision-Making Support**
Wautelet et al. [29]	Unified user story modeling in Agile Projects	UML	Cross-domain	No	No
Supavas et al. [30]	Enhancing requirements engineering in Agile software processes	UML	Software development	No	No
Júnior et al. [31]	Integration of applications for data-driven Scrum processes	UML	Software development	No	No
Kiv et al. [16]	Evidence-based tool to ease Agile practices adoption in software development	OWL	Software development	No	Partial (through SPARQL Queries)
The proposed APM Ontology	Knowledge management and decision-making in the whole Agile Project Management process	OWL	Cross-domain (both industrial and software development)	Yes	Yes (through SWRL inference rules)

Lastly, the proposed ontology effectively integrates knowledge management and decision-making processes, addressing gaps in previous works. For instance, while systems like those by [16] and [29] focus on specific aspects of Agile methods, such as systematic practice adoption or user story modeling, they do not leverage reasoning capabilities to enhance decision-making. The proposed OWL-based APM ontology uniquely combines standardized formalization, reasoning abilities, and cross-domain applicability, setting it apart from existing ontology-based systems in Agile project management.

5. CONCLUSION AND PERSPECTIVES

This paper presented an innovative smart knowledge-driven system designed to enhance Agile Project Management, particularly in the context of Industry 4.0 applications. By formalizing key concepts and integrating reasoning capabilities through SWRL rules, the proposed APM ontology provides a robust framework for project knowledge management, interoperability, and decision support. The proposed system was implemented in a real-world cobot integration project to demonstrate its ability to manage project knowledge and streamline resource allocation, sprint planning, and task prioritization, while leveraging data from external systems to reduce redundancy and ensure alignment with project objectives.

Looking forward, many future research can enhance the proposed ontology by integrating advanced machine learning (ML) techniques and digital twins. ML algorithms, such as supervised learning for delay prediction and reinforcement learning for dynamic resource allocation, can enable adaptive reasoning and predictive analytics, improving decision-making under uncertainty. Also, integrating the ontology with digital twins in Industry 4.0 environments could provide real-time monitoring, anomaly detection, and scenario analysis. This combination would allow continuous adaptation to operational changes, enhancing the management of complex, interconnected systems. These advancements position the ontology as a dynamic tool for intelligent, data-driven project management.

REFERENCES

1. D. Hindarto, *The Management of Projects is Improved Through Enterprise Architecture on Project Management Application Systems*, International Journal Software Engineering and Computer Science (IJSECS), vol. 3, no. 2, pp. 151–161, 2023.
2. J. Highsmith and A. Cockburn, *Agile software development: the business of innovation*, Computer, vol. 34, no. 9, pp. 120–127, 2001.
3. E. G. Chukwurah and S. Aderemi, *Elevating team performance with scrum: insights from successful US technology companies*, Engineering Science and Technology Journal, vol. 5, no. 4, pp. 1357–1371, 2024.
4. M. D. Kadenic, K. Koumaditis, and L. Junker-Jensen, *Mastering scrum with a focus on team maturity and key components of scrum*, Information and Software Technology, vol. 153, p. 107079, 2023.
5. T. Bergmann and W. Karwowski, *Agile project management and project success: A literature review*, in Advances in Human Factors, Business Management and Society: Proceedings of the AHFE, Springer International Publishing, pp. 405–414, 2019.
6. N. Chbaik, A. Khiat, A. Bahnasse, and H. Ouajji, *Analyzing Smart Inventory Management System Performance Over Time with State-Based Markov Model and Reliability Approach, Enhanced by Blockchain Security and Transparency*, Statistics, Optimization and Information Computing, 2024.
7. K. Conboy, *Agility from first principles: Reconstructing the concept of agility in information systems development*, Information Systems Research, vol. 20, no. 3, pp. 329–354, 2009.
8. D. West, T. Grant, M. Gerush, and D. D'Silva, *Agile development: Mainstream adoption has changed agility*, Forrester Research, vol. 2, no. 1, pp. 41, 2010.
9. R. Ouriques, K. Wnuk, T. Gorschek, and R. B. Svensson, *The role of knowledge-based resources in Agile Software Development contexts*, Journal of Systems and Software, vol. 197, p. 111572, 2023.
10. P. Abrahamsson, N. Oza, and M. T. Siponen, *Agile software development methods: a comparative review*, in Agile Software Development: Current Research and Future Directions, T. Dingsøyr, T. Dybå, and N. B. Moe, Eds., Springer Berlin Heidelberg, pp. 31–59, 2010.
11. P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen, *New directions on agile methods: a comparative analysis*, in 25th International Conference on Software Engineering, Proceedings, IEEE, pp. 244–254, 2003.
12. B. Boehm, *Get ready for agile methods, with care*, Computer, vol. 35, no. 1, pp. 64–69, 2002.
13. R. Steegh, K. Van De Voorde, and J. Paauwe, *Understanding how agile teams reach effectiveness: A systematic literature review to take stock and look forward*, Human Resource Management Review, p. 101056, 2024.
14. M. Annosi, M. Magnusson, A. Martini, and F. Appio, *Social conduct, learning and innovation: An abductive study of the dark side of agile software development*, Creativity and Innovation Management, vol. 25, no. 4, pp. 515–535, 2016.

15. S. Alcaraz-Corona, J. L. Cantú-Mata, and F. Torres-Castillo, *Exploratory factor analysis for software development projects in Mexico*, Statistics, Optimization and Information Computing, vol. 7, no. 1, pp. 85–96, 2019.
16. S. Kiv, S. Heng, Y. Wautelet, S. Poelmans, and M. Kolp, *Using an ontology for systematic practice adoption in agile methods: Expert system and practitioners-based validation*, Expert Systems with Applications, vol. 195, p. 116520, 2022.
17. E. Serna and A. Serna, *Ontology for knowledge management in software maintenance*, International Journal of Information Management, vol. 34, no. 5, pp. 704–710, 2014.
18. A. Bouhana, A. Zidi, A. Fekih, H. Chabchoub, and M. Abed, *An ontology-based CBR approach for personalized itinerary search systems for sustainable urban freight transport*, Expert Systems with Applications, vol. 42, no. 7, pp. 3724–3741, 2015.
19. B. Amini, R. Ibrahim, M. S. Othman, and M. A. Nematbakhsh, *A reference ontology for profiling scholar's background knowledge in recommender systems*, Expert Systems with Applications, vol. 42, no. 2, pp. 913–928, 2015.
20. W. Wang, S. De, R. Toenjes, E. Reetz, and K. Moessner, *A comprehensive ontology for knowledge representation in the internet of things*, in International Conference on Trust, Security and Privacy in Computing and Communications, pp. 1793–1798, 2012.
21. D. Yoo and S. No, *Ontology-based economics knowledge sharing system*, Expert Systems with Applications, vol. 41, no. 4, pp. 1331–1341, 2014.
22. Scrum.org, *[Online]*. Available:scrum.org. [Accessed: November 17, 2024].
23. A. Abadi, H. Ben-Azza, and S. Sekkat, *An ontology-based support for knowledge modeling and Decision-Making in Collaborative Product Design*, International Journal of Applied Engineering Research, vol. 12, no. 16, pp. 5739–5759, 2017.
24. H. Dong, N. Dacre, D. Baxter, and S. Ceylan, *What is Agile Project Management? Developing a new definition following a systematic literature review*, Project Management Journal, vol. 87569728241254095, 2024.
25. E. C. Daraojimba, C. N. Nwasike, A. O. Adegbite, C. A. Ezeigweneme, and J. O. Gidiagba, *Comprehensive review of agile methodologies in project management*, Computer Science and IT Research Journal, vol. 5, no. 1, pp. 190–218, 2024.
26. K. Schwaber and J. Sutherland, *Le Guide Scrum*, November 2020.
27. G. Verheyen, *Scrum: a pocket guide*, 2024.
28. A. Abadi, H. Ben-Azza, and S. Sekkat, *Improving integrated product design using SWRL rules expression and ontology-based reasoning*, Procedia Computer Science, vol. 127, pp. 416–425, 2018.
29. Y. Wautelet, S. Heng, M. Kolp, and I. Mirbel, *Unifying and Extending User Story Models*, in Jarke, M., et al., Advanced Information Systems Engineering. CAiSE 2014.
30. S. Sitthithanasakul and N. Choosri, *Using ontology to enhance requirement engineering in agile software process*, Proceedings of the 10th International Conference on Software, Knowledge, Information Management and Applications, IEEE, pp. 181-186, 2016.
31. P. S. S. Júnior, M. P. Barcellos, R. de Almeida Falbo, and J. P. A. Almeida, *From a Scrum reference ontology to the integration of applications for data-driven software development*, Information and Software Technology, vol. 136, p. 106570, 2021.