A conceptual architecture for optimizing IT Project Management using RPA and AI

Andreea-Izabela. Bostan [1,*], Ana Ramona. Bologa [1]

[1]Faculty of Cybernetics, Statistics and Economic Informatics, Bucharest University of Economic Studies, Bucharest, Romania
Email: izabela.bostan@csie.ase.ro (A.I.B); ramona.bologa@ie.ase.ro (A.R.B)
[*]Corresponding author

***Abstract*—The increasing complexity of IT development projects requires new approaches for optimizing project management processes. This paper proposes a conceptual framework for Project Management optimization in IT projects through the integration of Robotic Process Automation and Artificial Intelligence. The framework is structured into three layers: Process Layer, Automation Layer, and Optimization Layer, addressing process design, automated execution, and intelligent decision support. AI-based agents are introduced to analyze project data and generate recommendations for improved planning and resource utilization. A conceptual case study based on an IT project managed using Jira is proposed to illustrate the applicability of the framework. The proposed approach aims to reduce administrative effort, enhance transparency, and support continuous improvement in IT project management.**

***Keywords*—Project Management, Robotic Process Automation, Intelligent Agents, Framework, IT Development projects, Jira**

I. INTRODUCTION

IT project management has become increasingly complex due to rapid technological change, distributed teams, short development cycles, and high expectations regarding quality and position in the competitive market. Traditional project management approaches, while still relevant, often struggle to cope with the volume of operational data and the dynamic nature of modern IT development environments (Kerzner, 2022; PMI, 2021). Empirical studies consistently report high failure rates in IT projects, with common causes including inadequate planning, insufficient monitoring, inefficient resource allocation, and delayed decision-making (Standish Group, 2020).

In response to these challenges, organizations have increasingly adopted digital tools such as Jira, to support project execution. However, the mere adoption of tools does not guarantee process optimization. Manual interaction with these systems, fragmented data sources, and reliance on human interpretation still introduce inefficiencies, delays, and errors (van der Aalst et al., 2023).

In IT project management, Robotic Process Automation (RPA) can streamline administrative workflows, enabling faster implementation and minimizing human errors. This technology focuses on automating repetitive, rule-based tasks in business processes. By deploying software robots to handle repetitive activities such as data entry or report generation, RPA improves accuracy, reduces processing time, and frees human resources for strategic tasks (Aguirre & Rodriguez, 2017).

Artificial Intelligence (AI), in contrast, introduces cognitive automation, allowing systems to learn, predict, and optimize complex IT processes. AI can analyze large datasets to identify patterns, forecast issues, and support data-driven decisions (Russell & Norvig, 2021), while also enhancing project planning, resource allocation, and risk management, improving overall IT performance.

Despite the benefits of both RPA and AI, there are currently no integrated architectures or frameworks that leverage the opportunities of both technologies for IT project management. This gap limits the potential to combine RPA's efficiency with AI's predictive and adaptive capabilities, which could create more advanced automation solutions. As a result, the automation and intelligent optimization of project management processes have emerged as critical research and practice areas. This paper aims to address this research gap as a starting point, by examining the intersection of IT project management, automation technologies, and Artificial Intelligence. Building on these insights, the study proposes a conceptual framework that applies these technologies on Jira (a web-based tool commonly used by IT development teams in project development), while adhering to established project management principles. The framework aims to provide responses to the following research questions:

RQ1: What are the use cases that can be covered by integrating RPA and AI agents within a three-layer framework, in IT project management?

RQ2: To what extent can AI-driven optimization agents contribute to continuous improvement within Jira-managed IT projects.

II. LITERATURE REVIEW

A. *Project Management in IT Projects*

Due to the high degree of complexity and frequent changes in requirements, effective project management is critical for aligning technical development with organizational goals (Schwalbe, 2015; PMI, 2021).

Agile methodology has become a preferred approach for IT implementation due to its flexibility, iterative nature, and responsiveness to change. Unlike traditional waterfall models, which follow a linear and sequential process, Agile emphasizes incremental development through short cycles known as "Sprints". This iterative process allows IT teams to continuously test, refine, and deliver functional components of a system, ensuring that user requirements are consistently met and that errors or misalignments are detected early (Beck et al., 2001). The collaborative framework of Agile, which involves frequent communication between developers, stakeholders, and end-users, fosters transparency and adaptability, enabling organizations to respond promptly to evolving business needs or technological challenges (Highsmith, 2009). Moreover, Agile enhances the success of IT implementations by prioritizing user-centered design and

https://doi.org/10.34630/cip.vi.7663

continuous feedback loops. Through practices such as daily stand-ups, sprint reviews, and retrospectives, teams can gather real-time feedback from stakeholders and incorporate it into subsequent iterations, thereby aligning system functionalities with business objectives (Schwaber & Sutherland, 2020).

IT project management is generally conceptualized as progressing through a series of well-defined phases. The initiation phase establishes the project's objectives, scope, stakeholder engagement, and feasibility analysis (Kerzner, 2022). The planning phase involves decomposition of work into actionable tasks, development of schedules, estimation of durations and resource requirements, risk identification, and the creation of baselines against which performance will be measured (PMI, 2021). The execution phase focuses on the realization of project deliverables through development, testing, and integration activities, while maintaining alignment with the plan (Schwalbe, 2015). The testing phase operates in parallel with execution to track progress and implement corrective actions as needed (Kerzner, 2022). Finally, the deploy and hypercare phases require formal acceptance of deliverables, documentation of lessons learned, and release of team resources (PMI, 2021).

These phases form a lifecycle framework that provide the foundational structure upon which advanced automation and optimization mechanisms (such as Robotic Process Automation and Artificial Intelligence) can be integrated to improve efficiency and decision support in IT project environments. These improvements are needed because, although these phases establish a well-defined lifecycle, in practice many coordination, monitoring, and reporting activities remain manual or semi-manual. In highly dynamic IT environments, this reliance on manual effort often leads to operational overload and a diminished capacity to support timely and effective decision-making. Empirical evidence indicates that the existence of well-defined IT project lifecycle phases is insufficient to ensure project success in contemporary development environments. Studies (Standish Group, 2020) are reporting high rates of project failure and underperformance despite the application of traditional lifecycle models. Approximately 30-35% of projects are completed successfully, while nearly 50% are classified as exceeding planned cost, schedule, or scope, and around 15-20% fail entirely and are canceled before completion. These results highlight the existence of an execution gap between formal lifecycle planning and real-world project delivery, emphasizing the need for complementary mechanisms such as automation, and AI-supported decision-making to manage complexity, uncertainty, and dynamic change in IT projects.

B. RPA and Automation in Project Management

Robotic Process Automation (RPA) refers to the use of software robots that emulate human interactions with digital systems to execute rule-based, repetitive tasks (Aguirre & Rodriguez, 2017). In the context of project management, RPA has been applied to activities such as task creation, status updates, report generation, and data synchronization across systems (Syed et al., 2020). Several studies highlight that RPA is particularly suitable for structured processes with clearly defined inputs, outputs, and decision rules, characteristics commonly found in administrative project management activities (van der Aalst et al., 2018). By standardizing and automating workflow execution within Jira, RPA contributes to greater process consistency and reduces the risk of human error. For example, RPA bots can automatically extract task data from planning documents, populate project management platforms, monitor status changes, and notify stakeholders of predefined events (for example, blocked tasks) (Syed et al., 2020; Lacity & Willcocks, 2021; Bostan, 2024). Jira-based automation significantly reduces administrative work, since automating repetitive project management tasks in Jira (such as issue creation, status updates, notifications, and reporting) can substantially reduce the manual effort required from project managers and team members.

However, despite its benefits, RPA robots only executes predefined workflows and lack the ability to address higher-level optimization challenges such as adaptive resource allocation, predictive risk management, or strategic decision support (Lacity & Willcocks, 2021).

C. AI and Intelligent Agents for Project Control

A study we performed in 2023, by interviewing 105 practitioners involved in RPA implementations, reflects industry perspectives on current limitations and future enhancement directions for RPA technologies (Bostan A.I., Bologa A.R., 2023). The most frequently cited improvement is the integration of multiple Artificial Intelligence components, selected by a clear majority of respondents. This finding suggests that traditional rule-based RPA is viewed as insufficient for handling complex, dynamic business processes and intelligent capabilities are required for tasks such as decision support or contextual understanding.

Artificial Intelligence introduces analytical and cognitive capabilities that complement the execution-focused nature of RPA. AI techniques, such as Large Language Models (LLM) - intelligent agents, are advanced AI systems capable of understanding and generating natural language by learning from large-scale textual data (Brown et al., 2020). Their contextual reasoning abilities make them suitable for decision-support tasks in complex environments, such as interpreting project-related data and generating actionable recommendations in IT project management. Research in AI-supported project management demonstrates significant advancements in predictive and prescriptive analytics, particularly in schedule prediction, effort estimation, risk forecasting, and team performance analysis (Marnewick & Marnewick, 2020; Chou et al., 2022). AI-based models can learn from historical project data to detect underestimation of effort volume, recurring issues, thereby supporting more informed managerial decisions.

Intelligent agents, in particular, are increasingly discussed as autonomous entities capable of perceiving their environment, reasoning over available data, and acting toward defined objectives (Wooldridge, 2020). In IT project environments, such agents can monitor execution data from tools like Jira, analyze sprint performance, and propose corrective actions, such as reallocating resources or adjusting sprint scope. According to recent research, the use of AI for resource optimization can improve team productivity by up

to 15-25% (McKinsey, 2023).

The concept of hyperautomation has gained prominence as a combination of RPA, AI, and advanced analytics into integrated automation architectures (Gartner, 2022). Unlike isolated automation initiatives, hyperautomation emphasizes end-to-end process optimization, continuous feedback loops, and the orchestration of multiple automation technologies. Recent literature advocates for layered architectural models to manage the complexity of such systems. Layered approaches improve modularity and scalability, by separating business logic, automation execution, and optimization intelligence (Syed et al., 2020; van der Aalst et al., 2023). Empirical studies indicate that organizations adopting integrated RPA-AI architectures achieve superior outcomes compared to those implementing standalone automation, particularly in terms of process resilience and decision quality (Lacity & Willcocks, 2021). For example, McKinsey (2023) reports productivity gains of 15-25% in knowledge-intensive work when AI-supported decision systems are applied.

III. MATERIALS AND METHODS

This study adopts a qualitative, design-oriented research approach aimed at the conceptualization of a framework for optimizing IT project management processes through the integration of RPA and AI. Following the principles of Design Science Research (DSR) (Hevner et al., 2004), this study aims to design and formalize a layered architectural framework intended to improve decision support and process optimization in dynamic and complex project environments. The methodology combines process modeling using Business Process Model and Notation (BPMN) with an illustrative case study intended to demonstrate the applicability of the proposed framework in a realistic project management context. The methodology techniques were chosen due to the exploratory nature of the research, which focuses on the design of an architectural framework rather than on empirical testing or quantitative evaluation.

BPMN is used as the process modeling technique for designing and documenting the project management processes addressed by the proposed framework. BPMN was chosen because it offers a standardized notation that enables the clear and unambiguous representation of business processes, it is widely adopted by industry and also compatible with automation technologies. The use of BPMN ensures semantic consistency and improves communication between technical and non-technical stakeholders, while providing a solid foundation for automation and further optimization through AI-based components (Object Management Group, 2014). In this study, BPMN is applied to model project management activities, such as planning, task tracking, and reporting, in a structured and rule-based manner. These models define process flows, events, roles, and decision points, providing a formalized representation of the Process Layer (this concept will be presented in the next chapter). By explicitly specifying process logic and dependencies, BPMN facilitates the identification of automation opportunities and supports the translation of process models into RPA workflows.

To complement the conceptual framework and process models and to demonstrate its feasibility, an illustrative conceptual case scenario is designed around an IT development project managed using Jira, a widely adopted project management and issue-tracking platform. The project context includes typical Jira artifacts such as:

- user stories and development tasks;
- bug tickets and defect reports;
- task statuses and workflow transitions;
- sprint structures and backlog items.

Within this proposed scenario, the BPMN-modeled processes are mapped conceptually to Jira entities and workflows, illustrating how process data could be collected, interpreted, and acted upon by RPA bots and AI agents within the framework. For example, task status changes, ticket creation events, or backlog updates serve as representative data inputs for automation and optimization mechanisms. This mapping enables a demonstration-based validation of the proposed architecture, highlighting the logical consistency between the Process, Automation, and Optimization layers, as well as its practical applicability within a realistic IT project environment. The scenario serves as a proof-of-concept demonstration and supports the internal validation and logical coherence of the proposed model.

IV. RESULT AND DISCUSSION

A. Proposed Framework - Overall Architecture

In the context of increasing complexity in IT projects and sustained pressure for efficiency, the automation of project management processes is becoming a strategic necessity. To this end, the present chapter proposes a conceptual framework structured across **three distinct levels: Process Layer, Automation Layer, and Optimization Layer, which integrates RPA and AI** technologies to support and optimize Project Management activities. This approach enables a clear separation of functional responsibilities, facilitates solution scalability, and provides advanced decision-support capabilities for project managers. This multi-layer structure comes to address the fragmentation identified in the Literature review study, where an important conclusion was the separation of automation efforts from intelligent optimization research.

Figure 1 illustrates the operational flow of the proposed framework, highlighting the interaction between Robotic Process Automation, a LLM-based AI agent, and the human project manager within an IT project management context. The process begins at the level of **the project management platform - Jira**, which serves as the primary source of operational data.

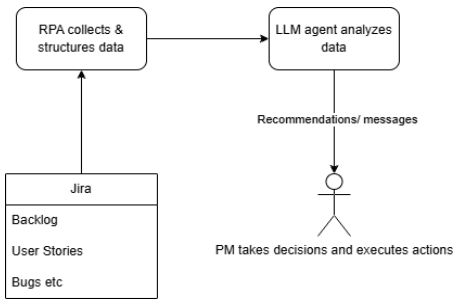


Figure 1. Overall process flow - Integration of RPA and AI to optimize project Management flows in the developments of IT projects. Source: authors' own result

Project artifacts such as backlogs, user stories, bugs, and task metadata represent structured and semi-structured data generated during the execution of IT development activities. An **RPA component is responsible for systematically collecting this data from Jira** and transforming it into a structured, analysis-ready JSON format. Data is retrieved based on a trigger action, the robot being executed on demand, with the opportunity to be scheduled on a time-based event. This step reduces manual data handling, ensures consistency, and enables continuous data acquisition across project phases. Once structured, the **data is transferred to an LLM-based AI agent operating within the Optimization Layer**. The AI agent analyzes the aggregated project data to identify patterns, deviations from planned schedules, potential risks, and inefficiencies related to task execution or resource allocation. Based on this analysis, the agent generates recommendations, insights, or messages expressed in natural language. These outputs may include risk alerts, improvement suggestions, or planning adjustments derived from the project state. The recommendations produced by the LLM agent are communicated to the project manager, who remains central to the decision-making process. In line with a **human-in-the-loop approach**, the project manager evaluates the AI-generated insights and determines the appropriate actions to be executed, such as reallocating resources, reprioritizing tasks, or initiating corrective measures. This design ensures that accountability and strategic judgment remain with human stakeholders while leveraging automation and AI to enhance decision quality.

Figure 2 illustrates the conceptual architecture of the proposed framework for optimizing project management in IT development projects. The framework is structured into three interrelated layers: **Process Layer**, **Automation Layer**, and **Optimization Layer**, each addressing a distinct level of functionality.

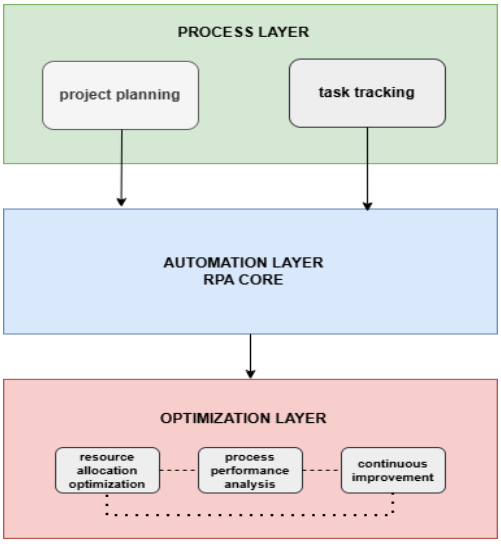


Figure 2. Structure of the three-layer framework for optimizing IT Project Management with RPA and AI agents. Source: authors' own result

At the top of the architecture, the **Process Layer** defines the core project management processes that govern IT project execution. In this framework, emphasis is placed on two fundamental processes: **project planning and task tracking**. Project planning integrates activities such as defining objectives, structuring work, estimating durations, and allocating resources, while task tracking focuses on monitoring execution progress, managing task statuses, and identifying deviations from the plan. These processes are designed to be standardized and explicitly defined, ensuring that they are suitable for automation and consistent execution across projects. The **Automation Layer**, positioned beneath the Process Layer, acts as the operational core of the framework and is implemented using Robotic Process Automation. This layer is responsible for **executing the processes defined above by automating repetitive, rule-based activities** such as: task creation, status updates, data synchronization, and reporting. By translating standardized process definitions into executable workflows, the Automation Layer ensures alignment between process design and execution while reducing manual effort and operational errors. At the lowest level, the **Optimization Layer** introduces advanced analytical and decision-support capabilities through the use of AI-based agents. This layer **leverages the data generated and collected by the Automation Layer to perform process performance analysis, resource allocation optimization, and continuous improvement**. Process performance analysis focuses on identifying inefficiencies and recurring deviations, while resource allocation optimization aims to improve the distribution of human and technical resources based on workload, competencies, and project constraints. The continuous improvement component enables learning across projects by incorporating historical insights into future planning and execution cycles.

The directional flow between layers highlights a clear separation of concerns: **processes are first defined, then executed automatically, and finally analyzed and optimized**. At the same time, the dotted feedback connections within the Optimization Layer emphasize the iterative and adaptive nature of the framework, supporting ongoing refinement of both processes and automation rules.

The three-layer architecture extends prior research that has addressed automation and intelligent decision support separately (Syed et al., 2020; Marnewick & Marnewick, 2020) and clearly delineates process formalization, operational automation, and intelligent optimization within the proposed architectural model.

B. Process Layer - Project planning

The planning process constitutes the foundation of the entire IT project lifecycle. It includes the definition of objectives, the structuring of activities, the estimation of durations, and the allocation of resources. Within the proposed framework, planning is treated as a standardized process, based on templates and clearly defined rules, in order to facilitate automation. The outputs of the planning process (the list of activities, deadlines, and assigning of resources) represent essential input data for the Automation Layer, where RPA robots can retrieve and translate this information into digital project management tools.

A distinctive aspect of the framework is the **design of the planning process** to ensure compatibility with automation. The data generated during planning is structured

in a format that enables the later automated execution of processes. For instance, at the next stage, the RPA robot will:

- retrieve the project plan from a structured file;
- automatically create tasks in Jira;
- set dependencies and deadlines;
- notify the team regarding assigned activities.

This integration reduces human error and ensures consistency between planning and execution, issues frequently identified as challenges in IT project management.

A core element of planning is the **decomposition of the project** into manageable activities, achieved through the use of a Work Breakdown Structure (WBS). Within the context of the proposed framework, the WBS is used as a standard when creating the Jira tickets, so that it can be easily interpreted by automation components. Tasks are defined according to predefined templates, including elements such as: activity description, estimated duration, responsible person from the team, dependencies, and priorities. This standardized structure is not only beneficial for the use of Jira, but also facilitates the automated ingestion of data by RPA systems.

Another essential element is the **estimation of durations and resource requirements**, a complex process characterized by a high degree of uncertainty, particularly in IT projects. Traditional estimation methods, such as analogous and parametric estimation, are frequently used but depend heavily on the project manager's experience (PMI, 2021). Within the proposed framework, estimates are documented in a structured format that enables: reuse of estimates from previous projects, analysis of deviations, integration with AI agents in the Optimization Layer. For example, historical data regarding task durations can later be analyzed by AI agents to identify patterns of overestimation or underestimation, thereby contributing to improved future planning. Below, in figure 3, a structured JSON representation of a planning output is provided, to illustrate how tasks can be represented in a machine-readable format, compatible both with automation and intelligent analysis. This will enable the Automation and Optimization layers to provide automated issue creation, real-time monitoring and predictive data analysis:

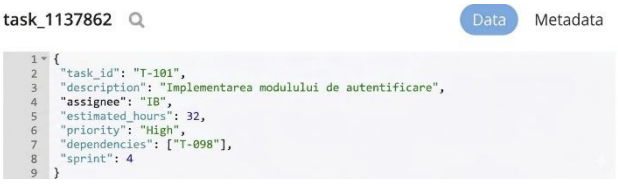


Figure 3. Example of structured representation of a project task. Source: authors' own result

Although the Process Layer does not include Artificial Intelligence components, the quality of planning directly influences the effectiveness of AI agents in the Optimization Layer. Well-structured data and planning history enable AI agents to perform predictive analyses and provide relevant recommendations regarding resource allocation and plan adjustments. Consequently, planning is not treated as a static activity, but as a dynamic process that feeds a continuous improvement cycle.

C. Process Layer - Task Tracking

Activity tracking plays a critical role in monitoring project progress against the initial plan. This process involves the periodic collection of information regarding task status, the identification of delays, and the highlighting of critical dependencies. Within the Process Layer, tracking is defined as a continuous process, based on the constant updating of data originating from multiple sources (Project Management platforms such as Jira, used in this case). In terms of tools, the **Jira platform is used for a clear structuring of activities** that contributes to more accurate duration estimates and to the reduction of execution risks.

An effective task tracking system contributes to increased transparency, reduced operational risks, and improved decision-making processes. Within the proposed framework, activity tracking is designed as a standardized process, oriented toward automated data collection and prepared for integration with RPA technologies and AI agents. It represents a fundamental process in IT project management, aimed at monitoring execution progress and identifying deviations from the initial plan. In IT projects, activities are often interdependent and distributed across multidisciplinary teams, making manual monitoring difficult and prone to error. The literature indicates that inadequate task tracking is one of the primary causes of delays and budget overruns in software projects (Standish Group, 2020).

In this context, Jira task tracking serves to:

- provide real-time visibility into project progress;
- identify issues and critical dependencies;
- support coordination among team members.

To enable automation of the tracking process, the proposed framework mandates the use of a **standardized set of task statuses**, aligned with established practices in both Agile and traditional project management. Examples of such statuses include: "not started", "in progress", "blocked", "testing", "completed". Status standardization is essential as it reduces ambiguity in progress reporting, enables automated data interpretation by RPA bots, and facilitates subsequent analysis performed by AI agents. The use of clear and consistent activity state definitions contributes to improved project control and increased reporting accuracy. A central technical element of the task tracking process is the collection of progress data from multiple sources. Within the framework, this process is designed to be performed automatically through RPA, thereby reducing reliance on manual updates.

The Process Layer is responsible for defining the formal Tracking process, and its operational implementation and execution is based on RPA automation mechanisms described in the next section.

D. Automation layer

The Automation Layer represents the operational level of the proposed framework, with the purpose of automatically executing the processes defined at the Process Layer, without direct human intervention. This layer is responsible for the orchestration, execution, and monitoring of software robots (RPA bots) that implement repetitive and rule-based activities within IT project management.

Within the context of this study, the Automation Layer functions as **an interface between business processes and the AI-based Optimization Layer**, ensuring the collection of operational data required for advanced analytics and continuous improvement mechanisms. RPA is employed as the primary mechanism for interacting with the project management platform, Jira. The RPA robot is configured to perform actions such as:

- automatic creation and updating of tasks in Jira;
- retrieve structured project data, including: backlog items, user stories, bugs, assignees, task statuses, priorities, and sprint metadata;
- collection of activity status information;
- collection of time-related attributes, such as: estimated effort, story points, and actual time logged, are extracted to enable comparisons between planned and realized execution;
- generation and distribution of periodic reports;
- data synchronization across systems.

Through the Automation Layer, the framework contributes to a significant reduction in the time allocated to administrative activities, enabling project managers to focus on strategic and decision-making tasks.

The Automation Layer of the proposed framework is implemented using **UiPath** as the Robotic Process Automation platform, selected for its native support for enterprise systems, API integration, orchestration capabilities, and extensibility toward AI services. This layer is responsible for automated data extraction, preprocessing, and orchestration of project management information originating from Jira. The UiPath implementation is designed to implement the process in four steps: **user authentication, data extraction, data validation, handover to the Optimization Layer**/ user notification. All four stages are being orchestrated centrally using **UiPath Orchestrator**, enabling controlled execution, logging, and scalability. For the first step, authentication is implemented using **token-based authentication mechanisms** supported by Jira's REST API, avoiding the use of hardcoded credentials within automation workflows. **UiPath ensures secure credential management** through UiPath Orchestrator Assets, where sensitive information such as API tokens, client secrets, and authentication headers is stored in encrypted form and accessed at runtime only by authorized robots. HTTP Request activity is configured with secure HTTPS communication and appropriate authorization headers. Access scopes are restricted to read-only or least-privilege permissions, ensuring that robots can retrieve project data (tickets, time logs, assignees, and statuses) without modifying Jira artifacts (at this point, at least). For the data extraction step, **the RPA robot is responsible of retrieving data such as: issue types (user stories, bugs, tasks), issue metadata (status, priority, assignee, sprint), time-related fields (original estimates, remaining estimates, time spent), workflow transitions and timestamps**. In terms of UiPath activities implemented, the robot is using "for each" activities for iterating through issue collections, combined with "assign", "if", "switch" activities for conditional logic. **Once Jira data is retrieved, UiPath workflows perform automated comparisons** between:

- Estimated effort versus actual time logged;
- Planned sprint scope versus completed tasks;
- Assigned resources versus workload distribution.

Figure 4 below illustrates an example for the BPMN flow corresponding to the automation of the task tracking process, highlighting the sequence of steps executed by the RPA robot, the decision points, and the interaction with the project manager. In the context of IT project planning, BPMN facilitates the alignment between conceptual process design and automation by providing executable-ready models that can be directly mapped to RPA workflows and system integrations. In this example, the process is automatically triggered by a scheduler-type event and involves collecting data from the project management system - Jira, analyzing deviations from the plan, and notifying the project manager when delays or blockers are identified. Human interaction is limited to the decision-making stage, while the remaining activities are fully automated.

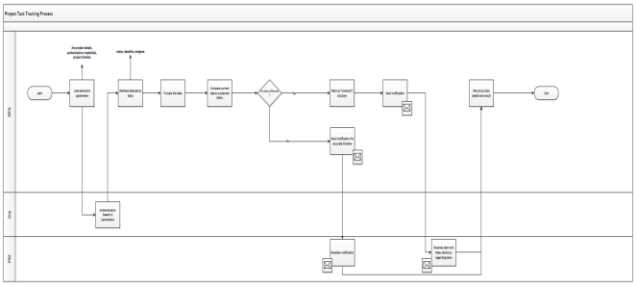


Figure 4. BPMN diagram - Automated project task tracking process. Source: authors' own result

The automated workflow described in Figure 4 includes the following actions:

- **Flow initialization:** The RPA robot is automatically triggered based on a scheduler-type event. It loads the execution parameters, such as the list of active projects and the analyzed time interval;
- **Authentication and access to the Project Management platform:** The RPA robot authenticates into Jira by means of secure credentials. After successful authentication, the robot accesses the list of tasks associated with the analyzed project;
- **Extraction of activity-related data:** For each identified task, the robot extracts the following information: task ID, current status, start date and

due date, assigned resource, and last update timestamp. The data are temporarily stored in a tabular structure to facilitate subsequent processing;

- **Identification of deviations from the plan:** The robot compares the extracted data with the initial project plan and identifies deviations such as overdue tasks, tasks marked as "Blocked", and tasks that have remained inactive for a period exceeding a predefined timeframe. This analysis is performed based on predefined business rules, consistent with standard RPA automation logic;
- **Data repository update:** The processed information is stored in a centralized structured data repository, ensuring a consolidated and up-to-date view of project progress;
- **Notification and escalation:** When critical deviations are identified, the RPA robot automatically generates notifications, specifically it sends emails to the project manager;
- **Execution finalization and logging:** At the end of execution, the robot records a detailed execution log of all performed actions, which can later be used for audit purposes or performance analysis.

The final output of the Automation Layer consists of **structured datasets** prepared for AI-based analysis. Also, the UiPath robot implements continuous monitoring processes that detect predefined conditions, that trigger automated actions using "trigger" or "Send Outlook Mail Message" activities, in order to highlight: overdue tasks, excessive variance between estimated and actual effort, high bug density in active sprints. This data is normalized and consolidated into a structured repository, allowing consistent analysis across sprints and projects. By automating data collection and validation, the Automation Layer minimizes manual reporting effort and ensures near real-time visibility into project execution.

E. Optimization layer - the level based on the AI agents intervention

The Optimization Layer constitutes the advanced level of the proposed framework and introduces AI agents with roles focused on analysis, prediction, and continuous improvement of project management processes.

Data coming from UiPath is then transformed into a JSON format, which serves as input for the Optimization Layer, where an AI-based agent is implemented using a LLM agent, accessed through an API-based integration. The LLM-based agent operates as a decision-support component rather than an autonomous controller. By invoking the LLM agent directly from UiPath, the **structured data and a carefully designed prompt** are sent to the model, which analyzes the sprint performance and generates actionable recommendations regarding resource allocation, process optimization, and automation opportunities. Once the data is received, the LLM performs a semantic and analytical interpretation of the project state, correlating historical trends with current execution data. Based on this analysis, the AI agent generates recommendations that are presented to the project manager, supporting a **human-in-the-loop governance model** who retains authority over whether and how adjustments are applied across the workflow.

Unlike rule-based automation, the LLM agent is capable of synthesizing multiple signals and generating explanatory, human-readable recommendations rather than deterministic outputs. For example, if the LLM detects a sustained increase in the number of blocked tasks or bugs associated with a specific epic during a sprint, it may generate recommendations such as:

- Temporarily allocating an additional testing resource to the affected sprint;
- Introducing testing tasks earlier in the development lifecycle;
- Adjusting sprint scope to reduce overload on developers;
- Replacing or augmenting human effort with additional RPA workflows (automated notifications or reporting tasks), thereby improving efficiency and reducing operational overhead. By updating the RPA workflow, a controlled feedback loop is enabled between the Optimization Layer and the Automation Layer.

The prompt designed for the LLM agent is dynamically constructed to include project information. The process of finding the most optimal prompt is iteratively refined using a sequence of prompts, thereby enhancing performance and ensuring the optimal outcomes. For example, for Resource Allocation Optimization, the RPA robot generates, as data entry, the following information: team components (for each member: the role, number of tasks completed, and average completion time), as well as the total number of tasks delayed. The LLM agent receives these indicators at a Sprint level and performs an analysis based on historical data. If the agent observes a pattern showing that testing tasks are consistently completed with delays, it may recommend the temporary allocation of an additional tester to the sprint. Implementing such a recommendation would lead to a significant reduction in delays and improved adherence to project deadlines.

Another technique used in order to maximize the AI capability is the **"Chain-of-Thoughts"** prompting technique, that encourages the model to generate intermediate reasoning steps before producing a final answer. In other terms, the LLM agent is encouraged to explain its reasoning, instead of directly providing a conclusion, which improves performance on such a task that requires problem solving and decision support. At the same time, the whole framework is based on a "**human-in-the-loop**" model, this way every decision suggested by the LLM agent is subject to managerial approval prior to implementation. This way, the framework supports efficient resource utilization and reduces the risk of both overloading and underutilization.

That being said, unlike RPA, which executes predefined tasks, AI agents are capable of interpreting project-generated data and providing intelligent

recommendations. Within the context of the proposed framework, AI agents operate on data collected by the RPA robot and aim to optimize key project dimensions, such as:

- adherence to deadlines;
- efficient resource utilization;
- reduction of operational risks;
- improvement of decision quality.

The proposed framework aligns with the concept of "hyperautomation", described in the Literature review section, as the integration between the two layers is achieved through bidirectional feedback:

- the Automation Layer (RPA robot) collects operational data;
- the Optimization Layer (AI agent) analyzes this data. Unlike the Automation Layer, which executes predefined processes, the Optimization Layer evaluates data generated during process execution and continuously optimizes project management performance.

V. Conclusion

The reviewed literature clearly demonstrates the individual benefits of RPA and AI in project management. However, existing studies often address these technologies in isolation or focus on narrow use cases. The three-level framework proposed demonstrates a clear separation of responsibilities: after project planning and tracking (Process layer), RPA handles data extraction and operational automation (Automation layer), the LLM agent (Optimization layer) provides analytical and decision-support capabilities, and the project manager retains control over execution. This interaction exemplifies how Intelligent Automation can augment, rather than replace, human project management in IT development environments. To answer the first research question, *what are the use cases that can be covered by integrating RPA and AI agents within a three-layer framework, in IT project management*, the framework proposed:

- standardizes project management processes to enable automation;
- leverages RPA for reliable operational execution;
- applies AI agents for continuous optimization;
- and supports a feedback loop between execution and decision-making.

Continuous improvement is a fundamental principle in IT processes and project management. A key element of the Optimization Layer is its ability to learn from previous projects. Within the proposed framework, continuous improvement is realized at the Optimization Layer through AI agents capable of analyzing the performance of automated processes and recommending optimizations to RPA workflows. The process ends in this state of "recommendation" and not "implementation" of suggested improvements, which can be an answer to research question number two: *To what extent can AI-driven optimization agents contribute to continuous improvement within Jira-managed IT projects?*. Once the AI-driven optimization is implemented, its operation is intentionally stopped at the point of decision-making, ensuring that all final choices are made by a human agent. This approach embodies both the "human-in-the-loop" and "chain-of-thought" principles, which are essential to apply rather than leaving decisions solely to the AI. By maintaining human oversight, the framework preserves accountability, contextual judgment, and strategic control, while still benefiting from AI's analytical and reasoning capabilities.

Regarding limitations and future work opportunities, this case study is based on a single tool-specific context (Jira), which may limit the generalizability of the proposed framework to other project management environments or organizational settings. At the same time, the effectiveness of the AI-based optimization agents depends on the consistency and quality of operational data (in this case mostly generated by Jira, and the prompt provided), so as a result, this proposed solution can be considered to be at an early stage. On a different note, this is more than a starting point for future research, with the aim to focus on implementing the framework in organizational settings and conducting studies to measure its impact on project performance and different metrics, such as: cycle time reduction, effort savings, team allocation (since resource allocation optimization represents a core functionality of the Optimization Layer). Also, the study can be extended across multiple other project management platforms (for example Azure DevOps) or could adapt other instruments/ technologies for automation or AI (for example Python instead of RPA, BluePrism instead of UiPath, Machine Learning for AI agents and so on). To enhance the methodological credibility of the framework and generate relevant outputs, the prompt designed for the LLM agent can be iteratively refined, through successive adjustments.

The framework has been designed to be tested on multiple Jira projects (web or mobile implementations), in order to observe results of the AI agent optimization on different environments and iterations. The introduction of such agents has the potential to overcome the limitations of deterministic automation specific to RPA, as they are capable of analysis, learning, and recommendation. This framework aligns with contemporary hyperautomation principles and addresses both operational efficiency and strategic project governance, providing a solid foundation for future empirical validation and practical implementation.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Both authors conducted the research, analyzed the data, wrote the paper, and approved the final version.

References

1. Aguirre, S., & Rodriguez, A. (2017). Automation of a business process using robotic process automation (RPA): A case study. *Communications in Computer and Information Science, 742*, 65-71.
2. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. Thomas, D. (2001). *Manifesto for agile software development*. Agile Alliance.
3. Bostan, A. (2024). Optimizing IT Project Management: Leveraging Jira Automation for IT Implementations. *COMMUNICATIONS.*

4. Bostan, A. I., & Bologa, A. R. (2023). Insights and proposals for RPA implementations. *Database Systems Journal*, *14*(1).
5. Brown, T. (2020). Language models are few-shot learners. Advances in Neural Information.
6. Chou, J. S., Nguyen, N. M., & Hsu, S. C. (2022). AI-based prediction models for project schedule and cost overruns: A systematic review. *Automation in Construction*, *134*, 104047.
7. Gartner. (2022). Hyperautomation: A framework for business-driven automation. Gartner Research.
8. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105.
9. Highsmith, J. (2009). *Agile project management: Creating innovative products* (2nd ed.). Addison-Wesley Professional.
10. Kerzner, H. (2022). Project management: A systems approach to planning, scheduling, and controlling (13th ed.). Wiley.
11. Lacity, M., & Willcocks, L. (2021). Robotic process automation and risk mitigation: The definitive guide. SB Publishing.
12. Marnewick, C., & Marnewick, A. L. (2020). Artificial intelligence for project management: The state of the art. *International Journal of Information Systems and Project Management*, *8*(4), 5-22.
13. McKinsey & Company. (2023). The economic potential of generative AI: The next productivity frontier. McKinsey Global Institute.
14. Object Management Group. (2014). Business process model and notation (BPMN), version 2.0.2. OMG.
15. Project Management Institute. (2021). A guide to the project management body of knowledge (PMBOK® Guide) (7th ed.). PMI.
16. Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
17. Schwalbe, K. (2015). *Information technology project management* (8th ed.). Cengage Learning.
18. Schwaber, K., & Sutherland, J. (2020). The Scrum guide: The definitive guide to Scrum: The rules of the game. Scrum.org.
19. Standish Group. (2020). *CHAOS report 2020: Beyond infinity*. Standish Group International.
20. Syed, R., Bandara, W., French, E., Stewart, G., Maynard, S. B., & Munro, J. (2020). Robotic process automation: Contemporary themes and challenges. *Computers in Industry*, *115*, 103162.
21. van der Aalst, W. M. P., Bichler, M., & Heinzl, A. (2018). Robotic process automation. *Business & Information Systems Engineering*, *60*(4), 269-272.
22. van der Aalst, W. M. P., Carmona, J., Chatain, T., & Taymouri, F. (2023). Process mining: A 360 overview. *ACM Computing Surveys*, *55*(5), 1-34.
23. Wooldridge, M. (2020). *An introduction to multiagent systems* (2nd ed.). Wiley.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).