UNIVERSITY
INSTITUTE
OF LISBON

Application of Knowledge Graphs in Project Management: An Enhance of the PMBOK 6 Framework

Margarida Beatriz da Conceição Correia

Master in Integrated Business Intelligence Systems

Supervisor:
PhD Luísa Cristina da Graça Pardal Domingues Miranda,
Assistant Professor,

Iscte – Instituto Universitário de Lisboa

October, 2025

TECHNOLOGY
AND ARCHITECTURE

Department of Information Science and Technology

Application of Knowledge Graphs in Project Management: An Enhance of the PMBOK 6 Framework

Margarida Beatriz da Conceição Correia

Master in Integrated Business Intelligence Systems

Supervisor:
PhD Luísa Cristina da Graça Pardal Domingues Miranda,
Assistant Professor,

Iscte – Instituto Universitário de Lisboa

October, 2025

Acknowledgements

I would like to express my appreciation to my family for their constant support, patience, and encouragement throughout this journey.

To my advisor, Professor Luísa Domingues, I thank you for all your guidance and availability and encouragement throughout this journey.

Finally, I would also like to thank my pugs, who kept me company and brought joy and calm to the most stressful moments.

To all that I enumerate my sincere "Thank you".

Resumo

O PMBOK é um guia reconhecido e utilizado por gestores de projetos durante a execução dos mesmos. Este guia possui várias edições, sendo a mais recente a 7ª edição. Embora não seja a versão mais atual, esta dissertação foca-se no PMBOK 6 devido à sua estrutura organizada, a qual define 49 processos com inputs, tools e outputs, todos interligados entre si.

O objetivo desta dissertação é auxiliar os gestores de projeto na interpretação e compreensão do PMBOK 6 e nas conexões existentes entre os seus 49 processos. Para isso, foi desenvolvido um Grafo de Conhecimentos. Este Grafo de Conhecimentos foi testado através de casos de uso que demonstram como uma única alteração num processo pode ter impacto e propagar-se por diferentes áreas.

Palavras-chaves: Gestão de Projetos; PMBOK 6; Grafos de Conhecimento; Neo4j

Abstract

PMBOK is a well-known standard used by project managers during project execution. This guide has several editions, with the most recent being the 7th edition. Although it is not the latest version, this dissertation focuses on PMBOK 6 due to its structured approach, which defines 49 processes with inputs, tools, and outputs forming a large interconnected network.

The goal of this dissertation is to help project managers have a clearer understanding of PMBOK 6 and the connections between its 49 processes. To achieve that, a visual tool was developed, a Knowledge Graph. This Knowledge Graph was tested through use cases that demonstrate how a single change in one process can impact and propagate across different areas.

Keywords: Project Management; PMBOK 6; Knowledge Graphs; Neo4j

Index

Tables Index

Figures Index

List of abbreviations

PM – Project Management

PMBOK - Project Management Body of Knowledge

PRINCE2 - Projects IN Controlled Environments

PG - Process Groups

KA – Knowledge Areas

KG – Knowledge Graph

SLR - Systematic Literature Review

DSR - Design Science Research

Chapter 1 – Introduction

1.1. Topic context

Project Management (PM) is a well-established and increasingly recognized subject within industry due to the essential role it plays in supporting the delivery of strategic initiatives [1]. It provides structure, organization and a systematic approach for managing projects of different types, scales and levels of complexity. Over the years, several standards and methodologies have emerged, such as PMBOK (Project Management Body of Knowledge), PRINCE2, Agile, Scrum and Kanban, helping organizations to select the practices that best fit their needs and project characteristics [1].

Among these standards, PMBOK is one of the most widely adopted, being recognized for its comprehensive guidelines covering the project lifecycle and the processes required to deliver value to the organization and its stakeholders [2]. This thesis focuses specifically on the **PMBOK Guide Sixth Edition** [3], where PM is structured into 49 processes, organized across five Process Groups (PG) and ten Knowledge Areas (KA). Despite its detailed guidance, PMBOK presents a significant challenge, its processes are highly interdependent, and managing the large volume of Inputs, Outputs and Tools can be demanding in terms of information interpretation and decision-making.

In practice, project managers often struggle to navigate these relationships efficiently. The complexity and scale of the processes, combined with insufficient training or limited experience, can make it difficult to understand the cascading effects that changes in one process may generate in others [4]. This reinforces the importance of developing mechanisms that improve visibility over these interconnections and support informed decision-making.

To address these difficulties, this thesis explores the application of Knowledge Graphs (KGs) as a way of representing PMBOK 6 in a more accessible and connected format. KGs are capable of structuring information through entities and relationships, making implicit dependencies explicit and facilitating knowledge discovery [5]. By mapping PMBOK processes into a KG, the main objective of this thesis is to provide project managers with a more intuitive visualization of the PMBOK, potentially improving decision-making and knowledge transfer in project environments [6]. Although PMBOK 7 is already available, this thesis focuses on the Sixth Edition, as it offers a more process-based and structured view of PM, which is essential for its representation in a KG.

1.2. Motivation and topic relevance

PM plays a vital role in ensuring organizational success. According to [7], the main goal of PM is to minimize time and resource wastage while maximizing process efficiency and value delivery. Achieving this requires effective planning, coordination and risk management throughout the entire project life cycle, supported by well-defined standards that guide best practices [8].

The PMBOK Guide Sixth Edition [3] remains one of the most widely adopted frameworks, but project managers still face difficulties when dealing with the scale and complexity of its structure. Interpreting the dependencies between processes and understanding how a change in one area may affect several others can become a demanding task, especially in large or dynamic projects [8]. This highlights the need for more advanced and intuitive ways to explore and connect the information defined in PMBOK 6.

One of the biggest challenges lies in navigating the interdependencies between project elements. The traditional format in which PMBOK is presented does not easily support the exploration of relationships between Inputs, Outputs and decision elements, making it harder to maintain a clear global view of the project [9]. Therefore, tools that allow a more accessible and integrated representation of these dependencies become increasingly relevant. Providing project managers with a clearer understanding of these relationships helps improve planning accuracy, supports risk anticipation and strengthens overall governance of the project.

Although KGs have already been successfully applied in domains such as healthcare, finance and semantic search, their application in PM is still limited and mostly unexplored. This gap highlights the relevance of investigating how KGs can be used to represent complex process-based standards such as PMBOK 6 [10].

KGs appear as a strong candidate to address this issue. They have proven ability to model complex relationships in a structured and visual manner [4], enabling users to understand how entities are connected and how information flows between them. By mapping PMBOK 6 into a KG, it becomes easier to observe how processes relate, identify hidden dependencies and strengthen decision-making [10].

Therefore, this thesis investigates how the integration of KGs into PM can improve the visualization of interdependencies and provide better support for timely and informed decisions. The proposed solution intends to contribute to a more connected and accessible representation of the PMBOK 6 framework, helping project managers deal with complexity more effectively and ultimately improving project performance [11].

1.3. Questions and research goals

The main goal of this thesis is to demonstrate how KG can improve PM by providing a more structured and connected way of handling the information defined in PMBOK 6.

To achieve this goal, three specific objectives were defined:

1. Analyze the structure of PMBOK 6 and determine how its processes and interdependencies can be represented in a Knowledge Graph.
    - Identify key processes , their relationships and dependencies.
    - Define how these elements can be modelled as graph entities
2. Develop and implement a KG that maps PMBOK 6 processes and their connections in Neo4j.
    - Develop a data model aligned with PMBOK 6.
    - Insert nodes and relationships representing the relevant elements.
    - Ensures the graph supports efficient querying and visualization.
3. Evaluate the usefulness of the KG through practical use case scenarios.
    - Analyze how changes in one process propagate to others.
    - Observe the benefits regarding visibility of interdependencies and support for decision-making.

Based on this objectives, the guiding research question of this work is: **How can a Knowledge Graph improve the understanding of dependencies in PMBOK 6 and support decision-making in project management?**

1.4. Methodologic approach

This research followed a Design Science Research (DSR) methodology, supported by a Systematic Literature Review (SLR). The SLR, presented in Chapter 2, ensured a structured and rigorous approach to identifying, selecting and synthesizing the most relevant literature to support the research motivation, define the problem and clarify existing gaps[12]. Meanwhile, DSR guided the development of the KG as a practical solution designed to address a real-world need in PM practice., aligned with the model described by Vaishnavi and Kuechler [13].

The combination of these two methodologies was considered appropriate due to their complementary contributions:

- The SLR enables a structuring understanding of the state of the art, helping to identify existing gaps and opportunities for improvement [12]
- DSR approach supports the development and evaluation of practical solutions to address organizational challenges, which made it suitable for designing and validating the KG developed in this work [13]

In this thesis, the SLR provided the theoretical foundations and justified the need for a visual and interconnected representation of PMBOK 6. Based on these insights, the DSR methodology structured the practical development phases of the KG, from conceptual modelling to implementation and evaluation through use case scenarios.

This combined approach ensured that the results achieved were aligned both with the academic context, by building on established knowledge and with professional needs, by offering a tangible contribution to PM practice.

1.5. Structure and organization of dissertation

This section provides a brief overview of the structure of this dissertation. The work is organized into six chapters, each addressing a different stage of the project:

Chapter 1 – Introduction: Presents the theme of the dissertation, the motivation behind the work, the research questions and the methodology adopted.

Chapter 2 – Literature Review: Explores the concepts related to PM, the structure of PMBOK 6 and the relevance of KG for representing and accessing complex information.

Chapter 3 – Research Methodology: Details the literature review process and the adopted research framework.

Chapter 4 – Knowledge Graph Implementation: Details the modelling process and the step-by-step construction of the KG in Neo4j.

Chapter 5 – Use Cases: Demonstrates how the KG can support PM by analyzing different change scenarios and observing their impacts across other processes.

Chapter 6 – Conclusions and Future Work : Summarizes the main findings, discusses the limitations of the study and proposes directions for future work.

Chapter 2 – Literature review

2.1. Project Management Practices

PM is a globally recognized concept presented in various forms across industries. For this work, we define PM as the application of knowledge, skills, tools, and techniques that meet the project requirements, helping organizations complete their projects effectively and efficiently [7]. To ensure successful outcomes when using this concept, there are several key characteristics that a PM method should present, such as well-structured models, clarity in detail levels, standardized techniques and reporting formats, adaptability in application, capacity for rapid evolution, and accessibility for both clients and internal stakeholders [7].

The PM approach is widely applied due to its many advantages over other management methods, particularly its adaptability to projects of any complexity, budget, size, or business type **[7]**. The two main standards in PM are the traditional approach and the agile approach, each extensively used and characterized by distinct features [7].

On one hand, according to [6], the Agile methodology offers a modern, dynamic approach in PM, particularly well-suited for handling complex innovation and technology projects, such as those in software development. It ensures that projects are completed on schedule with high-quality deliverables. Its goal is to deliver value continuously through each stage setting apart from traditional project management models [6].

On the other hand, traditional management provides a structure for managing projects, offering adaptable practices that can meet specific project needs. One widely recognized framework is the PMBOK, first introduced in 1996 and continuously updated to reflect industry's best practices. The 6th edition organizes PM into ten KA structured across five PG. The 7th edition, however, introduced a shift towards a more principle-driven approach, making both versions relevant for different contexts. Despite this evolution, the structure of PMBOK 6 makes it particularly effective for analyzing process interdependencies, which is why it is the focus of this work. The PMBOK is not seen as a guide with a rigid set of practices due to its collection of processes and practices that can be adapted to different project environments, whether predictive or adaptive [14]. This flexibility is seen by organizations as an incredible advantage, by adapting their PM strategies according to their specific needs.

2.2. PMBOK 6: Structure and Limitations

2.2.1 Structure of PMBOK 6

The PMBOK can be seen as a standard that consolidates the best practices, and procedures widely recognized in project management. Its processes and knowledge areas are designed to achieve successful project outcomes [15]. This guide, first introduced in 1996, has been updated periodically, with the latest being the seventh version presenting itself as a complement of the sixth version [16].

Effectively the PMBOK 6th Edition organizes its 49 processes within ten KA**(**such as Scope, Time, Cost, Quality, and Risk Management**)**, structured across five process groups: Initiating, Planning, Executing, Monitoring and Controlling, and Closing [17] [18]: Each process sits at the intersection of a PG and a KA and follows a defined structure: it consumes specific inputs, applies tools and techniques, and generates outputs.

These processes are interdependent and iterative. For example, Develop Project Charter (Initiating/Integration) formalizes the project's existence using inputs like contracts and business documents, and produces the Project Charter, which then guides planning activities. Similarly, Create WBS (Planning/Scope) structures project deliverables into manageable components, creating outputs that feed into scheduling and cost estimation.

As the project progresses, earlier outputs become inputs for execution and control. Execution processes rely on defined plans to coordinate people and resources, while Monitoring and Controlling processes assess performance, often triggering adjustments through integrated change control. Finally, the Closing group ensures formal project completion, producing closure documentation and lessons learned.

2.2.2 Challenges and Limitations of PMBOK 6

The PMBOK6 is widely used due to the numerous advantages it offers in PM practices. However, PMs may face specific challenges when trying to apply this guide in projects. Due to its detailed structure, and the complexity involved in adapting its processes to various project environments.

While PMBOK 6 is highly valued for its structured approach, the same structure can present several disadvantages. According to [19], [20] and [21] some of the challenges are:

- **Information Overload**: PMBOK6 contains 49 processes organized into ten knowledge areas and five process groups. This large amount of information can be overwhelming for PMs.
- **Difficulty in Customizing Processes**: Although PMBOK 6 provides guidelines for adapting its processes, PMs often struggle to determine the appropriate customization required for the projects which can lead to an over-complication of processes.
- **Navigation Complexity**: The vast documentation and the connection between different processes make it difficult for users to navigate and retrieve the specific information they need.
- **Limited Practical Guidance**: While PMBOK 6 offers a framework for what should be done, it often lacks detailed practical guidance on how to implement certain processes in specific project scenarios.

2.3. Knowledge Graphs

2.3.1 Knowledge Graph Definition

Accordingly, with [22], a KG is a visualization tool defined as graphs of data. This graph is composed of nodes and edges, where the nodes represent the entities of interest (a real object or abstract concept), and the edges represent the relations between the entities. In *figure 1*, we can see an example of a KG where in this KG, (e1, r1, e2) is a triplet that indicates e1 and e2 are connected by relation r1 [22].

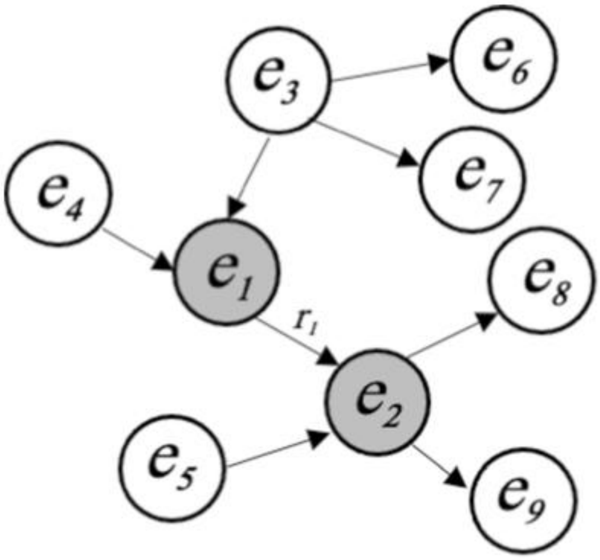

Figure 1 - KG Example

KGs have been applied across various domains to enhance data integration, information retrieval, and decision support. We can see the example of a KG in the cybersecurity field, for instance, they are utilized to model and analyse complex threat intelligence data, aiding in the detection and prevention of cyber threats. They enable the integration of diverse cybersecurity information, facilitating advanced analytics and improved situational awareness [23].

2.3.2 Benefits and Challenges of Knowledge Graphs

KGs have gained significant attention in recent years due to their ability to integrate and structure large volumes of data, enhancing decision-making processes in various domains. Despite these advantages, constructing and maintaining KGs presents several challenges, as we can see inside this subchapter.

On one hand, a major challenge consists of ensuring data quality. Since KGs rely on accurate relationships between entities, incomplete or inconsistent data can significantly impact the reliability of the graph. In healthcare, for example, a study [24] showed that incomplete data led to incorrect diagnoses, a risk with serious consequences. Another notable challenge is scalability, integrating multiple data sources increases complexity and can slow system performance, as seen in a financial fraud detection case [24]. Maintaining and updating a KG is also resource-intensive, as manual updates are prone to error and delay.

Still, the advantages of KGs are considerable. A key benefit is their ability to integrate diverse data sets, facilitating deeper connections and insights. For instance, a logistics company used a KG to merge data from suppliers, warehouses, and transport systems, which reduced operational costs [25]. KGs also offer flexibility in schema evolution, allowing for new relationships or data types without reconfiguring the entire structure.

2.4 PMBOK 6 and Knowledge Graphs

2.4.1 Integration of PMBOK 6 and Knowledge Graphs

While PMBOK 6 provides a structured and comprehensive framework for project management, its complexity can present challenges in terms of navigating, reusing information, and understanding the interconnections between its components. The KGs is presented as a helpful tool by offering an interactive visualization of structured information [26], [27].

Effectively, this data visualization method was already applied for the PMBOK 7 [28] where the main concepts and relationships were identified and extracted, resulting in a model with 599 components and 1,346 connections. This graph visually represented interactions between 12 project management principles, 8 domain components, 22 models, 60 methods, and 76 artifacts. While the visualization improved comprehension, the article lacked clear conclusions on the tangible benefits or limitations of the graph.

Nevertheless, several studies emphasize the potential of integrating KGs into project management practices. Paulheim [27] discusses how KGs support the refinement and formalization of complex domains. Zhao et al. [29] demonstrate improvements in project oversight and decision-making when combining PM practices with semantic technologies. Similarly, Smith and Jones [30] illustrate how KGs and ontologies can be used to tailor PMBOK guidance to specific organizational contexts. Moreover, visual interfaces powered by KGs allow for simplified navigation of complex frameworks [31], while their scalability supports continuous adaptation to dynamic environments [32].

2.4.2 Advantages of Integrating the PMBOK 6 and Knowledge Graphs

KGs are recognized as a powerful tool for organizing and understanding complex data and their relations across various domains, including PM [28], [27]. A few studies highlight how KGs can enhance data management, analysis, and risk assessment by structuring information.

When integrated with PMBOK 6 practices, KGs have demonstrated potential to overcome several challenges faced by project managers. For example:

1. **Enhanced Information Retrieval**: A KG can help PM to quickly locate relevant processes, tools, and techniques by linking related concepts within PMBOK [29], [30].

2. **Improved Customization Support**: By mapping relationships between processes, a KG can offer customized recommendations for applying PMBOK principles in different scenarios [30], [33].

3. **Simplified Navigation**: Instead of navigating through lengthy documents, project managers can use a KG's interface to explore interconnected concepts and retrieve necessary information [27], [31].

4. **Scalability and Adaptability**: KGs can evolve with changing PM practices, allowing continuous integration of new methodologies [27], [32].

Furthermore, as PM continues to evolve over time, the flexibility and scalability of KGs allow for continuous updates, ensuring that the knowledge captured remains relevant [29]. Consequently, the use of KGs can lead to more informed decision-making, and a more efficient application of PMBOK 6's best practices [33].

Chapter 3 – Research Methodology

This research adopts a dual methodological approach combining Design Science Research (DSR) with a Systematic Literature Review (SLR) to address the research question and objectives define in chapter 1. This combination was considered appropriate because DSR supports the creation and evaluation of practical solutions, while the SLR ensures that their design is based on structured and reliable evidence [34].

The study is structured around the DSR methodology, which supports both the identification of practical problems and the development of innovative solutions. In this case, the approach guided the development of a Knowledge Graph to represent and explore PMBOK 6 and its interdependencies.

3.1. Design Science Research Approach

The DSR approach proposed by Hevner et al. (2004) offers a structured, iterative process for developing and evaluating solutions that address real-world problems [34]. This approach was selected because it aligns with the purpose of this thesis, to design, implement, and evaluate a KG to address challenges in understanding PMBOK 6's interdependencies.

The DSR process adopted in this study followed the main phases proposed by **Hevner et al. (2004),** each phase, guided a distinct stage of the research, from identifying the problem to validating the solution. Below we in Figure 2 we can see a representation of the DSR diagram, based on [37]:

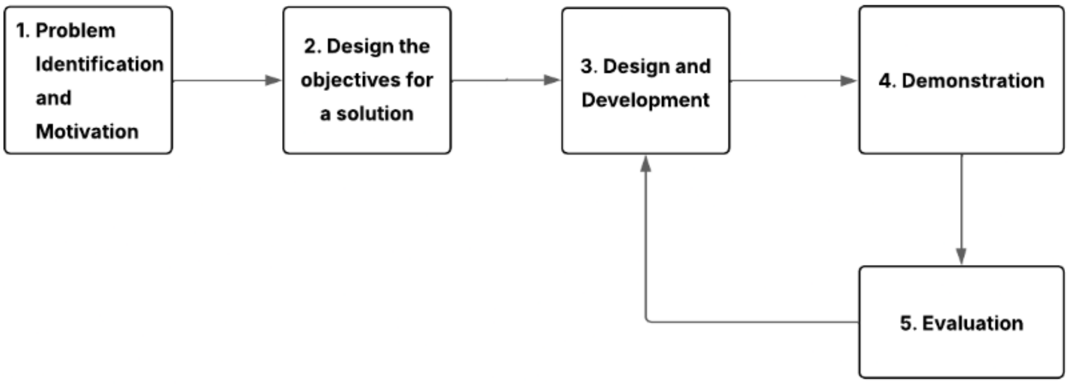

Figure 2 - DSR diagram

3.2. Problem Identification and Motivation

This phase is focused on identifying the research problem and understanding its relevance to PM practice. In this case, the main challenge was identified as the difficulty in interpreting the extensive and interconnected information structure on PMBOK 6, more precisely the 49 process and all of the information and connections associated to them. Although PMBOK provides a complete and descriptive view of PM processes, its structure makes it very difficult to visualize relationships or to understand how changes in one process may affect others.

The goal here was to define the research problem and propose a possible solution that could make these connections more explicit . As a result, a development of a KG was identified as a potential solution to represent and analyze PMBOK's complex network of relationships in a more visual and interactive way, helping project managers to better understand dependencies and process interactions.

3.3. Design the objectives for a solution

Following the identification of the problem, the next phase focus on defining what the proposed solution should accomplish. The goal was to design a KG capable of improving the visualization, retrieval, and interpretation of information within PMBOK 6. By doing so, it would become easier to explore the relationships between processes and understand how they influence each other across different KA and PG.

The main goals were to analyze PMBOK 6 processes, identify and map their dependencies and relationships, and structure them as interconnected entities within a graph-based model. This representation pretends to provide a clearer and more intuitive view of PMBOK 6's structure, allowing project managers to better understand how information flows between processes.

3.4.Literature Review

The SLR was an essential component of the DSR methodology, mainly supporting the Problem Identification and Objective Definition phases. The review goal was to establish a theoretical basis for the study, identify research gaps, and validate the need for a structured, graph-based approach to represent PMBOK 6.

The SLR focused on three main research domains: project management challenges, limitations in PMBOK implementation, and applications of KG in decision-support contexts. With that information the following steps were followed:

The databases Google Scholar, Scopus, and ResearchGate were used, applying the search strings:

- “Project Management” AND (“PMBOK” OR “Agile”)
- “PMBOK 6” AND (“limitations” OR “challenges”) AND “Project Management”
- “Knowledge Graphs” AND (“PMBOK” OR “Project Management”) AND (“Data Visualization” OR “Decision-Making”)

From an initial total of 287 articles, only 37 were selected after applying inclusion and exclusion criteria. The inclusion criteria covered studies related to PMBOK 6 and project management challenges, as well as research on KG for management or decision-making. Exclusion criteria included irrelevant or duplicate articles.

The review revealed a limited number of studies attempting to represent PMBOK in a structured way, despite its recognized complexity. This finding reinforced the motivation for the present research and confirmed the relevance of developing a KG as a means to visualize and explore PMBOK 6’s internal structure and dependencies.

3.5.Design and Development

Inside the design and development phase, the goal was to focus on the creation of a model that represents the processes and all of their elements defined in PMBOK 6 and represent all of the connections between processes.

The development started with a detailed analysis of the PMBOK 6 structure, where the 49 processes were identified and their corresponding elements such as PG, KA, inputs, outputs, and tools. After this analysis, a UML diagram was created to serve as a guide, visually organizing these elements and clarifying the relationships between them. Based on this diagram, all nodes and relationships were manually implemented in Neo4j, ensuring that each element accurately reflected the PMBOK framework.

This manual development process allowed for a careful validation of the data and guaranteed consistency across the entire graph. The resulting model provides a visual and navigable structure that simplifies the exploration of PMBOK's complex process and theirs interconnections, making it easier to analyze how processes interact across different areas.

3.6.Demonstration

The Demonstration phase pretended to validate the applicability of the developed KG in realistic PM scenarios. To achieve this, three use cases were designed to simulate common project changes: a change in risk level, a change of technology, and a budget reduction.

Each use case scenario was analyzed starting with one process and by using Cypher queries in Neo4j it was possible to identify which processes were directly or indirectly affected by each change as well as their elements. These analyses demonstrated how a single modification could propagate across multiple processes, revealing the dependencies and connections between processes described in PMBOK 6.

The use cases confirmed that the KG could be a useful tool by providing a clearer and more structured way to visualize relationships, helping project managers anticipate the impact of changes and make more informed decisions.

3.7.Evaluation

The final phase of the DSR approach focused on evaluating how effectively the KG addressed the research objectives and the problem initially defined. This evaluation was conducted by analyzing the results obtained in the use cases, assessing how well the model represented PMBOK 6 entities and relationships, and how useful it was for exploring dependencies and supporting decision-making.

The analysis was based on three main criteria:

- Accuracy, in representing processes, inputs, and outputs as defined in PMBOK 6;
- Usability, regarding how easily users could navigate and retrieve relevant information from the KG;
- Decision support capability, evaluated through how effectively the model helped identify process dependencies and the propagation of changes.

The evaluation confirmed that the KG improves the understanding of PMBOK 6 interdependencies and enhances accessibility to its information. These results validated the approach as a useful and innovative tool for representing complex standards like PMBOK in a more analytical and interactive way.

Chapter 4 – Knowledge Graph Implementation

4.1. Tool Selection

The first step in implementing the KG for the PMBOK 6, was to identify the most suitable graph database tool for this thesis. To ensure that the chosen tool met the project's technical and analytical requirements, a comparison was conducted based on recent academic studies evaluating graph database technologies [35–38]. Inside these studies there were several tools including JanusGraph, GraphDB, OrientDB and Neo4j[35].

To guide the selection, several key evaluation criteria were defined: [36–38]

- **Query performance:** how fast and responsive the tool is when analyzing large and interconnected models
- **Resource consumption:** how efficiently the tool uses CPU and memory during graph exploration.
- **Schema flexibility and adaptability:** the ability to adapt and evolve the data structure as new elements are added.
- **Usability and query language support**: how easy the tool is to use and which query languages it supports.
- **Visualization capabilities** how well the tool allows users to view and explore relationships visually.
- **Scalability**: the capacity to handle larger and more complex datasets without losing performance.

To support decision-making, a comparative analysis was carried out below in Table 1 [35-38]. While some tools offer strong performance or advanced semantic capabilities, they may require complex configuration or provide limited usability and visualization features.

Table 1- Comparison of Graph Database Tools

Tool	Query performance	Usability / Query language	Schema flexibility	Visualisation	Scalability	Notes
Neo4j	High	High	High	Built-in visual explorer	Moderate to high	Large community and documentation
JanusGraph	High	Medium	Very high	Requires external tools	High	Requires distributed backend
GraphDB	Medium	Medium	Medium	Limited	Medium	Optimised for RDF stores
OrientDB	Medium	Medium	High	Limited	Medium to high	Multi-model (document + graph)

Considering the criteria in Table 1, Neo4j appeared to be the most appropriate solution for this project. Its intuitive visual exploration of relationships, combined with the ease of learning the Cypher query language, significantly supported the iterative and manual construction approach required in this work [38]. Furthermore, the availability of extensive documentation and supporting tools facilitated efficient problem-solving throughout the construction of the graph. [35]

Neo4j also presents some limitations. For instance, there is no simple undo functionality, which meant that every mistake during graph construction had to be corrected manually. In addition, some features are limited compared to other graph database platforms, and even at the visualization level, node names are sometimes truncated, making it less practical when working with long labels. Nevertheless, these limitations did not impact the successful construction and exploration of the PMBOK 6 KG and were acceptable within the scope of this project.

In the end, Neo4j stood out as the best option, providing the right combination of functionality and usability. Its strong capabilities in representing complex graph-based models, efficient query language (Cypher), and easy visualization made it ideal to be implemented for this thesis. Neo4j allowed the PMBOK6 structure to be represented in a natural way without information loss, while also facilitating the manual and iterative construction approach chosen for this KG.

After this careful selection, it was time to move on to the next step of the project: UML modeling as a Blueprint.

4.2. UML Modelling as a Blueprint

Once the tool was selected, it was necessary to design a clear structure to define the nodes, relationships, and connecting rules of the KG. Given the complexity and interdependence of the PMBOK 6 processes, a UML class diagram was developed to act as a blueprint for the graph implementation.

This UML model included all 49 PMBOK 6 processes, the 10 KA and the 5 PG, as well as the information elements associated with each process, as shown in Figure 3. Each process is linked to exactly one KA and one PG, and it may be connected to multiple information elements that represent its inputs, outputs and tools, as described in PMBOK 6. The class Information Element generalizes all artefacts used by the processes, while the class Information Aggregator specializes this concept by grouping more detailed Component elements through a composition relationship.

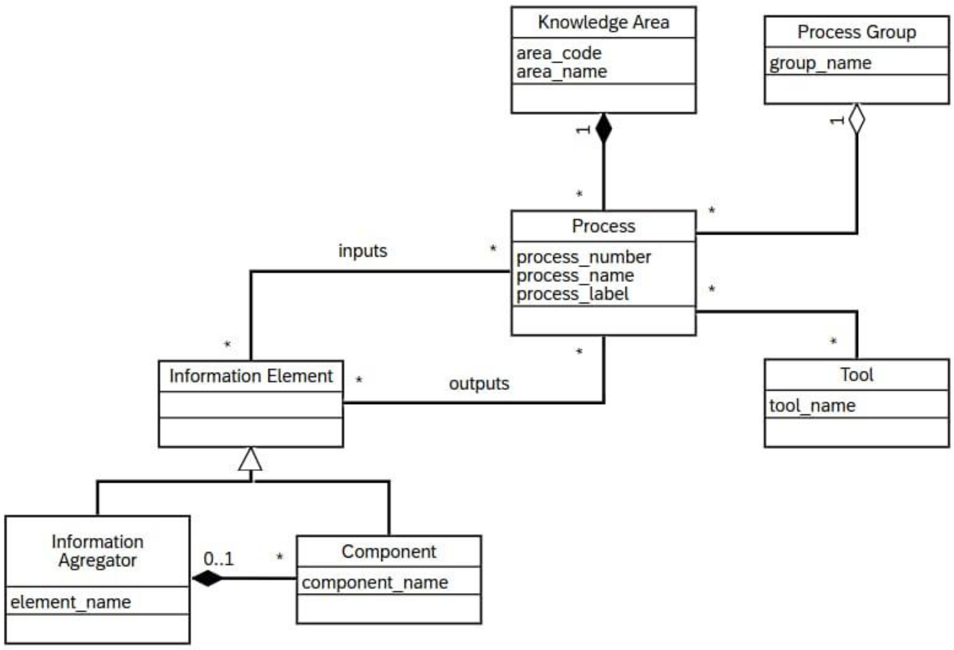

Figure 3 - UML diagram of the PMBOK 6 structure used for KG modelling

Although the UML model was not designed to capture every detailed relationship between processes, it provided a clear structural overview of the PMBOK 6 framework and served as a guide for translating its components into the KG.

The UML diagram served three main purposes throughout the project:

- **Clarity of Structure**: It provided a clear visual overview of the type of nodes, helping to identify overlaps or inconsistencies before moving to the graph implementation.
- **Guidance for Graph Construction**: By organizing how processes are connected to inputs, outputs, and tools, the UML made it easier to plan the Neo4j graph and reduced the risk of errors or omissions.
- **Consistency Check**: The diagram also helped define naming and entity types in advance, making sure everything would be uniform when nodes and relationships were created in Neo4j.

Overall, using the UML model as a blueprint ensured that the KG helped minimize manual errors, reduced rework, and provided a reliable foundation that guided the creation of all 49 PMBOK 6 processes and their associated elements in Neo4j.

4.3. Data Preparation and Modelling

Once the UML structure was defined, the next step involved preparing and modelling the data required to implement the PMBOK 6 KG. This phase required a detailed analysis of PMBOK 6 to identify which elements had to be represented in the graph to accurately reflect the structure and functioning of the PM framework.

PMBOK 6 identifies 49 processes as the core structuring elements of the framework, each belonging to a single PG and a single KA. These processes present a wide range of Inputs and Outputs and are supported by various Tools. Additionally, Components were included to represent the detailed sub-elements of Inputs, Outputs, and Tools, and they are the only elements that are not directly linked to the process. Based on this analysis, the different node types and relationships to be included in the KG were defined, as summarized in Table 2:

Table 2 - Entities and relationships in the PMBOK 6 KG

Nodes	Count	Relationship	Example
Process	49		"4.1 - Develop Project Charter"
Process Groups	5	BELONGS_TO_PG	“Planning”
Knowledge Area	10	BELONGS_TO_KA	“Scope Management”
Input	75	HAS_INPUT	"Enterprise Environmental Factors"
Output	119	HAS_OUTPUT	"Project Management Plan"
Tool	181	HAS_TOOL	"Expert Judgment"
Component	622	INCLUDES	"Marketplace Conditions"

The table above summarizes every node and relationship to be created inside the KG. All of this information was manually extracted and organized, in alignment with the relationships defined in PMBOK 6, ensuring that each process was accurately linked to the Inputs, Outputs and Tools.

Having all entities and relationships defined beforehand made the implementation in Neo4j much more structured and consistent. This preparation helped avoid missing information or creating inconsistent modelling information, ensuring that the KG was built on a clear and reliable structure.

In summary, the data preparation and modeling phase provided a clear and well-organized foundation for the KG. By organizing and defining all the PMBOK 6 elements in advance, it was possible to make sure that everything was accurately represented and ready for the next phase of implementation.

4.4.Knowledge Graph Construction in Neo4j

Once all the data was mapped and structure, the next step was to start the development of the KG inside the Neo4j tool. Inside this chapter we are able to see the mains steps followed, together with the Cypher commands used in Neo4j.

There were different ways to create the graph, but for this project the manual insertion approach was the one chosen as the main reason was to keep full control over what was being added, check every connection as it was created, and avoid introducing errors that could easily go unnoticed. Even though it took more time, this construction ensured that the graph stayed consistent and aligned with the PMBOK 6 structure.

Inside the KG there are the nodes as processes, KA, PG, inputs, outputs, tools, and components, while the edges represented the connections between these entities as it is describe in the PMBOK 6.

The first step was to create the five PG and the ten KA defined in PMBOK6. To generate these nodes efficiently, an UNWIND clause was used in Neo4j, allowing multiple nodes to be created within a single Cypher query, as seen in code block 1:

```
UNWIND [
  'Initiating',
```

```
  'Planning',
  'Executing',
  'Monitoring and Controlling',
  'Closing'
] AS group
MERGE (:ProcessGroup {name: group});
UNWIND [
  'Integration Management',
  'Scope Management',
  'Schedule Management',
  'Cost Management',
  'Quality Management',
  'Resource Management',
  'Communication Management',
  'Risk Management',
  'Procurement Management',
  'Stakeholder Management'
] AS area
MERGE (:KnowledgeArea {name: area});
```

Code Block 1 - Cypher query to create Process Groups and Knowledge Areas

Each of the 49 processes was then created and linked to its respective PG and KA, as it can be seen an example of the process '4.3 – Direct and Manage Project Work' in the code block 2:

```
MERGE (p:Process {id:'4.3', label:'4.3 – Direct and Manage Project Work'})
 MERGE (g:ProcessGroup {name:'Executing'})
```

```
MERGE (k:KnowledgeArea {name:'Integration Management'})

MERGE (p)-[:BELONGS_TO_PG]->(g) MERGE (p)-[:BELONGS_TO_KA]->(k);
```

Code Block 2 - Cypher Query to create processes

After inserting all the processes, PG, and KA, a verification query was used to confirm if everything was correctly created and connected. This was part of a verification step in order to ensured that no process was missing or incorrectly linked before adding the inputs, outputs, tools, and components. The cypher query used can be seen in the code block 3:

```
MATCH (p:Process)-[:BELONGS_TO_PG]->(g:ProcessGroup), (p)-[:BELONGS_TO_KA]->(k:KnowledgeArea)

RETURN p.name AS Process, g.name AS Group, k.name AS KnowledgeArea ORDER BY g.name, k.name, p.name;
```

Code Block 3 - Cypher Query to validate the nodes created

Once the process were created, the next step wat to add all the input, output, tool, and component associated with the corresponding process. This was with the use of **MERGE** which ensured that no duplicates were created. This approach kept the graph consistent, even when the same element was shared across multiple processes. The following query in code block 4 shows an example on how the process Direct and Manage Project Work including its inputs, outputs, tools, and components was created:

```
 MERGE (p:Process {id:'4.3'})

FOREACH (inputName IN [

  "Project Management Plan",

  "Change log",

  "Lessons learned register",

  "Milestone list",

  "Project communications",
```

```
  "Project schedule",
  "Requirements traceability matrix",
  "Risk register",
  "Risk report",
  "Approved change requests"
] |
  MERGE (i:Input {name: inputName})
  MERGE (p)-[:HAS_INPUT]->(i)
)
MERGE (eef:Input {name: "Enterprise Environmental Factors"})
MERGE (p)-[:HAS_INPUT]->(eef)
MERGE (eefc1:Component {name: "Stakeholder risk thresholds"})
MERGE (eef)-[:INCLUDES]->(eefc1)
MERGE (opa:Input {name: "Organizational Process Assets"})
MERGE (p)-[:HAS_INPUT]->(opa)
MERGE (opac1:Component {name: "Organizational standard policies, processes, and procedures"})
MERGE (opa)-[:INCLUDES]->(opac1)
MERGE (opac2:Component {name: "Issue and defect management procedures defining issue and defect controls, issue and defect identification and resolution, and action item tracking"})
MERGE (opa)-[:INCLUDES]->(opac2)
MERGE (opac3:Component {name: "Issue and defect management databases containing historical issue and defect status, resolution, and action item results"})
MERGE (opa)-[:INCLUDES]->(opac3)
MERGE (opac4:Component {name: "Performance measurement database used to collect and make available measurement data on processes and products"})
```

```
MERGE (opa)-[:INCLUDES]->(opac4)
MERGE (opac5:Component {name: "Change control and risk control procedures"})
MERGE (opa)-[:INCLUDES]->(opac5)
MERGE (opac6:Component {name: "Project information from previous projects"})
MERGE (opa)-[:INCLUDES]->(opac6)
MERGE (tool1:Tool {name: "Expert judgement"})
MERGE (p)-[:HAS_TOOL]->(tool1)
FOREACH (compName IN [
  "Technical knowledge on the industry and focus area of the project",
  "Cost and budget management",
  "Legal and procurement",
  "Legislation and regulations",
  "Organizational governance"
] |
  MERGE (tc:Component {name: compName})
  MERGE (tool1)-[:INCLUDES]->(tc)
)
FOREACH (toolName IN ["Meetings", "Project Management Information System"] |
  MERGE (t:Tool {name: toolName})
  MERGE (p)-[:HAS_TOOL]->(t)
)
MERGE (out1:Output {name: "Deliverables"})
MERGE (p)-[:HAS_OUTPUT]->(out1)
MERGE (out2:Output {name: "Work performance data"})
MERGE (p)-[:HAS_OUTPUT]->(out2)
MERGE (out3:Output {name: "Issue Log"})
```

```
MERGE (p)-[:HAS_OUTPUT]->(out3)
FOREACH (compName IN [
  "Issue type",
  "Who raised the issue and when",
  "Description",
  "Priority",
  "Who is assigned to the issue",
  "Target resolution date",
  "Status",
  "Final solution"
] |
  MERGE (c:Component {name: compName})
  MERGE (out3)-[:INCLUDES]->(c)
)
MERGE (out4:Output {name: "Change Requests"})
MERGE (p)-[:HAS_OUTPUT]->(out4)
FOREACH (cr IN ["Corrective action", "Preventive action", "Defect repair", "Updates"] |
  MERGE (crc:Component {name: cr})
  MERGE (out4)-[:INCLUDES]->(crc)
)
FOREACH (updateName IN [
  "Project Management Plan ",
  "Activity list ",
  "Assumption log ",
  "Lessons learned register ",
```

```
  "Requirements documentation ",

  "Risk register ",

  "Stakeholder register ",

  "Organizational Process Assets "

] |

  MERGE (uo:Output {name: updateName})

  MERGE (p)-[:HAS_OUTPUT]->(uo)

)
```

Code Block 4 - Cypher Query to create all process elements

Ater finishing the process creation, Figure 4 presents how the process would appear in the KG. This example makes it clear how processes are connected to their inputs, outputs and tools, as well as to their respective KA and PG.

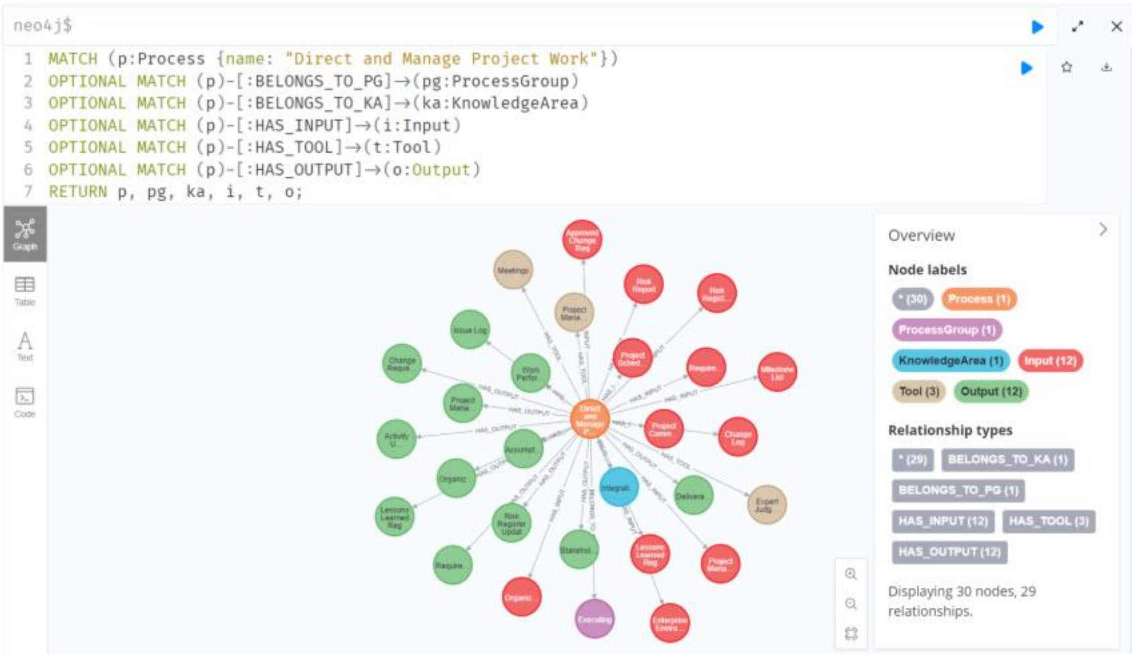

Figure 3 - 4.3 – Direct and Manage Project Work process

Overall, this manual construction process was essential to guarantee accuracy. By following this **step-by-step approach**, this process minimizes the risk of inconsistencies and ensures that the graph accurately reflects the PMBOK6 structure.

4.5.Challenges and Solutions

The KG development presented multiple challenges, starting from the initial data analysis to the manual creation of nodes and relationships in Neo4j. These challenges were due to the complexity of PMBOK 6, and the extensive amount of information associated with its 49 processes.

The first challenge involved analyzing and organizing the raw PMBOK6 data. Each process contains numerous elements such as inputs, outputs and tools which made the manual filtering, categorizing, and structuring this information time-consuming. The UML supported this step by helping to identify the different types of nodes that needed to be created and clarifying how they should be represented in the graph.

Another significant challenge was ensuring consistency and accuracy**.** To prevent duplicate nodes or incorrect connections, simple naming conventions were followed (for example, using capital letters consistently and keeping uniform names across all entities). Each relationship was carefully verified using Cypher queries, such as "MERGE**"** statements, this allowed checking if an element already existed in the graph and avoided the creation of duplicates or incorrect connections.

Finally, the manual data insertion process itself posed challenges. While automated imports could have been faster, they increased the risk of mistakes or misaligned relationships. The chosen approach of manually creating nodes and edges allowed for incremental verification, ensuring the graph accurately represented the PMBOK6 framework.

In summary, these challenges were mainly addressed through:

- A detailed analysis of PMBOK6 to identify what information needed to be represented.
- The creation of a UML diagram that helped to structure and visualize the elements before building the graph.
- The manual construction of nodes and relationships in Neo4j, following simple rules (for example, consistent naming conventions and the use of capital letters) to keep everything uniform.

By adopting this structured approach, the KG construction was able to reflect the PMBOK6 structure while minimizing errors. This careful methodology ensures the graph is reliable and ready for the use case analyses in the following chapter, where its practical advantages can be demonstrated.

4.6.Knowledge Graph Overview

After completing the UML modeling, data preparation, and the manual creation of nodes and relationships in Neo4j, the KG presents a complete visualization of the PMBOK6 49 processes with their associated elements which include inputs, outputs, and tools. To make the visualization clearer, a color code was applied to differentiate the types of entities represented in the graph, as the Figure 4 shows:

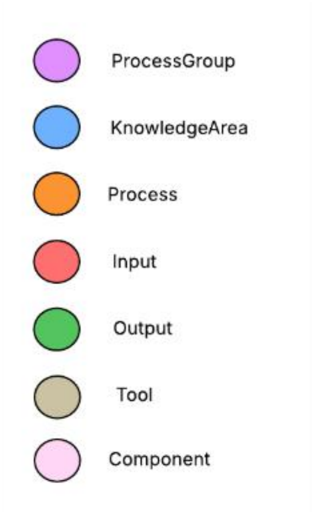

Figure 4 - KG Color Code

An example of a section of the KG is illustrated in the Figure 6 below. It is important to note that this visualization represents only a subset of the graph, since the full KG contains hundreds of nodes and relationships, which would be too dense to interpret effectively in a single image:

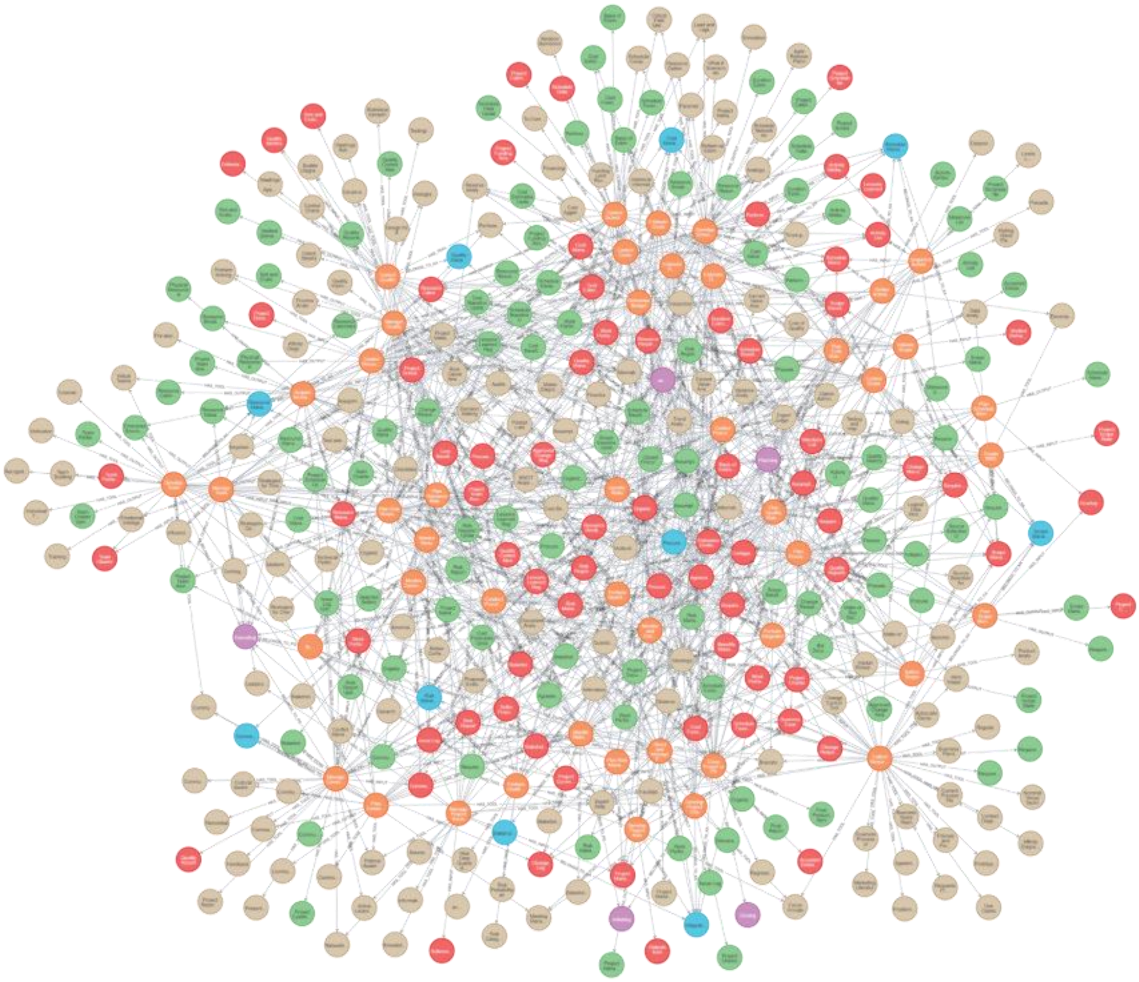

Figure 5 - PMBOK 6 Knowledge Graph

In the Figure 6 we can see all the elements represented inside the KG:

- **Nodes** representing processes, inputs, outputs, tools, components, KA, and PG.
- **Edges** showing the relationships between these entities, making clear how PMBOK6 elements are connected.

Although the complete KG is too large to analyze visually, its real value lies in the ability to query it using Cypher. The queries allow users to explore specific parts of the graph making it possible to extract insights that are not immediately visible in global visualization.

The KG enables users to explore the PMBOK 6 through an interactive way which shows processes and their connections and understand how changes in one element propagate across other processes. This interactivity highlights process interdependencies, making complex PM structures more comprehensible.

In conclusion, the careful modeling and construction process resulted in a functional and reliable KG, providing a foundation for practical applications and use cases in the next chapter.

Chapter 5 – Use case scenarios

5.1.Knowledge Graph Overview

In this chapter, three use cases are developed to demonstrate how the KG can support project managers in analysing the propagation of changes across PMBOK 6 processes. The goal is to show how the KG helps in the identification of dependencies between processes and the corresponding impacts on project artefacts.

Each use case begins by describing a project situation and identifying the process where the change is executed. From this starting point, the KG is queried to retrieve the outputs affected by the change and to trace the processes that depend directly on those updated outputs. This method reveals how a single change can trigger adjustments beyond its original context, exposing hidden dependencies in the project plan.

The use cases elaborated and subsequently selected were specifically designed to illustrate recurrent challenges in PM. These cases are presented in ascending order of complexity and impact: (i) a variation in risk level, (ii) the adoption of new technology, and (iii) a reduction in project budget. This sequencing was intended to provide a progressive analysis, beginning with challenges of a more contained scope and advancing towards issues with broader implications in PM practices.

5.2.Use Case 1: Change in Risk Level

During the execution of a project with a qualitative risk assessment, a change was detected in the probability of occurrence of a previously identified risk. This variation required an update to the project's risk documentation, which, according to the PMBOK 6, is handled within Process **11.3 – Perform Qualitative Risk Analysis**, located in the Planning PG and the Risk Management KA.

This process reviews and updates the assessment of each identified risk, estimating its probability and potential impact, and produces revised project documents that describe how the project's exposure to risk has changed.

To begin the analysis, the KG was queried to identify the outputs of Process 11.3, that were affected by the update in the assessed risk level. Inside the code block 5 we can see the cypher query used:

```
MATCH (p:Process {id:"11.3"})-[:HAS_OUTPUT]->(o:Output)
RETURN o.name;
```

Code Block 5 - Cypher query for 11.3 process output search

This query returned four updated outputs: Assumption Log, Risk Register, Issue Log, and Risk Report. These outputs consolidate the information resulting from the re-evaluation of risk probability and impact, and they represent the starting points through which this change may propagate across the project structure. To determine which processes directly depend on these outputs, the following query on code block 6 was executed:

```
MATCH (p1:Process {id:"11.3"})-[:HAS_OUTPUT]->(o:Output)
MATCH (i:Input)
WHERE toLower(i.name) = toLower(o.name)
MATCH (i)<-[:HAS_INPUT]-(p2:Process)-[:BELONGS_TO_PG]->(:ProcessGroup {name:"Planning"})
RETURN DISTINCT p2.label AS ImpactedPlanningProcess, i.name AS InputMatched, o.name AS OutputMatched
ORDER BY ImpactedPlanningProcess;
```

Code Block 6 - Cypher query for identifying dependent Planning processes for updated risk information

The query identified multiple processes that directly depend on the updated risk information. In total, 56 relationships were found between outputs and inputs, meaning that some processes appeared multiple times because they were connected to more than one output from Process 11.3. For example, a process could be linked once through the Risk Register and again through the Issue Log. After removing these duplicates and counting only unique processes, 15 distinct planning processes were found to be impacted at the first level of dependency. In Table 3, it can be seen a representative sample of impacted processes (the complete list is available in Appendix A).

Table 3 - Impacted processes from 11.3 process

Output (from 11.3)	Input	Impacted Process
Risk Register	Risk Register	11.5 – Plan Risk Responses
Risk Register	Risk Register	7.2 – Estimate Costs
Risk Register	Risk Register	7.3 – Determine Budget
Risk Register	Risk Register	6.5 – Develop Schedule

These results reveal that a change in risk probability propagates beyond the Risk Management KA. Processes such as Plan Risk Responses, Estimate Costs, Determine Budget, and Develop Schedule must be reassessed, as they rely on risk information when defining estimates and establishing project baselines.

Monitoring and control processes, including Monitor Risks, Monitor and Control Project Work, and Perform Integrated Change Control, are also affected, as they depend on updated reports and logs to supervise progress and authorize adjustments. Other processes, like Manage Communications or Plan Stakeholder Engagement, experience secondary effects through shared documentation.

To analyse the second level of dependencies, we extended the search to the outputs produced by the first level impacted processes and identified which additional processes use those outputs as inputs. This expanded search returned more than six hundred records, which, although not representing the same number of distinct processes, indicate that a large majority of PMBOK processes would eventually be connected to the updated risk information if the propagation were not limited. To preserve analytical clarity, the propagation was therefore limited to the first dependency level, focusing on practical, actionable impacts and leaving out connections that are only informational. This decision reflects the type of reasoning a project manager would apply when defining the practical boundaries of an impact assessment.

From a management perspective, the KG supports a structured approach to responding to this change. First, the Risk Register and related documents (Assumption Log, Issue Log, Risk Report) should be updated to reflect the new probability values.

Next, planning processes that depend on these data, Plan Risk Responses, Estimate Costs, Determine Budget, and Develop Schedule, should be reviewed to make sure that project baselines remain consistent with the updated risk information. Communication and stakeholder management activities must be informed of these updates, while control processes such as Monitor Risks and Perform Integrated Change Control verify that the revised information is correctly integrated into project governance.

In conclusion, the analysis of the change in risk probability highlighted how even a minor adjustment in qualitative risk evaluation can generate wide-ranging effects across project planning and control. The case showed that revising risk data does not remain confined to the Risk Management area but immediately influences cost, schedule, and communication planning.

This use case therefore illustrates the systemic nature of risk information within PMBOK 6 and the importance of treating risk updates as starting points for coordinated updates throughout the project.

5.3. Use Case 2: Change of technology

During the planning of a project, a change was requested regarding the technology to be implemented in one of the main deliverables. Since this modification alters the characteristics, functionality, or components included in the scope of the project, it required adjustments to scope definition. According to the PMBOK 6, such change is handled within **Process 5.3 – Define Scope**, located in the Planning PG and the Scope Management KA.

To begin the analysis, the KG was queried to identify the outputs of Process 5.3 that were affected by the technology change, and this can be seen in the code block 7 below:

```
MATCH (p:Process {id:"5.3"})-[:HAS_OUTPUT]->(o:Output)
RETURN o.name;
```

Code Block 7 - Cypher query for 5.3 process output search

This query returned five updated outputs: Assumption Log, Requirements Documentation, Stakeholder Register, the Project Scope Statement, and Requirements Traceability Matrix. These outputs consolidate the updated scope information and represent the starting points through which the change propagates across the project structure. To understand how these changes affect the rest of the planning phase, the KG was queried, code block 8, to identify which planning processes depend directly on these updated outputs:

```
MATCH (p1:Process {id:"5.3"})-[:HAS_OUTPUT]->(o:Output)

MATCH (i:Input)

WHERE toLower(i.name) = toLower(o.name) MATCH (i)<-[:HAS_INPUT]-(p2:Process)-[:BELONGS_TO_PG]->(:ProcessGroup {name:"Planning"})

RETURN DISTINCT p2.label AS ImpactedPlanningProcess, i.name AS InputMatched, o.name AS OutputMatched

ORDER BY ImpactedPlanningProcess;
```

Code Block 8 - Cypher query for identifying dependent Planning processes for updated scope information

The query identified 17 distinct processes within the Planning PG that directly depend on the updated scope information. These results demonstrates the interconnections between scope definition and other planning areas such as cost, schedule, resource, and stakeholder management. Inside table 4 there is a representative subset of these processes, while the complete list is available in Appendix B.

Table 4 - Impacted processes from 5.3 process

Output (from 5.3)	Input	Impacted Process
Assumption Log	Assumption Log	Develop Schedule
Assumption Log	Assumption Log	Plan Stakeholder Engagement
Requirements Documentation	Requirements Documentation	Create WBS
Requirements Documentation	Requirements Documentation	Plan Resource Management

Stakeholder Register	Stakeholder Register	Plan Communications Management
Stakeholder Register	Stakeholder Register	Identify Risks
Project Scope Statement	Project Scope Statement	Create WBS

These results from the cypher query show how a change in the technology can affect many more processes than might initially be expected. With this type of results we can look at processes such as Develop Schedule, Estimate Activity Durations, and Estimate Activity Resources understand they need to be revised in order to ensure that effort, duration, and resource plans are modify accordingly.

Other processes involving stakeholder engagement and communication (Plan Stakeholder Engagement, Plan Communications Management) are also impacted, andunderstand that they need to be revised of technological requirements may change stakeholder needs and communication priorities. Additionally, procurement planning (Plan Procurement Management) may require adjustments if new materials, licenses, or suppliers are introduced.

To gain a different perspective from the one explore so far and to simulate a more realistic decision-making scenario, a second query, in code block 8, was executed where the output was specified. In this case, it was selected the Project Scope Statement as the primary output affected by the change in technology, since it defines the project's deliverables and boundaries:

```
MATCH (p1:Process {id:"5.3"})-[:HAS_OUTPUT]->(o:Output {name:"Project Scope Statement"})

MATCH (i:Input) WHERE toLower(i.name) = toLower(o.name)

MATCH (i)<-[:HAS_INPUT]-(p2:Process)-[:BELONGS_TO_PG]->(:ProcessGroup {name:"Planning"})

RETURN DISTINCT p2.label AS ImpactedPlanningProcess, i.name AS InputMatched, o.name AS OutputMatched ORDER BY ImpactedPlanningProcess;
```

Code Block 9 - Cypher query for identifying dependent Planning processes for Project Scope Statement output

The Cypher query executed only returned one process, the **Process 5.4 – *Create WBS***, which defines the project's deliverables and forms the basis for schedule, cost, and resource planning. This result shows that the KG can be filtered and used according to what is needed in each situation. Once the WBS is updated, the change indirectly affects other areas such as Schedule Management, Cost Management, and Resource Management, since they all depend on the work packages defined within it.

From a management point of view, this helps to understand how the KG can be used for both a general and a more detailed analysis. The first query gives a complete view of all the processes affected by a change in technology, while the second focuses on a specific output, allowing project managers to analyze the impact of a single document or decision.

In the end, this use case shows how even a small technology change can spread across several planning areas, forcing adjustments to effort, procurement, and risk strategies. It also highlights the strong dependencies between scope definition and the other planning processes in PMBOK 6, showing the importance of carefully analyzing technological decisions to keep the project aligned.

5.4.Use Case 3: Budget Reduction

During the planning of a project, a reduction of the approved budget was imposed by the project sponsor. According to PMBOK 6, such a modification is handled within **Process 4.1 — Develop Project Charter**, located in the Initiating PG and the Integration Management KA. This process formalizes the project's existence and documents the high-level constraints under which it must be delivered, including the initial budget allocation.

To begin the analysis, the KG was queried, as it is shown on the code block 10, to identify the outputs of Process 4.1 affected by the revised budget decision:

```
MATCH (p:Process {id:"4.1"})-[:HAS_OUTPUT]->(o:Output)
```

```
RETURN o.name;
```

Code Block 10 - Cypher query for 4.1 process output search

This query returned two outputs: the Project Charter and the Assumption Log. These documents capture the fundamental definition of the project, including boundaries and major financial assumptions. Updating them is the first step of a budget change within the PMBOK process structure.

Once the main outputs were discovered, the next step wat to determine which planning processes depend directly on these outputs, and so a second query was executed, as shown in code block 11:

```
MATCH (p1:Process {id:"4.1"})-[:HAS_OUTPUT]->(o:Output)
MATCH (i:Input) WHERE toLower(i.name) = toLower(o.name) MATCH (i)<-[:HAS_INPUT]-(p2:Process)-[:BELONGS_TO_PG]->(:ProcessGroup {name:"Planning"})
RETURN DISTINCT p2.label AS ImpactedPlanningProcess, i.name AS InputMatched, o.name AS OutputMatched
ORDER BY ImpactedPlanningProcess;
```

Code Block 11 - Cypher query for identifying dependent Planning processes for updated budget information

This query identified 21 distinct planning processes that rely directly on the Project Charter or the Assumption Log. Inside the table 5 it is presented a representative subset of these processes. The complete list is provided in Appendix C.

Table 5 - Impacted processes from 4.1 process

Output (from 4.1)	**Input**	**Impacted Process**
Project Charter	Project Charter	Develop Project Management Plan
Project Charter	Project Charter	Plan Cost Management
Project Charter	Project Charter	Plan Schedule Management

Project Charter	Project Charter	Plan Resource Management
Assumption Log	Assumption Log	Estimate Activity Durations
Assumption Log	Assumption Log	Identify Risks
Project Charter	Project Charter	Plan Stakeholder Engagement

These results confirm that a reduction in budget does not only affect cost management alone. It requires re-alignment of nearly all planning processes that rely on financial viability. Cost, schedule, scope, procurement, quality, risk and stakeholder planning must all be revised to reflect the reduced resources and priorities.

From a management perspective, the KG supports a structured response to this change. First, the Project Charter and associated assumptions must be updated to formalize the new financial conditions. Then, planning processes that directly rely on these inputs, in particular cost, schedule and resource planning, should be reviewed to ensure that the revised funding limits are reflected consistently across the PM plan.

From the 21 impacted processes identified in the analysis, around six belong to the Cost Management KA, five to Schedule Management KA, and four to Resource Management KA, while the remaining ones are distributed along Scope, Risk, Quality, and Stakeholder Management. This distribution confirms that the impact of a budget reduction extends well beyond cost-related processes, influencing several dimensions of project planning and coordination.

In conclusion, this use case highlights that a budget reduction propagates across multiple planning activities and may compromise the project if not properly updated. This case highlights that the financial constraints defined in the Project Charter are key drivers of planning decisions, and therefore any modification must be thoroughly assessed and effectively communicated to maintain alignment and deliverability.

5.1.Use Case Conclusions

The analysis of the three use cases provided valuable insights into how the developed KG supports PM practice within the PMBOK 6 framework. Each case simulated realistic project situations, allowing the observation of how specific changes can propagate across Processes, Inputs, Outputs, and Tools.

Through this analysis, it became clear that the KG effectively makes process interdependencies visible and traceable. Even small modifications introduced in one process could be followed through the network of relationships to identify which other processes would be affected by the resulting change. This visibility is difficult to achieve using the traditional PMBOK documentation and demonstrates the value of the graph-based approach for understanding project dynamics.

As the KG was used and explored during the study of the use cases, several improvements were identified and progressively incorporated. For example, the inclusion of process numbers helped to facilitate navigation and interpretation, while additional attributes, such as the state of Outputs provided more detailed and accurate representations of process results. These adjustments appeared directly from the practical application of the KG and contributed to demonstrate its usability and consistency.

The results also showed that the KG supports one of the main objectives of this research: enabling a clearer and faster understanding of how information flows within the PMBOK structure. By querying the graph, it was possible to identify dependencies between processes and anticipate how updates in one area could affect others, addressing a key challenge in PM identified in the literature.

In summary, the use cases not only validated the usefulness of the KG but also served as an iterative stage for its improvement. They demonstrated that a KG can effectively support the analysis of dependencies and impact propagation in PMBOK 6, providing a more transparent and connected representation of PM knowledge.

Chapter 6 - Conclusions and future research

6.1. Main Conclusions

This thesis addresses one of the main challenges of PM practice, the difficulty of understanding and managing the complex interdependencies defined in PMBOK 6. The research question that guided this work was: How can Knowledge Graphs improve understanding, management, and decision-making in project management using the PMBOK 6 framework?

To answer this question, three main objectives were established:

1. to analyse the key processes and KA in PMBOK 6 and determine how they could be effectively represented through a KG;
2. to design and implement the KG model in Neo4j, mapping all entities and relationships between processes, Inputs, Outputs and Tools; and
3. to evaluate its usefulness through practical use cases simulating real PM scenarios.

The work began with an in-depth analysis of PMBOK 6, identifying its 49 processes, ten KA and five PG, as well as all related Inputs, Outputs and Tools. This analysis supported the design of a data model that was then implemented as a KG using Neo4j. The resulting model provided a structured and connected representation of the PMBOK framework, allowing its complex relationships to be visualised and queried interactively.

The use cases were an important component of this thesis, demonstrating the practical value that the KG can provide. By simulating situations such as a change in risk level, the adoption of a new technology, and a reduction in project budget, it was possible to observe how a modification affecting a single process can propagate to several others. By analysing the Outputs of the affected process, the KG made it possible to identify which other processes depend on that information, making the propagation of change explicit and traceable within the project structure.

This approach demonstrated several benefits. By making dependencies between PMBOK 6 processes explicit and visual, the KG presents a faster comprehension of project impacts and helps project managers anticipate the consequences of change more effectively. It also reduces the effort required to consult the traditional PMBOK documentation, offering a more intuitive and connected representation of the standard.

It was also possible to determine some limitations since the graph was manually constructed, which introduces the possibility of human error, and the scope of this work was restricted to PMBOK 6, even though PM has evolved and newer frameworks now exist, like is the case of PMBOK 7.

Despite these limitations, the goals defined for this dissertation were successfully achieved. The results confirm that a KG can serve as a valuable tool for representing complex PM frameworks such as PMBOK 6, improving their accessibility, usability and relevance in practical decision-making contexts.

6.2.Future Research Proposals

The work conducted in this dissertation opens several opportunities for future research and improvement.

An improvement could concerns the scope of the model. Although PMBOK 6 remains widely applied, PM practices continue to evolve. Extending the graph to incorporate PMBOK 7 or integrating additional frameworks, for example PRINCE2, would support other projects that adopt other frameworks.

Finally, the validation conducted in this work was based on simulated use case scenarios. A valuable next step would be to evaluate the KG with project managers in real operational contexts, gathering evidence on its usability, usefulness and impact on decision-making efficiency. Such feedback would support further refinement and confirm the practical contribution of the approach.

References

[1]. Silva, F. & Nuñez, A. (2023). Methodological proposal for the integration of agile methodologies and PMBOK. International Journal of Engineering Insights, 1, 09-12.

[2]. Amaro, F. & Domingues, L. (2023). PMBOK 6th meets 7th: How to link both guides in order to support project tailoring? Procedia Computer Science, 219, 1877-1884.

[3]. Tabassi, A. A., Bryde, D. J., Kamal, E. M., Michaelides, R., & Dowson, J. (2017). Challenges for project management in the 21st century. In P. A. J. Wahid, P. I. D. A. Aziz Abdul Samad, P. D. S. Sheikh Ahmad, & A. P. D. P. Pujinda (Eds.), Carving the future built environment: Environmental, economic and social resilience (Vol. 2, pp. 631-641). European Proceedings of Multidisciplinary Sciences.

[4]. Ji, S., Pan, S., Cambria, E., Marttinen, P., & Yu, P. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems, PP.

[5]. Vivar, J., Segarra, J., Villazon Terrazas, B., & Saquicela, V. (2020). REDI: Towards knowledge graph-powered scholarly information management and research networking. Journal of Information Science, 48.

[6]. Putri, P., Raharjo, T., Hardian, B., & Simanungkalit, T. (2023). Challenges and best practices solution of agile project management in public sector: A systematic literature review. JOIV: International Journal of Informatics and Visualization, 7.

[7]. Valadares, F. S., Moura, N. C., Pereira, T. N., & Arantes, M. D. (2024). Identification of the main traditional project management methods through a systematic literature review. International Journal of Advanced Computer Science and Applications.

[8]. Akhavan Tabassi, A., Bryde, D., Mustafa Kamal, E., & Dowson, J. (2019). Challenges for project management in the 21st century. European Proceedings of Multidisciplinary Sciences.

[9]. Simonaitis, A., Daukšys, M., & Mockienė, J. (2023). A comparison of the project management methodologies PRINCE2 and PMBOK in managing repetitive construction projects. Buildings, 13(7), 1796.

[10]. Isah, M., & Kim, B. (2023). Development of knowledge graph based on risk register to support risk management of construction projects. KSCE Journal of Civil Engineering, 27.

[11]. Chen, M., Tao, Z., Tang, W., Qin, T., Yang, R., & Zhu, C. (2023). Enhancing emergency decision-making with knowledge graphs and large language models. arXiv preprint.

[12] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering (EBSE 2007-001)," Keele University and Durham University Joint Report, 2007.

[13] V. Vaishnavi and W. Kuechler, Design Science Research Methods and Patterns: Innovating Information and Communication Technology, 2nd ed. Boca Raton, FL, USA: CRC Press, 2015.

[14]. Rosenberger, P. & Tick, J. (2018). Suitability of PMBOK 6th edition for agile-developed IT projects. CINTI 2018 - IEEE 19th International Symposium on Computational Intelligence and Informatics, 241-246.

[15]. Abdullah, A., Abdul-Samad, Z., Abdul-Rahman, H., & Salleh, H. (2021). Project management methods, guides and standards: A critical overview. Journal of Project Management Practice, 1, 35-51.
[16]. Project Management Institute. (2021). A guide to the project management body of knowledge (PMBOK Guide) – Seventh Edition. Project Management Institute.
[17]. Rosenberger, P. & Tick, J. (2019). Relevance of PMBOK v6 processes for tailored agile project categories. SACI 2019 - IEEE 13th International Symposium on Applied Computational Intelligence and Informatics, 87-94.
[18]. Lucidchart. (n.d.). What is PMBOK? Lucidchart. Retrieved November 3, 2024.
[19]. Bomfin, D. F., Nunes, P. C. A., & Hastenreiter, F. (2012). Gerenciamento de projetos segundo o guia PMBOK: Desafios para os gestores. Revista Universidade do Estado de Minas Gerais – UEMG.
[20]. Rosenberger, P. & Tick, J. (2018). Suitability of PMBOK 6th edition for agile-developed IT projects. CINTI 2018 - IEEE 19th International Symposium on Computational Intelligence and Informatics, 241-246.
[21]. Bomfin, D. F., Nunes, P. C. A., & Hastenreiter, F. (2012). Gerenciamento de projetos segundo o guia PMBOK: Desafios para os gestores. Revista de Gestão e Projetos – GeP, 3(3), 58-87.
[22]. Peng, C., Xia, F., Naseriparsa, M., & Osborne, F. (2023). Knowledge graphs: Opportunities and challenges. Artificial Intelligence Review, 56, 1-32.
[23]. Sikos, L. (2023). Cybersecurity knowledge graphs. Knowledge and Information Systems, 65, 1-21.
[24]. Xi, R. (2024). Applications of knowledge graph in medical and financial fields: Data integration and intelligent decision-making from an interdisciplinary perspective. Applied and Computational Engineering, 67, 300-306.
Data integration and intelligent decision-making from an interdisciplinary perspective. Applied and Computational Engineering, 67, 300-306.
[25]. Kalayci, T., Bricelj, B., Lah, M., Pichler, F., & Scharrer, M. (2021). A knowledge graph-based data integration framework applied to battery data management. Sustainability, 13, 1583.
[26]. Li, H., Appleby, G., Brumar, C., Chang, R., & Suh, A. (2023). Knowledge graphs in practice: Characterizing their users, challenges, and visualization opportunities. IEEE Transactions on Visualization and Computer Graphics, PP, 1-11.
[27]. Paulheim, H. (2017). Knowledge graph refinement. Semantic Web Journal, 8(6), 1011–1030.
[28]. Zambrano, L., Atencio, E., Mariani, C., & Mancini, M. (2024). Integration between PMBOK 7th concepts: A network analysis.
[29]. Zhao, Y., Chen, L., Wang, M., & Xu, H. (2022). Improving project management with knowledge graphs. International Journal of Project Management, 40(3), 215–228.
[30]. Smith, A., & Jones, R. (2020). Customizing PMBOK with ontologies: Enhancing flexibility in project methodologies. Project Management Journal, 51(2), 145–156.
[31]. Li, J., Sun, Y., & Wu, Q. (2019). Visual navigation in knowledge graphs: Techniques and challenges. IEEE Transactions on Visualization and Computer Graphics, 25(1), 248–257.
[32]. Garcia, M., López, D., & Torres, P. (2021). Evolving knowledge graphs in dynamic environments. Data Engineering, 44(7), 389–400.

[33]. Fernandez, C., Silva, N., & Pereira, T. (2023). Benefits of knowledge graph integration in project management. Journal of Systems and Software, 198, 111512.

[34]. Vaishnavi, V., & Kuechler, W. (2004). Design Science Research Process: A Model for Producing and Presenting Information Systems Research. Association for Information Systems, Technical Report.

[35]. Angles, R. (2012). A Comparison of Current Graph Database Models. In Proceedings of the IEEE 28th International Conference on Data Engineering Workshops (ICDEW). DOI:10.1109/ICDEW.2012.31

[36]. Timón-Reina, S., et al. (2021). An Overview of Graph Databases and Their Applications in the Biomedical Domain: A Systematic Review. Briefings in Bioinformatics.

[37]. Monteiro, J., Sá, F., & Bernardino, J. (2023). Experimental Evaluation of Graph Databases: JanusGraph, Nebula Graph, Neo4j, and TigerGraph. Applied Sciences, 13(9), 5770.

[38] Besta, M., et al. (2019). Demystifying Graph Databases: Analysis and Taxonomy of Data Organisation, System Designs, and Graph Queries. arXiv:1910.09017.

Appendix A

Impacted Process	Input Matched	Output Matched
4.7 – Close Project or Phase	Assumption Log	Assumption Log
4.7 – Close Project or Phase	Risk Register	Risk Register
4.7 – Close Project or Phase	Issue Log	Issue Log
4.7 – Close Project or Phase	Risk Report	Risk Report
5.2 – Collect Requirements	Assumption Log	Assumption Log
12.2 – Conduct Procurements	Risk Register	Risk Register
12.3 – Control Procurements	Assumption Log	Assumption Log
12.3 – Control Procurements	Risk Register	Risk Register
5.3 – Define Scope	Assumption Log	Assumption Log
5.3 – Define Scope	Risk Register	Risk Register
7.3 – Determine Budget	Risk Register	Risk Register
6.5 – Develop Schedule	Assumption Log	Assumption Log
6.5 – Develop Schedule	Risk Register	Risk Register
4.3 – Direct and Manage Project Work	Risk Register	Risk Register
4.3 – Direct and Manage Project Work	Risk Report	Risk Report
6.4 – Estimate Activity Durations	Assumption Log	Assumption Log
6.4 – Estimate Activity Durations	Risk Register	Risk Register
6.3 – Estimate Activity Resources	Assumption Log	Assumption Log
6.3 – Estimate Activity Resources	Risk Register	Risk Register
7.2 – Estimate Costs	Risk Register	Risk Register
11.2 – Identify Risks	Assumption Log	Assumption Log
11.2 – Identify Risks	Issue Log	Issue Log
13.1 – Identify Stakeholders	Issue Log	Issue Log
11.6 – Implement Risk Responses	Risk Register	Risk Register
11.6 – Implement Risk Responses	Risk Report	Risk Report
10.2 – Manage Communications	Issue Log	Issue Log
10.2 – Manage Communications	Risk Report	Risk Report
8.2 – Manage Quality	Risk Report	Risk Report
13.3 – Manage Stakeholder Engagement	Issue Log	Issue Log
9.5 – Manage Team	Issue Log	Issue Log
10.3 – Monitor Communications	Issue Log	Issue Log
11.7 – Monitor Risks	Risk Register	Risk Register
11.7 – Monitor Risks	Issue Log	Issue Log
11.7 – Monitor Risks	Risk Report	Risk Report
13.4 – Monitor Stakeholder Engagement	Risk Register	Risk Register
13.4 – Monitor Stakeholder Engagement	Issue Log	Issue Log
4.5 – Monitor and Control Project Work	Assumption Log	Assumption Log
4.5 – Monitor and Control Project Work	Risk Register	Risk Register
4.5 – Monitor and Control Project Work	Issue Log	Issue Log
4.5 – Monitor and Control Project Work	Risk Report	Risk Report
4.6 – Perform Integrated Change Control	Risk Report	Risk Report

11.3 – Perform Qualitative Risk Analysis	Assumption Log	Assumption Log
11.3 – Perform Qualitative Risk Analysis	Risk Register	Risk Register
12.1 – Plan Procurement Management	Risk Register	Risk Register
8.1 – Plan Quality Management	Assumption Log	Assumption Log
8.1 – Plan Quality Management	Risk Register	Risk Register
9.1 – Plan Resource Management	Risk Register	Risk Register
11.5 – Plan Risk Responses	Risk Register	Risk Register
11.5 – Plan Risk Responses	Risk Report	Risk Report
13.2 – Plan Stakeholder Engagement	Assumption Log	Assumption Log
13.2 – Plan Stakeholder Engagement	Risk Register	Risk Register
13.2 – Plan Stakeholder Engagement	Issue Log	Issue Log
6.3 – Sequence Activities	Assumption Log	Assumption Log

Appendix B

Impacted Process	Input Matched	Output Matched
5.2 – Collect Requirements	Assumption Log	Assumption Log
5.2 – Collect Requirements	Stakeholder Register	Stakeholder Register
5.4 – Create WBS	Requirements Documentation	Requirements Documentation
5.4 – Create WBS	Project Scope Statement	Project Scope Statement
5.3 – Define Scope	Assumption Log	Assumption Log
5.3 – Define Scope	Requirements Documentation	Requirements Documentation
6.5 – Develop Schedule	Assumption Log	Assumption Log
6.4 – Estimate Activity Durations	Assumption Log	Assumption Log
6.3 – Estimate Activity Resources	Assumption Log	Assumption Log
11.2 – Identify Risks	Assumption Log	Assumption Log
11.2 – Identify Risks	Requirements Documentation	Requirements Documentation
11.2 – Identify Risks	Stakeholder Register	Stakeholder Register
11.3 – Perform Qualitative Risk Analysis	Assumption Log	Assumption Log
11.3 – Perform Qualitative Risk Analysis	Stakeholder Register	Stakeholder Register
10.1 – Plan Communications Management	Requirements Documentation	Requirements Documentation
10.1 – Plan Communications Management	Stakeholder Register	Stakeholder Register
12.1 – Plan Procurement Management	Requirements Documentation	Requirements Documentation
12.1 – Plan Procurement Management	Stakeholder Register	Stakeholder Register
12.1 – Plan Procurement Management	Requirements Traceability Matrix	Requirements Traceability Matrix
8.1 – Plan Quality Management	Assumption Log	Assumption Log
8.1 – Plan Quality Management	Requirements Documentation	Requirements Documentation
8.1 – Plan Quality Management	Stakeholder Register	Stakeholder Register
8.1 – Plan Quality Management	Requirements Traceability Matrix	Requirements Traceability Matrix
9.1 – Plan Resource Management	Requirements Documentation	Requirements Documentation
9.1 – Plan Resource Management	Stakeholder Register	Stakeholder Register
11.1 – Plan Risk Management	Stakeholder Register	Stakeholder Register
11.5 – Plan Risk Responses	Stakeholder Register	Stakeholder Register
13.2 – Plan Stakeholder Engagement	Assumption Log	Assumption Log
13.2 – Plan Stakeholder Engagement	Stakeholder Register	Stakeholder Register
6.2 – Sequence Activities	Assumption Log	Assumption Log

Appendix C

Impacted Process	Input Matched	Output Matched
5.2 – Collect Requirements	Project Charter	Project Charter
5.2 – Collect Requirements	Assumption Log	Assumption Log
5.3 – Define Scope	Project Charter	Project Charter
5.3 – Define Scope	Assumption Log	Assumption Log
4.2 – Develop Project Management Plan	Project Charter	Project Charter
6.5 – Develop Schedule	Assumption Log	Assumption Log
6.4 – Estimate Activity Durations	Assumption Log	Assumption Log
6.3 – Estimate Activity Resources	Assumption Log	Assumption Log
11.2 – Identify Risks	Assumption Log	Assumption Log
11.3 – Perform Qualitative Risk Analysis	Assumption Log	Assumption Log
10.1 – Plan Communications Management	Project Charter	Project Charter
7.1 – Plan Cost Management	Project Charter	Project Charter
12.1 – Plan Procurement Management	Project Charter	Project Charter
8.1 – Plan Quality Management	Project Charter	Project Charter
8.1 – Plan Quality Management	Assumption Log	Assumption Log
9.1 – Plan Resource Management	Project Charter	Project Charter
11.1 – Plan Risk Management	Project Charter	Project Charter
6.1 – Plan Schedule Management	Project Charter	Project Charter
5.1 – Plan Scope Management	Project Charter	Project Charter
13.2 – Plan Stakeholder Engagement	Project Charter	Project Charter
13.2 – Plan Stakeholder Engagement	Assumption Log	Assumption Log
6.2 – Sequence Activities	Assumption Log	Assumption Log