Politecnico di Torino

Master’s Degree in Engineering and Management

A.Y. 2025/2026

Graduation Session March 2026

# Software Selection Process Design: Methodology and Application in a Corporate Context

Department of Management Engineering and Production

Academic Supervisor:

Prof. Marco Cantamessa

External Supervisors:

Chiara Pagnotta
Mauro De Maggi

Candidate:

Adriano Pella

# Software Selection Process Design: Methodology and Application in a Corporate Context


## Abstract

Organizations operating in complex and competitive environments frequently manage software acquisition through informal, fragmented practices that depend heavily on individual expertise rather than on shared governance frameworks. This reliance on informal approaches generates a range of operational and strategic risks: inconsistent requirement collection, ambiguous role allocation, late involvement of key technical and security functions, and limited traceability in purchasing decisions. This thesis addresses this gap by presenting the design and formalization of a standardized Software Selection Process developed within the context of a curricular internship at EY Advisory, for a multinational engineering consulting firm. The initiative was commissioned by the company's IT Architecture function, which had identified critical governance deficiencies in how software solutions were evaluated and procured across different business units.

The research adopts a qualitative methodology grounded in semi-structured stakeholder interviews and an iterative Agile framework, chosen to progressively capture operational knowledge that was not formally documented but embedded in daily practices. Starting from a diagnostic "As-Is" analysis, the work identifies four principal criticalities: absence of a governed workflow, informal and non-comparable requirement collection, unclear role boundaries, and the late engagement of IT Architecture and IT Security. These findings informed the design of a "To-Be" operating model structured across three macro-phases: Needs Identification and Initiation, Requirements Definition and Evaluation Criteria, and Market Analysis and Vendor Identification.

The primary deliverables produced are: a formalized BPMN process workflow comprising 31 activities and decision gates; a RACI matrix assigning explicit accountability across six organizational functions; a Business Requirement Template structured around thematic categories and a MoSCoW prioritization logic; a Decision Matrix grounded in a multi-criteria evaluation framework; and Operating Guidelines for structured vendor scouting. Each tool was designed to address a specific pain point identified during the diagnostic phase and was iteratively validated through stakeholder walkthroughs and joint review sessions.

The proposed framework has not yet been implemented end-to-end across live selection initiatives. Consequently, the benefits of the model, including improved governance, enhanced cross-functional coordination, and increased decision traceability, are grounded in the reference literature and supported by stakeholder validation during the design process but remain to be confirmed through quantitative empirical evidence. A set of KPIs, including Time to Selection, Requirements Coverage Rate, and Stakeholder Satisfaction, is proposed within the process documentation to enable future performance measurement once the model becomes fully operational. The thesis concludes by identifying two primary development directions: the digitalization of the workflow through implementation on a platform such as ServiceNow, and the integration of a dedicated evaluation module for AI-based software solutions. The methodological framework developed demonstrates sufficient generalizability to be adapted to software selection processes in different organizational contexts, representing a contribution to the intersection of Business Process Management and IT governance.

# LIST OF CONTENTS

# Introduction

The software selection constitutes a structured process aimed at identifying, evaluating, and choosing technological solutions that best fit an organization's operational and strategic needs. This activity, often underestimated in its complexity, requires the systematic analysis of business requirements, the definition of evaluation criteria, the scouting of potential vendors on the market, and the comparative assessment of available alternatives. The ultimate objective is to guarantee that the selected software represents the optimal choice from both a technical and economic standpoint, capable of supporting the enterprise's long-term goals. When executed without a formalized methodology, software selection frequently results in suboptimal decisions: tools that fail to integrate with existing systems, solutions that do not scale with organizational growth, or acquisitions driven by cost considerations alone rather than strategic alignment.

This thesis originates from my curricular internship experience at “EY Advisory SPA,” within the Technology Consulting division. During this engagement, I participated in a project commissioned by a multinational firm operating in the engineering consulting sector, tasked with designing and standardizing its software selection process. This work intends to offer a broader insight into the importance of formalizing industrial processes and its applicability: the methodological framework and operational tools developed during the project can be adapted to the selection of any enterprise software, from collaboration platforms to data analytics suites, from cybersecurity solutions to domain-specific applications.

In particular, the first chapter provides the theoretical foundation for the project. It analyses Business Process Management (BPM) as a discipline, illustrating its core principles and presenting examples of its most widely adopted modelling notations, including BPMN (Business Process Model and Notation). The chapter then examines the strategic role of the IT department within organizations, arguing that IT has evolved from a support function to a critical enabler of competitive advantage. Within this context, the importance of IT Strategy is discussed, with particular emphasis on how software selection decisions contribute to the alignment between technological investments and business objectives. The chapter concludes by positioning software selection as a strategic process that demands cross-functional governance.

In the second chapter is presented the case study to which I contributed during my internship. The company had identified inefficiencies and fragmentation in how its business units approached software acquisition. The chapter details the needs analysis conducted during the diagnostic phase, which revealed a fragmented "As-Is" state characterized by informal requirements gathering, ambiguous role definitions, and late engagement of the IT Architecture. Based on this analysis, a generic BPMN process was designed as a baseline reference model, intended to capture industry best practices before customization. The chapter also maps the business units and stakeholders involved in the process (Business, Business Partner, IT Architecture, Project Manager, IT Security, and Vendors) explaining their respective responsibilities and interdependencies.

The objective of the third chapter will be to focus on the methodology adopted to transform the generic process into an operational model tailored to the firm's specific context. The approach followed an Agile iterative framework, in which cycles of stakeholder interviews alternated with incremental deliverable development. The scheduling of interviews with operators from the various business units is described, together with the techniques used to understand requirements, validate assumptions, and collect feedback. This continuous loop of operativity and user validation ensured that the final process design reflected the actual organizational constraints and priorities rather than theoretical abstractions.

In the fourth chapter is presented the deliverables produced by the project and their critical evaluation. The main outputs include: a formalized BPMN workflow organized into three macro-phases (Needs Identification & Initiation, Requirements Definition & Evaluation Criteria, and Market Analysis & Vendor Identification); a RACI matrix that assigns clear accountability across all process actors; and a suite of operational instruments, specifically a Business Requirement Template, a Decision Matrix, and Operational Guidelines. The chapter evaluates the proposed model against the initial “pain points” identified during the diagnostic phase, assessing its effectiveness in enhancing governance, reducing cycle time, and ensuring strategic alignment.

Finally, critical evaluation and evolutions of the process are outlined, including the migration of the workflow to a digital platform to enable automation.

# 1. Literature Review

## 1.1 Business Process Management

Business Process Management (BPM) developed from the convergence of classical organizational theories in the early 1900s, when concepts like Adam Smith's division of labour and Frederick Taylor's scientific management laid the foundations for analysing activities as optimizable sequences. However, modern BPM tools and techniques only spread from the 1970s-80s, when process logic became pervasive, emphasizing interdependencies across organizational units, focusing on customer value creation in operations and quality. (*The History and Evolution of Business Process* Management *(BPM)*, 2024)

In today's context, marked by increasing business process complexity, needs of digitalization, and heightened competitiveness, BPM has emerged as an essential discipline to align operations with strategic goals, combining knowledge from information technology and management sciences applied to operational processes. (Bernardo, Galina and Pádua, 2017)

It represents a set of methodologies, techniques, and tools aimed at modelling, executing, monitoring, and optimizing business processes in compliance with constraints related to efficiency, costs, and service levels. In this perspective, it extends the concept of Workflow Management (WFM), which is primarily concerned with the automation and coordination of task execution, by encompassing a broader scope that includes process analysis, operations management, and the work organization. BPM not only facilitates cost reductions and productivity gains through generic BPM systems driven by explicit designs but is crucial for enhancing adaptability and stakeholder collaboration, enabling improvements even without new technologies, such as modelling and simulation. (Van Der Aalst, 2013)

The importance of BPM is further strengthened by the evolution of methodologies providing structured frameworks for process management, adapting to needs in areas like IT governance and digital transformation. It starts from traditional WFM's mechanistic automation with limited attention to human factors, shifting to modern approaches like the BPM lifecycle, design, and optimization, in an iterative loop for continuous improvement, particularly popular in software-enabled process development. Furthermore, it focuses on short, incremental analysis and adaptation phases, deriving from quality management and dynamic capabilities principles to minimize waste and maximize customer value, making it ideal for standardizing strategic decisions like software selection.

### 1.1.1 Core Principles and Objectives of BPM

Business Process Management is based on a set of core principles that sets it apart from traditional function-oriented approaches and enables an integrated, end-to-end management of organizational processes. The main principles of BPM can be summarized as follows (Dumas *et al.*, 2018, chs 1–2) :

- End-to-end process orientation: BPM focuses on managing processes as integrated value streams that span across organizational functions, rather than as isolated activities. This perspective enables coordination across business units.
- Process ownership and governance: BPM introduces explicit process ownership, assigning responsibility for planning, organizing, and monitoring processes to designated process owners. This principle supports accountability across functional boundaries and enables effective governance mechanisms within complex organizational structures.
- Performance measurability: the systematic measurement of process performance through indicators such as cost, time, quality, and flexibility. Measurability provides an objective basis for monitoring execution, identifying inefficiencies, and supporting data-driven decision-making.
- Continuous improvement: BPM promotes an iterative lifecycle approach in which processes are continuously analysed, redesigned, implemented, monitored, and refined. This ensures that processes are treated as dynamic assets capable of evolving in response to organizational and environmental changes.
- Strategic alignment: BPM ensures alignment between organizational strategy and operational execution by deriving process objectives and

performance metrics directly from strategic goals, thereby reinforcing the role of processes as instruments for strategy implementation.

The objectives of BPM are primarily oriented toward enhancing organizational performance and long-term competitiveness. By applying its principles in a systematic and continuous manner, BPM aims to improve operational efficiency through waste reduction and standardization, increase customer satisfaction by optimizing end-to-end value creation, and support innovation by enabling organizations to adapt their processes to changing internal and external conditions. Treating processes as strategic assets allows BPM to function as a dynamic capability that sustains alignment between strategy and execution, offering a structured and effective alternative to fragmented, or episodic management approaches.

### 1.1.2 The BPM Lifecycle

A successful BPM approach is promoted through a lifecycle made of six phases, beginning with planning and strategy and proceeding through analysis of business processes, design and process implementation, monitoring, and control and, finally, refining. A representation of BPM lifecycle is provided in Figure 1: BPM Lifecycle By understanding, documenting, modelling and analysing business processes, organizations can improve visibility and transparency and reduce costs and resource requirements, resulting in improved business performance and compliance. The BPM lifecycle therefore provides a structured articulation of BPM activities across time, linking the initial planning dimension to subsequent execution, control, and improvement stages. In addition, BPM is framed as an enabler of dynamic and flexible collaborations that support organizational adaptation to changing global market conditions.

Specifically, the lifecycle comprises the following phases (Bernardo, Galina and Pádua, 2017):

- Planning and strategy: Establish a strategy oriented toward business processes and develop a plan to direct BPM actions.

- Analysis of business processes: Use different methodologies to understand the current alignment of organizational processes with the stipulated objectives and goals.
- Design: Design the new process and its specifications; strive to adjust the specifications within a model that best contributes to the objectives established in the plan based on the status.
- Process implementation: Implement the adjusted model; this stage includes the challenges of change management and process optimization.
- Control and monitoring: Contrast the achieved results with the planned goals and offer suggestions for decision making by managers and continuous improvement.
- Refining: Adjust and improvements to contribute more effectively to cycle feedback.

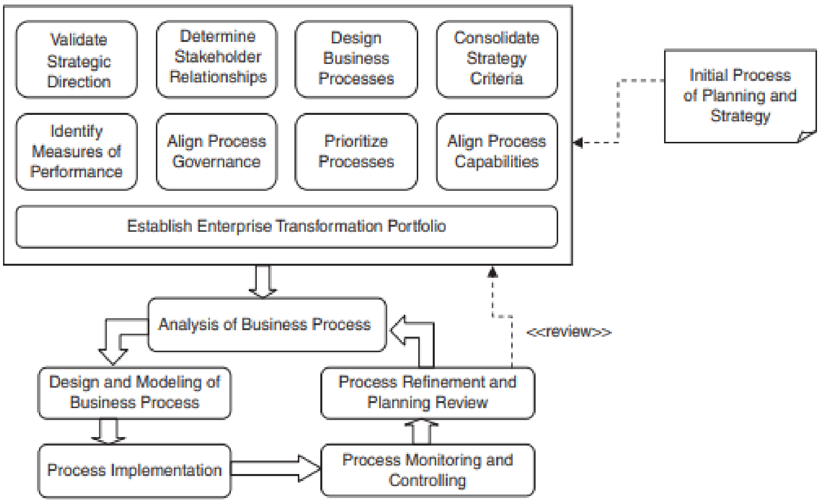


*Figure 1: BPM Lifecycle*

## 1.2 Process Modelling and BPMN Notation

Process modelling constitutes a fundamental component of BPM, furnishing a structured graphical depiction of organizational workflows that renders processes explicit and comprehensible to both business and IT stakeholders (Dumas *et al.*, 2018, ch. 3). Its principal objectives encompass delineating “As-Is” operations, devising “To-Be” configurations, facilitating performance and risk analysis, and establishing a common foundation for automation and governance. Through selective abstraction of irrelevant details and emphasis on pertinent activities, events, decisions, and responsibilities, process models promote perspective alignment for process enhancement.

Different notations can be used to represent business processes, reflecting distinct levels of abstraction and supporting different activities within the BPM lifecycle. A process model represents organizational reality through a selective representation that combines mapping and abstraction in relation to a defined purpose. In this perspective, process modelling can be positioned along two main uses: conceptual models adopted for organizational design and communication, and executable models adopted for application system design and process automation.(Dumas *et al.*, 2018, ch. 3)

- Business Process Model and Notation (BPMN): a standard for business process modelling that provides a graphical notation for specifying processes in Business Process Diagrams, designed to be understandable by business and technical stakeholders while being able to represent complex process semantics.
- Unified Modelling Language (UML) Diagrams: a UML notation used to represent activity flows and control logic, typically adopted to describe procedural behaviour in a form that can be aligned with software-oriented representations.
- Value Stream Mapping (VSM): a lean technique used to visualize end-to-end flows with emphasis on value creation and waste identification, supporting process improvement by highlighting inefficiencies along the stream.
- Swimlane Diagrams: a flowchart-based representation that partitions activities into lanes to indicate responsibility boundaries, supporting role/organizational handoff analysis at a descriptive level.

Within this context, Business Process Model and Notation (BPMN) has emerged as the "de-facto" standard for business process modelling, providing a common graphical language for specifying processes in Business Process Diagrams. BPMN was initially developed by the Business Process Management Initiative and its first version was released in 2004; it has since been maintained and evolved by the Object Management Group, which define both notation and execution semantics (Von Rosing *et al.*, 2015). The core idea of BPMN is to offer a notation that is intuitive for business users yet expressive enough to represent complex process behaviour, bridging the gap between high-level conceptual models and executable process definitions. An illustrative example of a BPMN-based workflow is provided in Figure 2: BPMN-based workflow

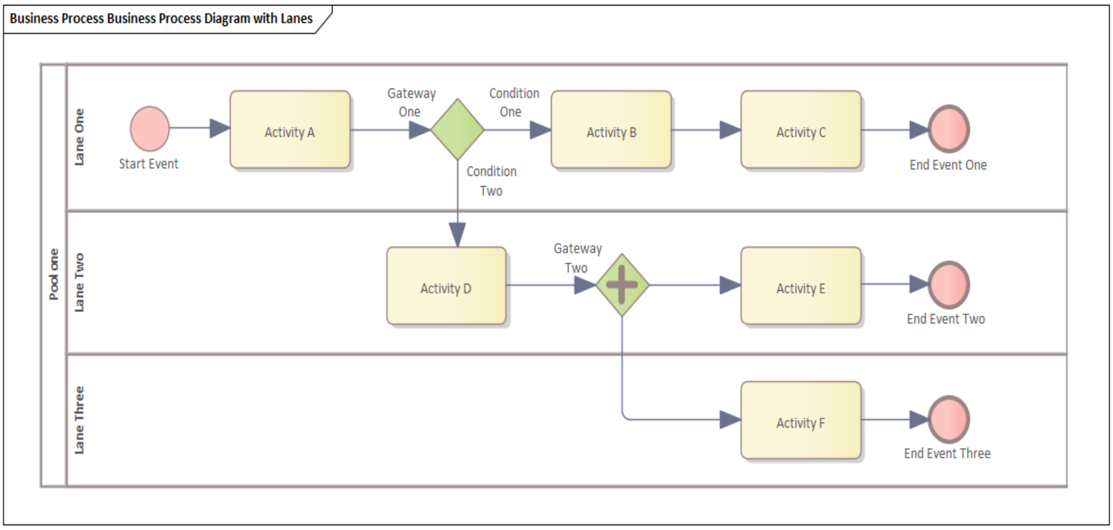


*Figure 2: BPMN-based workflow*

### 1.2.1 Key BPMN Elements

BPMN represents business processes through a limited set of graphical elements, each associated with a specific behavioural meaning and a corresponding symbol. The principal BPMN elements are illustrated in Figure 3: Key BPMN elements**Errore. L'origine riferimento non è stata trovata.**. The notation distinguishes between elements that determine the process logic (flow objects), elements that connect them (connecting objects), and complementary elements used to structure responsibilities and enrich the model with contextual information (swimlanes and artifacts).(Von Rosing *et al.*, 2015)

The following delineates these components.('Understanding the Core Elements and Key Concepts of BPMN', 2024)

Flow Objects

Flow objects define the core behaviour of a process model by representing occurrences, work, and routing logic.

- Events (circles): Represent things that happen during the process; BPMN distinguishes start events (thin border) and end events (thick border), which indicate when instances begin and complete. Intermediate and boundary events represent occurrences that happen between start and end and can be placed in the flow or attached to an activity boundary.
- Activities (rounded rectangles): Represent units of work; BPMN includes tasks and sub-processes, where a sub-process can be decomposed into a lower-level process. A call activity is a specific activity type used to invoke reusable (global) tasks or processes.
- Gateways (diamonds): Control divergence and convergence of sequence flow within a process. Typical gateway behaviours include exclusive routing (XOR), parallel routing and synchronization (AND), and inclusive multi-branch routing (OR), each used to structure decisions and concurrency.

Connecting Objects

Connecting objects specify how flow objects relate to each other in terms of control and interaction.

- Sequence Flow (solid line with solid arrowhead): Shows the order in which activities are performed within a process.
- Message Flow (dashed line with open arrowhead): Shows message exchange between different participants (pools) in a collaboration.

- Association (dotted line): Links text and artifacts to flow objects to add contextual information.

Swimlanes and Artifacts

Swimlanes structure the diagram according to participants and responsibility boundaries, while artifacts add information about data and explanatory notes.

- Pool / Lane: Pools are used to represent participants (resource classes), while lanes partition a pool into roles, units, or systems for responsibility allocation.
- Data objects and data stores: Data objects represent information or material used or produced by activities, while data stores represent persistent repositories (e.g., a database) beyond the lifetime of a single instance.
- Annotations: Text annotations provide additional explanatory information and are connected via associations; they do not alter token flow semantics.

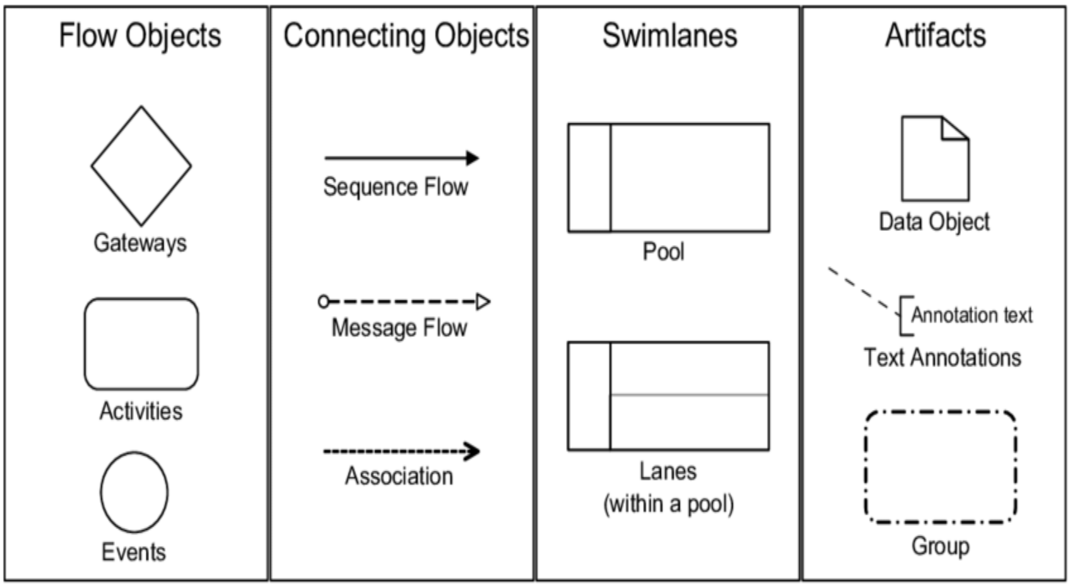


*Figure 3: Key BPMN elements*

## 1.3 Strategic Role of the IT

The centrality of Information Technology (IT) is associated with the increasing dependence of core business processes on digital platforms and with the need to govern data, applications, and infrastructure in an integrated manner. In the literature, IT is described as enabling automation, information processing, and business transformation, linking its operational role to broader organizational performance and change.(Rahimi, Møller and Hvam, 2016) IT strategy is defined as a set of choices that aligns technological capabilities with business objectives, covering architectures, applications, data, and competencies. The interdependence between processes and IT systems is frequently addressed through a process-oriented view, in which business processes connect business strategy to IT capabilities and, conversely, process designs can be translated into technical specifications informing system selection, configuration, and integration. In parallel, BPM is described as supporting digital innovation and digital transformation by helping organizations identify, optimize, and automate relevant processes and by facilitating collaboration among stakeholders.(Putra and Er, 2024)

### 1.3.1 IT as Organizational Function

Within a corporate context, the IT department is responsible for managing the technological resources required to support enterprise operations, including digital infrastructures, information systems, and software applications. Its primary mandate is to ensure the reliability, continuity, and coherence of the technological landscape through structured decision-making and management practices that govern IT assets and their evolution over time. Although traditionally associated with operational support, the IT function cannot be considered isolated from business process concerns. Information systems and business processes are structurally interdependent and co-evolve in the delivery of products and services, making IT an integral component of organizational functioning rather than a purely technical support unit.

### 1.3.2 IT Strategy in BPM-Business Alignment

IT also functions as a coordination mechanism that enables alignment between BPM and business strategy, as business processes represent the connection between strategic intent and the organization's IT capabilities and information requirements. In this context, alignment is operationalized through two complementary governance paradigms (Rahimi, Møller and Hvam, 2016):

- IT Governance (ITG), defined as the framework specifying decision rights and responsibilities for IT-related choices, including structures as steering committees and processes such as portfolio management, with the aim of aligning IT investments with business strategy.
- Business Process Governance (BPG), which establishes accountability and decision-making for BPM activities, covering process design, execution, and monitoring through clearly defined roles, such as process owners and governance bodies.

The literature emphasizes the importance of reciprocal integration between ITG and BPG, particularly in contexts where IT acts as a business enabler or strategic driver. This integration involves alignment at the strategic level and continuous interaction at the operational level. Mechanisms such as liaison roles, whereby process owners participate in IT decision-making bodies, ensure that BPM requirements inform IT investments, while IT specialists contribute to process redesign through the adoption of system-embedded best practices. Empirical evidence from mature organizations shows that the harmonization of ITG and BPG enables the prioritization of IT-enabled process innovations and supports the creation of measurable business value.

### 1.3.3 Agile approach in BPM Optimization

Agile methodology originates from software development, where it promotes short, iterative cycles, continuous stakeholder feedback, and rapid adaptation to change, in contrast to linear and plan-driven models. When transposed into organizational contexts, Agile shifts the focus from exhaustive upfront design to incremental value delivery, emphasizing collaboration, responsiveness, and continuous learning as key drivers of performance improvement. Within BPM, this perspective implies treating processes as dynamic systems subject to frequent inspection and adaptation along the BPM lifecycle.

The adoption of Agile principles in BPM, often referred to as Agile BPM, introduces iterative loops of analysis, design, implementation, and review that can be executed within short, time-boxed cycles. This approach supports continuous alignment between business processes, IT strategy, and digital innovation, as it enables organizations to experiment with limited-scope process changes, validate them rapidly through performance indicators and stakeholder feedback, and progressively consolidate only those solutions that generate measurable value.

Within this framework, prioritization plays a significant role in ensuring that process optimization efforts remain focused on the most impactful improvements. Agile BPM requires mechanisms that help distinguish essential process capabilities from secondary enhancements, avoiding scope expansion and unnecessary complexity. Among the tools supporting this objective, the “MoSCoW” prioritization technique (Must-have, Should-have, Could-have, Won’t-have) is widely adopted to structure decision-making and guide iterative development. The MoSCoW method enables process requirements, capabilities, and improvement actions to be classified according to their criticality for business outcomes.

Applied to BPM initiatives, MoSCoW facilitates process optimization by enabling teams to concentrate early iterations on critical process requirements that directly address major inefficiencies, governance gaps, or compliance constraints. By explicitly prioritizing improvements according to their business relevance, organizations can deliver simplified yet effective process versions in shorter timeframes, while maintaining transparency on trade-offs and deferring non-essential elements to subsequent cycles. This targeted focus reinforces the Agile principle of delivering value incrementally and supports a disciplined approach to continuous process improvement consistent with BPM governance objectives.

### 1.3.4 IT Strategy as a Bridge to Operational Software Decisions

IT strategy can be understood as the set of choices through which organizational direction and process priorities are translated into concrete guidelines for technological architectures, applications, and investment decisions. Rather than operating at a conceptual level, IT strategy plays a practical role by shaping how information systems are designed and used to effectively support business processes and organizational objectives. In this context, business process designs developed through BPM initiatives play a critical role, as they provide a structured representation of business requirements that can be translated into technical specifications for system selection, configuration, and integration. Consequently, IT strategy functions as a critical link between strategic view and software-related decisions, helping ensure that selected applications are consistent with both process structures and long-term organizational goals.

## 1.4 Software Selection as a Strategic Process

Software selection can be framed as a strategic and structured decision-making process, as inappropriate software choices may generate significant economic losses, constrain organizational flexibility, and negatively affect business processes. For this reason, the selection of software solutions cannot be reduced to a purely technical or procurement-driven activity but must be understood as a decision with long-term strategic implications. The literature commonly describes software selection as a multi-criteria decision-making problem, in which decision makers are required to evaluate and compare alternative solutions characterized by multiple and often conflicting attributes.(Jadhav and Sonar, 2009) Rather than identifying a universally optimal solution, recent studies emphasize the importance of selecting software whose suitability depends on the objectives for which it is adopted, the organizational context, and the intended use. This shift is particularly evident in domain such as simulation software (demo), where usability and decision-support capabilities vary according to managerial and operational needs, and where both technical experts and managers are involved in the decision process.(Fumagalli *et al.*, 2019) Within the BPM domain, software selection assumes an additional strategic relevance. BPM is positioned as a paradigm for operational excellence and organizational transformation, while BPM software (BPMS) is regarded as an enabling technology for implementing flexible, integrated, and adaptive business processes in environments shaped by digital transformation. Consequently, BPMS selection reflects broader transformation needs and must be aligned with process designs, governance mechanisms, and long-term organizational objectives.(Brkić, Tomičić Pupek, Bosilj Vukšić, 2020)

To address this complexity, the literature highlights the role of structured methodologies, such as the Analytic Hierarchy Process (AHP), which support decision makers in decomposing the selection problem into hierarchies of criteria and alternatives. These approaches allow the integration of qualitative and quantitative evaluations and produce explicit priorities and rankings. At the same time, research emphasizes that the definition of evaluation criteria is itself a strategic activity, as the absence of shared and well-defined criteria can introduce ambiguity and undermine decision quality.(Jadhav and Sonar, 2009) Beyond scoring mechanisms, robust software selection processes incorporate additional dimensions, such as vendor-related evaluation and empirical validation. Vendor involvement represents a critical aspect of the decision, as suppliers may introduce bias by promoting their own solutions, while vendor maturity, experience, and support capabilities directly affect implementation sustainability. Furthermore, pilot testing and trial use of shortlisted solutions are identified as essential stages of the selection process, particularly in contexts where usability and process fit can only be assessed through practical experimentation.

### 1.4.1 Cross-Functional Decision-Making

Strategic software selection requires inputs that cannot be confined to a single functional or technical perspective. Evaluation criteria typically span multiple dimensions, including functional fit, cost–benefit considerations, hardware and software constraints, integration requirements, and vendor-related aspects such as training, documentation, and long-term support. As a result, effective software selection depends on the involvement of multiple organizational actors and the coordination of diverse viewpoints.

In the context of BPM software selection, this multi-criteria nature is operationalized by prioritizing different BPMS dimensions, such as capabilities, compatibility, complexity, vendor maturity, and cost structures, and by relating them to broader digital transformation drivers. This perspective highlights that technological and organizational priorities must be addressed jointly, rather than sequentially or in isolation.(Brkić, Tomičić Pupek, Bosilj Vukšić, 2020) Vendor-related evaluation further reinforces the need for cross-functional decision-making. Factors such as vendor references, local presence, documentation quality, and maintenance capabilities affect not only the acquisition phase but also long-term implementation success. Consequently, vendor assessment represents an integral component of strategic software selection, requiring coordinated involvement from business, IT, and supporting functions.

### 1.4.2 Risks of Unstructured Software Selection Processes

The absence of a formalized and structured methodology for software selection exposes organizations to a wide range of risks that directly compromise decision quality, increase economic costs, and undermine long-term strategic alignment between technology and business objectives. In the literature, unstructured software selection processes are typically characterized by the lack of explicit stages, undefined or inconsistently interpreted evaluation criteria, unclear allocation of stakeholder responsibilities, and the absence of governance mechanisms capable of coordinating decision-making activities. One of the most evident and frequently cited risks concerns economic loss. Selecting an inappropriate software solution may lead not only to suboptimal operational performance but also to incorrect strategic decisions with long-lasting financial consequences. This risk is particularly significant in the case of large-scale enterprise systems, such as ERP solutions, whose acquisition and implementation costs can amount to hundreds of thousands or even millions of dollars. As highlighted by literature, investing substantial portions of capital budgets in unsuitable software packages often results in sunk costs that fail to deliver the expected organizational value, while simultaneously constraining future investment options(Jadhav and Sonar, 2009).

Beyond direct financial implications, unstructured selection processes severely affect decision-making quality. Vendor selection is widely recognized in the literature as a complex and unstructured multi-criteria decision-making problem, involving the evaluation of heterogeneous and often conflicting criteria. In the absence of a defined process, evaluation activities tend to become arbitrary, fragmented, and highly susceptible to bias. Uncontrolled interactions with vendors, in particular, increase the risk of self-promotional influence, whereby suppliers emphasize selective aspects of their offerings without being assessed against a shared and transparent evaluation framework(Niknamfar and Niaki, 2018). Moreover, when evaluation criteria are not clearly defined and consistently interpreted, their meaning remains open to individual judgment, generating ambiguity and reducing the comparability of alternatives. A further category of risk relates to vendor dependency and outsourcing vulnerabilities. Poor vendor selection has been identified as a critical positional risk in outsourcing decisions, alongside issues such as data security and loss of organizational knowledge. Inappropriate vendor choices may result in implementation failures, excessive customization efforts, lock-in effects, or escalating maintenance and support costs over time. When these aspects are not systematically assessed within a structured process, organizations expose themselves to long-term dependency risks that are difficult to reverse.

Unstructured selection processes also exacerbate organizational and coordination risks, especially in contexts characterized by strong cross-functional interdependencies. Without explicit stakeholder mapping and formal assignment of roles and responsibilities, business units, IT architecture, security, and procurement functions tend to operate in relative silos. This fragmentation often leads to conflicting requirements, misaligned expectations, and iterative rework, as each function applies its own criteria and priorities without a shared governance framework(Brkić, Tomičić Pupek, Bosilj Vukšić, 2020). The resulting misalignment not only delays decision-making but also undermines trust and collaboration among stakeholders.

### 1.4.3 Advantages of Strategic Software Selection

Compared to ad-hoc or purely technology-driven approaches, strategic software selection offers several advantages.

First, it is characterized by a structured and transparent methodology shaped by well-defined stages, including need identification, shortlisting, elimination of incompatible options, evaluation and ranking, empirical testing, negotiation, and implementation. This structured progression reduces arbitrariness and supports informed decision-making.

Second, strategic approaches improve alignment between software choices and organizational objectives. The literature on simulation software selection, for example, emphasizes that technological evolution has shifted attention from the intrinsic features of tools to their suitability for specific decision-making purposes(Fumagalli *et al.*, 2019). This focus strengthens the link between software selection and the intended use of the system, ensuring coherence between tools, processes, and managerial objectives.

Third, the adoption of multi-criteria decision-making methods enhances evaluation quality. By framing software selection as a multi-criteria decision method (MCDM) problem, approaches such as AHP enable the systematic weighting and comparison of heterogeneous criteria, integrating qualitative judgments with quantitative assessments and reducing the risk of biased or implicit trade-offs.

Finally, strategic software selection contributes to risk mitigation through explicit vendor evaluation and empirical validation. Vendor selection is identified as a sensitive decision, as inappropriate vendor choices may lead to implementation failures, dependency risks, or increased costs over time.(Niknamfar and Niaki, 2018) The inclusion of vendor-related criteria, together with trial use and pilot testing of shortlisted solutions, supports a more robust assessment of practical usability and organizational fit.

Overall, these advantages position strategic software selection as a mechanism to reduce ambiguity, manage trade-offs across technological and organizational dimensions, and support long-term business objectives through repeatable and governance-aware decision processes.

# 2. Case Study: “Software Selection Process Design”

## 2.1 Project Context and Background

The case study presented in this chapter was developed during my curricular internship at EY, within the Technology Consulting service line. The project, concerning the design and standardization of a software selection process, was commissioned by a multinational company operating in the engineering consulting and ICT services sector, with a consolidated international presence. The initiative originated from an explicit request by the IT Architecture department, which identified the need to review and optimize the way different business units managed the acquisition and selection of new software solutions, with the strategic objective of pursuing “better technological infrastructures for operational excellence.”

Within this context, the company requested the formalization of the software selection process and the definition of a structured workflow aligned with national and international best practices, to be adopted as a single reference framework across business units. The request also included the development of operational support tools, such as process guidelines, templates, and checklists, aimed at facilitating coordination among stakeholders and reducing dependency on individual expertise. These objectives can be summarized in three main pillars, as illustrated in Figure 4: Project Objectives. The objective of the intervention was therefore not to identify “the” correct software for a specific application context, but rather to design a standardized and repeatable process capable of making decision-making steps explicit, improving coordination among stakeholders, and reducing the ambiguity typically associated with technology-related decisions managed through informal practices. This perspective implies interpreting software selection as a cross-functional governance process, with organizational and architectural implications, rather than as a sequence of purely technical activities or a procedure limited to procurement. Consistently with this approach, the project deliverables were conceived to transform a set of fragmented practices into a formalized and traceable workflow, in which roles, responsibilities, dependencies, and evaluation criteria are clearly defined and replicable. In response to the client's request, EY proposed a structured project plan articulated into three main phases:

I. Identification of stakeholders involved in the software selection process and execution of interviews to tailor the process to the company's specific context.
II. Definition of the process design, roles, and responsibilities through iterative feedback and fine-tuning (progressive refinement across multiple phases).
III. Classification of project drivers and thresholds to address project-specific characteristics, such as strategic impact, global versus local scope, and standardization requirements.

The project was conducted by a consulting team composed of myself, in the role of consultant, and two managers who provided supervision and strategic guidance throughout the engagement. In the present chapter, the work is presented to motivate the structure of the case study and the rationale underlying the baseline model. The operationalization of this approach and its iterative execution are discussed in Chapter 3.


*Figure 4: Project Objectives*

### 2.1.1 EY Consulting

EY Technology Consulting, which conducted the project for the client company, is EY's service line specialized in strategic, technological, and digital transformation consulting. It provides services and solutions to align corporate organization, processes, and IT systems with business strategy, defining and supporting improvement actions on processes, governance, IT architectures, and technological implementations. Composed of professionals with experience in Management Consulting and Information Technology, EY Technology Consulting plays the role of integration between functional and technological aspects, offering solutions that combine strategic vision, process knowledge, and digital innovation.

### 2.1.2 Client Company

The client company is a multinational group with a strong international presence, operating across 70 countries and serving a wide range of industrial sectors and communities. The organization provides advanced ICT services and engineering consulting, supporting clients in addressing complex technological, regulatory, and operational challenges.

As part of its ongoing commitment to technological excellence and continuous improvement, the company launched a strategic initiative aimed at reviewing and optimizing its software selection process. This initiative was driven by the need to ensure greater consistency, transparency, and alignment with market best practices, while also supporting the organization’s evolving digital and organizational requirements. The objective was to strengthen governance mechanisms and enhance decision-making effectiveness in the selection of software solutions within a complex, multinational context.

### 2.1.3 Project Plan

Through the analysis of client needs, we established a project plan lasting 3 weeks, as detailed in Figure 5: Project plan. This timeline guided the entire software selection process project execution, setting operational phase guidelines and ensuring adequate project progression. The project was divided into three phases.

Phase 1- Set-up and Onboarding: focused on stakeholder identification and engagement through interviews with key business units, gathering insights on current practices and expectations to tailor the process design.

Phase 2- Process Design: involved creating the BPMN workflow based on IT best practices, developing a RACI matrix for roles and responsibilities, and iterating through feedback collection and fine-tuning sessions.

Phase 3- Guidelines and Templates: creation of operational tools to support the process, including the Decision Matrix, Business Requirement Template, and Operating Guidelines. Additionally, definition of drivers and thresholds for process variations (as strategic impact or global versus local applicability) to guide consistent decision-making across initiatives.

This phased approach, executed over the project duration, ensured that the final software selection process was not only theoretically feasible but practically aligned with the client's organizational context and governance requirements.

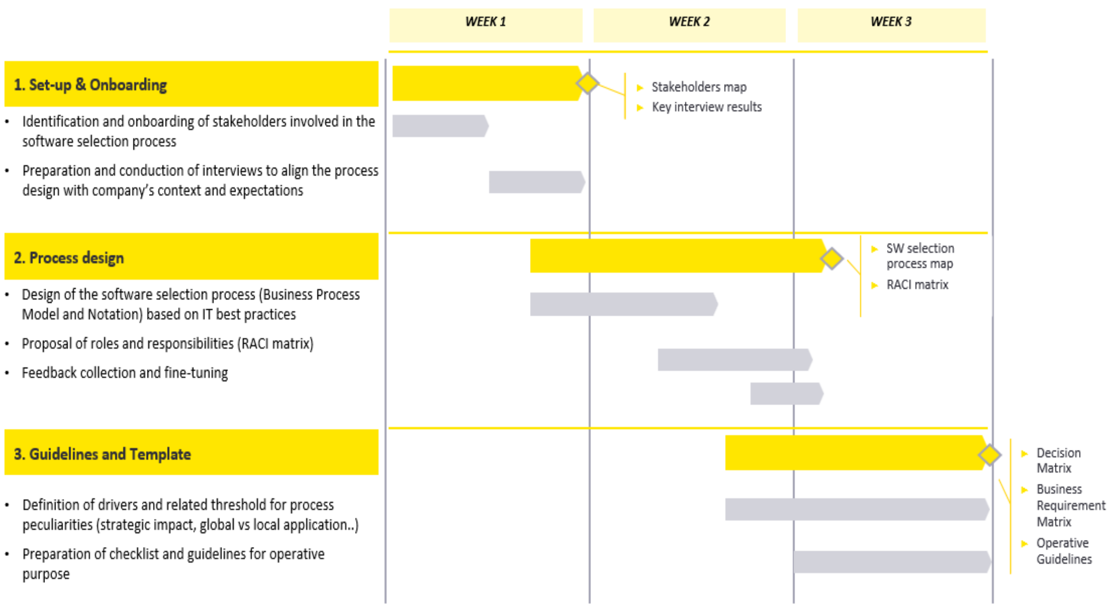


*Figure 5: Project plan*

## 2.2 Identification of Business Needs and “As-Is” Analysis

### 2.2.1 Initial Business Needs

The initial interactions with the IT Architecture function highlighted the need to formalize a shared and structured approach for managing software requests originating from the Business. In the absence of a common model, even conceptually recurring activities, such as requirements collection, technical validation, and market analysis, were performed with varying levels of depth and criteria, making it difficult to ensure consistency and comparability across initiatives.

Within this context, the client expressed needs that can be grouped into four principal areas.

The first need concerns the definition of a clear and repeatable workflow that guides the process from the initial request to the creation of a vendor shortlist, maintaining the selection within a governed perimeter before the formal engagement of procurement activities.

The second need relates to stakeholder coordination, with particular attention to the interface between business and IT. During the discussions, the significant role of the Business Partner emerged as a key element to ensure a structured and continuous translation of business demand into requirements that can be effectively evaluated.

The third area focuses on reducing decision-making ambiguities and role overlaps. Over time, unclear responsibilities were found to generate bottlenecks, duplicated efforts, and delays, with negative effects on both decision quality and process efficiency.

Finally, the client highlighted the need to anticipate the involvement of technical governance and risk control functions, such as IT Architecture and IT Security. Late validations were considered critical, as they often occur when the process is already advanced and corrective actions become more complex and costly.

### 2.2.2 Process Criticalities ("As-Is" Analysis)

To better understand the current situation, individual interviews were conducted with representatives of the main involved functions, with the objective of mapping the existing software selection process and identifying its main weaknesses. This activity made it possible not only to clarify the responsibilities of each function, but also to highlight the areas that currently generate the most significant criticalities.

The "As-Is" analysis revealed a process characterized by fragmentation and informality, which makes it difficult to reconstruct "ex post" the rationale, criteria, and responsibilities behind past decisions. A first critical issue concerns the informal collection of requirements, often performed without a shared framework. In this context, requirements may be incomplete, not verifiable, or not comparable, weakening the quality of evaluations and limiting the robustness of comparisons between alternative solutions.

A second criticality relates to the lack of common templates and checklists to support specific activities, such as market screening and vendor evaluation. When scouting activities rely on individual experience or non-standardized sources, the risk of bias increases and the repeatability of the selection process across initiatives is reduced.

A third issue concerns the late involvement of key governance functions. The evidence collected indicates that demos and proof-of-concept activities were often requested directly by the business and managed in an unstructured way, while IT Security and other control functions were involved inconsistently. This increases the likelihood of rework, as solutions that appear functionally adequate may later prove misaligned with architectural constraints or security requirements.

Finally, the current process suffered from limited decision traceability. Without a formalized process and a minimum set of standardized outputs, it becomes difficult to ensure auditability, support organizational learning, and clearly justify why a specific solution was selected over available alternatives.

"Lightweight" approaches based on informal guidelines were therefore considered inadequate, as they do not enforce decision gates, do not require minimum information standards and do not ensure the systematic production of comparable evidence. The value of a formalized process lies in its ability to make complex and multi-dimensional decisions repeatable and governed, transforming them from discretionary choices into a structured and transparent decision-making path.

## 2.3 Stakeholder Mapping and Organizational Scope

### 2.3.1 Stakeholder Identification

The stakeholder mapping activity was conducted immediately after project approval and represented the first structured step of the project approach. Stakeholders were identified through a series of structured interviews with different business units, with the objective of understanding how software selection initiatives were initiated, managed, and coordinated in practice. This activity was necessary to design a process tailored to the company's specific organizational and governance context, as well as to identify all business unit functions in the various stages of the software selection lifecycle. The interviews highlighted that software selection cannot be attributed to a single organizational function but instead requires the coordinated contribution of multiple actors with complementary responsibilities. In this sense, the software selection team emerges as inherently cross-functional and multidisciplinary, combining internal roles from different business and IT areas with external actors, whose involvement contributes specialized knowledge and an unbiased perspective on the solutions under evaluation.

Based on the evidence collected, the main stakeholders involved in the process were identified as Business, Business Partner, IT Architecture, Project Manager, IT Security, Vendors and Procurement, with the latter involved after the definition of the vendor shortlist.

Beyond identifying the involved roles, this mapping activity served a broader governance purpose. By formalizing which business units and functions participate in the process and how their contributions intersect, it delineated the organizational scope of software selection and provided the foundation for a customized process design. Furthermore, the stakeholder identification phase also supported the creation of organizational alignment around the initiative. Making roles and interactions explicit from the outset facilitated stakeholder engagement and awareness of the process objectives, a necessary condition to foster commitment and ensure that the redesigned process would be effectively adopted. This clarified that an effective solution could not rely on informal coordination practices but required structured interactions and clearly identified responsibilities to ensure consistency, traceability, and repeatability across the organization.

### 2.3.2 Roles, Responsibilities, and Interdependencies

Following the stakeholder mapping activity, the project team detailed the contribution of each role to the software selection process and clarified how these contributions interact throughout the process lifecycle. In this context, the specific roles and responsibilities assigned to each stakeholder involved in the process are outlined below.

Business: defines the functional needs and strategic goals the software must support. It provides use cases, prioritizes features, and validates whether proposed solutions align with operational objectives. Its input ensures the selected software delivers real value to end users and stakeholders.

Business Partner: acts as the primary interface between business and IT, capturing business needs and translating them into requirements that can be effectively assessed from a technical perspective. The Business Partner coordinates the evaluation process, supports timeline management, and ensures alignment with the overall IT strategy. By consolidating and validating requirements and serving as a single point of contact for involved stakeholders, this role helps balance business demand with available IT capabilities and reduces the risk of misalignment and rework during subsequent phases.

IT Architecture: ensures that software choices align with the current or target enterprise architecture and IT landscape. It evaluates integration capabilities, scalability, and adherence to architectural standards, with the objective of ensuring technical compatibility and long-term sustainability. Throughout the process, IT Architecture leads the main technical checkpoints and provides the technical framework that informs and guides the evaluations and decisions of the other stakeholders.

Project Manager: supports the overall process by focusing on operational execution and the management of interactions with potential vendors. This role coordinates preparatory and logistical activities, serves as the main point of contact for vendors to manage requests for clarification, and oversees the planning and organization of demos and workshops. In addition, they function as a bridge between the software selection process and the Procurement, to activate the purchase activities.

IT Security: assesses the security posture of each software solution, including data protection, access control, regulatory compliance, and potential risks. Its involvement is structured through predefined checkpoints to ensure security requirements are considered throughout the selection process.

Vendors: provide detailed information about their solutions, respond to requests for information (RFI), and participate in demos or proof-of-concept sessions. Their responsiveness and transparency contribute directly to the evaluation process.

Procurement: becomes involved once a vendor shortlist has been defined, managing qualification, registration checks, and subsequent negotiation and contracting activities. Although primarily engaged in the final stages, Procurement relies on the structured outputs of the selection process to ensure alignment with technical and business assessments.

These roles are interconnected through a network of interdependencies that governs information flow and decision-making across the process. For example, the quality of Business and Business Partner inputs directly affects IT Architecture's ability to conduct technical evaluations. In turn, IT Architecture provides the technical framing that guides IT Security assessments and supports Project Manager activities. Vendor interactions feed into these evaluations, ensuring decisions are based on complete and verified information.

Overall, the effectiveness of the software selection process depends less on isolated tasks and more on the orchestration of cross-functional collaboration. The RACI matrix, presented in Chapter 4, formalizes these relationships by translating the stakeholder network into an explicit accountability framework.

## 2.4 Definition of a Baseline Software Selection Process

### 2.4.1 Rationale for a Baseline BPMN Model

Following the findings of the "As-Is" analysis, a generic BPMN process was designed to serve as a baseline reference model. The purpose of this model was to capture industry best practices and provide a shared conceptual framework before tailoring the process to the specific organizational context. BPMN was selected as the modelling notation because it offers a standardized and widely recognized language that can effectively represent process flows, decision points, and interactions among stakeholders across the software selection lifecycle.

At this stage, the baseline model was intended to represent only the high-level structure of the process, highlighting its three main phases and the overall flow of decisions and stakeholder interactions. Its goal was to provide a clear graphical overview that could help align stakeholders on the conceptual approach and ensure that the subsequent process redesign would be grounded in best practices.

### 2.4.2 Overview of the Baseline BPMN Model

The baseline BPMN model represents a generic software selection process, organized into three main phases:

1. Needs Identification & Initiation – This phase focuses on understanding the motivations behind software requests and ensuring alignment with the organization's strategic objectives. It sets the foundation for the selection process by defining the initial scope and engaging relevant stakeholders.
2. Requirements Definition & Evaluation Criteria – In this phase, functional and technical requirements are structured and validated at a conceptual level. Evaluation criteria are defined to provide a framework for subsequent vendor assessment and to ensure that all stakeholders have a shared understanding of what constitutes a suitable solution.
3. Market Analysis & Vendor Identification – The final phase addresses the identification of potential vendors and the initial assessment of available solutions. Key decision points, stakeholder interactions, and responsibility

boundaries are conceptually outlined to guide the subsequent detailed evaluation and selection process.

The baseline BPMN model is not intended to represent a finalized process. Instead, it provides a conceptual reference that illustrates how software selection activities, decisions, and stakeholder interactions can be organized systematically, serving as a starting point for further refinement.

### 2.4.3 Baseline Model as a Foundation for Customization

The baseline BPMN model functions as a first visual representation of the software selection process, schematizing the main phases and providing an initial view of how activities, responsibilities, and governance logic are organized. Its purpose is to give stakeholders a clear overview of the process and establish a conceptual foundation for subsequent refinement.

During the project, structured interviews with the business units involved in software selection were conducted to gather insights into current practices, organizational priorities, and specific requirements. These insights later informed the progressive customization of the BPMN model, allowing activities to be assigned to the appropriate business units, decision gates to be placed at the correct points, and interdependencies between roles to be formalized. The methodological approach used to refine and validate the process is discussed in the following chapter.

# 3. Methodology

## 3.1 Methodological Overview

As outlined in the "Introduction," the aim of this chapter is to describe and validate the methodology adopted during the project. More specifically, the chapter explains the working approach used to transform the baseline BPMN model of the software selection process, presented in the previous chapter, into a concrete and operational model tailored to the client's organizational structure. To analyse the case, a qualitative research approach was adopted, primarily based on direct stakeholder interviews, discussion workshops, and the analysis of existing documentation. This approach proved to be the most suitable because the "As-Is" process was not formally documented through written procedures or structured guidelines, but rather embedded in the operational experience and daily practices of people belonging to different business units. In this context, a quantitative approach based on surveys or statistical analysis would not have been effective: structured numerical data were not available, and standardized questionnaires would have resulted in generic answers, failing to capture relevant dynamics such as role overlaps, unclear responsibilities and misalignments between the boundaries within which each business unit operates in the process. Semi-structured interviews, instead, allowed for an in-depth exploration of how the process worked in practice, making it possible to identify critical issues, operational exceptions, and informal decision-making mechanisms that could not have emerged through document analysis alone. This qualitative approach made it possible to understand not only what activities were performed, but also why certain decisions were taken and which organizational constraints influenced stakeholder behaviour.

To structure and manage this work, an iterative Agile approach was adopted and adapted to the context of strategic consulting and Business Process Management. In practical terms, this meant working in short, recurring cycles in which information gathering alternated with feedback analysis, process model updates, and validation sessions with stakeholders. This method proved to be particularly useful, as it allowed the process design to be progressively refined as new constraints and insights emerged from stakeholder interactions, avoiding the risk of developing a theoretical model disconnected from operational reality. At the end of each iteration, validated takeaway deliverables were produced and used as the foundation for subsequent phases of the work.

The following sections describe in detail how this methodology was applied, focusing on the structure of the iterative Agile framework, the interview methodology and stakeholder involvement, the incremental development of the project deliverables, and the validation mechanisms adopted, leading to the definition of the decision gates and the finalization of the "To-Be" process.

## 3.2 Iterative Agile Framework, Interview Methodology and Deliverable Development

### 3.2.1 Structure of the Iterative Agile Framework

The nature of the project under analysis was characterized by an initial condition of fragmented information and non-formalized processes. Given these factors, the project activities were organized according to an iterative Agile framework, adopted to manage evolving requirements and progressively structure the available knowledge. The suitability of this approach is also supported by empirical studies, which highlight the effectiveness of iterative Agile frameworks in industrial settings. In particular, practices such as learning loops, time boxing, and continuous customer involvement through stakeholder feedback have proven effective in adapting complex processes through short cycles and ongoing feedback, as demonstrated by empirical analyses conducted on a large number of industrial projects. (Diebold and Dahlem, 2014)

The iterative cycle was based on the following five main steps:

1. Definition of interview's objectives
   At the beginning of each cycle, specific objectives were defined and targeted questions were prepared according to the stakeholder or business unit to be involved. This phase aimed to clearly identify the purpose of the interview, guiding the interaction toward a defined outcome and avoiding the collection of generic or dispersed information.

2. Stakeholder interviews

   The cycle then continued with the execution of interviews or workshops aimed at gathering information on the process, operational requirements, identified critical issues and potential improvement proposals. During this phase, the objective was to identify the specific operational activities conducted by each stakeholder throughout the process.
3. Formalization and validation of takeaways

   At the end of each interview, the collected information was synthesized and formalized into structured takeaways. These preliminary outputs were subsequently shared with the involved stakeholders to validate their content and ensure alignment with what had emerged during the discussion.
4. Implementation into project deliverables

   Once the takeaways were validated, the confirmed information was translated into operational updates of the project deliverables. Depending on the content that emerged, the iteration could result in modifications to the BPMN process model and, when necessary, in the development or update of additional supporting deliverables.
5. Validation of deliverables and planning of the subsequent iteration

   The updated deliverables were finally submitted to the stakeholders for content validation. Upon completion of the validation, the stakeholder or business unit to be involved in the subsequent iteration was identified, thereby initiating a new cycle.

This structure was applied consistently throughout the entire duration of the project, allowing the work to be organized over time through short and manageable cycles. The iterative approach enabled the progressive integration of new inputs, the continuous adaptation of the process model to emerging evidence, and the maintenance of a strong alignment between project activities and the client's operational reality.

### 3.2.2 Interview Methodology and Stakeholder Engagement

Interviews represented the primary tool for collecting the qualitative knowledge required to understand the software selection process and to progressively validate the project deliverable. This choice was driven by the cross-functional and non-formalized nature of the process under analysis, which made direct interaction with the involved stakeholders essential to reconstruct operational practices, decision-making logics, and interaction points among different organizational functions.

The planning of the interviews followed a prioritization logic based on the degree of involvement of each function in the process and on the expected informational relevance. Consistently with the iterative nature of the work, the selection of the stakeholder to be involved in the subsequent interview was defined immediately after each session, prioritizing the functions that emerged as most interconnected with the activities analysed in the previous iteration. The rationale underlying this sequencing and mode of engagement was explicitly "anti-silos." By adopting an "anti-silos" logic, the project enabled a progressive yet coordinated involvement of organizational units, supporting the construction of an integrated end-to-end view of the process and reducing the risk of fragmented and potentially inconsistent requirement collection. Interviews were scheduled on a weekly basis, depending on the availability of both the client and the project team. The stakeholder engagement path started with IT Architecture, identified as the main process owner and sponsor of the initiative, with the objective of obtaining an initial overall contextualization of the process scope and of the existing operational practices. Based on the evidence emerging from the first discussions, the involvement was progressively extended to the business units deemed most critical to the process. In particular, the Business and Business Partner functions were included to address the interface between business needs and technological solutions; IT Security and IT Governance were involved to cover control, security, and governance aspects; Project Management contributed to clarifying operational and temporal interdependencies; while Procurement was initially engaged to define the boundaries of interaction between the software selection phase and the subsequent purchasing phase.

From an operational standpoint, interview sessions were conducted according to a recurring structure. The EY team managers were responsible for steering the discussion, posing targeted questions aimed at exploring the role of the involved function, the activities performed within the process, and the interactions with other organizational units. My contribution focused on the systematic collection of the information emerging during each session, with the objective of ensuring full traceability and consistency of the discussed content. At the end of each interview, I synthesized the collected information into a structured takeaways document, which was subsequently shared with the reference stakeholder for formal validation, to ensure accurate alignment with what had emerged during the discussion. In the absence of a structured methodological design and of information traceability and validation mechanisms, the project would have risked reproducing, also within the "To-Be" model, the same fragmentation characterizing the "As-Is" situation.

### 3.2.3 Development of Project Deliverable

The project deliverables were not defined in a static manner during the starting phase, but rather emerged and progressively consolidated throughout successive iterations. Each cycle of interviews and stakeholder engagement contributed to clarifying new needs, challenging initial assumptions, or refining previously identified requirements, leading to incremental updates of the project deliverables. Consistently with the methodological approach outlined in the previous sections, the design activities did not follow a "big design upfront" logic, but instead adopted an "incremental development" approach, whereby each version of the deliverable incorporated both newly acquired information and lessons learned from prior iterations.

This iterative mode of development also contributed to fostering a stronger sense of ownership among the actors involved, who were able to observe how their feedback was progressively synthesized and translated into concrete adjustments to the process and its supporting tools. This contributed to strengthening the legitimacy of the "To-Be" model and reducing the risk of it being perceived as an externally imposed solution. Conversely, an approach based on a single final release would have increased the distance between those designing the process and those responsible for its daily execution, reducing commitment and increasing resistance during the adoption phase.

### BPMN Process

The BPMN model was constructed and refined progressively through interviews with the business units involved in the software selection process. These interactions enabled a detailed reconstruction of the operational workflow, identifying performed activities, decision points, and inter-functional interactions across the three process phases: Needs Identification and Initiation, Requirements Definition and Evaluation Criteria and Market Analysis and Vendor Identification. In several instances, multiple interviews with the same business units were required to align different stakeholders on the emerging end-to-end process view, ensuring a shared understanding not only of the function-specific activities, but also of the complementary contributions of other units along the overall flow.

The iterative analysis and synthesis of the interview outcomes enabled the progressive consolidation of the BPMN model, ensuring consistency across phases and alignment among the involved organizational units. Emphasis was placed on the identification and formalization of decision points. Empirical evidence showed that, in the "As-Is" situation, several critical decisions were taken informally, with inconsistent stakeholder involvement and without clearly defined criteria or traceability mechanisms. Based on these findings, decision points were progressively translated into formal decision gates within the BPMN model. Examples include the distinction between new software selection and the implementation of an existing solution, the decision to initiate or skip a Request for Information (RFI) and go/no-go decisions for demos and deep-dive assessments.

From a methodological standpoint, these decision gates were not predefined "ex ante," but emerged incrementally through stakeholder interaction, with the objective of making explicit the ownership of decision moments that were already present in practice but not formally codified. Decision gates thus function as structuring elements of the process, enabling the explicit representation of minimum informational requirements and inter-functional alignment points required to proceed across phases, such as the validation of business and technical requirements prior to market analysis or the execution of preliminary security screenings before vendor demos.

The BPMN model was developed using Microsoft Visio, a professional diagramming tool that supports the standardized representation of complex workflows through native BPMN elements. Visio was selected for its flexibility in updating process models across iterations and for its ability to support collaborative review and validation activities with stakeholders, ensuring continuous alignment between the model and the evolving understanding of the process. In parallel with the graphical representation of the process, a descriptive support Word document was developed, detailing individual process steps and corresponding operational ownership. This document serves as an interpretative guide to the BPMN model, enhancing its accessibility for stakeholders less familiar with formal process notation.

**RACI Matrix**

The RACI matrix constitutes a complementary tool aimed at making operational responsibilities explicit within the process, distinguishing among the roles of Responsible, Accountable, Consulted, and Informed. Its development followed an incremental logic, grounded in the information collected during interviews and iteratively refined as ambiguities and overlaps emerged and were clarified through stakeholder validation.

For each process phase, responsibilities were assigned to the relevant stakeholders with the objective of clarifying roles and operational boundaries while ensuring consistency with the BPMN process logic. The RACI matrix was not conceived as a prescriptive instrument nor as a basis for performance evaluation; rather, it was designed to strengthen process ownership and to mitigate responsibility overlaps identified in the "As-Is" analysis, such as ambiguities between Business and Business Partner roles or uncertainties regarding accountability for RFI and demo-related activities.

**Operative Tools**

In parallel with process formalization and responsibility clarification, the interviews revealed recurring operational issues that could not be adequately addressed through BPMN modelling alone. These issues included unstructured requirement collection, limited comparability of vendor evaluations and a strong reliance on individual knowledge for market scouting and supplier identification. In response, a methodological approach oriented toward the development of operative support tools was adopted, conceiving them as functional extensions of the formalized process and as enablers of more structured and transparent decision-making. Templates and operational guidelines were therefore designed as flexible artefacts, aimed at supporting critical activities and reducing dependence on informal practices characteristic of the "As-Is" situation. The design of these operative tools was explicitly informed by the principles of Multi-Criteria Decision Making (MCDM), applied not as an optimization framework but as a conceptual lens to decompose decisions into explicit criteria, clarify trade-offs and make evaluation logics transparent and discussable among stakeholders.

Within this framework, the "Business Requirement Template" was conceived to structure business needs and objectives through a set of questions organized into thematic categories (context and scope, functional requirements, security and compliance, data management, non-functional requirements, strategic impact, and global versus local scope). These categories enable the explicit articulation of the multiple dimensions influencing software selection decisions and provide the conceptual basis for subsequent evaluation stages. The template also includes a dedicated section for the classification of detailed business requirements using the "MoSCoW method" (Must-have, Should-have, Could-have, Won't-have), employed as a prioritization logic rather than a scoring mechanism. Its purpose is to make explicit the relative criticality of requirements from a business perspective, distinguishing indispensable capabilities from desirable or explicitly out-of-scope ones. This prioritization supports the definition of minimum acceptance thresholds and guides the focus of later comparative assessments. From a methodological perspective, the template operationalizes an MCDM logic at the level of requirement structuring and prioritization, generating a coherent and traceable set of evaluation criteria.

The "Decision Matrix," in turn, was designed as an explicitly MCDM-based instrument to support the multidimensional evaluation of vendors throughout the software selection process. Its structure distinguishes between two primary decision dimensions: coverage of functional and non-functional requirements, articulated into domains and subdomains, and empirical evidence gathered during vendor demos, captured through qualitative criteria such as user experience and perceived added value. A balanced weighting scheme was adopted, assigning 50% of the overall evaluation to requirements coverage and 50% to demo-related evidence. This distribution is not intended as a mathematically optimal solution, but rather as a governance choice aimed at avoiding both an exclusively requirements-driven assessment and a purely demo-driven evaluation.

Within this methodological architecture, the Business Requirement Template, the Decision Matrix, and the Operating Guidelines were developed as complementary instruments to the BPMN process model and fully aligned with the responsibilities defined in the RACI matrix.

## 3.3 Methodological Consolidation and Validation of Project Deliverable

### 3.3.1 Validation Mechanisms and Acceptance Criteria

The validation of the project deliverables was conducted through a coordinated set of mechanisms designed to ensure consistency between the conceptual process model and the operational practices effectively adopted by the various functions involved. These mechanisms combined synchronous moments of structured discussion with asynchronous review activities, with the objective of balancing analytical depth and organizational sustainability.

First, process walkthroughs were conducted, during which the updated BPMN model was discussed using real or plausible software selection cases. Through a step-by-step reconstruction of activities, inter-functional interactions and key decision points, this approach enabled an immediate verification of the alignment between the process flow and the stakeholders' operational experience, as well as the rapid identification of potential deviations, omissions, or ambiguities in the process representation.

Second, joint review sessions were organized, focusing on the changes introduced following each iteration. During these sessions, the proposed modifications to the BPMN model, the RACI matrix, and the operational tools were explicitly discussed, collecting objections, requests for clarification, and suggestions for integration.

This approach made the link between evidence emerging from interviews and design choices transparent, reducing the risk that design decisions could be perceived as arbitrary or imposed top-down.

Finally, validation was further supported through the asynchronous sharing of a descriptive document accompanying the BPMN model, developed in Word format and containing a textual description of process phases, activities, and operational responsibilities. This document allowed stakeholders to conduct a more reflective review, enabling the submission of detailed comments and proposed revisions outside synchronous discussion moments. The combination of these validation channels reduced reliance on individual meetings and fostered more inclusive participation from functions with limited availability.

The acceptance criteria for the proposed solutions were defined consistently with the governance and standardization objectives identified in the previous phases. In particular, a modification was consolidated in subsequent versions of the process only if it: complied with existing architectural constraints and IT policies, as governed by IT Architecture and IT Security; ensured role clarity and alignment with the RACI assignments, avoiding overlaps or grey areas of responsibility; maintained consistency with the objectives of traceability, fragmentation reduction, and standardization identified during the diagnostic phase; and did not introduce new responsibility gaps or inter-functional conflicts requiring additional organizational compensations.

The decision to adopt a "continuous validation" mechanism embedded within each iteration was motivated by the need to progressively manage heterogeneous and, at times, divergent feedback, avoiding the concentration of all negotiations in the final phase of the project. A single final approval stage would have made it more difficult for conflicts between local needs and standardization requirements to emerge and be gradually resolved, increasing the risk of reopening discussions on already mature deliverables and delaying project completion. Conversely, the iterative integration of feedback made it possible to anticipate critical issues and to negotiate the necessary compromises among functions in time.

### 3.3.2 Methodological Limitations and Alternatives

The adopted methodological approach, while coherent with the project context, presents some limitations that must be explicitly stated for transparency and correct interpretation of results.

A first limitation stems from the strong dependence of the "Agile iteration cycles" on stakeholder availability. The quality, depth and timeliness of iterations were directly correlated with the ability to organize discussion sessions with business representatives whose agendas were often highly constrained. During periods of reduced availability, the pace of deliverable updates inevitably slowed, requiring careful prioritization of objectives within each cycle.

A second limitation is related to the cyclical nature of the approach. In the absence of rigorous discipline regarding iteration objectives, there is a risk of excessive proliferation of revisions, potentially extending the overall design timeline. To mitigate this risk, the objectives of each cycle were explicitly defined, associating each iteration with a limited and well-scoped set of decisions to be taken or issues to be clarified.

A third limitation concerns the trade-offs required to balance local needs and global standardization. In several instances, specific preferences expressed by individual functions, such as the sequencing of certain activities or the depth of specific controls, were not fully accommodated to preserve the coherence of a uniform, enterprise-wide workflow, and its future scalability in a digitalization perspective. These choices required explicit negotiation, with the awareness that full adherence to local preferences would have compromised the objective of developing a single integrable process model.

During the project, several methodological alternatives were also considered but not adopted. A first set of alternatives involved the use of strongly quantitative approaches to process design, such as the extensive application of Multi-Criteria Decision Making (MCDM) techniques not only for vendor evaluation but also for workflow structuring and responsibility allocation. This option was discarded because the primary objective of the project was the formalization of process governance (activities, roles, decision gates, and interfaces) rather than the numerical optimization of an already defined scheme. The MCDM paradigm was instead selectively incorporated into the design of operational tools, such as the Business Requirement Template and the Decision Matrix, where multi-criteria logic supports requirement structuring and vendor comparison, but was not used to drive the process modelling itself.

Approaches such as the Analytic Hierarchy Process (AHP) could have generated formal rankings of alternatives and explicit weighting of evaluation criteria; however, they would not have been adequate to address the central research problem, namely the absence of a shared governance framework for software selection. In the absence of a clearly defined process, formalized roles and explicit decision gates, sophisticated scoring mechanisms risk shifting attention toward decision-support tools without addressing the more fundamental issue of accountability and traceability along the decision flow. In other words, the extensive use of AHP or similar MCDM methods would have been premature and potentially misleading, as it would have provided an appearance of numerical precision on an incomplete process architecture.

Finally, a “big design upfront” approach was not adopted. It would have entailed the “ex ante” definition of a comprehensive process model primarily based on market best practices or generic guidelines. Although potentially faster in producing an initial design, this option was deemed less suitable given the objective of adhering closely to the client’s organizational specificities. It would have increased the risk of producing a theoretically model but poorly aligned with actual operational practices, making it more difficult to implement and gain stakeholder acceptance.

### 3.3.3 Methodological Coherence and Transition to Results

Despite the limitations identified, the adopted methodology is overall coherent with the project objectives, which aimed to transform fragmented practices into a formalized, traceable, and replicable workflow while maintaining alignment with the client’s specific operational and architectural constraints.

From the perspective of internal coherence, the sequence linking needs identification, baseline BPMN construction, iterative customization, the progressive introduction of decision gates, and continuous validation defines an organic methodological pathway. In this framework, each design choice derives from empirically observed issues in the "As-Is" state and is justified through explicit governance criteria. Decision gates are therefore not imposed "a priori" but represent formalized versions of decisions already present in practice (distinction between new software selection and the implementation of existing solutions, or the decision to activate or skip an RFI) rendered traceable and supported by clearly defined minimum informational requirements.

The integration of the MCDM paradigm into the design of operational tools further reinforces this coherence, as it translates the multi-criteria logic discussed in the literature into concrete mechanisms for structured requirement collection and comparative vendor evaluation. In this way, the quantitative dimension is not positioned in opposition to process formalization but is incorporated as an operational component supporting decision-making within an already defined governance framework.

Overall, the produced deliverables, the BPMN process, the RACI matrix, and the operational tools (Business Requirement Template, Decision Matrix, and Operating Guidelines), constitute the concrete translation of this methodological pathway into outputs that are usable and integrable within organizational practices.

# 4. Results

## 4.1 Introduction to results

This chapter presents the results of the thesis, in terms of the operational deliverables developed to support a structured “Software Selection process” within the analysed organisational context. The overall objective of the project was to formalise an end-to-end decision-making workflow, clarify roles and responsibilities across the stakeholders involved and introduce practical tools aimed at enabling traceable and repeatable vendor evaluation activities.

In line with these objectives, the chapter discusses the following deliverables:

- BPMN” To-Be”
- RACI matrix
- Business Requirement Template
- Decision Matrix
- Operating Guidelines

For each deliverable, the analysis focuses on two complementary aspects: what was developed and which specific issues it addresses with respect to the main pain points identified in the “As-Is” situation (Chapter 2). In particular, the results aim to mitigate process fragmentation, the use of informal and non-comparable requirements, ambiguity in decision-making roles, the late involvement of IT Architecture and IT Security functions and the strong dependence on individual expertise during market analysis and vendor evaluation activities. Critical reflections and potential future developments of the proposed framework are exposed in the “Discussion” chapter.

## 4.2 BPMN “To Be”

The BPMN “To-Be” process constitutes the central outcome of the design phase and represents the structural backbone of the proposed “Software Selection Process “project. It formalises a clear and repeatable workflow that guides the process from the initial business request to the definition of a final vendor shortlist, ensuring that selection activities remain within a governed perimeter before the formal engagement of Procurement. In doing so, the process responds to the objective of the project: defining a standardised path capable of supporting technically and economically informed decisions aligned with the organization’s strategic and operational objectives. The other project deliverables are conceived as complementary instruments that operate within this process model.

The resulting workflow is composed of 31 activities and decision gates, organised into three main phases that structure the Software Selection, as outlined in Figure 6: BPMN workflow The phases are described as follow.

The first phase, Needs Identification & Initiation, focuses on clarifying the business motivations underlying the request and assessing its consistency with the existing IT landscape. The Business and the Business Partner collect and consolidate needs while analysing current capabilities and applications. IT Architecture evaluates whether the request can be addressed through the extension of an existing solution or requires the activation of a new Software Selection initiative. A formal decision gate distinguishes between "vendor implementation" and "new software selection," ensuring that only aligned initiatives proceed further and that non-coherent requests are halted with documented justification. Once alignment is confirmed, the process is formally initiated, establishing a shared foundation for the subsequent phases.

The second phase, Requirements Definition & Evaluation Criteria, translates business needs into a structured and documented set of functional, non-functional, and technical requirements. These are drafted by the Business through the "Business Requirement Template" and validated in parallel by Business stakeholders and IT Architecture to assess feasibility and completeness. A consolidated requirements matrix is then developed to support the construction of the evaluation framework. This framework defines the criteria for assessing vendor solutions and supports the decision on whether issuing a Request for Information (RFI) is required. By structuring requirements and validation activities upfront, the phase aims to enhance alignment and reduce ambiguity in later evaluation stages.

The third phase, Market Analysis & Vendor Identification, focuses on identifying and assessing potential vendors in line with the defined evaluation framework. Where appropriate, an RFI is prepared and issued, and responses are analysed with the involvement of IT Architecture for technical assessment. A dedicated security checkpoint activates the IT Security process and determines whether additional cybersecurity analysis is necessary. Vendors are pre-shortlisted based on structured scoring, and their eligibility within the procurement system is verified. Where required, demo (proof of concept) sessions are organised to collect technical and business feedback before consolidating a final shortlist to be transferred to

Procurement, thereby concluding the Software Selection process with a validated set of options.

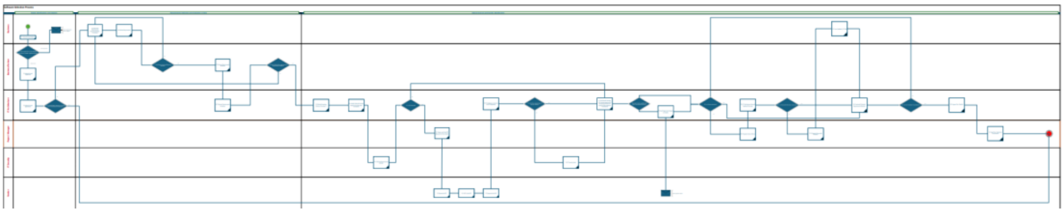

*Figure 6: BPMN workflow*

At an operational level, the BPMN diagram is supported by a dedicated descriptive document that specifies, for each activity, its purpose, expected outputs, responsible actors, and handover mechanisms. This complementary documentation strengthens governance by clarifying responsibilities and embedding quality checks at key stages of the workflow, extending process control beyond the graphical representation of the model.

From the perspective of the "As-Is" analysis, the BPMN "To-Be" process contributes to addressing several identified pain points. First, it mitigates process fragmentation by introducing an end-to-end structure with explicit phases and documented handovers. Second, it reduces decision-making ambiguities through clearly defined gates (e.g. distinction between implementation and new selection, the RFI checkpoint, the security review trigger, and demo go/no-go decisions) each associated with predefined inputs and accountable stakeholders. Third, it anticipates the involvement of IT Architecture and IT Security, positioning them at structurally embedded checkpoints rather than at late or discretionary stages. The role of the Business Partner as a stable interface between Business and IT further supports coordination and consolidation of requirements, thereby contributing to the reduction of misalignment and iterative rework.

Overall, the BPMN "To-Be" and its supporting operational description establish a governed and structured Software Selection workflow in which roles, activities and decision points are explicitly defined and coordinated across the three phases. At the same time, the model does not in itself ensure the quality of requirements nor the objectivity of vendor comparison, aspects that are addressed through the complementary tools presented in the following sections.

## 4.3 RACI Matrix

The RACI matrix constitutes the primary governance deliverable supporting the Software Selection process, as it makes explicit the distribution of responsibilities across the functions involved throughout the three phases of the BPMN “To-Be” model. It translates the process design into a structured accountability framework, defining for each phase who is Accountable (A), who is Responsible (R), and which actors are Consulted (C) or Informed (I). In doing so, it prevents the execution of the workflow from relying on informal interpretations of roles and implicit agreements among stakeholders. Operationally, as shown in Figure 7: RACI Matrix the matrix is structured by process phase and by stakeholders, assigning differentiated combinations of R, A and C according to the nature of the activities. During the Needs Identification & Initiation and Requirements Definition & Evaluation Criteria phases, the Business is Accountable for the definition of needs and requirements, while the Business Partner is Responsible for operational coordination and stakeholder engagement. IT Architecture is positioned as Consulted to ensure architectural coherence and technical feasibility. In the Market Analysis & Vendor Identification phase, the balance shifts: IT Architecture assumes an A/R role in the technical evaluation and shortlist construction, the Project Manager is Responsible for operational execution (including RFI management, demo coordination and vendor interaction), IT Security is Consulted for cybersecurity-related assessments, and Vendors are Responsible for providing information and participating in evaluation activities. Through this structure, the RACI matrix synthesises at a governance level the expected involvement scheme, ensuring consistency with the swimlanes and decision gates defined in the BPMN model.

With reference to the pain points identified in the “As-Is” analysis, the matrix addresses two critical dimensions. First, it strengthens stakeholder coordination by clarifying the interaction perimeter between Business and IT Architecture and by formalising the role of the Business Partner as a structured interface between the two domains. In the previous situation, the absence of explicit ownership often led to requirement collection, technical validation and market engagement being driven by individual initiatives, generating overlaps and uncertainty regarding process leadership. The systematic allocation of Accountability to the Business in the early phases and Responsibility to the Business Partner for operational orchestration makes visible who governs the content of the need and who governs the process flow, thereby reducing the risk of gaps or duplications.

Second, the RACI matrix contributes to reducing decision-making ambiguities associated with key process gates. By explicitly identifying who is Accountable in the qualification of the initial request, in the validation of the requirements matrix, and in the definition of the vendor shortlist, it limits the risk of "ex post" disputes regarding decision ownership. The alignment between BPMN decision gates and the assignments avoids functional overlaps (for instance between Business and IT Architecture in distinguishing vendor implementation from new selection, or between IT Architecture and the Project Manager in the management of demos) and reduces dependence on informal negotiations in borderline cases. In this sense, the matrix establishes a clear process-level map of accountability that supports consistent and predictable stakeholder engagement.

However, while the RACI matrix clarifies ownership and interaction patterns, it does not in itself impose performance metrics nor eliminate potential organisational resistance to role formalisation. Its effectiveness depends on consistent application and alignment with the broader governance framework established by the BPMN model and the complementary operational tools described in the following sections.

| Phase | Business | Business Partner | IT Architect | Project Manager | IT Security | Vendor |
|---|---|---|---|---|---|---|
| Needs Identification & Initiation | A | R | C | - | - | - |
| Requirements Definition & Evaluation Criteria | A | R | C | - | - | - |
| Market Analysis & Vendor Identification | C | C | A/R | R | C | R |

**Legend:**

- **R** = Responsible
- **A** = Accountable
- **C** = Consulted
- **I** = Informed
- **–** = Not involved

*Figure 7: RACI Matrix*

## 4.4 Business Requirements Template

The Business Requirement Template represents the operational tool supporting the "Requirements Definition & Evaluation Criteria" phase of the Software Selection process. It translates business needs into a structured and documented set of requirements that can be systematically assessed and shared with IT Architecture and other stakeholders. The template is directly used by the Business to formalise the initial request and to transmit information in a coherent and traceable manner, thereby reducing reliance on informal exchanges of heterogeneous documents or unstructured e-mail communication.

Structurally, the deliverable is organised into two complementary steps. In the first step, the Business completes a structured questionnaire organised into thematic categories identified through interviews and needs analysis, including: context and scope, functional requirements, security and compliance, data management, non-functional requirements, strategic impact, and global versus local scope. Each section provides guided completion instructions and highlights mandatory fields to ensure that critical aspects, such as system integrations, regulatory constraints, security requirements, data typologies, performance expectations, and strategic relevance, are systematically addressed and rendered comparable across different initiatives. In the second step, the template includes a "Detailed Business Requirements" section in which individual requirements are listed and classified according to the MoSCoW method (Must-have, Should-have, Could-have, Won't-have). This approach is employed as a prioritisation logic rather than as a quantitative scoring mechanism. By explicitly distinguishing essential capabilities from optional or out-of-scope functionalities, the template supports the definition of minimum acceptance thresholds and provides a structured input for subsequent vendor evaluation activities.

With reference to the pain points identified in the "As-Is" analysis, the Business Requirement Template addresses several critical dimensions. First, it mitigates the issue of informal and non-comparable requirement collection by replacing heterogeneous and unstructured documentation with a unified and standardised framework covering all relevant selection dimensions. Second, it reduces ambiguity in the translation of business needs between Business and IT Architecture: the standardised structure and prioritisation section make explicit which requirements are essential from a business perspective and provide IT Architecture with a structured input for technical validation and for the construction of evaluation criteria. Finally, by embedding categories and guided questions derived from lessons learned during the interviews, the template codifies organisational knowledge and contributes to reducing dependence on individual expertise in the requirements definition phase, thereby supporting greater consistency and repeatability over time.

However, while the template enhances structure and comparability, it does not in itself guarantee the completeness or quality of the information provided, as its effectiveness depends on the accuracy and engagement of the Business in the compilation phase. Moreover, being a document-based tool, it requires facilitation and consistent application to fully support the governance logic of the overall framework.

## 4.5 Decision Matrix

The Decision Matrix represents the primary quantitative support tool for the "Market Analysis & Vendor Identification" phase and is designed to translate vendor evaluations into a structured numerical format. The tool supports a systematic comparison among vendor alternatives based on predefined criteria derived from the earlier phases of the process. Project Manager is responsible for preparing the template and coordinating its completion, while evaluative inputs are provided by both Business and IT Architecture. Specifically, the Business completes the section related to demo assessment, expressing judgements on user experience and value demonstration criteria, whereas IT Architecture evaluates the Requirements Coverage section by assigning scores reflecting the degree of compliance with the functional and non-functional requirements defined in the Business Requirement Template. The deliverable is organised into two macro-categories of evaluation: Requirements Coverage and Demo, each further decomposed into sub-criteria with vendor-specific scores. The Requirements Coverage category includes sub-domains related to functional and non-functional requirements, while the Demo category captures the Business assessment of aspects such as alignment with real use cases, usability, perceived performance, clarity of vendor responses and evidence of added value. Individual scores are aggregated into weighted category scores, which are then combined into an overall vendor score through configurable category weights embedded in the template.

Methodologically, the Decision Matrix is grounded in a Multi-Criteria Decision-Making (MCDM) logic used as a conceptual framework for structuring the tool. The project did not develop or apply a formal MCDM method for weight elicitation or optimisation. Instead, the MCDM perspective was adopted to frame the vendor selection problem, identify the relevant evaluation dimensions, and organise them into coherent assessment categories. In this sense, the contribution lies in translating the vendor selection process into a structured multi-criteria problem, within which alternatives are assessed across clearly defined dimensions. The definition of relative weights is intentionally left to the organisation, as is expected to be determined at the moment of tool application by Business and the CIO, in alignment with strategic priorities and initiative-specific risk considerations. As an initial configuration, the project team proposes a 50%–50% distribution between the two macro-categories. This choice is based on qualitative reasoning and reflects the intention to avoid a selection process that is either purely requirements-driven or excessively demo-driven. By attributing equal relevance to formal requirement alignment and empirical evidence from demo sessions, the tool encourages a balanced evaluation logic.

With reference to the pain points identified in the "As-Is" analysis, the Decision Matrix addresses the reliance on purely qualitative and non-traceable vendor evaluations by providing a shared and documented assessment structure. It enables Business and IT Architecture to express their perspectives within a common framework, reducing the risk of vendor bias driven solely by demo impressions or, conversely, by technical documentation detached from user experience. Furthermore, its integration with the Business Requirement Template and the BPMN "To-Be" model ensures procedural coherence: formalised requirements become explicit evaluation criteria, RACI-defined roles regulate compilation responsibilities, and the aggregated score constitutes a structured input for the final decision gates related to vendor shortlisting.

## 4.6 Operating Guidelines

The Operating Guidelines constitute the standardisation and governance support tool for the market screening activity within the "Market Analysis & Vendor Identification" phase. The document originates from a specific need expressed by IT Architecture during the stakeholder interviews: the absence of a shared and structured reference for vendor identification meant that scouting activities were dependent on the individual experience and personal networks of the architects involved, with no common baseline across different initiatives.

The tool is structured into a set of macro-areas that organise the market screening activity according to different software typologies. The definition of these macro-areas is the result of a sector benchmarking analysis conducted during the project, which examined recurring categorisations in industry practices, analyst frameworks, and common corporate software adoption patterns. These macro-areas include High Strategic Impact Software, Global Adoption Software, Software for Local Adoption, High Innovation Software, Emerging Solutions, and AI Tools for Scouting.

For each macro-area, the document defines the applicative perimeter and suggests a set of tools and information sources considered appropriate for the specific scouting objective. For example, established analyst positioning frameworks (e.g., Gartner Magic Quadrant, Forrester Wave, IDC MarketScape) are associated with high strategic impact software; user review and comparison platforms (such as G2, Capterra, and TrustRadius) are suggested for globally adopted solutions; and startup intelligence databases (including Crunchbase and CB Insights) are proposed for emerging or innovation-driven domains. It is important to note that the sources, databases, and tools listed are not intended to be exhaustive: the IT Architecture team may, at its discretion, use a subset of these resources or adopt alternative ones depending on the specific technological domain, market context, or scope of the scouting activity.

A relevant integration element concerns the link with the Business Requirement Template: the information captured in the Strategic Impact and Global vs Local sections of the template directly orientates the selection of the most appropriate scouting area. Through this connection, market screening is not decoupled from the requirements definition phase but develops in continuity with the information formalised by the Business, reinforcing coherence across process phases and ensuring that the extensive list of vendors produced by IT Architecture is consistent with the strategic and organisational scope of the initiative.

With respect to the pain points identified in the "As-Is" analysis, the Operating Guidelines address two principal dimensions. On one side, it reduces the fragmentation and non-comparability of market screening activities by introducing a shared reference structure that clarifies where to search and according to which logic, while preserving sufficient flexibility for adaptation to specific technological or market contexts. On the other, by codifying the knowledge and tools previously embedded in the experience of individual architects, it transforms tacit organisational knowledge into an explicit and replicable asset: the quality and consistency of vendor identification activities are less dependent on the competencies of specific individuals but anchored to a shared and updatable governance tool.

# 5. Conclusions and Future Developments

This work set out to address a concrete and recurring gap in the technological management of complex organizations: the absence of a structured and replicable process for software selection. The aim of this project was to develop an applicable framework, grounded in the specific context of a multinational company operating in the engineering consulting sector, capable of transforming fragmented and informal practices into a formalized and governed workflow, consistent with the principles of Business Process Management applied to IT decision-making. The contribution of this study therefore lies at the intersection of technology governance and process design, with the purpose of translating established methodological principles into a model that can be effectively implemented within an organizational environment.

The project achieved the three main objectives defined at the outset. The first concerned the definition of a structured software selection process, with clearly assigned roles and responsibilities across the involved business units. The result is a BPMN-based workflow structured into three macro-phases: Needs Identification and Initiation, Requirements Definition and Evaluation Criteria, and Market Analysis and Vendor Identification. Each phase is governed by explicit decision gates that guide the process according to the specific characteristics of each request. This structure makes the interdependencies among stakeholders, activities, and decision points visible, introducing a governance logic that was previously absent. The second objective focused on the development of operational tools supporting the activities defined within the process. To this end, a Business Requirement Template was designed to collect functional and non-functional requirements in a structured manner, a Decision Matrix was developed to enable comparative and weighted vendor evaluation, and Operating Guidelines were prepared to support market screening through analyst sources and structured platforms. The third objective consisted in integrating the software selection process with adjacent frameworks already active within the organization, particularly Security processes, the Procurement system, and IT Architecture guidelines. This integration ensures that software selection does not operate as an isolated procedure, but as a coherent component of a broader IT governance ecosystem, reducing the risk of technical decisions misaligned with security, architectural, or contractual constraints. Overall, the developed framework has contributed to transforming an approach that was fragmented and highly dependent on individual knowledge into a standardized workflow, characterized by visible decision steps, coordinated stakeholder involvement, and formalized evaluation criteria.

The relevance of establishing a structured software selection process extends beyond operational efficiency and reflects a broader principle: technological choices should be treated as strategic decisions. In the identified "As-Is" configuration, the absence of clearly defined phases, shared evaluation criteria, and formal accountability exposed the organization to concrete risks, including tool duplication, delayed involvement of key functions, relational bias in vendor evaluation, and limited traceability in purchasing decisions. The proposed model introduces structural mechanisms aimed at mitigating these risks through explicit decision gates, a RACI matrix defining accountability at each phase, and standardized templates ensuring that evaluations remain verifiable and comparable over time. In this sense, the process does not merely reorganize existing activities, but redefines how technological decisions are structured, discussed, and documented.

However, a rigorous assessment requires acknowledgment of the limitations of the study. The first limitation is empirical in nature. Although the process was designed, iterated with stakeholders, and conceptually validated, it has not yet been implemented and executed end-to-end across a substantial number of real cases. Consequently, the expected benefits, such as reduced decision lead times, improved evaluation quality, and stronger alignment between IT and Business, are consistent with the reference literature but have not yet been supported by quantitative internal evidence. The KPIs proposed within the process documentation, including Time to Selection, Requirements Coverage, Stakeholder Satisfaction, and Vendor Responsiveness, will only enable empirical validation once the process becomes fully operational and generates a sufficiently robust dataset for comparative analysis. A second limitation concerns the Decision Matrix. The tool was designed following a MCDM logic to structure the evaluation problem and define coherent assessment categories. However, the specific weights assigned to individual criteria were not definitively established during the project. This was an intentional choice, reflecting the understanding that the relative importance of evaluation dimensions depends on the specific context of each initiative. While this approach increases flexibility and adaptability, it also introduces a degree of discretion that requires decision-making maturity from the involved stakeholders. The effectiveness of the matrix will therefore depend on the ability of Business and IT Architecture functions to calibrate evaluation weights in alignment with strategic priorities and the risk profile of each selection. A third limitation relates to the economic dimension. While the process provides a structured approach to technical and functional vendor evaluation, an in-depth analysis of Total Cost of Ownership, licensing models, and return on investment remains delegated to the Procurement phase following the vendor shortlist definition. This separation may create tensions between technical suitability and economic sustainability and represents a potential area for future integration, particularly in contexts where budget constraints significantly influence final decisions. Finally, the formalization of a process constitutes a necessary but not sufficient condition for organizational change. The effective adoption of the model depends on stakeholders' willingness to internalize newly defined roles and responsibilities, particularly the Business Partner role, which assumes a central function in coordinating and mediating between business needs and technological constraints. Without adequate investment in training and change management, there is a risk that the process may be progressively bypassed under operational pressure, reproducing informal dynamics like those observed in the "As-Is" configuration. At the same time, excessive formalization, if not balanced with flexibility, may slow down initiatives or generate perceptions of bureaucratization, highlighting the need for continuous monitoring of the balance between control and agility.

Looking forward, two main development directions can be identified. The first concerns the digitalization of the workflow through implementation on the ServiceNow platform. In its current configuration, process execution relies on email exchanges, shared folders, and manual template completion. A platform-based implementation would enable the configuration of a structured digital workflow with automated task activation, traceable approval management, and controlled parallelization of activities. Beyond improving operational efficiency, such digitalization would enhance transparency and monitoring capabilities, enabling systematic KPI collection and the identification of bottlenecks through dedicated dashboards. Moreover, the creation of a digital historical repository of past selections would support ex-post analysis and organizational learning. The second development direction involves the integration of a dedicated evaluation module for AI-based solutions. As AI-enabled tools increasingly become part of corporate application portfolios, additional evaluation dimensions emerge, including the degree of automation, explainability of outputs, exposure to algorithmic risks, and regulatory compliance. Introducing a specific framework to assess such solutions would enable more informed governance of this class of technologies and prevent the mechanical application of criteria originally designed for conventional software systems.

More broadly, the methodological framework developed in this project, based on phased structuring, formalized responsibilities, criteria-based evaluation, and iterative decision gates, demonstrates a level of generalizability that allows adaptation to different organizational contexts. This transferability represents a contribution to the broader discussion on Business Process Management applied to IT decision-making, while simultaneously highlighting the need for empirical validation in diverse environments to assess its effectiveness.

In conclusion, this work provides a structured and coherent model for governing the software selection process, addressing a concrete operational gap, and proposing a balanced approach between formalization and flexibility. Although full empirical validation is still pending, the developed framework constitutes a solid methodological foundation for future process evolution and for further research aimed at measuring its performance over time.

# References

Bernardo, R., Galina, S.V.R. and Pádua, S.I.D.D. (2017) 'The BPM lifecycle: How to incorporate a view external to the organization through dynamic capability', *Business Process Management Journal*, 23(1), pp. 155–175. Available at: https://doi.org/10.1108/BPMJ-12-2015-0175.

Brkić, Tomičić Pupek, Bosilj Vukšić (2020) 'A Framework for BPM Software Selection in Relation to Digital Transformation Drivers', *Tehnicki vjesnik - Technical Gazette*, 27(4). Available at: https://doi.org/10.17559/TV-20190315193304.

Diebold, P. and Dahlem, M. (2014) 'Agile practices in practice: a mapping study', *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. *EASE '14: 18th International Conference on Evaluation and Assessment in Software Engineering*, London England United Kingdom: ACM, pp. 1–10. Available at: https://doi.org/10.1145/2601248.2601254.

Dumas, M. *et al.* (2018) *Fundamentals of Business Process Management*. Berlin, Heidelberg: Springer Berlin Heidelberg. Available at: https://doi.org/10.1007/978-3-662-56509-4.

Fumagalli, L. *et al.* (2019) 'Framework for simulation software selection', *Journal of Simulation*, 13(4), pp. 286–303. Available at: https://doi.org/10.1080/17477778.2019.1598782.

Jadhav, A.S. and Sonar, R.M. (2009) 'Evaluating and selecting software packages: A review', *Information and Software Technology*, 51(3), pp. 555–563. Available at: https://doi.org/10.1016/j.infsof.2008.09.003.

Niknamfar, A.H. and Niaki, S.T.A. (2018) 'A binary-continuous invasive weed optimization algorithm for a vendor selection problem', *Knowledge-Based Systems*, 140, pp. 158–172. Available at: https://doi.org/10.1016/j.knosys.2017.11.004.

Putra, H. and Er, M. (2024) 'The Role of Business Process Management in Digital Innovation and Digital Transformation: A Systematic Literature Review', *Procedia Computer Science*, 234, pp. 829–836. Available at: https://doi.org/10.1016/j.procs.2024.03.069.

Rahimi, F., Møller, C. and Hvam, L. (2016) 'Business process management and IT management: The missing integration', *International Journal of Information Management*, 36(1), pp. 142–154. Available at: https://doi.org/10.1016/j.ijinfomgt.2015.10.004.

*The History and Evolution of Business Process Management (BPM)* (2024). Available at: https://www.emakin.com/the-history-and-evolution-of-business-process-management-bpm (Accessed: 13 January 2026).

'Understanding the Core Elements and Key Concepts of BPMN' (2024) *ArchiMetric*, 9 December. Available at:

https://www.archimetric.com/understanding-the-core-elements-and-key-concepts-of-bpmn/ (Accessed: 13 January 2026).

Van Der Aalst, W.M.P. (2013) ‘Business Process Management: A Comprehensive Survey’, *ISRN Software Engineering*, 2013, pp. 1–37. Available at: https://doi.org/10.1155/2013/507984.

Von Rosing, M. *et al.* (2015) ‘Business Process Model and Notation—BPMN’, *The Complete Business Process Handbook*. Elsevier, pp. 433–457. Available at: https://doi.org/10.1016/B978-0-12-799959-3.00021-5.

## Table of Figures