

A Framework for Wiki Selection from a Knowledge Management System (KMS), Software Architecture and Integration Perspective

Luciano Straccia*, Inés Casanovas and María Florencia Pollo-Cattaneo

Universidad Tecnológica Nacional, Buenos Aires, Argentina

Abstract

A wiki enables interaction between a knowledge management system (KMS) and users; it is a fundamental component, but from a systemic, software architecture, and integration perspective, it is part of a more complex technological ecosystem. Selecting the right software to implement a wiki can be challenging. This article proposes a framework for selecting wiki software from the perspectives of KMS, software architecture, and integration. The framework is organized into four elements: domain-specific definition, rules, activities, and a process; and, finally, artifacts. The artifacts propose six dimensions (functional, technical, integration, vendor, economic, and security and governance), 38 criteria and their attributes, along with a wiki selection matrix. These artifacts are applied in an illustrative case study using real tools available on the market.

Keywords

Knowledge Management, Wiki, Selection, Software architecture, Framework

1. Introduction

GEMIS is a research and development group focused on knowledge management. The group has presented models to define the scope of knowledge management systems and their key areas. From a software engineering perspective, previous work analyzed technologies for KMS. The literature on knowledge management presents the wiki as a fundamental element, but it is sometimes presented as a “Best of Suite” product rather than proposing an integrative vision. GEMIS has proposed integrated solutions from the perspective of software architecture and integration. This paper proposes a framework for selecting wiki software from the perspective of a knowledge management system, software architecture, and integration.

2. Methodology and Structure

This work is based on the premise that wikis are widely used in knowledge management, and that a software architecture and integration perspective is necessary for building a Knowledge Management System. The implementation of a KMS depends on identifying software products on the market that enable the KMS to be implemented in a real-world context and organization. The objective of this paper is to propose a framework for selecting wiki software that addresses the needs of a KMS from a software engineering perspective.

This paper presents the literature related to KMS, wikis, and frameworks; the construction of a framework; and an illustrative case study, and it is organized into the following sections: Conceptual Framework and Background (section 3), the discussion (section 4), a Wiki Selection Framework Proposal (section 5), a study case (section 6) and the conclusions (section 7).

ICAIW 2026: Workshops at the 9th International Conference on Applied Informatics 2026, October 13–17, 2026, Orlando, United States

*Corresponding author.

✉ lstraccia@frba.utn.edu.ar (L. Straccia); inescasanovas@gmail.com (I. Casanovas); flo.pollo@gmail.com (M. F. Pollo-Cattaneo)

ORCID 0000-0002-1183-7944 (L. Straccia); 0000-0003-4197-3880 (M. F. Pollo-Cattaneo)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

3. Conceptual Framework and Background

This section presents the conceptual framework and background related to knowledge management (3.1), a complex Knowledge Management System (KMS), the use of wikis (3.2) and the framework elements and structure (3.3).

3.1. Knowledge Management

Knowledge is defined as: “the blend of cognitive and contextualized beliefs, perspectives, judgments, methodologies, information, experiences, and expectations regarding an object, which are adapted and enhanced by the mind of an individual (the knower)” [1]. For Davenport [2], knowledge is a “combination of structured experiences, values, and non-contextual information that provides a framework for evaluating new experiences and information.” Ropain et al. [3] indicate that human intellectual capabilities, cultural values, skills, experience, and even mental models can become a powerful tool for creating value for the company. This reaffirms Davenport’s definition, as it includes individuals’ experiences and values. Pérez and Urbáez [4] analyze various existing definitions of the term “knowledge” and conclude that there are two epistemological camps regarding its nature: “an objectivist perspective, which assumes that knowledge can be a tangible physical resource of the organization, and another perspective based on the idea that knowledge is essentially personal and embedded in individual and organizational practices.”

GEMIS has developed projects related to different aspects of a comprehensive knowledge management system, recognizing that a comprehensive approach requires examining different perspectives and defining views on knowledge management: people, organizational aspects, activities and processes, measurement, and technology [5], with specific contributions to each perspective in various recent works, such as the analysis of the relationship between knowledge management and employee motivation [6], KM activities [7] and measurement models [8]. GEMIS is currently conducting specific analyses on issues related to technology, software architecture for knowledge management [9], and—in particular—the use of wikis, including the presentation of “Descriptors for the Technology Key Area in a Knowledge Management Maturity Model” [10], analyses of technologies, innovation, and, in particular, ontologies [11], and contributions from artificial intelligence [12]. Recently, there has been a particular focus on software architecture’s perspective and the view of wikis as a fundamental element of this perspective [13].

3.2. Knowledge Management System (KMS) and Wiki

A wiki is a technological tool that enables collaboration among team members. The wiki concept was created in 1995 by Ward Cunningham and was originally associated with a content repository [14]. A wiki is a virtual community for communication, documentation, and dissemination of knowledge, becoming a space for the development of online content that allows users to create, edit, and distribute information [15] and allows the accumulation of different contributions over time. Müller et al. [16] analyze the impact of wikis for KM, finding the following aspects: knowledge is free and sharable, content is evolving according to the dynamism of knowledge and its context, few barriers to knowledge use, no definition of knowledge management roles is necessary, the origin of content development is verifiable, and knowledge emerges. Annahi [17] suggests the following issues to be considered for the implementation of a corporate wiki: encouraging employees to participate (building critical mass), building culture (having a collaborative culture), ensuring updating (the wiki must be constantly updated to avoid being outdated), administration (there must be a manager responsible for the content, although all people have access and the possibility of participation) and basic research about the existing wiki software and its selection. GEMIS [13] proposes a Knowledge Management System (KMS) based on Wiki, considering the wikis as the main component from the user’s perspective, and the use of wikis as part of a complex system. The KMS incorporates other components and relationships with external tools, using a Software Architecture Perspective and using an Architecture Description Record

(ADR), a C4 Model, and UML notation. The KMS proposal has a wiki, a component called WIKI+, and external tools. The architecture proposal follows the Component-Based Software Engineering (CBSE) approach and uses a Layer Architecture Pattern, considering the following layers: presentation, access and authorization, knowledge management, taxonomy and knowledge maps, information management, and repository. Therefore, a WIKI is a part of a complex Knowledge Management System, and it requires integration with several components, as a client and as a server. From this perspective, a wiki is relevant for interaction with knowledge management users [18, 19], but the analysis of a complete KMS requires a broader structure.

3.3. Framework Structure

A framework is a “standardized set of concepts, practices, and criteria for addressing a particular type of problem” [20]. Sommerville [21] posits that a successful framework must model a specific problem domain (domain-specific framework), establishing clear boundaries regarding what it covers and what it delegates, and precisely defining [22] its invariant rules and its flexibility. A set of activities and processes is key to implementing a framework [23], and finally, as Schwaber and Sutherland [24] propose, a framework must define artifacts that guide action.

4. Discussion

This section presents some preliminary definitions, such as constraints to consider within the framework (4.1), as well as discussions on general criteria for software selection (4.2), a software selection process (4.3), and specific requirements and criteria for wikis (4.4).

4.1. Preliminary Decision

The IEEE 1062 [25] standard categorizes software according to the following classification: COTS (Commercial-off-the-shelf software), which refers to a commercially available product; MOTS (Modified-off-the-shelf software), which is characterized by being configurable—that is, it is pre-developed, but the vendor has the ability to modify certain features of the software product according to the customer’s requirements; and FD (Fully Developed Software), which refers to custom-developed software. Cloud computing makes it possible to reduce IT costs, improve agility and time to value, and scale more easily and cost-effectively [26], offering scalability and flexibility, cost savings, enhanced collaboration, advanced security, and data loss prevention [27]. Some of these characteristics are also mentioned by [28]. According to [29], cloud computing is an evolution of information technology and a dominant business model for the delivery of IT resources. There are three models (or types of services) of cloud computing that make up the acronym IaaS: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). Given the current market and the objective of considering the wiki as part of a comprehensive knowledge management system, it’s defined as a preliminary decision for the wiki selection framework that the software must be an off-the-shelf product (COTS) and available as cloud services (SaaS).

4.2. Software Selection Criteria

The research group has applied certain guidelines and recommendations from EvaluandoERP—a consulting and market intelligence portal specializing in the evaluation and selection of enterprise resource planning systems in Latin America—to its activities. Although the platform offers methodological guides and diagnostic tools widely used in the corporate sector, its commercial nature—based on brokering and generating leads for software providers—requires a critical evaluation of its data. In the context of this research, the site is considered a useful secondary source of industry reference for contextualizing the regional market offering and obtaining reference criteria.

EvaluandoERP [30] proposes four areas of analysis for software evaluation: functional, technical, vendor, and economic. Based on these areas, a preliminary model is defined, incorporating various contributions. The functional evaluation assesses the software's potential in relation to the organization's business objectives and core processes; the technical evaluation analyzes the state of the technology and how it relates to the technology already in place within the organization; the vendor evaluation assesses the companies providing both the software and the implementation services; and the economic evaluation analyzes the product's pricing [30]. Regarding functional and technological evaluation, [31] combines these into a single category, within which it proposes the following criteria to consider: the ability to work in different languages, cross-platform compatibility, support for multiple users, processes for data backup and recovery, compatibility with email, and support for integrations. Although the authors propose some additional criteria, these pertain specifically to data mining software; therefore, those specific criteria are not considered. For the purposes of this study, these aspects are considered part of the technical axis. Regarding technology, [32] proposes considering the following aspects: accessibility (all users can access the required information), security, customization and integration, scalability, and continuous updates. Finally, [30] incorporates several other aspects related to the technical dimension: maturity of the technology used, good customization options, and a wide range of modules that prevent the need to replace the system as the organization grows and its requirements evolve. It should be noted that this last requirement can be addressed through integration with other tools capable of meeting those requirements. Regarding supplier evaluation, [30] proposes: the supplier's financial stability, broad customer base, and a well-developed distribution channel. [31] further adds that "the supplier's market prospects must be good, because if the supplier struggles, we will end up purchasing a tool that will no longer be supported." In addition, the possible need to change the supplier and/or the tool used must be considered; therefore, the software must be technically capable of migrating data to another tool, either directly or through database access or file export; this aspect is included within the functional criteria. Regarding the economic evaluation, [31] proposes the following criteria: tool costs (prices), hardware costs, licenses, financing policies, and upgrade costs.

For a more complex and comprehensive consolidated analysis model, we are considering incorporating elements from ISO/IEC 25010 (software product quality model), the Red TIC UNAM Guide, and market references such as the Gartner Magic Quadrant and Completeness of Vision.

ISO/IEC 25010:2023 [33] defines a quality model applicable to software products. The current version addresses nine quality characteristics, along with their sub-characteristics, and serves as a reference for specifying, measuring, and evaluating products throughout their life cycle. This selection model provides a common "vocabulary" for transforming broad or imprecise requirements into evaluable criteria. The proposed characteristics are: functional adequacy (the software covers the required functions completely, correctly, and appropriately); performance efficiency (evaluates response times, resource utilization, and capacity); compatibility (evaluates coexistence and interoperability with other systems); interactivity and usability (evaluates ease of learning, operation, accessibility, and user experience); reliability (evaluates availability, fault tolerance, and recovery); security (evaluates information protection and appropriate access based on authorizations); maintainability (ease of analyzing, modifying, testing, and evolving the product); flexibility and portability (adaptation, installation, replacement, and scalability in new environments); and protection and safety (evaluates the prevention of harm to people, the business, systems, property, or the environment, where applicable).

The Red TIC UNAM Guide [34] proposes a consistent process for incorporating reliable, secure, and functional software applications, thereby reducing decisions based solely on perceptions. This process incorporates several ideas regarding criteria to consider. The activities and criteria outlined in the guide are: 1) needs analysis: definition of the problem, scope, stakeholders, and functional/non-functional requirements; 2) identification of alternatives: adoption of an existing solution, external SaaS, commercial or open-source solution, enhancement/migration of an existing system, or custom development; 3) evaluation of alternatives: assessment of requirements, weighting, cost-benefit analysis, and executive summary for decision-making.

Analyst reports do not replace an internal evaluation, but they can serve as input for the supplier assessment process; that is why it is important to consider relevant market reports. The Gartner Magic

Quadrant [35] positions vendors along two main dimensions: Ability to Execute and Completeness of Vision. The Forrester Wave compares vendors based on transparent criteria regarding their current offerings and strategy, with the option to tailor the evaluation to the buyer's priorities. In both cases, it is best to use them as an external reference for compiling a shortlist or analyzing supplier risk, not as an automatic final decision.

Finally, for this paper, considering the previous works, a preliminary five-dimension model is considered. The main change consists of separating Security and Governance into their own dimension, given that identity, permissions, auditing, compliance, and data protection typically cut across the entire decision-making process and, in many cases, can serve as deal-breakers. The identified dimensions are: functional (what the tool does and to what extent it covers the expected processes, use cases, and workflows), technical (how it is built, how it is deployed, how it scales, and how it integrates with the existing ecosystem), vendor (who is behind the solution, what capacity they have to support it, and what support they offer), economic (how much it costs to purchase, implement, operate, maintain, and evolve the tool throughout its lifecycle), and security and governance (how it protects data, identities, permissions, traceability, compliance, and operational continuity). Details on each dimension are presented in the section on the proposed framework in Section 5.

4.3. Software Selection Process

The software selection process model proposed by de Keravenant [36] consists of three phases: identifying the organization's requirements, identifying the characteristics of tools available on the market, and evaluating and selecting the best solution. The software acquisition process defined in the IEEE 1062 reference framework (IEEE Recommended Practices for Software Acquisition) [25] proposes various activities associated with the entire acquisition process, with the following being particularly relevant to the selection phase: determining software requirements (including the identification of criteria), identifying potential tools and vendors, and evaluating proposals and selecting the vendor. Meanwhile, PM [37] proposes compiling tool requirements (step 1), identifying candidate vendors (step 2), determining knockout criteria (step 3), creating a vendor shortlist (step 4), creating a tool selection criteria matrix (step 5) allowing the final decision, and collecting vendor information for acquisition and contract. To optimize the identification of candidate vendors, the framework proposal will consider the knockout criteria prior to identifying the candidate vendors.

4.4. Wiki Requirements

This section presents some specific requirements for the selection and implementation of wikis related to usability, collaborative editing and version control (4.4.1), data analytics (4.4.2), and integration (4.4.3).

4.4.1. Usability, Collaborative Edition and Version Control

The wiki pages are edited directly from the browser, where the users themselves create, modify, correct, or delete content that is usually shared by any other user. Wikis have the following fundamental principles: anyone can modify, they are flexible (they do not have a predefined structure); and the content has “no time and no end”, that is, it is timeless, it is never finished, and is being modified by new ideas [13]. For Cunningham [14], the principles of a wiki are simplicity, integration, being an open system, and allowing exponential growth.

Wikis can use markup/Markdown or incorporate WYSIWYG editors (“What You See Is What You Get”). WYSIWYG allows users to edit content in a more accessible visual format and view the generated content versions directly in the user interface. While the former may result in content creation being limited to engineers, the latter option enables standard users to edit content. In line with the approach presented in [13], the user interface is a very important aspect of a wiki.

Among the specific aspects that can be incorporated into the analytical frameworks proposed in the software selection criteria are concurrent editing, version comparison, version restoration, comments, and page history (functional dimension). In addition, a special analysis of integration aspects (technical

axis) and data analytics is presented below. [7] highlights the importance of processing data, performing data analytics, and refining knowledge; to this end, it is essential to have strategies for accessing information (based on integration strategies) and page history. Page history, in addition to security and traceability, enables the maximization of a knowledge management system's function.

4.4.2. Data Analytics

Data analytics enables the generation of insights that provide information and offer solutions to support organizations' analytical processes [38]. It is the process of discovering, interpreting, and communicating significant patterns present in the data [39]. Páez García [40] states that analytical techniques help ensure the viability of knowledge management.

According to [41], knowledge creation depends on the method of generating new knowledge through the repositioning and integration of existing knowledge to achieve objectives such as the introduction of a new product, the development of a service, or the improvement of management within an organization. For [41, 42, 43], knowledge creation arises from knowledge that already exists within the organization, setting in motion the spiral of organizational knowledge creation [44]. Dabic [45] presents a process of “conversion-oriented knowledge management” that refers to processes aimed at making existing knowledge useful. Justiniano Advíncula [46] proposes including a classification process that allows for filtering whether knowledge is relevant to the organization. [47] includes knowledge refinement. These aspects are presented in the KM activities outlined in [7], primarily within the activities of structuring, monitoring, and creation.

Data analytics and the pursuit of knowledge—for the purpose of creating new knowledge and for classification, filtering, refinement, or other related activities—must be considered in wikis, since their absence would severely limit the likelihood of success for knowledge management in the medium term.

In general, wikis do not include data analytics, knowledge search, and other similar features—as described in the preceding paragraphs—among their built-in functionalities. This gives rise to the need for them to be capable of being integrated with other tools, that is, to enable other tools to retrieve data from the wiki to carry out the functionalities described above. It is worth reiterating that, according to Cunningham, the principles of a wiki involve integration, being an open system, and allowing for exponential growth. Even if data retrieval were possible, the ability to write new content on the wiki must also be ensured to support the spiral of knowledge. To achieve the intended goal, this integration can be carried out using any of the existing integration strategies or integration patterns: data-centric, middleware, service-based, etc.

4.4.3. Integration

Considering the importance of implementing various activities within a knowledge management system and the value of software integrations, the following integration categories relevant to wikis are defined [13]:

- Data analytics tools: enable the visualization of usage patterns, identification of relevant or obsolete content, and the measurement of contributions. Allow evaluating the evolution of organizational knowledge and supporting continuous improvement processes.
- Knowledge classification and processing tools: facilitate the refinement, filtering, and restructuring of knowledge. They enable semantic classification, the identification of redundancies, and the extraction of tacit knowledge through NLP or text mining techniques.
- Document management systems: these complement wikis with rigorous version control, document approval workflows, and regulatory compliance.
- Automation and synchronization: task automation and synchronization with other platforms.
- Communication and notification tools: these ensure that knowledge shared on the wiki remains active and is disseminated to users in real time.
- Project and task management tools: these enable traceability between documented knowledge and resulting actions, integrating knowledge with operational execution.

The technical dimension defined in 4.2 includes aspects of integration and compatibility related to notifications and communication. For the specific selection of wikis, it is considered important to establish a new criterion for software selection called “integrations,” which is presented in the framework in Section 5.

5. Framework for Wiki Selection Proposal

Based on the framework structure presented in Section 3.3 and the discussions presented in Section 4, this section proposes a framework for wiki selection with its elements: domain detail (section 5.1), rules and knockout criteria (5.2), activities and process (5.3) and the artifacts (5.4).

5.1. Domain Definition

This framework focuses on selecting Wiki software, viewing the Wiki as part of a Knowledge Management System and from the perspective of software architecture and integration.

5.2. Rules and Knockout criteria

Based on the definitions presented in this paper, the following are considered knockout criteria: 1) it must be COTS, 2) it must be SaaS, 3) it must allow for the creation of integrations independent of those included in the out-of-the-box product, 4) it must have a WYSIWYG editor, 5) it must allow for data export either via the user interface or via API, 6) page history is required, and 7) It must allow users to create a private wiki where they can manage profiles and users.

5.3. Activities and Process

Based on the Software Selection Process proposed in Section 4.3, the process includes the activities presented in Figure 1: propose compile tool requirements (step 1), determine knockout criteria (step 2), identify candidate vendors (step 3), create vendor shortlist (step 4), create tool selection criteria matrix (step 5) and, finally, the final decision and collect vendor information for acquisition and contract.



Figure 1: Wiki Selection Process.

5.4. Artifacts

To construct the selection matrix, this framework proposes the selection dimensions (5.4.1), the criteria included in each dimension (5.4.2), and the evaluation attributes (5.4.3). Finally, the complete selection matrix is presented (5.4.4).

5.4.1. Wiki Selection Dimensions

As mentioned earlier, given the importance of integrations for a comprehensive wiki-based knowledge management system, it is considered important to incorporate the integration dimension into the software selection criteria proposed above. For this integration dimension, the criteria associated with the integration categories mentioned in 4.4 will be considered. Thus, the resulting axes are: 1) functional; 2) technical; 3) integration; 4) vendor; 5) economic; 6) security and governance as presented in Figure 2.

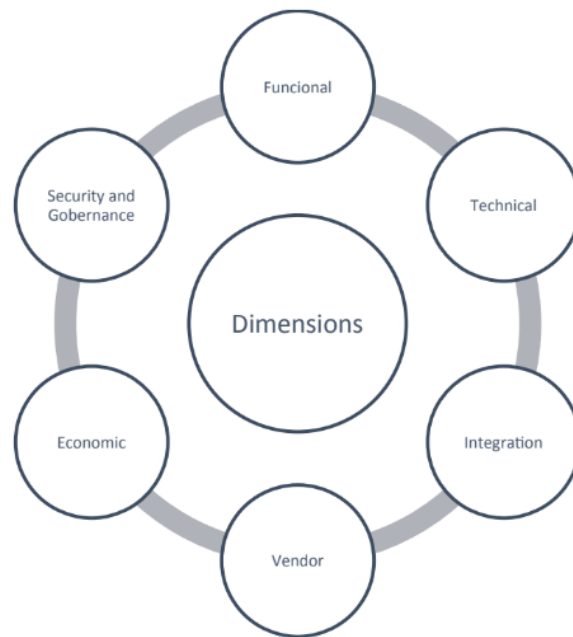


Figure 2: Dimensions for Wiki Selection.

5.4.2. Wiki Selection Criteria

Based on the criteria analyzed in 4.2 and the customizations required as outlined in this section, Table 1 presents the criteria for the functional dimension, Table 2 the criteria for the technical dimension, Table 3 the criteria for the integration dimension, Table 4 the criteria for the vendor dimension, Table 5 the economic criteria, and Table 6 the criteria for security and governance.

Table 1

Criteria for functional dimension.

#	Criteria	Description
1	Usability / UX	Evaluates the ease of learning and use for end users
2	Multi-language	Reviews support for multiple languages in both the interface and the data
3	Multi-platform / multi-device	Determines availability across different systems, browsers and devices
4	Collaborative Editing & Version Control	Enables join work with history and traceability
5	Data Import / Export	Facilitates the loading and extraction of data in reusable formats
6	Reporting & Information Outputs	Analyzes the capability to generate required queries, reports, or dashboards

Table 2

Criteria for technical dimension.

#	Criteria	Description
1	Deployment Model	Defines architecture such as Cloud, On-Premises, or Hybrid
2	Customization	Evaluate adaptation without costly custom development
3	Scalability	Examines the capability to grow in users, data, etc.
4	Technological Maturity	Reviews the stability and relevance of the technology stack
5	Maintainability	Assesses the ease of maintenance and continuous evolution
6	Continuous Update	Analyzes the frequency and method of releases/patches
7	Backups and Recovery	Evaluates backup and restoration capabilities

Table 3

Criteria for integration dimension.

#	Criteria	Description
1	Connectivity Mechanisms	Validates the availability of APIs, Webhooks, and native connectors
2	Documentation and Community	Reviews the clarity of technical manuals, code guides, and available support for developers
3	Rate Limits	Controls call volume restrictions and potential costs associated with intensive connection use
4	Security and Fall Management	Verifies authentication protocols, encryption, and the availability of logs to trace errors

Table 4

Criteria for vendor dimension.

#	Criteria	Description
1	Financial Solvency	Determines the vendor's economic capacity to sustain the product
2	Client Base & Success Cases	Reviews presence in comparable organizations
3	Distribution Channel & Partners	Measures local or regional implementation capabilities
4	Support & SLA	Evaluates the quality of post-sales support and assistance
5	Training and Documentation	Reviews of available resources for adoption and operation
6	Roadmap & Continuity	Analyzes the future evolution of the product
7	Market Positioning	Examines external references regarding vendor presence and market direction

Table 5

Criteria for the economic dimension.

#	Criteria	Description
1	Licensing / Subscription	Represents the direct cost of software usage
2	Hardware / Infrastructure	Includes the resources required to run the system
3	Implementation	Cost of configuration, migration and deployment
4	Maintenance & Update	Convers recurring support and evolution costs
5	Training	Includes the cost of training users and administrators
6	Total Cost of Ownership	Calculates the overall cost over a defined timeframe
7	ROI and Expected Benefits	Analyzes the ratio of benefits to costs
8	Financing & Commercial Terms	Defines payment structures and exit terms

5.4.3. Wiki Selection Attributes

Due to space constraints, the attributes corresponding to each criterion are not detailed in this text; each attribute can be found in the selection matrix on the wiki, which is presented in Section 5.4.4.

5.4.4. Wiki Selection Matrix

Based on the dimensions, criteria, and attributes presented, the Wiki Selection Matrix was developed, and the spreadsheet was made available¹.

6. Case Study

This section apply the Framework for Wiki Selection in a specific context and domain (presented in Section 6.1), following the process presented in Section 5.3: proposes compile tool requirements (step 1, Section 6.2), determine knockout criteria (step 2, Section 6.3), identify candidate vendors (step 3, in 6.4),

¹Available in www.grupogemis.com.ar/docs/wikiselectionmatrix.xlsx

Table 6

Criteria for the security and governance dimension.

#	Criteria	Description
1	SSO, OIDC and SAML	Ensures centralized authentication and single sign-on
2	OAuth and API Security	Requires secure authentication mechanisms for integrations
3	Permissions and Roles	Provides granular access control over actions and data
4	Encryption	Guarantees data protection in transit and at rest
5	Auditing and Traceability	Maintains a verifiable log of actions
6	Regulatory Compliance	Evaluates alignment with applicable standards and legal regulations

create vendor shortlist (step 4, in 6.5), create tool selection criteria matrix (step 5, in 6.6) and, finally, the final decision and collect vendor information to acquisition and contract.

6.1. Context

The context of application influences technological decisions and the presentation of the case, given that the feasibility of implementing a technological system depends on the resources that are available. In particular, the research group has focused its efforts on small and medium-sized enterprises (SMEs) in Argentina, considering their characteristics and challenges, and this work is part of the Research and Development Project titled “Creation of a Comprehensive Knowledge Management Model for Argentine SMEs.” Some of the problems encountered in this type of enterprise include an excessive level of management centralization in the business owner, levels of incremental investment that do not allow for an adequate level of technological updating, and low demand for support services [48]. There are also deficiencies in management, problems related to a lack of collaboration, negative influences from environmental variables, and a lack of credit and government support. Difficulties in accessing technology are highlighted in various studies [49]. In fact, many organizations do not have technology management teams. A recent study conducted in Spain [50] identifies access to technology as a problem for nearly one-third of small and medium-sized organizations, along with the desire for greater promotion of digital transformation, facilitation of cooperation processes, and stronger ties with other organizations—all aspects linked to knowledge management. Similarly, Colombia faces challenges in this regard, as standards for acquisition and development are geared toward large companies with different realities than the country’s small organizations, and a software model tailored to these organizations is needed [51]. Considering the context, the selection should also include a knockout criterion related to the user and data language interface: it must be in Spanish.

6.2. Compile Tool Requirements

This wiki selection considers the requirements related to wikis presented in Section 4.4.

6.3. Determinate Knockout Criteria

This wiki selection considers the knockout criteria presented in Section 5.2. Also included as knockout criteria is the language interface: it must be in Spanish. Given the context of this case, a maximum acceptable cost is also defined, with a maximum acceptable cost of \$30 per month per user for this illustrative example.

6.4. Identify Candidate Vendors

A market research phase aimed at broadly exploring the universe of available solutions. Through reviews of industry literature, specialized directories, and Requests for Information (RFIs), a long list of

vendors offering products within the required technology category is compiled. To quickly identify potential wikis for analysis, we use the Wikimatrix website, which specializes in comparative analysis of wikis. Using this website and applying certain fundamental knockout criteria, it is possible to conduct a rapid analysis of 82 wikis to select candidates for a full analysis. The aspects that Wikimatrix allows us to analyze through its Choice Wizard are presented in Table 7 and it is related to page history (step 1), WISIGWYG (step 2), Commercial Support (step 3), interface language (step 4), host (step 5), domain name (step 6) and corporate branding (step 7).

Table 7

Wikimatrix Choice Wizard Criteria.

Step	Option Selected	Source
Page History	Yes, a page history is needed	Knockout Criteria
WISIGWYG	Yes, WYSIWYG is crucial for me	Knockout Criteria
Commercial Support	No, I don't need Commercial Support	Not required
Interface Language	Spanish	Knockout Criteria
Host	I would like my Wiki to be hosted by someone else	Knockout Criteria
Domain Name	No, I don't need this	Not required
Corporate Branding	No, I don't need this	Not required

Based on the results obtained, Wikimatrix proposes the following wikis: Confluence, Xwiki and Zoho Wiki. In addition, other well-known wikis with a larger market share—such as Notion and Guru—are included in the analysis; together with Confluence, these are the ones with the largest market share in SaaS. Furthermore, some reports suggest considering Slab and Slite; however, Slab is not available in Spanish. Later a technical document from the research group was reviewed; the document analyzes technical aspects of various wikis: BookStack, Confluence, Docu Wiki, Document360, MediaWiki, Miraheze, Moin Moin, Notion, Nuclino, Outline, PMWiki, Slab, Slim Wiki, TikiWiki, TiddlyWiki, Vultr, Wikia, Wiki Hub, Wiki.js, XWiki, and Zoho Wiki. Of these, it was decided to include as candidates those not included in the quick analysis with Wimimatrix: Document360, Outline, Notion, and Slim Wiki.

Finally, the possibility of using GitHub was considered; it was ruled out for the reasons outlined below. Most wikis explicitly present their service as a tool for creating a knowledge base designed to be collaborative in real time, allowing multiple users to make changes to a document simultaneously, thereby aligning the service with the project's use cases and objectives. In contrast, GitHub's documentation explicitly presents the service as a management tool for software development and version control. The concepts of “commit” and “push” differ significantly from the collaborative contributions a knowledge base would require, since a commit is geared toward modifying code or software functionality within the repository. Although contributions in the form of commits followed by a push are reviewed before being accepted (merged) or rejected, the repository's primary purpose is to store functional software. The platform is not geared toward heuristics and knowledge. To make contributions, GitHub users who cannot directly edit the repository's content must do so via issues. An issue is not linked to a specific file (unlike a comment on a wiki); it is linked to a repository. This is designed as it is intended to address problems in software projects, where it would be necessary to reference multiple files.

Finally, the candidate vendors are Confluence, Xwiki, Zoho Wiki, Notion, Guru, Slite, Document360, Outline, and Slim Wiki.

6.5. Create Vendor Shorlist

In this convergent filtering step, the knockout criteria are rigorously applied to the initial longlist. As a result, the analysis pool is narrowed down to a qualified group of finalists (typically three to five vendors) that warrant a detailed technical and commercial evaluation.

SlimWiki is a minimalist, lightweight knowledge management platform for teams, designed to document processes, guides, and internal information using a simple editor, version history, and access control. It focuses on ease of use and clean navigation without the complexity of traditional corporate wikis; however, its official documentation and plan features do not mention the existence of a public API or an open extension system, so it would not be possible to perform external integrations or custom development beyond the native connections officially offered by the platform itself, such as its tools for Slack and Google. Given that the anticipated integration requirements were considered a knockout criterion, this wiki is excluded from the final analysis. Meanwhile, Document360 is a wiki tool that features a powerful Google-like search function for navigation that makes searching easier; it includes version control; offers integration with third-party applications such as Google Analytics, Segment, and Olark; and is cloud-based. Although it does not offer direct integration, it allows access via the API. Its cost is high compared to the other tools, at \$599 per project per month for 10 users, resulting in a minimum of \$59 per month per user—a cost that may be prohibitive for organizations depending on the context of the case study.

After applying the exclusion criteria, a vendor shortlist is defined, which can be found in Figure 3.

Candidate Vendors	Excluded	Vendor Shortlist
Confluence XWiki Zoho Wiki Notion Guru Slite Document 360 Outline Slim Wiki	Slimi Wiki Document 360	XWiki Confluence Notion Guru Slite Outline Zoho Wiki

Figure 3: From Candidate Vendors to Vendors Shortlist.

6.6. Create Tool Selection Criteria Matrix

To create the criteria matrix, the plans proposed by each of the identified vendors were reviewed. Based on the dimensions, criteria, and attributes presented—and considering a specific weight for each attribute—the “Wiki Selection Matrix” was completed, and the spreadsheet was made available². Each attribute is evaluated using a predefined ordinal scale and assigned a relative weight according to its relevance to the organizational context. Criterion and dimension scores are calculated as weighted aggregations to their corresponding attributes, and the final score is normalized to 0-100% scale.

AI-assisted analysis was used to support the preliminary completion of the matrix. The resulting values were manually reviewed and adjusted by the researchers. Therefore, the case study should be understood as an illustrative application of the proposed framework rather than as a definitive vendor evaluation.

After running the entire process using the values from the wiki selection matrix, the results shown in Figure 4 were obtained.

²Available in www.grupogemis.com.ar/docs/wkmit-2026-wikiselectionmatrix-studycase.xlsx. All details are available in www.grupogemis.com.ar/docs/wkmit-2026-wikiselectionmatrix-studycase.zip

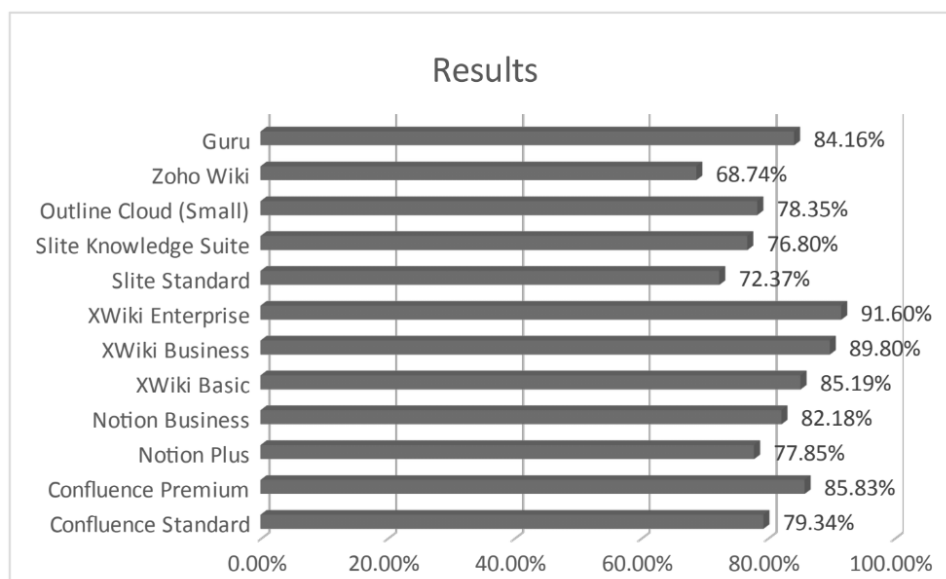


Figure 4: Study Case Results.

7. Conclusion

This paper proposes a framework for wiki selection from the perspective of a Knowledge Management System (KMS), software architecture, and integration. The article proposes a framework consisting of a definition of the domain scope, rules, and knockout criteria, as well as a set of activities that make up a process and its artifacts. The definition of artifacts includes a proposal for dimensions of analysis, their criteria, and attributes for each criterion, resulting in a wiki selection matrix. The proposed framework focuses on six dimensions: functional, technical, integration, vendor, economic, and security and governance. An application case study is presented, highlighting the importance of contextual specifics and their impact on the definition of knockout criteria. The parity in the results presented also demonstrates the importance of greater granularity in the analysis of attributes and their weighting in the matrix. Finally, the different licensing models proposed by the tools highlight the need for a specific analysis of cost curves in relation to the number of users.

Acknowledgements

This article was written in collaboration with Brenda Baruzzini, a researcher at the Universidad Tecnológica Nacional (UTN), who contributed to the discussion on the development of artifacts, and with Francisco Markiw and Tadeo Sorrentino, Information Systems Engineering students at UTN, for their analysis of the technical aspects of the integration.

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini to support the preliminary completion of the selection matrix values. The resulting values were manually reviewed and adjusted by the researchers. Therefore, the case study should be understood as an illustrative application of the proposed framework rather than as a definitive vendor evaluation.

References

- [1] M. T. R. Díaz, J. J. G. Millán, Gestión del conocimiento y capital intelectual, a través de modelos universitarios, *Económicas Cuc* 34 (2013) 85–116.
- [2] T. H. Davenport, Knowledge management and the broader firm: strategy, advantage, and performance, *Knowledge management handbook 2* (1999) 1–2.
- [3] E. Ropain, L. Amaya, E. Arroyo, Factores de la gestión del conocimiento en el desarrollo de software, 3. GESTIÓN DEL RIESGO TECNOLÓGICO EN EL USO DE LA ENERGÍA FOTOVOLTAICA EN VENEZUELA..... 4 (2014) 520.
- [4] V. A. Pérez, M. F. Urbáez, Modelos teóricos de gestión del conocimiento: descriptores, conceptualizaciones y enfoques, *Entreciencias: diálogos en la Sociedad del Conocimiento* 4 (2016) 201–227.
- [5] L. Straccia, M. Pollo-Cattaneo, A. Maulini, Visões da gestão do conhecimento para a construção de um modelo integral, in: *Proceeding in Congresso Brasileiro de Gestão do Conhecimento*, São Paulo, Brasil, 2022, pp. 19–22.
- [6] F. Nocenti, L. Straccia, N. Gonzalez, Impacto de la gestión del conocimiento sobre la motivación de los/as trabajadores/as, 2026.
- [7] L. Straccia, M. F. Pollo-Cattaneo, A. Maulini, Knowledge management model: a process view, in: *International Conference on Computational Science and Its Applications*, Springer, 2023, pp. 599–616.
- [8] L. Straccia, M. F. Pollo Cattaneo, A. Maulini, C. Berro, Gestión del conocimiento: actividades, tecnología y modelo de medición, 2024.
- [9] L. Straccia, M. F. Pollo-Cattáneo, M. Giorda, M. G. Bongiorno, A. Maulini, Architecture on knowledge management systems: Its presence in the academic literature, in: *International Conference on Applied Informatics*, Springer, 2022, pp. 411–423.
- [10] L. Straccia, M. F. P. Cattaneo, Descriptors for technology key area in a knowledge management maturity model., in: *ICAI Workshops*, 2023, pp. 135–149.
- [11] L. Straccia, M. F. P. Cattaneo, Knowledge management: Innovation, technology and ontologies., in: *ICAI Workshops*, 2024, pp. 264–279.
- [12] M. E. Ansalas, L. Straccia, C. Vegega, Aportes de la inteligencia artificial a la gestión del conocimiento en el ámbito empresarial, 2026.
- [13] L. Straccia, B. Baruzzini, B. Alegre, M. F. P. Cattaneo, A knowledge management system based on wikis from a software architecture perspective., in: *WAAI/AIESD/WDEA/WKMIT/SCTSD/WSM*, 2025, pp. 233–245.
- [14] B. Leuf, W. Cunningham, *The Wiki way: quick collaboration on the Web*, Addison-Wesley Longman Publishing Co., Inc., 2001.
- [15] L. B. Angarita, Librería de activos para la gestión del conocimiento sobre procesos software: PAL-Wiki, Ph.D. thesis, Universidad Carlos III de Madrid, 2010.
- [16] C. Müller, P. Dibbern, Selbstorganisiertes wissensmanagement in unternehmen auf basis der wiki-technologie-ein anwendungsfall, *HMD-Praxis der Wirtschaftsinformatik* 252 (2006) 45–54.
- [17] C. ANNAHI, ótimas dicas para um wiki de sucesso, *Portal gestão de pessoas e tecnologia da informação* (2015).
- [18] Á. L. M. Cerdán, Tecnologías de información y gestión del conocimiento: integración en un sistema, *Economía industrial* (2005) 107–116.
- [19] D. Pérez González, M. Dressler, et al., Tecnologías de la información para la gestión del conocimiento, 2007.
- [20] A. Angelini, Desafiemos la gravedad: la agilidad como un camino de rebeldía organizacional, 2025.
- [21] I. Sommerville, M. Fowler, K. Beck, J. Brant, W. Opdyke, D. Roberts, *Edition: Software engineering*, Instructor (2019).
- [22] W. Pree, *Design patterns for object-oriented software development*, ACM Press/Addison-Wesley Publishing Co., 1995.
- [23] E. Evelina, G. Pia, H. David, M. von Württemberg Liv, R. F. Waldo, *Process improvement framework*

- evaluation, in: 2010 International Conference on Management Science & Engineering 17th Annual Conference Proceedings, IEEE, 2010, pp. 319–326.
- [24] K. Schwaber, J. Sutherland, The scrum guide, Scrum. org 19 (2020).
 - [25] I. of Electrical, E. Engineers, IEEE Recommended Practice for Software Acquisition: IEEE Std 1062: 1998 Edition: Includes IEEE Std 1062-1993 and IEEE Std 1062a01998, IEEE, 1998.
 - [26] IBM, Cloud computing, Online, 2026. URL: <https://www.ibm.com/es-es/topics/cloud-computing>.
 - [27] Google Cloud, Advantages of cloud computing, 2026. URL: <https://cloud.google.com/learn/advantages-of-cloud-computing?hl=es-419>.
 - [28] S. E. Mamani Luque, Retos y oportunidades del cloud computing, Revista de Información, Tecnología y Sociedad (2012) 88.
 - [29] H. Nigro, Cloud computing: retos y oportunidades, Revista Ingeniería, Matemáticas y Ciencias de la Información 9 (2022) 11–16.
 - [30] Evaluando ERP, Evaluación y selección de un software erp, 2026. URL: <https://www.evaluandoerp.com/>.
 - [31] P. V. Britos, E. Fernández, M. A. Ochoa, H. Merlino, E. Diez, R. García Martínez, Metodología de selección de herramientas de explotación de datos, 2005.
 - [32] LatamReady, ¿qué tomar en cuenta para elegir un erp?, 2005.
 - [33] International Organization for Standardization, ISO/IEC 25010:2023 – systems and software engineering – SQuaRE – product quality model, 2023. URL: <https://www.iso.org/standard/78176.html>.
 - [34] Red TIC – UNAM, Guía para la evaluación y selección de soluciones de software, 2024. URL: https://www.red-tic.unam.mx/recursos/2024/2024_Guia_EvaluacionSW.pdf.
 - [35] Gartner, Magic quadrant faq, 2026. URL: <https://www.gartner.com/en/about/magic-quadrant-faq>.
 - [36] A. M. de Keravenant, P. Pytel, M. F. Pollo, Selección y evaluación de una solución software de gestión logística para una operadora en el rubro farmacéutico, DESARROLLO E INNOVACIÓN EN INGENIERÍA (2017) 8.
 - [37] ProjectManagement.com, Stage dts-dw tool selection, 2026. URL: <https://www.projectmanagement.com/content/processes/9238.cfm>.
 - [38] R. P. Astonitas, Í. M. Ramírez, C. R. Campos, R. C. S. C. Acosta, M. C. Alarcon, Analítica de datos para la gestión del conocimiento en la universidad nacional toribio rodríguez de mendoza de amazonas, Revista de la Universidad del Zulia 13 (2022) 76–84.
 - [39] Oracle, What is analytics?, 2026. URL: <https://www.oracle.com/ar/business-analytics/what-is-analytics/>.
 - [40] G. J. Páez García, Aplicación de técnicas de análisis de datos para obtener líneas de investigación específicas para el ecuador. caso de estudio: Computer science en scopus, 2019.
 - [41] A. A. H. Harandi, M. B. Nia, C. Valmohammadi, The impact of social technologies on knowledge management processes, Kybernetes 48 (2018) 1731–1756.
 - [42] I. Shehabat, The role of knowledge management in organizational performance and gaining sustainable competitive advantage, in: Proceedings of the 2020 Asia Service Sciences and Software Engineering Conference, 2020, pp. 133–139.
 - [43] J. Raudeliuniene, E. Albats, M. Kordab, Impact of information technologies and social networks on knowledge management processes in middle eastern audit and consulting companies, Journal of Knowledge Management 25 (2021) 871–898.
 - [44] I. Nonaka, H. Takeuchi, The knowledge-creating company, Harvard business review 85 (2007) 162.
 - [45] M. Dabic, T. Kiessling, The performance implications of knowledge management and strategic alignment of mnc subsidiaries, Journal of Knowledge Management 23 (2019) 1477–1501.
 - [46] L. M. Justiniano Advincula, Comportamiento de equipos de trabajo y gestión del conocimiento en las oficinas dependientes del vicerrectorado académico de la universidad nacional agraria de la selva, 2020.
 - [47] E. Hernández Prieto, Y. Sánchez Reyes, Diseño de un modelo de gestión del conocimiento para el departamento de comunicaciones del ejército nacional de colombia., 2021.
 - [48] P. G. Iavarone, R. N. LETTRY, Costos por órdenes de producción: su aplicación a la industria panificadora, Provincia de Mendoza-Argentina (2012).

- [49] E. E. Zapata Guerrero, Las pyme y su problemática empresarial. análisis de casos, Revista Ean (2004).
- [50] J. A. Clemente-Almendros, M. Blanco-Hernández, I. Pastor Sanz, Estudio de la Pequeña y Mediana Empresa (Pyme) 2024 Gestión del TALENTO en la PYME en la Región de La Rioja. Un análisis comparativo con España., Universidad Internacional de La Rioja, 2024.
- [51] M. Orozco Bohórquez, Guía metodológica de adquisición de software para pequeñas y medianas empresas del sector privado, 2010.