Enterprise Model for Requirements Management: Evaluation of Capability Gaps in Jira and Marketplace Applications

Agnese Rozenberga[1,*], Marite Kirikova[1,*]

[1]Institute of Applied Computer Systems, Riga Technical University, Kipsalas iela 6A, Riga LV-1048, Latvia

Abstract

Effective requirements management (RM) is pivotal for aligning projects with business objectives, yet enterprises frequently face capability gaps in their tool support. Jira is widely adopted for project management, but its native functionality does not fully address the RM lifecycle. This paper addresses the research question: What RM capability gaps can be recognized in Jira and in Marketplace applications designed for RM, and how can they be evaluated in a consistent, repeatable manner? An enterprise model, grounded in the 4EM framework, is proposed to systematically map tool components to RM lifecycle stages and assess their coverage against defined RM functions. The evaluation demonstrates that Jira provides only partial support across most RM activities, while Marketplace applications extend coverage but remain uneven and incomplete when considered individually. The contribution is an RM lifecycle-anchored diagnostics approach based on the evaluation enterprise model that offers a consistent way to surface RM capability gaps, guide evidence-based tool assessment, and support ongoing evaluation as tools evolve.

Keywords

Requirements management, Enterprise modeling, Jira capability gaps in requirements management

1. Introduction

Requirements management (RM) is intended to be a systematic activity within enterprises, ensuring that project needs are captured and maintained across the lifecycle. As defined in the PMBOK® Guide, "Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements" [1], emphasizing that RM forms a critical part of the project lifecycle and must be embraced by project management platforms. However, not all widely used project management platforms are capable to meet the needs of RM.

One such platform is Jira [2], which provides users with tools to effectively plan, track, collaborate, launch, and report on project progress. It has gained considerable popularity, with more than 300,000 companies worldwide using Jira to manage their projects [3].

Jira software is not an RM system and has limited RM capabilities [4]. To use Jira for RM, Atlassian recommends [5] using it in conjunction with Confluence. Confluence [6] is a platform for creating, organizing, and sharing project documentation. In this study, Confluence was not analyzed as part of the RM toolset, as it is primarily positioned and utilized as a documentation platform rather than an RM solution.

To extend the functionality of Jira, Jira Marketplace [7] applications can be used. In this paper, Marketplace applications are referred to as *Marketplace tools* for consistency of terminology throughout. The Jira Marketplace provides tools that integrate with Jira to extend its functionality and support project-specific needs. This paper examines Jira Marketplace tools with a focus on RM, specifically, looking at their functional capabilities.

PoEM2025: Companion Proceedings of the 18th IFIP Working Conference on the Practice of Enterprise Modeling: PoEM Forum, Doctoral Consortium, Business Case and Tool Forum, Workshops, December 3–5, 2025, Geneva, Switzerland

*Corresponding authors.

agnese.rozenberga@edu.rtu.lv (A. Rozenberga); marite.kirikova@rtu.lv (M. Kirikova)

0009-0001-9256-6938 (A. Rozenberga); 0000-0002-1678-9523 (M. Kirikova)

Acknowledging that (i) companies often use Jira for project management, (ii) RM is an essential constituent of project management, and (iii) Jira is not specifically designed for RM, the problem to be solved is how to evaluate which Marketplace tools would be helpful for a project. In this paper, the evaluation of the Jira platform and Marketplace tools is considered as an enterprise. The following research question was proposed: *"How can requirements management capability gaps be recognized in Jira and in Marketplace applications designed for RM, and how can they be evaluated in a consistent, repeatable manner?"*

To answer the research question, the paper employs enterprise modeling, which can be utilized for various purposes, such as visualizing the current situation, identifying capability gaps, and supporting decision-making [8]. By grounding the evaluation model in enterprise modeling principles of a 4EM model [8], the study ensures that the analysis of Jira and Marketplace tools is both systematic and adaptable to future developments. Enterprise in this research refers to the organizational setting in which Jira and its Marketplace tools are evaluated by mapping their components to RM stages, so that capability gaps can be identified and the potential impact of supplementary tools on the RM process can be revealed.

To find the capability gaps, Jira components and their functionality were mapped to the RM stages and respective requirement classes. The same mapping can be done with any Jira-compatible tool from Jira Marketplace, and the comparison of these mappings reveals the potential impact of the use of the tool on the RM process. Acknowledging existing shortcomings of platforms and tools is important, as these limitations may manifest in negative ways throughout the project lifecycle if left unaddressed. Given that Jira and its Marketplace tools are continuously evolving, it is essential to develop an evaluation approach that remains applicable over time.

The structure of this paper is organized to address the research question as follows: Background is briefly discussed in Section 2, Section 3 examines the expected functionality of requirements management tools; Section 4 analyzes functional capability gaps in Jira and its Marketplace tools; and finally, Section 5 presents an enterprise model for systematic requirements management gap analysis.

2. Background

The background used in this research roots in research of RM tools and Enterprise Modeling.

Research on RM tools has taken two complementary directions. The first direction concerns structured feature analyses that have sought to classify what functions tools provide and, in some cases, how they map onto the stages of the RM lifecycle. A key contribution in this stream is Carrillo-de-Gea et al. [9], who established a feature taxonomy spanning elicitation, analysis, specification, validation, change, and traceability, and used it to benchmark tool coverage. The second direction concerns comparative and empirical studies that have evaluated RM tools along broader dimensions or in real-world contexts. Özkaya et al. [10], for instance, compared 56 tools across collaboration, customization, interoperability, and project-management integration, confirming uneven support particularly for model-based specification and end-to-end interoperability. Case-based studies show how general-purpose platforms are adapted to RM use: Filion and Daviot [4] describe a large-scale Jira deployment where issue tracking and workflow were effective, but plugins were necessary to reach acceptable levels of specification, traceability, and test linkage. Broader empirical studies of RE practice (e.g., Wagner et al, [11]; Kasauli et al., [12] echo this pattern, showing that teams often bend Jira into RM roles, but coverage gaps remain.

Rozenberga, A. [13] provides a detailed functional comparison of Jira-compatible RM tools and an interactive Jira RM tool selector. This paper extends that research and contributes an approach for an enterprise-model-based tool capability gap analysis that generalizes across versions and supports repeatable evaluations. The proposed approach extends the previous research by introducing an enterprise-model-based evaluation, grounded in the 4EM frame [8]. By systematically mapping Jira components and Marketplace tools to RM lifecycle stages, it identifies capability gaps in a transparent and repeatable way, thus addressing the fragmentation observed in previous studies.

In the *enterprise modeling research* , the 4EM approach [8] has been chosen because of its explanatory capabilities [14]. This method provides a structured approach to representing and analyzing enterprises through interconnected models. A model is defined as a simplified representation of reality, focusing exclusively on properties that are relevant to the intended purpose of modelling. The 4EM model is constituted by a series of interconnected models [8]. The enterprise model comprises multiple interconnected sub-models, with each sub-model representing a distinct aspect of the enterprise. In this study, the following models were utilized: the Goal Model, which describes the enterprise's objectives; the Actor and Resource Model, which establishes the connection between individuals, resources, and responsibilities and goals and processes; the Business Rules Model, which defines rules that support goals and business processes; the Business Process Model, which specifies activities and information flows; and the Concepts Model, which clarifies key terms and relationships applied across other models. The proposed model is presented graphically in Section 5 and is also available in [15]. The elements of the concept will be examined in the subsequent section.

3. Expected functionality of requirements management tools

There are multiple objectives of systematic RM [16] such as - supporting the acquisition of requirements, the development of specifications and their grouping, facilitating the development and detailing of requirements, ensuring their flexible adaptation and maintenance, supporting the tracking of requirements, and supporting project development in different parts of the organization.

To gain a deeper understanding of RM needs, it is essential to examine the key stages of RM throughout the project lifecycle. In [17], six stages of RM are proposed. Below, these stages are listed, and each stage is referred to by its abbreviation (RE, RA, RS, RVV, RM, RT), respectively, also reflected in the 4EM Concepts Model discussed in Section 5 (Concepts 2–7):

Elicitation (RE): *Concept 2.* The purpose of requirements elicitation is to obtain all the information necessary for effective requirements specification and solution development.

Analysis (RA): *Concept 3.* Requirements analysis enables organization, classification, linking, and conflict identification. It involves reaching an agreement with stakeholders on the scope and content of requirements, where managing conflicts from differing goals is critical. A consistent set of requirements is necessary to formulate a coordinated solution concept.

Specification (RS): *Concept 4.* Requirements specification refers to the precise description of the object. Effective specifications of requirements require understanding their purpose, target audience, and role throughout the project lifecycle. They can take a textual or model-based form and support knowledge consolidation, change impact analysis, and risk mitigation.

Verification and validation (RVV): *Concept 5.* Requirements verification and validation are essential, as unverified and unvalidated requirements are considered invalid. Validation ensures clarity, correctness, and shared understanding. Reliable decisions require validated, business-aligned requirements.

Requirements (change) management (RM): *Concept 6.* includes building a requirements repository, defining structure, lifecycle understanding, and using attributes for classification. Version control supports change tracking during implementation.

Traceability (RT): *Concept 7.* Traceability links requirements to artifacts, enabling scope control, impact analysis, and alignment with business goals. Without traceability, managing changes becomes difficult, risking instability and failure to meet business needs. It becomes impossible to connect requirements to needs, design elements, tests, or results, leaving projects without control over critical information.

The depicted classification of stages in Table 1 builds on the functional framework established by Carrillo de Gea et al. [9], which maps tool capabilities to distinguish RM phases. In this paper, the functional framework is further aligned with the 4EM enterprise modeling approach. Specifically, the RM lifecycle stages themselves (Concepts 2–7) and their associated detailed functionalities (Concepts 32–46) are formalized in the 4EM Concepts Model, and together they constitute Information Sets 1 – 3

Table 1
RM functionality by lifecycle stage

Stage	Functions	Model
Requirement elicitation	Identification of stakeholders [RE1]	Concept 32
	Documenting business, user, functional, and non-functional requirements [RE2]	Concept 33
	Requirements traceability during elicitation [RE3]	Concept 34
Requirement analysis	Breaking down requirements into details [RA1]	Concept 35
	Feasibility assessment [RA2]	Concept 36
	Setting priorities [RA3]	Concept 37
	Conflict identification [RA4]	Concept 38
	Identification of unclear, incomplete, ambiguous, or conflicting requirements [RA5]	Concept 39
Requirement specification	Documenting the functions of the software or system that it must provide, and the constraints it must adhere to, stating them in a consistent, accessible, and transparent way to achieve this goal [RS1]	Concept 40
	Applying specific methods to create useful and verifiable requirements models [RS2]	Concept 41
Requirement verification and validation	Supporting various testing and evaluation tools used to verify and validate requirements [RVV]	Concept 42
Requirements (change) management	Ability to support change control and requirements maintenance to ensure that requirements accurately reflect the product [RM]	Concept 43
Requirement traceability	Documentation of the requirements life cycle [RT1]	Concept 44
	Ensuring traceability mechanisms between related requirements [RT2]	Concept 45
	Tracking changes to requirements [RT3]	Concept 46

of the Business Process Model.

To ensure alignment with enterprise modeling, each functionality is also represented as a concept in the 4EM Concepts Model. In Table 1, the final column ("Model") shows the corresponding concept. Thus, the RM lifecycle stages themselves (Concepts 2–7) and their detailed functional requirements (Concepts 32–46) are explicitly defined from the enterprise perspective.

Each functionality is further assigned a unique identifier that includes the initials of the RM stage and a sequential number (e.g., RE1, RA2). This systematic labeling supports traceability across requirements and enables direct comparison between tool capabilities and RM process needs.

In the 4EM Actors and Resources Model, Table 1, which provides an overview of the required tool functions for each stage, is represented as Resource 4 in the 4EM Concepts Model, the mapping of both the stages and functions ensures a consistent anchoring of RM needs in the enterprise context.

The functions in Table 1 serve as the basis for tool evaluation discussed in the remainder of this paper. These functions are first applied to Jira by mapping native components to RM lifecycle stages and assessing their coverage, and then to selected Marketplace tools. In addition, each functionality is positioned within the 4EM Concepts Model, ensuring that every elicitation, analysis, specification, validation, management, and traceability function is explicitly anchored in the enterprise perspective. This approach enables the identification of capability gaps in both Jira and Marketplace tools, ensuring they can be evaluated in a consistent and repeatable way.

4. Evaluation of requirements management support in Jira and its Marketplace tools

In this study, each Jira component is described based on Atlassian documentation and hands-on exploration in a sample project environment. The mapping to RM stages was derived by comparing observed functionalities with the RM lifecycle's theoretical aspects. Importantly, the mapping is many-to-many: a single component may support several RM stages, and each stage may rely on multiple components (see Table 2).

To structure the analysis, the 4EM Concepts Model was used, illustrating how Jira components link to RM stages and functionalities, and the Actors and Resources Model, where Resource 1 represents Jira and Resource 3 represents Documentation. Following Process 1 (Identify Jira components) in 4EM Process Model, detailed information on Jira's components and features was collected, forming Information Set 1. These data were then used in Process 2 (Map Jira components to RM stages) to establish connections between components and the RM lifecycle, resulting in Information Set 2. Finally, Process 3 (Evaluate Jira's coverage of each requirement) consolidated these mappings into Information Set 3, capturing the relationships between Jira components and RM stages.

In the 4EM Concepts Model, **Summary** [18] is represented as Concept 28 related to Concept 12, reflecting its role in consolidating project information and providing stakeholders with insights into requirements under development, completed, or pending. This aggregation supports the management stage (Concept 6) by enabling structured oversight of progress and workload.

The **Backlog** [19], central to agile projects, is Concept 22 related to Concepts 8–12, linking to multiple RM stages. It records, prioritizes, and schedules work items for sprints. As backlog items often begin as user stories, it supports elicitation (Concept 2) and analysis (Concept 3), while ongoing refinement relates to management (Concept 6). It also contributes to specification and verification (Concepts 5) through issue descriptions and acceptance criteria.

The **Kanban** [20] board, Concept 25 related to Concepts 9, 12, and 13, supports analysis (Concept 3) by organizing, decomposing, and relating requirements to one another, management (Concept 6) through workflow control and progress visibility, and traceability (Concept 7) by visualizing state transitions across requirement stages.

The **All Work** view, Concept 27 related to Concepts 10, 12 13, supports specification (Concept 4) by providing a structured issue list, management (Concept 6) by organizing lifecycle data, and traceability (Concept 7) by displaying links and dependencies.

The **Board** [21], Concept 26 related to Concepts 9, 12, and 13, supports analysis (Concept 3) by showing dependencies, management (Concept 6) by monitoring progress, and traceability (Concept 7) by exposing the history of changes and relationships across issues.

The **List** [22] view, Concept 31 related to Concept 13, aligns with the traceability (Concept 7), allowing structured tracking of requirements through fields, such as status, priority, and sprint.

The **Calendar** [23], Concept 29 related to Concept 12, provides visibility into deadlines and milestones, supporting the management (Concept 6) by ensuring visibility of due dates and potential scheduling impacts

Reports [24], Concept 23 related to Concepts 9, 11, and 12, link to analysis (Concept 3), verification (Concept 5), and management (Concept 6) by providing metrics such as cumulative flow, cycle time, release frequency, and sprint performance, supporting evaluation of progress and quality over time.

Forms [25], Concept 20 under Concepts 8–11, link to elicitation, analysis, specification, and validation (Concepts 2–5). They enable the capture, documentation, prioritization, and, when configured, validation of requirements in a structured manner.

Goals [26], Concept 21 related to Concepts 8, 9, and 13, relate to elicitation (Concept 2), analysis (Concept 3), and traceability (Concept 7), clarifying objectives and linking requirements to strategic goals.

Sprint [27], Concept 30 related to Concept 12, supports analysis (Concept 3) through scoping and management (Concept 6) by tracking requirement completion across iterations.

Table 2
Mapping of Jira components to RM stages

Requirement management stage	Jira component	Model
Elicitation	Forms	Concept 20
	Goals	Concept 21
	Backlog	Concept 22
Analysis	Backlog	Concept 22
	Reports	Concept 23
	Goals	Concept 21
	Forms	Concept 20
	Estimation	Concept 24
	Kanban	Concept 25
	Board	Concept 26
Specification	Forms	Concept 20
	All work	Concept 27
	Backlog	Concept 22
Verification and validation	Estimation	Concept 24
	Backlog	Concept 22
	Forms	Concept 20
	Reports	Concept 23
(Change) Management	Summary	Concept 28
	Backlog	Concept 22
	Kanban	Concept 25
	All work	Concept 27
	Board	Concept 26
	Calendar	Concept 29
	Reports	Concept 23
	Sprint	Concept 30
Traceability	All work	Concept 27
	Board	Concept 26
	List	Concept 31
	Goals	Concept 21
	Kanban	Concept 25

Finally, the **Estimation** [28], Concept 24, related to Concepts 9 and 11, links to analysis (Concept 3) and management (Concept 6) by quantifying work in story points or time, supporting feasibility assessment, scope definition, effort allocation, and change impact monitoring.

The mapping in Table 2 illustrates that Jira components contribute to multiple RM stages. These many-to-many relationships highlight the versatility of Jira's features when assessed against the structured RM lifecycle. By making these connections explicit, the evaluation provides a foundation for identifying capability gaps in Jira.

4.1. Jira's support for requirements management

To systematically identify capability gaps in Jira (Goal 1), it was necessary to evaluate Jira against RM stages (Process 3 in the 4EM Business Process Model). The analysis examined how each Jira component functionally supports requirements elicitation, analysis, specification, verification and validation, change management, and traceability.

To ensure consistency and comparability, the evaluation was guided by the following rules (represented in the Rules Model): Rule 1, which requires that all tool assessments follow a consistent scoring and mapping approach, and Rule 2, which specifies that RM functionality must be evaluated per lifecycle phase. The analysis includes the use of Information Set 3, which consolidates the mappings of

Jira components to RM stages and serves as the reference dataset for capability evaluation. Based on these rules and information, the specific requirements associated with each activity (as structured in the 4EM Concepts model) were applied, and the Jira environment was tested against them.

Building on these results, Process 4 (Identify unmet requirements) was performed to determine which RM functionalities are fully, partially, or not supported. The outcome corresponds to Information Set 4 in the Business Process Model, which consolidates the list of met and unmet RM requirements in Jira and provides the basis for evaluating Marketplace tools. The results are summarized in Table 3, with the concept numbers in parentheses referring to their corresponding entities in the 4EM Concepts Model.

The degree of fulfilment was rated using three designations: F – fully implements the functionality, P – partially fulfils it or requires configuration, and blank – not supported. For visual clarity, F is highlighted in green, P in yellow, and blanks are unshaded.

Overall, the findings show that Jira offers versatile yet uneven support across RM stages, fully covering stakeholder identification (RE1), feasibility assessment (RA2), prioritization (RA3), and traceability tasks (RT1–RT3), while most other functions are only partially supported. It should be noted that Jira is continuously updated, meaning that some capability gaps may shift over time as new features are introduced or existing ones are redefined. The results demonstrate both the platform's flexibility and its limitations, underscoring the need for a consistent and repeatable method for identifying RM capability gaps in line with the research question of this paper.

Table 3
Evaluation of Jira's support for RM functions

Functionality	Forms (Concept 20)	Goals (Concept 21)	Backlog (Concept 22)	Reports (Concept 23)	Estimation (Concept 24)	Kanban (Concept 25)	Board (Concept 26)	All work (Concept 27)	Summary (Concept 28)	Calendar (Concept 29)	Sprint (Concept 30)	List (Concept 31)
RE1 (Concept 32)	F	F	F									
RE2 (Concept 33)	P		P									
RE3 (Concept 34)	P											
RA1 (Concept 35)	P		P			P						
RA2 (Concept 36)		P		P	F							
RA3 (Concept 37)	F		F									
RA4 (Concept 38)	P											
RA5 (Concept 39)	P		P			P						
RS1 (Concept 40)	P		P					P				
RS2 (Concept 41)												
RVV (Concept 42)			P	P	P							
RM (Concept 43)			P	P		P	P	P	P	P	P	
RT1 (Concept 44)		P					P	P				F
RT2 (Concept 45)						F	P	P				F
RT3 (Concept 46)								F				F

4.2. Jira Marketplace tool support for requirements management

To better understand the support for RM within the Jira ecosystem (Jira together with its native components and extensions available through the Jira Marketplace), a scan was conducted using the keyword "requirements management." The search revealed diverse extensions with varying RM coverage: six

dedicated RM tools, one combining RM and Confluence integration, three including test management, five focused on testing with RM features, and five broader project or integration tools. This confirms that while Jira lacks full RM functionality, its Marketplace ecosystem offers partial solutions addressing these gaps in different ways.

Following the 4EM Model and, more specifically, the Business Process Model, the analysis proceeded with Processes 5–6: selecting RM Marketplace tools, evaluating their coverage of each requirement, and identifying unmet requirements. The evaluation constitutes Information Set 6, which contains the evaluation results of the Marketplace tools.

For the selection of RM Marketplace tools (Process 5 in the 4EM Business Process Model), four tools were evaluated using the same rating scheme applied to Jira components and the functions listed in Table 1. The selected tools: **rmCloud** [29], **RMsis** [30], **easeRequirements** [31], and **TraceCloud** [32], were identified through the Jira Marketplace search for "requirements management." This scoped subset is not intended to be exhaustive and can be extended.

In Table 4, the concept numbers in parentheses refer to entities defined in the 4EM Concepts Model, linking each evaluated functionality to its corresponding concept.

The consolidated results of this assessment are presented in Table 4. Jira's scores are reported as the highest score achieved for each requirement across its native components rather than as a cumulative total. This approach avoids double-counting overlapping functionality and ensures a fair comparison with Marketplace tools, highlighting capability gaps and areas where individual tools extend RM coverage.

Table 4
Evaluation of RM tools in the Jira Marketplace

Functionality	Jira	rmCloud	RMsis	EaseRequirements	TraceCloud
RE1 (Concept 32)	F				
RE2 (Concept 33)	P	F	F	F	F
RE3 (Concept 34)	P	F	P	F	F
RA1 (Concept 35)	P	F	F	P	F
RA2 (Concept 36)	F	P	F		F
RA3 (Concept 37)	F	P	F	P	
RA4 (Concept 38)	P		F		
RA5 (Concept 39)	P		F		
RS1 (Concept 40)	P	F	F	F	F
RS2 (Concept 41)					
RVV (Concept 42)	P	F	F		P
RM (Concept 43)	P	F	F	F	
RT1 (Concept 44)	F	P	F	F	F
RT2 (Concept 45)	F	P	F	F	F
RT3 (Concept 46)	F	P	F	F	F

Table 4 shows that Marketplace tools expand Jira's RM coverage, but none provide full lifecycle support individually. Most strengthen structured specification (RS1), documentation (RE2–RE3), and traceability (RT1–RT3), with some support for analysis and verification. Nevertheless, notable gaps remain, highlighting that coverage is uneven and requires consistent, repeatable evaluation to track improvements and remaining deficiencies.

5. Enterprise model for systematic requirements management gap analysis

As an answer to the research question *"How can requirements management capability gaps be recognized in Jira and in Marketplace applications designed for RM, and how can they be evaluated in a consistent, repeatable manner?"*, an enterprise model was developed that can guide the analysis.

Figure 1 presents this model, which captures, at a meta level, the steps required to systematically identify capability gaps in its Business Process Model. The Business Process Model illustrates the eval-

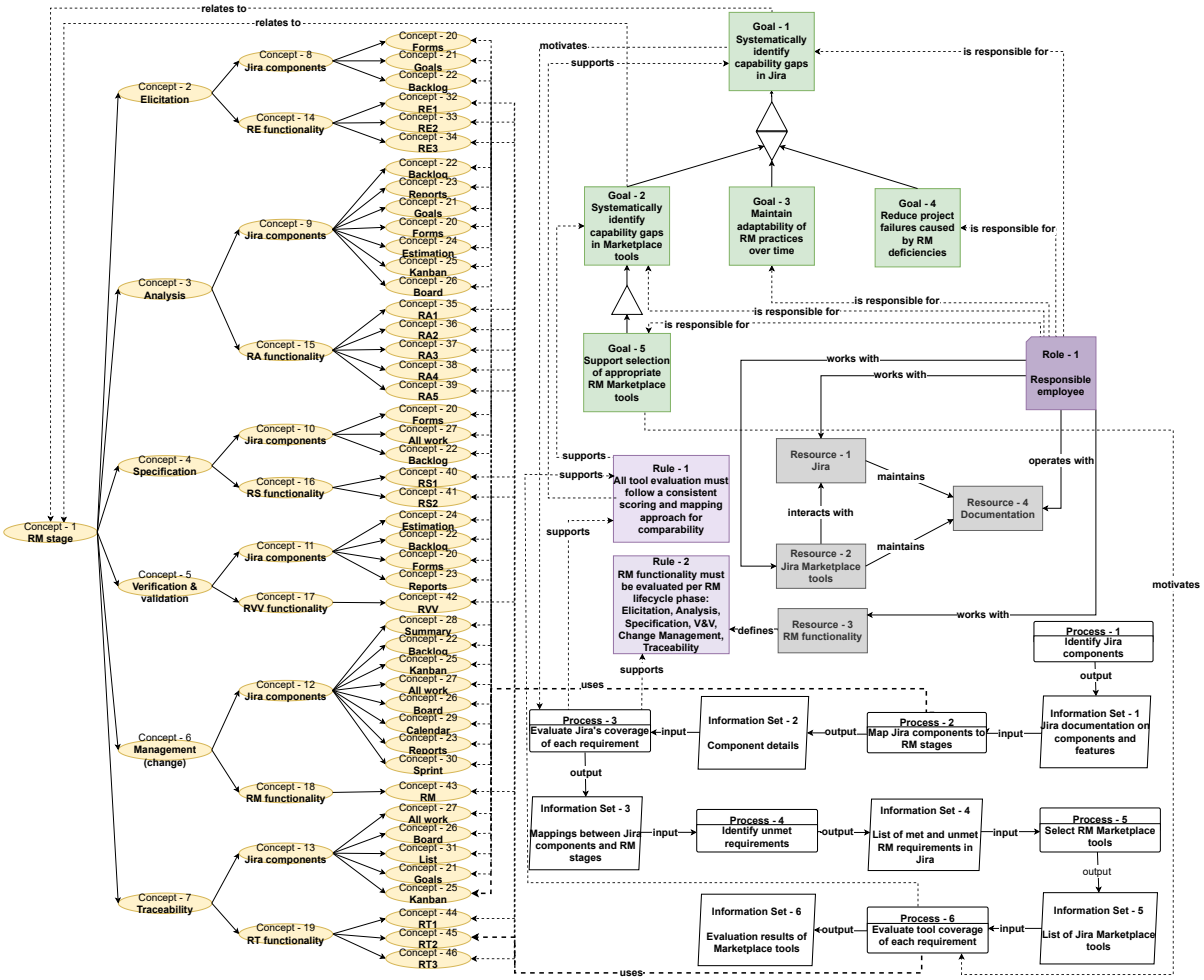

Figure 1: 4EM enterprise model for evaluating requirements management capability gaps in Jira and its Marketplace tools (available also at [15])

uation workflow, beginning with the identification of Jira's components, mapping them to RM stages, and assessing their coverage of functional requirements. Subsequent processes extend the analysis to Marketplace tools. Each process in the model consumes and produces specific information sets, which capture structured data, including component lists, lifecycle mappings, and evaluation results.

The Goals Model defines the intended outcomes (goals to be fulfilled): the systematic detection of RM capability gaps, adaptability of RM practices in evolving tool ecosystems, and mitigation of project risks caused by deficiencies in RM support.

The Actors and Resources Model identifies the employee responsible for the assessment and the key artifacts involved: the Jira platform, Marketplace tools, vendor documentation, and the RM functionality list (Concepts 32–46). The Rules Model ensures consistency: Rule 1 requires assessing all tools against the same RM functionality set, and Rule 2 requires reporting evaluations per lifecycle stage

(Concepts 2–7).

The Concepts Model defines the RM lifecycle stages and their relationships with Jira components and required RM functionalities used in the evaluation (Sections 3 and 4). Links between concepts illustrate these dependencies. Concepts 2–7 represent the RM stages, each associated with an RM functionality (Concepts 14–19), which is further specified through Concepts 32–46. Supporting Jira components are grouped under each stage through Concepts 8–13, with detailed elements captured in Concepts 20–31. Although this extends standard 4EM notation, it clarifies how these relationships ensure consistent capability evaluation across Jira and its Marketplace tools.

Model provides a structured and repeatable basis for assessing RM support and underpins the evaluations in Section 4. It also allows the assessment to be updated as Jira or Marketplace tools evolve, with modifications in RM features consistently reflected in the evaluation tables.

6. Conclusion

This study addressed the research question: *"How can requirements management capability gaps be recognized in Jira and in Marketplace applications designed for RM, and how can they be evaluated in a consistent, repeatable manner?"* To answer this question, a gaps evaluation model, grounded in the 4EM frame, was developed and aplied to Jira and selected Marketplace tools.

The evaluation showed that Jira provides versatile but incomplete RM support. Full support is limited to functions such as stakeholder identification, feasibility assessment, prioritization, and traceability, while most other RM functions are only partially supported. Marketplace tools extend Jira's functionality, particularly in specification, documentation, and traceability. However, none of the reviewed tools provided comprehensive lifecycle coverage, and persistent gaps remain. Consequently, enterprises using Jira for RM should not treat it or any single Marketplace tool as a complete solution. Tool selection should be based on systematic evaluation against project-specific needs, with awareness of strengths and remaining limitations.

In this regard, the enterprise model proved valuable for recognizing capability gaps in Jira and Marketplace tools and evaluating them. It helped to structure the analysis by integrating goals, processes, actors, resources, concepts, and rules. It enabled systematic mapping of tool functionalities to RM lifecycle stages and transparent identification of capability gaps. The 4EM-based model supports a structured, repeatable approach for recognizing these gaps, supporting consistent tool evaluation as Jira and its Marketplace evolve.

Future directions for the model's application include:

- Applying the model to a broader set of Marketplace tools, including those that combine RM with test management, project management, or Confluence integration.
- Applying the model beyond Jira to test its generalizability across platforms.
- Reapply the model over time to track how Jira and Marketplace tools evolve, and how updates or new releases affect capability coverage.
- Applying the model to competing RM platforms outside the Jira ecosystem to provide benchmarking and support evidence-based tool selection.

While this research is limited to the Jira ecosystem, it advances systematic support for selecting available software tools. No tools within Jira ecosystem were found to support graphical modelling beneficial for requirements engineering. Therefore, extending Jira with compatible modelling tools is an important direction for future research, given the platform's extensive use.

Declaration on Generative AI

During the preparation of this work, one of the authors, Agnese Rozenberga, used GPT-5 and DeepL for grammar checking, translation, and rephrasing assistance. All AI-assisted outputs were thoroughly reviewed and edited by the authors, who take full responsibility for the final content of the manuscript.

References

[1] Project Management Institute (PMI), A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 5 ed., Project Management Institute, Newtown Square, PA, 2013.

[2] Atlassian, Project management features for all teams, Atlassian, Sydney, 2025. URL: https://www.atlassian.com/software/jira/features, accessed 2025-10-31.

[3] Atlassian, Who uses jira?, Atlassian, Sydney, 2025. URL: https://www.atlassian.com/software/jira/guides/getting-started/who-uses-jira, accessed 2025-10-31.

[4] L. Filion, N. Daviot, J.-P. Le Bel, M. Gagnon, Using atlassian tools for efficient requirements management: An industrial case study, in: Proceedings of the IEEE International Systems Conference (SysCon 2017), IEEE, Montreal, Canada, 2017, pp. 1–6. doi:10.1109/SYSCON.2017.7934769.

[5] Atlassian, Using jira for requirements management, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/jira/kb/using-jira-for-requirements-management/, accessed 2025-10-31.

[6] Atlassian, Confluence basics, Atlassian, Sydney, 2025. URL: https://www.atlassian.com/software/confluence/resources/guides/get-started/, accessed 2025-10-31.

[7] Atlassian, An introduction to jira integrations: What is atlassian marketplace?, Atlassian, Sydney, 2025. URL: https://www.atlassian.com/software/jira/guides/integrations/overview, accessed 2025-10-31.

[8] K. Sandkuhl, J. Stirna, A. Persson, M. Wißotzki, Enterprise Modeling: Tackling Business Challenges with the 4EM Method, Springer, Berlin, Heidelberg, 2014. doi:10.1007/978-3-662-43725-4.

[9] J. Carrillo de Gea, R. Nicolás, J. Fernández-Alemán, A. Toval, C. Ebert, Requirements engineering tools: Capabilities, survey and assessment, Information and Software Technology 54 (2012) 1142–1157. doi:10.1016/j.infsof.2012.04.005.

[10] R. Özkaya, R. L. Nord, P. Kruchten, An analysis of the features of requirements engineering tools, Journal of Systems and Software 196 (2023) 111569. doi:10.3390/systems11120576.

[11] S. Wagner, D. M. Fernández, M. Felderer, M. Kalinowski, Status quo in requirements engineering: A theory and a global family of surveys, ACM Transactions on Software Engineering and Methodology 27 (2018) 1–48. doi:10.1145/3306607.

[12] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, F. Gomes de Oliveira Neto, Requirements engineering challenges and practices in large-scale agile system development, Journal of Systems and Software 172 (2021) 110851. doi:10.1016/j.jss.2020.110851.

[13] A. Rozenberga, Structuring complexity: A functional evaluation of jira tools for requirements management, in: BIR-WS 2025: Proceedings of the 24th International Conference on Perspectives in Business Informatics Research – Workshops and Doctoral Consortium, CEUR Workshop Proceedings, 2025.

[14] M. Kirikova, Explanatory capability of enterprise models, Data & Knowledge Engineering 33 (2000) 119–136. doi:10.1016/S0169-023X(99)00048-8.

[15] A. Rozenberga, M. Kirikova, 4em enterprise model for evaluating requirements management capability gaps in jira and marketplace applications, Developed as part of the present study, 2025. URL: https://drive.google.com/file/d/1uJjI5l6VZga5OK5vYqxhD_2ijurWt7YS/view, accessed 2025-10-31.

[16] M. Hoffmann, N. Kuhn, M. Weber, M. Bittner, Requirements for requirements management tools, in: Proceedings of the 12th IEEE International Requirements Engineering Conference (RE 2004), IEEE, Kyoto, Japan, 2004, pp. 301–308. doi:10.1109/ICRE.2004.1335687.

[17] K. Zmitrowicz, Business analysis done right: Lessons learned and pitfalls avoided, Springer Nature, Cham, 2024. doi:10.1007/978-3-031-62194-9.

[18] Atlassian, What is the summary view?, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/jira-software-cloud/docs/what-is-the-summary-view/, accessed 2025-10-31.

[19] Atlassian, Enable the backlog, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/jira-software-cloud/docs/enable-the-backlog/, accessed 2025-10-31.

[20] Atlassian, Kanban boards in jira, Atlassian, Sydney, 2025. URL: https://www.atlassian.com/

software/jira/features/kanban-boards, accessed 2025-10-31.
[21] Atlassian, What is the board?, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/jira-software-cloud/docs/get-started-with-team-managed-projects/#What-is-the-board, accessed 2025-10-31.
[22] Atlassian, What is the list view?, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/jira-software-cloud/docs/what-is-the-list-view/, accessed 2025-10-31.
[23] Atlassian, What is the calendar?, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/jira-software-cloud/docs/what-is-the-calendar/, accessed 2025-10-31.
[24] Atlassian, Enable reports, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/jira-software-cloud/docs/enable-reports/, accessed 2025-10-31.
[25] Atlassian, What are forms?, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/jira-service-management-cloud/docs/what-are-forms/, accessed 2025-10-31.
[26] Atlassian, What is a goal?, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/platform-experiences/docs/what-is-a-goal/, accessed 2025-10-31.
[27] Atlassian, What is a sprint?, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/jira-software-cloud/docs/what-is-a-sprint/, accessed 2025-10-31.
[28] Atlassian, What is estimation in jira?, Atlassian, Sydney, 2025. URL: https://support.atlassian.com/jira-software-cloud/docs/what-is-estimation-in-jira/, accessed 2025-10-31.
[29] Atlassian, Rmcloud – requirements management, Atlassian Marketplace, 2025. URL: https://marketplace.atlassian.com/apps/1217241/rmcloud-requirements-management, accessed 2025-10-31.
[30] Atlassian, Rmsis – requirements management for jira, Atlassian Marketplace, 2025. URL: https://marketplace.atlassian.com/apps/30899/rmsis-requirements-management-for-jira, accessed 2025-10-31.
[31] Atlassian, Easerequirements – requirements management for jira (r4j), Atlassian Marketplace, 2025. URL: https://marketplace.atlassian.com/apps/1213064/easerequirements-requirements-management-for-jira-r4j, accessed 2025-10-31.
[32] Atlassian, Tracecloud – projects & requirements, Atlassian Marketplace, 2025. URL: https://marketplace.atlassian.com/apps/1231062/tracecloud-projects-requirements, accessed 2025-10-31.