REQUIREMENTS ENGINEERING AND TECHNICAL SPECIFICATION FOR DIGITAL ACADEMIC MANAGEMENT SYSTEMS IN HIGHER EDUCATION

Siradjev Saloxiddin Negmatovich[1], Muminov Bahodir Boltayevich[2]

[1]Associate Professor of the Department of Information Systems and Technologies at Karshi State Technical University
Email: goldsagf@gmail.com orcid: 0009-0000-5912-7194
[2]Tashkent State University of Economics, UZ
doctor of technical sciences, professor.
E-mail: mbbahodir@tsue.uz orcid 0000-0003-4268-9750

K E Y W O R D S

requirements engineering, higher education, academic management system, technical specification, digital transformation, BPMN, requirements management, information system

A B S T R A C T

The digital transformation of higher education requires integrated information systems capable of supporting academic planning, curriculum management, scheduling, electronic document circulation, reporting, and communication among institutional stakeholders A critical prerequisite for the successful development of such systems is the systematic identification, analysis, documentation, verification, and management of software requirements. This study examines a requirements engineering framework for an academic management information system in higher education and proposes a structured approach to the development of a Technical Specification (TS). The proposed framework distinguishes functional and non-functional requirements and incorporates stakeholder analysis, business process modelling, requirements prioritization, prototyping, verification, documentation, and lifecycle management. The study also considers the applicability of ISO/IEC/IEEE 29148, ISO/IEC 25010, IEEE 830, and GOST 34.602-89 as methodological references for technical specification development.

The proposed approach establishes a common basis for communication between university administrators, academic departments, teachers, students, analysts, and software developers. It can contribute to reducing ambiguity, improving requirements traceability, supporting system scalability, and increasing the alignment between institutional processes and digital solutions.

1. Introduction

Digital transformation has become an important component of modern higher education management. Universities increasingly rely on information systems to manage educational plans, academic programs, teaching schedules, workload distribution, documentation, reporting, and communication between institutional stakeholders. [1-3]. The development of an integrated academic management system is, however, not limited to programming activities. The quality of the final system largely depends on the accuracy with which organizational processes and user requirements are identified and translated into technical specifications [4]. The source material emphasizes that the requirements and Technical Specification stage forms the foundation for subsequent system design, development, and testing [5]. In higher education, several user groups interact with academic information systems, including administrators, deans' offices, academic departments, teachers, and students [6]. Each group has different responsibilities and information needs. Therefore, a system designed without systematic requirements engineering may result in duplicated data entry, inconsistent information, insufficient access control, inefficient workflows, and difficulties in future system expansion.

At the same time, international standards can serve as a methodological basis for developing a more systematic Technical Specification. The initial material specifies ISO/IEC/IEEE 29148:2018 for requirements engineering, ISO/IEC 25010 for software quality specifications, IEEE 830 for software requirements specifications, and GOST 34.602-89 for technical specifications of automated systems. The

(cc) BY

combined use of these approaches can improve the consistency and verifiability of project documentation. Requests can also be managed using dedicated platforms such as Jira, Redmine, Trello, and Confluence, while GitHub or GitLab can support version management and tracking between requests and software artifacts [7-10].

The purpose of this study is to develop a structured methodological framework for identifying and documenting requirements for an integrated academic management information system in higher education.

2. Materials and methods

The study is based on the analysis of the functional structure and technical requirements of the academic management information system. The methodological framework includes five main stages: analysis of stakeholders and processes, identification and classification of requirements, modeling and verification of requirements, development and management of the terms of reference, and validation and acceptance criteria.

2.1. Stakeholder analysis

The main stakeholder groups considered in the proposed system include system administrators, university management, deanships, academic departments, teachers, and students.

The conceptual structure of the proposed integrated academic management system and the relationships among its main stakeholders and functional modules are illustrated in Figure 1.

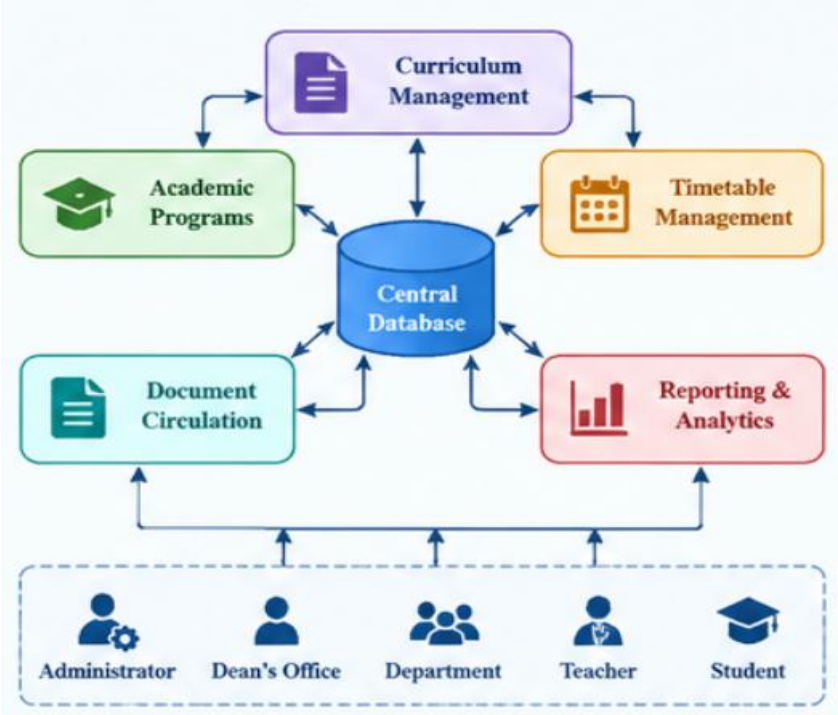


Figure 1. Conceptual Model of the Integrated Academic Management System

Interviews, questionnaires, observations, document analysis, workshops, and brainstorming can be used to identify stakeholder requirements. The source material specifically identifies interviews, observations, document analysis, workshops, brainstorming, and prototyping as appropriate requirement collection techniques.

2.2. Classification of requirements

Requirements are divided into functional and non-functional.

Functional requirements describe what the system should do. For academic management system includes creating and managing curricula, developing and approving training course programs, generating schedules automatically or semi-automatically, managing loads, electronic document flow, searching for and filtering

(cc) BY

information, importing and exporting data, and automatically generating analytical reports.

These functions correspond to the functional requirements described in the source material.

Non-functional requirements describe how the system should function. These include reliability, security, performance, scalability, cross-platform compatibility, and accessibility.

2.3. Business process modelling

Business process modelling is used to represent the processes that will subsequently be implemented within the information system. BPMN diagrams can be used to describe curriculum development, course-program approval, timetable generation, and reporting processes.

The use of process models allows system analysts and stakeholders to identify information flows, user actions, decision points, and dependencies before software implementation begins. An example of a BPMN-based workflow for curriculum development and approval is presented in Figure 2.

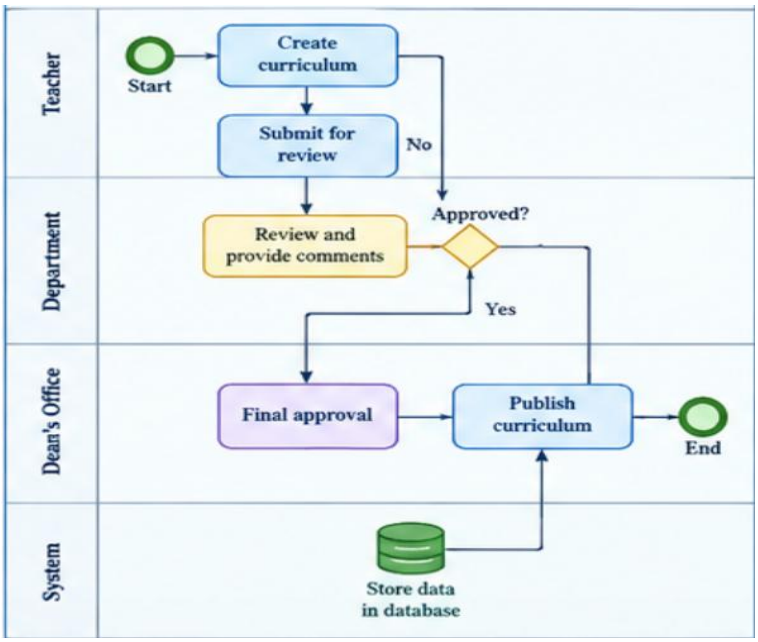


Figure 2. BPMN Diagram of the Curriculum Development and Approval Process

BPMN diagrams can be used to describe curriculum development, course-program approval, timetable generation, and reporting processes.

2.4. Requirements verification

After requirements have been collected, they should be analyzed according to completeness, consistency, feasibility, measurability, and absence of contradictions.

A requirement should therefore be sufficiently precise to permit later verification through testing. Requirements that cannot be objectively assessed should be reformulated in measurable terms. The overall requirements engineering framework developed for the proposed system is summarized in Figure 3.

(cc) BY

ISSN: 3030-3346

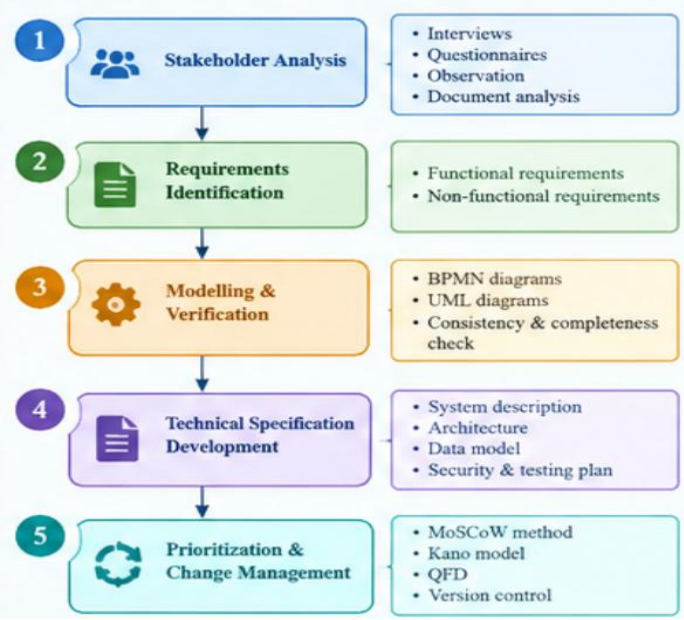


Figure 3. Requirements Engineering Framework

3. Technical Specification Framework

The Technical Specification serves as the principal regulatory document for system development, testing, and implementation. It combines functional, technical, organizational, and operational requirements into a unified project document.

A recommended structure is presented below.

Section	Main content
Project objectives	General purpose and expected outcomes
System description	Scope and boundaries
Functional requirements	Required system functions
Non-functional requirements	Performance, security, reliability, usability
Architecture	Client-server or microservices model
Database	Data structures and relationships
User roles	Administrator, dean's office, department, teacher, student
Security	Authentication, authorization, encryption, auditing
Testing	Functional, integration, load, security, acceptance testing
Deployment	Local, cloud, or hybrid infrastructure
Maintenance	Updating and modernization procedures

This structure is consistent with the technical specification components identified in the source material.

4. Priority of requirements

Not all requirements are equally important. Therefore, requirements must be prioritized before implementation. The MoSCoW method divides requirements into four categories.

Must have - mandatory requirements;

Must exist - essential, but not absolutely mandatory requirements;

Possibly - additional necessary functions;

No - features excluded from the current development phase.

In the main material, the Kano model and the implementation of the Quality Function (QFD) are also presented as approaches that support prioritizing requirements and transforming user needs into technical solutions.

(cc) BY

5. Results and Discussion

The proposed requirements engineering framework provides a structured transition from institutional needs to technical implementation.

The first important result is the separation of user needs from technical solutions. For example, a teacher's need for rapid access to an updated timetable should first be formulated as a user requirement and only subsequently translated into a specific interface or software function.

The second result is improved traceability. Each requirement can be assigned a unique identification code. This makes it possible to trace a requirement from its initial definition through design, implementation, and testing. The source material recommends assigning an identification code to each requirement to support change tracking and subsequent testing.

The third result concerns change management. A Technical Specification should not be regarded as a static document. Changes during development and operation should be documented using formal change requests and version control mechanisms.

The fourth result is the integration of iterative development practices. Model-Driven Development can employ UML, BPMN, and ER diagrams before implementation, while Agile and prototyping approaches allow requirements to be refined through continuous stakeholder feedback.

These approaches are particularly relevant to higher education because academic regulations, curricula, organizational structures, and reporting requirements may change over time.

6. Conclusion

A systematic requirements engineering process is a fundamental prerequisite for the successful development of integrated academic management information systems in higher education. The analysis demonstrates that requirements should be collected from all major stakeholder groups, classified into functional and non-functional categories, modelled using appropriate techniques, verified for completeness and consistency, prioritized, and documented within a structured Technical Specification.

The proposed framework combines stakeholder analysis, BPMN-based process modelling, requirements prioritization, prototyping, verification, version management, and acceptance criteria. Such an approach creates a common communication framework between university management, academic staff, analysts, and software developers.

The study is conceptual and methodological in nature. Therefore, quantitative claims concerning productivity improvements, cost reduction, or user satisfaction require empirical validation through implementation and controlled evaluation in a real university environment. Future research should focus on prototype development, usability evaluation, performance testing, and empirical comparison of traditional and requirements-driven approaches to academic information system development.

References

1. Jose Ermeson Silva Carmo, Daniel Pacheco Lacerda, Cristina Orsolin Klingenberg, Fabio Antonio Sartori Piran,Digital transformation in the management of higher education institutions,Sustainable Futures, Volume 9, 2025, https://doi.org/10.1016/j.sftr.2025.100692.
2. Silva, Ermeson & Lacerda, Daniel & Klingenberg, Cristina & Piran, Fabio. (2025). Digital Transformation in the Management of Higher Education Institutions Digital Transformation in the Management of Higher Education Institutions. Sustainable Futures. doi.org/10.1016/j.sftr.2025.100692.
3. Сираджев С.Н., Муминов Б.Б., Хайдаров Т.З. Инновационные подходы и технологические возможности в управлении высшим образованием // Universum: технические науки : электрон. научн. журн. 2025. 9(138). URL: https://7universum.com/ru/tech/archive/item/20804
4. Daun M, Grubb AM, Stenkova V, Tenbergen B. A systematic literature review of requirements engineering education.

(cc) BY

Requir Eng. 2023;28(2):145-175. doi: 10.1007/s00766-022-00381-9

5. Waruwu, Marinu & Halida, & Takdir, Muh & Dwikurnaningsih, Yari & Nuryani, Lilis & Satyawati, Sophia & Irnawati, Wati. (2025). Development of Digital-Based Academic Services System in Higher Education. International Research Journal of Multidisciplinary Scope. 06. 92-103. doi.org/10.47857/irjms.2024.v06i04.04750.
6. Johansson, Björn & Lahtinen, Markus. (2022). Getting the balance right between functional and non-functional requirements: the case of requirement specification in IT procurement. International Journal of Information Systems and Project Management. 1. 5-16. doi.org/10.12821/ijispm010101.
7. ISO/IEC/IEEE. (2018). *ISO/IEC/IEEE 29148:2018 Systems and software engineering — Life cycle processes — Requirements engineering*.
8. ISO/IEC. (2011). *ISO/IEC 25010: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*.
9. IEEE. (1998). *IEEE 830-1998: Recommended Practice for Software Requirements Specifications*.
10. GOST 34.602-89. *Technical Specification for Creation of an Automated System*.

(cc) BY