

---

# A Lightweight Requirements Engineering Process for Web-Based Competition Management Systems: The GOHIT v2 Case Study

Fadila Zeti Dewinta, Koko Wahyu Prasetyo\*

*Faculty of Science and Technology, Universitas Bhinneka Nusantara, Malang, Indonesia*

---

## **Keywords**

*requirements engineering; requirements elicitation; requirements analysis; requirements specification; software requirements;*

## **\*Correspondence Email:**

*koko@ubhinus.ac.id*

## **Abstract**

The GOHIT platform is a web-based system designed to support the management of student competitions within higher-education institutions. The first version of the platform lacked structured requirements documentation, resulting in unclear user roles and inconsistent workflows. This study aimed to establish a lightweight yet disciplined requirements-engineering (RE) process for the redevelopment of GOHIT v2. Using a qualitative case-study approach, data were collected through semi-structured interviews and document analysis involving three key stakeholders: the founder, the developer, and the program advisor. Thematic analysis was applied to transform stakeholder inputs into structured, verifiable requirements based on IEEE 830 documentation principles. The process yielded seventeen functional requirements, each traceable to stakeholder sources and internally evaluated using IEEE 830 quality attributes. The resulting Software Requirements Specification demonstrated high levels of completeness, consistency, and traceability, confirming that a structured yet adaptable RE approach can be effective for small or academic development teams. This study contributes a replicable model for implementing lightweight requirements engineering in resource-limited settings and establishes a baseline for future validation and quality assurance using a test-based mechanism.

---

## **1. Introduction**

GOHIT is a web-based platform designed to facilitate the management of student competitions within university environments. It allows organizers to create, manage, and validate competitions while enabling participants to register, form teams, and access results and certificates online. Despite its functional intent, the initial version of GOHIT (v1) was developed without formal requirements documentation, leading to unclear user roles, incomplete validation procedures, and inconsistencies between design and stakeholder expectations. These limitations underscored the need for a structured requirements-engineering (RE) process in the platform's redevelopment.

Requirements engineering provides a systematic approach to defining what a system should accomplish and how it should behave before implementation begins. When applied properly, RE ensures alignment between

stakeholder needs and technical design, preventing costly rework and scope creep. However, small or student-led development teams often lack formal RE training and resources to implement comprehensive industrial frameworks. Consequently, there is a pressing need for lightweight yet structured RE processes that retain traceability and quality assurance but remain feasible for limited-capacity teams, especially in academic or startup settings.

This study addresses that need by applying a disciplined yet adaptable RE process to the development of GOHIT v2. The focus lies on eliciting, analyzing, and specifying functional requirements that reflect stakeholder intent while maintaining internal quality through IEEE 830 documentation standards. The study contributes empirical evidence on how a structured RE approach can be applied effectively in small-team contexts, producing requirements that are complete, consistent, and verifiable. The research therefore seeks to understand how stakeholder needs can be systematically translated into formal specifications and to what extent a lightweight RE process can ensure internal quality attributes within the resulting Software Requirements Specification.

### **1.1 Literature Review**

Requirements engineering (RE) is a disciplined process that defines, analyzes, and documents software needs before implementation. It ensures that stakeholder expectations are explicitly stated and traceable to system features. Sommerville (2011) describes RE as an iterative process encompassing elicitation, analysis, specification, and validation. Each stage contributes to preventing rework and ensuring alignment between user intent and system functionality.

The IEEE 830 standard remains one of the most established frameworks for structuring Software Requirements Specifications (SRS). It defines key quality attributes (completeness, correctness, consistency, verifiability, and traceability) that ensure software requirements are measurable and unambiguous throughout development. Stephen and Mit (2020) empirically demonstrated that SRS documents compliant with IEEE 830 improve clarity and reduce semantic ambiguity through measurable quality indicators. Similarly, Elliott and Allen (2013) emphasized that applying formal SRS templates based on IEEE 830 enhances consistency and completeness across project artefacts. Together, these works establish IEEE 830 as a practical reference for maintaining documentation quality in small and large projects alike.

Beyond documentation standards, researchers have explored how lightweight approaches can deliver comparable rigor with reduced procedural overhead. Early studies by Gervasi and Nuseibeh (2000) and Easterbrook et al. (1998) demonstrated that lightweight validation and formal-methods techniques can identify inconsistencies in natural-language requirements without requiring full formalization. These approaches confirmed that meaningful verification can occur early in the requirements phase. Later, Farid (2012) introduced the NORMAP methodology, integrating non-functional requirement modeling into Agile workflows through a simplified yet systematic framework. More recently, Ågren et al. (2019) observed in the automotive domain that hybrid approaches combining lightweight upfront requirements work with post-development validation improved delivery speed while maintaining specification accuracy.

In academic and educational contexts, lightweight and Agile-inspired RE practices are particularly advantageous. Tam et al. (2022) applied Agile RE principles to analyze information-system requirements at STIKI Malang, demonstrating that iterative elicitation and prioritization improved requirement completeness and stakeholder involvement. Their findings resonate with the GOHIT v2 case, where small academic teams operate under similar constraints of time, manpower, and formal RE expertise. Such evidence supports the argument that Agile-adapted, minimal-overhead RE processes can enhance quality outcomes even in low-resource environments.

Requirement traceability also plays a central role in ensuring documentation integrity. Chikh and Aldayel (2014) proposed an SRS model that integrates IEEE 830 structure with hyperlinked traceability mechanisms, improving maintenance and comprehension for both developers and reviewers. These principles echo earlier arguments that requirement traceability strengthens system evolvability and auditability.

Collectively, these studies affirm that combining lightweight RE practices with IEEE 830-based quality assurance can produce specifications that are both practical and reliable. They provide the theoretical foundation for the present research on GOHIT v2, which seeks to demonstrate that disciplined yet adaptable RE methods are viable for small-scale software projects, especially within academic or startup ecosystems where resource efficiency is paramount.

## 2. Research Methods

This study employed a qualitative case-study approach to examine the early stages of requirements engineering for the GOHIT v2 platform, which is a web-based system designed to manage student competitions. The goal was to establish a lightweight yet traceable process for eliciting, analyzing, and specifying functional requirements. The research workflow followed a sequential model consisting of stakeholder input, elicitation, analysis, specification, and internal quality review, as illustrated in Figure 1.

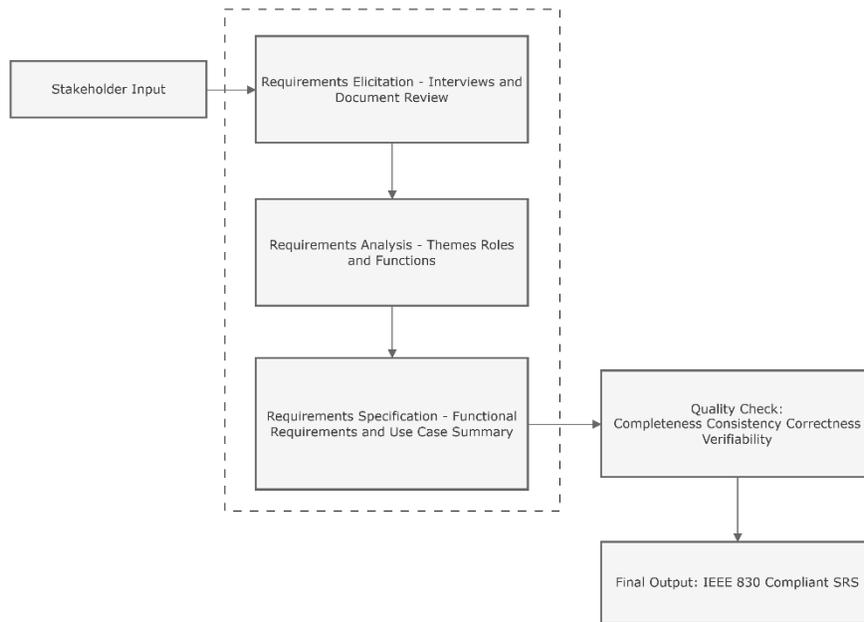


Fig. 1 Research Workflow

Three primary stakeholders participated in the study: the founders (product owner), the developer (engineer), and the program advisor (P2MW facilitator). They were selected through purposeful sampling based on their direct involvement in defining, developing, and validating the system. Each stakeholder contributed distinct perspectives on business goals, technical feasibility, and process conformity. Table 1 summarizes their respective roles, responsibilities, and the forms of data collected during the study.

Data were gathered through semi-structured interviews and document analysis. The interviews, lasting between 45 and 60 minutes, explored user roles, competition workflows, and system expectations. All sessions were recorded, transcribed verbatim, and reviewed by participants for accuracy. Complementary document analysis included the original GOHIT v1 proposal, mock-ups, and prior development notes to contextualize the interview findings and identify recurring process elements.

Thematic analysis was used to interpret the collected data. Following Braun and Clarke's (2006) procedure, transcripts were first coded line-by-line to identify meaningful statements, then grouped into themes representing user roles, functional flows, and system constraints. The themes were iteratively refined until data saturation was reached, forming a stable analytical basis for translating stakeholder needs into formal requirements during the specification stage.

Table 1. Stakeholders and Data Sources

Role	Responsibility / Contribution	Data Collection	RE Stage
Co-Founder / Product Owner	Defines business goals, user expectations, and competition process logic.	Semi-structured interview (≈60 min)	Elicitation / Analysis
Developer / Engineer	Provides technical feasibility input and feedback from GOHIT v1 prototype.	Semi-structured interview (≈45 min) + document review	Analysis / Specification
Program Advisor (P2MW)	Ensures alignment with entrepreneurship-program standards and acts as external validator.	Interview + review meeting notes	Elicitation / Analysis

To ensure methodological rigor, three complementary validation strategies were applied. Triangulation combined interviews and document analysis to confirm consistency across data sources. Member checking allowed stakeholders to verify the accuracy of interpreted findings. Finally, internal quality assurance applied IEEE 830 attributes (completeness, consistency, correctness, verifiability, traceability, and clarity) to assess the robustness of the documented process. These procedures collectively strengthened the credibility and reproducibility of the research design, ensuring that the subsequent specification activities were grounded in validated and well-structured empirical evidence.

### 3. Result and Discussion

The case study focused on the GOHIT v2 development initiative, a follow-up to the initial GOHIT v1 prototype which is a platform designed to manage university-level competitions. The first version lacked formal documentation of requirements, role definitions, and systematic validation. These weaknesses motivated a structured requirements-engineering (RE) process for version 2. The study produced a clear definition of system actors, functional requirements, and an evaluation of requirement quality, as described in the following subsections.

#### 3.1 Elicitation Results

Interviews and document reviews revealed several operational shortcomings in GOHIT v1: incomplete competition data, overlapping user roles, and inconsistent approval workflows. Through successive elicitation sessions, stakeholder statements were consolidated into nine main themes representing business and system expectations. These themes covered user management and authentication, competition creation and validation, team and bracket management, publication of results and certificates, and event notifications.

Each stakeholder emphasized different priorities. The founder focused on participant flow and monetization via premium organizer accounts. The developer stressed automation and feasibility, particularly for bracket generation and validation. Finally, the program advisor emphasized transparency and verifiable approval mechanisms. Together, these inputs defined the conceptual baseline for the subsequent requirement analysis.

#### 3.2 Requirement Analysis

Thematic coding transformed the elicited statements into structured requirements. Three principal actors were identified: Participant, Organizer, and Admin. Each actors responsible for specific process areas. Their interactions are represented in Figure 2, which summarizes the system’s functional scope through a UML use-case diagram.

Most system functions centered on the organizer role, reflecting the platform’s orientation toward event administration. Participant-side functions addressed usability whereas the admin handled verification and monitoring. Cross-cutting functions such as Notifications and Payment Processing ensured integration between all three actors. Logical dependencies between functions were also identified: for example, Publish Results includes Update Match Results, while Generate Certificate depends on result publication. These relationships reinforced structural consistency throughout the specification process.



Fig. 2 Use Case Diagram

### 3.3 Functional Requirement Specification

The analysis yielded seventeen functional requirements (FR\_001–FR\_017) documented according to IEEE 830 structure. Each requirement was described by its identifier, title, primary actor, and priority level. The formal specification ensured every FR could be traced back to at least one stakeholder statement, establishing empirical grounding for each feature. Table 2 presents the complete list of requirements developed from this process.

This traceability guarantees that the resulting specification is both comprehensive and verifiable, thereby addressing the documentation gap that existed in GOHIT v1. The approach aligns with the IEEE 830 principle that each requirement must be testable, unambiguous, and traceable to its origin.

### 3.4 Discussion and Quality Evaluation

The internal quality review, summarized in Table 3, evaluated the SRS against IEEE 830 attributes: completeness, consistency, correctness, verifiability, traceability, and modifiability. Results showed complete coverage and strong internal consistency across all seventeen requirements. Traceability achieved full marks, indicating that every FR is connected to a specific stakeholder or document source. Slightly lower scores in verifiability reflected the need for clearer acceptance criteria for user-interface-related functions.

These findings align with previous studies that highlight the role of structured documentation in preventing ambiguity and rework. Kamaruddin and Rozlis (2019) found that stakeholder participation enhances correctness and acceptance, while Stephen and Mit (2020) demonstrated that adherence to IEEE 830 properties improves requirement precision. The GOHIT v2 case supports both observations: iterative feedback and standardized templates significantly reduced inconsistencies between stakeholder expectations and technical interpretation.

Table 2. Identified Functional Requirements

ID	Requirement	Primary Actor	Priority	Description
FR_001	Register User Account	Participant / Organizer	High	Enable new users to register basic profile information.
FR_002	User Login and Authentication	Participant / Organizer / Admin	High	Provide secure login and session handling for all users.
FR_003	Create and Manage Competition	Organizer	High	Allow organizers to create, edit, and manage competitions.
FR_004	Admin Validation of Competition	Organizer / Admin	High	Require admin approval before competition publication.
FR_005	— Removed / Merged	-	-	Redundant with FR_003 scope.
FR_006	Generate Bracket	Organizer	Medium	Automatically generate competition brackets based on participant data.
FR_007	Update Match Results	Organizer	Medium	Allow entry and updates of match outcomes.
FR_008	Publish Results	Organizer	High	Publish verified results for participant viewing.
FR_009	View Competitions	Participant	High	Display a list of available competitions.
FR_010	Join Competition (Individual or Team)	Participant	High	Permit participants to register individually or within teams.
FR_011	Manage Team (Invite / Leave)	Participant	Medium	Manage team creation, invitations, and membership.
FR_012	Download Certificate	Participant	Medium	Allow participants to download participation or award certificates.
FR_013	Generate Certificate	System / Organizer	Medium	Automatically create certificates after result publication.
FR_014	Search / Filter Competition	Participant	Medium	Enable search and filtering by category or status.
FR_015	Notification System	All Actors	High	Send notifications for approvals, updates, and results.
FR_016	Upgrade Organizer Account	Organizer	Medium	Permit organizers to upgrade to premium accounts.
FR_017	Payment Processing	Organizer / Admin	Medium	Handle online payment transactions for upgrades.

Methodologically, this study demonstrates that a lightweight yet disciplined RE process can achieve high documentation quality even with limited resources and participants. Thematic analysis provided flexibility for capturing nuanced stakeholder intent while maintaining empirical rigor. Such an approach is particularly suitable for academic and startup software projects that lack formal RE teams but require traceable development artefacts.

Future research may extend this work in several directions. First, a follow-up study could examine how the elicited requirements evolve during design and implementation phases, measuring volatility and change management in small-team environments. Second, further work could integrate non-functional requirements (e.g. usability, reliability, and scalability) into the same lightweight RE pipeline. Finally, the quality review performed here provides a foundation for applying a test-based evaluation in subsequent research. By converting each functional requirement into executable test cases, future work can evaluate the extent to which the SRS supports automated verification and continuous quality assurance throughout system development.

Table 3. Requirements Quality Assessment

Quality Attribute (IEEE 830)	Evaluation Criteria	Assessment Method	Remarks
Completeness	Each FR defines input, process, and output elements.	Cross-check FR list with interview themes.	All FRs fully mapped to stakeholder statements.
Consistency	No contradictory or duplicated requirements.	Peer review of SRS by two validators.	Terminology standardized (e.g., “competition”, “organizer”).
Correctness	Requirements represent stakeholder intent accurately.	Member checking and expert validation.	Minor lexical revisions recommended.
Verifiability	Each requirement testable by defined criteria.	Draft test scenario per requirement.	UI-related FRs need clearer acceptance criteria.
Traceability	Each FR linked to its source (stakeholder or document).	Traceability matrix (Actor → FR → Source).	Fully covered within SRS appendix.
Modifiability / Clarity	Requirements written in atomic and simple form.	Manual review using IEEE 830 checklist.	Language refined for clarity and atomicity.

#### 4. Conclusions

This study presented a structured, lightweight requirements-engineering process for the development of the GOHIT v2 platform, a web-based system for managing academic competitions. Through qualitative case-study methods, the research successfully elicited, analyzed, and specified seventeen functional requirements grounded in stakeholder input and verified through internal quality evaluation.

The results demonstrate that even in small, resource-constrained development contexts, a disciplined yet flexible RE process can produce a Software Requirements Specification that is complete, consistent, and traceable. The integration of stakeholder validation and internal quality checks proved effective in minimizing ambiguity and ensuring alignment between business goals and technical interpretation. This approach contributes a practical model for academic or startup teams seeking to formalize requirements without the overhead of heavy documentation frameworks.

Building on these outcomes, future studies may extend the work in several directions. Subsequent research could evaluate the evolution of the documented requirements through the design and implementation phases, examine the management of requirement changes in agile settings, and integrate non-functional attributes into the same lightweight RE pipeline. Finally, the validated SRS developed in this study provides a foundation for applying test-based requirements validation as a future method for systematic verification and continuous quality assurance during software development.

#### 5. References

- Ågren, S., Knauss, E., Heldal, R., Pelliccione, P., Malmqvist, G., & Bodén, J. (2019). The impact of requirements on systems development speed: A multiple-case study in automotive. *Requirements Engineering*, 24(3), 315–340. <https://doi.org/10.1007/s00766-019-00319-8>
- Chikh, A., & Aldayel, M. (2014). Reengineering requirements specification based on IEEE 830 standard and traceability. In A. Abraham (Ed.), *Advances in Intelligent Systems and Computing* (Vol. 265, pp. 211–227). Springer. [https://doi.org/10.1007/978-3-319-05951-8\\_21](https://doi.org/10.1007/978-3-319-05951-8_21)
- Easterbrook, S., Lutz, R., Covington, R., Kelly, J., Ampo, Y., & Hamilton, D. (1998). Experiences using lightweight formal methods for requirements modeling. *IEEE Transactions on Software Engineering*, 24(1), 4–14. <https://doi.org/10.1109/32.663994>
- Elliott, R., & Allen, E. B. (2013). A methodology for creating an IEEE standard 830–1998 software requirements specification document. *Journal of Computing Sciences in Colleges*, 29(1), 123–131

- Farid, W. (2012). The NORMAP methodology: Lightweight engineering of non-functional requirements for agile processes. In Proceedings of the 19th Asia-Pacific Software Engineering Conference (pp. 322–325). IEEE. <https://doi.org/10.1109/APSEC.2012.23>
- Gervasi, V., & Nuseibeh, B. (2000). Lightweight validation of natural language requirements: A case study. In Proceedings of the Fourth International Conference on Requirements Engineering (pp. 140–148). IEEE. <https://doi.org/10.1109/ICRE.2000.855601>
- Stephen, E., & Mit, E. (2020). Evaluation of software requirement specification based on IEEE 830 quality properties. *International Journal on Advanced Science, Engineering and Information Technology*, 10(4), 1396–1402. <https://doi.org/10.18517/ijaseit.10.4.10186>
- Sommerville, I. (2011). *Software engineering* (9th ed.). Pearson Education.
- Tam, N. R. A., Prasetyo, K. W., & Kristanto, B. K. (2022). Analisis kebutuhan sistem informasi manajemen kegiatan kemahasiswaan STIKI Malang menggunakan Agile requirements engineering. *J-INTECH: Journal of Information and Technology*, 10(1), 21–29. <https://doi.org/10.32664/j-intech.v10i1.651>