A Procedure for Requirements Analysis

Dr. Bipasha Maity
Professor, Masters In Business Administration (General Management), Presidency University, Bangalore, India,
Email Id-bipasha@presidencyuniversity.in

ABSTRACT:

Requirements analysis is a critical step in the system engineering process and establishes the groundwork for the effective design and deployment of complex systems. This abstract provides a detailed method for doing requirements analysis in system engineering, emphasising important phases and factors to guarantee a methodical and successful approach. The process starts by highlighting how crucial it is to have a clear grasp of the system's goals and purpose, as well as to identify stakeholders and their requirements. The methodical collection and documenting of needs, using a variety of methodologies including interviews, questionnaires, and workshops, follows. The analysis phase focuses on establishing performance measurements, investigating system limitations, and determining functional and non-functional needs. The approach emphasises the need of verifying and prioritising requirements via engagement with stakeholders and subject matter experts in order to guarantee correctness and completeness. Additionally, it emphasises the need of traceability and the creation of a strong change management procedure to deal with changing requirements throughout the course of the system's lifespan.

KEYWORDS:

Documentation, Enterprise, Project Constraints, Technical Performance Measures.

I. INTRODUCTION

In the discipline of system engineering, requirements analysis is a crucial step since it establishes the groundwork for the effective creation and implementation of complex systems. Clare and succinct requirements must be defined in order to direct the design and development of the system. This requires comprehending, recording, and analysing the needs and expectations of stakeholders. The technique integrates best practices from recognised frameworks and methods, such as the International Council on Systems Engineering (INCOSE) and the Systems Engineering Body of Knowledge (SEBoK). It also recognises that requirements analysis is iterative, encouraging ongoing development via feedback loops.

In order to assure successful system development and deployment, the suggested technique provides an organised and repeatable approach to requirements analysis. It directs system engineers in efficiently gathering, analysing, and managing requirements. Organisations may improve their capacity to provide solutions that satisfy stakeholder demands, reduce risks, and accomplish project goals by adhering to this method. It is customary to adhere to a technique or methodology while doing requirements analysis to guarantee a methodical and efficient approach. Throughout the system development lifecycle, this technique offers a standardised framework for compiling, evaluating, validating, and maintaining requirements [1], [2].

The following essential stages are often included in the requirements analysis process:

Elicitation: In this stage, stakeholders are actively engaged in order to determine their wants, goals, and ideal system features. To obtain information and guarantee a thorough grasp of the needs, techniques including interviews, seminars, questionnaires, and observations are used.

Documentation: After requirements have been acquired, they must be clearly and unambiguously recorded. This involves recording both functional and non-functional requirements, or the characteristics and limitations of the system, which describe what the system should be able to perform. Textual explanations, use cases, graphs, and other suitable forms may all be used as documentation [3], [4].

Analysis: The requirements are then examined to make sure they are consistent, exhaustive, and feasible. To find any conflicts or gaps, this stage entails assessing, prioritising, and defining the requirements. The analytical process is supported by tools like requirement traceability and prioritisation matrices.

Validation: This stage makes sure that the criteria correctly reflect the wants and needs of the stakeholders. This is accomplished by conducting reviews, walkthroughs, and talks with stakeholders to ensure that the requirements are accurate and clear. The criteria are improved and refined with input from stakeholders [5], [6].

Verification: After the requirements have been confirmed, the verification phase examines how well the system has been designed and implemented. Inspections, simulations, and prototyping are just a few of the methods used to make sure the system meets the requirements.

Management: Throughout the system development lifecycle, requirements must be managed via the establishment of a regulated procedure. This entails keeping a requirements traceability matrix, handling change requests, and making sure that stakeholders collaborate and communicate effectively.

System engineers may increase the possibility of a successful system implementation by adhering to a method for requirements analysis. This approach will verify that the system's design and development activities are in line with stakeholder demands. Through the system engineering process, the technique offers an organised and methodical approach to acquiring, documenting, analysing, verifying, and maintaining requirements, facilitating successful collaboration, decision-making, and communication [7], [8].

II. DISCUSSION

An outline of the tasks that make up a strategy for requirements analysis may be found in the section that follows. The 15 requirements analysis tasks mentioned in IEEE P1220 provide the basis for a portion of this hypothetical methodology. When planning engineering activities, this industry standard and others should be referenced to assist identify and arrange acceptable tasks.

The learner should be cautious to modify, that is, add or delete as appropriate to the unique system being constructed, as they do with any procedures. Furthermore, even while these jobs build on one another, they shouldn't be seen as strictly sequential. Every task adds knowledge that can make it necessary to go back and review earlier task selections. All System Engineering activities have this type [9], [10].

Establishing and maintaining a Decision Database

Make sure a mechanism is in place to capture and maintain the decision database before starting a system engineering process. The decision database is a repository of previous technological choices and specifications for future use. It serves as the major tool for keeping requirements traceability.

To suit the next stage of product development, this database decision management system must be created, or the current system must be examined and modified as appropriate. A Requirements Traceability Matrix, which links requirements to subsystems, configuration items, and functional domains, is a crucial component of this database management system. This has to be created, revised, and released often. Every necessity must be documented. An exhaustive approach for completing requirements analysis that fully identifies the crucial activities that must be completed is provided by the IEEE Systems Engineering Standard. The following 15 task areas that need to be examined are presented in Figure 1.

1. Customer expectations
2. Project and enterprise constraints
3. External constraints
4. Operational scenarios
5. Measure of effectiveness (MOEs)
6. System boundaries
7. Interfaces
8. Utilization environments
9. Life cycle
10. Functional requirements
11. Performance requirements
12. Modes of operation
13. Technical performance measures
14. Physical characteristics
15. Human systems integration

Figure 1: IEEE P1220 Requirements Analysis Task Areas [ocw.mit.edu].

Task 1: Customer Expectations

Define and quantify the expectations of the client. The eight core roles, operational requirements papers, mission demands, technology-based opportunities, direct customer contacts, or requirements at a higher system level are all possible sources for them. This task's goal is to ascertain what the system's intended use is and how successfully each function must be carried out. This should include the natural and artificial environments in which the system's product(s) must function or be used, as well as any restrictions (such as financial, cost, or price goals, schedules, technological advancements, non-developmental and reusable items, physical properties, daily operating hours, on-off patterns, etc.).

Task 2: Enterprise and Project Constraints

Determine and specify the restrictions that affect design solutions. The following project-specific constraints may apply: Costs, Updated Technical and Project Plans, Team Assignments and Structure, Control Mechanisms, and Required Metrics for Measuring Progress. Approved Specifications and Baselines Developed from Prior Applications of the Systems Engineering Process. Enterprise limitations may be anything from management choices resulting from a technical assessment to general requirements for the enterprise, standards or guidelines, policies and processes, domain technologies, and the project's allotment of physical, financial, and human resources.

3rd Task: External Limitations

Determine and specify any external limitations that have an influence on design choices or the Systems Engineering Process activities. Technology base, compliance requirements, industry, international, and other general specifications, standards, and guidelines that require compliance for legal, interoperability, or other reasons, threat system capabilities, and interface system capabilities are just a few examples of external constraints.

Task 4: Operational Scenarios

Determine and specify operating scenarios that cover the intended applications for the system product(s). Define predicted interactions with the environment and other systems and physical interconnectivities with interacting systems, platforms, or products for each operating scenario.

Task 5: Measures of Effectiveness and Suitability, or MOE/MOS

Determine and specify the metrics for system effectiveness that capture overall consumer expectations and satisfaction. The system's ability to carry out the customer's purpose is reflected in MOEs. Mission effectiveness, safety, operability, dependability, and other factors are important MOEs. Measures of supportability, maintainability, simplicity of use, and other factors are included in MOSs, which are connected to how effectively the system functions in its intended setting.

Task 6: System Boundaries

Define system boundaries by indicating which system components are under the performing activity's design control and which are outside of it, as well as the anticipated interactions between those components and external or higher-level interacting systems outside the system boundary (including open systems approaches).

Task 7: Interfaces

Define quantitatively (using the open systems method) the functional and physical interfaces to external or higher-level and interacting systems, platforms, and/or products. Mechanical, electrical, thermal, data, control, procedural, and other interactions are examples of functional and physical interfaces. The internal and exterior perspectives of interfaces are also possible. Internal interfaces are ones that speak to objects contained inside the limits of the system being addressed. The contractor in charge of system development typically recognises and manages these interfaces. The government generally defines and regulates external interfaces, which are those that entail entity connections outside of the specified limits.

Task 8: Environments for Use

Set up the settings for every possible operating situation. It is necessary to identify and specify any environmental elements (natural or artificial) that might affect system performance. The following are some examples of environmental factors: temperature ranges, topologies (such as ocean, mountains, deserts, plains, and vegetation), biological (such as animal, insects, birds, and fungi), time (such as dawn, day, night, and dusk), and induced (such as vibration, electromagnetic, and chemical).

Task 9: Concepts of the Life Cycle Process

Determine the essential life cycle process needs based on an analysis of activities 1 through 8 to design, create, test, distribute, operate, maintain, train, and dispose of system goods under development. Organise integrated teams that reflect the eight fundamental functions. The cost-drivers and higher-risk components that are predicted to affect supportability and affordability throughout the system's useful life should be the centre of attention.

Task 10: Functional Requirements

Specify what the system needs to achieve or be able to perform. During functional analysis and allocation, the functions discovered via requirements analysis will be further broken down.

Task 11: Performance Standards

Establish the performance specifications for each higher-level system function. Priority should be given to KPPs created in test plans or flagged as of interest by oversight authorities, as well as performance criteria that address MOEs.

Task 12: Modes of Operation

Describe the several operating modes for the system products that are currently being developed. Included in this definition should be the factors (such as environmental, configurational, operational, etc.) that govern the modes of operation.

Task 13: Technical Performance Measures (TPMs)

Determine the important system performance metrics that will be monitored during the design process. The selection of TPMs should be restricted to vital technical milestones and objectives that, if missed, might jeopardise the project's budget, schedule, or performance. TPMs entail monitoring the KPPs' actual and planned development so that the management may assess the technological advancement on an exception-by-exception basis. Phase influences TPM selection to some degree. They must be reexamined at the start of each phase and at each stage of the systems engineering process.

Technical Performance Measures (TPMs)

System engineers utilise technical performance measures (TPMs), which are quantitative measurements, to gauge and evaluate a system's technical performance throughout the course of its whole lifespan. To determine if a system is reaching its performance goals and requirements, TPMs provide objective and quantifiable criteria. They are used to monitor and assess the system's performance during the development, testing, and operating stages. Typically, they are established during the requirements analysis process.

TPMs are chosen in accordance with the system's objectives and unique features. They might include metrics like speed, precision, dependability, availability, maintainability, safety, and other pertinent factors. They are often generated from system requirements. The important functions and operational goals of the system should be taken into consideration while choosing TPMs.

When defining TPMs in system engineering, the following factors and qualities should be taken into account:

1. To allow for an unbiased assessment, TPMs should be represented in quantifiable terms. This enables the gathering of data and the evaluation of actual performance in relation to predetermined goals or benchmarks.
2. TPMs should be directly related to the performance goals and requirements of the system. They must to concentrate on the essential system features and characteristics that are essential for attaining operational success.
3. TPMs should be specified using suitable units and scales that appropriately reflect the performance of the system. To enable accurate comparisons and analysis, the scales and units employed should be compliant with accepted practises and norms in the industry.
4. TPMs should be viewable and collectible using procedures for acquiring data such measurements, testing, simulations, or monitoring. This makes it possible to gather information to evaluate the system's effectiveness in comparison to the specified TPMs.

Setting baselines and thresholds for TPMs helps in deciding what performance levels are considered acceptable. While thresholds specify the allowable limits within which the system's performance should fall, baselines act as a point of comparison. Throughout the lifespan of the system, TPMs should be continually tracked and examined. This permits the early detection of performance problems, variances, or trends that may call for improvements or remedial measures.

System evaluation and assessment depend heavily on TPMs. They provide unbiased proof to support judgements on system acceptability, operational readiness, or the need for upgrades and adjustments. System engineers may objectively evaluate the technical performance of a system and make choices based on quantitative data by developing and monitoring TPMs. TPMs provide a way to keep track of development, pinpoint areas in need of improvement, and make sure the system achieves its goals and specifications.

III. CONCLUSION

For system engineering, a technique for requirements analysis offers a planned and methodical means of acquiring, evaluating, and managing requirements. By following this process, organisations may improve stakeholder engagement, improve the clarity and quality of requirements, and raise the possibility that a system will be developed that satisfies user needs and expectations. Successful system development depends on effective requirements analysis, which also paves the way for later phases of the system engineering process.

REFERENCES

[1] M. Kamalrudin, L. L. Ow, and S. Sidek, “Requirements defects techniques in requirements analysis: A review,” J. Telecommun. Electron. Comput. Eng., 2018.

[2] E. Febriani and W. S. Dewobroto, “Problems and requirement analysis as a first step to connect researchers and small and medium enterprises (SMEs),” Cogent Bus. Manag., 2018, doi: 10.1080/23311975.2018.1513774.

[3] R. Sheh and I. Monteath, “Defining Explainable AI for Requirements Analysis,” KI - Kunstl. Intelligenz, 2018, doi: 10.1007/s13218-018-0559-3.

[4] S. T. Demirel and R. Das, “Software requirement analysis: Research challenges and technical approaches,” in 6th International Symposium on Digital Forensic and Security, ISDFS 2018 - Proceeding, 2018. doi: 10.1109/ISDFS.2018.8355322.

[5] U. Ulfiah, “Functional requirements analysis of E-counseling,” J. Theor. Appl. Inf. Technol., 2018.

[6] L. Liu, B. Li, S. Zlatanova, and H. Liu, “The path from bim to a 3d indoor framework ⇓ a requirement analysis,” in International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 2018. doi: 10.5194/isprs-archives-XLII-4-373-2018.

[7] S. Kang, H. M. Kim, and H. K. Kim, “Trustworthy smart band: Security requirement analysis with threat modeling,” arXiv. 2018.

[8] G. Junming, “Requirements analysis and management of green logistics and green package under circular economy,” J. Adv. Oxid. Technol., 2018.

[9] K. Slhoub and M. Carvalho, “Towards process standardization for requirements analysis of agent-based systems,” Adv. Sci. Technol. Eng. Syst., 2018, doi: 10.25046/aj030311.

[10] J. Shirogane, J. I. Panach, and O. Past, “Method to define user interfaces in the requirements analysis phase,” in Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2018, 2018. doi: 10.1145/3220134.3220137.