

**ÇANKAYA UNIVERSITY**

**COMPUTER ENGINEERING DEPARTMENT**

**CENG 407**

**LITERATURE REVIEW**

**CSEK Requirement Management System**

**Members**

KIVILCIM IŞIK 202011006

SARPER ERBAR 202011001

ERAY EMİR 202011016

CAN METE BOZAR 202011052

# CONTENTS

Abstract .....	3
Introduction .....	3
Navigating Complexity: The Essential Role of Requirements Management in Modern Projects .....	4
• Requirements Structuring and Traceability: .....	4
• Version Control and Change Management: .....	4
• Document Generation and Reporting: .....	5
• Scalability and Adaptability: .....	5
• Enhanced Collaboration and User Accessibility: .....	5
Overview of IBM DOORS Features and Structure .....	6
1. Project and Folder Structure .....	6
2. Requirement Trace Model .....	7
2.1. Modules .....	7
2.2. Linking .....	8
2.3. Folder, Module, and Linking Models .....	9
3. Change Management .....	10
4. Reporting and Formatting .....	10
Market Research: IBM DOORS Alternative Applications .....	11
1. Primavera .....	11
2. Microsoft Project .....	12
3. GanttProject .....	12
4. Requirements Analysis Tool (RAT) .....	13
5. QuARS (Quality Analyzer for Requirements Specifications) .....	14
From User Requirements to System Specifications: An AI-Based Approach to Requirement Analysis .....	15
❖ Objective .....	16
❖ Methodology .....	16
➤ Linguistic Analysis: .....	16
➤ Pre-Processing: .....	17
➤ Classification with ANN: .....	17
➤ Evaluation: .....	17
➤ Results and Discussion .....	18
Adaptation of this Approach for Transforming User Requirements to System Requirements .....	18
Conclusion .....	19
References .....	20

## **Abstract**

Project success in software and systems engineering is largely dependent on efficient requirements management. A number of alternative tools have surfaced as a result of cost, complexity, or particular project requirements, however IBM DOORS has been a well-known tool for tracking, evaluating, and managing complicated requirements. The structure and functionality of IBM DOORS are examined in this paper, along with its reporting features, requirement trace models, and change management integration. A comparison of alternatives like Primavera, Microsoft Project, and specialized tools like RAT and QuARS is also carried out in order to evaluate their advantages and disadvantages in terms of enabling requirements engineering. Additionally, to improve efficiency and accuracy in requirement analysis, the paper suggests an AI-based method that combines Artificial Neural Networks and Natural Language Processing to automate the conversion of user needs into system specifications. Even though handling complicated linguistic structures can be challenging, the AI-based approach has the potential to greatly enhance the software development process' demand analysis step.

## **Introduction**

Requirements management is a foundational discipline in software and systems engineering, ensuring that all project stakeholders' needs are effectively captured, documented, and tracked throughout the lifecycle of a project. IBM DOORS has long been a leading solution for managing complex requirements, offering robust features such as requirement traceability, version control, and automated reporting. However, due to its high cost and complexity, many organizations seek alternative tools to meet their specific project needs. In this report, we explore the features of IBM DOORS and examine several alternatives, focusing on their strengths and limitations in various project contexts. Additionally, we introduce an innovative approach that integrates Artificial Intelligence to automate the transformation of user requirements into system specifications, highlighting the potential benefits and challenges associated with this technology. By combining insights from industry-standard tools and AI techniques, this report aims to provide a comprehensive understanding of the current landscape in requirements management and offer new avenues for enhancing the efficiency and accuracy of requirement analysis.

# Navigating Complexity: The Essential Role of Requirements Management in Modern Projects

Requirements management is an important and critical discipline in software and systems engineering, offering a structured approach for documenting, tracking, and adapting requirements across a project's lifecycle. Requirements management is essential for ensuring that all project stakeholders' needs are met, reducing the risk of misalignment or project scope issues that can arise when requirements are unclear or inconsistent. In order to reduce risks, eliminate scope creep, and enhance project outcomes, effective requirements management promotes traceability, upholds consistency, and encourages strong documentation.

The role of requirements management becomes significant in large and complex projects, where maintaining control over numerous, interrelated requirements is essential. As a result, requirements management systems have become extremely effective supporters, enabling teams to approach complicated requirements methodically and guaranteeing coordination amongst distant development teams. According to Kühn, Hoffmann, and Weber, requirements management tools provide significant features that facilitate and assist requirements management procedures, meeting a number of essential requirements:

- **Requirements Structuring and Traceability:** Linking requirements to other project artifacts, such as design documents, testing specifications, and implementation records, is a crucial feature of requirements management solutions. Project teams can track changes through each phase, comprehend and manage dependencies, and guarantee consistency throughout development phases thanks to this traceability. In order to improve visibility and lower the possibility of discrepancies, the tools provide bidirectional traceability, which is essential for determining the source of each requirement and its influence on later phases. (Kühn et al., 2004).
- **Version Control and Change Management:** Requirements management solutions help with version control, which makes it simpler to monitor changes over time because updates are frequently needed, particularly in agile or iterative development contexts. Kühn claims that these tools keep thorough change histories that differentiate between significant and small alterations in addition to tracking changes at the level of individual requirements. The level of documentation supports a strong audit trail and helps prevent

the loss of important information by enabling stakeholders to go back to earlier versions if needed. Change requests can be

handled methodically thanks to integrated workflows, which also encourage accountability and offer real-time status information.

- **Document Generation and Reporting:** Requirements management tools simplify documentation by automatically generating reports, specifications, and other essential documents from the requirements. Document consistency and dependability are ensured by this automation, which also saves time and lowers human mistake. According to Kühn, document generation is especially useful in sectors where thorough documentation is required for regulatory compliance or as a component of agreements with outside parties like suppliers in the automotive industry.
- **Scalability and Adaptability:** Requirements management tools are designed to be highly scalable, catering to the diverse and growing needs of large projects. These tools can manage vast amounts of data, and their adaptability allows them to be customized to fit specific project requirements, processes, and organizational needs. Kühn note that requirements management software may be set up to support domain-specific needs, which is especially advantageous for sectors like aerospace and automotive that have unique project workflows or specialized regulatory standards.
- **Enhanced Collaboration and User Accessibility:** Collaboration is fundamental to successful requirements management, particularly in globally distributed teams. Requirements management software includes features that enhance teamwork, such as user-specific views and centralized databases, and allows team members to work on various project components simultaneously while remaining consistent. Additionally, some solutions provide web-based access, which improves accessibility and lowers administrative overhead by enabling infrequent users or outside contributors to interact with requirements without requiring full software installations. (Kühn et al., 2004).

The requirements management process has been revolutionized by the incorporation of requirements management tools into project management workflows. These technologies offer a simplified, automated way to handle complicated requirements, preserve uniformity, and encourage teamwork. Integration features that enable requirements management technologies to work in unison with other development tools, such configuration management and testing software, are becoming more and more important as these tools advance. This integration

supports end-to-end traceability and ensures that each requirement is continuously linked to its corresponding artifacts throughout the product lifecycle.

Research indicates that by decreasing errors, guaranteeing requirements consistency, and promoting adherence to legal requirements, requirements management technologies, when used properly, can greatly improve project outcomes. These solutions are now essential in industries like aerospace and automotive, where big project teams and frequent changes call for a more structured approach to requirements management. In order to better support complicated project needs, requirements management solutions will continue to be developed with an emphasis on accessibility, scalability, and integration with artificial intelligence for predictive analytics.

## **Overview of IBM DOORS Features and Structure**

IBM DOORS (Dynamic Object-Oriented Requirements System) is powerful software designed to manage and track requirements in large-scale projects. It helps organize requirements in a clear project and folder structure. DOORS offers a requirement trace model that includes modules, linking, and specific folder, module, and linking models to ensure that requirements are connected and easy to trace. In addition, its reporting and data export features allow project teams to analyze and share important information efficiently. Together, these features make DOORS a flexible, traceable, and effective tool for requirement management.

### **1. Project and Folder Structure**

IBM DOORS projects and folders serve as containers for information, typically containing formal modules and link modules. Folders are hierarchical (parent and child folders) and are listed alphabetically by default. In smaller projects, placing all requirement modules in one folder may be sufficient.

In some projects, the requirements traceability hierarchy may differ from the physical product structure or bill of materials, often resembling a document structure more closely. If requirements are traditionally managed in a document-based model, this structure may be replicated in DOORS, though complex document structures may not easily translate to DOORS' object-oriented database.

To meet traceability and reporting needs, the structure must be modeled and tested. For this, the ReqMAPS team uses Object Role Modeling (ORM) methodology to model project and folder structures, attributes, views, and link sets. Each sub-folder in the requirements folder structure contains information at the same level within the linking model.

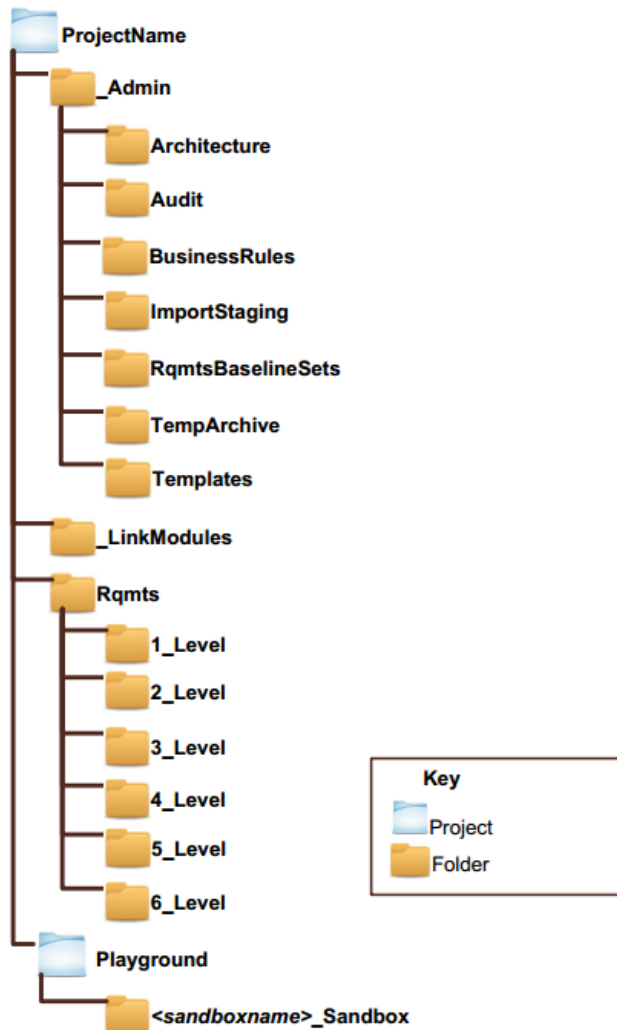


Figure 1: The Standard Folder Structure

## 2. Requirement Trace Model

2.1. **Modules:** Whether requirements are created directly in DOORS or imported from other files, a module/linking model should be established to support linking, traceability, and reporting. Creating requirements in DOORS with a defined information structure helps clarify which modules should be linked and ensures requirements are built to support

relationships, such as satisfies relationships. Reviewing all requirement information and relationships within a model also simplifies validating the linking model.

2.2. **Linking:** In the traceability hierarchy, requirements are allocated downward and traced upward. For traceability reports to function correctly, levels should not be skipped. The same link set is reused across levels to enable full traceability. The direction of linking is essential because linking data is stored in the source module; thus, the source module requires editing permissions, while read-only access is enough for the target module. Different types of relationships, such as between requirement modules and modules like Qualification or Document Reference, can use link sets in opposite directions. All link sets should be created in DOORS before any linking; otherwise, DOORS will create a default link module, which should be removed using an audit script after confirming no links were created with it.

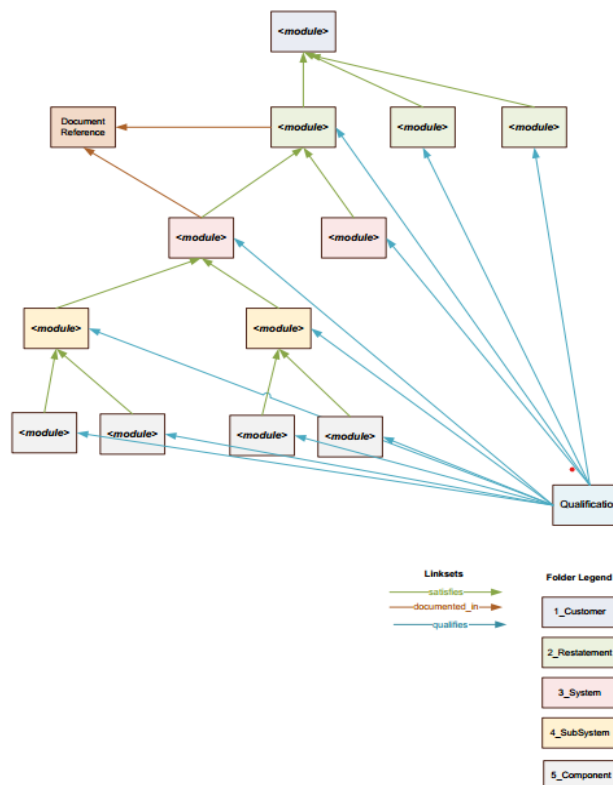


Figure 2: Linking Model Example



2.3. **Folder, Module, and Linking Models:** Two general examples of folder, module, and link set models are provided. The 1\_Level folder might contain customer requirements, while the 2\_Level folder could hold system requirements. To maintain trace integrity, link sets should never skip levels. The examples illustrate traditional vertical structures and flattened structures using the satisfies linking model, suitable for cases where the document set does not fit traditional traceability.

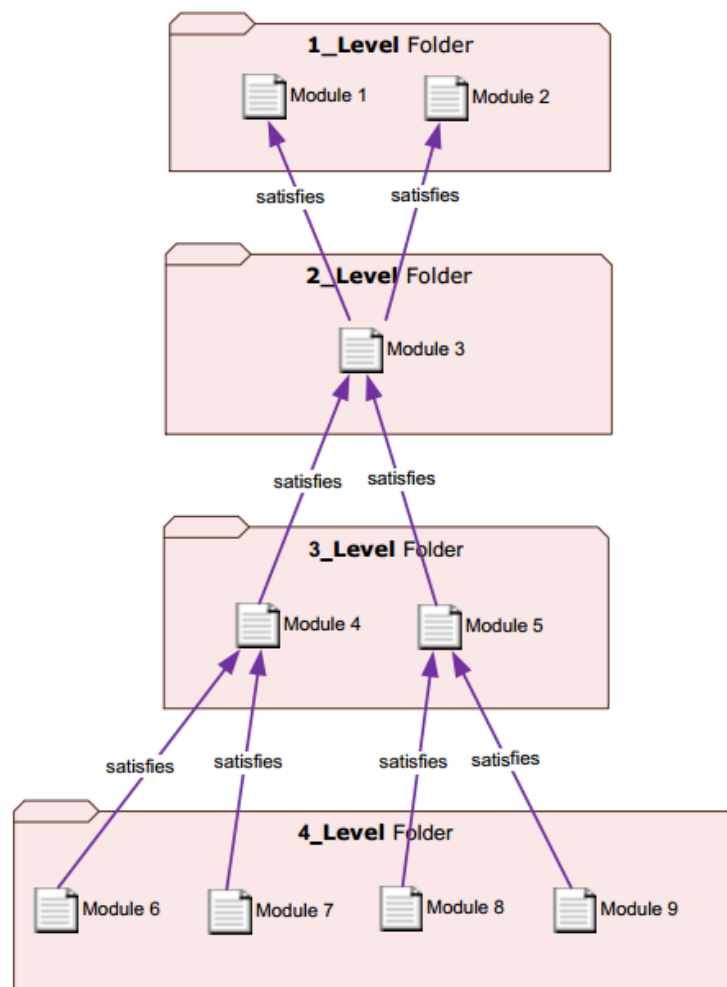


Figure 3 Folder, Module, and Link set Model for the satisfies Vertical Structure

### **3. Change Management**

IBM DOORS integrates change management with traceability, enabling effective tracking of updates to requirements. When changes are made, the system tracks which requirements are affected, what the changes entail, and how these changes may impact the project scope. Documentation of changes and notification of relevant stakeholders are supported by DOORS' robust traceability structure. Users can view the history of changes and understand which modules, requirements, or links are impacted by each update. This ensures that all changes can be reviewed and approved before they impact project requirements, enhancing the reliability of change management.

### **4. Reporting and Formatting**

IBM DOORS provides powerful options for exporting data, such as requirements and traceability links, into MS Word format, enabling users to share project details with stakeholders and maintain clear, accurate documentation. Customizable report templates help tailor reports to specific needs, and traceability information can be included to enhance report completeness. This integration facilitates easy tracking of project progress through detailed documentation.

When exporting data from DOORS to MS Word, formatting can be influenced by the method used. If DOORS' internal tools or custom scripts are employed, the data is sent in a DOORS-specified format, such as indented paragraphs. While this may be suitable for some, it may cause formatting inconsistencies with elements like tabs, bullets, or Rich Text Formatting (RTF), which may not be fully compatible with MS Word.

To address these issues, the Reporting and Publishing Engine (RPE) can be used to export DOORS data with more advanced formatting. By utilizing the `att.RPEStyle` attribute, which corresponds to MS Word's paragraph styles, RPE can instruct MS Word on how to format the content. In this approach, DOORS objects are not pre-formatted, allowing RPE to apply more sophisticated formatting and enabling quick adjustments. This method offers greater flexibility and control over the final appearance of the report, overcoming the limitations of DOORS' native formatting options.

## Market Research: IBM DOORS Alternative Applications

In the field of requirements engineering, IBM DOORS has long been a prominent tool for managing, analyzing, and tracking requirements in complex projects. However, while DOORS is comprehensive, some organizations seek alternatives due to its cost, complexity, or limitations in specific project contexts. This part explores several alternative tools to IBM DOORS, examining their capabilities in requirements management, project scheduling, and requirements analysis. Each tool is reviewed for its strengths and functionalities, but we also highlight critical shortcomings—whether in traceability, integration, usability, or cost-effectiveness—that may impact their suitability as an alternative for IBM DOORS in various engineering environments.

### 1. Primavera

Primavera is primarily a project management tool that is frequently adapted for requirements and project scheduling, particularly in large-scale projects where multi-project planning and tracking are needed. Key features include:

- **Scheduling and Tracking:** Primavera uses the Critical Path Method (CPM) to calculate project schedules based on activity duration and interdependencies. This tool provides flexibility in adjusting project timelines and viewing progress.
- **Resource Management:** The software supports detailed management of labor, material, and non-labor resources, making it suitable for projects with complex resource allocations.
- **Comprehensive Reporting:** Users can generate a variety of reports, enhancing transparency and enabling detailed performance tracking.

#### Drawbacks:

- **High Cost:** Primavera is relatively expensive compared to other tools, which can be a barrier for small to mid-sized projects or companies.
- **Limited Issue Tracking:** Users report limitations in defect and issue tracking, especially in the early stages. Many users noted the need to dedicate separate sprints just for resolving bugs and issues.
- **Complexity and Usability:** Primavera's extensive features can make it cumbersome and difficult to navigate, especially for those not deeply trained in project management.

## 2. Microsoft Project

Microsoft Project is a versatile tool commonly used in requirements and project management due to its intuitive interface and functionality. It is well-suited for planning, scheduling, resource allocation, and tracking project progress. Key functionalities include:

- **Milestone and Task Management:** The tool enables users to define milestones, track project progress against set timelines, and adjust schedules based on project needs.
- **Resource Allocation and Management:** Microsoft Project allows users to assign resources, manage workloads, and track time, giving project managers a clear view of resource utilization.
- **Gantt Chart Visualization:** The tool's Gantt chart feature provides a clear, graphical representation of project timelines, dependencies, and task status.

### Drawbacks:

- **Rigid Scheduling Constraints:** New users may inadvertently apply constraints to tasks, leading to inflexible scheduling. These constraints, particularly when placed on too many tasks, can restrict the ability to adapt to changing project demands.
- **Tracking Limitations for Short Tasks:** Tracking tasks that span only minutes or hours may be impractical, as the tool is more suited for longer-duration projects and milestones.
- **Limited Integration with Other RE Tools:** Microsoft Project lacks strong integration with specialized RE tools, making it challenging to manage requirements tracking and traceability effectively within the same system.

## 3. GanttProject

GanttProject is an open-source, cross-platform tool mainly designed for task scheduling and resource management. Known for its lightweight interface and ease of use, GanttProject is popular in smaller projects or educational settings. Key features include:

- **Gantt and PERT Charting:** It provides Gantt charts for project scheduling and PERT charts for dependency analysis, both useful for visualizing project timelines and relationships between tasks.

- **Resource Allocation:** While basic, the resource allocation feature allows for effective task management across teams.
- **Work Breakdown Structure (WBS):** GanttProject offers a WBS feature that breaks down complex projects into manageable tasks, facilitating project planning.

#### **Drawbacks:**

- **Limited Features for Large-Scale Projects:** GanttProject lacks advanced features, such as intricate reporting and automated tracking, which may make it less effective for large-scale or enterprise-level projects.
- **Basic Resource Management:** Resource management capabilities are limited and may not support complex resource allocation needs as effectively as other RE tools like Microsoft Project.
- **Compatibility and Export Issues:** Although the software supports export to PDF and HTML, interoperability with other RE and project management tools remains limited, potentially creating issues for projects that require extensive reporting.

## **4. Requirements Analysis Tool (RAT)**

Developed by Accenture, the Requirements Analysis Tool (RAT) is specifically designed to analyze requirements documents by performing both syntactic and semantic analysis. RAT combines structured content extraction with domain-specific ontologies, making it highly useful for identifying ambiguities and inconsistencies within requirements. Key features include:

- **Controlled Syntax and User-Defined Glossaries:** RAT enforces structured syntax and user-defined glossaries, which improve consistency and clarity across requirements documentation.
- **Semantic and Syntactic Analysis:** RAT utilizes a semantic engine to perform deep semantic analysis, identifying gaps, conflicts, and dependencies among requirements, which reduces the potential for misinterpretation or ambiguity.
- **Problem Phrase Glossary:** RAT has an extensive glossary of problematic phrases, helping to detect vague terms that could lead to misinterpretations (e.g., ambiguous timeframes like "daily" or terms like "quickly").

**Drawbacks:**

- **High Dependency on Glossary Maintenance:** While effective, RAT's reliance on user-defined glossaries means it requires continuous maintenance to stay relevant to specific project needs and terminologies.
- **Complex Setup for Non-Technical Users:** RAT's features may be challenging for non-technical users due to its emphasis on structured syntax and controlled language, making it less accessible for stakeholders without a technical background.
- **Limited Integration with RE and Project Management Tools:** RAT primarily focuses on requirements analysis and lacks robust integration with tools for broader project management activities, such as scheduling and task tracking.

## **5. QuARS (Quality Analyzer for Requirements Specifications)**

QuARS is a requirements analysis tool known for its support in improving the quality of requirements specifications. It focuses on identifying linguistic and syntactic issues within requirements documents, which helps organizations enhance the clarity and consistency of their requirements. Key features include:

- **Linguistic Analysis:** QuARS evaluates the language used in requirements to identify ambiguities, inconsistencies, and incompleteness, which often lead to misinterpretations during development.
- **Support for Requirements Quality Control:** The tool offers built-in metrics for assessing the quality of requirements, allowing users to measure aspects like clarity, precision, and consistency.
- **Phrasal Analysis:** It uses specific algorithms to scan for problematic phrases, making it especially useful in projects where requirements are captured in natural language.

**Drawbacks:**

- **Limited Scope of Analysis:** QuARS is restricted to linguistic and syntactic analysis, lacking deeper semantic analysis capabilities. This means it may not detect complex interdependencies between requirements effectively.

- **No Native Traceability Features:** QuARS does not support requirements traceability natively, which is a crucial feature for managing changes and tracking requirement lifecycle.
- **Dependency on Well-Written Requirements:** The effectiveness of QuARS depends significantly on the initial quality of requirements. If requirements are already well-structured, the tool's benefit may be limited.

In comparison to IBM DOORS, which provides extensive support for requirements management, traceability, and collaboration in large projects, many alternative tools focus on specific aspects of requirements engineering, such as project management (Primavera, Microsoft Project), linguistic analysis (QuARS), or requirements analysis (RAT). While these tools offer unique advantages, they often lack the comprehensive traceability and integration that DOORS provides.

Each of the tools reviewed has limitations that could affect its suitability depending on project size, complexity, and specific RE needs:

- **For Project Management-Centric Needs:** Primavera and Microsoft Project offer robust scheduling, resource management, and milestone tracking. However, they lack detailed requirements analysis features and may not integrate seamlessly with traceability needs.
- **For Requirements Quality and Analysis:** Tools like RAT and QuARS provide strong analytical capabilities but are often limited by lack of integration with broader project management tools. These tools are suitable for ensuring requirements clarity and consistency but may require additional tools for full project lifecycle support.
- **For Lightweight Project Management:** GanttProject is a good choice for simpler projects but lacks the advanced features necessary for complex requirements management.

## **From User Requirements to System Specifications: An AI-Based Approach to Requirement Analysis**

In this project, we aim to leverage artificial intelligence (AI) technology to automatically generate system requirements from user requirements. The methods and results derived from reviewing similar research are summarized in this section of the report.

This study proposes an approach for automatically extracting key use-case components—specifically actors and actions—from software requirement documents written in natural language. By combining Artificial Neural Networks (ANN) and Natural Language Processing (NLP) techniques, the goal is to enhance the speed and accuracy of requirement analysis. The study utilizes the GATE (General Architecture for Text Engineering) platform for linguistic analysis, followed by classification of extracted features using ANN.

### ❖ Objective

In software development, requirement documents play a crucial role in conveying detailed information about system functionality. However, manually extracting and processing this information is time-consuming and error-prone. The aim of this study is to automate the detection of actors and actions expressed in natural language within these documents, thus improving efficiency.

### ❖ Methodology

The methodology integrates NLP and ANN for requirement analysis, divided into four key stages:

#### ➤ Linguistic Analysis:

- *Tokenization*: The text is broken down into tokens (e.g., words, sentences), which serve as input for further analysis.
- *Syntax Analysis*: The grammatical structure of sentences is analyzed using tools like GATE and Stanford-CoreNLP.
- *Semantic Analysis*: Actions and their associated actors are analyzed based on semantic relationships to identify use-case components.



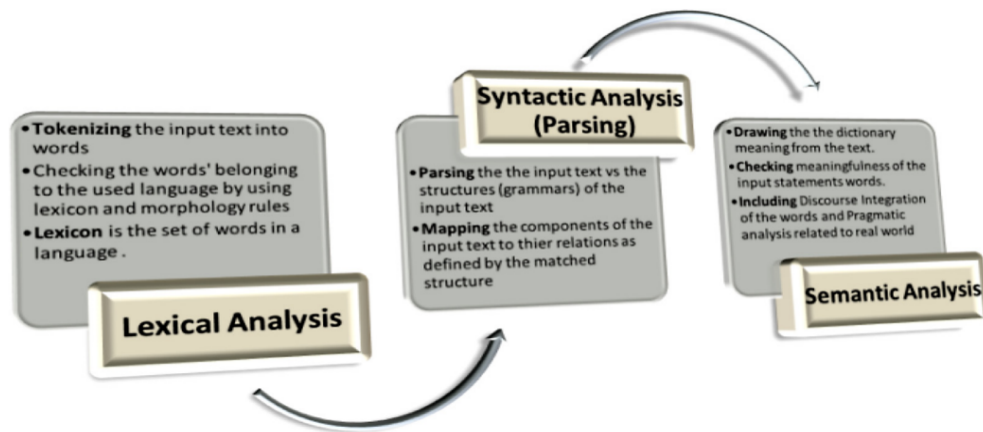


Figure 4 NLP components

➤ **Pre-Processing:**

- Tokens, categories (e.g., nouns, verbs), and dependencies are encoded numerically to prepare them for classification by the ANN.

➤ **Classification with ANN:**

- A Back-Propagation Neural Network (BPNN) classifies linguistic components into use-case labels (actor, action, other).

- The network includes an input layer, hidden layers, and an output layer, with training carried out using MATLAB's neural network tools.

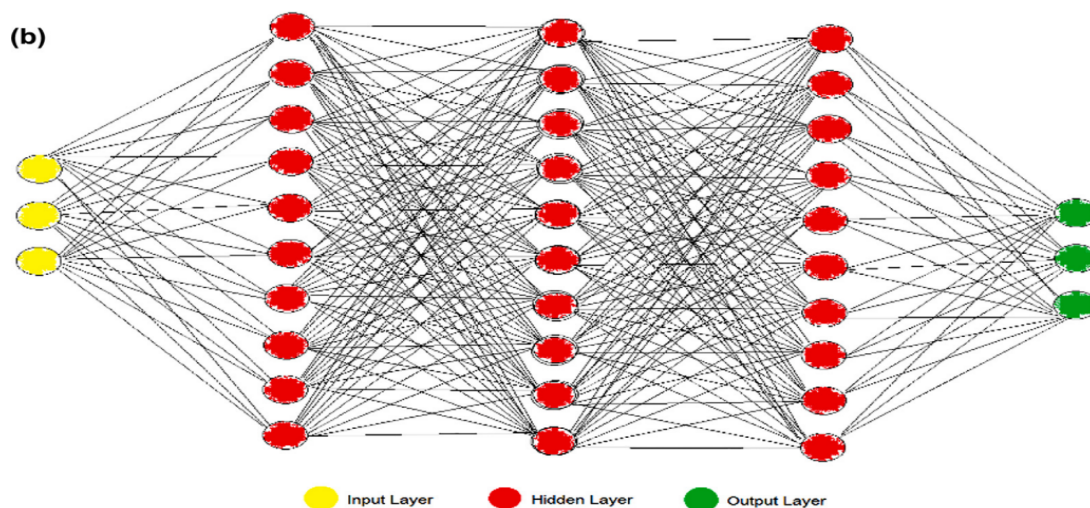


Figure 5 BPNN topology of IT4RE

➤ **Evaluation:**

- The network's performance is assessed using precision, recall, and F-measure metrics to evaluate the accuracy of classification.

## ➤ Results and Discussion

The results indicated that precision varied between 17% and 63%, while recall ranged from 5% to 100%. The average F-measure was 55%, suggesting that the method performed less effectively with complex sentence structures and specialized vocabulary due to limitations in NLP tools. However, this approach demonstrates an efficient and accurate method for requirement analysis by integrating NLP and ANN techniques. Enhancing NLP tools and model parameters is necessary to improve accuracy, especially for handling complex linguistic structures.

### **Adaptation of this Approach for Transforming User Requirements to System Requirements**

This combined ANN and NLP approach can be adapted to automatically convert user requirements into system requirements by interpreting and specifying them in a more technical, system-level language. The process would involve the following steps:

- *Extended Linguistic Analysis*: User requirements, often expressed in high-level terms, would be analyzed for key concepts and inferences. Using Concept Mapping, general terms are mapped to specific system functions or modules, while Inference is used to derive more detailed system requirements.
- *Pre-Processing and Encoding*: Key terms and dependencies are encoded in a technical language, facilitating the definition of system-level dependencies and features.
- *Mapping User Requirements to System Requirements*: A customized ANN model would categorize user requirements into system requirement categories such as functionality, performance, and security. Specific rules guide the transformation process based on recognized patterns.
- *System Requirement Generation*: The model generates detailed system requirements, extracting technical specifications and organizing them into functional modules, thereby providing a structured transition from user requirements to system design.

This method, combining ANN and NLP, offers a promising solution for automating the conversion of user requirements into system requirements. However, to achieve greater accuracy, especially in handling complex language structures, further improvements in NLP tools and model parameters are needed. This approach has the potential to significantly enhance the efficiency of requirement analysis processes in software development projects.

## **Conclusion**

Effective requirements management is essential for the success of complex software and systems engineering projects. IBM DOORS has proven to be a comprehensive solution for handling the intricacies of requirements traceability, version control, and reporting. However, due to its cost and complexity, many organizations turn to alternative tools such as Primavera, Microsoft Project, and specialized analysis tools like RAT and QuARS, each with distinct advantages and drawbacks. While these alternatives may be suitable for specific project needs, they often lack the comprehensive traceability and integration that IBM DOORS offers. Furthermore, the application of Artificial Intelligence, specifically Natural Language Processing (NLP) and Artificial Neural Networks (ANN), presents an exciting opportunity to automate the conversion of user requirements into system requirements. This AI-based approach, though not without challenges in handling complex linguistic structures, shows promise in enhancing the speed and accuracy of requirement analysis, offering significant improvements in project efficiency. Continued advancements in AI and NLP technologies will further refine this approach, enabling more seamless transitions from user requirements to system specifications in the future.

## References

- [1] Aragon, K. M., Eaton, S. M., McCornack, M. T., Shannon, S. A., & Sandia National Laboratories. (2014). Generalized information Architecture for managing requirements in IBM's Rational DOORS® application. In SANDIA REPORT (No. SAND2014-20563). Sandia National Laboratories. <http://www.osti.gov/bridge>
- [2] Kuutti, T. (2019). Comparing Requirements Management Tools – IBM Rational DOORS & HP ALM. In Metropolia University of Applied Sciences, Metropolia University of Applied Sciences.
- [3] Al-Hroob, A., Imam, A. T., & Al-Heisa, R. (2018). The use of artificial neural networks for extracting actions and actors from requirements document. *Information and Software Technology*, 101, 1–15. <https://doi.org/10.1016/j.infsof.2018.04.010>
- [4] Hoffmann, M., Kühn, N., Weber, M., DaimlerChrysler AG Research & Technology, Bittner, M., & TU Berlin, Fak. IV, ISTI, SWT. (2004). Requirements for Requirements Management Tools. In *Proceedings of the 12th IEEE International Requirements Engineering Conference (RE'04)* (pp. 1090-705X) [Conference-proceeding]. IEEE.
- [5] Verma, K., Kass, A., & Accenture Technology Labs. (2008). Requirements Analysis Tool: A Tool for Automatically Analyzing Software Requirements Documents. In *ISWC 2008* (Vol. 5318, pp. 751–763) [Journal-article]. Springer-Verlag Berlin Heidelberg.
- [6] Sajad, M., Sadiq, M., CRIDS (Center for Research in Distributed and Supercomputing) RIU, Naveed, K., Iqbal, M. S., CRIDS (Center for Research in Distributed and Supercomputing) RIU, & School of Computer Science, Anhui University, Hefei, China. (2016). Software Project Management: Tools assessment, Comparison and suggestions for future development. In *IJCSNS International Journal of Computer Science and Network Security* (pp. 31–32).
- [7] The Ultimate Guide to Requirements Management. (n.d.). <https://specinnovations.com/blog/guide-to-requirements-management>
- [8] Weller, J. (2022, August 23). Requirements Management 101: Processes, Plans, and Best Practices. Smartsheet. <https://www.smartsheet.com/content/requirements-management>
- [9] Arvindpdmn. (2024, January 29). Requirements Management. Devopedia. <https://devopedia.org/requirements-management>