



Principles of Requirements Management and Analysis for Supporting Software Engineering Development: A Literature Review

Ashraf Abdulmunim Abdulmajeed ¹ , Atica M. Altaie ²  and Rasha Gh. Alsarraj ³ 

^{1,2,3} Department of Software, College of Computer Sciences and Mathematics, University of Mosul, Mosul, Iraq

Article information

Article history:

Received September 12, 2024
Accepted October 28, 2024
Available December 1, 2024

Keywords:

Requirements Management
Software Development
Software Engineering
Project Management, Software
Development Project Success
Requirements Gathering, Analysis
Requirement Engineering

Correspondence:

Ashraf Abdulmunim Abdulmajeed
ashraf_althanoon@uomosul.edu.iq

Abstract

Requirements management and analysis principles for software engineering projects are of enormous importance in the development of software systems. This article concentrates on examining the literature that highlights the significance of requirements management and analysis principles in bolstering software engineering development. The requirements management process includes a set of important processes that focus on identifying, documenting, analysing, reviewing, and managing the requirements of the software system during the stages of its construction or development. These requirements are the cornerstone of the success of any software project for software companies, as effective and correct management leads to the development of products that meet the needs and expectations of users. This process includes requirements analysis, which transforms user or lab or market requirements into technical specifications for implementation in later stages of the system construction process. This process concentrates on precisely studying and analysing requirements to eliminate any conflict or ambiguity, and to confirm their validity and the extent of their implementation. In general, this literature review reviews the different methods used in managing and analysing requirements by the work team, including requirements collection techniques, requirements verification methods, and managing changes during the requirements collection stage. The review also addresses the challenges facing requirements management in software projects and how to deal with them to ensure the success of the project.

DOI [10.33899/ijqjoss.2024.185257](https://doi.org/10.33899/ijqjoss.2024.185257), ©Authors, 2024, College of Computer Science and Mathematics University of Mosul.
This is an open access article under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years requirements analysis and management has been categorized as a fundamental enabler to any software project and is assumed as a capacity or characteristic of the systems to be developed. In other words, it is the process of getting out functional and non-functional requirements that the customer or consumer demands for a specific system that is to be implemented to meet its goals and objectives within certain standards. Another definition of requirements can also be viewed as specification, expectation, parameter or factor which the system should offer or achieve (Maliha et al., 2023).

Requirements engineering focusses primarily on the way the software team (from within the organisation) interacts internally, which includes the relationship with stakeholders regarding the activities of the systems to be built, such as requirements gathering, analysis, and research and development (Krzysztof et al., 2012).

Evident that the requirements are the most tangible product of software engineering activity. Due to the fact that requirements are the core of the final product all the consequent product steps reflect the requirement. Some mistakes done at this stage would have a completely overwhelming impact on the remaining processes of the project (Mohammad et al., 2010) In this paper, researchers focus on important principles in requirements management and analysis to support software development. We summarize the research review's objectives around four key points (Sujatha et al., 2017):

- 1- Achieving accurate compatibility with user needs: It seeks to make sure that a developed software has all the required customer needs that the users and other stakeholders want, which are managed properly.
 - 2- Reducing errors and costs: Effective requirements analysis in the initial phases increases the probability of error occurrence in subsequent development phases thereby minimizing the cost and time taken to rectify those errors.
 - 3- Dealing with changes: Creating a structure for dealing with alterations to requirements that may be perceived at some point in the software development life cycle and avoiding a situation where these changes compromise the quality of the end product.
 - 4- Enhancing the quality of the final product: With proper control of the specifications and requirements introduced in the subject's framework, the latter helps enhance the quality of the created software and its compliance with recognized standards.
- Figure 1 describes the requirements analysis and documentation stages in system development, showing the different stages involved in the requirements phase, from feasibility study to requirements verification and documentation (Anthony,2010).

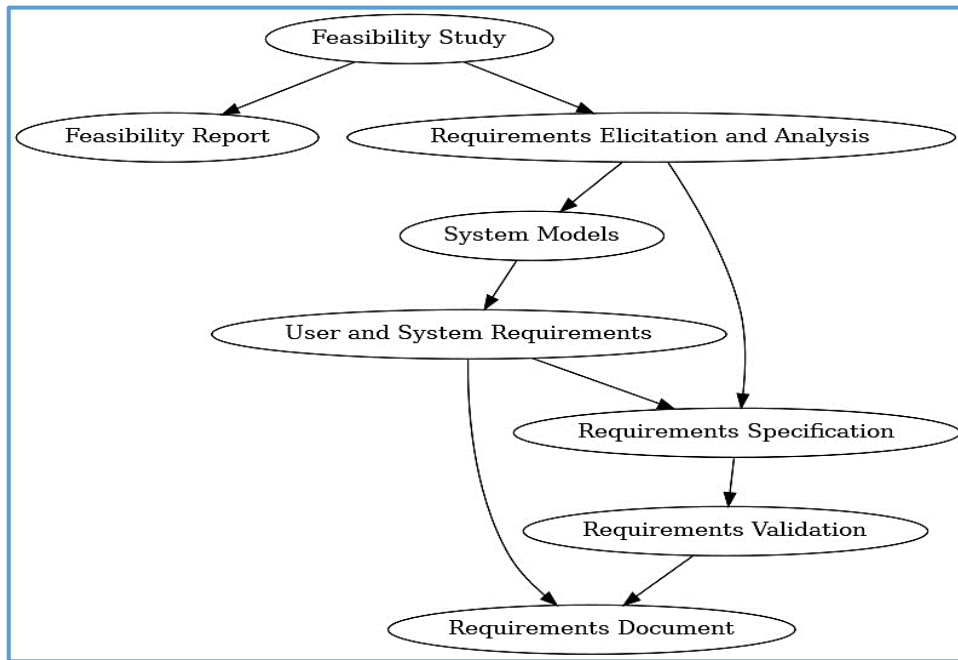


Figure (1) Conceptual of Requirements engineering

In this paper, we review key principles to help in managing and analyzing requirements for effective software development. The structure of the manuscript is organized as follows: In the first part, therefore, we present a brief review of the literature on requirements management and its place in the field of software engineering. The second section looks at techniques and instruments involved in elicitation, verification, and documentation of requirements. The third section of the paper focuses on the Issues and approaches towards handling evolving requirements in project life cycle. Lastly, the last section presents general recommendations on possible further research and development in the area of requirements management. This manuscript seeks to present a synthesis of current and developing practices in order to improve portfolio management and operational performance of software development projects.

2. Related Works

The requirements engineering, as defined in the IEEE glossary is the concept of how well a system or component or process is capable of meeting defined requirements and the customer or user needs or expectations (M. R. 2008). Requirements engineering relates to all those aspects of a product or service that influences its capacity to meet needs- stated or inherent. While in the former one, the software quality is defined by conformance to requirements, this is a manufacturer's point of view and by fitness for use, which is a user's point of view. Therefore, the definitions of the software product's quality also share a significant correlation with the requirements. In other words, it becomes reasonable to take a position that perhaps what is considered as requirements realization in the real sense is none other than software quality (A. S. 2007).

In this section, we will look at a variety of articles that cover the principles of requirements management and analysis to support software engineering development.

Zagajsek and et al (B. Zagajsek et al., 2007). suggest a requirements management process Where it shows Major cause for any project failure is related to inadequate requirements definition and the use of wrong approach/ methodology towards requirements. That is also the main reason for cost and schedule overruns which in turn influences conflict with the clients. The success of a project is usually determined by 5 common performance factors they are time, cost, performance, and user convenience and last but not the least organizational influence. When proper communication of the project starts as well as distributing roles and responsibilities, it is necessary to start building the requirements management team to continue with the requirements elicitation process and develop the requirements elicitation process. Some of the members required for forming a requirements management team are System Architects for designing the system architecture, end users, business domain specialists, Technology consultants well aware of new innovative technologies, line management and developers.

Chu and et al (C-W Lu et al., 2007) presented a new strategy to handle with the uncertainties in the requirements documentation and described a framework known as Model based Object oriented Requirement Engineering (MORE). The useful domain knowledge shall be captured by employing MoReq phases for Requirement Engineering Quality in Modeling & Object-Oriented technologies.

Another study Aziz and et al (Ruhaya et al., 2021) Thus, requirements relationships knowledge that describes the information of the relationships between requirements is defined. concern stems on how requirements are connected to other requirements and other artefacts throughout the SDP. Indeed, it is thought that for this purpose, that is, in the process of categorizing and structuring of the requirements documentation and specification, requirements relationships knowledge may be useful. Since requirements relationships knowledge has the direct effects on the requirements quality and requirements quality has the direct significant on this basis it can be stated that the requirements relationships knowledge that affects success is the other factor that has the effect on the requirements quality and hence this factor influences the success of this project.

Another study (Othman et al., 2021) provides a comprehensive review that shall outline the difficulties experienced in Requirements Engineering (RE) when in Agile Software Development (ASD) context. The authors present a structured approach to the ways of making the requirements management and elicitation challenging in the context of the agile methodologies, which are based on the use of lightweight and iterative processes. Some of the issues that the research raises include handling of scenarios where requirements are open and may be in the process of being refined and the effective communication between stakeholders, and how the dynamic nature of requirements in agile can be managed with the fast pace of development. The implications of the paper are that significant improvements in collaboration practices and the use of automated requirements engineering tools would offer ways of addressing the challenges identified in the paper.

Mishra and et al (Mishra et al., 2021) determining the nature and extent of effects of Requirements Engineering (RE) practices on software projects' success from an empirical perspective. In their studies the authors analyse correlations between different practices of RE including requirements elicitation, analysis, specification and validation practices and their influence on overall results of a project. It is an important note for the study that clear RE processes play a large extent of contribution for software projects, such as enhanced communication among the stakeholders, project risks minimized, and the final deliverables achieved in accordance to the intended goals. The authors propose that it could be possible for organisation to apply these practices systematically in order to enhance on project performance and customer satisfaction.

Another study (Li et al., 2022) refers to the trends regarding the automation of the Requirements Engineering (RE) process and examines the respective issues and prospects. The authors recognize that automation, in general, in RE has benefits, such as; increased productivity, reduction of errors, and harmonization of the requirements lifecycle. However, they report on some concerns and issues: Issues like, requirement elicitation based on natural language processing, the problem with tacit knowledge acquisition, and incorporation of tools into the work environment. The study also discusses the current technologies' inability to capture and adequately analyze the stakeholder needs. This paper provides the following recommendations for future work: the enhancement of extant machine learning for RE, design and development of versatile tools for RE automation and improvement of human-machine interfaces for RE.

Lee and et al (Alrub et al., 2020) focus on enhancing the process of the prioritization of software requirements via machine learning algorithms. Thus, the next crucial component that needs to be addressed is the issue of requirements prioritization since often there are more requirements than is possible to implement the software development project. The authors suggest that, instead of manually claiming priorities and applying such to PubMed records, machine learning algorithms should be

introduced in order to enhance and speed up the prioritization step in opposition to previous approaches. The findings show how the integration of machine learning might work for prioritizing the software requirements based on historical data, stakeholders' preference and project constraints. The results show that the use of the machine learning-based approach minimizes the human bias in decision-making, ensures prioritization of the feature set for the development teams. Therefore, the research establishes that incorporating machine learning in requirements prioritisation paradigm presents advantages within large software development projects especially in environments that are prone to change. It is very important is that requirements changes should be a very critical step in any of the software developing phase. Regarding the requirements changes management this is an open issue in the literature for many decades. Of course, there are some requirements change approaches which has been introduced in literature but none of the seal alone handle with this requirement change. Leveraging on this, Lu and Yuan (G. Lu et al., 2010) have proposed an algorithm for determining whether there is a change in requirement or not. The new proposed algorithm is named LCS-NP++ and this approach produces several variants of the requirement change. Arguing from the accuracy measures, the version of requirements change is selected which is most effective and accurate. First, this technique is not only capable to identify the requirements changes but also can be really effective if someone has to look for the HTML structure.

Another study (Nemat, 2019) discusses ways through which Agile Requirements Engineering (RE) enables incorporation of dynamic stakeholder requirements into a system. The authors draw attention to the fact that that agility is based upon flexibility, and is therefore particularly suitable for contexts with fluctuating stakeholder demands. The paper looks into practices that are applied in agile RE including; the use of the iterative processing, stakeholder interactions and feedback processing that helps the teams to make constant enhancement of the requirements throughout the processing of the project. Thus, the study's focus on the importance of integrating change as early and as often as possible is used to minimize project risks, as well as enhance the fit between the final product and stakeholders' requirements. The paper is concluded by explaining that communication, tools and flexibility of a team are critical success factors in managing requirements in fast changing and fast paced development contexts.

Pavanasam and Subramaniam (V. Pavanasam et al., 2010) suggest a model of requirements inspection and explain that both the stages of requirements change management and requirements analysis are quite significant in the process. Each of the requirement engineering techniques involves stages of requirement transformation and system requirement representation, which in this case is verified using JDesigner tool. The authors identify proper weights for the elements under the amalgamation, abstraction, and the clarity aspects in the transformation of the requirements. Subsequently, this data is distributed to system requirements which bring consistency, completeness, correctness factors respectively 0. 5%, 0. 7%, and 0. 6% weights. System requirements are the functional requirements in essence. Finally, the membrane computing model quantifies the system needs as the product of a constant with the abstraction, amalgamation, and clarity factors. It is also important to note that the model requires knowledge of mathematics which any times might be difficult to comes up with.

Aminu and et al (Muhammad et al., 2024) discusses as a systematic review of developments in the automation of various facets of Requirements Engineering (RE). It reviews how automation tools have grown to meet the need to enhance certain RE activities such as elicitation, analysis, validation and management. A close examination of the knowledge will indicate the advantages of the automated system, which includes better efficiency, fewer mistakes by workers, and high quality of requirement documentation. It also discusses issues that organisations face when adopting tools in RE, including tool integration and compatibility with existing processes, coping with intricate demands, and coordinating between the automated tools and RE teams. In the future, introduction of AI and machine learning are expected to enhance automation of RE tools since the paper implies that the efficiency of these tools will be enhanced in the future.

Obaidi and et al (Deters et al., 2024) focuses on the enhancement of quality of requirements in software engineering through the application of quality models that improve the understandability and explainability of systems requirements. The primary purpose of the research is to guarantee that the requirements are properly defined, described, and documented, and the related problem is solved on the example of the software project with the guidance of the main principal. This can result in improved requirements analysis and, as a consequence, in the design of systems that are more capable to address the issue and satisfy stakeholder requirements and project objectives. Thus, the paper focuses on minimizing the ambiguity and increasing stakeholders and development teams' understanding during the whole software development process applying these quality models. Table 1. Shows the summary for each paper that included in our review is concluded.

Table (1) is a summary of information for the different sources used in this review.

Author(s)	Study Focus	Key Findings
Wojcik et al (B. Zagajsek et al., 2021)	Requirements Management Process	Inadequate requirements definition causes project failures; effective requirements management team is crucial.
Wang et al (C-W Lu et al., 2007)	Handling Uncertainties in Requirements	Reduced Model Based Object-Oriented-Requirement Engineering (MORE) to capture domain knowledge in modelling.
Alzahrani et al (Ruhaya et al., 2021)	Requirements Relationships Knowledge	Understanding relationships between requirements can aid in documentation structuring and project success.
Alshaykh (Othman et al., 2021)	Software Development & Requirements Engineering	Challenges include managing evolving requirements and improving communication; automated tools could help.
Alkhatib et al (Mishra et al., 2021)	Empirical Study on RE Practices	Clear RE processes enhance project performance and customer satisfaction; systematic application is beneficial.
Alshaykh (Li et al., 2022)	Automation in Requirements Engineering	Automation benefits productivity and error reduction but faces challenges like natural language processing.
Alshaykh et al (Alrub et al., 2020)	Prioritization of Requirements using Machine Learning	Machine learning can improve prioritization by reducing human bias and adapting to changing project needs.
Alshaykh & Yuan (G. Lu et al., 2010)	Requirements Change Management	Proposed LCS-NP++ algorithm to identify and manage requirements changes effectively.
Alshaykh (Nemat , 2019)	Agile Requirements Engineering	Agile supports dynamic stakeholder requirements; emphasizes iterative processing and flexibility.
Alshaykh Pavanam & Subramaniam (V. Pavanam et al., 2010)	Requirements Inspection Model	Developed model using JDesigner for requirements transformation emphasizes accuracy and clarity.
Alshaykh et al (Muhammad et al., 2020)	Automation in Various Aspects of RE	Automation tools enhance productivity and accuracy; future advances in AI and machine learning are anticipated.
Alshaykh et al (Deters et al., 2024)	Improving Quality of Requirements	Formal quality models to enhance clarity and alignment with business objectives; reduces ambiguity and improves communication.

Through analyzing previous works, we conclude that requirements engineering focuses on the capability of the system or component to satisfy a set of requirements or user needs. Meeting these requirements is closely linked with quality in software, as quality is recognized as one of the measures which determine the success of the project. Not being able to define requirements properly results in more costs and customer conflict hence making it crucial to manage requirements well. Some current researches suggest the requirement management methods using some approaches, MORE for instance to tackle issues of uncertainty in documents Besides, the associations between requirements are considered as the key to increase the quality of the requirements. As the requirements are dynamic in agile systems the new challenge arises to manage that, the usage of the automated tool is recommended to provide better performance with less chances of error. Other researchers show its ability to prioritize, plan, and outline changes with high efficiency, which gives increased flexibility in project management and minimizes risks. Conversely, automation in requirements engineering improves efficiency and quality because all methods prone to errors, but it poses the problem of how to integrate the new tools with current tools at the workplace.

3. Practices In Requirements Management

Managing software requirements is a delicate process even in the most favorable of settings. Thus, it is quite important to simplify in some cases and enhance the probability of success before applying such approaches. The practices followed in requirements management can be classified as follows (Ali ,2015):

A- Requirements Definition: It is important to define all requirements accurately and clearly from the beginning, including functional and non-functional requirements. Requirements should be understandable by all stakeholders such as stakeholders, developers, and designers, and should be well documented to avoid any ambiguity (B. W. Boehm, 1991). Requirements Defined Process in Software Engineering Through Verification and Approval. Figure (2) shows the requirements definition process in software engineering in addition to Verification and Validation

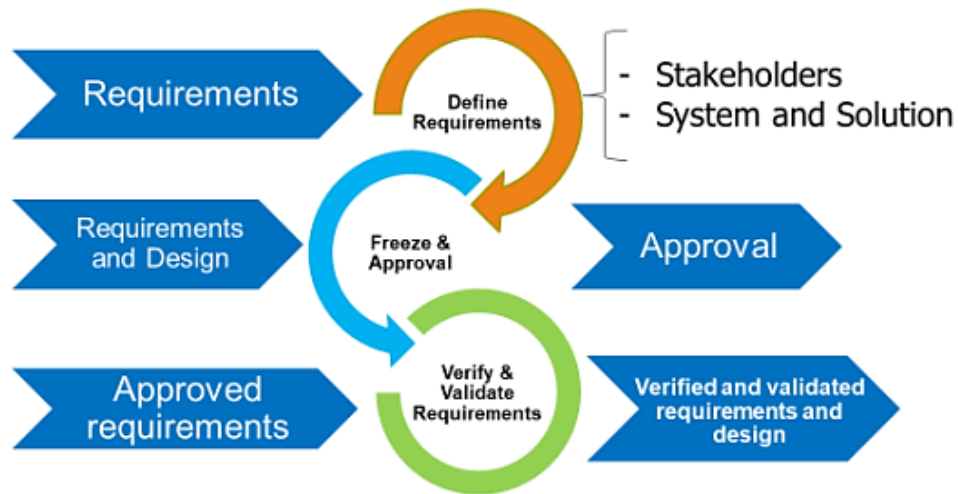


Figure (2) Requirements Defined Process in Software Engineering

B- Requirements Analysis and Documentation: Requirements Analysis and Documentation step in software engineering, and it includes the following (Catania ,2006):

1- Requirements Analysis: This stage concentrates on elaboration, since the requirements gathered during the requirements gathering stage need to be studied in order to meet the needs of all stakeholders or partners and to provide a clear understanding of the requirements. It also assists in the prioritization process of identifying which requirements are important and need to be met most urgently. There must also be integration and consistency by ensuring that the requirements are integrated and do not overlap as well as ensuring the requirements are in line with the project goals.

2- Requirements Documentation: The documentation stage is executed by producing an exhaustive requirement document after clearly assessing the requirements in a formal document that encompasses each of the requirements in terms of description, objective, standard, and performance expectation (Catania ,2006).

3- Functional Requirements Specification: In this stage, the team that is charged with the responsibility of developing the product has an understanding of what the system or product is supposed to do given the understanding that the requirements are dynamic and are subjected to change from time to time; therefore, the documents are to be reviewed at certain intervals so that any changes are captured and made available to all parties (Kurtanovi et al., 2017).

The above steps are critical in preventing misconceptions when it comes to the implementation of the constitution. As the technical details are thoroughly reviewed and explained with the help of documents, the risk of miscommunication between the technical team and stakeholders is minimized.

C- Requirements Verification and Validation: Verification and validation are some of the key activities in software engineering that aims at improving the quality of the developed system during general system development and during requirement elicitation (Elizabeth et al., 2012).

1- Verification of requirements: It is meant to confirm that the system meets the agreed requirements, and to conduct checks to confirm that the developed product meets the required specifications at a particular phase of development.

2- Validation of requirements: It intends to guarantee that the system provides satisfaction to users and customers as desired. In other words, it is aimed at guaranteeing that the final system delivers what is expected from it, its goals and objectives being met.

The purpose of these processes is a minimization of mistakes that may occur during production and enhancement of the final result.

4. Requirements Management Tools

Requirement management tools are implemented kinds of applications, serving the objective of requirements management. A toolkit can contain at least one tool, and their purpose is to monitor activities that the human brain cannot handle. Below is a list of factors that should be taken into consideration when developing a requirements management tool (Shams et al., 2021). Use of requirements management tools is based on:

1- Number of Requirements: For projects with less than 200 requirements, it is possible to use simple tools such as spreadsheet, wiki or simple database. This relates more to background and rationale but to specify it, a lightweight requirements management tool should be used in order to preserve : the requirements knowledge-base of the organization, historical data from previous projects remains for future reuse. A project that involves 200-2000 requirements is considered as a medium size and such a project requires the use of a commercial tool. tables, databases, or wikis do not meet the versatility and reliability of commercial requirements management tools. Huge extent projects with over 2000 requirements in effect require sound and sturdy commercial requirements management tools to avoid any potential unnecessary project hold up, defects and failures (Shams et al., 2021).

2- Dimensions of the project team’s participants: When it comes to measuring the project, most organizations will use the number of people in a software development team. A commercial requirements management tool is not necessary for the project as it has less than 5. that can be used to provide feedbacks to members in a team and they are co-located. Regular table applications, word process applications, wiki or even index carding is enough for addressing the requirements of the project. A commercial requirements management tool is greatly benefited for larger teams that are spread across different cities or even different countries (Shams et al., 2021).

Such tools are essential for proper product lifecycle management, project and change management, quality control improvement as well as compliance enforcement. Requirements management can be combined with task management to create a one-to-one relationship between the project goals and drills down into the actual labor, which drives efficiency. This is useful in maintaining links between requirements and test cases which will help in testing the application to meet required objectives. Collaboration among the teams- it helps to adapt clear communication and a shared understanding of what is required so as to achieve project goals, making work together fun and motivating. It also serves in risk management by defining tracking and managing requirements to help with scope creep and miscommunication. Thirdly, it offers a defined structure in change management that helps capture, communicate and implement changes which reduce ambiguity avoiding misalignment with stakeholder expectations (M. Cadena-Romero et al., 2019).

Table (2) shows a summary of the best requirements management tools in building and developing software systems.

Table (2) Summary of Top Requirements Management Tools.

Tool Name	Version	Company
Agosense.fidelia	1.8.19	agosense GmbH
Aligned Elements	2.5 SP1	Aligned AG
ALM / Quality Center	15.0.1	Micro Focus
ALM Octane	15.0.40	Micro Focus
CodeBeamer ALM	9.5	Intland Software GmbH

Cognition Cockpit	9.2	Cognition Corporation
Cradle	7.5	3SL
Dimensions RM	12.7.1	Micro Focus
DocSheets	AI-driven	Goda Software
ENOVIA 3DEXPERIENCE	R2020x	Dassault Systèmes
Enterprise Architect	15.1	Sparx Systems
Helix RM	2019.5.0	Perforce
IBM Doors	Latest	IBM
Innoslate	4.2.1	SPEC Innovations
Innovator for Business Analysts	14.1	Innovator
in-STEP BLUE	6.5	microTool GmbH
iRise	11.6	iRise
Jama Connect	8.47	Jama Software
Kovair ALM Studio	9.8	Kovair Software, Inc.
Modern Requirements4DevOps	2019 Update 2	eDev Technologies
objectiF RM	6.0	microTool GmbH
objectiF RPM	5.2	microTool GmbH
OneDesk Product Management	January 2020	OneDesk Inc
Polarion Requirements	19.3	iemens Industry Software GmbH
PREEvision	9.0.11	Vektor Informatik GmbH
Psoda	8.03.1	Psoda
Requirements Management for JIR	Latest	JIRA
Rally Software	Q1 2020	Broadcom
RaQuest	4.4	SparxSystems Japan Co., Ltd
ReqEdit	2.15	REQTEAM GmbH
ReqSuite RM	3.0	OSSENO Software GmbH
ReQtest	Latest	ReQtest AB
RequirementONE	Latest	RequirementOne Inc.
ReqView	2.8.1	eccam s.r.o.
RMSis	2.0.5	optimizory Technologies Pvt. Ltd.
Specification Wizard	UML based	MeetAdd
SpiraTeam	6.3	Inflectra
Visure Requirements	5.2	Visure Solutions, Inc.
Windchill RV&S	12.3.0.0	PTC
Yonix	Latest	Yonix Ltd.

Requirements management tools in software engineering improve, manage and allocate critical resources by organizing the management of project resources to be developed and significantly by providing several key benefits. (Rajat et al., 2003):

- 1. Clear Insight on Project Requirements:** Due to the fact RM tools provide an in-depth project requirements overview, you know what resources are needed.
- 2. Requirement Prioritization:** These tools provide you with the capability to prioritize requirements so that resources are allocated towards tasks which can help complete projects more efficiently.
- 3. Requirements and resource tracking:** Real-time requirement status of requirements and resources along with over allocation or underutilization will be immediately visible.
- 4. For forecasting and planning:** Through an analysis of past projects, RM tools support in calculating what kind of resources required for future generation.
- 5. Facilitating Collaboration:** They ensure the collab across team, by putting in one place all requirements to get clarity on roles and responsibilities so that resources will be utilized appropriately.

Conclusion

Requirements engineering is crucial in determining how well a system, component, or process meets defined requirements and user expectations. This process is integral to both software quality and project success, as quality often correlates closely with how well requirements are met. Inadequate requirement definition can lead to project failures, cost overruns, and client conflicts, highlighting the importance of effective requirements management. Various studies suggest that using structured approaches, such as the Model-Based Object-Oriented Requirements Engineering (MORE) framework, can address uncertainties in requirements documentation. Understanding the relationships between requirements helps improve documentation and project success. Managing software requirements involves a series of critical practices to ensure project success. Initially, clear and accurate requirements definition is essential, covering both functional and non-functional needs, and ensuring these requirements are well-documented and understood by all stakeholders. The next step, requirements analysis and documentation, involves elaborating on gathered requirements, prioritizing them, and ensuring they align with project goals. This stage also includes producing a comprehensive document that details each requirement and periodically reviewing these documents to capture any changes. Verification and validation processes are crucial for confirming that the system meets the agreed requirements and provides the desired user satisfaction. Effective requirements management also relies on tools that help manage and track requirements throughout the project lifecycle. These tools vary in complexity based on the number of requirements and team size. They facilitate resource allocation, prioritization, tracking, and collaboration, ultimately enhancing project efficiency and success.

Acknowledgment

The authors are very grateful to the University of Mosul, and College of Computer Science and Mathematics, which helped improve this work's quality.

References

- 1- A. S. White, (2007) "Control Engineering Analysis of Software project Management," BCS SQM Conference, Stafford, Section 3, April.
- 2- Ali Altalbe, (2015), "Software Requirements Management", (IJARAI) International Journal of Advanced Research in Artificial Intelligence, Vol. 4, No.4, 2015.
- 3- Alrub, A., & Lee, S. P. (2020) "Enhancing Software Requirements Prioritization Using Machine Learning Techniques", Applied Soft Computing, 96, 106674.
- 4- Anthony White, (2010), "A Review of the Impact of Requirements on Software Project Development Using a Control Theoretic Model", J. Software Engineering & Applications, JSEA.
- 5- B. W. Boehm, (1991), "Software risk management: principles and practices," Software, IEEE, vol. 8, no. 1, pp. 32–41.
- 6- B. Zagajsek, K. Separovic and Z. Car, (2007) "Requirements Management Process Model for Software Development Based on Legacy System Functionalities", 9th International Conference on Telecommunications (ConTEL 2007) pp. 115 - 122, 13-15 June.
- 7- Catanio, J.T. (2006). "Requirements Analysis: A Review. In: Sobh, T., Elleithy, K. (eds) Advances in Systems, Computing Sciences and Software Engineering". Springer, Dordrecht. https://doi.org/10.1007/1-4020-5263-4_64
- 8- C-W Lu, W. C. Chu and C-H Chang, (2007), "A Model-based Object-oriented Approach to Requirement Engineering (MORE)", 31st Annual International Computer Software and Applications Conference (COMPSAC 2007), Hsiuping Institute of Technology, Taichung, Volume 1, pp. 153 - 156, 24-27 July.
- 9- Deters, H., Droste, J., Obaidi, M., & Schneider, K. (2024), "Quality Models for Requirements Engineering".
- 10- Elizabeth Bjarnason, Per Runeson, Markus Borg, Michael Unterkalmsteiner, Emelie Engström , Björn Regnell , Giedre Sabaliauskaite, Annabella Loconsole, Tony Gorschek, Robert Feldt, (2012), "Challenges and Practices in Aligning Requirements with Verification and Validation: A Case Study of Six Companies
- 11- G. Lu and F. Yuan, (2010), "Comparison of Requirement Items based on the Requirements Change Management System of QONE", Second World Congress on Software Engineering (WCSE), Institute of Computer Technology, Chinese Academy of Science, Beijing, China.
- 12- Krzysztof Wnuk, Dietmar Pfahl, David Callele, Even-Andre Karlsson, (2012) " How can Open Source Software Development Help Requirements Management Gain the Potential of Open Innovation: An Exploratory Study" , In Proceedings Of The ACM-IEEE International Symposium On Empirical Software Engineering And Measurement (ESEM'12) (pp. 271-279), IEEE.
- 13- Kurtanović, Z.; Maalej, W. (2017), "Automatically classifying functional and non-functional requirements using supervised machine learning". In Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference (RE), Lisbon, Portugal, 4–8 September; pp. 490–495.

- 14- Li, J., Heindl, M., & Lock, R. (2022). "Towards Automated Requirements Engineering: Challenges and Future Directions", IEEE Transactions on Software Engineering, 48(3), 623-640.
- 15- M. Cadena-Romero, J. O. Ocharán-Hernández, M. de los Ángeles Arenas-Valdés and J. C. Pérez-Arriaga, (2019) "Tool-Supported Requirement Engineering: A Categorization of the State of the Art and Research Trends," 2019 4th International Conference on System Reliability and Safety (ICSRS), Rome, Italy, pp. 514-523, doi: 10.1109/ICSRS48664.2019.8987610.
- 16- M. R. McGee-Lennon, (2008) "Requirements Engineering for Home Care Technology," Proceeding of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems, Florence, Italy, pp. 1439-1442.
- 17- Maliha Tasnim, Maruf Rayhan, Zheyang Zhang, Timo Poranen, (2023) "A Systematic literature Review on Requirements Engineering Practices and Challenges in Open-Source Projects Maliha", Conference Paper, ResearchGate.
- 18- Mishra, D., & Mishra, A. (2021). "Impact of Requirements Engineering Practices on Software Project Success: An Empirical Study", Journal of Systems and Software, 176, 110927.
- 19- Mohammad Yamin, Venera Zuna, Moteb Al Bugami, (2010), "Requirements Analysis and Traceability at CIM Level", J. Software Engineering & Applications JSEA.
- 20- Muhammad Aminu Umar, Kevin Lano, (2024), "Advances in automated support for requirements engineering: a systematic literature review", Requirements Engineering (2024) 29:177–207
- 21- Nemat Dehghani, (2019), "Defining Requirements Management Process for Product Development Projects", Helsinki Metropolia University of Applied Sciences, Master's Thesis.
- 22- Othman, M., Ghani, I., & Ahmad, R. (2021). "A Systematic Review of Requirements Engineering Challenges in Agile Software Development", Journal of Software: Evolution and Process, 33(1), e2303.
- 23- Rajat R. Sud, J. Arthur, (2003) "Requirements Management Tools: A Quantitative Assessment", Department of Computer Science, Virginia Polytechnic Institute & State University.
- 24- Ruhaya Ab. Aziz, Bernard Wong, (2021), "The Impacts of Requirements Relationships Knowledge on Requirements Quality and Software Development Project Success", 9th International Workshop on Quantitative Approaches to Software Quality, Dec 06, 2021, Taipei, Taiwan.
- 25- Shams Tabrez Siddiqui, M.U. Bokhari, (2021), "Needs, Types and Benefits of Requirements Management Tools", International Journal of Trends in Computer Science Volume 2, Issue 11.
- 26- Sujatha Alla, Pilar Pazos, (2017) "The Impact of Requirements Management Documentation on Software Project Outcomes in Health Care", Proceedings of the 2017 Industrial and Systems Engineering Conference K. Coperich, E. Cudney, H. Nembhard, eds.
- 27- V. Pavanam, C. Subramaniam, T. Srinivasan, J. K. Jain, (2010) "Membrane Computing Model for Software Requirement Engineering", Second International Conference on Computer and Network Technology (ICCNT), Sathyabama University, Chennai, India, pp. 487 - 491, 23-25 April.

مبادئ إدارة المتطلبات وتحليلها لدعم تطوير هندسة البرمجيات: مراجعة الأدبيات.

اشرف عبد المنعم عبد المجيد، عاتكة محمد الطائي و رشا غانم السراج

قسم البرمجيات، كلية علوم الحاسوب والرياضيات، جامعة الموصل، الموصل، العراق

الخلاصة: تعتبر إدارة وتحليل المتطلبات لمشاريع هندسة البرمجيات ذات أهمية كبيرة في تطوير أنظمة البرمجيات. حيث تركز هذه المقالة على دراسة الأدبيات التي تسلط الضوء على أهمية مبادئ إدارة وتحليل المتطلبات والأدوات المساعدة في دعم تطوير هندسة البرمجيات. تتضمن عملية إدارة المتطلبات مجموعة من العمليات المهمة التي تركز على تحديد وتوثيق وتحليل ومراجعة وإدارة متطلبات نظام البرمجيات أثناء مراحل بنائه أو تطويره. تشكل هذه المتطلبات حجر الزاوية لنجاح أي مشروع برمجي لشركات البرمجيات، حيث تؤدي الإدارة الفعالة والصحيحة إلى تطوير منتجات تلبي احتياجات وتوقعات المستخدمين. تتضمن هذه العملية تحليل المتطلبات، والذي يحول متطلبات المستخدم أو سوق العمل إلى مواصفات فنية قابلة للتطبيق في المراحل اللاحقة من عملية بناء النظام في هندسة البرمجيات. تركز هذه العملية على دراسة وتحليل المتطلبات بدقة لإزالة أي تعارض أو غموض، والتأكد من صحتها ومدى تنفيذها. بشكل عام، تستعرض الورقة هذه الأساليب المختلفة المستخدمة في إدارة وتحليل المتطلبات من قبل فريق العمل، بما في ذلك تقنيات جمع المتطلبات، وطرق التحقق من المتطلبات، وإدارة التغييرات أثناء مرحلة جمع المتطلبات. كما يتناول التقرير التحديات التي تواجه إدارة المتطلبات في مشاريع البرمجيات وكيفية التعامل معها لضمان نجاح المشروع.

الكلمات المفتاحية:

إدارة المتطلبات، تطوير البرمجيات، هندسة البرمجيات إدارة المشاريع، نجاح مشاريع تطوير البرمجيات، جمع المتطلبات، تحليل هندسة المتطلبات.