A Requirement Gathering Framework for Electronic Document Management Systems

Omdeep Gokhool
Software and Information Systems Department
University of Mauritius
Reduit, Mauritius
omgokhool@gmail.com

Soulakshmee Devi Nagowah
Software and Information Systems Department
University of Mauritius
Reduit, Mauritius
s.ghurbhurrun@uom.ac.mu

***Abstract*- The practice of requirement engineering has been used for several years in software development. The process of requirements engineering can be broken down into four stages namely elicitation, analysis and negotiation, specification, and validation. Because requirements engineering varies from organization to organization, it may be difficult for project managers to adopt the correct usage of the various stages of a project. Moving towards digital transformation, many organizations are moving towards electronic document management system (EDMS). EDMS has become a substantial tool in many organizations, when it comes to integrated data gathering and sharing. Nevertheless, requirement engineering for implementing EDMS projects should be carefully done to avoid project failure or poor-quality software. This paper thus presents a framework for requirement gathering for EDMS to promote successful implementation using a hybrid approach. The framework has been evaluated in a well-known company in Mauritius and results have been presented in the paper.**

***Keywords*- Requirement Engineering, Requirement Gathering, Electronic Document Management System, Agile Framework**

I. Introduction

EDMS can be described as digital filling cabinets that provide a platform to arrange, organize and store digital documents [1]. EDMS has been introduced as a solution to reduce paper works in organizations. As a result, EDMS improves organizational communication and productivity by facilitating the processing of documents and the flow of information [1]. However, Ismael and Okumus [2] argued that the proper installation of an EDMS begins with a thorough understanding of the organization's structure and specialized requirements. Similarly, Abdulkadhim et al. [3] highlighted that one of the most important factors to consider while implementing an EDMS is the user needs/ requirements. A faulty requirements process can lead to software project failure [4].

A requirement is a distinct desirable characteristic of a software. According to Sonbol et al. [5], a requirement can be defined as a comprehensive conceptualization of an objective, a condition, or a function. All software projects, including document management systems, include requirements as one of the most important components. Hence understanding and depicting their meaning, plays a crucial role towards the success of such projects [6]. The process of obtaining user needs and producing software specifications is known as requirement engineering [7]. Mohabuth [8] highlighted that requirement engineering assesses if the system aligns with business objectives (feasible study), uncovers requirements (elicitation and analysis), translates them into conventional format (specification), and verifies that the requirements define the system as per user needs (validation).

Waterfall or agile frameworks are commonly used as requirement engineering frameworks for implementing document management systems. Chen et al. [9] reported several challenges and issues related to the adoption of these frameworks for requirements gathering. Furthermore, it was observed by Zamudio et al. [10] that formal procedures are not being applied despite the various methods for obtaining user requirements and understanding user needs throughout requirement engineering. Companies are relying on common sense and team experience instead. Given the dynamic nature of organizations and fast shifts in demand, these methodologies have shown to deliver low-quality software.

The rest of the paper is structured as follows: Section II describes the related works on different methods for requirements gathering. Section III describes the proposed framework for requirements gathering for implementing EDMS projects. Section IV consists of the evaluation of the proposed framework. Results and discussion are presented in section V. Finally, section VI concludes the paper and presents future works.

II. Related Works

This section presents related works done in the field of requirements gathering. Kautz [11] evaluated the function of users and customers in agile software development by conducting a case study of a large agile development project. Participatory design methods were employed by Kautz [11] in XP. The agile team identified issues related to misunderstanding of requirements early by using an on-site customer and iteratively reviewing them with users and customers.

Rivero et al. [12] proposed the use of Mockup-Driven Development (MockupDD). They adopted a scrum-integrated strategy, which supports model-driven web engineering (MDWE). MockupDD begins with a brief requirement gathering phase that culminates with a set of user stories. Customers and users created mock-ups to illustrate these user stories. These prototypes serve as a basis for the modelling process that follows.

Olsson et al. [13] built a process model based on the conceptual paradigm of qualitative/quantitative customer-driven development in the early stages of development. The model highlights the significance of combining qualitative consumer feedback with quantitative data. The authors consider needs as hypotheses that are tested with consumers prior to development. Hypotheses were influenced by business strategy, innovation activities, customer input, and continuing validation cycles.

978-1-6654-9291-1/22/$31.00 ©2022 IEEE

Schön et al. [14] conducted a comprehensive review of the available literature. The review focused on the current status of the literature on agile requirements engineering, with an emphasis on stakeholder and user participation. There were 27 relevant papers found after a thorough quality assessment of the research considered. Stakeholder and user participation, data collection, user perspective, integrated approaches, shared knowledge, artefacts, documentation, and non- functional requirements (NFR) were among the agile requirements engineering components that the authors thoroughly investigated.

Heck and Zaidman [15] conducted a comprehensive review of the literature. The purpose of their study was to find out if there are any quality standards for evaluating the correctness of written agile requirements, as requirements quality is often regarded to be a critical aspect in the quality of the final product. For the purpose of creating agile requirements, this study looked at 16 previous studies and identified 28 different quality criteria. Using common requirements engineering criteria, a comparison was made following the analysis and categorization. Based on their findings, the authors also made recommendations for practitioners in the field of agile requirements measurement.

In a study by Chen et al. [9], empirical data were collected through survey results to identify the limitations with existing requirements gathering frameworks. The frameworks consisted of the waterfall model, agile methodologies and hybrid approaches which combined both agile methods and waterfall model. A number of issues were reported such as inconsistent and fragmented business and application expertise. Another issue relates to lack of coordination among all key corporate entities. Furthermore, difficulties in observing the long-term effects of changes, in engaging stakeholders, in identifying important stakeholders and business units, in defining end-to-end use cases and in securing final approval among others, were highlighted.

III. Proposed Framework

This section describes a framework to facilitate requirement gathering for EDMS in an attempt to overcome the limitations identified in existing frameworks described in Section II. The enhanced framework makes use of scrum, Joint Requirements Development and mind mapping. A model-driven method is used to design the framework by mixing agile and non-agile practices. Different hybrid approaches are commonly used in the software industry, for example, a hybrid of agile and conventional software systems. Pre-gathering, mid-gathering, and post-gathering are the three phases of the proposed framework. The following paragraphs briefly discuss the most significant components of the proposed framework:

- *Scrum Agile framework*

Scrum is a project management model that focuses on teamwork, responsibility, and incremental progress to a well-defined objective. The framework begins with a straightforward principle: start with what is visible or understandable. After that, the progress is tracked and adjustments are made as required. Scrum is part of agile and this methodology promotes teamwork within software development firms. The scrum development approach uses the product backlog to keep track of requirements for a project. The product backlog serves as a repository for project needs. Prioritized requirements (in the form of user stories) are requirements sent to the product backlog. The different ceremonies involved in scrum are sprint planning, daily stand-up, sprint review and sprint retrospective.

- *Mind maps*

Mind mapping is a technique that is regarded as the greatest model for requirement gathering by practitioners and researchers. It is a cognitive method that teaches people how to collaborate with others in their own way of thinking. Likewise, according to Mahmud and Veneziano [16], mind mapping technique has been considered as an appropriate technique for collecting and representing requirements within the scrum model. The authors have investigated on how mind maps could contribute to the formation of an acceptable product backlog, which serves as an initial SRS document in scrum. Mind mapping is also common psychological procedure, which comprises the graphical representation of cases in written means. Planned concepts, keywords, actions, or a unit in application development phrases are shown as a supplemental tool for requirements gathering.

- *Joint Requirements Development (JRD) & Joint Application Development (JAD)*

Collaborative meetings with customers and end users are useful for swiftly establishing requirements. Improved session efficacy and early prototype acquisition can be achieved by using software strategies in conjunction with JAD and JRD sessions. Despite these benefits, the sessions are the most difficult to carry out properly. The most important factor in successfully conducting these sessions rely on the performance of the team chosen to engage in the sessions. Cross-functional impacts can also be estimated using JRD processes in a controlled environment. The stakeholders work together to deal with the cross- functional effects. To document the discussion, business analysts, requirement engineers and facilitators should be present.

In the proposed framework shown in Fig. 1, two methods (agile and non-agile) are incorporated to create a unified concept of JRD, concept mapping (mind mapping), and scrum. The use of closed interview sessions in the JRD phase is emphasized to prioritize requirements along with subsequent usage of mind mapping to gather all essential specifications before delivering final specifications (user stories) to the product backlog (as a base process). The second phase generates output data in form of user stories that are adapted for the framework. As shown in Fig. 1, the entire process consists of three components, each of which is further sub divided into subsections. The distinct operations that occur within each phase as well as the flow of data between processes are illustrated.

A. JRD-based pre-requirement gathering phase

Pre-requirement collecting (also known as "no-defined requirements phase") is the initial step in the software development lifecycle, and JRD methodologies are employed during this stage. A business analyst oversees the entire process of gathering needs from executives, clients, partners, and end users. Requirements documentation begins with this

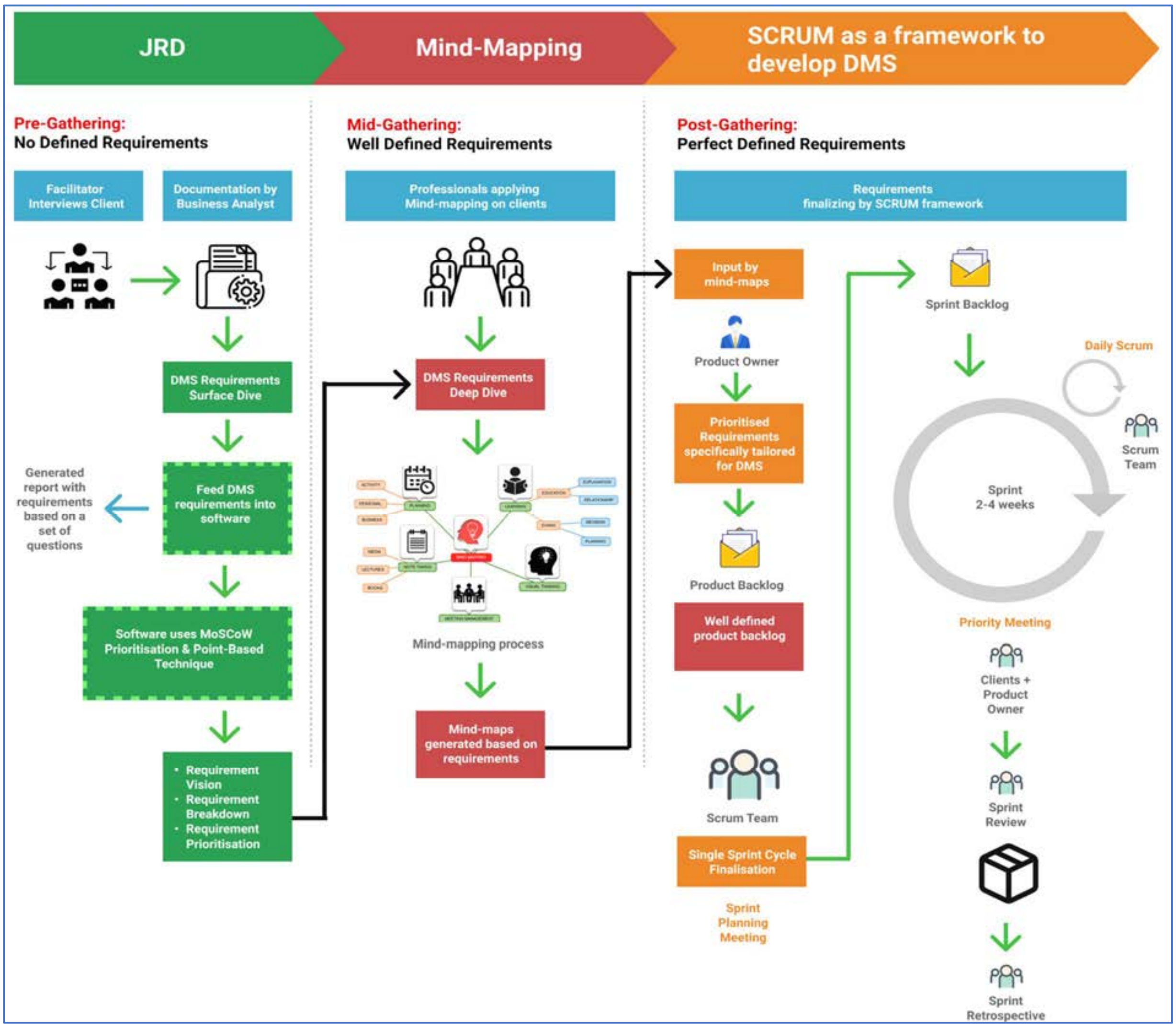
JRD
Mind-Mapping
SCRUM as a framework to develop DMS
Pre-Gathering:
No Defined Requirements
Facilitator Interviews Client
Documentation by Business Analyst
DMS Requirements Surface Dive
Feed DMS requirements into software
Generated report with requirements based on a set of questions
Software uses MoSCoW Prioritisation & Point-Based Technique
• Requirement Vision
• Requirement Breakdown
• Requirement Prioritisation
Mid-Gathering:
Well Defined Requirements
Professionals applying Mind-mapping on clients
DMS Requirements Deep Dive
Mind-mapping process
Mind-maps generated based on requirements
Post-Gathering:
Perfect Defined Requirements
Requirements finalizing by SCRUM framework
Input by mind-maps
Product Owner
Prioritised Requirements specifically tailored for DMS
Product Backlog
Well defined product backlog
Scrum Team
Single Sprint Cycle Finalisation
Sprint Planning Meeting
Sprint Backlog
Daily Scrum
Scrum Team
Sprint 2-4 weeks
Priority Meeting
Clients + Product Owner
Sprint Review
Sprint Retrospective

Fig. 1. Proposed Framework

essential initial step. The interactions between the facilitator and the final client are documented by a business consultant. A great benefit that JRD provides to the current project's development team is a reduction in documentation issues from the beginning of the requirements gathering phase.

In the JRD meetings, specific questions to EDMS are asked to the client such as: *What is the nature of business for the EDMS being built for?*; *Is the data in the company in digital format or any other format?*. The same data are fed into the software, which will keep track of all the requirements, the priority, the story points and shall generate a report, which will then be used in the second phase. During JRD sessions, a requirements engineer or business analyst will document the conversations that are specifically focused to gather needs from the client, and this discussion will take place in an environment that is controlled before employing mind mapping. The fundamental concept is to include mind mapping strategies into the JRD process. After the JRD meetings, mind mapping is employed and scrum is used at the conclusion of the process.

B. *Mind mapping – Mid-requirement gathering phase*

The use of the mind mapping approach demonstrates that cooperative requirements development sessions produce positive results and assist in better indulgence in solving the challenge. Furthermore, mind mapping methodology facilitates in the segmentation of functional requirements into sub-functional specifications during the second phase.

The goal of the "mid- requirements gathering phase" is to obtain a detailed list of user requirements. As part of the framework's second step, the user and the requirement engineer (professional) using the mind map will collaborate to brainstorm solutions based on the basic needs that were gathered during the JRD phase. Mind mapping begins with the user's most significant requirement, and the research engineer selects the rest of the user's requirements starting from the top and continuing into sub-details as much as possible to capture all the aspects of the user's requirements conceivable to offer at the conclusion. All user stories are extensively reviewed and documented in mind map style to ensure that any probable requirement is not missed in this phase.

C. *Scrum framework – Post –requirement gathering phase*

Scrum receives its first input for iteration and sprint production after obtaining all possible sub branches of mind maps for each user requirement. The "post-requirement gathering phase" also known as the scrum product backlog, is where the completely described user requirements are ready to be supplied for a sprint. The software sprint backlog is the third phase of the process. The trimmed chart can be seen in the sprint backlog. For each scheduled activity, this simplified chart shows the responsibilities, and exact time periods. During this phase, a sprint group meeting is organized where all members of the team, as well as the end-users participate in order to reflect on the established requirements stories. Final product backlog work is utilized to kick off a sprint following a meeting when agreed-upon user stories are added. The emphasis of this phase is to prioritize user stories.

IV. Evaluation of Framework

The proposed framework in section III was deployed in operation within a team of Company X (fictitious name), which is a leading IT company based in Mauritius. The name of the company is not revealed because of confidentiality of information. The company provides solutions, which range from developing custom software for ATM to payment solutions. It is also known to implement EDMS for small-medium size companies around the island. Company X was chosen because it produces its EDMS using agile approaches, particularly scrum methodology. The members of the company agreed to use the recommended framework in their ongoing sprints of projects to find solutions to the challenges they were experiencing.

The proposed framework was utilized in investigations where a scrum sprint iteration was completed over a 30-day lapse to assess its validity. Statistical data were acquired following the implementation of the proposed approach. The framework was used to gather requirements for a new DMS project. The following methods were used to collect data on a continuous basis: (1) A series of discussions at Company X helped to define the subject area, (2) Numerous clarifications from the literature followed, and the evolution with real obstacles were discussed in greater detail and (3) A survey was compiled when Company X concerns were properly identified. The survey was given to the Company X main contact person, who then passed it on to other members of the team. The team consisted of project managers, developers, product owners, and members of their support staff.

The DMS requirements were broken down into three groups at "No-defined requirements" phase of requirement gathering: end-user requirements, administrator requirements and system requirements as listed in Table I. This was the first stage in visualizing the three primary categories. An accurate depiction of the breakdown phase could be gained in doing so.

Table I. Requirement Pre-gathering Phase With Three Major Categories (End-user, Administrator, System)

End-user Requirements		**Administrator Requirements**	
REQ1	Manage documents	**REQ1**	Manage storage of users
REQ2	Search within documents	**REQ2**	Manage access of users
REQ3	Create documents	**REQ3**	Manage permissions groups
REQ4	Modify documents	**REQ4**	Manage overall organization files shared publicly
REQ5	Share documents between collaborators	**REQ5**	View system statistics
REQ6	Archive documents	**REQ6**	View audit trail of documents
REQ7	Manage storage	**REQ7**	Manage workflows
REQ8	Version control documents	**REQ8**	Ensure compliance
REQ9	Manage security of documents	**System Requirements**	
REQ10	Manage collaboration	**REQ1**	Perform indexing on documents
REQ11	Manage workflow for collaboration	**REQ2**	Classification
REQ12	Integrate third party apps such as Google Docs	**REQ3**	Perform version control
		REQ4	Capture metadata
		REQ5	Perform update
		REQ6	Run backup

Requirements mid-gathering phase: Mind mapping techniques were used to gather in-depth requirements for each

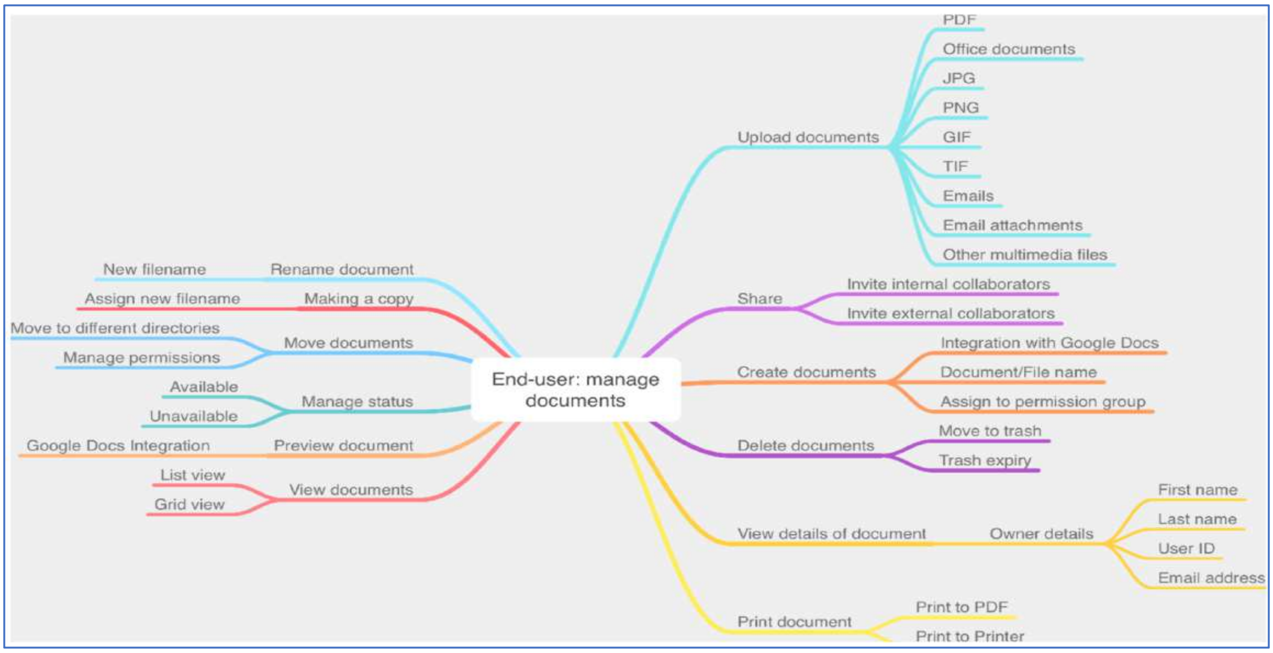

Fig. 2. End-user: Managing document requirements using mind-maps

user's specific needs. Many branches and sub-branches of any specified requirement were located using a cognitive method so that functional or non-functional requirements were not missed. Different mind maps of end-user, administrator and system requirements, with their subdivisions were defined. An example of a mind-map for end-user is shown in Fig. 2.

To bring the requirements gathering process to a conclusion, the user stories and finalized product backlogs created from the detailed requirements captured in deep dive mind maps were fed into the scrum. The sprint backlog was assigned to the scrum process, which includes frequent sprints and a 30- day cycle. Tables II - IV highlight the DMS requirement backlogs that have been completed. End- user requirements, administrator requirements, and system (EDMS) needs are all included in the requirement categories.

TABLE II. PART OF PRODUCT BACKLOG FOR USERS OF EDMS (FIRST FIVE REQUIREMENTS)

ID	Work Item Type	Title	Priority
1	End user backlog item	As an end-user, I want to upload documents (PDFS, Office files, JPG, PNG, GIF, TIF, Emails, multimedia files)	1
2	End user backlog item	As an end-user, I want to share documents with other collaborators	2
3	End user backlog item	As an end-user, I want to create documents	1
4	End user backlog item	As an end-user, I want to integrate Google Docs	3
5	End user backlog item	As an end-user, I want to view the details of the uploaded files	3

TABLE III. PART OF PRODUCT BACKLOG FOR ADMINISTRATORS OF EDMS (FIRST FIVE REQUIREMENTS)

ID	Work Item Type	Title	Priority
1	Administrator Product backlog item	As an administrator, I want to create permission groups	1
2	Administrator Product backlog item	As an administrator, I want to view permission groups	1
3	Administrator Product backlog item	As an administrator, I want to assign users to permission groups	1
4	Administrator Product backlog item	As an administrator, I want to edit group permissions	1
5	Administrator Product backlog item	As an administrator, I want to delete group permissions	1

TABLE IV. PART OF PRODUCT BACKLOG FOR SYSTEM (FIRST FIVE REQUIREMENTS)

ID	Work Item Type	Title	Priority
1	System Product backlog item	Check files if can be indexed and flag if cannot be indexed	1
2	System Product backlog item	Manage manual indexing by enabling/disabling	1
3	System Product backlog item	Automate indexing and check if system is not overutilizing system resource	3
4	System Product backlog item	Update database based on collected metadata such as keywords, sentences & tags	2
5	System Product backlog item	Perform OCR on files	1

V. RESULTS AND DISCUSSION

The results gathered following surveys and interviews are explained and analysed in this section based on the data collected further to the adoption of the proposed framework in section IV in Company X. The efficiency and effectiveness of the framework and the different phases within the framework were tested following numerous EDMS projects within a 30-day lapse. A questionnaire via Google Forms was sent to the project teams working in Company X. 28% of replies came from requirements engineers or business analysts, 13% from testers, 14% from scrum masters, 34% percent from developers, 7% from project managers, and 4% from product owners. The participants were asked to respond to a predefined set of questions and responses as summarised in Table V. The options for answers were:

(A) This was not an issue in our project
(B) The issue was satisfactorily resolved and did not cause problems with the project delivered
(C) The issue was normally discussed, but it occasionally caused issues with the project at hand
(D) The issue was rarely addressed, but it frequently resulted in problems with the given project
(E) The matter was not discussed, which resulted in major problems with the completed project

Results from Table V show that responses have changed following the adoption of the proposed framework. Prior to adoption of the framework,

- 70.6 % responded that lacking and clashing requirements (questions 1 and 2) were major concerns that were not discussed, which resulted in major problems with the completed project
- 70.6 % responded that questions 5, 7-10 were issues that were rarely addressed but they frequently resulted in problems with the given project.

When the proposed framework was adopted,

- most issues did not arise or were satisfactorily resolved and did not cause problems with the project delivered
- with respect to questions 8, 9 and 10, 5.9%, 5.8% and 6.8% reported that the issues were usually addressed, but they occasionally resulted in complications with the given project respectively.

The findings of the study show that with the introduction of this hybrid framework, both the requirement engineering and development teams experienced a significant reduction in requirement gathering challenges. The surveys carried out quantify that the proposed framework is indeed effective and efficient. The framework was tested for a period of 30 days within Company X and was used with different clients to test the efficiency. The proposed hybrid framework proved to solve the issues faced using conventional requirement gathering processes. Apart from collecting data from questionnaires, data were collected through focus groups with the team that adopted the framework for testing. The result of the discussion showed that this framework can be used not only for EDMS purposes but for global software development requirement engineering purposed. The questions asked to the client by the facilitator during the pre-requirement gathering phase will only differ.

TABLE V. RESPONSES W.R.T SURVEY QUESTIONS

No.	Question	Response (%) prior to adoption of framework					Response (%) following adoption of framework				
		A	B	C	D	E	A	B	C	D	E
1	Were lacking requirements a concern in your project?			17.6	11.8	70.6	99.5	0.5			
2	Were clashing requirements a challenge in your project?			5.9	23.5	70.6	94.1	5.9			
3	Was a lack of measurability or testability of user requirements identification a concern in your project?			11.8	29.4	58.8	82.4	17.6			
4	Were changing requirements a problem in your project?			6.2	37.5	56.3	94.1	5.9			
5	Did over-specified requirements cause problems in your project?				70.6	29.4	88.2	11.8			
6	Were project scope issues of concern in your agile project requirements gathering?			6.7	46.6	46.6	52.9	47.1			
7	Was the issue of early usability forecasting of requirements a concern in your project?			11.9	70.6	17.5	55.5	44.5			
8	Was there the difficulty of comprehending a user a problem during requirements gathering in your project?			23.5	70.6	5.9	52.9	41.2	5.9		
9	Was the issue of requirements prioritizing a concern in your project's requirements gathering?			5.9	70.6	23.5	82.4	11.8	5.8		
10	Was the issue of insufficient requirements a concern in your project's requirement gathering?			23.5	70.6	5.9	82.4	10.8	6.8		

VI. CONCLUSION AND FUTURE WORKS

Different methods have been used to gather requirements during the requirement engineering process. However, none of the strategies can be categorized as either effective or ineffective because they are applicable in a variety of scenarios. There is thus room for improvement. This paper presents a framework for requirements gathering using a hybrid approach namely the scrum framework with mind mapping and JRD. The framework was tested for EDMS projects but can surely be adapted for other types of projects. The framework was evaluated in a well-known company in Mauritius and has reported to be efficient based on surveys carried out. As future works, AI can be used for automatic mind mapping in the deep dive requirement phase. Furthermore, other frameworks such as Kanban can be adopted in the post-requirement gathering phase. Moreover, Key Performance Indicators (KPIs) can be integrated to measure the framework effectiveness at a granular level.

REFERENCES

[1] R. Andriansyah and F. Elmi, “Analysis of the Effect of Electronic Document Management System”, Organizational Commitment and Work Satisfaction on Employee Performance PT. Graha Fortuna Purnama, International Journal of Innovative Science and Research Technology, vol. 5(8), pp.944-952 , 2020.

[2] A. Ismael and I. Okumus, “Design and Implementation of an Electronic Document Management System”, Mehmet Akif Ersoy Universitesi Uygulamalı Bilimler Dergisi, vol. 1(1), pp.9-17, 2017.

[3] H. Abdulkadhim, M. Bahari, A. Bakri and H. Hashim, “Exploring the common factors influencing Electronic Document Management Systems (EDMS) implementation in Government”, ARPN Journal of Engineering and Applied Sciences, vol.10(23), pp.17945-1795, 2015.

[4] A. Hussain and E.O. Mkpojiogu, “Requirements: Towards an understanding on why software projects fail”, In AIP Conference Proceedings, vol. 1761(1), p.020046, AIP Publishing LLC, 2016.

[5] R. Sonbol, G. Rebdawi and N. Ghneim, “Towards a Semantic Representation for Functional Software Requirements”, In 2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), pp. 1-8, IEEE, 2020.

[6] F. Nazir, W.H. Butt, M.W Anwar and M.A.K Khattak, “The applications of natural language processing (NLP) for software requirement engineering-a systematic literature review”, In International conference on information science and applications, pp. 485-493, Springer, Singapore, 2017.

[7] H. M. Chen, R. Kazmanand and S. Haziyev, “Agile big data analytics development: An architecture-centric approach”, In 2016 49th Hawaii International Conference on System Sciences (HICSS), pp. 5378-5387, IEEE, 2016.

[8] A. Q. Mohabuth, “Software Practitioners Challenges in the Requirement Engineering Phase of Software Development”, 15th Int'l Conference on Advances in Science, Engineering & Natural Resources (PSENR-19), 2019.

[9] F. Chen, N. Power, J.J. Collins and F. Ishikawa, “Contemporary requirements challenges and issues: an empirical study in 11 organizations”, In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 1592-1599, 2019.

[10] L. Zamudio, J.A Aguilar, C. Tripp and S. Misra, “A requirement engineering techniques review in agile software development methods”, In International Conference on Computational Science and Its Applications, pp. 683-698, Springer, Cham, 2017.

[11] K. Kautz, “Participatory design activities and agile software development”, In IFIP Working Conference on Human Benefit through the Diffusion of Information Systems Design Science Research, pp. 303-316, Springer, Berlin, Heidelberg, 2010.

[12] J. M. Rivero, J. Grigera, G. Rossi, E. Robles Luna, F. Montero and M. Gaedke, “Mockup-Driven Development: Providing Agile Support for Model-Driven Web Engineering”, Information and Software Technology, 56, 670-687, 2014. doi:10.1016/j.infsof.2014.01.011

[13] H. H. Olsson and J. Bosch, “Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation”, LNBIP 210, International Conference of Software Business, Braga, pp. 154-166, 2015. doi: 10.1007/978-3-319-19593-3_13

[14] E. M.Schön, J. Thomaschewski and M.J. Escalona, “Agile Requirements Engineering: A Systematic Literature Review”, Comput. Stand. Interfaces, 2016. doi:10.1016/j.csi.2016.08.011

[15] P. Heck and A. Zaidman, A systematic literature review on quality criteria for agile requirements specifications, Software Quality Journal, Springer US. 2016. doi:10.1007/s11219- 016- 9336-4

[16] I. Mahmudand and V. Veneziano, “Mind-mapping: An effective technique to facilitate requirements engineering in agile software development”, In 14th International Conference on Computer and Information Technology (ICCIT 2011), Dhaka, IEEE, pp. 157-162, (2021). doi:10.1109/ICCITechn.2011.6164775