Elinvo (Electronics, Informatics, and Vocational Education)
Vol 9, No. 2, 2024, pp. 253-264
ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

A Survey on Software Requirements Engineering in Information Technology Institutions

Muhammad Tulus Akbar[1*], Maria Ulfah Siregar[1]

[1]Universitas Islam Negeri Sunan Kalijaga, Yogyakarta, Indonesia

Article Info

Article history:
Received January 19, 2024
Revised November 27, 2024
Accepted Oktober 23, 2024

Keywords:
Qualitative approach; requirements elicitation; software development; specification; validation

Abstract

Requirements engineering is a crucial phase in software development. In theory, requirements gathering should follow structured and systematic steps to ensure all requirements are complete, consistent, and clear. However, there is a lack of research in real-world contexts, particularly in small and medium-sized software development organizations, which often face challenges related to human resources, infrastructure, and financial constraints. This study aims to explore the practices of requirements engineering in software development agencies. A qualitative approach was used, employing a narrative (descriptive) research design. Data were gathered through interviews, observations, and documentation from six institutions involved in developing various types of software. The data were then analyzed using Yin's Five Cycle content analysis method, supported by Nvivo 12 software. The findings reveal that the time required for gathering requirements varies between institutions (four institutions take 1 week to 1 month, two take 1 to 3 months, and two take over 3 months). Regarding the analysts' backgrounds, all institutions consider requirements elicitation a critical stage in software development, leading them to hire analysts with IT educational backgrounds, most of whom possess significant work experience. This is further evidenced by the fact that all institutions consider staff experience when forming requirements analysis teams and assigning senior staff to specific roles. There is limited practical variation in the sources of requirements and elicitation techniques. All institutions rely on objectives, domain knowledge, stakeholders, and the operational environment as sources of requirements. As for elicitation techniques, they primarily use interviews, facilitated meetings, and document reviews.

Corresponding Author:
Email: 22206051003@student.uin-suka.ac.id

INTRODUCTION

The increasing needs of the industrial market in software development require companies that produce software to provide functions and features for owners, managers, and end-users to benefit from the software built [1]. Software development institutions must be able to collect system requirements as a whole to carry out the development objectives [2].

The activity of building software needs to go through the requirements elicitation stage by applying a socio-technical approach [3]. They assisted in mediating between developer users and managers to meet all needs [4], [5]. Constructive dialogue with stakeholders while collecting system requirements is the key to success in system development [6]. This is important because it will get precise information about the system and its limitations. The activity of detailing system requirements is found in the requirements specification phase; all needs are documented in the form of natural language, which can be described in the form of narratives and concept maps as an element of

ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

decomposition, verification, work contracts, costs, scheduling, sharing knowledge and identifying and reducing errors or mistakes in system requirements [7], [8]. Essential parts of the requirements specification summarize functional and non-functional requirements, data resilience, system security, connection, and service availability [2], [3]. In practice, knowledge of requirements engineering involves intuition and inspiration from understanding stakeholders when seeking solutions to system needs. The requirements and characteristics are related to the individual's analytical ability, reflection, experience, and creativity. These characteristics can manifest in ideas, solutions, innovations, and interface designs [9], [10].

This study uses a case study approach focusing on software development companies and universities. Data were collected through interviews and observations, as well as by documenting institutional archives. A similar study examined the understanding of User Experience (UX) in Indonesia's technology industry [11]. Using a descriptive qualitative approach and the Nvivo software as a data processing tool to present the qualitative research findings [12].

The process of collecting requirements for the needs of the system must be carried out by a professional who specializes in in-depth needs analysis in the field of requirements engineering because the critical phase in designing a system is in the software requirements process when the needs are loaded, to ensure that all needs are sufficient and have been met [5], [6], [13]. Ensuring that the statement of requirements has been fulfilled, then validation and verification are carried out in the requirements validation phase to ensure that the system requirements are guaranteed and have high-quality [14]. In the practice of engineering system requirements, software engineering engineers face many problems. It is often found that the occurrence of errors in the requirements engineering process causes the needs to be inaccurate requirements and results in ambiguity in system requirements [15]. This process may occur because some requirements engineers assume that stakeholders are the sole source of requirements in engineering system requirements [9]. Stakeholders as a source of requirements can trigger this problem released by Macaulay; it is mentioned that many stakeholders need more knowledge about the technology perspective coupled with improper and incomplete documentation, so many requirements end up disappointed [16], [17]. Other problems can arise from too many requirements, which are prone to errors and can be time-consuming and costly [18].

To solve the above problems, it is crucial to make changes in system requirements engineering with a more effective and efficient model by analyzing system requirements documents, designing UI (user interface) forms, or building prototypes that can easily communicate with stakeholders. Implementing structured requirements will include the ability to control projects that can be adequately managed, measure the performance of the results of the requirements, make improvements if errors or damage occur, and reduce productivity costs during the requirements collection cycle. Documenting requirements to limit the need to be unambiguous and costly to produce. The validation and verification process ensures that all requirements have been met, are in line with the needs, and is the gateway into the system development process [4], [7], [19], [9], [20], [21], [22].

METHODS

This research aims to identify the technical processes and resources of requirements engineering in a real-world context, specifically in micro, small, and medium-sized software development institutions. Our research step is in Figure 1. This research uses a qualitative approach. This research is designed using narrative research (descriptive). Data is collected from interviews, observations, and documents. Data is obtained from information whose daily task is to collect information related to software requirements [11], [23]. data sources are described in the following section.

Interviews

Interviews are intended to obtain information from informants. The interview process was conducted formally. In this study, the interviews aimed to get the broadest possible information about

the practice of user requirements engineering in institutions [11]. Interviewees are allowed to answer interview questions and provide examples if needed [24].

Observations

After a series of interview activities were completed, the next stage was to conduct observations to test the accuracy of the data collected during the interviews by trying directly at the locations of the six institutions [25]. The observation process was carried out to collect additional data to correct and revise the inaccurate data collected during the interviews.

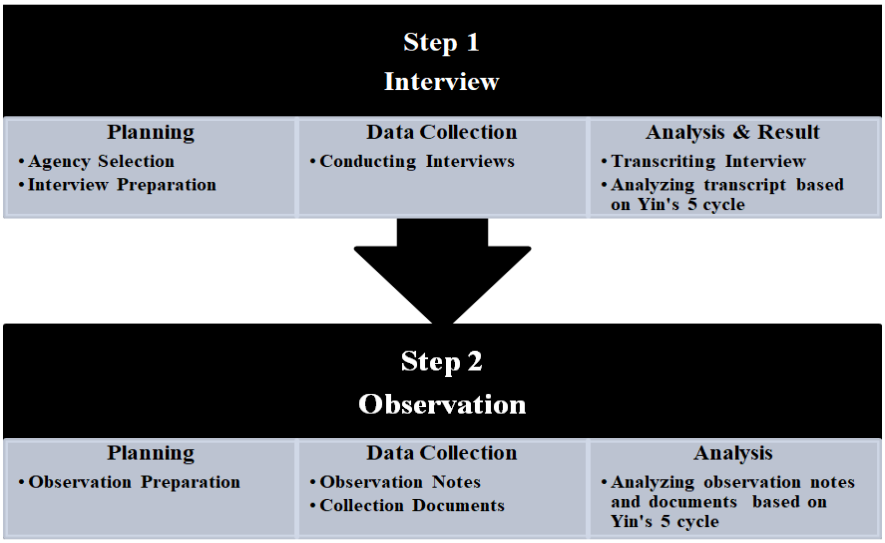

Figure 1. The employed research approach utilized qualitative methods

Data was also collected from the organization's documents. Searches were conducted through the organization's official website and social media platforms [26]. The website provided formal resources such as reports, policies, and press releases, while social media offered insights into public engagement, customer feedback, and real-time updates. To analyze the data, we employed Yin's five-stage analytical framework [11]. These stages included compiling the database, unpacking the data, reorganizing the data, interpreting the data, and drawing conclusions. The process is visually summarized in Figure 2, which outlines each stage in detail.

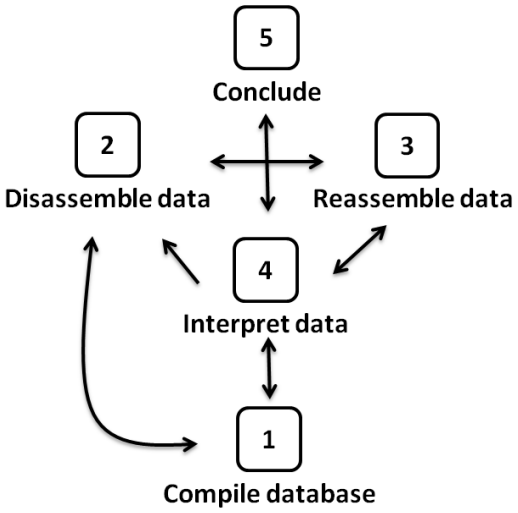

Figure 2. Yin's Five Phases of Analysis

Initially, we collected all the information from the interviews and searched for previous research sources that were related. This was then organized using Nvivo 12 software for further processing. Second, disassembling the data. After collecting all the information and putting everything in Nvivo, the loaded data in the form of audio, video, and various types of text then was sorted out. The first

stage is to determine the parent code as a variable so that the data can be fragmented again into child codes to get the relationship between variables and sub-variables, for example, the relationship between institutions and the software being developed. Third, reassembling the data. After the data was made in fragments, we reorganised it based on parent and child codes. They are compiled with maps tools available in Nvivo to see the relationship between parent codes, child codes, and cases. Cases themselves represent and contain descriptive information on the relationship between parent codes and child codes with field conditions, for example, the relationship between requirements analysts and educational backgrounds. Fourth. interpreting the data. During this phase, we used the maps feature in Nvivo. This feature serves to read the data in the form of a framework that is easy to read, understand and interpret. Fifth, concluding the data. All data that has been collected, processed, analyzed, and interpreted from the previous four phases were then summarized and concluded.

RESULT AND DISCUSSION

The data for this research was obtained from institutions selected based on the following criteria: (1) Information Technology companies that provide application and software development services or Information Technology units of higher education institutions [5]; (2) The institution has been operating for over two years. Based on the selection, there are six institutions, as given in Table 1.

Table 1. Institutions Background

Name of institution	Institution Type	Year of establishment
Institution 1	Unit IT Universitas	2020
Institution 2	Unit IT Universitas	2019
Institution 3	Android Developer	2012
Institution 4	Android Developer	2013
Institution 5	Software House	2021
Institution 6	Software House	2010

The selection of institutions was based on study requirements. Institution 1, established in Jambi in 1992, became a university in 2020. Its software development, tailored to campus needs, involves a team comprising a System Analyst, UI/UX Designer, Frontend, and Backend Developers [1]. Institution 2, also in Jambi and founded in 1993, attained university status in 2019. Its software focuses on improving organizational services with a team consisting of a System Analyst, Frontend, and Backend Developers [11]. Institution 3, founded in 2012 in Jambi, collaborates with government and private sectors, employing a Project Manager, System Analyst, and Mobile Programmer (Andhika et al., 2021) [13]. Institution 4, established in 2013 in Jambi, has a similar partnership model and a development team including a Project Manager, System Analyst, Mobile Programmer, Frontend, and Backend Developers [20]. Institution 5, based in Yogyakarta and founded in 2021, partners with government and private sectors, featuring a team of a System Analyst, Project Manager, UI/UX Designer, Quality Assurance, Frontend, and Backend Developers [21]. Institution 6, established in 2010 in Yogyakarta, partners with government, private sectors, and MSMEs. Its team includes a Project Manager, Quality Assurance, UI/UX Designer, Mobile Programmer, Frontend, and Backend Developers [11].

Table 2. Informants Profile

Informants	Job Description	Educational Background	Working Experience
1	Programmer	Master of Computer Science	9 Years
2	Programmer	Bachelor of Computer Science	8 Years
3	Software Developer	Master of Computer Science	10 Years
4	Programmer	Master of Computer Science	3 Years
5	Requirements Analyst	General	2 Years
6	Programmer	Master of Computer Science	5 Years

ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

Interviews were conducted with six resource persons representing their respective institutions. Table 2 presents background information on needs analysts at each institution. The interviews were conducted with six informants from different institutions, all with backgrounds in needs analysis, using the "swim-along" method for comprehensive information gathering [11]. The interviews were recorded in audio format for transcription and analysis purposes [27]. The results were classified based on similar answers to simplify interpretation and facilitate understanding.

Institutions' Background

First, we asked the interviewees about their institutional background. Institutional background in requirements engineering for their systems. An institution's background in requirements engineering influences the processes and techniques used for software development. Based on the interviews, we classified their answers regarding the time to analyze requirements, the software they developed, the type of software developed, and the costs incurred for requirements analysis. Table 3 presents information about the background in requirements engineering at each institution.

Table 3. Institution System Background Development

Variable	Description	Institutions	Percentage
Time Needed for Requirements Analysis	1 Week – 1 Month	1,4,5,6	67%
	1 -3 Month	3,4	33%
	3 Month - ∞	2,4	33%
Type of Software Being Developed	Software for Internal Usage	1,2,3,4,5,6	100%
	Market-based (Product) Software	3,4,5,6	67%
	Made-to-Order (Project) Software	3,4,5,6	67%
Platform Type	Web Apps	1,2,5,6	67%
	Mobile Apps	3,4	33%
Costs Incurred	Monthly Wage	1,2	33%
	Project-Based Wage	3,4,5,6	67%

The time required for needs analysis varies across institutions. Four out of six institutions took 1 week to 1 month, which they deemed sufficient [28]. Institutions 3 and 4 took longer, 1 to 3 months, due to intensive communication with clients for accurate requirements gathering. Institutions 2 and 4 took over 3 months; institution 2 cited annual regulations for gathering requirements, while institution 4 based the timeline on system complexity, categorizing them as small, medium, or large.

Most institutions develop internal software, such as company profile websites, while institutions 1 and 2 also create educational systems like academic software, repositories, and registration pages [11]. Institutions 3 and 4 focus on ticket sales, financial transaction services, and cashier systems integrated with messaging apps [13]. Institution 5 develops financial recording apps for MSMEs, and Institution 6 creates hotel management software [21]. These projects reflect the institutions' strengths, whether in producing market-based or custom-made solutions and involve partnerships with government and private entities [28]. In terms of platform type, institutions 1, 2, 5, and 6 primarily produce web-based applications, while institutions 3 and 4 focus on mobile apps [20].

Background of Requirements Analysts

After knowing the background of the institution, we dug up information regarding the background of the requirements analyst and the ability of the requirements analyst to influence the results of the software being developed. Apart from that, requirements analysts are required to understand requirements in functional and non-functional forms so that the software developed is appropriate and runs according to user expectations.

In this interview, we asked about educational background, experience as an analyst, and experience with software developed by requirements analysts from each institution. Table 4 outlines the background of the requirements analyst.

Table 4. Background of Requirements Analysts

Variable	Description	Institutions	Percentage
Type of Software that the Analysts Ever Get Involved in Production	Software for Internal Usage	1,2	33%
	Market-based (Product) Software	3,4,5,6	67%
	Made-to-Order (Project) Software	1,3,4,5,6	83%
Job Experience	1 – 5 Years	4,5,6	50%
	5 – 10 Years	1,2,3	50%
Educational Background	IT	1,2,3,4,5,6	100%
	Non-IT	5	17%

Field data and interviews reveal that analyst teams from Institutions 1 and 2 focus on internal software development, aligning with their main duties and their ability to manage teams and understand complex needs [11]. Institutions 3, 4, 5, and 6 are involved in product development, leveraging technical expertise and conducting market analysis [28]. This involvement results from collaborations with government and private sectors. Institution 4 noted, “Project-based systems must address detailed needs, as they involve multiple stakeholders and can be vulnerable if not handled carefully" [29].

The success of requirements analysts depends on the number of projects completed and their tenure. Accurate elicitation is crucial; incomplete or unclear requirements may lead to project difficulties or contract cancellations [17]. Field data shows that three analysts have 1-5 years of experience, while the rest have over 5 years. All six analysts have IT backgrounds, with five institutions employing two analysts per system. In Institution 5, the IT analyst handles technical tasks, while a non-IT analyst manages social engineering aspects [21].

Analysis Team

Next, identify the analysis team at each institution. We dug up as much information as possible about teams focused on systems analysis and teams involved in software development. Based on information from six sources, it can be depicted in Table 5.

Table 5. Analysis System

Variable	Description	Institutions	Percentage
Staff Assigned for Analysis Team	Project Manager	3,5,6	50%
	System Analyst	1,2,3,5,6	83%
	UI/UX Designer	5	17%
	Programmer	4	17%
Type of Staff’s Main Task	Project Manager	3,5,6	50%
	System Analyst	1,2,3,5	67%
	UI/UX Designer	1,5,6	50%
	Quality Assurance	5,6	33%
	Programmer	1,2,3,4,5,6	100%

Observations reveal that three institutions assign project managers to conduct needs analysis, leveraging their experience in requirement identification. Most institutions rely on system analysts for this task, as they not only collect requirements but also prepare documentation, conduct market analysis, and manage budgets [1]. At Institution 5, the UI/UX Designer team manages requirements, reflecting a multi-tasking approach typical in developing institutions. In contrast, Institution 4 delegates need analysis by its programming team due to limited resources and the need to optimize staff utilization.

Research indicates that all institutions have programmer teams as their core software development units, supplemented by system analysts who explore system requirements and project

managers overseeing development. Additionally, the UI/UX Designer team handles system interface design, while Institutions 5 and 6 include quality assurance teams to assess system feasibility [14].

Resource Elicitation

At this stage, we describe the sources of need for building software at each institution. Resource elicitation will present data regarding the sources used, the most frequently used, and rarely used by each institution. Based on the results of field data analysis, six sources were used by each institution, which will be explained in Table 6.

Table 6. Requirements Sources

Variable	Description	Institutions	Percentage
Source of Requirements that are Normally Used	Goals	1,2,3,4,5,6	100%
	Domain Knowledge	1,2,3,4,5,6	100%
	Stakeholders	1,2,3,4,5,6	100%
	Business Rules	1,2,3,4,5,6	100%
	The Operational Environment	1,2,3,4,5,6	100%
	The Organization Environment	2,3,4,6	67%
Sources of Requirements that are Frequently Used	Goals	1,2,3,4,5,6	100%
	Domain Knowledge	2,3,4,5,6	83%
	Stakeholders	1,2,3,4,5,6	100%
	Business Rules	1,3,4,5,6	83%
	The Operational Environment	2,3,4	50%
	The Organization Environment	3	17%
Sources of Requirements that are Rarely Used	Goals	4	17%
	Domain Knowledge	1,4	33%
	Business Rules	2	17%
	The Operational Environment	1,5,6	50%
	The Organization Environment	2,4,6	50%

Analysis of resource elicitation data from six institutions shows that goals, domain knowledge, stakeholders, business rules, and the operational environment are all utilized at a rate of 100%. Four institutions leverage environmental organizational resources, reflecting a 67% usage rate among the six. All institutions consistently employ goals and stakeholder resources (100%), while domain knowledge and business rules are used by five institutions (83%). Three institutions frequently access operational environment resources, but overall usage is lower, with only 17% of institution 3 using organizational environment resources.

Operational and organizational environments are infrequently utilized, with a 50% usage rate across three institutions. Additionally, two institutions rarely use domain knowledge resources, resulting in a 33% frequency among the six. Financial resources are utilized by only one institution (17%), alongside goals and business rules, which are also rarely employed [17].

Elicitation Techniques

This discussion will examine how the analyst team gathers requirements sources with existing elicitation techniques. The research results in the data field will be presented in the form of techniques used, often used, and rarely used. The field data processing results can be seen in Table 7.

Field data analysis reveals that all institutions (100%) utilize interview techniques, facilitated meetings, and documents for elicitation. Additionally, prototype, observation, and user stories techniques are employed by five out of six institutions (83%). Scenario techniques are used by four institutions, resulting in a 67% usage rate. The client-hire consultant technique is exclusive to Institution 6, with a usage rate of 17%.

The commonly used elicitation technique is interviews, employed by all institutions (100%). Scenarios and document techniques are frequently used by four institutions (67%). Facilitated meetings and user stories are utilized by three institutions (50%), while observation techniques are

used by two institutions (33%). The prototype technique is infrequently utilized, with only four institutions using it (67%). The methods of facilitated meetings, observation, and user stories have a usage rate of 33%, and one institution rarely employs document techniques (17%) [17].

Table 7. Elicitation Techniques

Variable	Description	Institutions	Percentage
Elicitation Techniques that are Normally Used	Interviews	1,2,3,4,5,6	100%
	Scenarios	2,3,4,6	67%
	Prototypes	1,3,4,5,6	83%
	Facilitated Meetings	1,2,3,4,5,6	100%
	Observation	2,3,4,5,6	83%
	User Stories	1,3,4,5,6	83%
	Document	1,2,3,4,5,6	100%
	Client hires Consultant	6	17%
Elicitation Techniques that are Frequently Used	Interviews	1,2,3,4,5,6	100%
	Scenarios	2,3,4,6	67%
	Prototypes	3	17%
	Facilitated Meetings	1,3,4	50%
	Observation	5,6	33%
	User Stories	3,5,6	50%
	Document	1,2,3,4,6	83%
	Client hires Consultant	6	17%
Elicitation Techniques that are Rarely Used	Prototypes	1,4,5,6	67%
	Facilitated Meetings	2,5	33%
	Observation	2,3	33%
	User Stories	1,4	33%
	Document	5	17%

Requirements Specification

At this stage, we describe the requirements and specifications concerning system analysts in software development at each institution. Excavating requirements specifications will present data related to functional requirements, non-functional requirements, technical limitations in system development, user and system interactions, and system processes in carrying out main functions. The field data processing results are shown in Table 8.

Analysis of requirements specifications indicates that all system analysts consider functional requirements like data integration, feature access, security, and availability, each with a usage rate of 100%. Five analysts (83%) include responsive speed, while Institution 2 excludes this requirement. Automatic updates are considered by 50% (three out of six analysts).

All system analysts (100%) emphasize performance, high-level security, and responsiveness for non-functional requirements. Five analysts (83%) incorporate scalability, while browser compatibility and support are noted by 67% (four analysts). Battery optimization is considered by 50% (three analysts).

Technical limitations identified by all analysts (100%) include data storage, hosting, and server security. Budget limits are set by five analysts (83%), and device and connection technique limitations are acknowledged by 33% (two analysts).

Regarding user and system interactions, all analysts focus on user input and input/output images (100%). Four analysts (67%) cover user interactions and interface systems, while audio input/output is noted by 33% (two analysts). Hardware loading interactions are recognized by only 17% (one analyst).

Lastly, specifications for primary functions include text input, text output, and interaction, all at 100%. Report specifications receive attention from 83% (four analysts) [17].

ISSN 2580-6424 (printed) | ISSN 2477-2399 (online)

Table 8. Elicitation Techniques

Variable	Description	Institutions	Percentage
Functional Requirements	Integrasi data	1,2,3,4,5,6	100%
	Access features	1,2,3,4,5,6	100%
	Automatic updates	2,3,6	50%
	Responsive speed	1,3,4,5,6	83%
	Security	1,2,3,4,5,6	100%
	Availability	1,2,3,4,5,6	100%
Non-functional requirements	Performance	1,2,3,4,5,6	100%
	High level of security	1,2,3,4,5,6	100%
	Documentation	1,2,3,4,5,6	100%
	Battery optimization	3,4,6	50%
	Responsive	1,2,3,4,5,6	100%
	Compatibility	1,2,4,5	67%
	Browser support	1,2,5,6	67%
	Scalability	1,2,3,5,6	83%
Engineering constraints in system development	Data storage	1,2,3,4,5,6	100%
	Budget	1,3,4,5,6	83%
	Device	1,2	33%
	Connection	1,2	33%
	Hosting	1,2,3,4,5,6	100%
	Server security	1,2,3,4,5,6	100%
User and system interaction	User Interface	1,3,5,6	67%
	Input user	1,2,3,4,5,6	100%
	Audio input/output	3,4	33%
	Images input/output	1,2,3,4,5,6	100%
	Hardware	4	17%
User and system interaction	Input text	1,2,3,4,5,6	100%
	Output text	1,2,3,4,5,6	100%
	Report	1,2,4,5,6	83%
	Interaction	1,2,3,4,5,6	100%

Requirements Validation

This discussion will look at validating and verifying system requirements collected and documented for testing. The results of research in the field data will be presented in the form of techniques for the stages of testing suitability checks, actions to be taken if there are errors in the determination test, how to test business reviews, implementation of technical feasibility tests and stakeholders who approve the validation results. Table 9 shows the field data processing results.

Table 9. Requirements Validation

Variable	Description	Institutions	Percentage
Testing suitability checks	Functional	1,2,3,4,5,6	100%
	Performance	1,2,3,4,5,6	100%
	Responsive	1,2,3,4,5,6	100%
	Compatibility	1,2,4,5	67%
Actions to be taken if an error occurs in the determination test	Immediate repair	1,4,5	50%
	Scheduled repairs	3,5,6	50%
	Return	2	17%
How to test business reviews	Employment contract	4	17%
	Letter of assignment	1,2	33%
	Feature	3,5,6	50%
	System complexity	3	17%
Do use technical due diligence	Yes	1,5,6	50%
	Sometimes	3,4	33%
	No	2	17%
The stakeholders involved approve the validation results	Field	1,2,3,4,5,6	100%
	Client	3,4,5,6	67%
	Institutional officials	1,2,3,4,5,6	100%

The analysis of requirements validation reveals that compliance checks, functional checks, performance checks, and responsiveness checks are each conducted by 100% of system analysts. Compatibility checks are performed by 67% (four out of six) analysts.

When errors occur in determination tests, 50% of analysts opt for immediate repairs and scheduled maintenance, while 17% (Institution 2) choose to implement returns. Regarding business review methods, 50% of analysts review features, and two analysts evaluate assignment letters, resulting in a 33% allocation for business reviews. Additionally, 17% of analysts assess work contracts and system complexity during reviews. In terms of technical feasibility tests, 50% of analysts conduct them regularly (three out of six), while 33% do so occasionally. Notably, 17% of analysts from Institution 2 do not perform technical feasibility tests. Finally, stakeholder validation is achieved by 100% of institutional officials, while 67% of clients (four out of six institutions) support the validation results [17].

CONCLUSION

This research found that each institution needed different amounts of time to explore needs (four institutions took 1 week to 1 month, the other two took 1-3 months, while two institutions took more than 3 months). Four institutions that took 1 week - 1 month stated that this was sufficient to identify needs. Two institutions take 1-3 months to gather client needs. They are busy, so it takes a long time to communicate actively. Finally, the two institutions took more than 3 months due to regulations from the parent organization, which regulates the collection of needs, which has a longer duration.

All institutions consider requirements mining a critical phase in software development, so they employ analysts with an educational background in IT, most of whom already have sufficient experience. This also helps the process of gathering requirements because, in software development, someone with an IT background can quickly analyze and break down problems and find the main essence of the needs of a desired application request. Apart from that, it can also save costs because one person can do one job. Person.

All institutions consider staff work experience when assisting needs analysis teams, where they assign senior staff to specific teams. Every institution needs an accelerated process in carrying out analysis, and this aims to make it easier to distribute each burden to the officers responsible for carrying out tasks so that it does not interfere with the manufacturing process in other fields. as can be illustrated in the feature creation process, the analyst team gives tasks to programmers to create predetermined features within a predetermined period.

All institutions use goals, domain knowledge, stakeholder information, business rules, and operational environment as needs sources. Each institution ensures that the series of processes in building an application are guaranteed and remain focused on the needs of the desired application. And ensure that stakeholder needs can be accommodated well and conveniently when using an application. These business rules are written to regulate everything from parties interested in each other so that it is easy to arrange agreements that can run well in the application creation process.

All institutions use interviews, facilitated meetings, and planning documents to obtain requirements. The interview technique is traditional for gathering needs, but this technique is straightforward, fast, and precise in getting core information. They facilitated meeting techniques that can bring together all stakeholders to obtain comprehensive and transparent information so that complete and accurate requirements will be accepted. Documentation techniques are used throughout because they save time and have definite final results.

All institutions document system requirements in design specification documents, thoroughly documenting functional requirements relating to data integration, feature access, security, and availability. Non-functional needs considered comprehensively by all institutions are related to system performance, security, documentation, and responsive systems. Limitations that all institutions consider before the system is built are preparing for data storage, hosting, and server security. The

process of user and system interaction concerning all institutions is related to user input and input/output images. Determine the process of each feature built and implemented by all institutions relating to text input, output, and interaction. The specification process must be detailed to avoid everything running smoothly regarding planned requirements.

All institutions carry out a validation process for the specification documents that have been prepared to ensure that requirements are complete and can be continued to the development stage. Test and check the suitability of the system implemented by all institutions, namely functional, performance, and responsiveness. Suppose there is an error in the assessment test. In that case, each institution has its policy in dealing with assessment test issues: 3 institutions make immediate repairs, three institutions make changes at a scheduled time, and finally, one institution returns and reprocesses. Testing business reviews for industry-based institutions review the business based on the features built, while for educational institutions, it reviews the assignment letter given to IT unit staff. All institutions implement policies to carry out reviews of technical feasibility. Finally, the validation of the results is carried out by the head of the field with an interest in the system and ratified by institutional officials directly.

REFERENCES

[1] A. A. Setiawan, Y. E. Silaen, T. Andreas, and T. Oktavia, "Analysis of Intention to Use on Pay Later Payment System During COVID-19 Pandemic," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 12, no. 3, pp. 119–129, 2022, doi: 10.46338/IJETAE0322_14.

[2] F. S. Bizarrias, L. F. da Silva, R. Penha, and R. de Fátima Segger Macri Russo, "Relationship between marketing and project management success through cognitive process lens," *IEEE Access*, vol. 8, pp. 169810–169821, 2020, doi: 10.1109/ACCESS.2020.3023897.

[3] S. Lim, A. Henriksson, and J. Zdravkovic, *Data - Driven Requirements Elicitation : A Systematic Literature Review*, vol. 2, no. 1. Springer Singapore, 2021. doi: 10.1007/s42979-020-00416-4.

[4] M. Gillani, H. A. Niaz, and A. Ullah, "Integration of Software Architecture in Requirements Elicitation for Rapid Software Development," *IEEE Access*, vol. 10, pp. 56158–56178, 2022, doi: 10.1109/ACCESS.2022.3177659.

[5] M. Daun, A. M. Grubb, V. Stenkova, and B. Tenbergen, *A systematic literature review of requirements engineering education*, vol. 28, no. 2. Springer London, 2023. doi: 10.1007/s00766-022-00381-9.

[6] A. Ferrari, P. Spoletini, and S. Debnath, *How do requirements evolve during elicitation ? An empirical study combining interviews and app store analysis*, vol. 27, no. 4. Springer London, 2022. doi: 10.1007/s00766-022-00383-7.

[7] M. S. Farooq, M. Ahmed, and M. Emran, "A Survey on Blockchain Acquainted Software Requirements Engineering: Model, Opportunities, Challenges, and Future Directions," *IEEE Access*, vol. 10, pp. 48193–48228, 2022, doi: 10.1109/ACCESS.2022.3171408.

[8] C. Taramasco, "Using Dissemination and Implementation Strategies to Evaluate Requirement Elicitation Guidelines : A Case Study in a Bed Management System," pp. 145787–145802, 2020, doi: 10.1109/ACCESS.2020.3015144.

[9] H. A. Al-Alshaikh, A. A. Mirza, and H. A. Alsalamah, "Extended Rationale-Based Model for Tacit Knowledge Elicitation in Requirements Elicitation Context," *IEEE Access*, vol. 8, pp. 60801–60810, 2020, doi: 10.1109/ACCESS.2020.2982837.

[10] F. M. Khan, J. A. Khan, M. Assam, A. S. Almasoud, A. Abdelmaboud, and M. A. M. Hamza, "A Comparative Systematic Analysis of Stakeholder's Identification Methods in Requirements Elicitation," *IEEE Access*, vol. 10, pp. 30982–31011, 2022, doi: 10.1109/ACCESS.2022.3152073.

[11] I. Nugraha and A. Fatwanto, "User Experience Design Practices in Industry (Case Study from Indonesian Information Technology Companies)," *Elinvo (Electronics, Informatics, Vocat. Educ.*, vol. 6, no. 1, pp. 49–60, 2021, doi: 10.21831/elinvo.v6i1.40958.

[12] K. Alinani, D. Liu, D. Zhou, and G. Wang, "Service Composition and Optimal Selection in Cloud Manufacturing: State-of-the-Art and Research Challenges," *IEEE Access*, vol. 8, pp. 223988–224005, 2020, doi: 10.1109/ACCESS.2020.3045008.

[13] W. Andhika, M. Iqbal, M. Nur, and R. D. Nugroho, "Teknik Representasi Kebutuhan Pengguna Menggunakan User Persona (Studi Kasus: Relasi antara Presensi dengan Kemampuan Kompetensi)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 2, p. 245, 2021, doi: 10.25126/jtiik.0813444.

[14] S. Maalem and N. Zarour, "Challenge of validation in requirements engineering," *J. Innov. Digit. Ecosyst.*, vol. 3, no. 1, pp. 15–21, 2016, doi: 10.1016/j.jides.2016.05.001.

[15] J. S. van Remmen *et al.*, "Natural Language Processing in Requirements Engineering and Its Challenges

for Requirements Modelling in the Engineering Design Domain," *Proc. Des. Soc.*, vol. 3, no. July, pp. 2765–2774, 2023, doi: 10.1017/pds.2023.277.

[16] L. Macaulay, "Requirements for requirements engineering techniques," *Proc. IEEE Int. Conf. Requir. Eng.*, no. May 1996, pp. 157–164, 1996, doi: 10.1109/icre.1996.491440.

[17] M. Bano, D. Zowghi, A. Ferrari, P. Spoletini, and B. Donati, *Teaching requirements elicitation interviews: an empirical study of learning from mistakes*, vol. 24, no. 3. Springer London, 2019. doi: 10.1007/s00766-019-00313-0.

[18] I. Gräßler, D. Preuß, L. Brandt, and M. Mohr, "Efficient Formalisation of Technical Requirements for Generative Engineering," *Proc. Des. Soc.*, vol. 3, no. July, pp. 1595–1604, 2023, doi: 10.1017/pds.2023.160.

[19] F. Halim and D. Siahaan, "Detecting Non-Atomic Requirements in Software Requirements Specifications Using Classification Methods," *2019 1st Int. Conf. Cybern. Intell. Syst. ICORIS 2019*, vol. 1, no. August, pp. 269–273, 2019, doi: 10.1109/ICORIS.2019.8874888.

[20] A. Chaipunyathat and N. Bhumpenpein, "Communication, culture, competency, and stakeholder that contribute to requirement elicitation effectiveness," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 6, pp. 6472–6485, 2022, doi: 10.11591/ijece.v12i6.pp6472-6485.

[21] H. M. O. Zamili, N. Setiyawati, D. H. Bangkalang, and Y. A. Susetyo, "Requirement Engineering Aplikasi Pengelolaan Proses Pertanian Pada Komunitas Tani Menggunakan Loucopoulos Dan Karakostas Iterative Model," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 8, no. 2, pp. 572–584, 2023, doi: 10.29100/jipi.v8i2.3590.

[22] H. F. Martins, A. Carvalho, D. O. Junior, and E. D. Canedo, "Design Thinking : Challenges for Software Requirements Elicitation," pp. 1–27, 2019, doi: 10.3390/info10120371.

[23] J. Ahmad *et al.*, "Quality Requirement Change Management's Challenges: An Exploratory Study Using SLR," *IEEE Access*, vol. 10, no. December, pp. 127575–127588, 2022, doi: 10.1109/ACCESS.2022.3224593.

[24] N. Harahap, *PENELITIAN KUALITATIF*. Wal ashri Publishing, 2020.

[25] S. N. K. Benson, "A Review of 'Observation Techniques: Structured to Unstructured,'" no. October 2014, pp. 21–23, 2010, doi: 10.1080/00220670903231284.

[26] Q. A. Shreda and A. A. Hanani, "Identifying Non-functional Requirements from Unconstrained Documents using Natural Language Processing and Machine Learning Approaches," *IEEE Access*, vol. 4, pp. 1–22, 2021, doi: 10.1109/ACCESS.2021.3052921.

[27] G. S. Walia and J. C. Carver, "A systematic literature review to identify and classify software requirement errors," *Inf. Softw. Technol.*, vol. 51, no. 7, pp. 1087–1109, 2009, doi: 10.1016/j.infsof.2009.01.004.

[28] V. Gupta, J. M. Fernandez-Crehuet, and T. Hanne, "Freelancers in the software development process: A systematic mapping study," *Processes*, vol. 8, no. 10, pp. 1–25, 2020, doi: 10.3390/pr8101215.

[29] M. Romadhoni and W. A. Kusuma, "Elisitasi Kebutuhan Untuk Meningkatkan Kepuasan Pengguna Pada Sistem E-Learning Universitas Muhammadiyah Malang," *Infotek J. Inform. dan Teknol.*, vol. 4, no. 2, pp. 228–237, 2021, doi: 10.29408/jit.v4i2.3602.