ISSN : 1978-8282, Online ISSN: 2655-4275

Development of GoEliTool using Iterative Incremental Approach

Rosa Delima[*1], Joko Purwadi[2]
[1,2] Departement of Informatic, Faculty of Information Technology, Duta Wacana Christian University, Indonesia
E-mail: [*1]**rosadelima@staff.ukdw.ac.id**, [2]jokop@staff.ukdw.ac.id

Abstract

GoEliTool is a tool for requirements elicitation during the requirements engineering process. Such a tool is specifically developed for elicitation through a goal-oriented requirements engineering (GORE) approach. GoEliTool was developed in the previous study. However, the first version of the tool still had many limitations to be corrected so that the process of data elicitation from stakeholders could be carried out more easily and effectively. In this study, the development of the tool was performed through an iterative, incremental approach based on the limitations found in the previous version. The development process consisted of three cycles, and there were three iterations in the second cycle. The tool test was conducted using a black box approach, and the user satisfaction test was conducted using two test questionnaires, namely the User Experience Questionnaire (UEQ) and the System Usability Scale (SUS) questionnaire. The results of the black box test showed that the tool was able to work according to the test scenario. Furthermore, user satisfaction using the UEQ questionnaire showed that the tool was in the above-average category. Regarding the SUS questionnaire, it obtained a grade C result, which indicated the marginal acceptance category.

Keywords *— Iterative Incremental Method, AREM, Requirements Engineering, Requirements Elicitation, Elicitation Tool*

1. INTRODUCTION

Requirements Engineering (RE) collects, analyzes, and specifies stakeholder requirements for software development. This activity is very important and must be performed in the software development cycle. Requirements engineering work needs a lot of interaction with stakeholders, including the organization's owner or management, system operators, or end users. A model that can automate most of the RE processes has been developed in a previous study called AREM (Automatic Requirements Engineering Model)[1].

AREM has four model parts: elicitation, analysis, specification, and analysis models [1]. To support the elicitation model, the previous study developed a tool with a web-based platform called GoEliTool (Goal-Oriented Elicitation Tool) [2]. GoEliTool was developed based on the RE process which applied the Goal-Oriented Requirements Engineering (GORE) approach. GORE is an approach that focuses on the goals of stakeholders to support the RE process [3]. The development of GoEliTool began with the preparation of input document standards for the requirements elicitation process [2]. Document standards could meet all elements of the GORE approach, namely goals, activities, procedures/operations, and

resources needed to carry out applicable activities and procedures [4]. Relationship for each GORE element is made in the form of a goal tree as presented in Figure 1.

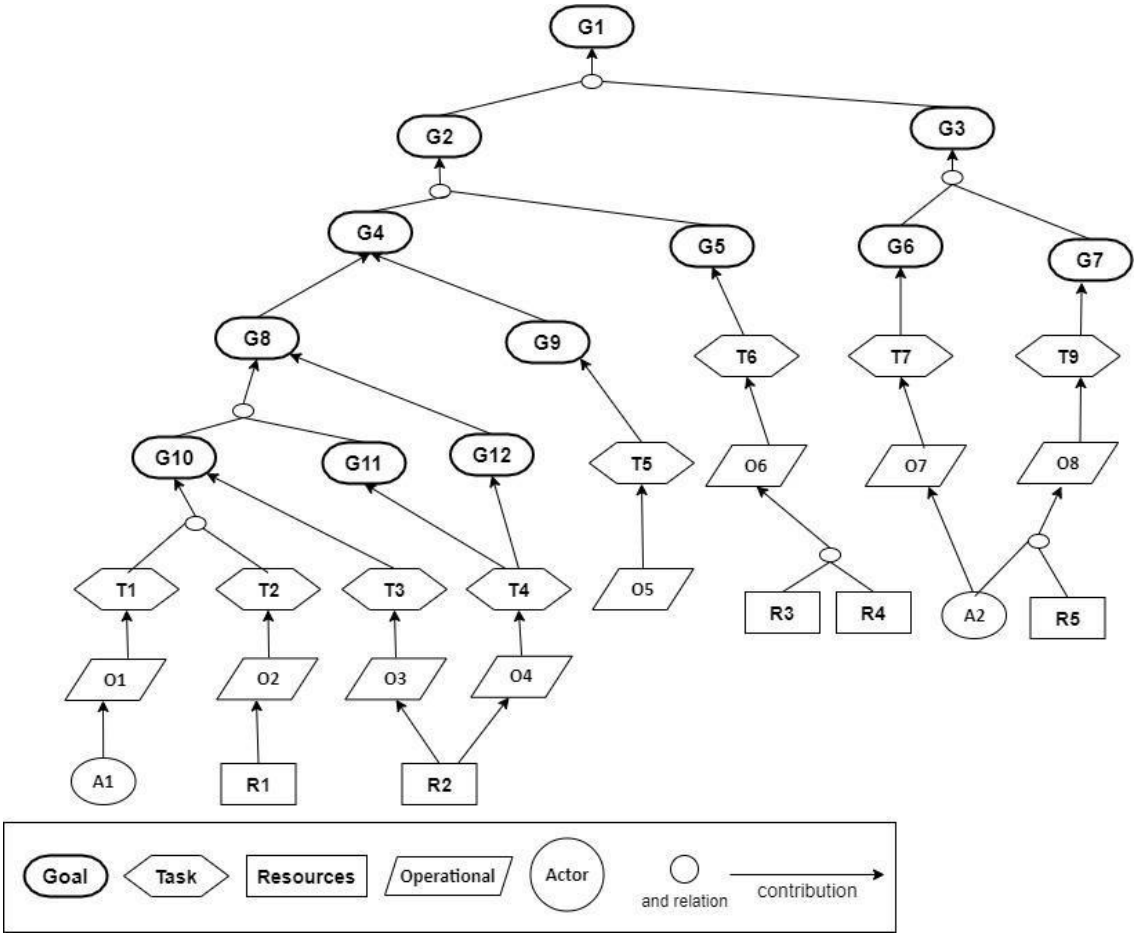

Figure 1. Representation of Requirements Data in Tree Shape [5].

GoElitool has two main features: a feature for entering data on requirements and a feature for extracting data on requirements. The requirements data entry feature is used to record the requirements data from stakeholders, while the requirements data extraction feature is a feature that produces output results from the requirements data extraction to be used for requirements analysis. GoEliTool system architecture can be observed in Figure 2.

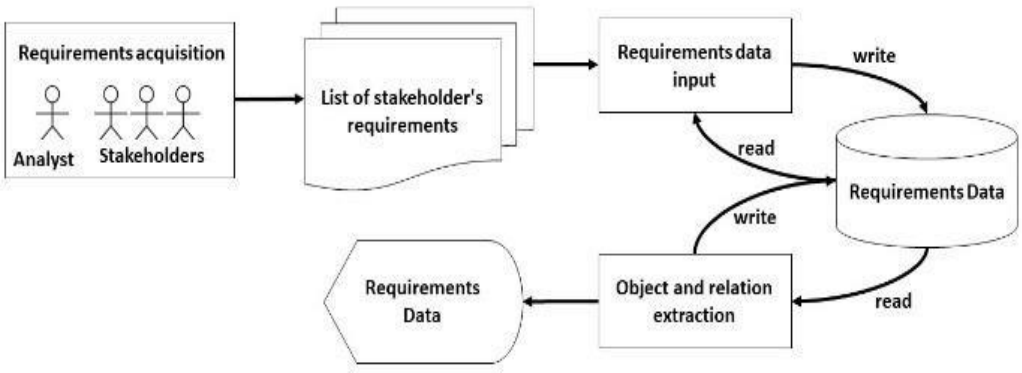

Figure 2. GoEliTool System Architecture [2].

GoEliTool has been used to perform data entry for several information system development projects. The tool test and implementation resulted in a number of recommendations for improving the tool, one of which was regarding tool development to be able to accept data entry for multiple projects. The tool interface for data entry should also be improved so that the tool can minimize the repeated data entry process when a 'one to many' condition exists on the GORE element. As an illustration, each goal can have several activities, the tool user must define the parent goal repeatedly when having to enter activity data. Other limitation of tool was the use of words that were not consistent for each part of the tool and the need for improvement on the dashboard page view [2].

The current study focuses on the development of GoEliTool to improve and enhance the ease and effectiveness of entering data into a certain model. Tool development applied an agile approach with the Iterative and Incremental Development (IID) method.

Software engineering is a systematic approach to developing, operating, and maintaining software [6][7]. Software engineering includes several activities. According to Pressman and Maxim, there are five software engineering activities: communication, planning, modeling, construction, and deployment[6] . Meanwhile, Sommerville defines four software engineering activities: specification, development, validation, and evolution [7].

All activities in software engineering are linked in an engineering process model. There are several models in software engineering, including waterfall, iterative, prototyping, evolutionary, and agile models. Iterative and Incremental Development (IID) is a model that combines iterative and incremental models in the software development process.

The iterative model is a development model that repeats one or more software engineering activities [6]. On the other hand, an incremental model is a software development process in stages within several software versions. Each version is an evolution of the previous version. Each version that has been successfully developed will be tested and implemented. Then, the user can provide feedback on the system, and the system will be evaluated in the next version until all system requirements are met [7]. These two models provide several advantages, including speeding up the software development process, making it easier to test the system, and simplifying risk management [8]. The iterative and incremental models scheme is presented in Figure 3 and 4.

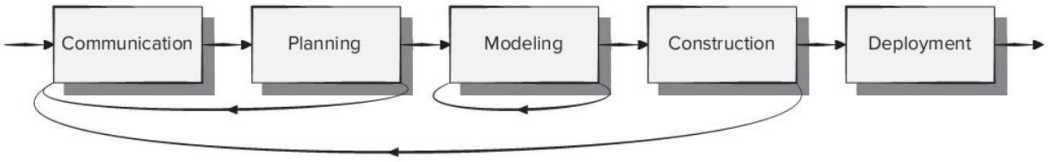

Figure 3. Iterative Process Model [6].

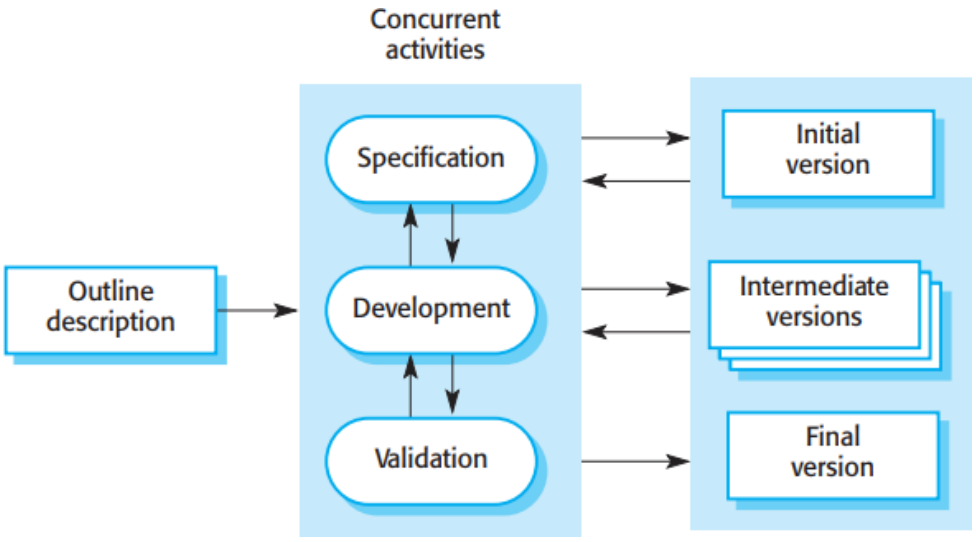

Figure 4. Incremental Process Model [6] [7].

The IID model implements software development in several cycles, wherein each cycle consists of all system development activities. In addition, an iterative process on one or more activities can be carried out in each cycle. The IID model is a model that accommodates an agile approach in software engineering [9]. In essence, the IID model seeks to ensure high system quality, ensure that project management is running well and optimize the development team's work [10]. This study evaluated user satisfaction with the tool using the User Experience Questionnaire (UEQ) and the System Usability Scale (SUS) questionnaire.

The paper is organized into five sections. The first section is the introduction. The second section contains related works, followed by research methods in the third section and results and discussion in the fourth section. The last section of the paper is conclusions and future work.

2. RESEARCH METHOD

This study applied iterative and incremental methods in tool development. The development team consisted of two programmers, one analyst, and one evaluator. The project ran in 4 cycles, and each cycle involved the stages of requirements analysis, design, coding, and testing. The study process scheme can be seen in Figure 5. The first cycle began with the evaluation and application of GoEliTool, which has been developed in a previous study. This evaluation and analysis resulted in a formulation of the requirements for developing the tool. Furthermore, the formulation of such requirements was developed incrementally, starting from the first cycle to the third. The list of features developed in each cycle can be observed shown in Table 1.

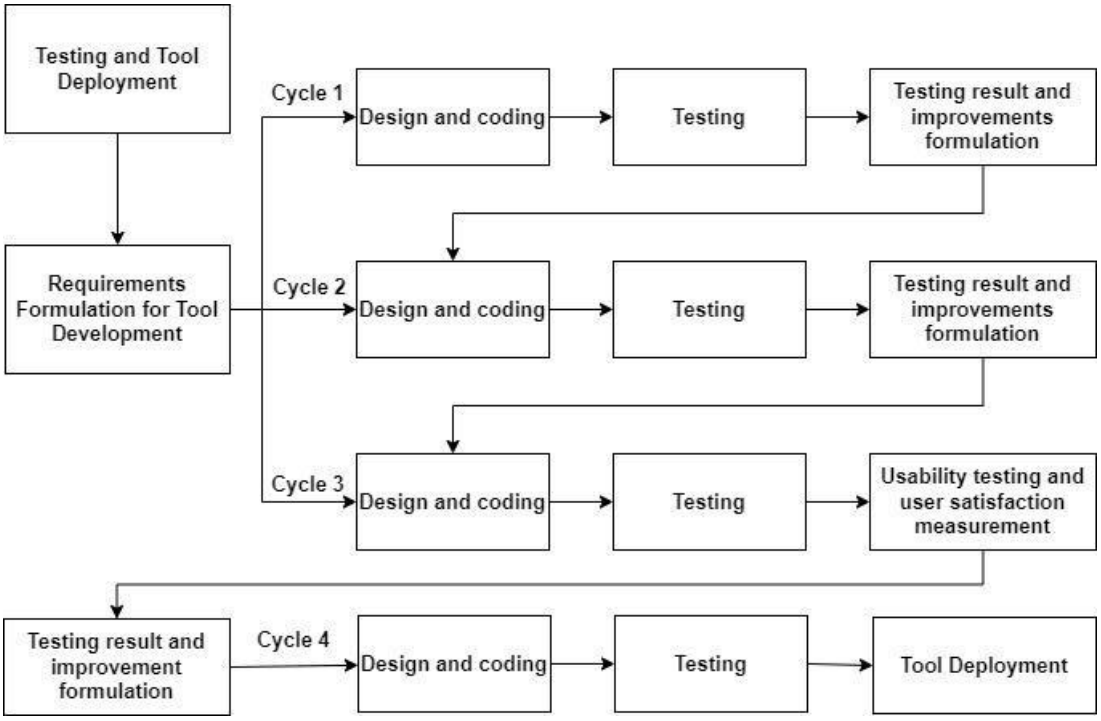

Figure 5. GoEliTool Development Process.

Table 1. List of Features Developed in Each Cycle

Cycle	Feature Improvement
1	Added user management
	The tool's capacity to take entries for a variety of system development projects has been improved
	Enhancing the tool's capacity to receive contributions from a variety of stakeholders
	Improved the features for managing project entry and stakeholders
2	Add Setting Feature
	Improved assist feature for parent selection of goal-oriented elements
	Enhancing the tool's capacity to enter data for all (multiple) sub-goals for the same parent goal
	Enhancing the tool's capacity to enter data for all (multiple) activities for the same parent goal
	Enhancing the tool's capacity to enter data for all (multiple) sub-activities for the same parent goal
	Enhancing the tool's capacity to enter data for all (multiple) procedures for the same parent activity
	Enhancing the tool's capacity to enter data for all (multiple) procedure details for the same parent procedure
3	Improved the dashboard and added information on the number of elements of the goal model
	Equation of terms used in tool
	Improved the icon of the goal element to make it more suitable
	Added report feature for element recapitulation information from the tool
4	Improved the buttons and dashboard page view
	Added features to views and entered data goals and features
	Improved the sidebar menu view
	Added error management

Each cycle was conducted along with the designing, coding, and testing processes. The results of the analysis of the system trials became the input for the next cycle to iteratively improve the tool. A similar process was applied for the second and third cycles. At the end of the third cycle, a usability test and user satisfaction test was conducted using the UEQ and SUS questionnaire. The results of this test became the input for the development of the tool in the fourth cycle. The final tool product was produced from the fourth cycle.

3. RESEARCH RESULTS AND DISCUSSION

3.1. Results of GoEliTool Development

The development of GoEliTool took place in three cycles, wherein the first cycle involved additional features for user registration as well as project and stakeholder sets. In this cycle, addition and improvement were made to project data entry and stakeholder data.

In the second cycle, addition and improvement were made to the goal data entry feature, activities, procedures, and detailed procedures. In the third cycle, the dashboard was added, and the tool interface was improved. Furthermore, in the fourth cycle, the tool interface was improved according to input from the user. Details of features added and improved in each cycle can be seen in Figure 6. Meanwhile, each tool's registration and dashboard pages can be seen in Figure 7 and 8.

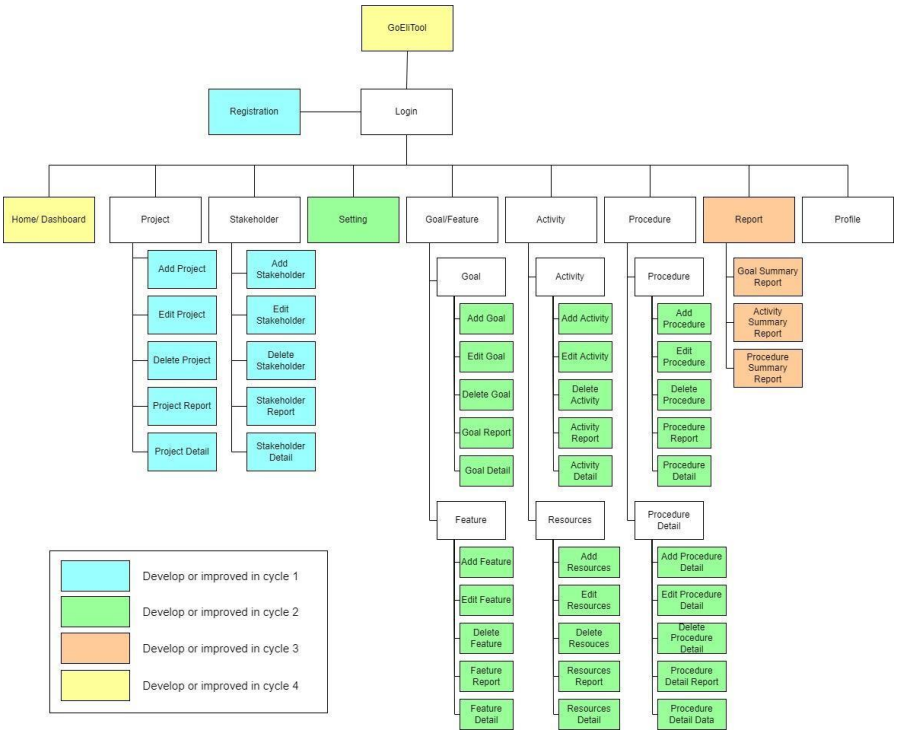

Figure 6. GoEliTool Site Map and Development Process.

3.2. Tool Test

The test was conducted on the functionality of the system. The test was conducted using a black-box approach through test cases by checking whether the input and output of each feature on the system were as expected. In this study, 88 test cases were generated. System tests and improvement were conducted simultaneously whenever errors were found in the tool. The final test results showed that the tool could run all the features according to the functions defined.

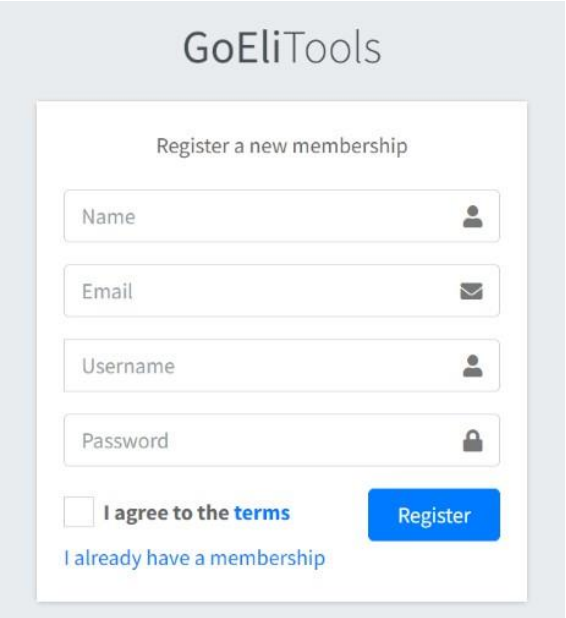

Figure 7. GoEliTool Registration Page.

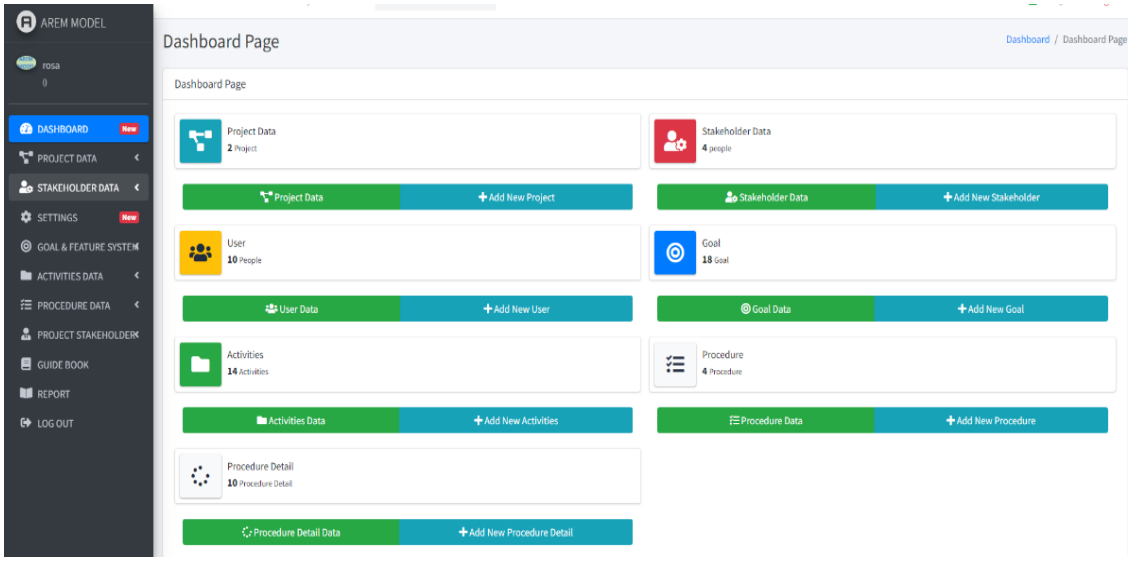

Figure 8. GoEliTool Dashboard Page.

3.3. Results of Usability and User Satisfaction Tests

Usability and user satisfaction tests involved 33 respondents, wherein the respondents were divided into three types analysts, assistant analysts, and general users/operators. Four analysts, 24 assistant analysts, and five general users were involved.

The tool usability test included four test variables: task success, time on task, error, and efficiency. The test results showed that the tool had an effectiveness value of 94% and an efficiency value of 97%.

Evaluation of user satisfaction was conducted using the UEQ and SUS questionnaires. The users of these two questionnaires were intended to complement each other in order to obtain satisfaction levels from various test variables.

UEQ is a questionnaire used to measure user experience (UX). UEQ comprises 26 question items divided into six categories: attractiveness, perspicuity, efficiency, dependability, simulation, and novelty. Attractiveness refers to the overall impression of the product. Do users like or dislike the product? Perspicuity refers to the ease of learning and using a product. Efficiency is a value measuring whether the user can complete the task without making an unnecessary effort. Dependability measures whether the user feels in control of the interaction with the product. Simulation measures whether the product is attractive and motivates users to use the product. Novelty measures whether the product is innovative and creative and matches the user's interests [9]. The list of questions for the UEQ questionnaire can be observed in Figure 9. Analysis of the results of the UEQ questionnaire was conducted using the UEQ tool, which can be accessed on the https://www.ueq-online.org/ page.

Analysis of the data collection results through the UEQ questionnaire was conducted using an excel tool from the https://www.ueq-online.org/ page. The results of the analysis showed that respondents gave positive responses to all categories and all questions. The graph

regarding the interpretation of UEQ results for all categories and all questions can be observed in Figure 10 and Figure 11.

	1	2	3	4	5	6	7		
annoying	O	O	O	O	O	O	O	enjoyable	1
not understandable	O	O	O	O	O	O	O	understandable	2
creative	O	O	O	O	O	O	O	dull	3
easy to learn	O	O	O	O	O	O	O	difficult to learn	4
valuable	O	O	O	O	O	O	O	inferior	5
boring	O	O	O	O	O	O	O	exciting	6
not interesting	O	O	O	O	O	O	O	interesting	7
unpredictable	O	O	O	O	O	O	O	predictable	8
fast	O	O	O	O	O	O	O	slow	9
inventive	O	O	O	O	O	O	O	conventional	10
obstructive	O	O	O	O	O	O	O	supportive	11
good	O	O	O	O	O	O	O	bad	12
complicated	O	O	O	O	O	O	O	easy	13
unlikable	O	O	O	O	O	O	O	pleasing	14
usual	O	O	O	O	O	O	O	leading edge	15
unpleasant	O	O	O	O	O	O	O	pleasant	16
secure	O	O	O	O	O	O	O	not secure	17
motivating	O	O	O	O	O	O	O	demotivating	18
meets expectations	O	O	O	O	O	O	O	does not meet expectations	19
inefficient	O	O	O	O	O	O	O	efficient	20
clear	O	O	O	O	O	O	O	confusing	21
impractical	O	O	O	O	O	O	O	practical	22
organized	O	O	O	O	O	O	O	cluttered	23
attractive	O	O	O	O	O	O	O	unattractive	24
friendly	O	O	O	O	O	O	O	unfriendly	25
conservative	O	O	O	O	O	O	O	innovative	26

Figure 9. UEQ Questionnaire.

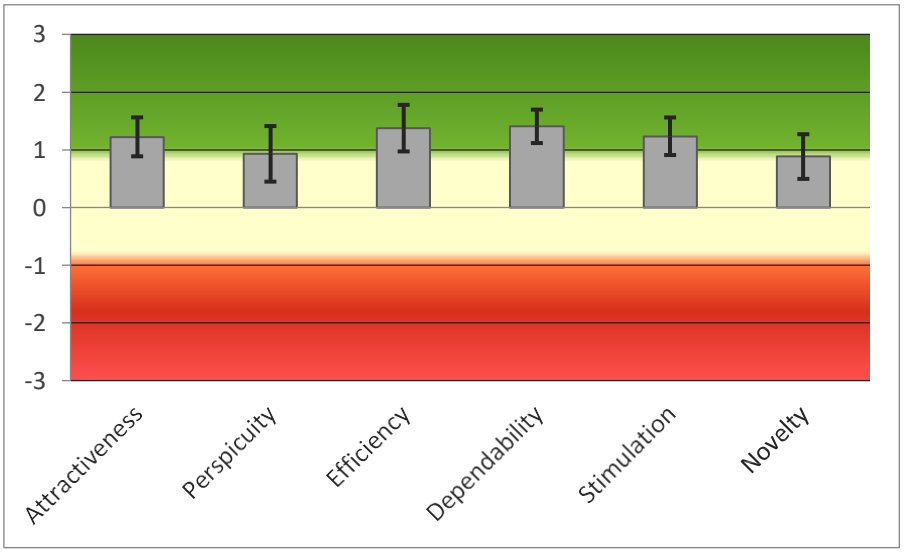

Figure 10. Interpretation of UEQ results on GoEliTool in normal distribution standard

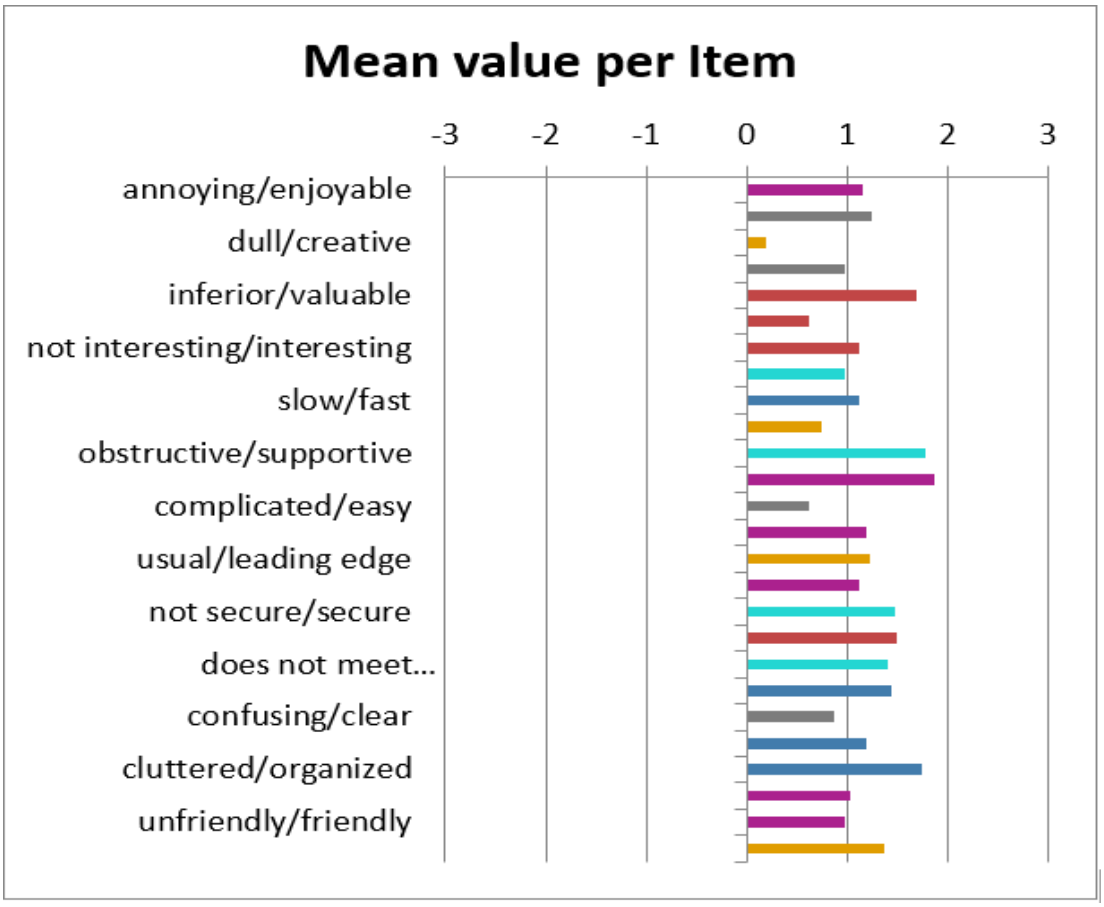

Figure 11. Representation of the mean values of the 26 assessment elements in UEQ on GoEliTool.

Based on UEQ data analysis, it was found that GoEliTool had a level of satisfaction of Above Average for the categories of attractiveness, efficiency, dependability, stimulation, and novelty. However, the perspicuity category showed a below-average response. This indicated that the tool still needs improvement regarding the clarity of the tool in terms of ease of use, to be understood, and clarity of the elements contained in the tool. The graph regarding analysis results for each category can be observed in Figure 12 and Table 2.

The results of the analysis for respondents' assessment of GoEliTool using a Likert scale of 1 – 7 can be seen in Figure 13. Based on this figure, it was known that the mean rating for each category was above 5.

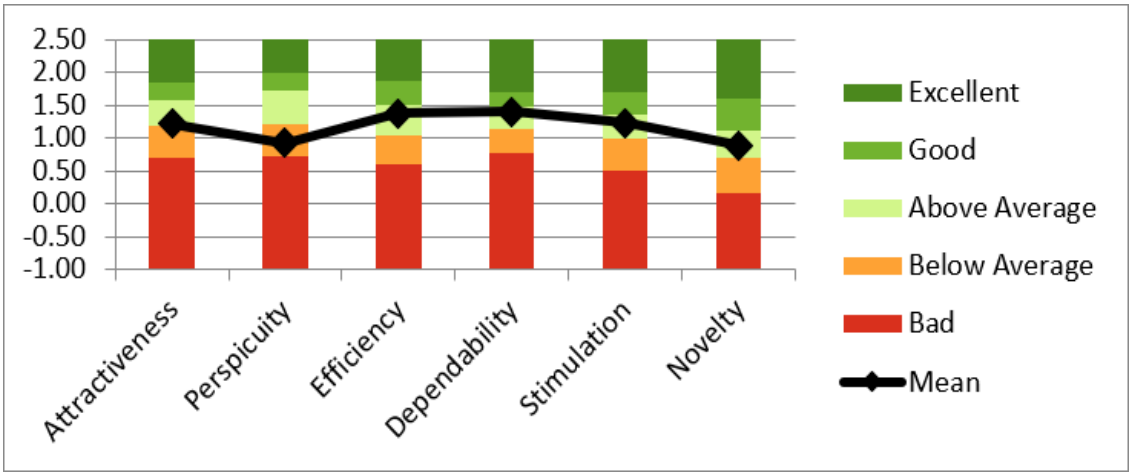

Figure 12. Categories of UEQ questionnaire results for GoEliTool.

Table 2. The Mean Value of each Category, along with the Interpretation of Results based on Benchmarks

Scale	Mean	Comparison to benchmark	Interpretation
Attractiveness	1.22	**Above average**	25% of results better, 50% of results worse
Perspicuity	0.93	**Below Average**	50% of results better, 25% of results worse
Efficiency	1.38	**Above Average**	25% of results better, 50% of results worse
Dependability	1.41	**Above Average**	25% of results better, 50% of results worse
Stimulation	1.23	**Above Average**	25% of results better, 50% of results worse
Novelty	0.88	**Above Average**	25% of results better, 50% of results worse

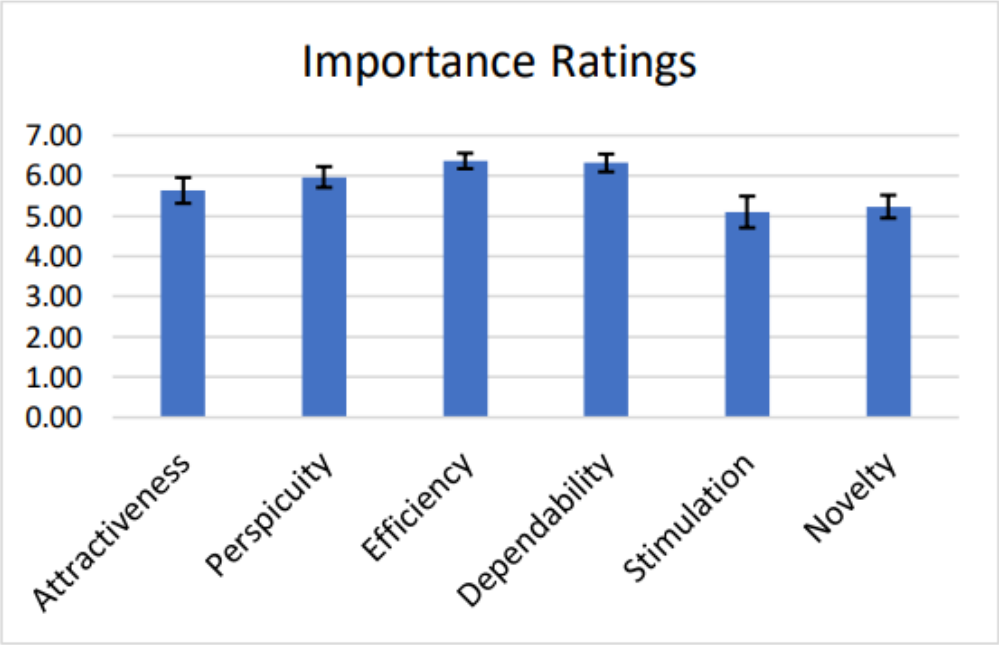

Figure 13. Ratings for each category in 7 Likert scale

Evaluation of user satisfaction was also performed using the SUS questionnaire. SUS is a questionnaire developed by John Brooke in 1986. The SUS questionnaire consists of 10 questions, where odd-number questions are positive questions, and even-number questions are negative questions. Each question has five scales ranging from "strongly disagree" to "Strongly agree" [10]. The list of questions on the SUS questionnaire is presented in Table 3.

Table 3. List of Questions on the SUS Questionnaire [11]

Num.	Pertanyaan
1.	I think that I would like to use this system frequently.
2.	I found the system unnecessarily complex.
3.	I thought the system was easy to use.
4.	I think that I would need the support of a technical person to be able to use this system.
5.	I found the various functions in this system were well integrated.
6.	I thought there was too much inconsistency in this system.
7.	I would imagine that most people would learn to use this system very quickly.
8.	I found the system very cumbersome to use.
9.	I felt very confident using the system.
10.	I needed to learn a lot of things before I could get going with this system.

SUS questionnaire analysis was performed by applying the calculation process as follows:

- Each question had a score of 1 for strongly disagree to a score of 5 for strongly agree.
- For each odd-number question, the user's score for each question was subtracted by 1.
- For each even-numbered question, a score of 5 was subtracted from the user's score for each question.
- The SUS score was obtained from the sum of the scores for each question multiplied by 2.5.

A qualitative assessment of the final SUS score was made based on the categorization of SUS scores, as shown in Figure 14.

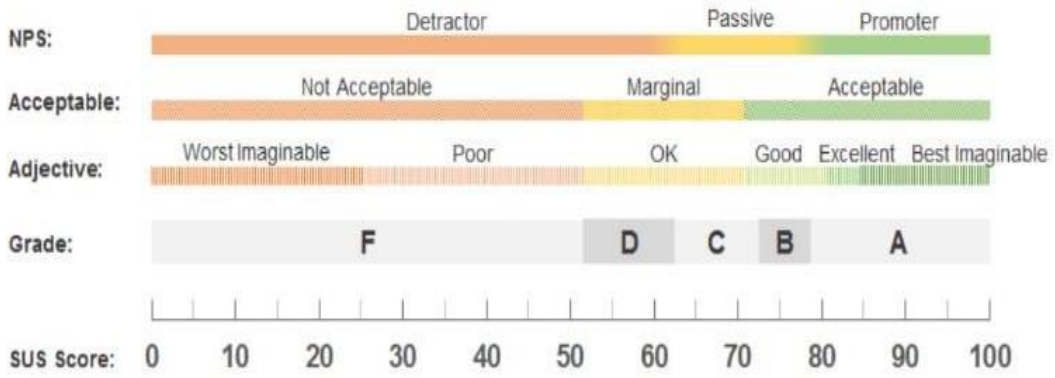

Figure 14. Rating Scale for SUS Questionnaire [12].

The results of the analysis of the SUS questionnaire showed a mean average SUS value of 62.75 which indicated that the tool was in grade C- with a percentile rank of 62.7–64.9, and quality was involved in the OK group with acceptable, marginal variable and Net Promoter Score (NPS) in the passive class. Based on such findings, it can be said that the tool still needs to be improved since the SUS score was still below 68. The results of the SUS questionnaire analysis can be seen in Table 4.

Table 4. Tool category groups by SUS value

Grade	SUS	Percentile Range	Adjective	Acceptable	NPS
C-	62,7 – 64,9	35-40	Ok	Marginal	Passive

During the evaluation process, recommendations for improvement were also collected from users. Tool improvement according to recommendations derived from users was conducted in the fourth cycle of the system development process.

Based on the usability test results, it was known that the tool was effective and efficient in supporting the elicitation of requirements in AREM. However, based on the satisfaction evaluation, it was found that the tool still needs interface improvement to increase user satisfaction. Such limitation is understandable because the tool was not developed with a user-based approach, so the interface could not perfectly meet the user's expectations.

4. CONCLUSION

The current study has developed GoEliTool as a tool that functions to record data on user requirements at the requirements elicitation stage. The development of GoEliTool was performed through iterative and incremental methods. The development process consisted of four cycles involving requirements analysis, design, coding, and testing stages in each cycle.

The results of the black box test showed that the tool was able to work according to the test scenario. Evaluation of user satisfaction was conducted using the UEQ and SUS questionnaires. The results of the satisfaction evaluation showed a moderate level of satisfaction. Furthermore, the results of user satisfaction using the UEQ questionnaire showed that the tool was in the above-average category for 5 of the 6 UEQ categories. Regarding the SUS questionnaire, it obtained a grade C result, which indicated the marginal acceptance category.

5. SUGGESTED

The final development conducted in the fourth cycle included recommendations for improvement derived from respondents. However, this cycle was not re-tested. Therefore, the future study can design an evaluation to determine the satisfaction response from the user. Besides, it is necessary to integrate the tool with the analysis, specification, and requirement validation models in AREM.

6. REFERENCES

[1] R. Delima, R. Wardoyo, and K. Mustofa, “Automatic Requirements Engineering Model using Goal-Oriented Modelling with Text Pre-Processing Technique,” in Sixth International Conference on Informatics and Computing (ICIC), 2021, pp. 1–8. doi: 10.1109/icic54025.2021.9632980.

[2] R. Delima and J. Purwadi, “GoEliTool for Software Requirements Elicitation using Goal-Oriented Approach,” J. Edukasi dan Penelit. Inform., vol. 8, no. 3, pp. 432–440, 2022.

[3] A. Van Lamsweerde, “Goal-oriented requirements engineering: A guided tour,” in Proceedings of the IEEE International Conference on Requirements Engineering, 2001, pp. 249–261.

[4] P. Giorgini, B. W. Eds, I. Conference, and D. Hutchison, “Advanced Information Systems Engineering,” in 31st International Conference, CAiSE 2019 Rome, 2019, vol. 932. doi: 10.1007/3-540-59498-1.

[5] R. Delima, R. Wardoyo, and K. Mustofa, “Requirements Conflict Detection and Resolution in AREM Using Intelligence System Approach,” JUITA J. Inform., vol. 10, no. 2, p. 191, 2022, doi: 10.30595/juita.v10i2.14855.

[6] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach (Ninth Edition), Ninth Edit. New York: Mc Graw Hill Education, 2020. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84865607390&partnerID=tZOtx3y1%0Ahttp://books.google.com/books?hl=en&lr=&id=2LIMMD9FVXkC&oi=fnd&pg=PR5&dq=Principles+of+Digital+Image+Processing+fundamental+techniques&ots=HjrHeuS_

[7] I. Sommerville, Software engineering (10th edition), Tenth Edit. Pearson Education, 2016.

[8] D. Singh, A. Thakur, and A. Chaudhary, "A Comparative Study between Waterfall and Incremental Software Development Life Cycle Model," Int. J. Emerg. Trends Sci. Technol., vol. 02, no. 04, pp. 2202–2208, 2015, [Online]. Available: www.ijetst.in

[9] M. Zeller and S. Klabes, "Iterative and Incremental Development of Reliable Systems," in Proceedings of the 29th European Safety and Reliability Conference, 2019, pp. 2433–2439. doi: 10.3850/978-981-11-2724-3.

[10] I. M. Ibrahim, "Iterative and Incremental Development Analysis Study of Vocational Career Information Systems," Int. J. Softw. Eng. Appl., vol. 11, no. 5, pp. 13–24, 2020, doi: 10.5121/ijsea.2020.11502.

[11] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," Usability Eval. Ind., no. November 1995, pp. 207–212, 1995, doi: 10.1201/9781498710411-35.

[12] J. Sauro, "5 Ways to Interpret a SUS Score," Measuring U, 2018. https://measuringu.com/interpret-sus-score/