Improvement Quality of Software Requirements Using Requirement Negotiation System for Supporting Decision

Egia Rosi Subhiyakto[1], Yani Parti Astuti[2]

[1,2]*Informatics Engineering Department, Faculty of Computer Science, Dian Nuswantoro University*
Semarang, Indonesia
[1]`egia@dsn.dinus.ac.id`, [2]`yanipartiastuti@dsn.dinus.ac.id`

Abstract - The Requirement Engineering phase, where all requests and software requirements of the user and the client are delivered, understood and agreed upon. However, often the developers are just too focused on implementing the software, even though the Requirements Engineering phase is a phase that can have a big impact. The impact is not only on the final product but also on the development process itself. In this study, the authors conducted software development negotiation of software requirements as a medium for stakeholders to negotiate the requirements of software products. In the negotiation system, the author will provide a means of decision support or group decision support system that a method of resolving conflicts. The main objectives of this work are twofold: 1) to assist the negotiation process between stakeholders and 2) to improvement quality software after negotiation. The workings of the E-Voting method are by giving choices to each sub-specification that has been chosen by stakeholders. We will select the choice that has the highest number of votes as a specification. We used prototyping as a method of developing a system life cycle because prototyping is very open to improvements that might occur after it releases the prototype version system. The results of evaluations show that the system has a high success rate based on 3 dimensions of testing, Performance (80.5%), Usability (78.5%), and User Satisfaction (78%).

Keywords: Requirements, negotiation, improvement, prototyping

I. INTRODUCTION

In a software system development process, the Requirement Engineering phase is the earliest phase where the software requirements of the user and client are collected, understood together, and an agreement will occur between the developer and the client. At this stage, it can be called a crucial stage because the developer is faced by various stakeholders with their various requirements, which are tailored to the background of the company or client. This stage is complex in a series of software engineering processes[1]. It often refers this process to as the most difficult phase because the validation percentage of the Requirements Engineering process itself must reach a high number. If a system failure occurs when the development process is complete, it will cause various consequences. On a small scale, it will take a lot of time to correct errors, system performance, and functionality. Meanwhile, for a large scale, it will cause losing the trust of stakeholders, losing orders and reducing profits, also risking the reputation of the developer[2].

We can use negotiation as a method and tool for negotiation in various disciplines. We can say negotiation to be a phase in the decision-making process. In decision making there are firm foundations related to the field of science itself and the situation at hand. However, negotiating in a group-decision is challenging and produces its complexities for negotiators[3]. In the [4] introduced a tool called C-FaRM is useful in managing knowledge of requirements and various types of requirement artifacts. Within this framework uses an ontology-based recommendation system that helps prioritize, visualizing, and negotiating requirements.

In the other hand, using the latest meta-heuristic algorithm Owl Search Algorithm (OSA), and chaos theory, in the [5] offer a solution based on an automated bilateral negotiation model. The Chaotic Owl Search Algorithm (COSA) was used to adapt the negotiation strategy to calculate bids during the negotiation process. Based on the comparison results, the COSA algorithm used is proven to be accurate in terms of utility, average round of negotiations and processing time. Cooperative negotiations based on fuzzy inference used as approach [6], authors propose a fuzzy-based two-layer control architecture. There are pairwise negotiations between agents according to the couplings and the communication network. The resulting pairwise control sequences are sent to a coordinator in the upper control layer.

In recent decades, various methods have been proposed to manage requirements, there are several related studies, including [7] which conducted research on solution-based requirements management based on the git control system. They developed a tool that supports managing requirements for large-scale development using agile methods. In the [8] authors investigate application of three multi-criteria decision-making methods. In the other hand, trust negotiation is important. In the other hand, [9] introduced a type of trust management model for establishing trust between entities by a mutual exchange of credentials. Author's discusses and presents a model that uses UML notation to design trust negotiations. The specifications created will become part of the SDLC, which provides a solid and reliable foundation in terms of software development. In another discussion, related to cloud computing mentioned that from the consumer's point of view, they want their business needs to be met and provided with the best quality service. Meanwhile, from the cloud provider side, they want to sell services that suit their preferences[10]. And the last is [11] which uses the multi classifier voting method.

II. METHOD

A. Research Method

The data collected is primary data because the authors get it directly from the research subject and are qualitative. In this study, the data used comprised 2 types, namely from the developer and data by the pharmacy (client).

Developer data is got by distributing a questionnaire form regarding the respondent's opinion on a software requirements negotiation system. In this questionnaire, the authors target respondents, namely software developers and Informatics Engineering students who have a background in software development. This questionnaire uses several questions (parameters) as the major focus of respondents' opinions on negotiations, including:

- Respondent's agreement regarding the negotiation system conducted via the web, the respondent can answer "Yes" to agree and "No" to disagree.
- Respondents' preferences in pre-negotiation, whether respondents prefer to recognize the problem first or to identify the stakeholder first.
- Respondent's agreement regarding whether it is necessary to describe the interface design / UML / illustration of the system being built. Respondents can answer "Yes" to agree and "No" to disagree.

- Respondent's agreement regarding the role of voting as a problem-solving method in negotiations, respondents can answer the respondent, can answer "Yes" to agree and "No" to disagree.
- Respondents' preferences regarding the applicable Voting system, whether it is the Plurality Method or Majority Rule.

At the end of the questionnaire, it includes a column name and email to identify the respondent.

Client data is got by distributing a questionnaire form regarding the respondent's opinion on the needs of a pharmacy system. In this questionnaire, the authors target respondents, namely pharmacy employees/owners, both pharmacies that already have a pharmacy management software system or pharmacy websites and pharmacies that don't have both. This questionnaire uses several questions (parameters) as the primary focus of respondents regarding their needs, including:

- Respondents' preferences regarding the software to be built are based on the pharmacy's actual priority. Respondents can choose the website as a pharmacy information system or management system.
- The question is valid only if the respondent answers that they already have a software system in the pharmacy. This question discusses features that are not yet available in the existing system. The answer choices in this questionnaire include Data Security (Database Backup, Restore Database, and Privacy User), Data Menu (Supplier data, products, concoctions, similar drugs, doctor data, and stock hospitalization), Reports Menu (Recap transaction data, etc.), and Transactions with Credit, Accounts Payable, Loss / Profit Report and others which the respondent can write himself. The choice of answer to this question is a checkbox so that the respondent can choose over 1 choice.
- Continuation of the previous question, respondents are required to choose a maximum of 3 choices of answers regarding the most important features that are respondents' preferences for the pharmacy system based on the actual needs of the pharmacy.

At the end of the questionnaire, the authors include a column for the respondent's name and the respondent's pharmacy to identify the respondent's answer.

From the data collected, the authors performed the data analysis technique by:

- Both primary data will be grouped into developer data and client data.

- The two data were analyzed based on the existing parameters.
- Conclusions are drawn on each data source.
- The conclusion from the two data sources will the main parameter in the negotiation's development system requirements in this study, where the conclusions from the developer data sources will a reference in building a comfortable negotiation space and system in the system, while the client data sources a reference in forming Group decision support system to assist the voting method process in making negotiation conclusions.

B. Requirement Negotiation Method

Requirements Negotiation Method in this system will run through 3 stages of the negotiation process pre-negotiation, negotiation, and post-negotiation. The parameter points of the pre-negotiation process will lead to the negotiations that will be carried out by the two stakeholders.

1) The pre-negotiation process that will be carried out in this research is:

- Problem definition: Identifying problems that exist in the problem object. Here, an example is a pharmacy, the problem at a pharmacy is in the form of a need for an administration system or information system.
- Stakeholder Identification: Identity who is involved in negotiating the need for pharmacy software. Here, the developer needs to know who he is dealing with, whether the owner of the pharmacy or just a pharmacy manager.

2) The negotiation process will run in which the two stakeholders will exchange offers based on the needs and desires of each stakeholder. Matters related to negotiating needs in research are:

- Stakeholders will communicate and negotiate in a forum/means of negotiation in the form of a chat forum that can be accessed via the web.
- Stakeholders can provide/read comments and send/access pictures and product design illustrations.

3) In the post-negotiation process itself, voting will be carried out as group-decision support from the negotiation forum, both parties will test which to ensure the certainty of the negotiation results. Key stakeholders are actors who have a role to input points of need and choose to be selected.

C. E-Voting Method

Voting is one of the most important acts through which a community can make a collective decision [12].

Based on the results of research conducted [13] shows that e-voting has no effect on the number of voters. In this research, innovation is carried out by analyzing citizen participation. They estimated a multi-level Bayesian model on official data on Swiss citizen participation. The results showed that e-voting has increased the number of voters.

Below is class e-voting, consist of name of class, attribute, and method. Fig. 1 Shows the class diagram e-voting process, and will use these algorithms. Meanwhile in the Fig. 2, the notation of the E-Voting algorithm begins with the function for the input (vote) given by the user.

In the Fig. 3 show the results of the sum of votes will be accumulated into the results of negotiations with the following notation.

III. RESULT AND DISCUSSION

A. Data Analysis

One part of a software project is to carry out the software management process. The important thing related to it is the appraisal of the developed project, and failure due to wrong management practices. In the [14] discuss about software estimation tools built to find efficient and accurate methods for estimating effort. So we also have to follow the appropriate development stage process.
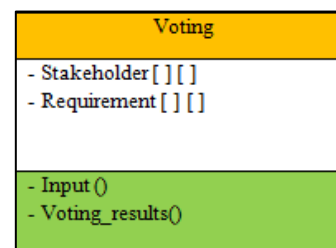

Fig. 1 Class diagram e-voting

```
for r in requirement
    Input choice
    Requirement [r][choice]++
    Stakeholder [s][r] =choice
```

Fig. 2 Source code function vote

```
Voting_results=[ ]
For r requirement
Max= requirement [i]
      if requirement [r][i] > requirement [r][max]
      max = i
Voting_result[r] = max
```

Fig. 3 Source code e-voting results

Based on the data collection process through the methods mentioned in the previous chapter, the authors got data results from both types of data. We present the results for client data and developer data below.

In collecting developer data, we use several question parameters that will support the development and construction of this negotiation system. All the parameters of the questions given to 17 respondents, which comprise Informatics Engineering students with an interest in Software Engineering and software developers who understand the negotiation process. Table I below is a table of overall developer data collection.

In collecting client data, we use several question parameters that will support the development and construction of this negotiation system, where all parameters of the questions are answered by 9 respondents who work, all of whom have status as employees of several pharmacies. Following are Table II and graphs of client data collection results.

B. Design

In this section, we have design for this system such as high level architecture, and design user interface. Fig. 4 shows the feature of our negotiation systems. The main components of this tool are: 1) registration for guest and login after register, 2) creating the negotiation for stakeholders, and discuss in the forum, stakeholders also can search negotiation forum that has been done. To describe system functionality, you can use UML as shown in [15]. Next, we will model the system using use case diagrams. Use case diagram is one of the diagrams in UML, and is often taught in universities, especially computer science[16].

TABLE I
QUESTIONNAIRE OVER THE DEVELOPERS

Parameters	Results
Respondent's agreement regarding the negotiation system conducted via the web	Agree: 82,4% Disagree: 17,6%
Respondent preference in pre-negotiation	Identify Stakeholders first: 58,8% Identify problem first: 41,2%
Respondent's agreement regarding whether it is necessary to describe the interface design / UML / illustration of the system is built	Agree: 94,1% Disagree: 5,9%
Voting as a negotiation method	Agree: 76,5% Disagree: 23,5%
Respondents' preferences regarding the applicable Voting system	Plurality Method: 52,9% Majority Rule: 47,1%

TABLE III
QUESTIONNAIRE OVER THE CLIENTS

| Parameter	Results
Respondents' preferences regarding the software to be built are based on the actual priority of the pharmacy	Pharmacy Management System: 55% Website Profile: 45%
Respondents' preferences regarding features that are not available in the existing system (For pharmacies that are computerized)	Credit Transaction: 46% Income Statement: 23% System Security: 15% Reports: 15%
The most important feature for the Pharmacy.	Master Data: 33% Reports: 26% Security System: 20% Transaction: 13% Income Statement: 6%
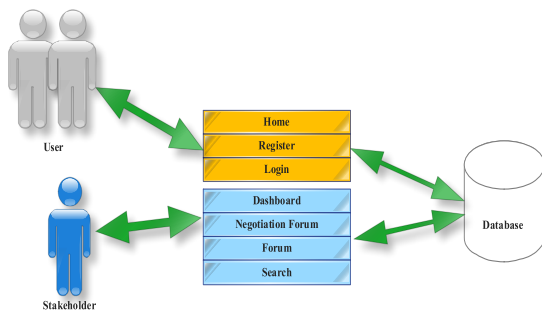
Fig. 4 High-level architecture negotiation systems

Fig. 5 is a use case diagram illustrating application functionality and shows the relationship between actors and associated use cases. There are two actors, namely moderators and stakeholders. Each actor has a different role. Moderators can add forums, add and remove stakeholders, add and delete comments, provide model illustrations, guide the voting process, and view voting results. Meanwhile, stakeholders can add and delete comments, provide model illustrations, conduct the voting process and view the voting results.

C. Implementation

The result of implementing this research is a Web-based application. From input, processing to output, all users will get on the web. We chose the name, "RENego" as the branding name of this Software Negotiation System web application, which stands for "Requirements Software Negotiation". This application has 3 principal parts that cover the stages in negotiations:

1) Pre-negotiation shows in Fig. 6, where key stakeholders create a negotiation forum and include stakeholders. Includes the Forum Dashboard page. The following is a display of the Forum Dashboard page.

2) Negotiation shows in Fig. 7, where each stakeholder will negotiate such as giving comments,

sending illustrations of software requirements design, covering the Negotiation Forum page.

3) Post-negotiation shows in Fig. 8, where each stakeholder will vote as a step for solving problems in negotiating software requirements. Includes voting pages and negotiation results.

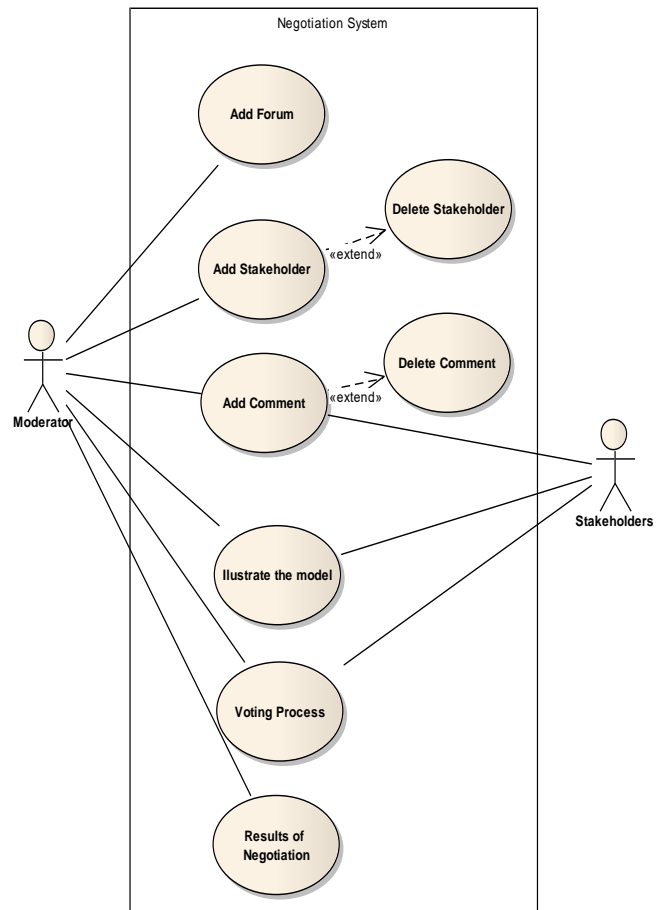While in the Fig. 9 shows the source code for voting method.
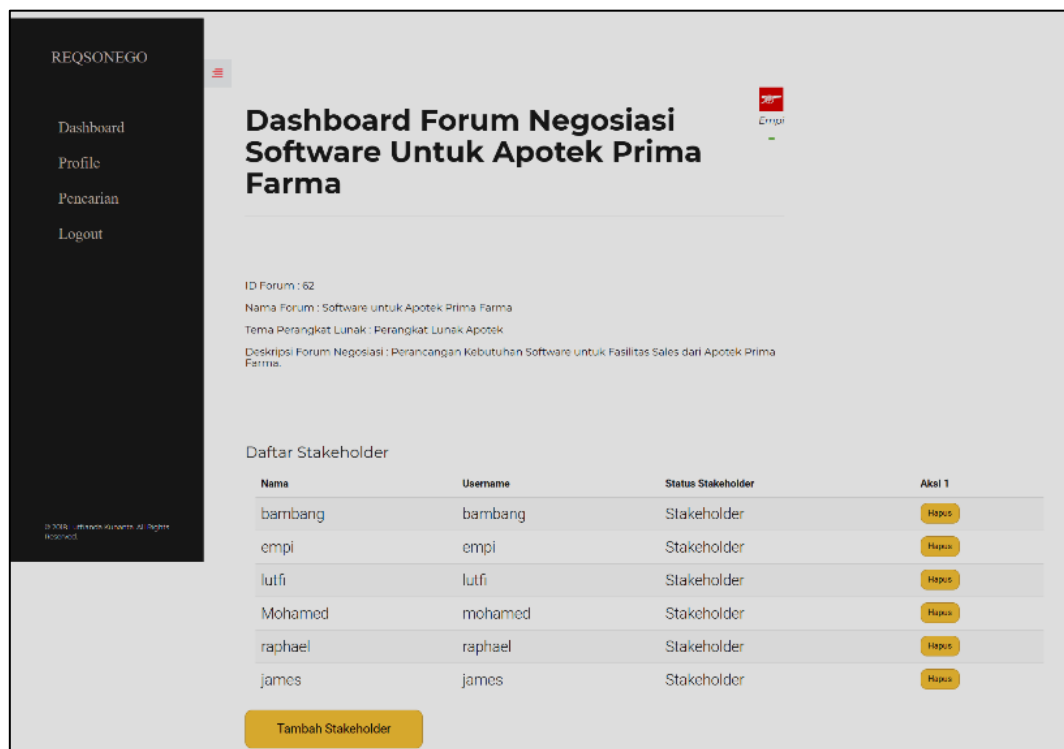

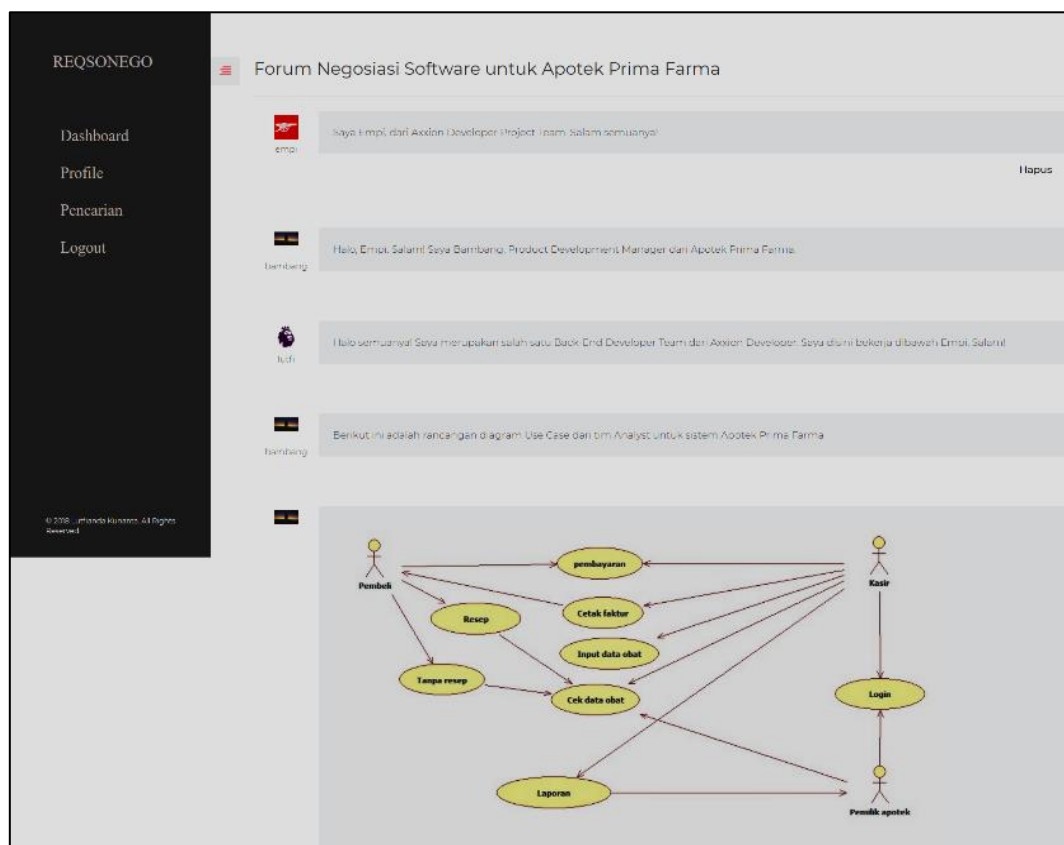**Fig. 5 Use case diagram negotiation systems**

Fig. 6 Dashboard page

Fig. 7 Negotiation page

Fig. 8 Voting page

```
{
        $cekUser = Vote::where('id_forum', $request->id_forum)->where('id_user',
Auth::user()->id)->get();
        if(!$cekUser->isEmpty()){
            return redirect('/voting?id_forum='.$request->id_forum);
        }
        $vote = new Vote();
        $vote->id_forum = $request->id_forum;
        $vote->id_user = Auth::user()->id;
        $vote->save();
        for($i=0;$i<$request->index;$i++){
            $voteDetail = new VoteDetail();
            $voteDetail->id_user = Auth::user()->id;;
            $voteDetail->id_vote = $vote->id;
            $voteDetail->id_requirement = $request['requirement_'.$i];
            $voteDetail->id_option = $request['option_'.$i];
            $voteDetail->save();
            $option = Option::find($request['option_'.$i]);
            $jumlah = $option->score+1;
            $option->score = $count;
            $option->save();
        }
        return redirect('/voteresult?id_forum='.$request->id_forum);
    }
```

Fig. 9 Source code voting page

D. User Acceptance Testing

We need user Acceptance Test to increase the usefulness and advances of the system. There are many testing strategies and method can used [17]. In this test, the author provides a piece of paper having several questions that will be held out by respondents who will later be involved in this system, 10 respondents comprising 5 each from software developers (software engineering students, business analysts, and programmers) and 5 clients (assistant-pharmacist and the public). The questions that stand describe several dimensions of the system, cognitive (convenience), performance, and user satisfaction in accepting this

system. The following are the test results of the software requirements negotiation system.

Fig. 10 shows the results for the performance survey. The graph showed positive results for each parts of the survey with agreement easy access of the system (96% strongly agree and agree on its easy access), the system accuracy (over 90%), voting accuracy (over 90%) and data input definition (over 90%). However, in terms of error handling, this system should be better and improved (70%).

Fig. 11 shows the results tests on the cognitive/ eases of System dimensions, from graphs and tables show that positive results from all respondents, 76% think that the features in the system are easy to use, 86% of data management is easy, and 90% consider the process for voting needs easy. However, to learn and understanding the system, this system should be better and improved (62%).

Fig. 12 shows the user satisfaction dimension, the small percentage of disagreement (less 5%) over the performance and satisfaction is related to ability of the participant to understand to using this system. The results of the User Acceptance test give a value on a fairly high likert scale. Respondents agree that the "RENego" software requirements negotiation system has a positive impact both in terms of performance, cognitive, and user satisfaction.
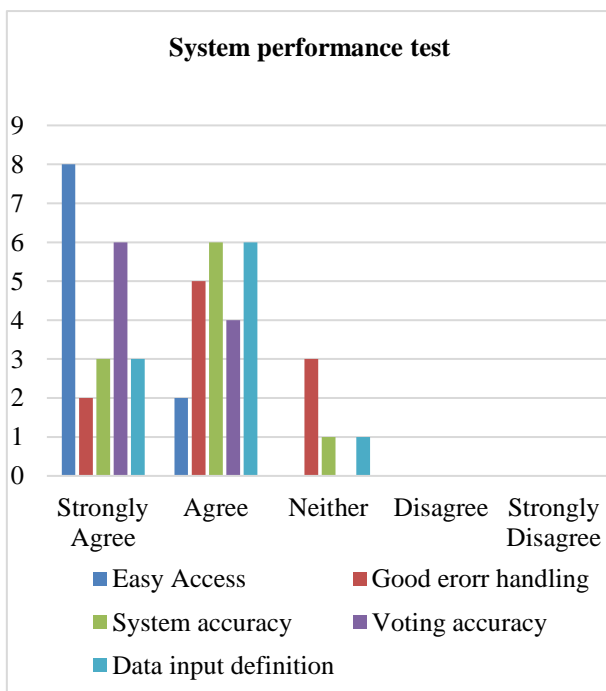

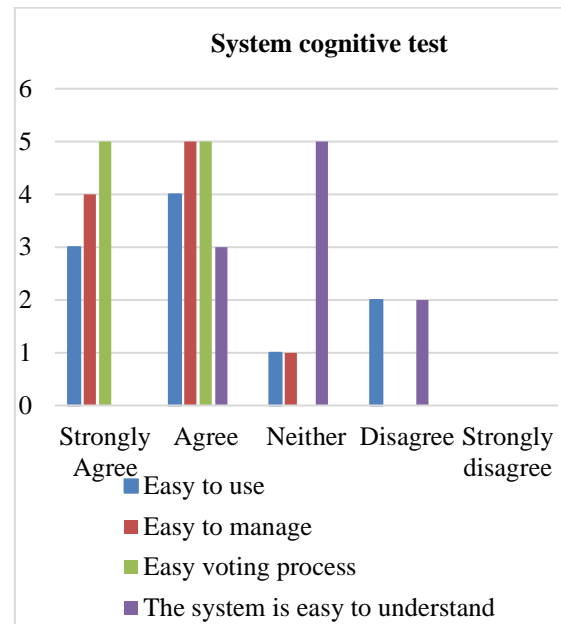**Fig. 11 Cognitive dimension test chart**

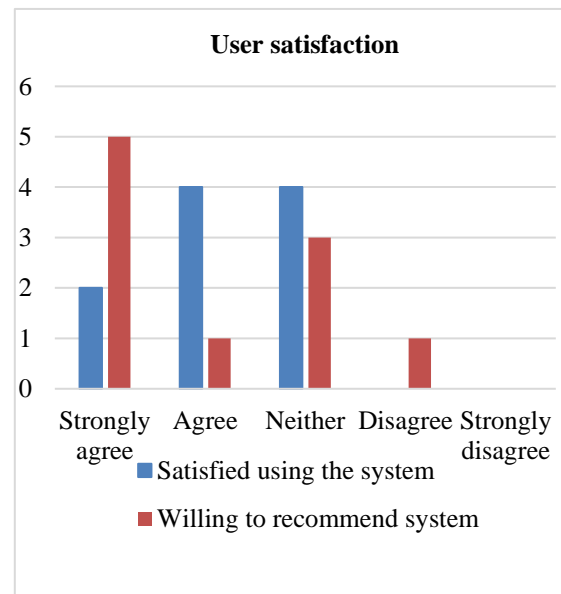

Fig. 12 User satisfaction dimension test chart

Fig. 10 Performance dimension test chart

IV. CONCLUSION

Based on a series of research steps, analysis, design, implementation, and testing of software requirements negotiation systems using the E-Voting method as decision support, we have described all of which in this paper, the following conclusions are: Based on the data analysis process, 82.4% of stakeholders agreed to negotiate through the web and the system testing process, the web application forum for the software requirements

negotiation system can be applied as a forum for communication and negotiation of stakeholders comprising developers and clients. The system is used to discuss it and understand each other's needs based on the recommendation for a forum with a particular software theme.

The E-Voting method can be applied properly as a decision-making method of negotiating software requirements, based on the results of data analysis which states that 76.5% of respondents choose to vote as group decision support, and also where the results of applying to vote for negotiations are an absolute decision of choice. And the agreement between the two stakeholders and the results got as reports, to reduce the possibility of failure of the developer, both on a small scale (correcting errors, performance, functionality) and a large scale (loss of trust from clients, orders, reduced profits). This is also because we will develop later the materials and needs that result from a negotiated agreement between the two parties. In addition, based on the User Acceptance Test that the author has done, the software requirements negotiation system that uses the E-Voting method has a high success rate based on 3 test dimensions, namely Performance (80.5%), Ease (78.5%), and User Satisfaction (78%).

ACKNOWLEDGEMENT

REFERENCES

[1] S. Ahmad, "Measuring the Effectiveness of Negotiation in Software Requirements Engineering," *Sch. Comput. Sci. Softw. Eng.*, pp. 1–203, 2012.

[2] S. Ahmad, "Measuring the Stakeholders' Agreement Level in Negotiation through Experiment," *Imecs 2009 Int. Multi-Conference Eng. Comput. Sci. Vols I Ii*, vol. I, pp. 1069–1074, 2009.

[3] A. Aurum and C. Wohlin, *Engineering and Managing Software Requirements*. 2006.

[4] S. Ghosh, A. Dubey, and S. Ramaswamy, "C-FaRM: A collaborative and context aware framework for requirements management," *2011 4th Int. Work. Manag. Requir. Knowledge, MaRK'11 - Part 19th IEEE Int. Requir. Eng. Conf. RE'11*, pp. 29–30, 2011.

[5] W. H. El-Ashmawi, D. S. Abd Elminaam, A. M. Nabil, and E. Eldesouky, "A chaotic owl search algorithm based bilateral negotiation model," *Ain Shams Eng. J.*, vol. 11, no. 4, pp. 1163–1178, 2020.

[6] E. Masero, M. Francisco, J. M. Maestre, S. Revollar, and P. Vega, "Hierarchical distributed model predictive control based on fuzzy negotiation," *Expert Syst. Appl.*, vol. 176, no. January, 2021.

[7] E. Knauss *et al.*, "T-Reqs: Tool support for managing requirements in large-scale agile system development," *Proc. - 2018 IEEE 26th Int. Requir. Eng. Conf. RE 2018*, pp. 502–503, 2018.

[8] X. Zhu, H. Zhu, D. Liu, and X. Zhou, "Criteria Making in Role Negotiation," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 50, no. 10, pp. 3731–3740, 2020.

[9] M. Kolar, C. Fernandez-Gago, and J. Lopez, "A model specification for the design of trust negotiations," *Comput. Secur.*, vol. 84, pp. 288–300, 2019.

[10] B. Shojaiemehr, A. M. Rahmani, and N. N. Qader, "Automated negotiation for ensuring composite service requirements in cloud computing," *J. Syst. Archit.*, vol. 99, no. February, p. 101632, 2019.

[11] G. A. Pradipta, R. Wardoyo, A. Musdholifah, and I. N. H. Sanjaya, "Improving classifiaction performance of fetal umbilical cord using combination of SMOTE method and multiclassifier voting in imbalanced data and small dataset," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 5, pp. 441–454, 2020.

[12] V. Agate, A. De Paola, P. Ferraro, G. Lo Re, and M. Morana, "SecureBallot: A secure open source e-Voting system," *J. Netw. Comput. Appl.*, vol. 191, no. May, p. 103165, 2021.

[13] A. Petitpas, J. M. Jaquet, and P. Sciarini, "Does E-Voting matter for turnout, and to whom?," *Elect. Stud.*, vol. 71, no. February, p. 102245, 2021.

[14] S. I. Khaleel*, "Designing a Tool to Estimate Software Projects Based on The Swarm Intelligence," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 4, pp. 524–537, 2021.

[15] E. R. Subhiyakto and Y. P. Astuti, "Design and development meeting schedule management application using the rad method," *Proceeding - 2019 Int. Conf. Artif. Intell. Inf. Technol. ICAIIT 2019*, pp. 60–64, 2019.

[16] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, "A Different Approach on Automated Use Case Diagram Semantic Assessment," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 1, pp. 496–505, 2021.

[17] E. R. Subhiyakto and D. W. Utomo, "Software Testing Techniques and Strategies Use in Novice Software Teams," *J. Sist. Inf.*, 2016.