

Defining and Prioritizing Software Requirement Using gIBIS and AHP Method

Falahah

Department of Informatics, Widyatama University, Indonesia

Copyright©2019 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract Requirement engineering is an important process in a software development project. During the process there are two activities that are very significant that determine the success of the software development: requirement definition and requirement prioritization. Through this process we can set the scope of the software product and the limitations of product features for each released version. Defining and prioritizing requirements can be done using same approach. gIBIS (graphical Issue-Based Information System) is a proposed approach for the design rationale to describe the reasoning process for the solution that is chosen in order to answer a problem or issue. The graphical representation of the reasoning process using gIBIS can be supported by tools such as Compendium. This paper proposes the implementation of the gIBIS approach in defining software requirements by assigning a score for each alternative and selecting the highest score as a solution. The approach is combined with AHP (Analytical Hierarchical Process) method to set the priority and as the result we can determine the software requirements and also rank the requirements based on its priority. The criteria used in setting the priority include cost, risk, difficulty and stability. The proposed approach (combining of gIBIS and AHP) is demonstrated with a case study of prioritizing requirements for a website development of "IndoSoccerSchool". The result show that the proposed approach can help us generate a list of requirements and give them a score based on priority and the list can give a reference for developing the software based on credible priorities and then manage the software release.

Keywords Requirement Engineering, gIBIS, AHP, Priority

1. Introduction

Software requirement can be defined as the

functionalities that should be provided by the software so it can meet the needs of stakeholders or users that will use the software. Requirement engineering is a series of activities in identifying and defining the requirement. Requirement engineering consists of activities such as eliciting the requirement, identification, analysis, validation and requirements management (Sommerville, 2010). Analysis of requirements can consist of activities such as defining and prioritizing the requirements.

Requirement analysis in software requirement management is very important, because during the software development process, changes in requirements often occur due to several causes, such as change in organization environment, change in user's priority, misinterpretation in early stages, different view of the business process, and so on. Conflict can happen when change of requirements must be completed in the same original timeline or the requirements are in conflict each other (Davis, 2013). In this case, the software developer needs a method to set the requirements priority.

Prioritizing requirements is a decision making process. During this process, the software developer needs to identify alternatives based on considerations such as cost, impact, importance, and relation with other requirements (Khari and Kkumar, 2013). The developer also needs to negotiate with other stakeholders to determine the chosen alternatives because an alternative may have significant impact to the stakeholder. So, requirement prioritization is not a standalone decision making process and it would be better if supported by good structured approach and method.

When negotiating with the stakeholder, the software developer usually needs a reasoning process to explain why they choose the alternative. Explanation of the reasoning can be achieved by a textual description, a schematic diagram or a graphical visualization. The graphical usage is often regarded as more familiar for the user because the idea can be presented better and the audience can understand faster. gIBIS (graphical Issue-Based Information System) has been recognized as a tool for the

graphical description of the reasoning process in design rationale (Liu, Chandra and Barnes, 2014). On the other side, AHP (Analytical Hierarchical Process) is well known as a model for decision making to choose the solution based on multiple criteria (Alexander, 2012).

Based on the description above, this paper proposes an approach in defining and prioritizing requirements using gIBIS reasoning techniques combined with the AHP method. The proposed approach is then implemented in the development of an "IndoSoccerSchool" portal, a web based application which supports the soccer school community in Indonesia. The portal will be used as an integrated web site for soccer schools from different areas or different sports clubs to promote their program, achievement, team, and facilitate the user to communicate with the soccer school management.

2. Literature Review

2.1. Requirement Management

The first step in software development is to define the requirement. Davis (Davis, 2013) defined the requirement as an externally observable characteristic of a desired system. The requirements needs to pass two criteria to prove validation, which are the requirement must be observable from the point of view of external perspective, and the requirement must help satisfy some need of stakeholders.

Davis (2013) explains that requirement management is the set of activities that are divided into three subsets: requirement elicitation, requirement triage, and requirement specification. Requirement triage is the activity to analyze the requirement in the context of the resource available, time to market, revenue goals, and return on investment. In this phase the project leader should determine the priority of the requirement in order to analyze the cost and benefit of selected requirement. The process of managing requirements becomes more practical and easier when using software tools. Zainol and Mansoor (2012) mention the benefits of using requirement management tools such as follows to: manage version and change, store requirements attributes, facilitate impact analysis, track requirement status, control access, communicate with stakeholders and reuse requirements.

2.2. Requirement Prioritization

The quality of a software product is often determined by the ability to satisfy the needs of the customers and users (Bertnsson-Svensson, 2010). Hence, eliciting and specifying the correct requirements and planning suitable releases with the right functionality is a major step towards the success of a project or product. Most software projects have more candidate requirements than can be realized

within the time and cost constraints. Prioritization helps to identify the most valuable requirements from this set by distinguishing the critical few from the trivial many. The process of prioritizing requirements has some advantage such as to decide the core requirements for the system, to plan and select an ordered, optimal set of software requirements for implementation, to trade off desired project scope against sometimes conflicting constraints, and many more (Berander and Andrews, 2005). This approach in requirements prioritization is typically applied to determine the requirements or features that should be included or should be implemented first. Prioritization usually includes some criteria such as risk, business value, or implementation cost (Mead, 2006). Each aspect can breakdown again into more detailed aspects. For example, risk can be sub-divided into business risk, process risk, and schedule risk (McManus, 2012). Often, the aspects interact and changes in one aspect could result in an impact to others. Aspects are usually evaluated by stakeholders in a project. The selection criteria of prioritization are usually rated by different stakeholders or automatically determined using others information (Calienes, 2013). For each criteria, ranking for importance can be done by assigning descriptors to the requirements using some point of consideration such as cost, risk, difficulty, ROI and many more. Stability can be measured by identifying the requirements that will most likely change, and identify the potential risk based on requirement descriptor (Wong, 2015).

The prioritization can be achieved with various measurement scales and types. The least powerful prioritization scale is the ordinal scale, where the requirements are ordered so that it is possible to see which requirements are more important than the others. The ratio scale is more powerful since it is possible to quantify how much more importance one requirement is than the others (by assign the percentage 0-100) (Riegel & Joerg, 2015). Numerous method in requirement prioritization exists such as the analytic network process (ANP), analytic hierarchy process (AHP), hierarchy AHP, spanning tree matrix, bubble sort, binary search tree and priority group (Sadiq, Shadid, and Ahmad, 2010). Saaty developed AHP as a method for decision making using multiple objectives. Joachim Karlsson and Kevin Ryan implemented AHP method in software engineering (Mead, 2006). ANP is the generalization of AHP that deals with dependency and feedback among clusters (Calienes, 2013). Among these methods, AHP is suggested as the most suitable method for requirements prioritization, because it provides efficient results on a ratio scale, has fault-tolerance and also provides consistency checks. The preconditions should be prepared before using AHP as a prioritization method are firstly that the stakeholders (user, developer or system analyst) have to understand the requirement clearly, and secondly know the interconnections between requirements and criteria (Khan, et.al, 2015).

2.3. Design Rationale with gIBIS

Reasoning is a process of decision making that allows us to explain the reason behind the solution we had chosen. Design rationale (DR) is the reasoning and argumentation that needs to be done during the design process. DR systems are intended to support people in the design process that can allow designers to share structure and record their thought process (Horner and Atwood, 2006).

The argumentation structure in DR is designed to provide a natural framework in which the designer can reflect on decisions. This structure can help focus the search for design alternatives, making cognitive processing more effective (Schubanz et al, 2015). Although designers cannot consider all possible alternatives, if the rationale is recorded, maintainers will be better able to identify which ideas were deliberated upon. Recording DR allows the opportunity for people to perform a post-hoc analysis of the design decisions.

The problem with DR is how to present the argumentation in a well-formatted structure. Rittel developed IBIS (Issue Based Information System) as an approach to structure the argumentation process (Kunz, Wenner, and Rittel, 1979). The IBIS method is based on the principle that the design process for complex problems is fundamentally a conversation among stakeholders in which they bring their respective expertise and view point to the resolution of the design issue. Each issue can have many positions, where position is defined as a statement or assertion which resolves the Issue. Each of an Issue's positions, in turn, may have one or more arguments which either support that position or object to it. Thus, each separate issue is the root of a (possibly empty) tree, with the children of the Issue being positions and the children of the position being argument. Figure 1 show the structure of argumentation in IBIS (Ebadi, 2009). As shown in figure 1, the conceptual relationship between issue, argument and position is connected by some activity such as position → responds-to → issue, or issue → is suggested by argument.

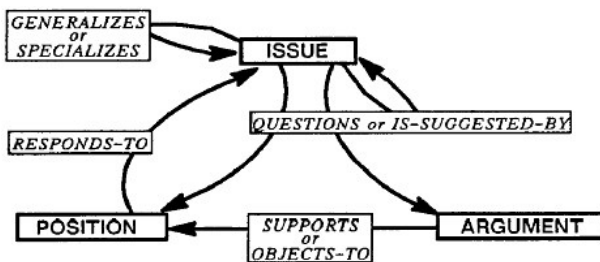


Figure 1. Structure of argumentation process in IBIS (Kunz, Wenner, and Rittel, 1979)

According to Liu, Chandra and Barnes (2014), IBIS is used in issue mapping, an argument visualization technique related to argument mapping. It is also the basis of a facilitation technique called dialogue mapping. Along with

the improvement in computer technology, attention on and use of the IBIS reappeared in the form of computer-based IBIS - type system, which applies the concept of IBIS in the form of graphic visualization, and is referred to as gIBIS. gIBIS can be used to describe the dialogue as an image featuring argument, issues and positions in the form of connected symbols, as well as to facilitate tracking of the argument.

Several other graphical IBIS-type systems were developed once it was realized that such systems facilitated collaborative design and problem solving. These efforts culminated in the creation of the open source Compendium (software) tool which supports - among other things - a graphical IBIS notation. Compendium is a computer program and social science tool that facilitates the mapping and management of ideas and arguments (Sierhuis, 2006). The software provides a visual environment that allows people to structure and record collaboration as they work through "wicked problems". The software is currently released by the not-for-profit Compendium Institute. The current version of Compendium software supports modeling through gIBIS. The features of compendium in graphical modeling of argument tracking can be implemented as a tool to define alternatives in requirements.

2.4. Analytical Hierarchy Process (AHP)

The Analytic Hierarchy Process (AHP) is a systematic decision-making method that has been adapted for prioritization of software requirements. It is conducted by comparing all possible pairs of hierarchically classified requirements, in order to determine which has higher priority, and to what extent (usually on a scale from one to nine where one represents equal importance and nine represents absolutely more important). The total number of comparisons to perform with AHP are $n \times (n-1)/2$ (where n is the number of requirements) at each hierarchy level, which results in a dramatic increase in the number of comparisons as the number of requirements increases (Bunruamkaew, 2012).

In its original form, the redundancy of the pair-wise comparisons allows a consistency check where judgment errors can be identified and a consistency ratio can be calculated. When reducing the number of comparisons, the number of redundant comparisons is also reduced, and consequently the ability to identify inconsistent judgments is also reduced. When using other techniques, a consistency ratio is not necessary since all requirements are directly compared to each other and consistency is always ensured. Some studies indicate that persons who prioritized with AHP tend to mistrust the results since control is lost when only comparing the requirements pair-wise. The result from a prioritization with AHP is a weighted list on a ratio scale.

3. Requirement Definition and Prioritization Using Compendium and AHP

3.1. Problem Definition

The problem of requirement definition and prioritization is still topic of interest in software engineering research. Although there are many methods already proposed and implemented, it is still open for new potential approaches to inform an alternative approach in requirement definition and prioritization. The process for requirement definition and prioritization consists of two steps, first is requirement definition and second is prioritization. These two processes will become the main discussion in this paper. The problem statement is:

1. How to define or generate the requirements options using gIBIS approach?
2. And then, how to implement AHP as an approach for the prioritization of the requirements?

The definition and generation of requirement alternatives will be achieved through gIBIS approach utilizing Compendium. After it produces alternatives including the score to reduce the alternatives, the selected alternative will assign as an input into prioritizing process using AHP approach.

3.2. Requirement Definition Using Compendium

DR process using compendium is an interesting presentation of the idea but the method did not provide a way to choose using the options based on compendium

presentation. This paper proposed the approach to choose the option on DR process using Compendium.

The approach is based on simple assumptions: it is easier for us to quantify the option if we assign the score for each alternative, so we can argue why we choose the solution. The approach is using the sign of node in compendium as a score of an option. As describe above, there are three types of node to support an option, con as negative node, pro as positive node and argument or idea as neutral node. If we put the score for each option, by summarizing the total score of node then we can scoring the alternative, and then give us the reason to choose an option, which has the biggest score or has positive final score. Figure 2 shows an example of an argument structure in compendium and relation between nodes. The figure shows one issue/problem node which is thus resolved or answered by 5 options. Each option has some nodes which present pros and cons for each option, and also, may generate new problems/issues. We will use the case on figure 2 as an example of implementation of giBIS in DR using Compendium. The problem that will be addressed in this example is to answer the issue “Videophone Marketing Plan?” The alternative solutions for the issue as shown in the figure 2 are: print advertising, press release, radio advertising, trade publication articles, and trade show. Each alternative has their own score such as pro (positive score) or con (negative score) as a part of argumentation process. So for the fifth alternative “trade show”, it has three arguments, two are cons (insider only and expensive) and one is pro (live demo sells). Each argument is represented by node.

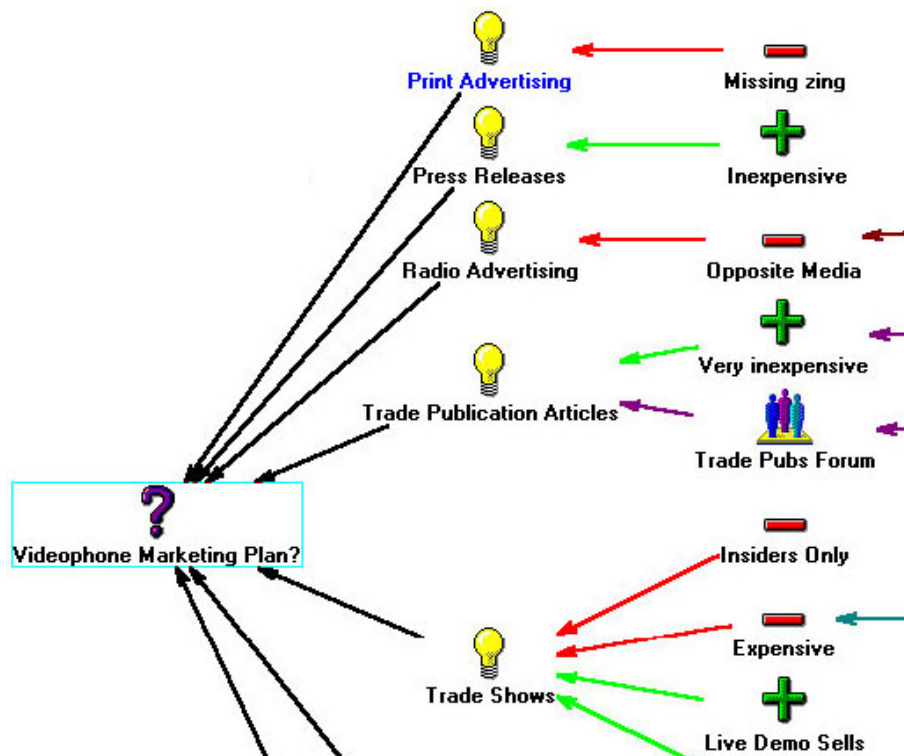


Figure 2. Nodes in Compendium**Table 1.** Example of Mapping Node for Design Rationale, Derived from Compendium Diagram

Issue	No	Option	Statement	Status	Score	Total
Videophone Marketing Plan	1	Print Advertising	Using print advertising to execute marketing plan, but it would not give us a passion	Con	-1	0
			It is quite easy	Pro	1	
	2	Press Release	It is easy and inexpensive	Pro	1	0
			It is not popular in recent trend	Con	-1	
	4	Trade Publication	It is very inexpensive	Pro	1	1
Need trade pubs forum			Issue	n/a		

After generating some alternatives and recording arguments for each option, then each option is written in a table as described in Table 1. We also can add the statement or description for each option. For each argument which supports the alternative, we assign score -1, 1 for positive and negative node respectively. If there is new issue or problem arising, we assign the note "n/a". The last column show the calculation of total score for each alternative, and then we can choose the biggest one as a solution. In this example, we have one candidate alternative as a solution. We would not choose the first and second option because the total score is zero. The best alternative we choose is the fourth option.

In the context of requirement definition, this approach can help us to determine which requirement we want to explore, which option to potentially ignore because it has a negative score and which requirements we decide to define with detailed requirement. It is the first step in requirement definition and selection.

3.3. Requirement Prioritization Using AHP Method

After defining some requirements and ensuring that each requirement can be explained in a structured manner using the DR approach by mapping in compendium and calculating the score, the next step is to choose which requirement will have the highest priority. This goal can be achieved by implementing a decision model approach such as AHP. With a short description, of each requirement we have defined, we can use it as a criteria in a AHP model.

The AHP modeling consist of the steps as described below:

Step 1. Develop the weights for the criteria by:

- Developing a single pair-wise comparison matrix for the criteria;
- Multiplying the values in each row together and calculating the n th root of said product;
- Normalizing the aforementioned n th root of products to get the appropriate weights;
- Calculating and checking the Consistency Ratio (CR).

Step 2. Develop the ratings for each decision alternative for

each criterion by:

- Developing a pair-wise comparison matrix for each criterion, with each matrix containing the pair-wise comparisons of the performance of decision alternatives on each criterion;
- Multiplying the values in each row together and calculating the n th root of said product; Normalizing the aforementioned n th root of product values to get the corresponding ratings;
- Calculating and checking the Consistency Ratio (CR)

Step 3. Calculate the weighted average rating for each decision alternative.

Step 4. Choose the one with the highest score.

As an example suppose we have 3 criteria that need to be evaluated, which are A1, A2, and A3. Table 2 show the example of pair wise matrix.

Table 2. Example of Pair Wise Matrix using 3 Criteria

	A1	A2	A3
A1	1		
A2		1	
A3			1

Comparisons were made based on the judgement of decision makers to evaluate the importance of the element relative to the other elements. The pairwise comparison process is started from the top level of the hierarchy is intended to select the criteria, for example A, then take the elements to be compared, e.g. A1, A2, and A3. Then the arrangement of the elements being compared will look like the matrix in Table 3.

Table 3. Importance / Preference Scale

Scale	Degree of Preference
1	Equal importance
3	Moderate importance of one factor over another
5	Strong or essential importance
7	Very strong importance
9	Extreme importance
2,4,6,8	Values for inverse comparison

The first step is to decompose the problem into a hierarchy of criteria and alternatives. Based on 8 quality measures for requirements (Saavendra et al., 2013; Kumar & Cristin 2018), we choose ranked for importance and stability as criteria to set requirement priority.

The importance level can be measure by evaluating the cost, risk and difficulty of the requirement. The cost can be measured using function point (FP) approach to define the size of functional specification which should provide by the system to fulfill the requirements. The rule of thumb is the higher the FP score, the greater the cost of development, because each FP need effort to be provided. The risk can be measured using for example the SERIM method to measure the risk of requirement from the view point of software development project. To measure difficulty we can use a simple scale (derived from criteria such as complexity of a module, number of database accessed and the interface with other system). Actually, difficulty can be

measured implicitly using FP approach, but in this case we just put a simple score 1-3 by discussing with the programmer using expert judgment to evaluate the detail design of each requirement. The goal is to answer how important the requirement is. As mentioned above, we will use 2 criteria, importance and stability. Importance can be measured by three parameters which are cost, risk and difficulty. From the table above, we have 4 factors or criterias for evaluated, which is cost (C), risk (R), Difficulty (D), and Potential to changes (P). Figure 3 show the breakdown of requirement prioritization goals based on two categories of criteria: importance and stability. Importance is divided into three categories which are cost, risk and difficulty, and each category is then divided into three sub categories based on the range for each criteria, which are High (H) (for example high risk, high cost, low cost, etc), Medium (M) and Low (L).

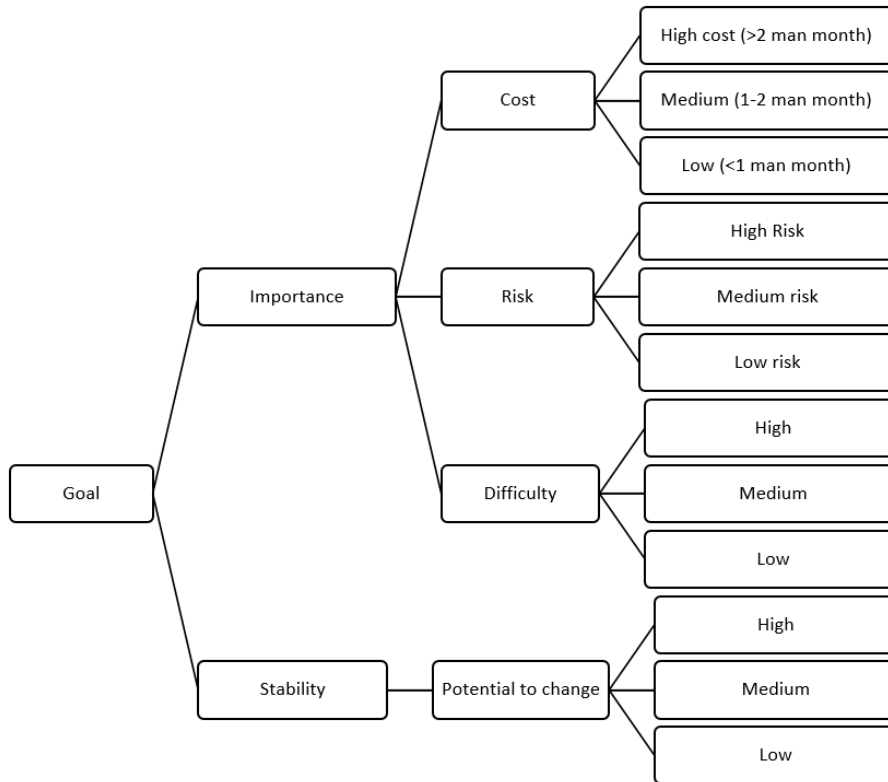


Figure 3. Breakdown of criteria and sub criteria

The next step is to assign a comparative scale of importance. Saaty and Vargas (2000) use the scale as mentioned on Table 3.

Table 4 show details result of a pair wise comparison between 4 criteria: C, R, D and S is stand from Cost, Risk, Difficulty and Stability criteria. The score is assigned based on assumption that risk is more important than cost, and difficulty is more important than risk because the difficulty could increase risk, and stability is the most important criteria, because usually the developer expect to produce software that is relatively stable.

Table 4. Pair Wise Comparison for 4 Criteria in Goal

	C	R	D	S
C	1.00	0.50	0.33	0.20
R	2.00	1.00	0.33	0.25
D	3.00	3.00	1.00	0.50
S	5.00	4.00	2.00	1.00
Total	11.00	8.50	3.67	1.95

Table 5. Normalization, Average and Consistency Measure

	C	R	D	S	Avg	Cm
C	0.09	0.06	0.09	0.10	0.09	0.35
R	0.18	0.12	0.09	0.13	0.13	0.52
D	0.27	0.35	0.27	0.26	0.29	1.18
S	0.46	0.47	0.55	0.51	0.50	2.02
Total	1.00	1.00	1.00	1.00		

The next step is calculating the normalization matrix, by

Table 6. Weighted Score for Sub Criteria

Criteria	Priority	Score	Sub Criteria	Priority	Score
Cost	4	0.09	High cost(>2 man month)	3	0.05
			Medium	2	0.14
			Low	1	0.23
Risk	3	0.13	High Risk (>impact to change more than 70% specification)	3	0.05
			Medium (70% - 30%)	2	0.26
			Low (<30%)	1	0.55
Difficulty	2	0.29	High (FP>30)	3	0.11
			Medium (FB between 10-30)	2	0.32
			Low (FP<10)	1	0.62
Stability	1	0.50	Possibility to change > 70% (70%-30%)	3	0.04
				2	0.17
				1	0.33

dividing the value of each cell with the total for each column. The result is shown in table 5.

Consistency measure (Cm) is calculated from pair wise matrix to average column as in example below: For row C: $Cm = (1*0.09)+(0.5*0.13)+(0.33*0.29)+(0.2*0.47) = 0.35$

To examine the consistency, we can calculate the consistency index (CI) using the formula

$$CI = (t-n)/(n-1) \tag{1}$$

where is :

$$t = (1/n) \sum (*) \tag{2}$$

Based on Random index table of CI for i=4, the value for RI(4) is 0.9, we can calculate the consistency ration (C-Ratio):

$$CR = CI/RI(4) \tag{3}$$

Hence we get the result:

$$T = 4.06$$

$$CI = 0.02$$

$$RI(4) = 0.90$$

$$CR = 0.02$$

Because the value of CR is ≤ 0.1 , we can conclude that the pair wise matrix is consistent so we can use the average (avg) as weighted score for each criteria.

The process is done repeatedly for each sub criteria so we have the weighted score for each sub criteria as describe in table 6.

For each requirement, we then set the assessment for sub criteria. For example, for requirement R1, for sub criteria Cost, if we fulfill the requirement it will cause the high value of Function Point which consequently will raise the cost, so we assign the “H” (high) as value for sub criteria cost, it refers into “high risk cost”.

4. Case Study

The approach proposed above i.e. requirement definition using gIBIS and the process of choosing alternatives using Compendium will produce a defined requirement. The next step is how to prioritize the requirements using AHP. The proposed method, combining the result of gIBIS into decision making model through AHP model, will utilized to process a requirement definition of IndoSoccerSchool, a portal which provide information for young soccer player schools in Indonesia. The background for developing this portal is because of the lack of information about soccer schools in Indonesia. Despite some famous soccer schools already developing their own sites, most soccer schools do not have suitable human resource and ability to create well formatted sites to publish their program, achievements and their players. The portal also serves as a communication channel between the community members and in evolution, will provide a database for young soccer player to record their performance and portfolio.

The case study that we will examine is restricted into two steps. The first is how to use the gIBIS method in

generating the option for the issue that is to be addressed and the second is how to implement the AHP approach on prioritizing the requirements. For the first step, we will choose one issue as a case study: define the channel that will be used to communicate with the user. The second step starts with the assumption that we already have defined some requirements, and need to prioritize them. The scope of requirements that will be discussed in the second step is what kind of information should be displayed by the system and what information can be given by the user to the system.

4.1. Resolving Issues in Requirement Definition Using gIBIS

The issue that will be addressed in this case study is: what kind of channel we want to use to communicate with the user? We explore three ideas as alternatives solution, which are: email, discussion forum or SMS (short message services) gateway. We also discuss the pros and cons of each alternative and then mapping the argumentation into compendium diagram as shown in figure 4. To calculate the score for each alternative, we summarize the map into a table as shown in Table 7.

The chosen alternative is the third, the system needs an SMS gateway to communicate with the user. This decision raised the new issue: where is the location of the server? We also need to define some alternatives to resolve this issue. Based on mapping node above, we can construct the mapping process using Compendium as shown in figure 4.

Table 7. Mapping Node for Issue “What channel we want to use to communicate with the user?”

Issue	No	Option	Statement	Status	Score	Total
What channel we want to use to communicate with the user?	1	Email	Easiest way	pro	1	-1
			Not interactive	Con	-1	
			Rarely access by the user	Con	-1	
	2	Discussion group	Lots of work	Con	-1	-3
			Need moderator to filtering the threat	Con	-1	
			Potential to arise conflict between the users	Con	-1	
	3	SMS	Where is the gateway location of the server?	N/a	0	0
			Need additional hardware	Con	-1	
			Familiar for most user	Pro	1	

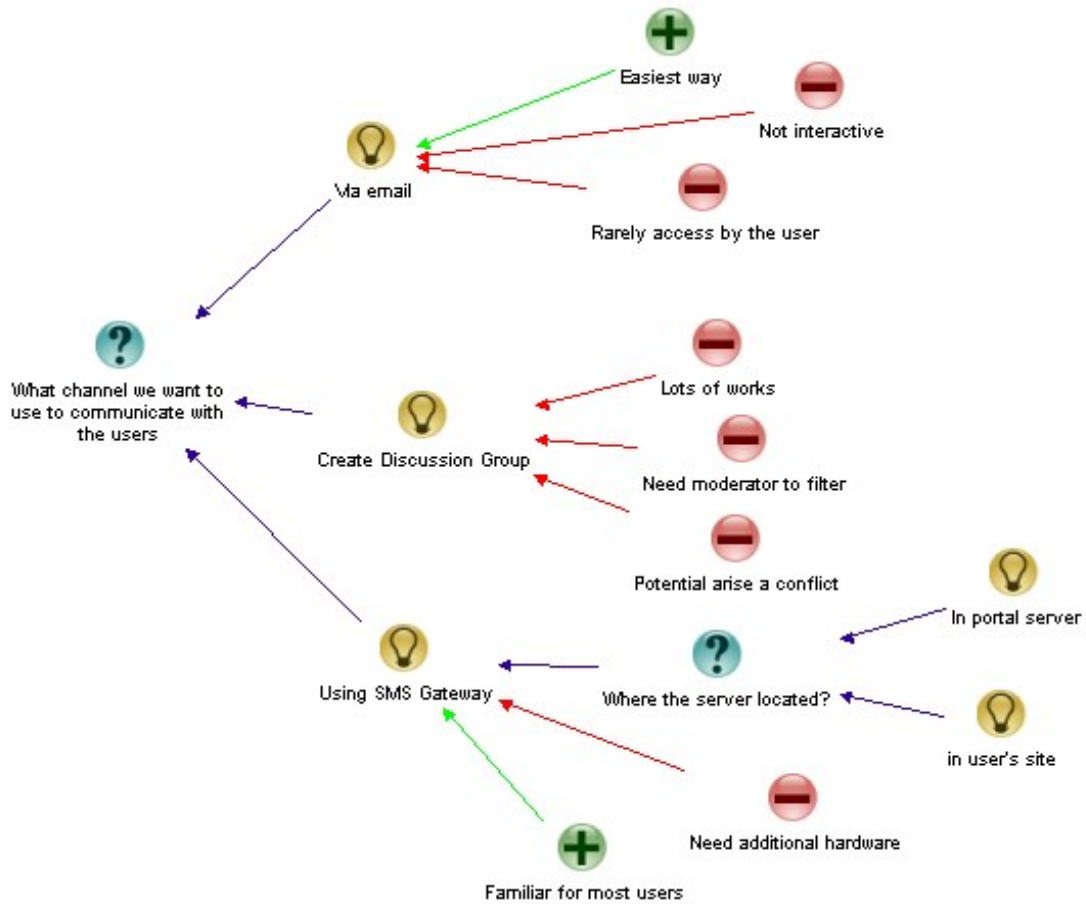


Figure 4. Mapping DR to resolve the issue

4.2. Prioritizing Requirement Using AHP

We assume that after doing some processing on the requirement definition, we can define some requirements for the IndoSoccerSchool, namely requirements of what kind of information should be able to be displayed by the system and what information can be given by the user to the system. Table 8 shows the example of defined requirements that would be an input for prioritizing requirement using AHP.

The scores from table 7 are then used to assess the defined requirement to set the priority. For example, requirement R1 “Users can register themselves into the site”, two criteria is assigned as “high” which is Cost and Risk, “medium” for Difficult and “low” for stability. From table 7, the score for sub criteria H (high) for criteria C (cost) is 0.05. Also, from Table 7 we can see that the weight for C is 0.09. The final score for sub criteria Cost is: Score = 0.05 x 0.09 = 0.005

We do the same way for sub criteria R, D and S, and after calculate the total score we got the value for Requirement

R1 is 0.27. In this case, the higher the score, the higher is importance of requirements. Table 9 show the score calculation for each requirement.

Table 8. Example of Defined Requirements

Req.No	Description
R1	Users can register themselves into the site
R2	System can display the soccer school by regional area, both text and graphic
R3	User can enter the tournament event information
R4	Users can search and sort the event by date
R5	System can record the team member information and their portfolio
.....
....
Rn

Table 9. Score Calculation for Requirements

Req	Sub Criteria assessment			Weight	score	Total	Rank
R1	C	H	0.05	0.09	0.004	0.268	1
	R	H	0.05	0.13	0.007		
	D	M	0.32	0.29	0.093		
	S	L	0.33	0.50	0.165		
R5	C	H	0.05	0.09	0.004	0.238	2
	R	M	0.26	0.13	0.033		
	D	L	0.62	0.29	0.18		
	S	H	0.04	0.50	0.021		
R4	C	H	0.05	0.09	0.004	0.151	3
	R	M	0.26	0.13	0.033		
	D	M	0.32	0.29	0.093		
	S	H	0.04	0.05	0.021		
R6	C	L	0.23	0.09	0.019	0.106	4
	R	M	0.26	0.13	0.033		
	D	H	0.11	0.29	0.032		
	S	H	0.04	0.50	0.021		
Rn						

Based on the requirement priority assessment we can then define the list of requirements as describe in Table 10

Table 10. Requirement Priority Assessment

Req.No	Description	Rank
R1	Users should be able to register themselves	1
R2	System can inform the competition event by sms	5
R3	System can record the summarize of event	8
R4	Users can search and sort the event by date	2
....
....
Rn	N

5. Result and Discussion

So far, we have selected the requirements using gIBIS DR approach and then analyzed the priority of requirements using the AHP methods. The process that we proposed in this paper can be divided into two main processes as describe in figure 5.

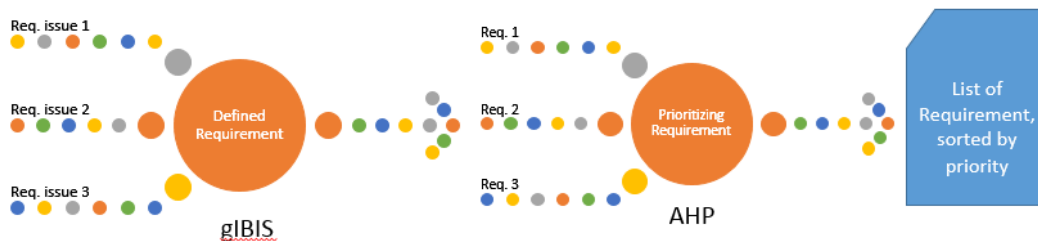


Figure 5. Processes of Requirement definition and prioritization using gIBIS and AHP

Based on the study case above, we can see the benefit of using gIBIS on resolving each issue in the defining requirement process, because this approach can represent the DR process, by showing graphically the reasoning process in selecting the solution from some alternatives. Visual representation can help us select the solution objectively, measurably and traceably, based on our reasoning process.

We can also implement the AHP in prioritizing requirements. The first step in implementing AHP in requirement prioritization is defining the criteria and to score the importance criteria. To set the importance score, we need to adjust it rationally or complete some specific research, such as asking the opinion of developers or users about the impact of important aspects. In this research we only use 4 criteria, which are based on practical considerations due to the limited time and cost of this research. The result would be more interesting if we can add some other criteria such as based on Quality model from IEEE 830:2009 (correct, unambiguous, complete, consistent, verifiable, traceable, modifiable, and ranked for importance and stability).

The results of the priority setting process will help us to focus on higher priorities to be fulfilled in the early stages of software release. In this case, the next step is designing and developing the IndoSoccerSchool portal based on defined requirements.

5.1. Design of Indo Soccer School Portal

After defining the requirement, the next step is to design the portal that can fulfill the requirements. The figures below are the screen shoot of IndoSoccerSchool Portal that was designed based on requirements R1, R2 and R3. Figure 6 is the realization of Requirement R1 (User can register themselves into the portal). The screen is the registration form for Soccer Schools that want to be included in this portal.



Figure 6. Registration form (Req R1)

Figure 7 show the sms interface for news and events. This is the solution of the issue “What channel we want to use to communicate with the user?” as the result of DR process using gIBIS approach. We define this requirement as R2.



Figure 7. SMS for news and event (Req R2)

Figure 8 show the summary of the event as the result of requirement R3 “System can record the summarized of event”. The record will display the winner of tournament, top scorer and best player.

Event Final Result	
EventId	U12_001
Name	LIGA ANAK NUSANTARA
Date	22-24 Aug 2013
Total team participants	43
The Winner	SSB ABCD
Runner Up	SSB XXX (Team B)
Top Scorer	Udin Sedunia
Best Player	Jaenudin Zidane

Figure 8. Summary of the event (Req R3).

Prioritizing requirements using the AHP method can give the developer reasonable guidance in setting what features that need to provided first and can help us to make appropriate release versions of the software features.

6. Conclusions

Based on the discussion above, we can conclude that the prioritization of requirements is an important stage in the requirement engineering process. By determining priority,

we can manage resources better and allocate sufficient resources to fulfill the priority. Setting the priority gives us the strategy for release of the software in several versions, which is the means that features can grow incrementally. In addition, defining requirements also is a process that involves reasoning. During requirement elicitation, we can do some research for the design rationale (DR) that gives us the reasons to choose a selected requirement. gIBIS is a tool for visualizing the process of design rationale which can display the process of decision making in determining the requirement. By graphic visualization, gIBIS can help us to explain the reason behind the selected requirement to the customer or stakeholders. Compendium is software that adopts the gIBIS approach to visualize the reasoning process. Compendium can help us to choose the most appropriate solution easily, by adding the score for each alternative, so we can decide to choose the highest score as the best alternative or the need to find additional information to support our argumentation.

Prioritizing the requirement can be done using several approaches. AHP is one approach that can be implemented in selecting the requirement. Using the AHP method, we can set certain criteria for some requirements to choose the highest priority (Jabarullah et al., 2019). On this research, we choose 4 criteria that are cost, risk, difficulty, and stability. To define the importance of each criteria, we use expert judgement. The method combines the approach of requirement design rationale using gIBIS with prioritization using AHP method than implemented in requirement definition of "IndoSoccerSchool" portal. The result of selected requirement is listed and sorted by level of priority. The list then can give us a reference to develop the software based on priority and manage the software release.

REFERENCES

- [1] Alexander, M., (2012). Decision-Making using the Analytic Hierarchy Process (AHP) and SAS/IML®, *SESUG 2012: The Proceedings of the South East SAS Users Group*, Durham, NC, 2012.
- [2] Bunruamkaew, W., & Khwanruthai. (2012). "How to do AHP analysis in Excel", University of Tsukuba. Accessed from: http://giswin.geo.tsukuba.ac.jp/sis/gis_seminar/How%20to%20do%20AHP%20analysis%20in%20Excel.pdf date: 10 July 2013.
- [3] Berander, Patrik, and Andrews, Anneliese, (2005), Requirement Prioritization, in *Engineering and Managing Software Requirements*, pp.69-94, Springer Verlag.
- [4] Bertnsson-Svensson, Richard, 2010. Managing Quality Requirements: A Systematic Review", in *EUROMICRO SEAA*, 2010.
- [5] Calienes, G. G. (2013). Requirements Prioritization Framework for developing Green and Sustainable Software using ANP-based Decision Making, *EnviroInfo 2013: Environmental Informatics and Renewable Energies*, 2013.
- [6] Davis, A., (2013), *Just Enough Requirements Management: Where Software Development Meets Marketing*, Addison-Wesley, 2013.
- [7] Ebadi, T., Purvis, M.A., Purvis, M.K.,(2009).A collaborative Web-based Issue Based Information System (IBIS) Framework, *The Information Science Discussion Paper Series, November 2009*, ISSN 117-455X.
- [8] Horner, J., & Atwood, M. E. (2006). Effective design rationale: understanding the barriers. In *Rationale management in software engineering* (pp. 73-90). Springer Berlin Heidelberg.
- [9] Jabarullah, N.H., Shabbir, M.S., Abbas, M., Siddiqi, A.F. & Berti, S. (2019) Using random inquiry optimization method for provision of heat and cooling demand in hub systems for smart buildings, *Sustainable Cities and Society*, 47, 101475.
- [10] Khan, J.A., Rehman, I.U., Khan, Y.H., Khan, I.J., Rashid, S., (2015). Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique. *I.J. Modern Education and Computer Science*, 2015,11, pp.53-59.
- [11] Khari, M., and Kumar, N., (2013), Comparison of Six Prioritization Techniques for Software Requirements, *Journal of Global Research in Computer Science*, Volume 4, No. 1, January 2013.
- [12] Kumar, B. S., & Cristin, R. (2018). A Survey on Efficient Power Management Using Smart Socket and IoT. *Review of Computer Engineering Research*, 5(2), 25-30.
- [13] Liu, X., Chanda, N., & Barnes, E. C. (2012). Software Architecture Rationale Capture through Intelligent Argumentation. In *SEKE* (pp. 156-161).
- [14] McManus, J., (2012), *Risk Management in Software Development Projects*, Routledge, 2012.
- [15] Mead, Nancy, (2006), Requirement Prioritization Introduction, Software Engineering Institute, Carnegie Mellon University.
- [16] Mead, Nancy, (2006), Requirement Prioritization Case Study Using AHP, Software Engineering Institute, Carnegie Mellon University.
- [17] Riegel, N., and Joerg, D., (2015), A Systematic Literature Review of Requirements Prioritization Criteria, in *Requirement Engineering Foundation for Software Quality, 21st International Working Conference*, REFSQ 2015.
- [18] Saaty, T.L. and Vargas, L.G. (2000). *Models, Methods, Concepts and Applications of the Analytic Hierarchy Process*, Boston: Kluwer Academic Publishers.
- [19] Saavendra, R., Ballejos, L., Ale, M., (2013), Software Requirements Quality Evaluation: State of the art and research challenges, *14th Argentine Symposium on Software Engineering*, ASSE 2013.
- [20] Schubanz, M., Pleuss, A., Jordan, H., Botterweck, G., (2014), Guidance for Design Rationale Capture to Support Software Evolution, *Workshop Software-Reengineering and Evolution*, 2014.

- [21] Sierhuis, M., (2006). Compendium, Introduction Slide, Stanford University.
- [22] Shum, S.B., Sierhuis, M., Park, J., Brown, M., (2010). Software Agents in Support of Human Argument Mapping, *Computational Models of Argument, Proceedings of COMMA 2010*.
- [23] Sommerville, I, 2010. Software Engineering, Pearson 2010.
- [24] Wong, B.,(2013), Measuring the Quality of the Requirement Specification Document for an e-Commerce Project, E-Business and Organizations in the 21st Century 102, accessed from <http://epress.lib.uts.edu.au/research/bitstream/handle/10453/2659/20030>. date: 10 July 2013.
- [25] Zainol, A., & Mansoor, S., (2012), Requirements Management Tool Elements for The Malaysian Software Industry, Azida Zainol, Sa'ad Mansoor, *Journal of Information and Communication Technology*, Vol 11, 2012, pp.178-192.