Available online at: https://ijact.in

Date of Submission	15/05/2019
Date of Acceptance	24/07/2019
Date of Publication	31/08/2019
Page numbers	3317-3323(7 Pages)

Cite This Paper: M Yaseen, A Mustapha, N. Ibrahim. (2019). Minimizing inter-dependency issues of requirements in parallel developing software projects with AHP, 8(8), COMPUSOFT, An International Journal of Advanced Computer Technology. PP. 3317-3323.

ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

MINIMIZING INTER-DEPENDENCY ISSUES OF REQUIREMENTS IN PARALLEL DEVELOPING SOFTWARE PROJECTS WITH AHP

Muhammad yaseen, Aida Mustapha, Noraini Ibrahim

Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, 86400 Batu Pahat, Johor, Malaysia

yaseen_cse11@yahoo.com, aidam@uthm.edu.my, noraini@uthm.edu.my

Abstract: Many of the organizations often fail to deliver big size software projects in allocated time and budget. One of the main reason of exceed in time estimation is improper implementation of requirements and their integration as software requirements are inter-related and thus waiting time of requirements for their pre-requisite requirements can delay whole project . As compare to small size software's, requirements of big size software systems such as Enterprise Resource Planning (ERP) are normally implemented by parallel and distributed team members so prioritization of requirements become necessary in parallel development for timely delivery of whole project. In this research work, AHP is efficiently applied to prioritize software functional requirements with minimum time complexity. Differences in time estimation without prioritization and with prioritization shows significance of prioritization with AHP.

Keywords: Requirements prioritization; Functional requirements (FRs); Directed acyclic graph (DAG); Spanning Tree; Analytical hierarchical process (AHP).

I. INTRODUCTION

Functional requirements (FRs) of any software system do not exist in isolation but are inter-related. As implementation of some requirements need other requirements so pre-requisite requirements should be implemented first[1]. According to[2][3][4], prioritization of requirements is necessary before implementation. In the context of this research work, dependency should mean the requirement can't not be implemented in the absence of other requirement[5]. This type of dependency is also known as constraint dependency. This type of dependency represents stronger relationship between two requirements. Requirements traceability is important phase of requirements management and SRS should include all kinds of dependencies because sometimes developers may not know about the nature of these dependencies. Assigning low priority to important requirements can increase waiting time of other requirements in parallel developing projects. This shows important or needed requirements should be implemented first while depended requirements should be given low priority [6][7]. FRs should be prioritized in parallel developing projects so that projects can be delivered to clients in estimated time.

Although a lot of techniques are presented by many of the authors for prioritizing different types of requirements but they are not applied to prioritize FRs that can capture dependencies between FRs in parallel developing projects. AHP is a technique that can be efficiently applied for this purpose. AHP is an organized decision-making method that is intended to compute complex multi-criteria decision problems. In fact, AHP is the utmost frequently discussed prioritization technique within decision making in requirements engineering. The purpose of this research is to apply AHP on FRs and identify timely important requirements [8].

For instance, AHP is suitable technique for prioritizing small set of requirements where it calculates priority of requirements through pair-wise comparisons.AHP is simple in use and yield more accurate results as compare to other suggested techniques [9]. Although AHP cannot be applied efficiently for large size FRs but when the size of requirements is reduced before applying AHP then technique works better with less number of pair wise comparisons. In big software systems like ERP as all requirements are not interrelated, so we can compare only depended requirements and thus in this way AHP can be efficiently used with reduced time complexity.

II. BACKGROUND STUDY

The Analytical Hierarchy Process (AHP) is the most famous, most used and simplest technique for Requirement Prioritization (RP). AHP-based prioritization is performed pair wise by comparing each and every requirement against each other. For n requirements, then $n(n-1)/2$ comparisons will be needed. For example, if the number of requirements is ten, then AHP will perform forty-five times comparisons of the requirements. If the requirements increase in size, so does the processing time. If the requirements size is in thousand, there will be 1000*(1000-1)/2 = 499,500 comparisons, which is both very time consuming and difficult to execute. Because the technique is time consuming, it is not scalable for big requirements due to the pair wise comparisons for every requirement [10].

In one of the research study, AHP technique is applied for prioritization. According to the author, although we assign priorities to FRs but can assign priorities also on the basis of process requirements. This work discusses prioritization of the process requirements by considering both local priority and perspective priority. CBPA framework was develop to prioritize requirements of a different stakeholders from different perspectives and to highlight the key issues in them. Two types of process requirements are considered i.e. from business point of view and from management point of view. Increased profit, lead in competition, reduce cost of development, reduce time to development are business oriented process requirements while within budget, on schedule, high customer satisfaction, increase productivity are management oriented process requirements that are considered and prioritized in the paper. The relationship between different requirements, its prioritization and impact are discussed in the paper in the form of matrix. Apart from process requirements, prioritization of requirements from multiple stakeholder's point of view is also discussed. High priority requirement needs more attention and leads to project success [11].

Another author proposed an enhanced AHP method of improving total time of calculations for pair wise comparisons of the requirements. Using Eigen values and matrix evaluation, the author proposed improved model which suggest solutions for prioritizing requirements. During this method some errors like inconsistency can occur so inconsistency removal method is also explained in the paper through implementing consistency index (CI). The requirements will be arranged in groups called bins in the form of hierarchy. This form of prioritization although be helpful in those cases where requirements are not too much and we need to prioritize with the help of AHP. Number of comparisons will be less as compared to traditional AHP [12].

Intelligent based solution is provided for prioritization of requirements collected from stakeholders by applying machine learning technique first to categorize requirements into similar group, then ANN was applied for further prioritization and then AHP was applied at the end to do final comparisons. In first step, before clustering, stakeholders are requested to prepare requirements, then on the basis of profiles of stakeholders and through expert opinions using ANN, requirements can be prioritized [13].

III. GRAPHICAL REPRESENTATION OF REQUIREMENTS

Our research work proposes a graph-based approach to prioritization of FRs. The inputs are the FRs collected from any sources using appropriate elicitation technique and must be specified in the form of Software Requirement Specification (SRS). In this research, the FRs are represented as alphabets R1, R2, . . . , Rn enclosed in circles as nodes. Figure 1 shows notations used for representing the requirements.

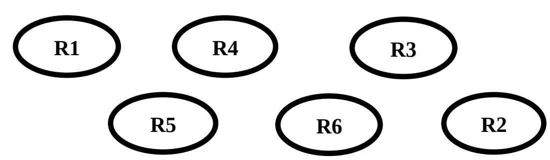

Figure 1: Notations for representing requirements

A. Directed acyclic graph

A graph G = (V;E) consists of a finite set of vertices V and a finite set of edges E. Graphs are useful for the representation of any kind of data in particular sequence.

This research uses directed acyclic graphs (DAG). DAG is a set of ordered pairs of vertices (u; v). The arrows in the graph indicate the dependency of a requirement on another requirement. The requirement generates arrow and points to another requirement indicating that it is necessary or required for another requirement.

Requirements are connected and inter-related with DAG as shown in Figure 2. E.g. R5 is required for R3 and R6 or we can say that both R3 and R6 need R5 for implementation.

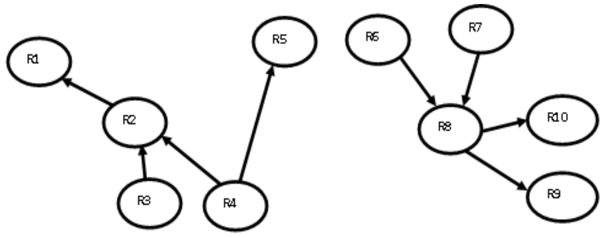

Figure 2: Representing requirements with DAG

B. Adjacency matrix

In graph theory and computer science, an adjacency matrix is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph. For a simple graph with vertex set V, the adjacency matrix is a square |V| × |V| matrix A such that its element Aij is one when there is an edge from vertex I to vertex j and zero when there is no edge. From adjacency matrix as shown in Table 1, we can identify easily which requirement is necessary for other requirements of Figure 2 easily.

Table 1: Adjacency matrix for the requirements of Figure 2

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
R1	0	0	0	0	0	0	0	0	0	0
R2	1	0	0	0	0	0	0	0	0	0
R3	0	1	0	0	0	0	0	0	0	0
R4	0	1	0	0	1	0	0	0	0	0
R5	0	0	0	0	0	0	0	0	0	0
R6	0	0	0	0	0	0	0	1	0	0
R7	0	0	0	0	0	0	0	1	0	0
R8	0	0	0	0	0	0	0	0	1	1
R9	0	0	0	0	0	0	0	0	0	0
R10	0	0	0	0	0	0	0	0	0	0

IV. GRAPHICAL REPRESENTATION OF REQUIREMENTS

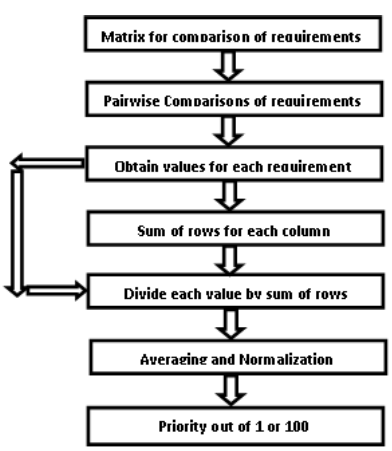

Figure 3: Step by step approach of applying AHP

Figure 3 shows step by step approach of applying AHP to requirements.

As stated, AHP calculate priority of requirements through pair wise comparisons. As from adjacency matrix, we can see that there exist limit number of depended relations in DAG so we don't need to compare all requirements together. In calculating priority of requirements with AHP, the following steps are included.

In first step, we take matrix and start comparing all requirements together. From adjacency matrix, we can see which requirement is needed for other requirements. If one requirement is needed for other requirement, we will put value greater than 1. For this research, we have considered value of 2. This means the pre-requisite requirement priority is taken as two times as compare to requirement for which it is needed while the priority of depended requirement will be half i.e. ½ as compare to pre-requisite requirement. For independent requirements, we have put value of 1. Table 2 shows values for each requirement against other requirements for Figure 3. Table 3 shows rows sum for each column of Table 2. After adding column values for each row, net priority of requirements will be calculated. Table 4 shows how we have calculated net priority of all requirements after averaging over normalized values for each requirement. Priority sum of all requirements will be equal to the number of requirements i.e. 10.

Table 2: Pair wise comparison of requirements

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
R1	1	.5	.25	.25	1	1	1	1	1	1
R2	2	1	.5	.5	1	1	1	1	1	1
R3	4	2	1	1	1	1	1	1	1	1
R4	4	2	1	1	2	1	1	1	1	1
R5	1	1	1	.5	1	1	1	1	1	1
R6	1	1	1	1	1	1	1	2	4	4
R7	1	1	1	1	1	1	1	2	4	4
R8	1	1	1	1	1	.5	.5	1	2	2
R9	1	1	1	1	1	.25	.25	.5	1	1
R10	1	1	1	1	1	.25	.25	.5	1	1

Table 3: Rows sum of comparison values

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
R1	1	.5	.25	.25	1	1	1	1	1	1
R2	2	1	.5	.5	1	1	1	1	1	1
R3	4	2	1	1	1	1	1	1	1	1
R4	4	2	1	1	2	1	1	1	1	1
R5	1	1	1	.5	1	1	1	1	1	1
R6	1	1	1	1	1	1	1	2	4	4
R7	1	1	1	1	1	1	1	2	4	4
R8	1	1	1	1	1	.5	.5	1	2	2
R9	1	1	1	1	1	.25	.25	.5	1	1
R10	1	1	1	1	1	.25	.25	.5	1	1
	17	11.5	8.75	8.25	11	8	8	11	17	17

Table 4: Averaging and Normalization

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Sum / priority out of 10
R1	.058	.043	028	.03	.09	.125	.125	.09	.058	.058	0.705
R2	.12	.087	.057	.06	.09	.125	.125	.09	.058	.058	0.87
R3	.23	.173	.114	.121	.09	.125	.125	.09	.058	.058	1.184
R4	.23	.173	.114	.121	.18	.125	.125	.09	.058	.058	1.274
R5	.058	.087	.114	.06	.09	.125	.125	.09	.058	.058	0.865
R6	.058	.087	.114	.121	.09	.125	.125	.18	.235	.235	1.745
R7	.058	.087	.114	.121	.09	.125	.125	.18	.235	.235	1.745
R8	.058	.087	.114	.121	.09	.0625	.0625	.09	.117	.117	0.919

R9	.058	.087	.114	.121	.09	.03	.03	.045	.058	.058	0.691
R10	.058	.087	.114	.121	.09	.03	.03	.045	.058	.058	0.691

V. ANALYSIS AND DISCUSSION

In above section it is already explained that how AHP can be applied to requirements of DAG. Priority value is assigned to requirements on the basis of its position in graph i.e. how much they are needed and depended on other requirements. Requirements need can either increase breadth wise or depth wise, in either case priority can increase but priority values in both cases can be different.

Similarly priority of requirement reduces if requirements dependency on other requirements increases. The increase priority of requirements with maximum need is associated because maximum requirements wait for it while its priority should be decreased with maximum dependency on other requirements as depended requirements are waiting for their pre-requisite requirements and this can increase the waiting time and delay in projects. Now when AHP is applied, net priority of requirement is calculated by comparing all requirements and for particular requirement net priority value after applying AHP will increases when we put maximum values for it against other requirements. Similarly it will decreases when either we put less values for it against other requirements or either put values less than 1. The net priority will reduces. In order to explain how priority of requirements is associated with AHP, the following scenarios are considered.

Scenario 1: In this scenario we are going to check priority of requirement when its need for other requirements increases breadthwise. Breadthwise contain all requirements on same level with same priority. We take two cases, one with seven requirements and other with five requirements and calculate priorities.

Case 01: In this case, R1 is required for six other requirements with all requirements on same level with same priority as shown in Figure 4. Through AHP, priority of R1 becomes 1.75. The priority of all other requirements is shown in Table 5.

Case 02: In this case, R8 is required for four other requirements with all requirements with same priority on same level as shown in Figure 5. Now priority of R8 is reduced to 1.32 as shown in Table 4.5.

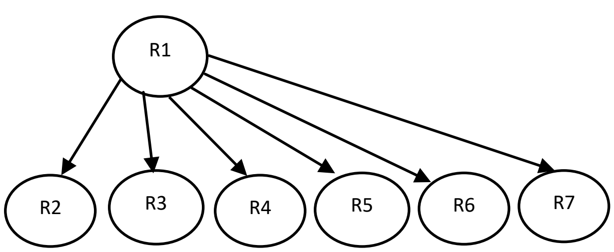

Figure 4: R1 is needed for six other requirements

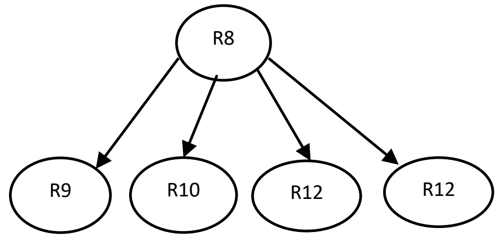

Figure 5: R8 is needed for four other requirements

Priority of requirement increases when its needed increases because of increasing weight score during comparisons e.g. In case 01 we put six values against other requirements for R1 and in case 02, we put four values of 2 each.

In parallel developing projects when requirements are distributed in different team members for development, more needed and demanded requirement should be implemented first as more and more other requirements and developers can wait for these requirements which can delay the project.

Scenario 2: Priority of requirement decreases when dependency of requirement for its implementation increases on other requirements. The reason is that during comparison against other requirements, sum of values are reciprocal of 1 e.g. in Figure 6, priority of R1 against R6 and R7 will be equal to ½. As number of requirements increase, sum of reciprocal values will reduce the priority of requirement. Priority of R1 is now 1 which is minimum as compared to all cases. Priority of other requirements is shown in Table 5 below.

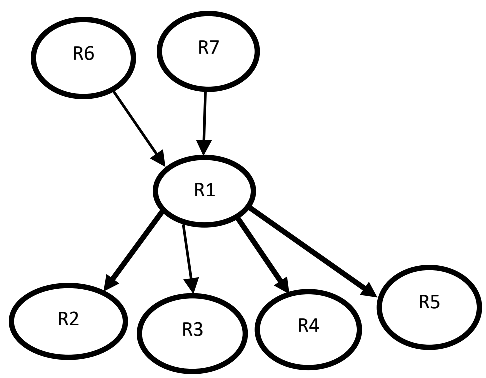

Figure 6: Number of pre-requisite connected with requirement

Deceasing priority of R1 when its dependency on other requirements increases has advantages because it can decrease waiting time. If we implement high depended requirement first as compare to less depended requirements, it can delay the project so it is better to implement fist low depended requirements first.

Table 5: Priority values for Figure 4, Figure 5 and Figure 6

Requirements	Scenario 1		Scenario 2
	Case 1	Case 2	Case 1
R1	1.750	x	1.0
R2	0.875	x	0.5
R3	0.875	x	0.5
R4	0.875	x	0.5
R5	0.875	x	0.5
R6	0.875	x	1.995
R7	0.875	x	1.995
R8	x	1.32	x
R9	x	0.68	x
R10	x	0.68	x
R11	x	0.68	x
R12	x	0.08	x

Similarly if number of pre-requisite requirements and number of requirements for which particular requirement is need are equal then priority of requirement will be almost equal. E.g. in Figure 7, number of backward (depend on these requirements) and forward requirements (needed for these requirements) for R1 are equal, in all these cases priority of R1 will be equal i.e. nearly equal to 1.

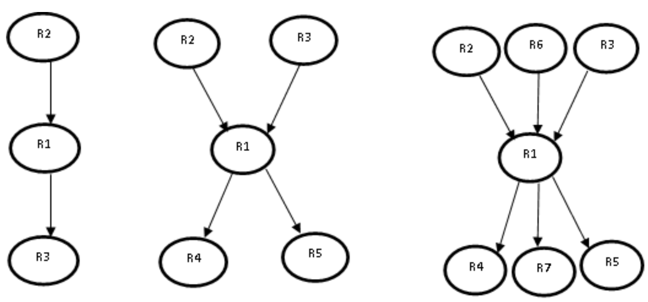

Figure 7: Distribution of requirements with same ratio

If requirement need for other requirements is greater than zero and dependency on other requirements is 0 then priority will be greater than 1 and if dependency of requirement on other requirements is greater than 0 and need is 0 then priority of requirement will be less than 1. If requirement need and dependency as shown in Figure 7 become equal, then priority will become equal to 1. Priority of requirement will be considered 1 when it is totally independent i.e. neither need for other requirements and neither dependent on other requirements for implementation.

If we have two independent requirements then giving preference to requirement with greater priority is better and can reduce delays in projects but if we have two dependent requirements then we must have to implement first the pre-requisite requirement and in this case priority of pre-requisite requirements must be greater than the requirement for which it is needed. AHP will automatically assign pre-requisite requirements with high priority.

VI. EXPERIMENT

In order to validate the significance of AHP during prioritization, experiment was conducted on requirements of mobile phones inventory management system. The presented technique was applied on requirements collected from mobile sales shop as shown in Table 6. In our previous published paper, same dataset was consider in other context[1]. From Table 6, we can see which requirement is required for other requirements.

Table 6: Requirements of mobile shop

Functional Requirement	Notation	Required for	Priority (AHP)	Efforts Required (Time estimation)	Assign Team member
Supplier	R1	R7,R9,R11,R12	1.207	30 hrs.	A
Customer	R2	R8,R10,R13,R14	1.03	30 hrs.	A
Product category	R3	R5	1.9	30 hrs.	A
Company	R4	R5	1.9	30 hrs.	A
Product	R5	R7,R8,R9,R10	1.161	30 hrs.	A
Sale man	R6	R8,R10	1.09	30 hrs.	A
Purchase	R7	R11	0.723	30 hrs.	A
Sale	R8	R13	0.64	30 hrs.	A
Purchase return	R9		0.695	30 hrs.	B
Sale return	R10		0.637	30 hrs.	B
Supplier debit	R11		0.608	30 hrs.	B
Supplier payment	R12		0.887	30 hrs.	B
Customer debit	R13		0.554	30 hrs.	B
Customer payment	R14		0.89	30 hrs.	B
Expenses	R15		1	30 hrs.	B
Employee basic information's	R16		1	30 hrs.	B

AHP was applied successfully on first fourteen requirements that were related with graph. Priority values as a result of AHP are for all these requirements are shown in Table 6. Apart from it, average time estimation of all requirements is taken as 30 hours. Requirements are equal distributed in two team members i.e. A and B as shown in Table 6 such that requirements of developer B are depended on A. Now two different cases are considered for implementing these requirements. In case 01, all requirements of developer A are prioritized while requirements of B are totally un-prioritized i.e. in ascending order of priority. Time estimation of A becomes 240 hours and B become 390 hours. Increase in time estimation of B is due to delay as many of these requirements are depended on A. In case 02, requirements of developer B are fully prioritized. Now in case 02, time estimation of B is reduced to 300 hours. Figure 8 shows time estimation of developer A and B in both of these cases.

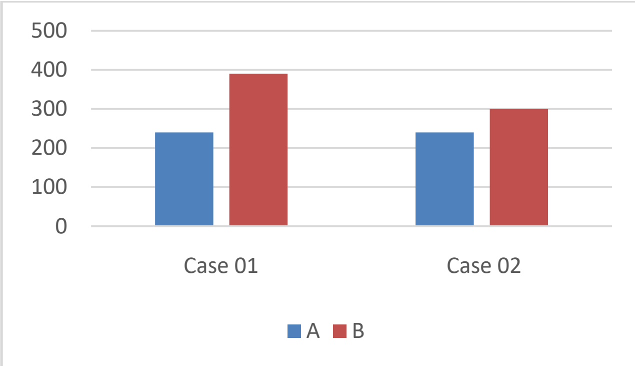

Figure 8: Time estimation of two team members in case 01 and case 02

Total estimation time in both cases will be equal to maximum time taken by any developer to implement all requirements. In case 01, total estimation time is 390 hours and in case 02, total estimation time of project will be equal to 390 hours as shown in Figure9.

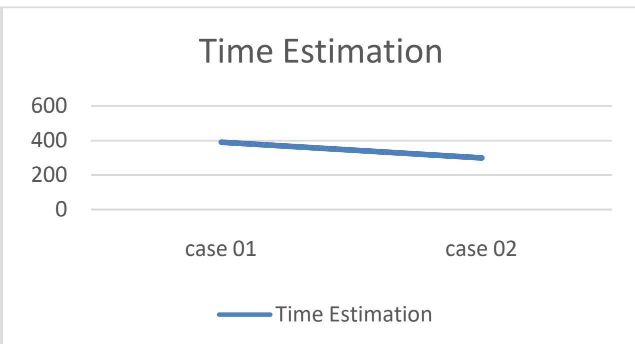

Figure 9: Total time estimation of case 01 and case 0

VII. CONCLUSION

In this research work AHP is applied on FRs to reduce the effect of dependency issues and for timely delivery of parallel developing software projects. It was shown that requirements priority with AHP increases when requirement need increases and it decreases when requirements dependency on other requirements increases. High priority requirements should be implemented earlier as compare to lower priority requirements in order to reduce the waiting time or delay in projects to assure timely implementation and delivery of software projects. AHP was applied on requirements of mobile shop inventory system. Requirements were distributed in two developers such that requirements of one developer were dependent on the requirements of other developer. We have noted a significant difference in time estimation with prioritized requirements and with requirements that were not prioritized.

VIII. REFERENCES

[1] M. Yaseen, N. Ibrahim, and A. Mustapha, 'Requirements Prioritization and using Iteration Model for Successful Implementation of Requirements', *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 1, pp. 121–127, 2019.

[2] M. Yaseen, A. Mustapha, and N. Ibrahim, 'An Approach for Managing Large-Sized Software Requirements During Prioritization', *2018 IEEE Conf. Open Syst.*, pp. 98–103, 2019.

[3] D. E. Perry, H. P. Siy and L. G. Votta, "Parallel changes in large scale software development: an observational case study," Proceedings of the 20th International Conference on Software Engineering, Kyoto, Japan, 1998, pp. 251-260. doi: 10.1109/ICSE.1998.671134.

[4] M. Yaseen, A. Mustapha, and N. Ibrahim, 'Prioritization of Software Functional Requirements : Spanning Tree based Approach', vol. 10, no. 7, pp. 489–497, 2019.

[5] Dahlstedt Å.G., Persson A. (2005) Requirements Interdependencies: State of the Art and Future Challenges. In: Aurum A., Wohlin C. (eds) Engineering and Managing Software Requirements. Springer, Berlin, Heidelberg

[6] P. Carlshamre, K. Sandahl, M. Lindvall, B. Regnell, and J. Natt och Dag, 'An industrial survey of requirements interdependencies in software product release planning', *Proc. IEEE Int. Conf. Requir. Eng.*, pp. 84–91, 2001.

[7] Åsa G. Dahlstedt and Anne Persson, Requirements Interdependencies - Moulding the State of Research into a Research Agenda, Ninth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2003), held in conjunction with CAiSE 2003, 71-80.

[8] E. Bagheri, M. Asadi, D. Gasevic, and S. Soltani, 'Stratified Analytic Hierarchy Process : Prioritization and Selection of Software Features', pp. 300–315, 2010.

[9] M. Sadiq, S. Ghafir, and M. Shahid, 'An approach for eliciting software requirements and its prioritization using analytic hierarchy process', *ARTCom 2009 - Int. Conf. Adv. Recent Technol. Commun. Comput.*, pp. 790–795, 2009.

[10] Serral E., Sernani P., Dragoni A.F., Dalpiaz F. (2017) Contextual Requirements Prioritization and Its Application to Smart Homes. In: Braun A., Wichert R., Maña A. (eds) Ambient Intelligence. AmI 2017. Lecture Notes in Computer Science, vol 10217. Springer, Cham

[11] Xiaoqing (Frank) Liu, Yan Sun, Chandra Sekhar Veera, Yuji Kyoya, Kunio Noguchi, Priority assessment of software process requirements from multiple perspectives, Journal of Systems and Software, Volume 79, Issue 11, 2006, Pages 1649-1660.

[12] M. A. Iqbal, A. M. Zaidi, and S. Murtaza, ‘A new requirement prioritization model for market driven products using analytical hierarchical process’, *DSDE 2010 - Int. Conf. Data Storage Data Eng.*, pp. 142–149, 2010.

[13] Muhammad Imran Babar, Masitah Ghazali, Dayang N.A. Jawawi, Siti Maryam Shamsuddin, Noraini Ibrahim, PHandler: An expert system for a scalable software requirements prioritization process, Knowledge-Based Systems, Volume 84, 2015, Pages 179-202.