International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

© Global Society of Scientific Research and Researchers

http://ijcjournal.org/

Feature Prioritization-A Novel Method for Prioritization

Sandhia Valsala[*]

University of Buraimi, Sultanate of Oman

Email: sandhia_v@yahoo.com

Abstract

One of the most important activities during features engineering is the selection and prioritization of features satisfying various explicit and implicit objectives and constraints. One of the most important steps in incremental software development is the Selection and Prioritization of features satisfying various constraints like stakeholder preference, resources, cost of development etc. As it is practically impossible to implement all the available features in the coming release there is a need to prioritize and choose only those features satisfying these technical and non-technical constraints? In addition, Stakeholder's interest should also be considered in selecting the features that has to be included in the current version. After analyzing the different prioritization techniques, clearly understanding the drawbacks of traditional methods, the proposed algorithm for Feature prioritization can handle multiple stakeholders, feature dependency; a large feature set and provides an ordering of the generated optimum features.

Keywords: Features; Prioritization; Stakeholders; Dependency.

1. Introduction

In Incremental software development it is really a challenge to select the right group of features to be implemented from among several or many available options. A feature can be defined as a demand, a need or a description of what the system should do. For a product to be successful it's important to include the right set of features. The most appropriate group of features that can be selected for implementation largely depends on how successfully feature prioritization is done on that product release. There are different feature prioritization techniques available and it's often a crucial decision to choose the most appropriate prioritization technique. It is impossible to implement all the features in a release due to various constraints like time, budget and resources. So choosing the best prioritization technique helps to isolate the most important features. When we develop software, stakeholder's needs and expectations should be given utmost attention.

----

[*] Corresponding author.

We need to efficiently and effectively manage stakeholders' system Features and produce a software system that meets the expectations of both customers and users. But a question arises "Is it possible in real world?".Given stakeholder's preference and other constraints it will not be possible to include all the features at the same time in a system. Thus, project managers face a dilemma, how can we select a subset of the customers' Features and still produce a system that meets their needs? Here comes the importance of Feature prioritization techniques or feature prioritization techniques.

2. Literature Review

A number of researches has been done to analyze and evaluate software feature prioritization techniques .One of the study [1] discusses an analysis of the following prioritization techniques namely AHP, SERUM, EVOLVE and VOP and concludes that VOP is the best approach for Feature prioritization. Another study [2] on Feature prioritization was carried out at three types of industries, to find out the most appropriate Features prioritization technique that gives high level of satisfaction to the customer and it was found out that stakeholder preference is the most important factor to be considered while prioritizing Features. Reference [3] does a systematic literature study on various considerations that have influenced the prioritization of software Features, along with the characterizations of Feature prioritization techniques. Reference [4] in his master's thesis carried out a systematic study of the following Feature prioritization techniques namely AHP, NA and Cumulative Voting. A study [5] comparing six Feature prioritization techniques has been carried out by Ms Khari and Nikunj Kumar and it was concluded that Value oriented Prioritization (VOP) gives the most accurate result when compared with Analytic Hierarchy Process (AHP), Cumulative Voting (CV), Numerical Assignment Technique (NAT), Binary Search Tree (BST) and Planning Game (PG). Reference [6] studies Feature prioritization techniques used in Agile software development but focuses only on genetic algorithm based techniques and proposes a new technique "Interactive Genetic Algorithm" that can produce a prioritized list satisfying all the constraints. Siddiqui, Reference [7] does a comparative study on two Feature prioritization techniques namely AHP and Planning Game and concludes PG as the most appropriate technique for Feature prioritization. Most of the studies concentrated on prioritizing functional Features but the study [8] done by Saranya and Subha concentrates on prioritization of nonfunctional Features. Three algorithms namely prioritization using architectural feedback, NFR algorithm and Goal decomposition algorithm was studied and it was concluded that NFR out rates the other two algorithms in prioritizing nonfunctional Features. Ma [9] did a systemic review of Feature prioritization techniques that could prioritize medium to large scale Features. A novel approach of Feature prioritization was discussed by Bajaj and Arora which evaluates various Features prioritization methods with different characteristics and proposes a novel technique that could assist the managers to achieve better customer satisfaction by selecting the most appropriate method for their application.

3. Comparison of various Feature Prioritization Techniques.

3.1. Requirement prioritization

A number of prioritization techniques are available in literature that uses a wide variety of factors in selecting the features that need to be included in a release. Practically it is impossible to implement all the needed features in software. An ideal situation is to choose the best from the available large pool of Features. From the analysis listed below are the major factors considered in feature prioritization.

Stakeholder Preference

System stakeholders are defined as "people or organizations who will be affected by the system and who have a direct or indirect influence on the system Features". Suppose we have n different stakeholders abbreviated by s1, s2,… sn, where each stakeholder will be assigned a relative importance. Given are some Stakeholder's examples but are not limited to the following: user (novice, advanced, expert or other classifications of users), manager (project, product), developer, or sales representatives.

Feature dependency

Features cannot be considered isolated and they share a complex relationship with other Features. The relationship that exists between these Features is termed Feature dependencies. Previous studies have revealed that 80% of the Features are interdependent and only a very few are standalone Features. The mere fact that most of the Features are dependent on each other makes it one of the most important factor to be considered while selecting or prioritizing Features. Identified in (Carlshamre and his colleagues 2001) literatures are the three basic types of dependencies. These dependencies are listed below.

- Implication: This dependency between Features says that given two Features i and j, Feature i needs Feature j to function but Feature j does not need Feature i to function. In other words the first Feature needs the second one to function but the second one does not need first one to function.

- Revenue-based: Two Features i and j are said to have revenue based dependency if Feature i influences the revenue of Feature j.

- Cost-based : Two Features i and j are said to have cost based dependency if Feature i influences the cost of Feature j

4. Proposed Algorithm for Feature Prioritization

The proposed algorithm called the "Best_Choose Algorithm" uses two major factors in selecting the best features to be included in the release.The first factor being the stakeholder preference and second being Feature dependency.Stakeholder are the main people involved in the application who are directly involved in the system.Feature dependency refers to the interdependency that exist between the Features.

Proposed Best_Choose Algorithm.

1. Accept the list of features to be implemented from the stakeholders
2. Group the Features based on dependency into Feature subsets
3. Elimainate all repeated Feature subset
4. Calculate priority value of each Feature group by considering the stakeholder priority.

5. Sort the Features based on stakeholder priority.
6. Choose the best Feature group for implementation.

The list of features to be implemented is the initial input .These features will be interdepenedent to each other. In our simulation we are considering three types of depenedency namely Implication,Cost-Related and Revenue–Related. Implication is denoted as IMP ,Cost-related dependency is represented as CT and Revenue Releated dependency is represented as RT. A sample data set of 20 Features is used in this research as given in Table 1. Stakeholders are categorized into different categories based on their positions. We have identified eight Stakeholders category namely Project Manager, Project Lead, Team Head, Team Members, Users, Sales Team, Representative and Software Supplier. The Table 2 shows the sample list of Stakeholders considered. These stakeholders assign their priority score to each Feature with scores ranging from 1 to 5 where 5 denotes high priority and 1 denotes lease priority as given in Table 3.

Table 1: Sample Data Set

Feature Id	Description	Dependency	Stakeholder Priority
'1'	System Should handle multiple users	IMP 6	0.4
'2'	System should handle errors		0.2
'3'	System should identify loyal customers		0.4
'4'	System should generate reports		0.2
'5'	System should calculate leave	IMP 2	0.3
'6'	System should do validations		0.4
'7'	System should do automatic updations		0.4
'8'	System should be dynamic		0.4
'9'	System should send messages to customers	CT 12	0.2
'10'	System should handle payment		0.4
'11'	System should have handle customer contact		0.2
'12'	Automatic response		0.3
'13'	Food description updation		0.4
'14'	Payment System	CT 7	0.4
'15'	Live chat		0.4
'16'	Live camera view		0.2
'17'	Chat with Chef	RB 6	0.2
'18'	Cookery show		0.4
'19'	Add location	CT 16	0.2
'20'	Upload pictures	IMP 16	0.4

All the stakeholders will not have the same priority. So the stakeholders S1 –S8 were weighted using AHP from a general project management perspective. A Matrix of pair-wise comparison of stakeholders on a nine-point scale is done based on the Table 4 given. Weightings are used to discriminate between stakeholders. For greater flexibility and objectivity these weightings are further calculated using the pairwise comparison method from AHP.

Table 2: Stakeholder categories

Stakeholder. No	Stakeholder Categories
S1	Project Manager
S2	Project lead
S3	Team Head
S4	Team Members
S5	User
S6	Sales
S7	Representative
S8	Software Supplier

Table 3: Sample Stakeholder Assigned Values

Stakeholder	***Features***																			
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20
S1	5	1	4	1	1	3	5	5	1	4	1	1	3	5	4	1	1	4	1	5
S2	3	5	4	3	4	5	4	3	5	4	3	4	5	4	4	3	5	4	3	4
S3	4	2	4	2	3	3	3	4	2	4	2	3	3	3	4	2	2	4	2	3
S4	5	1	3	1	4	3	2	5	1	3	1	4	3	2	3	1	1	3	1	2
S5	3	3	4	3	5	3	1	3	3	4	3	5	3	1	4	3	3	4	3	1
S6	2	1	2	1	2	4	4	2	1	2	1	2	4	4	2	1	1	2	1	4
S7	1	2	5	3	3	2	3	1	2	5	3	3	2	3	5	3	2	5	3	3
S8	5	5	3	4	5	4	5	5	5	3	4	5	4	5	3	4	5	3	4	5

Table 4: Scale of absolute numbers

Numerical values	Verbal Judgment
1	Equal Importance
2	Weak or slight
3	Moderate importance
4	Moderate plus
5	Strong importance
6	Strong plus
7	Very strong or demonstrated importance
8	Very, very strong
9	Extreme importance
Reciprocals of above : If activity *i* has one of the above non-zero numbers assigned to it when compared with activity *j*, then *j* has the reciprocal value when compared with *i*	

After representing the comparison matrix ,priorities are computed by finding the principal Eigen values and the corresponding Eigen vector of the pairwise comparison matrix.

The normalized principal Eigen vector is the priorities vector. To compute the priorities the following two steps are to be carried out.

- Normalize each column
- Find average over each row

The stakeholder weightings are computed from the eigenvalues of the matrix shown in Table 5. The computed stakeholder weightings is shown in the Table 5.The technique of averaging over normalized columns can be used to approximate the eigenvalues.

The normalized principal Eigen vector [21] is also called priority vector. Since it is normalized, the sum of all elements in priority vector is 1.

The priority vector shows relative weights among the considered stakeholders. Combining Stakeholder priority value and the stakeholder's Feature preference score, we arrive at Stakeholder Priority value for each Feature. Stakeholder_ Feature priority for individual Features is calculated and represented in Table 7.

Combining the tables the final Feature priority is represented in Table 8.

Table 5: Matrix of pair-wise comparison of stakeholders on a Nine-point Scale of AHP

Stake Holder	**Stake Holder**							
	S1	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
S1	1	7	4	3	9	2	8	6
S2	1/7	1	3	2	8	2	7	5
S3	1/4	1/3	1	3	8	2	8	5
S4	1/3	1/2	1/3	1	7	2	7	6
S5	1/9	1/8	1/8	1/7	1	2	8	7
S6	1/2	1/2	1/2	1/2	1/2	1	9	7
S7	1/8	1/7	1/8	1/7	1/8	1/9	1	6
S8	1/6	1/5	1/5	1/6	1/7	1/7	1/6	1

Table 6: Calculated Stakeholder Priority

Stakeholder	**Stakeholder Weight**
S1	.31
S2	.15
S3	.14
S4	.11
S5	.06
S6	.09
S7	.03
S8	.02

Table 7: Stakeholder Priority for each feature

Stakeholder	Features																			
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
S1	1.6	0.3	1.2	0.3	0.3	0.9	1.6	1.6	0.3	1.2	0.3	0.3	0.9	1.6	1.2	0.3	0.3	1.2	0.3	1.6
S2	0.5	0.8	0.6	0.5	0.6	0.8	0.6	0.5	0.8	0.6	0.5	0.6	0.8	0.6	0.6	0.5	0.8	0.6	0.5	0.6
S3	0.6	0.3	0.6	0.3	0.4	0.4	0.4	0.6	0.3	0.6	0.3	0.4	0.4	0.4	0.6	0.3	0.3	0.6	0.3	0.4
S4	0.3	0.1	0.2	0.1	0.2	0.2	0.1	0.3	0.1	0.2	0.1	0.2	0.2	0.1	0.2	0.1	0.1	0.2	0.1	0.1
S5	0.2	0.2	0.2	0.2	0.3	0.2	0.1	0.2	0.2	0.2	0.2	0.3	0.2	0.1	0.2	0.2	0.2	0.2	0.2	0.1
S6	0.2	0.1	0.2	0.1	0.2	0.4	0.4	0.2	0.1	0.2	0.1	0.2	0.4	0.4	0.2	0.1	0.1	0.2	0.1	0.4
S7	0.0	0.1	0.2	0.1	0.1	0.1	0.1	0.0	0.1	0.2	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.2	0.1	0.1
S8	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Table 8: Final Stakeholder Priority for features

Feature	Stakeholder Priority
Feature1	0.4
Feature2	0.2
Feature3	0.4
Feature4	0.2
Feature5	0.3
Feature6	0.4
Feature7	0.4
Feature8	0.4
Feature9	0.2
Feature10	0.4
Feature11	0.2
Feature12	0.3
Feature13	0.4
Feature14	0.4
Feature15	0.4
Feature16	0.2
Feature17	0.2
Feature18	0.4
Feature19	0.2
Feature20	0.4

On analyzing the Table -1 we can see various dependencies set. Given the sample dataset with Feature dependencies and stakeholder priority ,the first step is to group the Features based on dependency.

STEP 1: Group the Feature based on dependency and thus form Feature Subsets. The following subsets are generated after grouping the Features based on dependency and remove the redundant Feature subsets.

Table 9

Generated Feature Subset
Feature1, Feature6, Feature17
Feature2, Feature5
Feature3
Feature4
Feature5, Feature2
Feature7, Feature14
Feature8
Feature9, Feature12
Feature10
Feature11
Feature13
Feature15
Feature16, Feature19, Feature20
Feature18

STEP2: Compute the Stakeholder priority of the generated Feature subsets

Table 10

Feature Subset	Features in Subset	Stakeholder Priority
FeatureSubset1	Feature1, Feature6, Feature17	1
FeatureSubset2	Feature2, Feature5	.5
FeatureSubset3	Feature3	.4
FeatureSubset4	Feature4	.2
FeatureSubset5	Feature5, Feature2	.5
FeatureSubset6	Feature7, Feature14	.8
FeatureSubset7	Feature8	.4
FeatureSubset8	Feature9, Feature12	.5
FeatureSubset9	Feature10	.4
FeatureSubset10	Feature11	.2
FeatureSubset11	Feature13	.4
FeatureSubset12	Feature15	.4
FeatureSubset13	Feature16, Feature19, Feature20	.8
FeatureSubset14	Feature18	.4

STEP3: Final Sorted Feature Set

Table 11

Feature Subset	Features in Subset	Stakeholder Priority
FeatureSubset1	Feature1, Feature6, Feature17	1
FeatureSubset2	Feature7, Feature14	0.8
FeatureSubset3	Feature16, Feature19, Feature20	0.8
FeatureSubset4	Feature2, Feature5	0.5
FeatureSubset5	Feature5, Feature2	0.5
FeatureSubset6	Feature9, Feature12	0.5
FeatureSubset7	Feature3	0.4
FeatureSubset8	Feature8	0.4
FeatureSubset9	Feature10	0.4
FeatureSubset10	Feature13	0.4
FeatureSubset11	Feature15	0.4
FeatureSubset12	Feature18	0.4
FeatureSubset13	Feature4	0.2
FeatureSubset14	Feature11	0.2

STEP 4: Choose the Feature number of features for implementation

The given scenario requires 8 features to be selected and the resultant features based on the two selection factors will be Feature1, Feature6, Feature17,Feature7, Feature14, Feature16, Feature19,Feature20.

5. Graphical Output

In order to make the output more presentable and easy to understand, we pictorially represent the generated output using the following graphs. The first graph shows the computed stakeholder priority for each Feature.

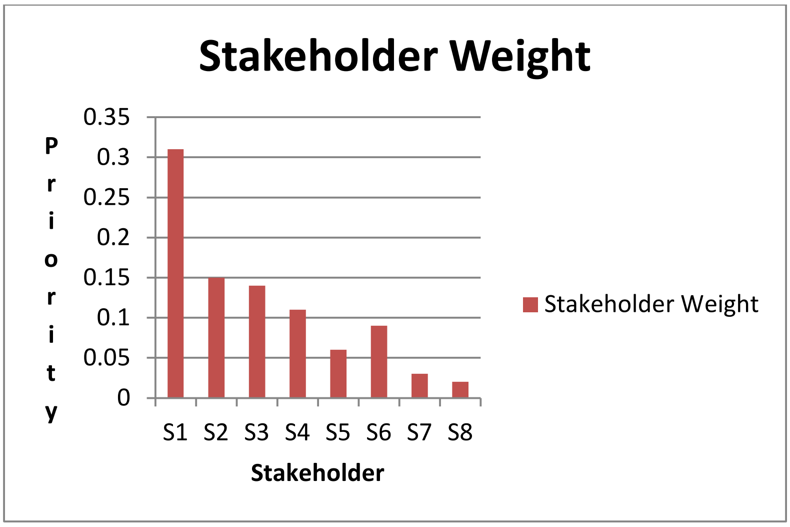

Figure 1: Computed stakeholder priority for each Feature

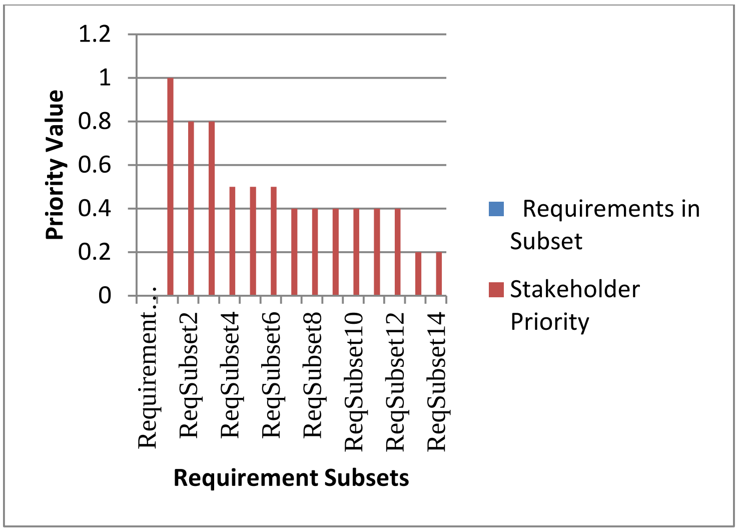

Figure 2: Calculated Priority value of Feature Subset

Two main approaches of software development are iterative and incremental. The first and foremost process in software development is Feature selection and prioritization. Feature prioritization is the process of ordering the Features for implementation in a release [18]. Several Features will be available for implementation but it is impossible to have them all implemented in the current release [17]. Choosing the best among them becomes a crucial factor when this selection is subject to various constraints like cost, revenue, budget, and dependency. Each release includes a set of Features satisfying certain constraints of the organization. There are so many factors that are considered in the selection of Features like cost, effort, resources, time and stakeholder's preference. Different prioritization techniques consider different technical and non-technical factors for Feature selection.After analyzing the different prioritization techniques, clearly understanding the drawbacks of traditional methods,[19] the proposed algorithm for Feature prioritization can handle multiple stakeholders, Feature dependency; a large Feature set and provides an ordering of the generated optimum Features.

6. Conclusion and Future Recommendations

We have identified taxonomy of selection factors that need to be considered during Feature prioritization. The proposed algorithm for Feature prioritization can handle multiple stakeholders, feature dependency; a large feature set and provides an ordering of the generated optimum features. The research can be further extended to include the following. The research work can be further extended to include additional selection factors as the existing model considers only two main factors for Feature prioritization namely stake holder preference and Feature dependency. It can be further extended to include additional factors while performing prioritization.

References

[1] Aaqib Iqbal, Farhan M Khan, Shahbaz A Khan (2009). “A Critical Analysis of techniques for Feature Prioritization and Open Research”, International Journal of Reviews in Computing, Vol. 1, No. 1.

[2] Shams ul Hassan , Salman Afsar Awan (2012) , “Software Engineering : Factors Affect on Feature

Prioritization”, Global Journal of Computer Science and Technology Software & Data Engineering Volume 12 Issue 15 No 1

[3] Rahul Thakurta (2014),”Research Trends on Software Feature Prioritization”,International Journal of Software Engineering and Its Applications Vol.8, No.6 (2014), pp.287-298.

[4] “A Systematic Review of Software Features Prioritization”, Thesis Report , School of Engineering Blekinge Institute of Technology ,Sweden ,2006

[5] Manju Khari, Nikunj Kumar (2013) ,”Comparison of six prioritization techniques for software Features , Journal of Global Research in Computer Science,Volume 4, No. 1, January 2013

[6] Nupoor Garg ,Neha Malhotra (2013) , “Prioritization of Features based upon Quality Approach and Interactive Genetic Algorithm focused on Agile methodologies”, International Journal of Computer Science & Communication Networks,Vol 3(2), 111-113

[7] Shadab Siddiqui, Dr.Mohd. Rizwan Beg, Shahin Fatima (2013),Effectiveness of Feature Prioritization Using Analytical Hierarchy Process (AHP) And Planning Game (PG): A Comparative Study, International Journal of Computer Science and Information Technologies, Vol. 4 (1) , 2013, 46 – 49.

[8] Saranya. B , Subha. R. , Dr. Palaniswami. S.(2014), “ A Survey on Prioritization Methodologies to Prioritize Non-Functional Features”, International Journal of Computer Science and Business Informatics, Vol. 12, No. 1

[9] “The Effectiveness of Features Prioritization Techniques for a Medium to Large Number of Features”,Thesis Report , Auckland University,2009

[10] Kaur, G., and Bawa, S. (2013). “A survey of Feature prioritization methods”. Int. J. Eng.Res. Technol, pp.958-962.

[11] Achimugu, P., Selamat, A., Ibrahim, R., and Mahrin, M. N. (2014). A Systematic Review of Software Feature Prioritization Research. Information and Software Technology, Vol 56, pp.568-585.

[12] Thakurta, R. (2012). A framework for prioritization of quality Features for inclusion in a software project. Softw. Qual. J., pp.573–597.

[13] Wohlin, C., and Aurum, A. (2005). What Is Important when Deciding to Include a Software. Proceedings International Symposium on Software Engineering ,IEEE, (pp. 246-255). Austarlia.

[14] Voola, P., and Babu, A. (2013). Comparison of Features Prioritization TechniquesEmploying Different Scales of Measurement. ACM SIGSOFT Software Engineering Notes, Vol 38, pp. 1-10.

[15] Pitangueira, A., Maciel, R., Barros, M., and Andrade, A. (2013). A systematic review of software

Features selection and prioritization using SBSE approaches in Search Based Software Engineering (Vol. 8084). Springer, Berlin Heidelberg.

[16] Meenakahi. (2014). Reengineering a Software Process by using Unified Foundation. International Journal of Inventions in Computer Science and Engineering, Vol 1(3), pp.301-307.

[17] Srivastava, P. R., Sankaran, S., and Pandey, P. (2013). Optimal Software Release Policy Approach Using Test Point Analysis and Module Prioritization. MIS Review, an International Journal, Vol 18(2), pp.19-50.

[18] Sharma, C., Sabharwal, S., and Sibal, R. (2011). Applying Genetic Algorithm for Prioritization of Test Case Scenarios Derived from UML Diagrams. International Journal of Computer Science, Vol 8(3), pp.433-444.

[19] Elsayed, S., Sarker, R., and Essam, D. (2014). A new GA for solving Optimization Problems. Elesevier, pp.57-69.

[20]. Dabbagh, M., and Lee, S. P. (2014). An Approach for Integrating the Prioritization of Functional and Nonfunctional Features. The Scientific World Journal Volume, Article ID-737626, pp.1-12.

[21]. http://web.cjcu.edu.tw/~lcc/Courses/TUTORIAL/AHP%20Tutorial.doc