A New Requirement Prioritization Model for Market Driven Products Using Analytical Hierarchical Process

Muhammad Atif Iqbal
Department of computer Sciences, NUST, Pakistan
atif09@gmail.com

Athar Mohsin Zaidi
Department of computer Sciences, NUST, Pakistan
athar-mcs@nust.edu.pk

Dr. Saeed Murtaza
Department of computer Sciences, NUST, Pakistan
shouzib@yahoo.com

Abstract- *As compared to bespoke software development, in market driven software development huge numbers of requirements that comes from many different users, customers, competitors through surveys, interviews, focus groups, and competitor analysis intimidate to over burden the requirement engineering process. So it is vital for product management to select the requirements that aligned with the overall business goals and dispose of others as early as possible. This paper presents a new model for requirement prioritization using AHP (Analytical Hierarchical Process). Number of comparisons as compare to traditional AHP techniques are significantly reduced which make it practicable and applicable. In creation of this model qualitative and quantitative research methodologies are used. To evaluate the model both static and dynamic validation techniques are used to assess the model in industry where the aspects of practicality, applicability, Implementation, effectiveness, and efficiency were investigated, yielding promising initial results.*

Keywords: AHP, Bespoke, Dynamic validation, Market-driven, Model, Requirements, Static validation

I. INTRODUCTION

Due to the emergence of markets for off-the-shelf or packaged software [1], market-driven software development is gaining increased significance in comparison to traditional bespoke software development [9]. Due to this shift, the requirements engineering area is very much affected [9]. In conventional requirements engineering for bespoke software products the focus is on satisfying the specific customer's requirements whereas in market-driven requirements engineering the number of requirements in most cases is very huge. In market-driven software development environment, requirements do not only come from the developers, marketing groups, sales team, support groups, bug reports etc [10] but also from many different users, customers, competitors through surveys, interviews, focus groups, and competitor analysis.

Prioritizing software requirements helps to determine which requirements that are most important, and in which order requirements should be developed and tested throughout the development lifecycle. By prioritizing the requirements, software engineers can put focus on a subset of all requirements, and implement these in a particular release but the biggest challenge is how to prioritize large number of requirements in case of market-driven software development.

This paper presents a MDRPM (Market Driven Requirement Prioritization Model) which uses AHP. The layout of the paper is as follows, section II describes the overview of existing requirement prioritization techniques, section III go through the steps involved in making if MDRPM, section IV explains the functionality of MDRPM and implementation of all the steps involved in it, Validation process and result of this validation is discussed in section V, VI and in the last section VII gives us the conclusion.

II. RELATED AREA AND LITERATURE STUDY

As discussed in the earlier section, in case of market-Driven software products, it is not unusual to have hundreds and even thousands of individual requirements. Whereas all requirements are mandatory but some are more critical than others. For example, failure to implement certain requirements may have grave business ramifications that would make the system a failure, while others although contractually binding would have far less serious business consequences if they were not implemented or not implemented correctly [4]. Requirements prioritization is recognized as a most important and difficult activity in Market-Driven Requirement Engineering [11] [13]. The most vital and decisive decision making is needed in requirement prioritization activity. The selection of requirements from the given set of features requires a very high decision power because it is done most of time by ad-hoc methods [9]. The incorrect decision leads to wrong implementation of selected requirements [9].

Findings indicate that currently used prioritization methods have limited ability to support decision-making in a complex area like requirements prioritization in market-driven product development [13]. In addition, there are practical difficulties in the usage of methods, and therefore, prioritization results should be taken more as being indicative than as an ultimate truth [14]. Current approaches are not capable of handling large number of requirements as in case of market driven products or large system [11][14]. On the basis of above facts, Therefore, a need for a model of good practices in market-driven requirements prioritization for market driven products has

978-0-7695-3958-4/10 $26.00 © 2010 IEEE
DOI 10.1109/DSDE.2010.49

been identified, which could come to help cater the challenges faced by MDRE and transfer of knowledge from academia to industry.

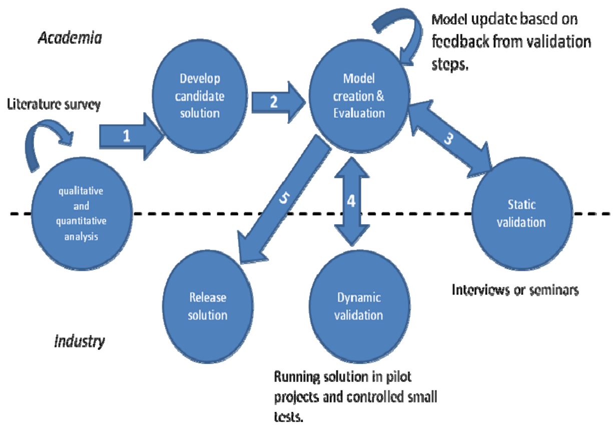

Figure 1 Steps involved in making of MDRPM

The steps involved in the making of MDRPM are shown in Figure 1. In the first step both quantitative and qualitative analysis [23] are performed which include brief literature survey, interviews from industry representatives, analysis of requirement specification documents of previous software projects done by different companies. On the basis of step 1, a candidate solution is developed in step 2. Once the candidate solution is developed, it is validated in both industry and academia through static and dynamic validation [23] in step 3 and 4 respectively. Model is updated on the basis of static and dynamic validation. Finally Model is released in step 5.

III. MDRPM (MARKET DRIVEN REQUIREMENT PRIORITIZATION MODEL)

This section presents MDRPM and brief description of steps involved in it.

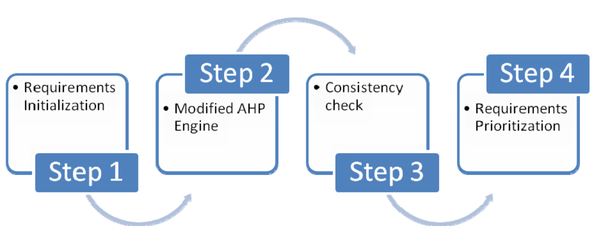

Figure 2 Steps involved in MDRPM

The MDRPM is based upon the following assumptions.

1. **Abstract requirements:** Abstract requirements [2] are available to be comparable to other requirements.
2. **No Requirements dependencies:** There are no dependencies [4] between requirements, all requirements are independent.
3. **No conflict:** There is no conflict if pair wise comparison to be performed by more the than two persons during step-2.

4.1. *Step 1 (Requirement Initialization)*

Pre-defined bins are created as a result of qualitative and quantitative analysis and industry input. In step-1 one all the requirements will be traversed by a requirement engineer or a team performing requirement prioritization and using their knowledge and experience every requirement will be assigned a bin on the principle of best fit [8].

4.2. *Step 2 (Modified AHP Engine)*

Once all the requirements are inserted in their corresponding bins according to best fit approach, if there are n number of bins than all the bins from i to n is processed in AHP engine [18] where i is the first bin.

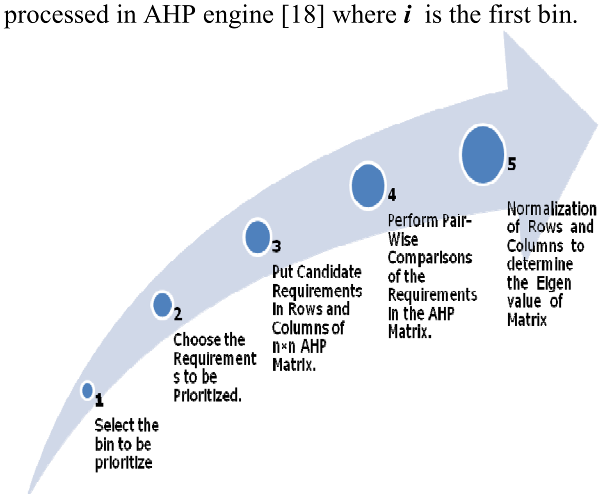

Figure 3 Steps involved in AHP Engine

If the total numbers of requirements in bin i is n, then the total number of comparisons are $n \times (n-1) / 2$. Place the candidate requirements in the $n{\times}n$ AHP matrix. In order to set up the priorities, the scale shown in Table 2 is used.

S No	Requirement related to Bin	Description
1	Accessibility	Degree to which a system is accessible by as many people as possible.
2	Audit and control	Information system auditing
3	Availability	The degree to which a system is operable and in a committable state at the start of a mission.
4	Backup	Making copies of data so that these additional copies may be used to restore the original after a data loss event.
5	Usability	Ease with which people can make use of a particular system in order to achieve a particular goal.

6	Certification	Requirements that must be met in order to be able to get the system certified.
7	Compliance	A state in which someone or something is in accordance with established guidelines, specifications, or legislation.
8	Reliability	The ability of a person or system to perform and maintain its functions in routine circumstances, as well as hostile or unexpected circumstances.
9	Resilience	The ability to provide and maintain an acceptable level of service in the face of faults and challenges to normal operation.
10	Documentation	System related manuscript.
11	Performance / Response time	The amount of useful work accomplished by a system compared to the time and resources used.
12	Platform compatibility	Hardware architecture or software framework (including application frameworks), that allows system to run.
13	Recovery	Retrieve data loss refers to the unforeseen loss of data or information.
14	Effectiveness	Resulting performance in relation to effort
15	Escrow	A trust account held in the borrower's name to receive.
16	Extensibility	Design principle where the implementation takes into consideration future growth.
17	Legal and licensing issues	Requirements related to contract between a producer and a purchaser of computer software that is included with software.
18	Interoperability	The ability of diverse systems and organizations to work together (inter-operate).
19	Failure Management	Attempts to reboot the system failed to correct the problem and, when the backup system failed to cut-in.
20	Modifiability	Customization of the system according to one`s need or to meet new requirements
21	Open Source	Offering practical accessibility to a software's source code
22	Operability	The ability to keep a system in a safe and reliable functioning condition, according to pre-defined operational requirements.
23	Usability by target user	Ease with which selected people can make use of a particular system in order to achieve a particular goal.
24	Efficiency	
25	Privacy	Built-in capabilities to protect the *confidentiality* of its users
26	Portability	Making system available from an environment to another
27	Disaster Recovery	The process, policies and procedures related to preparing for recovery or continuation of technology infrastructure critical to an organization after a natural or human-induced disaster.
28	Testability	Can be applicable to an empirical hypothesis
29	Resource constraints	Whether or not the processes lead to the desired outcome within resource constraints.
30	Response time	The time a system or functional unit takes to react to a given input.
31	Robustness	Capability of coping well with variations (sometimes unpredictable variations) in its operating environment with minimal damage, alteration or loss of functionality.
32	Scalability	Ability to either handle growing amounts of work in a graceful manner or to be readily enlarged
33	Security	The degree of protection against danger, loss, and criminals
34	Compatibility	System that runs on one of the models can also be run on all other models of the family.
35	Supportability	The ability to support install, configure, and monitor system.

Table 1 Name and Description of Bins

This scale is very much similar to the one that was designed by Saaty [21] for comparing requirements. The scale not only tells us which requirement is more important than the other, but also how much one requirement is more important than other.

How Important	**Description**
1	Equal Importance
2	Intermediate value between 1 and 3
3	moderate difference in importance
4	Intermediate value between 3 and 5
5	Essential difference in importance
6	Intermediate value between 5 and 7
7	Major difference in importance
8	Intermediate value between 7 and 9
9	Extreme difference in importance

Table 2 Scale for prioritizing requirements

Let us suppose that there are ***'n'*** numbers of requirements, R1, R2 ….Rn. Once we put these into ***n×n*** matrix, the relative value for requirements are presented as Rij, where rows have index of ***'i'*** and columns have index of ***'j'*** [21]. For each pair of requirement Rij, the requirements are compared and the relative value is inserted from Table 2 [20].
If relative importance of requirement Ri is equal to requirement Rj than Rij=1, Rji =1. Hence, we insert '1' in all the positions where Ri is compared to Ri (main diagonal). As shown in Table 3 we have inserted '1' at all diagonal positions (R11, R22, R33, R44 . . . Rnn).

Requirements	**R1**	**R2**	**R3**	**R4**	**. . .**	**R*n***
R1	1					
R2		1				
R3			1			
R4				1		
. . .					1	
R*n*						1

Table 3 AHP nxn Priority Matrix

If requirement Rij has priority value of X, then requirement Rji has priority value of 1/X (reciprocal value of Rij). For each pair of requirements (starting with R1 and R2) inserts their relative intensity of value from Table 2. Example of how this step will be performed in shown in table 4, in the position (R12) where the row R1 intersects column R2. As shown in Table 4 we have inserted values of relative intensity in columns (R12, R13, R14, R23, R24, and R34). For example we have put value '5' in position (R12) which shows that R1 is '5 times' more important than R2.Inspite of this selection one can also select the values that are less important rather than more important.

When this is done, the reciprocal values are automatically calculated. We have put reciprocal values in columns (R21, R31, R41, R32, R42, and R43) as shown in Table 4. For instance in position (R21) insert the reciprocal value, in this case value in (R21) is 0.2 which shows that requirement R2 is 0.2 times less important than R1.Once the prioritized values are inserted into the rows and columns of AHP matrix, the next task is to normalize the values by using a method called 'Averaging over normalized columns' [18] [21]. The steps to perform Averaging over normalized columns are shown below. First we calculate the sum of columns in the comparison matrix. As shown in Table 5, we have calculated sum of columns for four requirements.

Requirements	R1	R2	R3	R4
R1	1	**5**	5	7
R2	0.2	1	3	3
R3	0.2	0.33	1	9
R4	0.14	0.33	0.11	1
Total	**1.54**	**6.66**	**9.11**	**20**

Table 4 Sum of columns in the comparison matrix

After calculating the sum of columns, the next step is to normalize the sum of rows convert each value to the percentage of its column total. We divide the value of Rij by the sum of columns. As shown in Table 5, the value of column (R11) is '1' and sum of that column (Total) is 1.54.We divide 1 by 1.54 which results in the value 0.65, see Table 6. The value 0.65 is normalized value for columns (R11). Similarly we calculate normalized values for the whole matrix.

Requirements	R1	R2	R3	R4
R1	1/1.54 =0.65	5/6.66 =0.75	5/9.11 =0.55	7/20 =0.35
R2	0.2/1.54 =0.13	1/6.66 =0.15	3/9.11 =0.33	3/20 =0.15
R3	0.2/1.54 =0.13	0.33/6.66 =0.05	1/9.11 =0.11	9/20 =0.45
R4	0.14/1.54 =0.09	0.33/6.66 =0.05	0.11/9.11 =0.01	1/20 =0.05

Table 5 Normalized values for the whole matrix.

The normalized values calculated above are used to calculate the priority ordering for every requirement. To calculate row average value we calculate sum of rows. For each row, we add every normalized values of that row and divide it with the number of requirements. As shown in Table 7 for row 1, sum of rows is (R11 +R12+R13+R14) which equals to 2.3. Similarly we calculate the sum of rows for other three rows.

Requirements	R1	R2	R3	R4	Sum of rows
R1	0.65	0.75	0.55	0.35	**2.30**
R2	0.13	0.15	0.33	0.15	**0.76**
R3	0.13	0.05	0.11	0.45	**0.74**
R4	0.09	0.05	0.01	0.05	**0.20**

Table 6 Calculate the sum of rows for AHP Matrix.

The sum of rows of each row is divided by four (number of requirements). The priority value we got for row 1 is 0.575. Similarly we calculate values for other three rows. These values are called Eigen values of the matrix.

Requirements	R1	R2	R3	R4	Priority
R1	0.65	0.75	0.55	0.35	2.3/4 =0.575
R2	0.0.13	0.15	0.33	0.15	0.76/4=0.19
R3	0.13	0.05	0.11	0.45	0.74/4=0.185
R4	0.09	0.05	0.01	0.05	0.2/4=0.05
Total					**1.00**

Table 7 Eigen values of the matrix.

The final matrix is represented as normalized eigenvector of the comparison matrix. The final result shows the degree of importance of one requirement to another, with respect to the particular aspect customer keeps in mind before performing prioritization [21][22].

R1 contains 57.5% of total importance.
R2 contains 19% of total importance.
R3 contains 18% of total importance.
R4 contains 5% of total importance.

The resulting values not only reflect that R1 is more important than R2, R2 is more important than R3 and R3 is more important than R4; it also shows how much one requirement more/less important than another. For instance, R1 is 11.5 times more important than R4 and R4 is 3.6 times less important than R3.

4.3. Step 3 (Consistency Check)

While performing prioritization there may be likelihood that Eigen values are not consistent. Let us assume that if in final result we determined that requirement R1 is more important than R2, requirement R2 is more important than R3, and R3 is more important than R1.This shows that there is inconsistency while performing pair wise comparisons, in reality R1 is more important than R3.These kind of errors that take place in performing AHP, are called judgmental errors [22].

When the prioritization is done; next task is to check consistency of Eigen values. Consistency Index (CI) is the indictor of accuracy of pair wise comparisons performed. To overcome judgmental errors we need to perform a consistency check [18]. The CI is calculated by using the following formula.

CI (Consistency Index) = (λmax – n)/ (n-1)

Where
λmax = Maximum Eigen value of the comparison matrix
n = Number of requirements

The closer the value of λmax is to n; greater is the possibility of having consistent results and little judgmental errors. Steps to calculate the value of λmax are as follows [20] [21]:

1) To facilitate the calculations we have transformed the data presented in Table 4 into Matrix A (4 × 4) and the Priority column of Table 7 into Matrix B (4 × 1). Next, we have multiplied two matrixes which results in Matrix C (4 × 1).

$$\begin{pmatrix} 1 & 5 & 5 & 7 \\ 0.2 & 1 & 3 & 3 \\ 0.2 & 0.33 & 1 & 9 \\ 0.14 & 0.33 & 0.11 & 1 \end{pmatrix} \times \begin{pmatrix} 0.575 \\ 0.19 \\ 0.185 \\ 0.05 \end{pmatrix} = \begin{pmatrix} 2.42 \\ 1.01 \\ 0.81 \\ 0.21 \end{pmatrix}$$

Matrix A **Matrix B** **Matrix C**

I Divide the first element of Matrix C with first element of Matrix B, second element of Matrix C with second element of Matrix x B and so on. These results into Matrix D (4× 1).The elements of Matrix D are D1, D2, D3 and D4.

$$\begin{pmatrix} 2.42/0.575 \\ 1.01/0.19 \\ 0.81/0.185 \\ 0.21/0.05 \end{pmatrix} = \begin{pmatrix} 4.20 \\ 5.31 \\ 4.39 \\ 4.27 \end{pmatrix}$$

Matrix C **Matrix D**

2) The next step is to calculate the average sum (λmax).

The following formula is used to calculate the value of λmax.

λmax = (D1 + D2 +D3…+Dn)/ n
Where n= Number of requirements

In this case we have four requirements

λmax = (4.20 + 5.31 + 4.39 + 4.27)/4 = 4.54
Once value of λmax is calculated we calculate value of CI using formula (1)
CI = (4.54 – 4) /3 = 0.18
To further evaluate whether CI value is acceptable or not, we calculate Consistency Ratio (CR). CR is the ratio of consistency index to random index (RI).

Saaty has proposed the scale for random index to calculate consistency ratio [21]. As shown in Table 8, the value of RI varies with the number of objects to prioritize and that its RI vector is fixed.
The formula to calculate Consistency Ratio (CR) is:
CR = CI/RI
Where
CI = Consistency Index
RI = Random Index
For example the order of matrices is (4×4); therefore we issue RI value of order (4) from Table 9. The RI value for order (4) is 0.9.
The Consistency Ratio (CR) for this example is: CR = 0.18/0.9 = 0.2

According to [21], if the value of CR is equal or less than 0.1, than it results in to very little judgmental errors and yields accurate results. The results in this case, show the moderate degree of judgmental errors.

Requirements (n)	RI
1	0
2	0
3	0.58
4	0.9
5	1.12

6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.48
13	1.56.
14	1.57
15	1.59
16	1.60
17	1.62
18	1.62
19	1.63
20	1.65
21	1.66
22	1.68
23	1.69
24	1.72
25	1.73

Table 8 Scale for random index

4.4. Step 4 (Requirement Prioritization)

Prioritize all requirements in ***n*** bin, processing one bin at a time. When all the requirements are prioritize now it's time to prioritize the bins itself. For this purpose step 2-3 is used and the final outcome is prioritized requirements.
For example we have five bins; first we prioritize all the requirements in b1. Demonstration of how to do this step is shown in figure 4; this is the step we performed in step 1, 2 and 3.

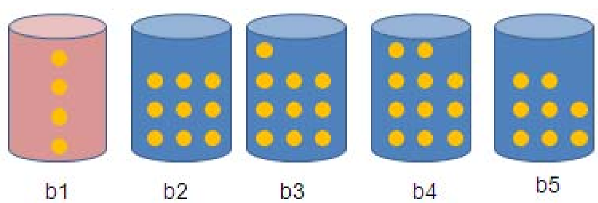

Figure 4 Example showing bin b1 is prioritized.

Then using step 2-3 all requirements in all the bins (b1, b2, b3, b4, b5) are prioritized.

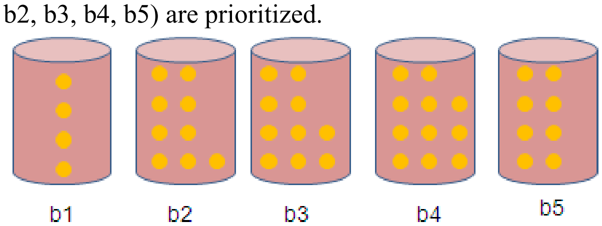

Figure 5 Example showing bin b1, b2, b3 and b4 are prioritized.

In the first step we prioritize all requirements within each bin; Next step is to prioritize bins itself. For this purpose we will again use step 2 and 3 but this time we are not prioritizing requirements but bins. Let us suppose the order in which the bins are prioritized is b5,b2, b4, b1, b3. Bin b5 has the highest priority where as bin b3 has the lowest priority. Now use the appropriate representation to show prioritizes requirements.

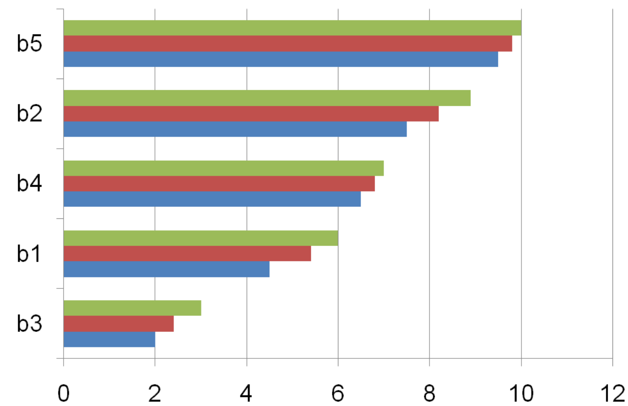

Figure 6 Graph showing prioritized requirements

V. VALIDATION OF MDRPM

Purpose of the validation step is to evaluate the aspects of practicality, applicability, Implementation, effectiveness, and efficiency of MDRPM. Both static and dynamic validation techniques are used. Static validation involved presentation of the candidate solution in industry or academia and collecting feedback from practitioners [23]. A set of qualitative and quantitative questions is designed to validate the proposed model from the interviewees. A thorough literature study of product management and requirements engineering literature is done by the author. Brainstorming is done by the authors to identify suitable questions to be included for the interview. Based on the outcome of the brainstorming session, a set of qualitative and quantitative questions is developed by the authors.
Steps involved in validation of MDRPM are shown in figure 8.
Dynamic validation means that the proposed solution is actually applied through case study in the industry in some project(s) by the roles suggested [23]. This realistically evaluates the scalability and potential of the proposed solution in the industry. Refinements can be done based on the practical validation of the proposed solution. Two companies A and B were chosen for the validation.

V.I Validation results

In this section summary of static and dynamic validation results is presented, the goal of the validation process was to investigate Model`s practicality, applicability, Implementation, effectiveness and efficiency.

V.I.I Practicality and applicability

The model is found to be practical and easy to use once it is clearly understood. It is also found that the model is customizable based on the organization's individual goals and objectives. In order to customize the model new bins has to be introduced or modification of existing ones according to company`s need. During the validation interviews the management personnel from company A suggested that applying model on more case studies would help in understanding and customization of the model. Both the companies A and B showed interest in implementing MDRPM as it is a strong candidate for requirements prioritization for their new product. Representatives from both the companies showed no reservations in declaring this model practical and applicable.

V.I.II Model Implementation

The pre-requisite for implementing this model is not considered as overhead. The pre-request includes requirement abstraction, no requirement dependencies and no-conflict. All the existing requirement prioritization models are unable to cope with large number of requirements in case of market-driven products which make them non practical, and so they are ignored by companies but in case of MDRPM it has all the strength of AHP and by adding few modifications made it scalable. The representative from both the companies agreed that this model is implementable. Representative from company B stressed on developing an automated tool to facilitate requirement engineers to prioritize requirements using MDRPM, as it can save lot of time. Representative from company A suggested that it would be a great advantage if MDRPM could use history data of company's previous projects and propose relevant bin to a requirement, it will assist requirement engineer in selecting the most appropriate bin for a particular requirement.

V.I.III Effectiveness and Efficiency

The MDRPM is found to be capable of producing an intended result and as it reduces the number of pair wise comparisons as compared to traditional AHP technique which make it more efficient. The model is based upon AHP which is known to be a most reliable and effective technique among all requirement prioritization methods [12]. The outcome of dynamic validation showed that it produced accurate results and in half time as compared to AHP. The representatives from both the companies rate it better than AHP in terms of effectiveness and efficiency.

VI. CONCLUSION

MDRPM will help to make right decisions regarding requirement prioritization. The validation was performed to assure that the model complied with the real needs of the industry. As MDRPM does not suggest fit for every organization, but is rather customizable. The customizable nature of the model is based on introducing new bins or modifying existing ones. MDRPM is validated through both static and dynamic validation, the results of static and dynamic validation shows that it can be practically implemented in the industry and the model is customizable based on the organization's individual goals and objectives. Numbers of comparisons are drastically reduced as compared to traditional AHP technique which was unusable once the requirements reaches over a particular threshold but in case of MDRPM it can handle large number of requirements. This research paper has also described the process of creating MDRPM, as well as the validation procedure employed in order to ensure that the model is useful for industry practitioners. The model could be useful for driving improvement efforts in requirements prioritization.

VII. REFERENCES

[1] Karlsson L, Dahlstedt G, Natt J, Regnell B, Persson A, "Challenges in Market-Driven Requirements Engineering - an Industrial Interview Study", In Proceedings of Eighth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02), Universitat Duisburg-Essen, Essen, Germany, 2002, PP 101-112.

[2] Gorschek T, "Requirements Engineering Supporting Technical Product Management". PhD Thesis No. 2006:01, ISBN 91-7295-081-1, Blekinge Institute of Technology, Ronneby, 2002.

[3] McGrath, Michael E, "Product Strategy for High Technology Companies". McGraw-Hill Professional Book Group, 2nd edition, 2000.

[4] Aurum A, Wohlin, "Aligning Requirements with Business Objectives: A Framework for Requirements Engineering Decisions", Proceedings Requirements Engineering Decision Support Workshop, Paris, France, 2005.

[5] Vähäniitty J, "Product Strategy Decisions in Small Software Product Businesses – A Framework and Experiences", Master's Thesis, Helsinki University of Technology, Institute of Software Business and Engineering, 2003.

[6] Davis A, "The Art of Requirements Triage", IEEE Computer, Volume 36, Issue 3, PP 42-49. [12] Meyer M, Lopez L (1995) Technology Strategy in a Software Products Company. The Journal of Product Innovation Management, Volume 12, Number 4, 2003, PP 294-306.

[7] Kostoff R N, Schaller R, "Science and Technology Roadmaps", IEEE Transactions on Engineering Management, Volume 48. No. 2. 2001, PP. 132-14.

[8] Wieringa R, Ebert C (2004) Guest Editors' Introduction: RE'03: Practical Requirements Engineering Solutions. IEEE Software Volume 21. Issue 2, 2004, PP 16-18.

[9] Karlsson, Lena, et al, "Challenges in Market-Driven Requirements Engineering - an Industrial Interview Study". In Proceedings of the Eight International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ '02). 2003, pp. 101-112.

[10] Carlshamre, Pär, "Release Planning in Market-Driven Software Product Development: Provoking an Understanding", Requirements Engineering. 2002, Vol. 7, 3, pp. 139-151.

[11] Gomes, Andrigo and Pettersson, Andreas, "Market-Driven Requirements Engineering Process Model – MDREP", Karlskrona: Blekinge Institute of Technology, 2007.

[12] Sommerville, Ian and Sawyer, Pete, "Requirements Engineering - A good practice guide", West Sussex: Wiley & sons, 2003. 0471974447.

[13] Lehtola, Laura and Kauppinen, Marjo, "Suitability of Requirements Prioritizations Methods for Market-Driven Software Product Development", Software Process: Improvement and Practice. 2006, Vol. 11, 1, pp. 7-19.

[14] Karlsson, Lena, et al, "Requirements engineering challenges in market-driven software development - An interview study with practitioners", Information and Software Technology. 2007, Vol. 49, 6, pp. 588-604.

[15] Brall, Aron and Gardner, Landis, "Marketing with Reliability and Maintainability: The Business Case of Hiring a Reliability Engineer When There are No R&M Requirements", Reliability and Maintainability, 2004 Annual Symposium - RAMS. 2004, pp. 480-484.

[16] Azar, Jim, Smith, Randy K and Cordes, Davis, "Value-Oriented Requirements Prioritization in a Small Development Organization", IEEE Software. 2007, Vol. 24, 1, pp. 32-38.

[17] Ebert, Christof, "Understanding the Product Life Cycle: Four Key Requirements Engineering Techniques", IEEE Software. 2006, Vol. 23, 3, pp. 19-25.

[18] Saaty T.,"The Analytic Hierarchy Process: Planning, Priority Setting, Resource, Allocation", McGraw-Hill, New York, 1980

[19] Saaty, L.T, "Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World", New Edition 2001, Publisher: RWS Publications, ISBN-13: 978-0962031786

[20] Paetsch F., Eberlein A., Maurer F.,"Requirements Engineering and Agile Software Development", Proceedings of the 12 IEEE International workshop, IEEE Computer society, 2003, pp 1-6

[21] Saaty T,"The Analytic Hierarchy Process: Planning, Priority Setting, Resource, Allocation", McGraw-Hill, New York, 1980.

[22] Gass S., Ripsaw T," Singular value decomposition in AHP", European Journal of Operational Research, 2004, Vol. 2004, No. 3, pp 573-584.

[23] Robson C, "Real World Research", 2nd Edition, Blackwell Publishing, Malden, 2002