

Cite as: Shehzad, K., Awan, M. D., Rizvi, S. S., & Khiyal, S. H. (2014). A hybrid technique based on standard SRS modules for software requirement prioritization. *Proceedings of Informing Science & IT Education Conference (InSITE) 2014* (pp. 279-294). Retrieved from <http://Proceedings.InformingScience.org/InSITE2014/InSITE14p279-294Shehzad0564.pdf>

A Hybrid Technique based on Standard SRS Modules for Software Requirement Prioritization

**Khurram Shehzad, M. Daud Awan, Sanam Shahla Rizvi, and
Malik Sikandar Hayat Khiyal**
Preston University, Islamabad, Pakistan

arizon45@yahoo.com, drdaudawan@preston.edu.pk,
s.s.rizvi@preston.edu.pk, m.sikandarhayat@yahoo.com

Abstract

Requirement prioritization is an important consideration. Normally solutions do not focus on some specific considerations like unique priority to a requirement and standard conformance in prioritization. Previously, some of ideologies of requirement prioritization have been designed to provide an end-to-end solution for system development life cycle. Ranking, n-based requirements and much more contain many ambiguities. AHP, Weiger's method, Kano's method, QFD are the latest requirement prioritization solutions that focus on quality to choose requirements. This study considers an overview of different prioritization techniques commonly practiced by software industry and their flaws/ambiguities. Study focuses on solutions those conform the SRS document a symbol of standard to be followed in software industry. Factors taken are cost, volatility and risk. In nature of requirement criticality, importance and optional be the part of the discussion. Finally, we testify the suggested solution with a case study of video renting system and find out results in graphical form.

Keywords: Requirement Engineering, Requirement Prioritization, Decision Tree, Video Renting System.

Introduction

Requirement Engineering (RE) is the process of accumulating, scrutinizing and sculpting software requirements in an orderly managed process, ambiguities and errors interlinked with each other (Yang et al., 2011). Graphical notations like prototypes use cases and descriptive models like story boarding are employed (Sutcliffe et al., 2011). Project complexity, insufficient requirements, volatility of requirements, and competitive market edges towards issues in requirement prioritization (RP) decision making (Mishra et al., 2008). Problems of requirement elicitation are

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

scope understandability, volatility. Early defects in requirements (functional and non-functional) leave serious operational consequences in design phase (Christel et al., 1992). Face-to-Face conversation used to elicit user's tacit knowledge and to illustrate software developer's knowledge is an iterative negotiation process (Ping et al., 2009). An expert to take requirement needs to solve the conflicts arise between two intersecting re-

quirements (Kulkarni, 2008). The Goal-Based Requirements Analysis Method uses goal topography to structure and organize such requirements information as scenarios, goal obstacles, and constraints (Sajid et al., 2010). Requirements uncertainty is expected to have a negative impact on the software developer performance as well (Srinivasan, 2009). Skilled team and professional attitude in RE process is required to decrease the rate of failure of projects in Pakistan (Hayat et al., 2010). Unnecessary requirements are the requirement 'gold plating (Mayer, 1997). Probabilistic Risk Analysis is widely used for quantitative risk assessment, while approaches like FMECA quantify risk into qualitative values: frequent, reasonably probable, occasional, remote, and extremely unlikely (Ansar et al., 2011). A deep focus is required to monitor deviation of functional requirements (FRs) from normal or ideal state to deviated or misinterpreted state in project (Sutcliffe, 1998). Poorly defined or missing requirements must be in focus and removed before the design level (WestFall, 2005). An adaptive selection process can support human effort to input information and domain knowledge in requirement (Perini et al., 2013). Another question lies in prioritization over time, i.e., prioritization of improved quality over the entire lifecycle for the product as opposed to prioritization in the beginning of the project (Aurum et al., 2011). Prioritization approaches are divided into two categories: methods that base on giving values to different values to requirements (pair wise comparisons, e.g. AHP) and negotiation approaches. There are also several other RP methods, which are not particularly designed for collaboration. Careful requirement elicitation and prioritization reduces 40% of cost and time rather implementing ambiguous requirements (Karlsson et al., 1997). Requirements are prioritized effectively in agile methodologies as compared to traditional processes due to extensive user involvement (Racheva et al., 2010). Machine learning techniques induce requirements ranking approximations from the acquired data (Avesani et al., 2005). Quantitative data referred to cost and value used to compute the set of most promising requirement rankings (Avesani et al., 2004). RP aims at ranking the requirements to trade off user priorities and implementation constraints (Tonella et al., 2010). There is an increasing need for methods capable of prioritizing candidate requirements to calculate the cost-value ratio for the requirement that represents, total cost is not solution, and to cut down budget each requirement must have its own cost of implementation (Mohamed et al., 2008).

Good aspect of this article is considering software requirement specification document (SRS). Considering this can divide the requirements in to different sections like NFRs, FRs, design requirements, interface requirements and further more specifications which are included in the document. With this consideration, development can be restricted to more focused prioritization consideration by the user.

In Section II, literature review is discussed. Section III proposes the RP technique (RPT) methodology. Implementation of proposed idea is discusses in Section IV. Conclusion and future work of the paper is presented in Section V.

Literature Review

Ambiguous, incomplete and missing requirements should have been written in formal and precise without formulating a track to organize the work, most of the effort may loss and have to pay back in terms of cost plus time.

Feasibility Study

Feasibility study starts when the stakeholder/user countenances the dilemma in legacy system and hence be acquainted with a need for developing/improving the system. Revealing the stableness and vulnerability of a legacy system or suggested endeavor impartially and sensibly, contingencies and hazards as conferred by the experiments and context, the assets need to accomplish the goals, and finally the outlook for future success. Time and money are two considerable resources for change. An accurate feasibility consists of many domains such as finance, marketing, work

evaluation, circumstances, causes of failures, difficulties in execution in plans and overall management. Hence, in feasibility we have to consider whether the project is in a position to enhance, one has to provide technical considerations of stakeholder's desires. Failure of projects in software industry is mostly blamed on the RE process. In RE where gathering and analyzing requirements is important, another important consideration is RP. Stakeholders must have some interest in some part of their application. Which has more importance than others? Dependent requirements should also be in consideration. So considering all these, a solution is to prioritize requirements. Prioritization is not a common practice in whole process. A small precedence is provided to prioritization. This must have to consider in important terms.

Prioritization Techniques

There are numerous RPTs like AHP, Ranking, VOP, CV, QFD (Aurum et al., 2011; Karlsson et al., 1997; Avesani et al., 2004; Bebensee et al., 2010) used practically in different projects. Most of them focus on single user perspective. Due to these considerations, there are so many problems faced during the prioritization such as time, cost, volatility (Karlsson et al., 1997; Racheva et al., 2010; Avesani et al., 2005). Some Selected techniques in detail are given in Table 1.

Aspect	Prioritization Technique	Perspective
Strategic importance	AHP	Product Manager
Customer importance	100-dollar / Top-ten1	Customers
Penalty	AHP	Product Manager
Cost	100-dollar	Developers
Time	Numerical Assignment (7)	Project Manager
Risk	Numerical Assignment (3)	Requirements Specialist
Volatility	Ranking	Requirements Specialist

Set prioritized feature list

In this technique, there are three types of requirements; baseline, optional and new version. Features are assigned a term like critical, important, useful, high, low medium to get a standard against three considerations of priority, effort and risk. Technique's (Ansar et al., 2011; Bebensee et al., 2010) have advantage to have descriptive consideration of a feature. They can divide and conquer the requirements in functional, optional and future considerations; and can prioritize requirement in consideration of future requirement. However, factors as time and cost are not considered, non- FR are ignored, but presented that modular division of requirements is better than sequential.

A prioritization scale

A scale of single RPT is a simple technique to get a quick solution for the prioritization. Let us check its advantages and disadvantages of (Karlsson et al., 1997; Bebensee et al., 2010) in short description. It is a quick method and singular RPT. Imposes various limitations like it cannot entertain N requirements simultaneously, time delay by analyzing one by one and it cannot compare between two high priority requirements.

Numerical assignment

In numerical assignment, there must be some value related to the mark assigned to a requirement as shown in Figure 1. Requirements relate to critical/High, standard/Medium, optional/Low marks for a reliable classification. We can assign numerical values to such classifiers for requirements like critical/High has 9 number, standard/Medium has 6 and optional/Low has 3. Some advantages of (Karlsson et al., 1998) are; it has fixed numerical values against classifiers and it can separate different groups. However, assigning the same value to different requirement cannot distinguish between the same groups and cannot prioritize in a same group.


	Important	Not Important
Urgent	High Priority must be included in next release	
Not Urgent	Medium Priority must be included, but can wait for later release	Low Priority would be nice to have if we can fit it in

Figure 1: Numerical Assginent [23]

Ranking

It is similar to numerical assignment but based on an ordinal scale the requirements are ranked without ties in rank. Like the most important requirement is ranked 1 and the least important is ranked n (for n requirements). Some advantages of (Bebensee et al., 2010; Karlsson et al., 1998) are: it has various levels and each requirement has its own numerical value to distinguish. However, it has no modular distinguish, focused on a single aspect and no criteria for the categorization.

Sample Quantification Matrix to Prioritize n-Requirements

It is similar to numerical assignment, but based on an ordinal scale the requirements are ranked getting 10-15 members of a team to agree on the rankings of as many as 50 items can be a laborious, frustrating, time-consuming challenge (Karlsson et al., 1997; Lubars et al., 1993).

Three administrative tasks need to occur before establishing priorities:

- Display all requirements in plain view of all members of the cross-functional team.
- The team needs to closely scrutinize the list to merge any similar requirements and to reword requirements needing additional clarity. Revisions to the list must be done with the consent of the entire team.
- Finally, the new list of requirements with designations has been finalized. These designations can be converted to numerical values (for example, high = 3, medium = 2, low = 1) to develop a quantification matrix like the sample shown in the Figure 2. The requirements can then be ranked according to their total values.

Advantages of this technique are: unique numerical values assigned by different team members, marking requirement individually, marking single requirement from multiple members, average value calculation and N requirements can be entertained simultaneously. The limitations are; no modular approach, reparative work by many members and no taking average criteria.

	Team Member #1	Team Member #2	Team Member #3	Team Member #4	Team Member #5	Totals
Requirement #1	2	3	1	2	2	10
Requirement #2	1	3	2	3	2	11
Requirement #3	1	2	1	2	3	9
Requirement #4	2	3	2	3	2	12
.
.
.
Requirement #n	1	2	1	2	2	8

Figure 2: Sample Quantification Matrix to Prioritize n-Requirements [26]

Proposed Requirement Prioritization Technique

Proposed RPT is conforming IEEE SRS standard as follows.

Matrix A: Structuring Matrix

Redesign structuring matrix as per project requirement document that cannot be framed into this matrix until it complies with a standard SRS. The results of following two techniques using Matrix A are given in Table 2.

- Set prioritized feature list (Karlsson et al., 1997).
- Sample quantification matrix to prioritize N-requirements (Karlsson et al., 1998).

Some of the features included are not part of these two techniques. In consideration of features involved, time, cost, volatility, risk and effort. Priority is replaced with nature is how a reader takes level of requirements. We have considered all those features which comply with a SRS Standard. We defined type of requirements like functional, non-functional, design constraints and interface. Functional requirements will carry all modules of project. There are certain fixed considerations for non-function requirements like design constraints and interfaces are provided. Index of requirements is the unique number for acknowledgement that N number of requirements will be entertained. Previously, we just placed requirements in a prioritization technique. Here, we confirm RP to a standard form of requirements, i.e. SRS. Now how and what to assign to features?

- Precedence wise nature can be one of these; critical, urgent, important, normal, optional.
- Each of these parameters time, cost, volatility, effort and risk must have a value high, medium and low.

Matrix B Value Assignment Matrix

Column “value assigned” contains the possible values of the combination. Values are assigned according to combination matching from Matrix B. Value column is assigned numerical value as ranking technique. It is assigned numbers for possible combinations of six parameters considered in terms of high, medium and low. Parameters are time, cost, volatility, effort and risk. See Table 3 for Matrix B. Figure 3 represents a decision tree for Matrix B. It seems a hectic process to search from a large table. To avoid this, we consider a machine learning technique where average values from Matrix B can be represented as a decision tree and can be manipulated through this tree.

Table 2: Matrix A

Requirements			Parameters						Value from Matrix B
Type of Requirement	Features/modules	Req. Index	Nature	Effort	Risk	Time	Cost	Volatility	
FR's	Module A	A1							
	Module B *All Modules	B1							
NFR's	Usability	C1 C2 ...							
	Performance *All NFR's	D1 D2 ...							
Design Constraints	Software constraints	E1 E2 ..							
	Hardware constraints	F1 F2							
Interfaces	User Interface	G1 G2							
	Hardware Interface	H1 H2							

Table 3: Matrix B

Features					Nature				
Effort	Risk	Time	Cost	Volatility	Optional	Normal	Important	Urgent	Critical
					Value assigned by ranking				
High	High	High	High	High	243	486	729	972	1215
High	High	High	High	Medium	242	485	728	971	1214
High	High	High	High	Low	241	484	727	970	1213
High	High	High	Medium	High	240	483	726	969	1212
Low	Low	Low	Medium	Low	4	247	490	733	976
Low	Low	Low	Low	High	3	246	489	732	975
Low	Low	Low	Low	Medium	2	245	488	731	974
Low	Low	Low	Low	Low	1	244	487	730	973

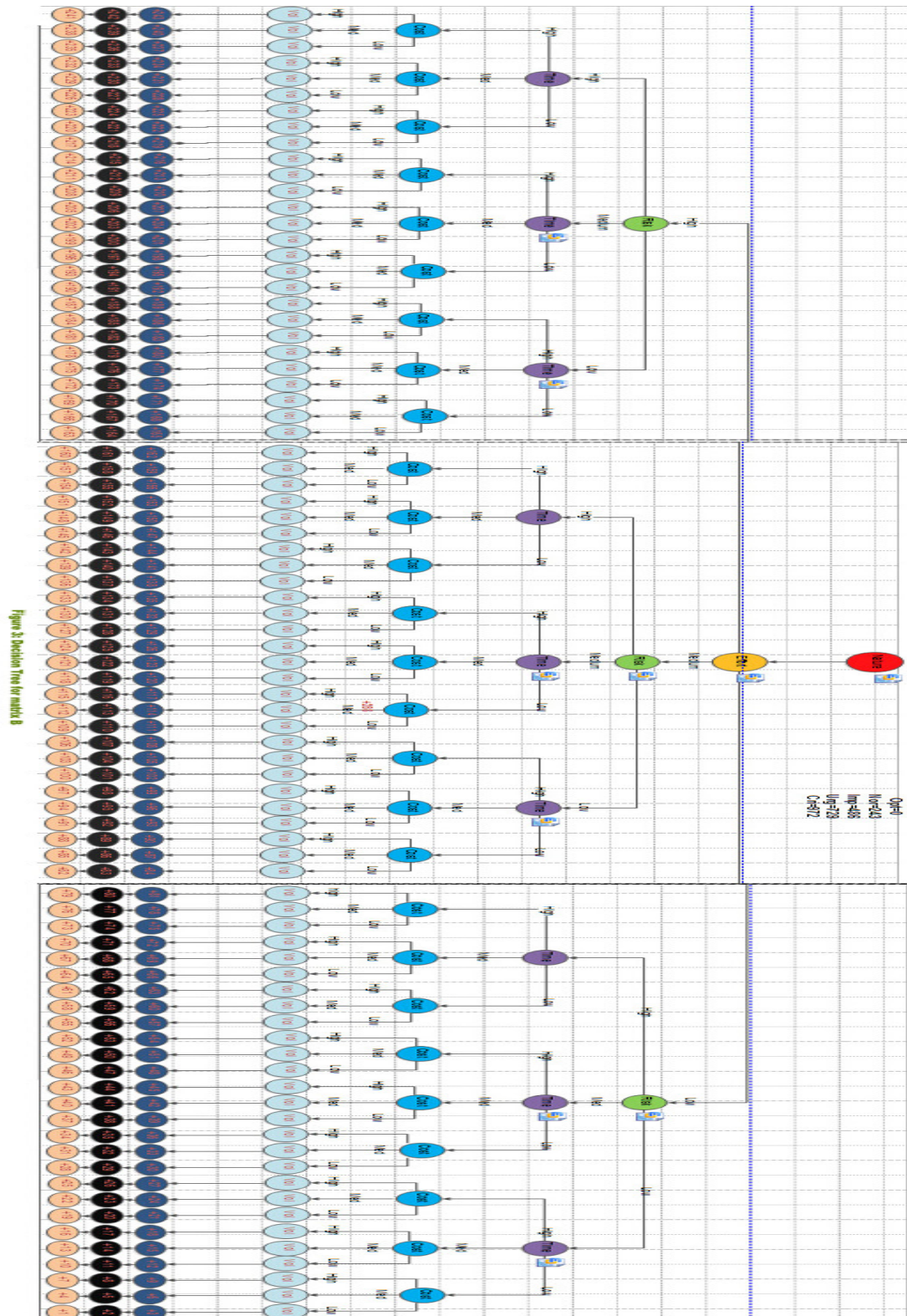


Figure 3: Decision Tree for matrix B

Figure 3: Decision Tree for Matrix B

Algorithm

Algorithm for Decision tree is defined as below:

Function PriReqTree (A[243]: an array contacting values of leaf nodes,

Min: Minimum index of node,

Max: Maximum index of node,

LN: Leaf_Node selected in tree,

UV: Upper_Value of node in tree,

LV: Lower_Value of node in tree,

NV: Nature_Value Selected in tree,

) returns a decision tree;

Begin

 If NV is not empty

 If NV is equal to “Optional” assign 0 to NV and return NV;

 If NV is equal to “Normal” assign 243 to NV and return NV;

 If NV is equal to “Important” assign 486 to NV and return NV;

 If NV is equal to “Urgent” assign 729 to NV and return NV;

 If NV is equal to “Critical” assign 972 to NV and return NV;

 End

//We have 5 levels of tree to pass on i.e. effort, risk, time, cost and volatility. Therefore, we will consider five iterations:

 For Loop for five iterations

 Val is a string assign on each loop with the value of High, Medium or Low

 If Val is equal to “High”

 Max = max/3

 LV = UV – max +1

 End

 If Val is equal to “High”

 Maximum = Maximum / 3;

 Lower_Value(LV) = Upper_Value(UV) – Maximum + 1;

 End

 If Val is equal to “Medium”

 Maximum = Maximum / 3;

 Minimum = Lower_Value(LV);

 Upper_Value(UV) = Lower_Value(LV) + Maximum;

 Lower_Value(LV) = Upper_Value(UV) – Maximum;

 End

 If Val is equal to “Low”

 Maximum = Maximum / 3;

 Upper_Value(UV) = Lower_Value(LV) + Maximum - 1;

 End

 If last iteration of loop is true


```

        If Upper_Value is equal to Lower_Value
            Assign value of array A at index [Upper_Value]
            To Leaf_Node (LN);
        End
    End
End
Return Leaf_Node + Nature_Value;
// gives priority of a specific requirement
EndPriReqTree;

```

This algorithm is representation of decision tree shown in Figure 3 and corresponding of Table 3. Algorithm traversed and verified the information retrieved and given by Table 3 and Figure 3.

In description of algorithm, a defined set of array containing leaf nodes values up to 243 according to solution. Core logic is to assign the nature numerical value based on the text value in NV. Then based on the high, medium or low value, we have defined the formulas to search beneath the tree and pick the node which is leaf node. Finally, values of leaf node and nature node are added that can be verified by Table 3.

Matrix C: Evaluation Matrix

“Value assigned” column of Matrix A is filled according to Matrix B. Now take this column to map it in Matrix C against each member and take mean of each requirement of N numbers to get single value of each requirement, see Table 4.

Stepwise description

Stepwise diagrammatical explanation can be viewed in Figure 4. Theoretical description of diagram is as follows.

Step 1: Provide a copy of Matrix A after adding all FRs to team members more over this matrix is derived from standard SRS document. It contains functional, non-functional, design constraints and interface requirements.

Step 2: Each member shall fill the criteria for each requirement and assign a factor to each feature provided in Matrix A. To assign nature as critical, urgent, important, normal and optional, above is the given fixed order for nature of requirements. To assign other features like time, risk, cost, effort and volatility, we fix following values as among of these i.e. high, medium and low.

Step 3: Now look in Matrix B and provide numerical values in value column of Matrix A.

Step 4: Provide value column of each member against the requirements mentioned.

Step 5: Get the mean value of each requirement provided against in Matrix C.

Step 6: Sort the mean value column in Matrix C.

Step 7: Plot graph for presentation.

Table 4: Matrix C

Requirements			Value column from matrix A of each member						Mean
Type of Requirement	Features/modules	Req. Index	Member 1	Member 2	Member 3	Member 4	Member n	Value
FRs	Module A	A1							
	Module B *All Modules	B1 .							
NFRs	Usability	C1 C2 ...							
	Performance *All NFR's	D1 D2 ...							
Design Constraints	Software constraints	E1 E2 ...							
	Hardware constraints	F1 F2							
Interfaces	User Interface	G1 G2							
	Hardware Interface	H1 H2							

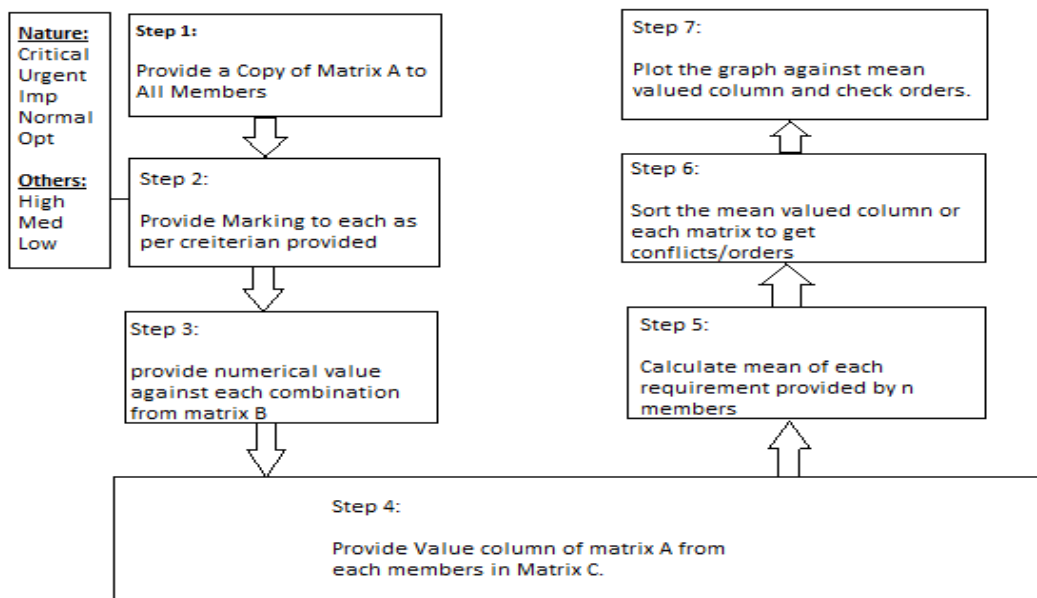


Figure 4: Step Wise Description

Performance Evaluation

For evaluation purpose, we selected a video renting system (VRS). After formalizing whole requirement gathering process, SRS document has been finalized by the stakeholders. RP stands in the middle of requirement analysis and the design phase. Stakeholders (software developers/team lead) must prioritize requirements before workflow decision. We applied our technique on VRS, and plotted a graph as shown in Figure 5 where values are extracted from Table 5 as follows.

Table 5: Requirement Prioritization Table for VRS Project

Requirements			Value column from matrix A of each member					Mean
Type of Requirement	Features/modules	Req. Index	Member 1	Member 2	Member 3	Member 4	Member n	Value
FRs	Module A	A1	720	523	741	988	183	631
		A1.1	943	491	258	254	426	474.4
		A1.2	1195	623	963	345	759	777
	Module B	B1	667	229	485	168	689	447.6
	Module C	C1	1205	548	596	756	356	692.2
	Module D	D1	962	500	951	587	214	642.8
		D1.1	652	712	753	654	547	663.6
	Module E	E1	1094	362	456	159	785	571.2
	Module F	F1	851		987	244	452	506.8
	Module G	G1	352	257	369	125	159	252.4
	Module H	H1	712	189	123	459	753	447.2
		H1.1	955	399	147	752	456	541.8
		H1.2	1198	488	547	136	789	631.6
		H1.3	1207	886	965	146	123	665.4
		H1.4	401	682	851	167	155	451.2
	Module I	I1	958	589	358	349	326	516
I2		697	995	751	729	467	727.8	
I3		379	475	953	183	215	441	
Module J	J1	728	751	759	159	658	611	
Module K	K1	728	689	351	753	489	602	
NFRs	Usability	L1	670	244	650	456	597	523.4
	Supportability	M1	111	654	983	852	124	544.8
		M2	87	258	187	359	102	198.6

Software Requirement Prioritization

	Frequently Asked Questions	N1	244	951	329	751	352	525.4
	Online Customer Service	O1	636	753	781	153	124	489.4
		O2	576	456	582	759	458	566.2
	Interface	P1	580	741	693	741	410	633
		P2	592	963	471	852	777	731
	Resource	Q1	277	852	882	963	458	686.4
		Q2	264	410	1054	987	658	674.6
	Security	R1	1215	624	987	541	452	763.8
		R2	1210	369	1158	236	652	725
	Reliability	S1	958	258	669	755	412	610.4
		S2	820	147	724	12	369	414.4
	Maintainability	T1	457	987	1199	46	874	712.6
		T2	463	478	201	167	456	353
	Design Constraint	Software Constraint	U1	14	698	964	796	321
U2			1	214	354	459	789	363.4
Interfaces	User Interfaces	V1	369	632	496	743	852	618.4

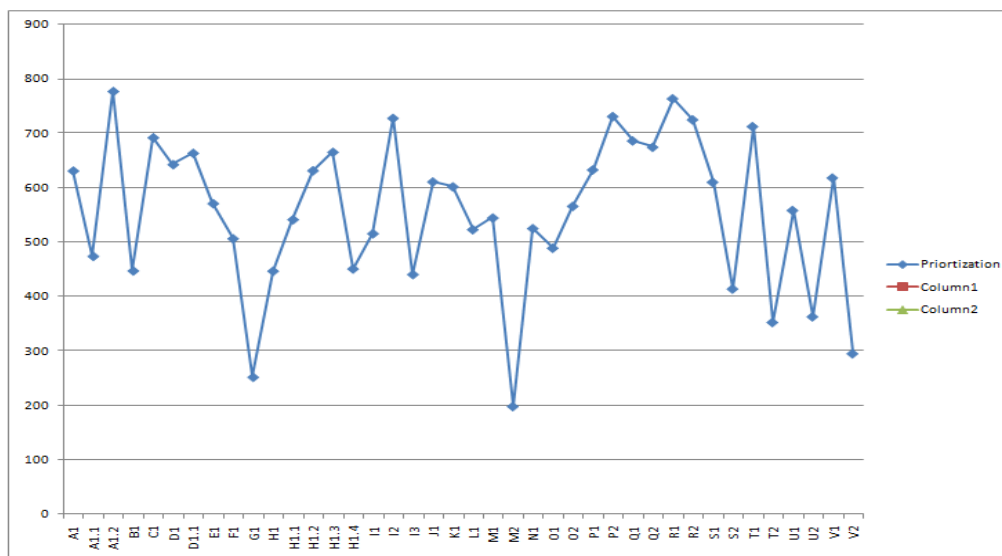


Figure 5: Requirement Prioritization Graph for VRS Project

Requirements are assigned indexes and Matrix A is formed. Copies of matrix A are provided to 5 team members each. The values assigned to each matrix as per combinations of parameters from Matrix B. Finally, value column from each Matrix A of members has been placed in Matrix C. Mean value is calculated and graph is plotted against those. Table 5 can be arranged in ascending order to get numerical values. In evaluation of presented methodology, a comprehensive comparison can show this a better solution. This technique covers following flaws which are not present before in previous studies.

- No limited numbers of requirements to prioritize i.e. 1 to 10.
- Not dependent on the number of requirements to prioritize i.e. from 1 to n.
- Avoids the repetition of priority within the requirements order i.e. many requirements are High, Medium, or Low level.
- Not focusing some specific attributes of nature of requirements.
- Not only focuses nature but also covers five important factors that may affect a requirement priority.
- Unlike common practice of formulating SRS document, it will enforce stakeholders to conform the standards.
- Includes the multiple RP views from different stakeholders.
- Prioritize each and every requirement as well as the module based RP due to consideration of SRS documentation standard.
- Hybrid solution of different common practiced solutions to prioritize requirements.
- Follow specific steps to process.
- Enforce RP as a common practice step in RE phase.

Summary & Future Work

We focused the execution of RP as a common practice phase of RE process. Previously, RP is not much in practice. Even if it is considered, it has not given much importance. In earlier studies, numerous techniques are available but each has a different flaw in consideration. We focused on solutions, considering more than one requirement at a same priority. Some of them do not consider the stakeholders some involve but have not any authentic way to cover all. This problem statement arises in our mind to provide a solution to software industry.

In this paper, the prose of foremost RP considerations described with RE aspect. A significant framework for RP proposed and evaluated using VRS. Paper clearly considers more pinching factors and complex nature of requirements. We conform that our proposed prioritization technique follows the IEEE SRS standard document.

Future work for this can be involvement of RP in design. View based consideration of RP, Scenario based RP and module based RP where dependency of requirements on each other is so much involved that no single requirement could be focused, or taken a start. A bit module based prioritization has been considered in proposed solution. However, some of complexities have been left to focus for future consideration.

Machine learning techniques like classification and regression make automated RP tool by answering questionnaires on specific requirements. Except might be lengthy or hectic. Surveys have proven that project failure is the cause of bad/poor/ambiguous requirements. In Pakistan, the ratio of project failure is much more than other countries due to lack of consideration of RP in analysis phase.

References

- Asnar, Y., Giorgini, P., & Mylopoulos, J. (2011). Goal-driven risk assessment in requirements engineering. *Requirement Engineering*, 16(2), 101-116.
- Aurum, A., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., & Feldt, R. (2011). Prioritization of quality requirements: State of practice in eleven companies. *Requirements Engineering Conference (RE)* (pp. 69 - 78). Trento: IEEE Society .
- Avesani, P., Bazzanella, C., Perini, A., & Susi, A. (2004). Supporting the requirements prioritization process. A machine learning approach. *16th International Conference on Software Engineering and Knowledge engineering (SEKE)*, 1, pp. 23 - 32. Banff, Alberta (CAN).
- Avesani, P., Bazzanella, C., Perini, A., & Susi, A. (2005). Facing scalability issues in requirements prioritization with machine learning techniques. *Requirements Engineering* (pp. 297 - 305). IEEE Computer Society.
- Bebensee, T., Weerd, I. v., & Brinkkem, S. (2010). Binary priority list for prioritizing software requirements. *Requirements Engineering: Foundation for Software Quality*, 6182, 67-78.
- Christel, M. G., & Kang, K. C. (September 1992). *Issues in requirements elicitation, Technical report*. Carnegie Mellon University, U.S. Department of Defense. Pittsburgh, Pennsylvania: Software Engineering Institute.
- Hayat, F., Ali, S., Ehsan, N., Akhtar, A., Bashir, M. A., & Mirza, E. (2-5 June 2010). Requirement elicitation barriers to software industry of Pakistan (Impact of cultural and soft issues). *IEEE International Conference of Management of Innovation and Technology (ICMIT)* (pp. 1275 - 1278). Singapore: IEEE Publisher Society.
- Karlsson, J., & Ryan, K. (Sep/Oct 1997). A cost-value approach for prioritizing requirements. *Software, IEEE*.14, pp. 67 - 74. IEEE Computer Society.
- Karlsson, J., Wohlin, C., & Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, 39(14-15), 939-947.
- Kulkarni, V. (12-14 Dec, 2008). A conceptual model for capturing stakeholders' wish list. *Computer Science and Software Engineering* (pp. 275-278). Wuhan, Hubei: IEEE Computer Society.
- Lubars, M., Potts, C., & Richter, C. (1993). A review of the state of the practice in requirements modeling. *Requirements Engineering*, (pp. 2 - 14). San Diego, CA.
- Meyer, B. (April 13, 1997). *Object-oriented software construction* (2nd ed.). Prentice Hall.
- Mishra, D., Mishra, A., & Yazici, A. (2008, Aug 4 - 6). Successful requirement elicitation by combining requirement engineering techniques. *International Conference Applications of Digital Information and Web Technologies (ICADIWT)*, 258 - 263.
- Mohamed, S. I., El-Maddah, I., & Wahba, A. M. (July 14-17, 2008). Criteria-based requirements prioritization for software product management. *International Conference on Software Engineering Research & Practice, SERP*, 2. Las Vegas Nevad.
- Perini, A., Susi, A., & Avesani, P. (2013). A machine learning approach to software requirements prioritization. *IEEE Transactions on Software Engineering*, 39(4), 445-461.
- Ping, W. J., De-yi, H., & Dan, W. (26 - 28 Dec, 2009). Knowledge conversion in software requirement elicitation. *Information Science and Engineering (ICISE)*, (pp. 2328 - 2331). Nanjing.
- Racheva, Z., Daneva, M., Sikkil, K., & Herrmann, A. (2010). Do we know enough about requirements prioritization in agile projects: Insights from a case study. *Requirements Engineering Conference (RE)* (pp. 147 - 156). Sydney: IEEE computer society.
- Sajid, A., Nayyar, A., & Mohsin, A. (2010). Modern trends towards requirement elicitation. *NSEC '10 Proceedings of the 2010 National Software Engineering Conference* (p. Article 9). New York: ACM Digital Library.

- Srinivasan, S. S. (July 10-12, 2009). Performance under requirements uncertainty: A personality perspective. *Pacific Asia Conference on Information Systems, PACIS*, (Paper 84). Hyderabad, India.
- Sutcliffe, A. (1998). Scenario-based requirements analysis. *Requirements Engineering*, 3(1), 48-65.
- Sutcliffe, A., Thew, S., & Jarvis, P. (2011). Experience with user-centred requirements engineering. *Requirements Engineering*, 16(4), 267-280.
- Tonella, P., Susi, A., & Palma, F. (7-9 Sept. 2010). Using interactive GA for requirements prioritization. *Search Based Software Engineering (SSBSE), Second International Symposium*, (pp. 57 - 66). Benevento.
- WestFall, L. (2005, September). Software requirements engineering: What, why, who, when, and how. *Software Quality Professional*, 7(4), 17-26.
- Yang, H., Roeck, A., Vincenzo, G., Willis, A., & Nuseibeh, B. (2011). Analysing anaphoric ambiguity in natural language requirements. *Requirements Engineering*, 16(3), 163-189.

Biographies



Mr. Khurram Shehzad is candidate of PhD in Software Engineering at Preston University, Islamabad.



Prof. Merit. Dr. Mohammad Daud Awan (Salam Teacher) Dean Faculty of Computer Sciences, Preston University, Islamabad, born at Mansehra, Pakistan. He is full Professor of Computer Science, since 1988 and Meritorious Professor of Computer Science, since 2000. He served as Vice Chancellor, Hazara University, Mansehra (2002 - 2006) and Allama Iqbal Open University (AIOU) (Acting) from 2006 – 2009, Founding Director, Computer Centre, AIOU, Founding Chairman Math's, Stats and Computer Science Deptt. AIOU, Secretary University Grants Commission (UGC) / HEC (1993 – 1995), Director CASDU, UGC (1989 – 1993), Senior Academician of Oslo, Norway (1977 - 1988), Project Evaluator, World Bank Team, 1988 and Consultant GTZ/GIZ. His area of interest is Artificial Intelligence, Software Engineering, Packages, Human Computer Interaction; Programming and Computer Science Research. He has more than 100 research reports and publications to his credit and supervised MPhil and PhD Scholars.



Dr. Sanam Shahla Rizvi received the B.C.S. degree in Computer Science from Shah Abdul Latif University, Khairpur Pakistan, in 2003, and the M.C.S. degree in Computer Science from KASBIT, Karachi Pakistan, in 2004, and M.S. degree in Computer Science from Mohammad Ali Jinnah University, Karachi Pakistan, in 2006, and Ph.D. degree in Information and Communication Engineering from Ajou University, Suwon, South Korea, in 2010. She is currently working as assistant professor at Department of Computer Sciences at Preston University, Islamabad, Pakistan. Her research interests include flash memory storages,

data management, database systems, indexing structures, and wireless sensor networks.



Dr. Malik Sikandar Hayat Khiyal born at Khushab, Pakistan, is currently Professor of Faculty of Computer Science, Preston University, Islamabad. He remained Chairman Department of Computer Sciences and Software Engineering in Fatima Jinnah Women University, Pakistan from 2007 to 2012 and in International Islamic University, Islamabad from 2002 – 2007. He served Pakistan Atomic Energy Commission for 25 years (1978 - 2002) and continuously was involved in different research and development projects of the PAEC. He developed software for underground flow and advanced fluid dynamic techniques. He was also associated with teaching at Computer Training Centre (PAEC) and International Islamic University. His areas of interest are

Numerical Analysis, analysis of Algorithms, Theory of Automata and Theory of Computation. He has more than hundred and forty research publications to his credit in National and International Journals and Conference proceedings. He has supervised three PhD and more than more hundred and fifty research projects at graduate and postgraduate levels. He is member of SIAM, ACM, Informing Science Institute, IACSIT. He is associate editor of IJCTE, IJMO, JACN, LNSE and Coeditor of the journals JATIT and International Journal of Reviews in Computing. He is reviewer of Journals, IJCSIT, JIISIT, IJCEE and CEE of Elsevier.