Master Thesis

Software Engineering

July 2012

A HYBRID TECHNIQUE FOR REQUIREMENTS PRIORITIZATION

By

Aaqib Iqbal

Reg # 212-FBAS/MSSE/F08, DCS&SE, IIUI

Supervised By

Mr. Shahbaz Ahmed Khan Ghayyur

Assistant Professor, DCS&SE, IIUI

Department of Computer Science & Software Engineering
Faculty of Basic & Applied Sciences
International Islamic University
Islamabad, Pakistan.

1. Hybrid computers ; architecture
2. Hybrid computer ; system design

International Islamic University, Islamabad
Faculty of Basic & Applied Sciences
Department of Computer Science

Dated: *July* 19, 2012

FINAL APPROVAL

It is certified that we have read the thesis, entitled **"A HYBRID TECHNIQUE FOR REQUIREMENTS PRIORITIZATION"**, submitted by Aaqib Iqbal Reg. No. 212-FAS/MSSE/F08.It is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for MS Degree in Software Engineering.

PROJECT EVALUATION COMMITTEE

External Examiner:
Dr. Muhammad Umer Khan
Assistant Professor, Department of Mechatronics,
Faulty of Engineering, Air University,
Islamabad

Internal Examiner:
Mr. Imran Saeed
Assistant Professor, DCS&SE, IIUI,
Department of Computer Science,
Faculty of Basic and Applied Sciences,
International Islamic University,
Islamabad, Pakistan.

Supervisor:
Mr. Shahbaz Ahmed Khan Ghayyur
Assistant Professor, DCS&SE, IIUI,
Department of Computer Science,
Faculty of Basic and Applied Sciences,
International Islamic University,
Islamabad, Pakistan.

Dedicated to

HAZRAT MUHAMMAD (sallal laho alaihi wasallam)

& To

My Loving Parent & Family

Who are a source of courage, honesty and love,
without their prayers it was not possible for me to
complete this research

& To

Precious Friendship

That has made us laugh, held us when we cried
and always, always, be among us

Declaration

I hereby declare and affirm that this thesis neither as a whole nor as part thereof has been copied out from any source. It is further declared that I have completed this thesis entirely on the basis of my personal effort, made under the sincere guidance of our supervisor. If any part of this report is proven to be copied out or found to be a reproduction of some other, we shall stand by the consequences. No portion of the work presented in this report has been submitted in support of an application for other degree or qualification of this or any other University or Institute of learning.

Aaqib Iqbal
212-FAS/MSSE/F08

ACKNOWLEDGEMENTS

I will always remain grateful to Allah for his abundant blessings during the course of this study. As I move towards the final destination of this remarkable journey of learning, I acknowledge the invaluable contributions of my teachers, family and friends.

This endeavor would not have been possible without the guidance and support of my Research Supervisor Mr. Shahbaz Ahmed Khan Ghayyur. Words are inadequate to express my profound sense of gratitude to him for his patience and support during the course of this study. His suggestions and comments have helped me immensely to be an independent researcher. He was a true mentor who kept motivating me whenever I was slowing down in the race.

I am deeply indebted to all the software professionals, from whom the data for this study was collected, for their co-operation and patience in responding to the research questionnaire.

I remember my family with gratitude, for the love and encouragement I received from them. For my parents, it is a dream coming true. My wife supported me with hearts and minds and all my colleagues who has given me constant support during the period of my study.

Lastly, I would like to acknowledge the contribution of those whose names have not been mentioned but who have, nevertheless, played their part in making this research work successful.

Aaqib Iqbal (212-FBAS/MSSE/F08)

PROJECT IN BRIEF

Project Title: A Hybrid Technique For Requirements Prioritization

Organization: International Islamic University, Islamabad, Pakistan.

Objective: The focus of this research is to provide engineers with better understanding of limitations present in existing prioritization technique and how these gaps and limitations can be eliminated (minimized) using the proposed technique.

Undertaken By: Aaqib Iqbal (212-FBAS/MSSE/F08)

Supervised By: Mr. Shahbaz Ahmed Khan Ghayyur
Assistant Professor
Department of Computer Science & Software Engineering
Faculty of Basic & Applied Sciences,
International Islamic University, Islamabad.

Research Area: Software Requirement Engineering

ABSTRACT

Requirement prioritization is an important part of development process. Literature shows us that Requirements prioritization plays important role in decision making whilst development process and it has also been a crucial step in software development process in terms of minimizing software failure. Researchers have proposed many prioritization techniques and there are many drawbacks and limitations associated with these techniques. One of the major disadvantages is that the current prioritization techniques do not consider both technical and business perspective associated with each and every requirement. This study will help engineers to better understand limitations presented in existing prioritization technique, how these gaps and limitations can be eliminated (minimized) using the proposed technique. Proposed technique will help engineers focus on both business and technical perspective during prioritization process. I've performed two experiments to prove the effectiveness of proposed technique.

A HYBRID TECHNIQUE FOR REQUIREMENTS PRIORITIZATION

Thesis for the degree of Master of Studies in Software Engineering
Faculty of Basic & Applied Sciences, Department of Computer Science & Software Engineering,
International Islamic University Islamabad

Table of Contents

List of Figures

List of Tables

Chapter # 1

INTRODUCTION

1. Introduction

The software business community is fully involved in innovative development due to the rapid increase in the usage of software applications. The very existence of innovation induces the factor of complexity in designing such systems effectively and this complexity comprises in terms of the lack of clear objectives or user requirements which have a deeper effect on the design of the system and it becomes hard to develop such systems or products. Such complexities can be controlled by giving due consideration to the user or stakeholder's requirements. A set of user requirements is selected on the basis of importance of the requirements with the help of a requirements prioritization method. The requirements prioritization plays a vital role in decision making (Aurum, 2003) for elimination of the complexities caused by unclear requirements or objectives. Requirements prioritization is itself a process of complex decision making (Carlshamre, 2002, Karlsson, 1996, Lehtola et al. 2004. Moisiadis, 2002) so for proper implementation of a prioritization technique there is the need of professional skills and sound domain knowledge (Karlsson et al. 2004).

1.1 What is Requirement Prioritization?

Requirement Prioritization is "An activity during which the most important requirements for the system (release) should be identified" (Sommerville 1996). Requirement prioritization facilitates requirement engineering process. It helps engineers make crucial decisions about requirements in a software development process. Requirement prioritization process is used to determine which candidate requirements should be included in a certain release, for this purpose different techniques are used (Aaqib I, 2009) such as AHP (Analytic Hierarchy

Process)(Saaty, Thomas L 1994), a cost-value approach (Karlsson, 1997), SERUM (Greer D, 1999), VOP (Value-Oriented Prioritization) (J.Azar, 2007) etc.

1.2 Factors / Criterion

Requirement prioritization techniques use different approaches and consider different factors (Henry, 1993) or criterion (Barragans, 2005) for prioritization e.g. cost, value, benefit, risk etc (Aaqib I, 2009). On the basis of these aspects the unimportant requirements are not given much importance rather they are totally rejected and value added requirements are added in the requirement set in order to develop a system of high quality. There are two perspectives of requirements, technical perspective that shows the technical importance of requirement which can be measured in terms of factors like cost, value, benefit, risk etc and then there is business perspective of requirement that shows the business value of requirement which can be measured in terms of core business values like customer satisfaction, sales, marketing, strategic, integrity etc. For effective and optimal requirement prioritization engineer should consider both technical and business perspective of requirements. Below is a brief explanation of prioritization factors that I will use later in this research for requirements prioritization:

1.2.1 Technical Factors

1.2.1.1 Cost

Cost is the cost of successfully implementing the candidate requirement. In practice, software developers often calculate cost purely in terms of money (Lehtola, Kauppinen, 2004). The implementations of different requirements have different development or lifecycle costs. Given limited budgets, cost can be an important and even overriding factor when prioritizing requirements. Thus, the highest priority requirements may be those that the project can afford to implement first. (J.Azar, 2007, Aaqib I, 2009).

1.2.1.2 Benefit

Priorities provide management and engineering with a rough estimate of the benefit of the different requirements, which is useful when performing cost/benefit analysis of the requirements to determine whether to expend limited project resources in preparation for requirements negotiation (J.Azar, 2007), (Aaqib I, 2009).

1.2.1.3 Risk

Risk calculation is a crucial part of requirement prioritization process. Prioritizing the requirements according to the risk related with there Implementation may seems to be a logical choice. For instance, it's possible to attempt to implement high risk requirements first, in order to cope with the resulting challenges throughout development process. Alternatively, it may seems logical to implement requirements with minimum risk first, to be able to increase the amount of system implemented by making certain that minimal resources aren't wasted in attempting to implement the high-risk aspects of the system which may not be possible to successfully implement. Delaying the implementation of high-risk requirements may also maximize the period available to investigate the risks and to establish suitable risk mitigation strategies (J.Azar, 2007, Aaqib I, 2009).

1.2.1.4 Value

One of the main considerations in requirement prioritization process is determining the value of each requirement. Nearly every prioritization technique compares value of individual requirements with other requirements or criteria's. Value indicates the significance of each requirement in the software product. All requirements are generally not equal when it comes to importance. Some are important as compare to others. It is critical to determine the significance and value of every single requirement in order to implement the most significant and important requirement first. (Aaqib I, 2009).

1.2.1.5 Dependency Constraints

In incremental product development approach, dependency constraints occur between requirements. Dependency means that one requirement or set of requirements is dependent on another requirement or set of requirements. Dependency Constraints are of two types Precedence and Coupling Constraints (Aaqib I, 2009).

Precedence Constraints: "*Precedence constraints occur when one requirement can only be started if another is completely finished. Consider we have two requirements R1 and R2. Requirement R1 can only start if requirement R2 is completely finished. There is a precedence constraint between requirements R1 and R2*" (Aaqib I, 2009).

Coupling Constraints: "*Such requirements that must be implemented in the same increment. Consider we have four requirements R1, R2, R3, and R4. Requirements R1 and R2 must be in the same increment. So there is a coupling constraint between requirement R1 and R2*" (Aaqib I, 2009).

1.2.1.6 Effort

Engineers have to deal with effort estimation issue whilst prioritizing requirements. Effort estimation for certain release or releases involves estimation of overall available effort and total effort required. *"Estimate effort led by the technical leader and architecture team, the development team that must actually implement the requirements creates and records realistic estimates of the effort required to implement each requirement"* (J.Azar, 2007).

1.2.1.7 Resources

86% of companies sited resource constraints as top 3 strategic. This was steady around companies despite of the quantity of employees, income or perhaps variety of products for sale (Lubars 1993). Resources refer to the budget, staff and schedule.

1.2.2 Business Factors

1.2.2.1 Business Values

"A business value is a belief, a mission, or a philosophy that is really meaningful to the organization. Every organization has one or more values, whether they are consciously aware of it or not. Another way of saying it is: a value is a statement of the company's intention and commitment to achieve a high level of performance on a specific qualitative factor" (Aaqib I, 2009).

Few business values are listed below (Yourdon E, 1997), (J.Azar, 2006), (Aaqib I, 2009):

- Continuous Improvement
- Creativity
- Customer Delight
- Develop People
- Resourcefulness

- Trust
- (A Will to) Succeed
- Being a market leader in terms of sales
- Being of being the first to market with a product
- Customer retention

Prioritization process is used mostly in the starting phases of development process. Effective use of requirement prioritization prevents project failure by addressing high-priority requirements first and then move towards the low-priority requirements. Prioritization helps build set of core requirements around which the project is built. Ed Yourdon believes that prioritization of requirements is an extremely important issue (Yourdon E, 1997), Lubars et al. stated that prioritization of requirements was a topic that came up repeatedly among the market-driven projects they surveyed (Lubars 1993), and Siddiqi et al. identified prioritization as a key but neglected issue in requirements engineering research (Siddiqi, J. 1996).

> *"Customers are never thrilled to find out they can't get all the features they want in release 1.0 of a new software product (at least, not if they want the features to work). However, if the development team cannot deliver every requirement by the scheduled initial delivery date, the project stakeholders must agree on which subset to implement first. Any project with resource limitations has to establish the relative priorities of the requested features, use cases, or functional requirements. Prioritization helps the project manager resolve conflicts, plan for staged deliveries, and make the necessary trade-off decisions. (Wiegers 1992)"*

In order to gain the advantage in the market the process of software requirements prioritization is also considered as vital and it helps in understanding the ups and downs of the market in terms of loss and profit (Aurum, 2003). In software industry the consideration of all requirements, in order to develop a software system, is not possible because of the constraints like time to market, budget, and other resources so instead of considering all the user requirements in a single release the consideration of important requirements is taken into account (Karlsson, 1997, Siddiqi, J. 1996).

Different techniques for requirement prioritization address different factors. AHP uses pair-wise evaluation (Saaty, Thomas L 1994), Zultner proposed a requirement prioritization technique of multiple stakeholders or customers (Zultner, R.E. 1997) using AHP for determining the priorities of multiple customers, Karlson also used the AHP concept and developed a cost-value approach for prioritizing requirements (Karlsson, 1997), SERUM (Greer D, 1999) uses estimation for cost, benefit, development risk and operational risk to inform the prioritization process, VOP (J.Azar, 2007) uses a framework for prioritizing and making decisions about requirement and the list goes on.

There are many advantages, disadvantages and limitations associated with these techniques. These methods consider different factors while prioritization, so it is very important to select prioritization method according to given constraints. It is essential to decide what is important before these requirements are incorporated into the development process. And the term important may have different meaning for different people. E.g. sometimes project completion on time is more important than cost. Importance is highly subjective thing that varies considerably from one stakeholder to another (Lehtola, 2004), (Lehtola, Kauppinen, 2004). In RE, there are multiple proposals for defining what the term 'importance' means. Two key factors are benefit and/or cost associated with each individual requirement (Erdogmus, 2006), (Karlsson, 1997), (Lehtola, Kauppinen, 2004), (Wiegers 1992), (Herrmann, 2008). Prioritizing the needs of stakeholders is best way to know what is important for stakeholders. Besides stakeholder viewpoint Requirements prioritization should also consider business and implementation issues like financial benefits for the developing organization, competitors, regulations whereas implementation issues mostly involve implementation cost, cost if not implemented, available resources etc. (Lehtola et al. 2004).

In order to develop a successful innovative software system there is the need to select right and relevant user requirements. For right selection of user or stakeholders' requirements there exist different prioritization approaches which are discussed in detail in the next chapter.

Chapter # 2

LITERATURE SURVEY

2. Literature Review

Literature shows the importance of requirement prioritization. It is essential to decide what is important before these requirements are incorporated into the software development process. By addressing the high-priority requirements before considering the low-priority ones, one can significantly reduce both the costs and duration of a project (Hofmann H.F., Lehner F., 2001). In 2004 the Standish Group (Johnson J, 2006) surveyed 50,000 completed commercial and governmental software projects. Of all projects surveyed only 29% were found to finish close to or on time/budget, while 18% failed to produce a usable product. The remaining 53% were classified as challenged, meaning they finished late and/or over-budget. Reasons for these overruns include unrealistic goals, inaccurate estimates, an ill-defined system, poor monitoring of project status and poor project management (Charette, R). Through the implementation of an honest and reliable release plan the chance of completing a project within the allocated time and budget can increase considerably (Cohn M, 2006).

According to annual report named 'CHAOS Summary 2009' prepared by Standish Group, ten main factors causing challenged or failed projects are unveiled. Four of them are: (Mohammad S. H., Abdullah M)

- Lack of user involvement
- Lack of resources
- Unrealistic expectations
- Changing requirements and specifications

Requirements prioritization has become an essential step in the software development process in order to reduce software failure. (Hatton S., 2007)

Requirements prioritization has been recognized as one of the most important decision making processes in the software development process. (Ngo-The A., Ruhe, G., 2005)

In order to select right set of requirements researchers has developed many requirements prioritization techniques AHP, SERUM, VOP, EVOLVE, Cumulative Voting or Hundred Dollar Test, Binary Search Tree, Numerical Assignment, Ranking, Binary Search Tree etc. Different methods or ways are used to prioritize the requirements e.g. Wiegers presented a technique in which quantitative ranking from 1-9 was used and for ranking the factors like cost, risk and importance were taken into account (Wiegers K, 1999). Some techniques requires stakeholders to prioritize requirements on the basis of factors like which requirement is mandatory, desirable or essential one and which one is not (J.W. Brackett, 1990), while some have to adopt quantitative ranking system for requirements. Kent Beck presented the planning game method to prioritize the requirements in extreme programming (Kent B, 2000).

A research conducted by Karlsson (Karlsson L, Regnell B, Wohlln C, 1998) in which prioritization methods are applied to prioritize 13 well-defined quality requirements on a small telephony system. The results provided by most of the techniques, except AHP and Bubblesort, are unreliable and prone to errors while on the other hand AHP and Bubblesort are more reliable but time consuming due to the decisions which one must have to consider during requirements prioritization methods. AHP is reliable but its time consuming due to pair wise comparison and its is complex to implement. The spanning tree technique is more prone to faults so the results are not reliable. Hierarchy AHP is not reliable and results provided by Hirerarchy AHP are error prone and faulty. The prioritization techniques of Binary search and Priority Groups are also prone to errors and the results are not reliable. The role of consistency index is very important because it helps to reduce the human error so the results provided by Bubblesort may be faulty and unreliable. AHP and Bubblesort is problematic to scale up (Karlsson L, Regnell B, Wohlin C, 1998). AHP is not suitable when the requirements exceed the limit of 20 because the number of comparison grows so it becomes difficult to manage the requirements (Lehtola L, Kauppinen M, 2004).

Peng Shao proposed an algorithm for requirements prioritization and this algorithm or "technique allows customers to rank a relatively small set of items then combines these rankings over a large number of customers to determine an ordering for a large requirement set" (Peng S, 2008). The proposed algorithm is just simulated and not implemented in the real environment. The simulation is performed on small number (10) of customers with 10 items. Initially the technique is focusing for small range of requirements and not for large and there is no statistical evidence of the results. About implementation of the technique writer says "In the future, we will try to implement out algorithm to improve requirements prioritization" (Peng S, 2008).

Laurent et al. presented a technique called "Requirements Triage". According to Laurent et al. "The limitations of the approach are closely related to the limitations of the underlying traceability, classification, and clustering algorithms, all of which are based upon data mining and information retrieval techniques that are probabilistic in nature and that therefore do not return perfect precision or recall in the results" (Laurent P, Cleland-Huang J, Duan C, 2007). The requirements prioritization model for market driven products (M. A. Iqbal, A. M. Zaidi, 2010) is an extension of AHP and is proposed to solve the issue of scalability. The model is implemented on companies A and B but there is no data evidence or statistics for the experimentation which is performed and most of the results are just opinions. The technique or model is actually focusing the issue of scalability (large scale requirements) but the proposed model is not for each and every organization and also not telling the focused domain (M. A. Iqbal, A. M. Zaidi, 2010). Further there is no evidence that at which extent the proposed model is solving the issue of scalability. In another research a prioritization technique is presented which is for legal requirements (Aaron K Massey, 2010). The very technique is focusing the legal implications of requirements in the domain of healthcare. The major focus of this technique is HIPPA (Health Insurance Portability and Accountability Act) which regulates the Electronic Health Records (EHRs). So the technique is focusing the domain of Healthcare systems and it is not suitable for all projects.

SERUM (Software Engineering Risk: Understanding and Management) (Greer D, 1999) uses estimations for cost, benefit, development risk, and operational risk reduction to inform the prioritization process. SERUM focuses on some of the crucial factors while prioritization but still neglects important factors like value, dependency constraints, effort etc. Same is the case with VOP (J.Azar, 2007) and EVOLVE (Greer D, 2004). EVOLVE consider value, benefits, risk, dependency constraints, effort and resources but neglects some important factors like business value and also it's not cost effective to Implement (). VOP is also an effective approach by considering important factors like cost, value, risk but neglects benefits of implementing certain requirement, dependency constraints, effort, resources etc (Aaqib I, 2009).

According to the literature review presented above problem with the existing prioritization techniques are:

- Existing techniques neglects many important factors/criterions (cost, benefits, value, risk, time etc) while prioritization process.
- The existing techniques don't provide a scalable solution when the requirements scaled up in case of large number of requirements.
- Most of the techniques are time consuming.
- The results are faulty or error prone.
- Most techniques are only suitable for only small set of requirements.

We have conducted deep literature review (Aaqib I, 2009) and picked up commonly used prioritization techniques AHP, SERUM, VOP, EVOLVE. And the factors considered by these techniques cost, value, benefit, risk, dependency constraints, business values, effort, resources, approach (relative, absolute). We evaluated these techniques and the result of our evaluation shows the limitations of these techniques. By keeping these limitations in mind we will propose a new prioritization approach trying to eliminate these limitations.

Chapter # 3

PROBLEM STATEMENT

3. Problem statement

Karlsson and Ryan quotes from Software Requirements, Objects, Functions and States by Davis A. M. (1993) and say in their paper that:

"Unfortunately, many software systems end up being developed without meeting their requirements, and thus cannot be used as initially expected (Davis A. M. 1993). In particular, when there are limited resources but an almost infinite range of candidate requirements, it is essential to choose those requirements that give the best return, in terms of customer satisfaction, on the investment involved" (Karlsson, Ryan, 1996).

According to Laurent et al:

"Lack of an effective prioritization and triage process can lead to problems such as missed deadlines, disorganized development efforts, and late discovery of architecturally significant requirements. Existing prioritization techniques do not provide sufficient automation for large projects with hundreds of stakeholders and thousands of potentially conflicting requests and requirements" (Laurent et al., 2007).

Literature review shows that there are many limitations associated with existing requirements prioritization techniques. There is an issue that because of "infinite range of candidate requirements" (Karlsson, Ryan, 1996) "many software systems end up being developed without meeting their requirements and thus cannot be used as initially expected" (Davis A. M. 1993). In order to provide solution to this problem the software requirements must be prioritized in an innovative way instead of the traditional one. There is the need of a highly efficient and cost effective software requirements prioritization method which will be suitable for projects with "infinite range of candidate requirements" (Karlsson, Ryan, 1996). Most techniques are suitable for small set of requirements, only few techniques considers some of many crucial factors while prioritization, existing techniques are complex and error prone. There are two sides of requirements, technical side that shows the technical importance of requirement which can be measured in terms of factors like cost, value, benefit, risk etc and then there is business side of requirement that shows the business value of requirement which can be measured in terms of factors like customer satisfaction, sales, marketing, strategic, integrity etc. For effective and optimal requirement prioritization engineer should consider both technical and business perspective of requirements. Most techniques only cover the technical factors of requirement and neglect the business factor and some techniques like VOP considers both business and technical factors but still neglects many important technical factors as shown in table below. We proposed a technique which focus on both business factors and important technical factors identified in the literature while requirement prioritization process. Proposed solution has been discussed in the following section thoroughly.

Table 1 shows the important factors that are present in almost every software project (Aaqib I, 2009). Figure also shows the gap in the existing techniques, not even a single technique considers all the important factors while prioritization. We will propose a prioritization approach trying to eliminate/minimize the gaps/limitation associated with existing prioritization techniques.

		AHP	SERUM	EVOLVE	VOP
Cost		✓	✓		✓
Value		✓		✓	✓
Benefit			✓	✓	
Risk			✓	✓	✓
Dependency Constraints	Precedence			✓	
	Coupling			✓	
Business Values					✓
Effort				✓	
Resources				✓	
Approach	Relative Assessment	✓			
	Absolute Assessment		✓	✓	✓

Table 1: Techniques/Factors Comparison (Aaqib I, 2009).

Literature review presented in chapter 2 is the motivation for this research. There are many limitations and gaps associated with existing prioritization techniques (Aaqib I, 2009). In this research we'll try to minimize or remove these limitations.

3.1 Research Question

- RQ#1
  - What techniques are reported in literature for requirements prioritization?
- RQ#2
  - What are the limitations in requirements prioritization techniques?
- RQ#3
  - How these limitations cost, effort, resources, risk, value, constraints can be eliminated and incorporated in a single technique?
- RQ#4

- To What extent the new proposed technique prove to be effective in eliminating or minimizing the limitation in the existing prioritization techniques.

3.2 Proposed Solution

We've proposed a new approach for requirement prioritization (Figure 1) showing the tasks that should be performed for effective prioritization (Aaqib I, 2009).

Current prioritization techniques neglect many important factors that must be taken into account during prioritization process (Yourdon E, 1997). In literature review section we've identified some of the important factors that are present in almost every software project. The process we proposed only shows how the important factors can be incorporated in requirement prioritization process. We've modified the process into implementable technique "Hybrid technique" by dividing it into different phases. In each phase we focused on different factors. At the end we'll perform experiments implementing hybrid technique and will compare the results of the experiments with expert judgments to prove the effectives of proposed technique.

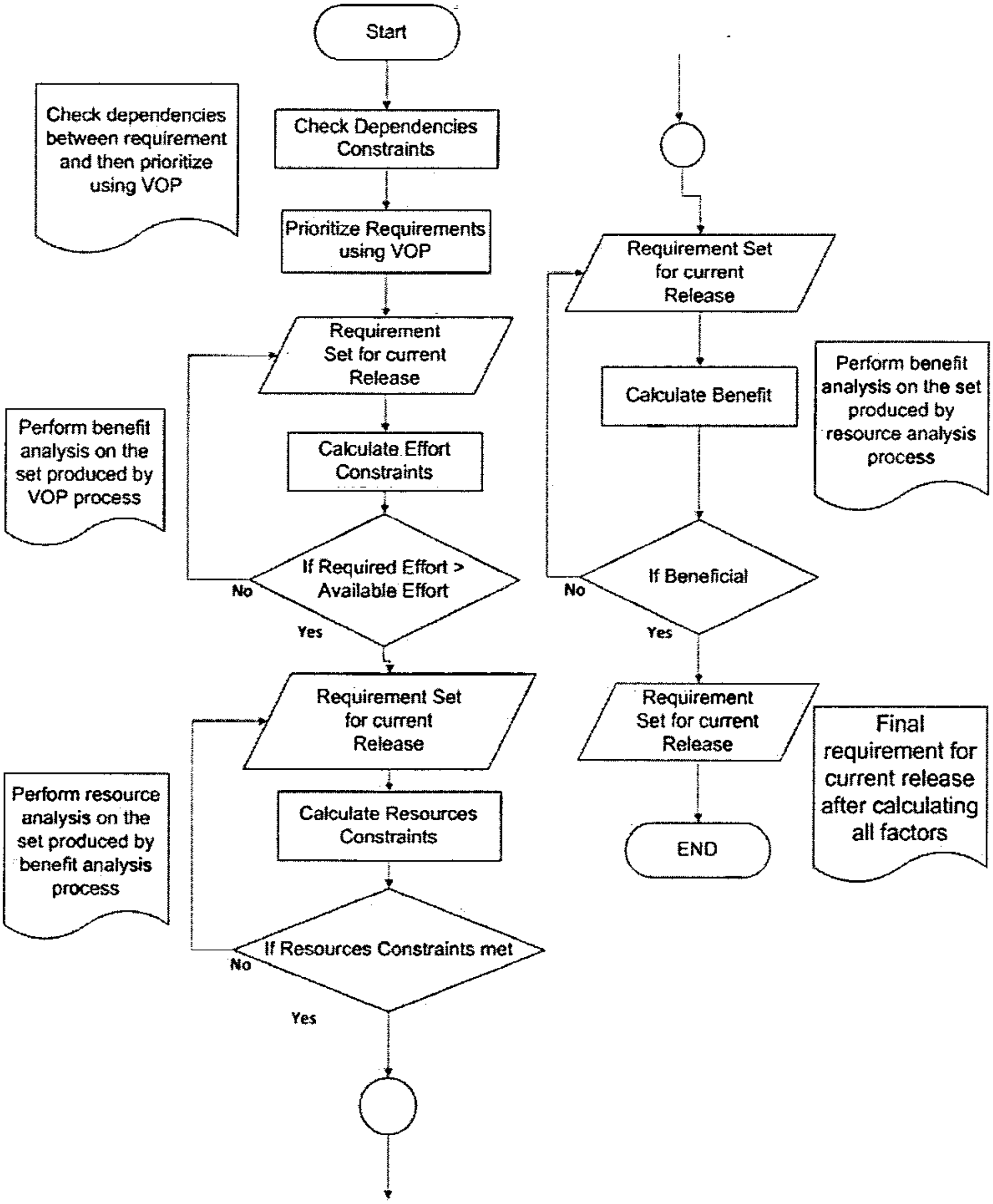

Start
Check dependencies between requirement and then prioritize using VOP
Check Dependencies Constraints
Prioritize Requirements using VOP
Requirement Set for current Release
Perform benefit analysis on the set produced by VOP process
Calculate Effort Constraints
If Required Effort > Available Effort
No
Yes
Requirement Set for current Release
Perform resource analysis on the set produced by benefit analysis process
Calculate Resources Constraints
If Resources Constraints met
No
Yes
Requirement Set for current Release
Calculate Benefit
Perform benefit analysis on the set produced by resource analysis process
If Beneficial
No
Yes
Requirement Set for current Release
Final requirement for current release after calculating all factors
END

Figure 1: Proposed approach for requirements prioritization (Aaqib I, 2009)

Figure 2 below shows the flow of hybrid technique. Proposed solution is discussed in detail in the following sections.

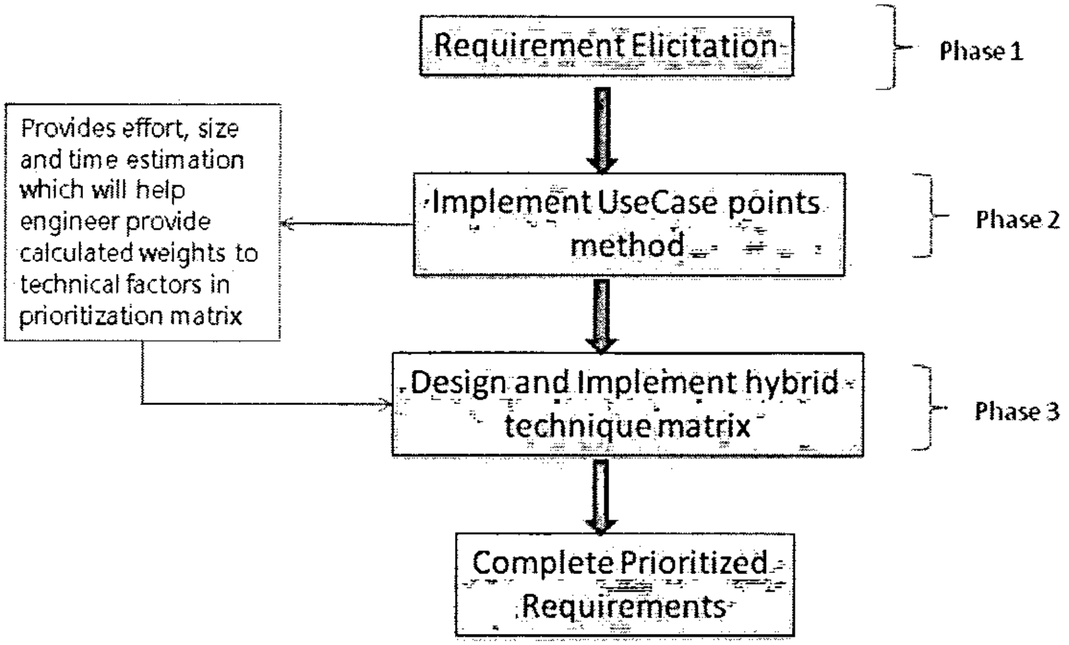

Requirement Elicitation
Phase 1
Provides effort, size and time estimation which will help engineer provide calculated weights to technical factors in prioritization matrix
Implement UseCase points method
Phase 2
Design and Implement hybrid technique matrix
Phase 3
Complete Prioritized Requirements

Figure 2: Flow of hybrid technique

3.3 Intended Output

- This should result in better understanding of limitations present in existing prioritization technique and how these gaps/limitations can be eliminate/minimized.
- This should provide engineers better understanding of technical and business perspective of requirements and how these perspectives can be incorporated in prioritization process.
- The proposed technique should provide more optimal and effective results after focusing both business and technical perspective in requirement prioritization process.

Chapter # 4

RESEARCH METHODOLOGY

4. Research Methodology

Experiment is chosen as method for study design for carrying out research. It is an empirical method for conducting research in software engineering. A controlled experiment is an investigation of a testable hypothesis where one or more independent variables are manipulated to measure their effect on one or more dependent variables. Controlled experiments allow us to determine in precise terms how the variables are related and, specifically, whether a cause-effect relationship exists between them. Each combination of values of the independent variables is a treatment (Easterbrook S, 2008).

"Controlled experiments offer several specific benefits. They allow us to conduct well defined, focused studies, with the potential for statistically significant results. They allow us to focus on specific variables, measures, and the relationships between them. They help us formulate hypotheses by forcing us to clearly state the question being studied and allow us to maximize the number of questions being asked. Such studies usually result in well defined dependent and independent variables and well-defined hypotheses. They result in the identification of key variables and good proxies for those variables. They allow us to measure the relationships among variables." (Victor R, 2006)

A checklist is constructed according to the guidelines provided in (Jedlitschka A, 2005):

4.1 Project Title

A Hybrid Technique for Requirements Prioritization - A controlled experiment.

4.2 Structured Abstract

Purpose of this thesis is to provide better understanding of the limitation in the current prioritization techniques. Based on these limitations we have proposed a new prioritization technique "Hybrid technique for requirement prioritization". We've tried to eliminate/minimize the limitation in the existing prioritization techniques. To prove the effectiveness of the proposed technique we'll perform two an experiment to implement the proposed technique. We'll also get an expert estimation on the same software projects upon which we implemented the Hybrid technique. At the end we'll perform an analysis based on our finding from experiment and expert estimation in the results and discussion chapter.

4.3 Hypothesis

This should result in better understanding of the limitation in the current prioritization techniques. Proposed software prioritization technique will eliminate/minimize the limitation in the existing prioritization techniques.

4.4 Experiment Design

We'll get the project description from stakeholder and formalize the raw description into usecases. At phase 1 we'll use UseCase point method to calculate size, effort and time required to complete the project. In phase 2 we'll use hybrid technique matrix to prioritize requirement. At the end we'll get the same matrix filled out by industrial experts to match the results obtained by hybrid technique.

4.5 Execution

Experiment execution consists of following phases:

- calculate size of the project, effort and time required for the selected set of requirements using Use Case point method
- Use hybrid prioritization matrix to prioritize requirements set.
- Result and analysis

4.6 Study Analysis and Validation

Table 2 shows the matrix that will be used for data collection. Data will be populated after implementing the Hybrid Technique. Figure shows the value distribution across each factor for the selected set of requirements. All requirements will be given value from scale 1-10, 10 being the highest and 1 being the lowest.

	Business Perspective					**Technical Perspective**				Score
	Business Values (V_1 .... V_n)					Benefit	Value	Risks (R_1 ... R_m)		
Rqmt	Sales	Marketing	Competitive	Strategic	Customer Retention			Tehnical	Business	
	V_1=7	V_2=6	V_3=8	V_4=10	V_n=7	TF_1=7	TF_k=8	R_1=-8	R_{m2}=-5	
r1										
r2										
...										
rN										

Table 2: Hybrid Technique matrix

Hybrid technique will provide us the value for each requirement against each factor. Cumulative value will provide us combined score of each requirement. Requirement with highest cumulative score will be the top priority requirement and requirement with the lowest cumulative score will be the lowest priority requirement. Results validation will be qualitative in nature and requirement engineer from the industry will be involved in validation. After getting the values in matrix by implementing Hybrid technique the same matrix and requirements set will be used to get value from the requirement engineer (Industry experts). This will show provide us with the expert priority for each requirement against each factor. The results can then be validated with the data obtained from the expert to show the effectiveness of Hybrid technique.

4.7 Interpretation

In this section we'll interpret the findings from the analysis presented in the previous section. This includes an overview of the results, threats to validity, generalization (where are the results applicable?), as well as the (potential) impact on cost, time, and quality. We suggest structuring the Interpretation section into the following subsections: Evaluation of Results and Implications, Limitations of the Study, Inferences, and Lessons Learned. This section shall be covered in results and analysis chapter.

4.8 Significance of Study

This study will help engineer's better understanding of limitations present in existing prioritization technique and how these gaps/limitations can be eliminate/minimized using the proposed technique. Proposed technique will help engineers focus on both business and technical perspective during prioritization process.

Chapter # 5

PROPOSED SOLUTION

5. Proposed Solution

In this chapter we'll discuss proposed solution in detail and then we will perform an experiment to prove the effectiveness of proposed technique. As discuss in previous sections for proper and effective prioritization it's very important to consider both technical (Henry, 1993)/criterion (Barragans, 2005) and business (J.Azar, 2007) factors associated with requirement set to be prioritize. There are many requirement prioritization techniques that focus on technical perspective of requirement and some focus on business perspective but as shown in literature review for effective and optimal prioritization engineer should focus on both technical and business perspective of requirement. Therefore we've proposed a technique that will consider both technical and business perspective while performing prioritization. It's very hard to incorporate both these perspectives in prioritization process because existing prioritization techniques are time consuming and error prone and if we include more factors for prioritization the process will become more complex. We need a technique which is simpler and at the same time focus on both perspectives of requirements, therefore we've selected VOP technique. Case study performed in (J.Azar, 2007) shows that VOP provides effective results but it still lacks some major areas. We proposed a technique "Hybrid technique for requirement prioritization" which focuses on both business factors and important technical factors identified in the literature during requirement prioritization process. We've modified VOP process and the matrix used in VOP technique. VOP calculates business values and two technical factors risk and cost. VOP still neglects important technical factors individual value of each requirement, benefit, resources and effort. As hybrid technique is based on VOP it's important to provide brief description of VOP technique.

"VOP uses a framework that gives requirement engineers a foundation for prioritizing and making decision about requirements. It provides visibility for all stakeholders during decision making, eliminating lengthy discussions and arguments over individual requirements by emphasizing the core business values. VOP is a quantitative method in which requirements are prioritize based on numerical values assigned by engineer and stakeholders to each requirement against core business values.

The first step in setting up a value oriented prioritization process is to establish a framework for identifying the business's core values and the relative relationships among those values. VOP uses the relationships that exist between core business values to assess and prioritize requirements and ensure their traceability. The VOP framework establishes a mechanism for quantifying and ordering requirements for an application increment, a prototype, or a software requirements specification. Company executives identify the core business values and use a simple ordinal scale to weight them according to their importance to the organization. The framework also supports the identification and weighting of business risk categories. Weighted risk categories represent the organization's tolerance for engaging in those risks. For example, an organization might be more tolerant with respect to taking a business risk and less tolerant with respect to taking a technical risk. In VOP, risk is measured on a negative scale.

Using the core business values and risks, VOP constructs a prioritization matrix. Table 3 illustrates such a matrix, incorporating five business values and two types of risks. In this matrix, V_i is the weight of business value i and R_j is the weight of risk j. VOP then weights each requirement relative to each specific business value and risk; the weighting scale is 0 (not *important) to 10 (critical). $W_{i,j}$ is the weight assigned to requirement ri with respect to business value V_j. Likewise, $W_{i,j}$ is the weight assigned to the requirement r_i with respect to R_j. Formally, we can express the score, Sr, for each requirement r, in the set R of all possible requirements, as the sum of its contribution to core business values minus the sum of its perceived risks. The total core business value contribution is simply the product of each value's weight times the weighting of that requirement with respect to that perceived risks is the product of each risk's weight times the requirement's weighting with respect to that risk. Table B shows an example of applying business value and risk weights within the VOP framework. In the example, requirement r2 has a higher overall score than requirement r1, so it has a higher priority and is a more logical choice for inclusion in the project's core requirements set"* (J.Azar, 2007).

	Business Values (V_1 … V_n)					Risks (R_1 … R_m)		
Rqmt	Sales	Marketing	Competitive	Strategic	Customer Retention	Technical	Business	*Score*
–	V_1=7	V_2=6	V_i= 8	V_{i+1}=10	V_n=7	R_1 = -8	R_m=-5	
r_1								
r_2			$W_{i,j}$			$W'_{i,j}$		
…								
r_N								

Table 3: VOP matrix

5.1 Hybrid technique for requirement prioritization

VOP is one of the best approaches for requirement prioritization. It evaluates requirements according to their impact on core business values. It also supports identification and weighing of business risks and cost of implementing each requirement. VOP builds a prioritization matrix using the core business values, requirements values and risk. This matrix shows the priority of each requirement and the requirements that has higher priority is a more logical choice for inclusion in the requirement set for the current release. The output of VOP process is the requirement set that should be implemented in the current release, but there are still few major factors that are not judged in this process: the effort and resources required for each requirement, the benefit of implementing the requirement set and individual value of each requirement. We have modified matrix used in VOP to calculate benefit and value associated with individual requirement and for effort and size we'll use usecase points method. *"Use Case Points is a project estimation method that employs a project's use cases to produce an accurate estimate of a project's size and effort"*. The reason of using use case point method is because we are mainly focusing on technical factors and use case point method analyzes technical and environmental method for calculating size and effort. This will help engineer to better understand and assign correct weight to factors like risk, effort and resources in prioritization matrix.

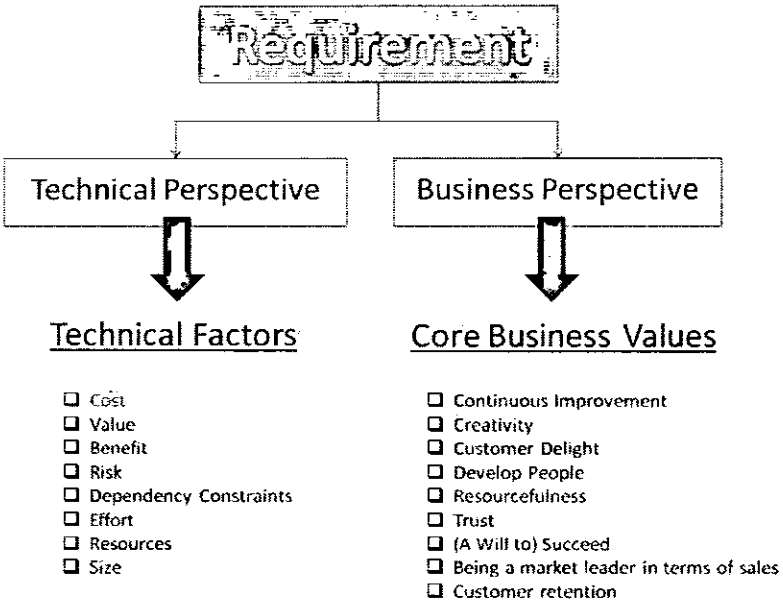


Figure 3: Requirement perspectives for hybrid technique

Figure 3 shows both the perspective that hybrid technique will focus during prioritization process.

Technical factors listed under column 1 has been identified from the literature that is present in almost every software project and business values listed under column 2 will be selected by clients/stakeholders based on their business.

Hybrid technique consists of following steps:

5.1.1 Step 1: Use Case Point Method

In step one we'll use UseCase point method to calculate effort and size of the project. To calculate UCP following equation is used:

UCP = UUCP * TCF *ECF * PF

UUCP (Unadjusted Use Case Points)

TCF (Technical Complexity Factor)

ECF (Environment Complexity Factor)

PF (Productivity Factor)

UCP is the Size of the project. To calculate Time and effort:

UCP / 40 hours *"for one man work week"*.

Through Use Case point we calculated total size of project and by providing the project size we calculated time and effort required to complete the project. This also help engineer to

5.1.2 Step 2: Hybrid Technique Prioritization Matrix

In step 2 we'll use prioritization matrix that hybrid technique use to prioritize requirement set focusing on both business and technical perspective.

	Business Perspective					Technical Perspective				Score
	Business Values (V_1 .... V_n)					Benefit	Value	Risks (R_1 ... R_m)		
Rqmt	Sales	Marketing	Competitive	Strategic	Customer Retention			Tehnical	Business	
	V_1=7	V_2=6	V_3=8	V_4=10	V_n=7	TF_1=7	TF_k=8	R_1=-8	R_{m2}=-5	
r1										
r2										
...										
rN										

Table 4: Finalized hybrid technique data collection matrix

In this matrix (Table 4), V_i is the weight of business value *i*, TF_k is the weight of technical factors *k* and R_j is the weight of risk *j*. Each requirement then weight relative to there business values, technical factors and risk; weighing scale is 0 (not important) to 10 (critical). $W_{i,j}$ is the weight assigned to requirement r_i with respect to business value V_j. Likewise, $W'_{i,j}$ is the weight assigned to the requirement r_i with respect to risks *Rj*, and $W''_{i,j}$ is the weight assigned to requirement r_i with respect to technical factor TF_k. Finally we can express the score *Sr*, for each requirement r, in the set R of all possible requirements, as the sum of its contribution to core business values plus sum of technical factors values minus the sum of its perceived risks. The total core business value contribution is simply the product of each value's weight (for both business values and technical factors) times the weighting of that requirement with respect to that perceived risks is the product of each risk's weight times the requirement's weighting with respect to that risk.

$$\mathbf{Sr = W_{i,j} + W'_{i,j} + W''_{i,j}}$$

5.2 Experiments

We've implemented hybrid technique on two sample projects online book shop and online mentoring system.

5.3 Experiment #1: Online book shop

Following are the requirements for online book shop project as explained by client.

5.3.1 Project Description

New customers need to register first to get one account ID. After registration, the customer will be assigned one account ID and he/she can login using account ID and password. One customer can only register one account and each account must belong to exact one customer.

The bookstore keeps a large amount of books. Each book is identified by ISBN. For each book, the bookstore also needs to record its authors' names, title, edition, year, category, publisher, quantity-in-stock, and price.

One customer can place any number of orders. For each order, the bookstore needs to record who places this order, when, the order status, total price, shipping address, payment method, bill address, and ordered books. Note there is only one shipping address and one billing address for each order though the shipping address may not be the same as the billing address. Currently for payment method, it only accepts credit card, hence the bookstore needs to record credit card information.

Customers can also manage their shopping carts. One customer can have any number of shopping carts. However, each shopping cart has exactly one customer. The shopping cart contains the following info: cart-ID, name, date-created, date-last-updated, books contained in this shopping cart, etc.

Now we'll implement hybrid technique to prioritize the requirements mentioned by client.

5.3.2 Step 1: Use Case Point Method

In step 1 we'll calculate size of the project, effort and time required for the selected set of requirements using Use Case point method. This method consists of following phases:

Phase 1: Classification of use cases

Phase 2: Determine and calculate the TCF.

Phase 3: Determine and calculate the ECF.

Phase 4: Determine UUCP

Phase 5: Calculate UCP

We are not showing full procedure for each phase as it will take lot of time. We are just showing the values and the formulas we used to calculate these values during each phase.

Phase 1: Classification of use cases

UC01: Register with Book Shop
UC02: Login
UC03: Request UnListed Book
UC04: Locate Book by Title or Author
UC05: Browse Book Catalogue

UC06: Pay for Order

UC07: View Cart

UC08: Remove Item from Cart

UC09: Add title to Cart

UC10: Cancel Order

UC11: Enquire on Order Status

Phase 2: Determine and calculate the TCF.

Metric	Description	Weight	Perceived Complexity	Calculated Factor (weight*perceived complexity)
TCF01	Distributed System	2	5	10
TCF02	Response or throughput performance objectives	1	4	4
TCF03	End user efficiency (online)	1	2	2
TCF04	Complex internal processing	1	4	4
TCF05	Code must be re-usable	1	2	2
TCF06	Easy to install	0.5	5	2.5
TCF07	Easy to use	0.5	3	1.5
TCF08	Portable	2	3	6
TCF09	Easy to change	1	3	3
TCF010	Concurrent	1	2	2
TCF011	Includ special security features	1	2	2
TCF012	Provide direct access for third parties	1	5	5
TCF013	Special user training faciities are required	1	3	3
			Total:	47

Table 5: Compute technical factors

TCF = 0.6 + (.01*TF). For table 5, the TCF = 1.07

Phase 3: Determine and calculate the ECF.

Metric	Description	Weight	Perceived Impact	TCF = (weight*perceived complexity)
ECF01	Familiar with Rational Unified Process	1.5	4	6
ECF02	Application experience	0.5	3	1.5
ECF03	Object-oriented experience	1	4	4
ECF04	Lead analyst capability	0.5	4	2
ECF05	Motivation	1	3	3
ECF06	Stable requirements	2	4	8
ECF07	Part-time workers	-1	0	0
ECF08	Difficult programming language	-1	3	-3
			Total:	21.5

Table 6: Compute environmental factors

ECF = 1.4 + (-0.03*TF). For table 6, the ECF = 0.75

Phase 4: Determine UUCP

Package	Name	Type	Complexity
UC01: User Management	Register with Book Shop	UseCase	5
UC02: User Management	Login	UseCase	5
UC03: Select Books	Request UnListed Book	UseCase	5
UC04: Select Books	Locate Book by Title or Author	UseCase	5
UC05: Select Books	Browse Book Catalogue	UseCase	10
UC06: Manage Order	Pay for Order	UseCase	5
UC07: Manage Order	View Cart	UseCase	5
UC08: Manage Order	Remove Item from Cart	UseCase	5
UC09: Manage Order	Add title to Cart	UseCase	5
UC10: Order Status	Cancel Order	UseCase	5
UC11: Order Status	Enquire on Order Status	UseCase	5
		Total	60

Table 7: Unadjusted use case points

For table 7 UUCP=60.

Phase 5: Calculate UCP

The UCP is calculated by multiplying the obtained variables in previous phases:

UCP = TCP * ECF * UUCP * PF

"The Productivity Factor (PF) is a ratio of the number of man hours per use case point based on past projects. If no historical data has been collected, a figure between 15 and 30 is suggested by industry experts. A typical value is 20".

For the values used in this experiment:

UCP = 1.07 * 0.75 * 60 * 20 = 963 hours.

Divide the UCP by 40 hours (for one man work week) = 24 man-weeks. Hence, for the values used in this experiment, it would take one developer 24 weeks to complete the application.

5.3.3 Step 2: Hybrid Technique Prioritization Matrix

In step 2 we'll use hybrid prioritization matrix to prioritize requirements set focusing on both business and technical perspective.

Rqmt	Business Perspective						Technical Perspective				Score
	Business Values (V_1 .... V_n)						Benefit	Value	Risks (R_1 ... R_m)		
	Sales	Marketing	Customer Satisfaction	Continuous Improvement	Time to market	Customer Retention			Tehnical	Business	
	V_1=7	V_2=6	V_3=6	V_4=9	V_5=4	V_6=10	TF_1=7	TF_2=8	R_1=-8	R_2=-5	
Register with Book Shop	5	4	6	7	7	5	6	6	7	7	235
Login	4	3	7	6	5	6	4	5	6	7	207
Request UnListed Book (2)	6	7	6	8	9	8	7	8	5	5	356
Browse Book Catalogue (1)	7	8	5	9	10	7	9	8	5	4	385
Pay for Order	3	2	6	8	7	5	5	4	8	8	188
View Cart	7	5	9	8	6	6	7	5	7	6	281
Remove Item from Cart	3	2	5	6	6	4	3	5	8	7	143
Add title to Cart (3)	5	6	8	7	7	5	6	7	6	5	285
Cancel Order	2	3	4	3	5	2	4	3	8	9	56
Enquire on Order Status	5	4	6	7	6	5	6	7	10	9	205

Table 8: Experiment #1: ranking of potential top 3 requirements after implementing hybrid prioritization matrix

Under business perspective we have organization core business values; every organization can identify their own business values. Company executives identify the core business values and use a simple ordinal scale to weight them according to their importance to the organization. For technical perspective we have Risk (technical and business), benefit and individual value of each requirement. In hybrid technique risk is measured on a negative scale. Weighted risk categories represent the organization's tolerance for engaging in those risks. For example, an organization might be more tolerant with respect to taking a business risk and less tolerant with respect to taking a technical risk. Stakeholders assigned weight to each requirement relative to there importance for each business value. Requirement engineer assigned weight to each requirement according to their impact for each technical factor.

According to the scores in table 8, the top three requirements are Browse Book Catalogue (385), Request UnListed Book (356), and Add title to Cart (285). The Result shows the benefit of considering technical perspective along with core business values, if the process only considered the weighted business values, then requirement "View Cart" would have at number three; the risks associated with this requirement lowered its score. The more logical choice of inclusion would be requirement "Add title to Cart".

5.4 Experiment #2: Online Mentoring System

5.4.1 Project Description

Project "Online Mentoring System" is a purpose built mentoring system dedicated to an individual university. Users can search for mentors and request some mentoring time from them. Students by default get a certain amount of tuition time free, as do Mentors. Commercial users must purchase this time in a pre-pay system, which counts down with use. During a mentoring session users have the ability to discuss via text, sound or video.

5.4.2 Step 1: Use Case Point Method

Calculate size of the project, effort and time required for the selected set of requirements using Use Case point method. This method consists of following phases:

Phase 1: Classification of use cases

UC01: An email request for appointment for both mentors and mentees.

UC02: An email request confirmation for both mentors and mentees.

UC03: Create multiple account types.

UC04: Login

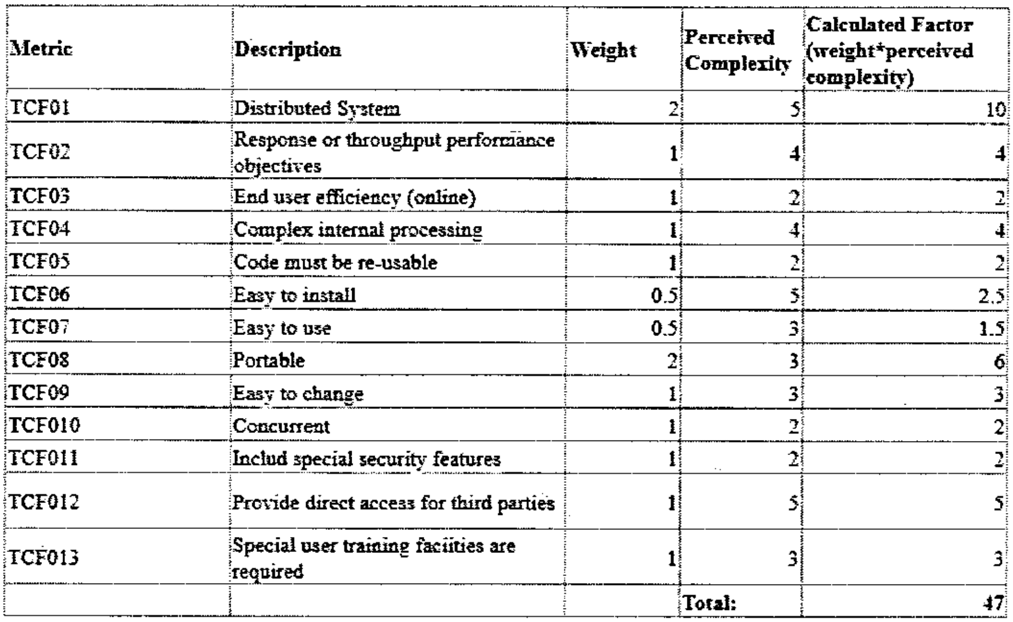
UC05: Video/Sound/Text chat on a one-to-one basis.

UC06: Payment method for purchasing credits.

UC07: Credits request.

UC08: An automated, traceable timing system for both Mentors and Mentees.

Phase 2: Determine and calculate the TCF.



Metric	Description	Weight	Perceived Complexity	Calculated Factor (weight*perceived complexity)
TCF01	Distributed System	2	5	10
TCF02	Response or throughput performance objectives	1	4	4
TCF03	End user efficiency (online)	1	2	2
TCF04	Complex internal processing	1	4	4
TCF05	Code must be re-usable	1	2	2
TCF06	Easy to install	0.5	5	2.5
TCF07	Easy to use	0.5	3	1.5
TCF08	Portable	2	3	6
TCF09	Easy to change	1	3	3
TCF010	Concurrent	1	2	2
TCF011	Includ special security features	1	2	2
TCF012	Provide direct access for third parties	1	5	5
TCF013	Special user training faciities are required	1	3	3
			Total:	47

Table 9: Experiment #2: Compute the technical factors

TCF = 0.6 + (.01*Total Factor). For table 9, the TCF = 1.07

Phase 3: Determine and calculate the ECF.

Metric	Description	Weight	Perceived Impact	TCF = (weight*perceived complexity)
ECF01	Familiar with Rational Unified Process	1.5	4	6
ECF02	Application experience	0.5	2	1
ECF03	Object-oriented experience	1	5	5
ECF04	Lead analyst capability	0.5	2	1
ECF05	Motivation	1	1	1
ECF06	Stable requirements	2	5	10
ECF07	Part-time workers	-1	0	0
ECF08	Difficult programming language	2	1	2
			Total:	26

Table 10: Experiment #2: Compute the environmental factors

ECF = 1.4 + (-0.03*TF). For Figure 10, the **ECF** = 0.75

Phase 4: Determine UUCP

Package	Name	Type	Complexity
UC01: Email Management	An email request for appointment for both mentors and mentees	UseCase	5
UC02: Email Management	An email request for appointment for both mentors and mentees	UseCase	5
UC03: User Management	Create multiple account types	UseCase	5
UC04: User Management	Login	UseCase	4
UC05: Select Books	Video/Sound/Text chat on a one-to-one basis	UseCase	10
UC06: Manage Payment	Payment method for purchasing credits	UseCase	10
UC07: Manage Payment	Credits request	UseCase	8
UC08: User Management	An automated, traceable timing system for both Mentors and Mentees	UseCase	7
		Total	54

Table 11: Experiment #2: Unadjusted use case points

For table 11, UUCP=54.

Phase 5: Calculate UCP

The UCP is calculated by multiplying the obtained variables in previous phases:

UCP = TCP * ECF * UUCP * PF

For the values used in this experiment:

UCP = 1.07 * 0.62 * 54 * 20 = 716.427 or 717 hours.

Divide the UCP by 40 hours (for one man work week) = 18 man-weeks. Hence, for the values used in this experiment, it would take one developer 18 weeks to complete the application.

5.4.3 Step 2: Hybrid Technique Prioritization Matrix

In step 2 we'll use hybrid prioritization matrix to prioritize requirements focusing on both business and technical perspective.

Rqmt	Business Perspective						Technical Perspective				Score
	Business Values (V_1 .... V_n)						Benefit	Value	Risks (R_1 ... R_m)		
	Resourcefulness	Develop People	Customer Satisfaction	Continuous Improvement	Time to market	Marketing			Tehnical	Business	
	$V_1=7$	$V_2=8$	$V_3=4$	$V_4=6$	$V_5=10$	$V_6=5$	$TF_1=7$	$TF_2=8$	$R_1=-8$	$R_2=-6$	
email request for appointment	3	2	6	0	4	4	5	5	5	7	134
email request confirmation	3	2	6	0	4	4	5	4	6	7	118
Create multiple account types (2)	5	3	6	4	7	4	6	7	6	5	237
Login (3)	3	2	8	5	7	5	8	9	6	7	232
Video/Sound/Text chat (1)	7	6	8	5	9	3	7	10	7	8	289
Credits payment methods	4	0	10	6	2	5	5	6	5	4	168
Credits request	4	3	9	3	6	5	5	4	7	8	164
Timing traceable system	5	4	7	5	8	3	7	8	8	7	227

Table 12: Experiment #2: ranking of potential top 3 requirements after implementing hybrid prioritization matrix

According to the scores in table 12, the top three requirements are Video/Sound/Text chat (289), Create multiple account types (237), and Login (232). The Result shows the benefit of considering technical perspective along with core business values, if the framework considered only the weighted business values, then requirement "Timing traceable system" which is currently at number four, would have been at number two; the risks associated with this requirement lowered its score. Top three requirements are the more logical choices of inclusion. Logically its not possible to implement "Timing traceable system" requirement at number two because we can only trace time of registered users. That is why we have to first complete requirement "Create multiple account types" and "Login" and then we can trace time for registered users. Results of both experiments are discussed in details in chapter 6 (Results and Analysis).

Chapter # 6

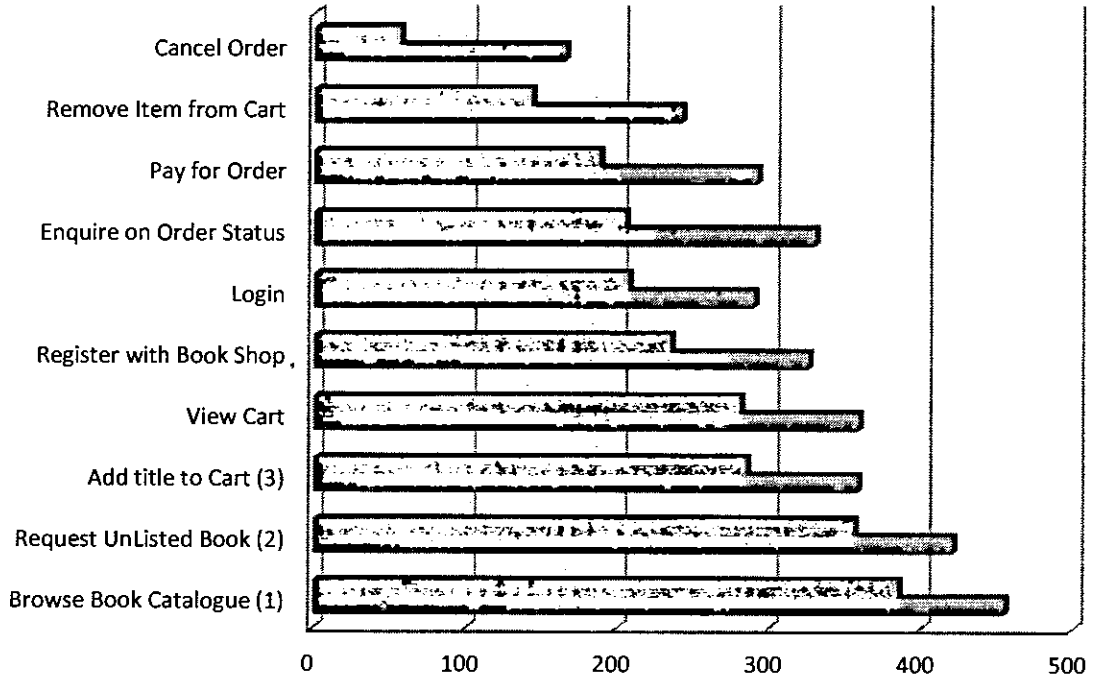
RESULTS AND ANALYSIS

6. Results and Analysis

Chapter 5 shows the results of implementing hybrid technique by performing two experiments on two different projects. To validate and prove the effectiveness of results provided by hybrid technique we've used industrial expert opinion on the same requirement set we performed experiment. We've selected 3 industrial expert based on their experience: expert #1 (beginner - less than 1 year experience), expert #2 (intermediate - between 2-3 years) and expert #3 (advance - more than 4 years). All three experts have experience in the selected project domain. The experts experience in the selected domains let them better understand and prioritize the requirements. Following are the opinion provided by experts:

6.1 Results Experiment #1; Online Book Shop

The scores in figure 4 shows the top three requirements to be Browse Book Catalogue (385), Request UnListed Book (356), and Add title to Cart (285). The Result shows the benefit of considering technical perspective along with core business values, if the framework considered only the weighted business values, then requirement "View Cart" would have been at number three; the risks associated with this requirement lowered its score. The more logical choice of inclusion would be requirement "Add title to Cart" which is currently at number 3.



	Browse Book Catalogue (1)	Request UnListed Book (2)	Add title to Cart (3)	View Cart	Register with Book Shop	Login	Enquire on Order Status	Pay for Order	Remove Item from Cart	Cancel Order
■ risks	385	356	285	281	235	207	205	188	143	56
■ business value/benefit/value	455	421	358	359	326	290	330	292	242	165

Figure 4: Graph showing ranked prioritized requirements – Experiment #1

In experiment #1 we have taken a sample project with 10 requirements and implemented hybrid technique to prioritize selected requirement set. The end results of experiment are a prioritized set of requirements according to their importance and possible sequence of implementation. We've selected three industrial experts to prioritize the same requirements set that we used in experiment according to their views and experience. Following are the prioritized order of requirements according to expert's opinion:

6.1.1 Expert #1

1. Browse Book Catalogue (1)
2. Register with Book Shop
3. Login
4. Request UnListed Book (2)
5. Add title to Cart (3)
6. View Cart
7. Cancel Order
8. Enquire on Order Status
9. Pay for Order
10. Remove Item from Cart

6.1.2 Expert #2

1. Browse Book Catalogue (1)
2. Request UnListed Book (2)
3. Add title to Cart (3)
4. View Cart
5. Remove Item from Cart
6. Register with Book Shop
7. Login
8. Enquire on Order Status
9. Pay for Order
10. Cancel Order

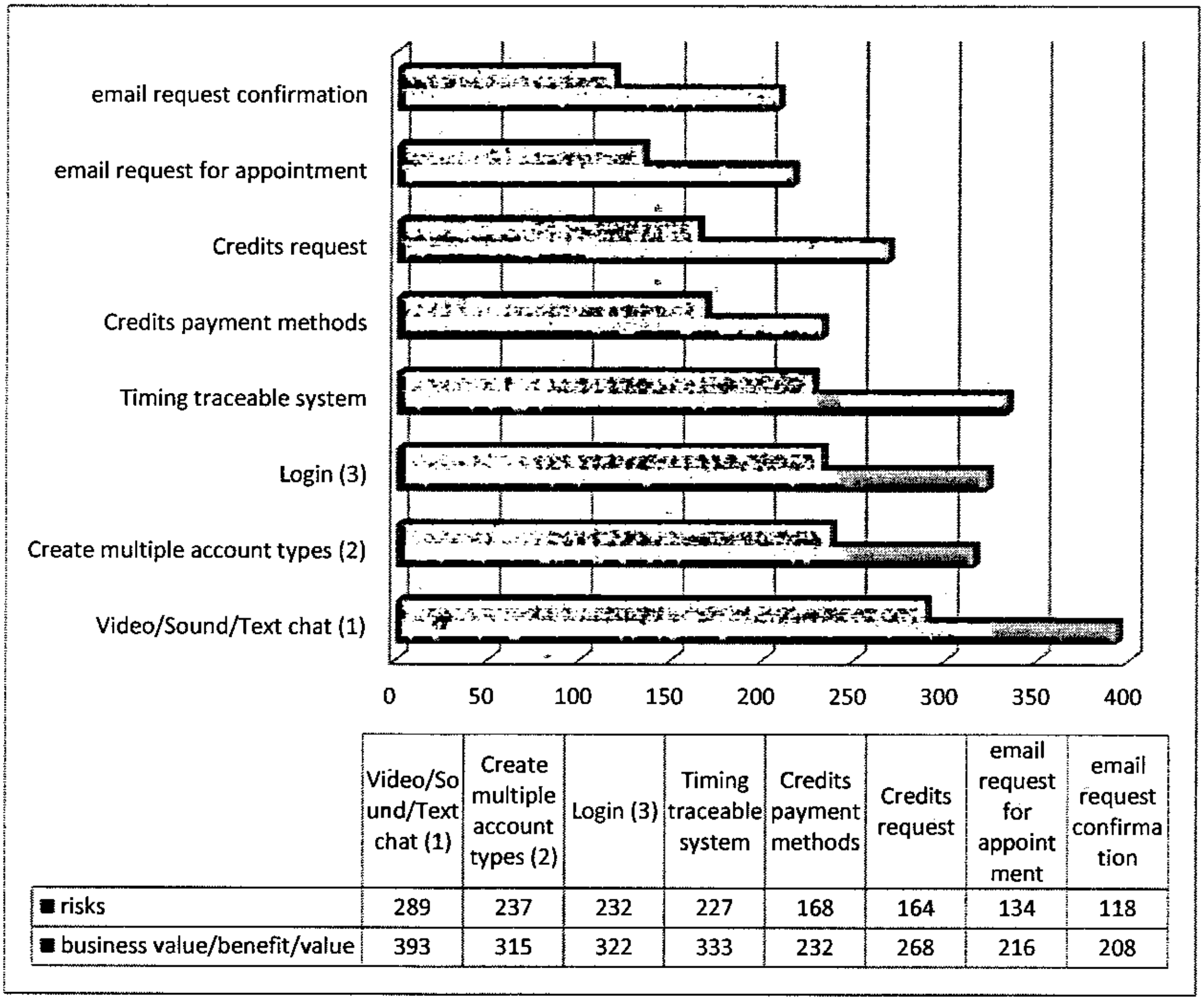
6.1.3 Expert #3

1. Browse Book Catalogue (1)
2. Request UnListed Book (2)
3. Add title to Cart (3)
4. View Cart

5. Register with Book Shop
6. Login
7. Enquire on Order Status
8. Pay for Order
9. Cancel Order
10. Remove Item from Cart

There is slight difference between the expert 1 prioritized requirement set and the one provided by hybrid technique. "Browse Book Catalogue" is on top of list in both cases but According to expert 1 opinion user registering and login process should be implemented first and then "Request UnListed Book" and "Add title to Cart". For expert #2 and 3 top 3 requirements are same. The prioritized requirement set for expert #2 and 3 are almost same apart from some minor requirements at bottom of list.

6.2 Results Experiment #2; Online Mentoring System



	Video/Sound/Text chat (1)	Create multiple account types (2)	Login (3)	Timing traceable system	Credits payment methods	Credits request	email request for appointment	email request confirmation
■ risks	289	237	232	227	168	164	134	118
■ business value/benefit/value	393	315	322	333	232	268	216	208

Figure 5: Graph showing ranked prioritized requirements – Experiment #2

6.2.1 Expert #1

1. Video/Sound/Text chat (1)
2. Create multiple account types (2)
3. Login (3)
4. Timing traceable system
5. email request for appointment
6. email request confirmation
7. Credits payment methods

8. Credits request

6.2.2 Expert #2

1. Video/Sound/Text chat (1)
2. Create multiple account types (2)
3. Login (3)
4. Credits request
5. Credits payment methods
6. Timing traceable system
7. email request for appointment
8. email request confirmation

6.2.3 Expert #3

1. Video/Sound/Text chat (1)
2. Create multiple account types (2)
3. Login (3)
4. Timing traceable system
5. Credits request
6. Credits payment methods
7. email request for appointment
8. email request confirmation

Top three requirements are same according to all three experts' opinion as prioritized by hybrid technique. There are some variations in sequence after top three requirements apart from that all three experts opinion suggests that current top three requirements should be implemented for timely and successful completion of requirements set.

The discussion in the above sections shows the successful execution of Hybrid technique on two projects. Expert's opinion proves the effectiveness of results provided by hybrid technique.

There is no major change in prioritized requirements set by experts and hybrid technique. The selected candidates for opinion are experts in the selected project domains thus they have proper understanding of projects selected for experiments. There experience in the specific domain let them effective prioritization of requirements as they have better understanding of risks and values associated with every requirement. This shows that focusing on both business and technical perspective of requirement during prioritization process helps engineers better understand and prioritize the requirements, and experts opinions proves that the validity of proposed approach.

6.3 Results and Evaluation

Based on our experimentation we've created profiling of various prioritization approaches as shown in charts below.

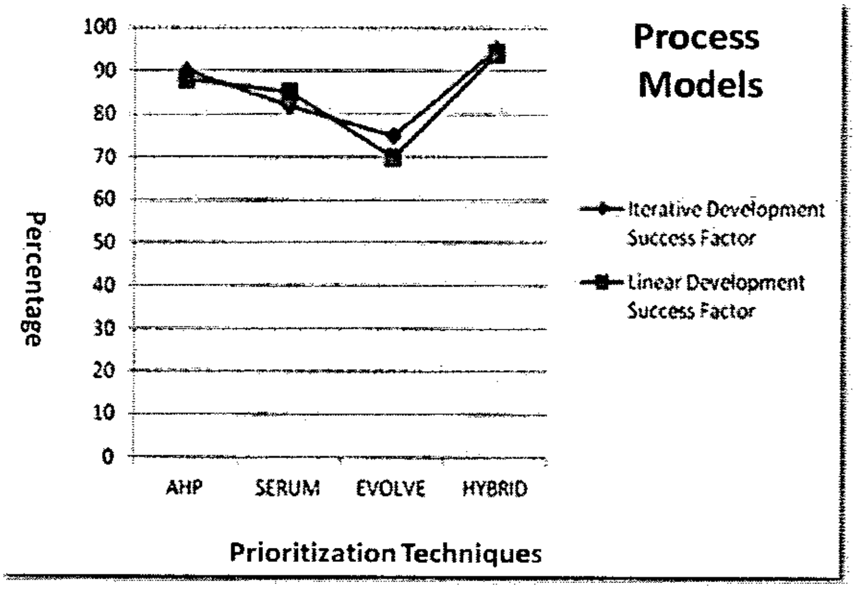


Figure 6: Experimental Evaluation Results of Various Prioritization Techniques with Hybrid technique (constraint – process models)

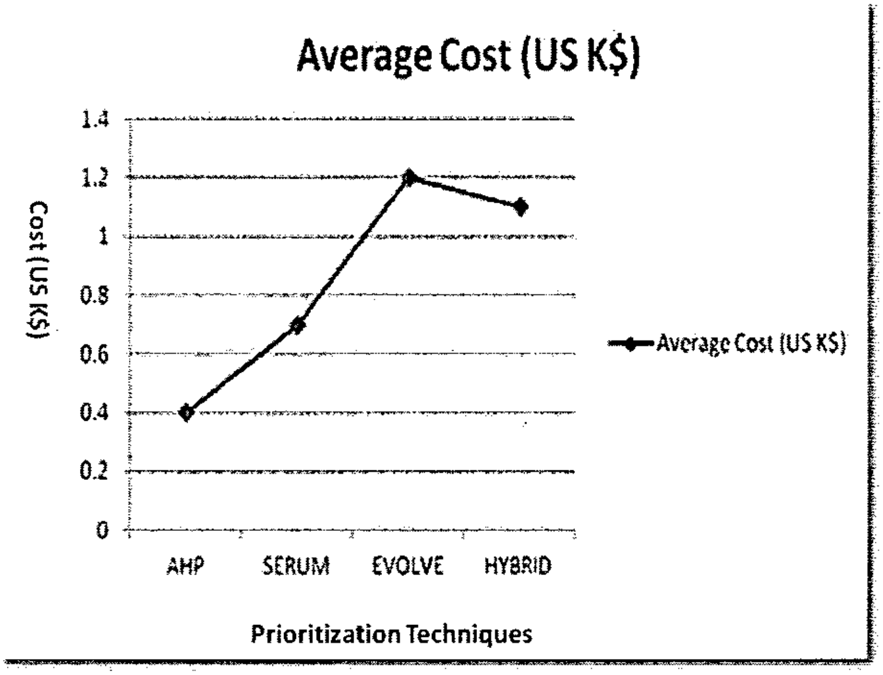


Figure 7: Experimental Evaluation Results of Various Prioritization Techniques with Hybrid technique (constraint – cost)

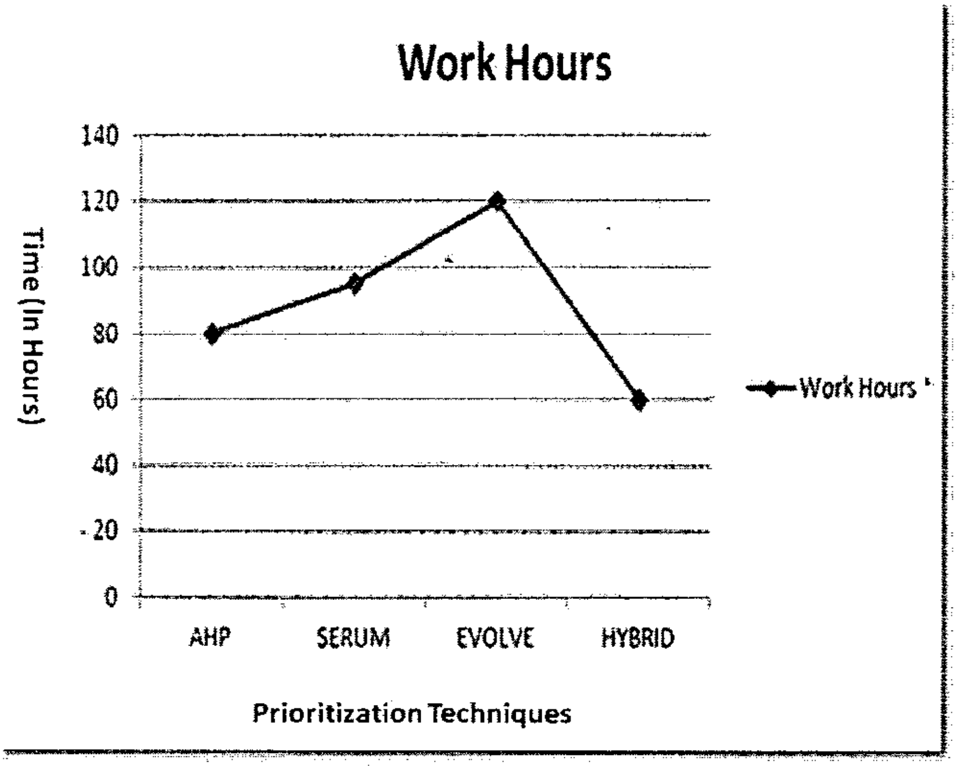


Figure 8: Experimental Evaluation Results of Various Prioritization Techniques with Hybrid technique (constraint – work hours)

Results clearly reveal that Hybrid technique is less time consuming, cost effective and produce more precise results with high success ratio for both linear and iterative process models as compared to other prioritization techniques.

6.4 Drawbacks

Previous section clearly demonstrates that hybrid prioritization technique is an effective and optimal method for requirements prioritization. However there are some drawbacks associated with hybrid technique listed below:

- Integrating technical and business perspective and stakeholders and experts involvement makes prioritization process bit complex.
- Engineer must be familiar with UseCase point method because step 1 of hybrid technique use UseCase point method to calculate size, effort and time required for completion
- Some mathematical skills are required as the process is manual

Chapter # 7

CONCLUSIONS AND FUTURE WORK

7. Conclusions

Literature shows the importance of requirement prioritization process and how the properly prioritized requirements are crucial for successful and timely project completion. Researchers have proposed many prioritization techniques and every technique is suffering from some major drawbacks and limitations. One of the major disadvantages is that the current prioritization techniques do not consider both technical and business perspective associated with each and every requirement. The experiments performed in the previous section clearly show the importance of focusing on both business and technical perspective during prioritization process. The results provided by experiments shows that the proposed technique provides optimized and effective way of requirements prioritization. Letting the stakeholders decide the business factors that are important to their organization success and let them score the requirements accordingly covers the business perspective of requirement and after that requirement engineer scores the requirements against each technical factor for better inclusion and exclusion criterion for each requirement. Experiment also shows the change in sequence of requirements after considering technical factors: benefits, individual requirement value and the risk associated with each requirement. Even though the resulting order of prioritized requirements changed slightly as scored by stakeholder but they still approves the resulting requirement set as it provides more logical inclusion criteria and clear process of requirement selection. Also, consideration of risk factor significantly improves the end results. Engineers always have to use their domain experience to assess the risk factor. If the requirement engineer has no experience in selected project domain it's not possible for them to assess the risk factor properly. In Hybrid technique engineers does not necessarily have the domain experience for risk assessment because hybrid technique provides formulated and structured approach for risk calculation by using UseCase point method during earlier phases of hybrid technique, by considering environmental complexity factor, technical complexity factor, required time and effort for successful completion of project let them better understand and score the risk factor properly in later phase of hybrid technique. This is also very helpful for engineers who don't have much experience in specific project domain. Following are some key benefits and conclusion of this research work:

- Hybrid technique provides a structured way for requirements prioritization by covering both business and technical factor and proper and effective method of risk assessment.
- Value based decision making
- More confident & comfortable decision making as decisions are taken on the basis of actual requirement value rather than mere guess and also risks are assessed based on formulated and structured approach rather than just engineer's experience and perception of risk.
- Proposed technique is beneficial for both for experienced and non-experience domain experts.

7.1 Future Work

Experimental results clearly demonstrated that hybrid technique is effective and optimal method for requirements prioritization. However there's still room for improvement. Although this process is time and cost efficient and by automating the process we can reduce the complexity of hybrid technique and make it more time and cost efficient.

References

Aaqib I, Farhan M. Khan, Shahbaz A. Khan (2009): "A Critical Analysis of Techniques for Requirement Prioritization and Open Research Issues", International Journal of Reviews in Computing.

Aaron K. Massey, Paul N. Otto, and Annie I. Anton. (2010): "Prioritizing Legal Requirements". 2009 Second International Workshop on Requirements Engineering and Law (relaw'09). IEEE

Aurum, A., Wohlin, C. (2003): "The Fundamental Nature of Requirements Engineering Activities as a Decision-Making Process", Information and Software Technology, 45(14), pp. 945-954.

Barragans, M. A. B.; Pazos, A. J. J.; and Fernandez, V. A. (2005): "Merging Requirements Views with Incompleteness and Inconsistency", Proceedings Software Engineering Conference, pp: 58-67

Carlshamre, P. (2002): "Release Planning in Market-driven Software Product Development - Provoking an Understanding". Requirements Engineering Journal 7 (3), pp. 139-151

Charette, R., "Why Software Fails". Website: http://spectrum.ieee.org/sep05/1685

Cohn, M., (2006): "Agile Estimating and Planning". Prentice Hall.

Davis, A. M. (1993): "Software Requirements, Objects, Functions and States." Prentice-Hall International, Inc., Englewood Cliffs, New Jersey.

Easterbrook S, Singer J, Storey M-A, Damian D (2008): "Selecting empirical methods for software engineering research, Chapter 11" in Shull et al.

Erdogmus, H., J. Favaro, M. Halling (2006): "Valuation of Software Initiatives Under Uncertainty: Concepts, Issues, and Techniques", Value-Based Software Engineering, S. Biffl, ed., Springer, Berlin, Germany, pp. 39-66.

Greer D., & Ruhe, G. (2004): "Software release planning: An evolutionary and iterative approach", Journal Information and Software Technology, 46(4), 243-253.

Greer, D., Bustard, D., &Sunazuka, T. (1999b): "Prioritization of system changes using cost-benefit and risk assessments", Proceedings of the Fourth IEEE International Symposium on Requirements Engineering, 180-187. (SERUM).

Hatton, S. (2007): "Early prioritization of goals. In Advances in conceptual modeling – Foundations and applications", (pp. 235-244).

Henry, J..and Henry, S. (1993): "Quantitative assessment of the software maintenance process and requirements volatility". In Proceedings of the ACM Conference on Computer Science, pp: 346–351.

Herrmann A. Daneva M. Software Eng. Group, Univ. of Heidelberg, Heidelberg (2008): "Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research". International Requirements Engineering, RE '08. 16th IEEE

Hofmann, H.F., Lehner, F. (2001): "Requirements engineering as a success factor in software projects", IEEE Software

J.Azar, R.K. Smith, & D. Cordes (2006): "Value-Oriented requirements Prioritization, tech. report, Department of Computer Science, Univ. of Alabama; http://cs.ua.edu/research.Reports.asp".

J.Azar, R.K. Smith, & D. Cordes (Jan/Feb 2007): "Value Oriented requirements Prioritization in a Small Development Organization", IEEE. Software.

J.W. Brackett. (1990): "Software Engineering", Proceedings of Software Engineering Institute, 19(1.2), Carnegie Mellon University, Pittsburgh, PA.

Jedlitschka, A. and Pfahl, D. (2005): "Reporting Guidelines for Controlled Experiments in Software Engineering, Proceedings of the 4th International Symposium on Empirical Software Engineering" (ISESE'05), Noosa Heads, Australia, 17-18 Nov, IEEE Computer Society, pp. 95--104.

Johnson, J., (2006): "My Life is Failure: 100 Things You Should Know to be a Successful Project Leader", Standish Group International.

K. Beck. (2000): "Extreme Programming Explained: Embrace Change", Addison-Wesley Laurent, P., Cleland-Huang, J., Duan, C. (2007) "Towards Automated Requirements Triage". 15th IEEE International Requirements Engineering Conference.

Karlsson, J. & Ryan, K. (1997): "A Cost-Value Approach for Prioritizing Requirements", IEEE Software September/October 1997, 67-74

Karlsson, J. (1996): "Software Requirements Prioritizing". In: Proceedings of the 2nd IEEE International Conference on Requirements Engineering (ICRE1996), Colorado, USA, pp. 110-116

Karlsson, J., Ryan, K. (1996): "Supporting the Selection of Software Requirements". In: Proceedings of IWSSD-8. IEEE.

Karlsson, L., Berander, P., Regnell, B., Wohlin, C. (2004): "Requirements Prioritization: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning". In: Proceedings of Empirical Assessment in Software Engineering (EASE2004), Edinburgh, Scotland.

Karlsson, L., Regnell, B., Wohlin, C. (1998): "An Evaluation of Methods for Prioritizing Software Requirements." Information and Software Technology 39, pp 939-947.

Laurent, P., Cleland-Huang, J., Duan, C. (2007): "Towards Automated Requirements Triage". 15th IEEE International Requirements Engineering Conference.

Laurent, P., Cleland-Huang, J., Duan, C. (2007): Towards Automated Requirements Triage. 15th IEEE International Requirements Engineering Conference.

Lehtola L., Kauppinen M., and Kujala, S. (2004): "Requirements Prioritization Challenges in Practice". Proceedings of the 5th International Conference on Product Focused Software Process Improvement. pp: 497-508

Lehtola L, Kauppinen M (2004): "Empirical evaluation of two requirements prioritization methods in product development projects". In: Proceedings of the European Software

Process Improvement Conference (EuroSPI 2004), Springer-Verlag, Berlin Heidelberg, pp.161-170

Lehtola L, Kauppinen M (2004): "Empirical evaluation of two requirements prioritization methods in product development projects". In: Proceedings of the European Software Process Improvement Conference (EuroSPI 2004), Springer-Verlag, Berlin Heidelberg, pp.161-170

Lehtola, L., Kauppinen, M., Kujala, S. (2004): "Requirements Prioritization Challenges in Practice". Proceedings of 5th International Conference on Product Focused Software Process Improvement, Kansai Science City, Japan, pp. 497-508

Lubars, M., Potts, C., and Richter, C. (1993): "A Review of the State of the Practice in Requirements Modeling". Proceedings of the IEEE Intemational Symposium of Requirements Engineering, pp: 2-14

M. A. Iqbal, A. M. Zaidi, Dr. S. Murtaza. (2010): "A New Requirements Prioritization Model for Market Driven Products Using AHP". International Conference on Data Storage and Data Engineering. IEEE.

Mohammad Shabbir Hasan, Abdullah Al Mahmood, Md. Jahangir Alam, Sk. Md. NahidHasan: "An Evaluation of Software Requirement Prioritization Techniques (pp. 83-94)" FarinRahman Panacea Research Lab, Dhaka, Bangladesh and Dhaka International University, Dhaka, Bangladesh.

Moisiadis, F. (2002): "The Fundamentals of Prioritizing Requirements". Proceedings of Systems Engineering/Test and Evaluation Conference (SETE2002).

Ngo-The, A., &Ruhe, G. (2005): "Decision support in requirements engineering", In A. Aurum& C. Wohlin (Eds.), Engineering and managing software requirements (pp. 267-286): Springer Berlin Heidelberg.

Peng Shao. (2008): "Sample Selection: An Algorithm for Requirements Prioritization", ACM.

Saaty, Thomas L (1994): "Fundamental of decision making and priority theory with analytical hierarchy process", RWS publication, Pittsburgh , PA 1994.

Siddiqi, J. and Shekaran, M. (1996): "Requirements Engineering: The Emerging Wisdom", IEEE Software, March, pp: 15-19.

Sommerville (1996).

Victor R. Basili (2006): "The role of controlled experiments in software engineering research". Proceedings of the 2006 international conference on Empirical software engineering issues: critical assessment and future directions, June 26-30, Dagstuhl Castle, Germany.

Wiegers, K., "First things first: prioritizing requirements", Software Development, 7(9), 1999. 247

Yourdon, E. (1997): "Death March Projects", Prentice Hall.

Zultner, R.E. (1997): "Project QFD - Managing Software: Development Better with Blitz QFD", 9th Symposium on. QFD, pp. 15-26