A STUDY OF FACTORS AND PROPOSAL OF NEW FRAMEWORK FOR REQUIREMENTS PRIORITIZATION FOR SUCCESSIVE RELEASES OF APPLICATION SOFTWARE PRODUCTS

Doctoral Thesis submitted
In partial fulfillment of the requirements for the award of the degree of

DOCTOR OF PHILOSOPHY

in

MANAGEMENT

by

DEVULAPALLI SITA

Under the guidance of

Research Co-Supervisor
Dr. Akhil Khare
Professor, CSE Department
MVSR College of Engineering
Hyderabad

Research Supervisor
Prof O R S Rao
Professor, FMS
ICFAI University, Jharkhand
Ranchi

ICFAI UNIVERSITY JHARKHAND
RANCHI

July 2016

THESIS COMPLETION CERTIFICATE

This is to certify that this research thesis titled "**A Study of Factors and Proposal of new Framework for Requirements Prioritization for Successive Releases of Application Software Products"**, submitted by Devulapalli Sita in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy in Management by the ICFAI University Jharkhand, Ranchiis an original work carried out by her under our joint guidance. It is certified that the work has not been submitted anywhere else for the award of any other Degree or Diploma of this or any other University. We also certify that she complied with the Plagiarism Guidelines of the University.

Dr.Akhil Khare

Professor, CSE Department
MVSR College of Engineering
Hyderabad, Telangana

Co- Supervisor

Prof ORS Rao

Professor, FMS
ICFAI University Jharkhand
Ranchi, Jharkhand

Supervisor

DECLARATION OF AUTHORSHIP

I declare that this research thesis titled "**A Study of Factors and Proposal of new Framework for Requirements Prioritization for Successive Releases of Application Software Products"**, submitted by me in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy in Management by the ICFAI University Jharkhand, Ranchi is my own work. It contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text. I further state that I complied with the Plagiarism Guidelines of the University, while preparing the thesis.

Devulapalli Sita

Date

Place

ACKNOWLEDGEMENTS

I am thankful to the ICFAI University Jharkhand for enabling me pick up my academic pursuits. I thank Professor ORS Rao, Research Supervisor, for providing encouragement throughout the research period. His review comments added value to my studies. I am grateful to Dr.Akhil Khare, Co-Supervisor, for standing by me all through my efforts to steer towards meaningful research studies. Their encouragement and guidance helped me reach the destination without getting lost. My Special Thanks are to Dr. Sukanta Swain for facilitating the research throughout. I also thank Dr. KK Nag for advising on systematic process of research.

Thanks to Prof.KCS Murthy, BITS, Hyderabad, Dr.Gopinath, Independent Software Quality Consultant, Mr. M RajeswaraRao from FedEx, and Dr. Ramansingh from Amazon Web Services for providing their valuable suggestions. Thanks to all my ex-colleagues from different organizations and other supporters who participated in my studies.

Thanks to Webonise team, for enabling me carry out my studies at their organization. Thanks go to PMP, Pune chapter and Product Management chapter, Pune for facilitating my studies during their seminars.

Devulapalli Sita

Date
Place

ABSTRACT

Software Products development through multiple release cycles goes through requirements analysis, prioritization, design, development, testing and augmenting existing products in order to meet the increasing customer needs, expanding customer base and changing technological trends and improving device platforms.

Each time a software release is planned, the contention of many requirements to be incorporated in the release needs to be resolved. .In this decision making process "Requirements Prioritization" assumes significance. The constraints of available time and resources need to be balanced with realizable Business Value in long term and short term. While a number of methodologies are proposed by Researchers, practical application of these methods for requirements prioritization is hindered as they do not indicate to easy, simple methods that are scalable, flexible and facilitate release planning.

This research focuses on identifying factors that impact requirements prioritization for the software products/applications' building and continuing to meet customers' needs. A study has been carried out with a questionnaire designed based on industry experience to gather information on practices related to "Requirements Prioritization" in software development in different organizations. Based on the Information gathered, Qualitative analysis has been carried out grouping the parameters to reflect relevant areas in product development and identify important factors for requirements prioritization. The goal of the present research effort is to evolve effective, simple and scalable Framework for Requirements Prioritization for software products development undergoing continuous changes and releases.

The thesis provides improved understanding of requirements prioritization in the context of off-the-shelf products and custom made products, based on qualitative analysis of the factors effecting prioritization of requirements. The data is grouped in to 3 datasets – large, medium, small - based on the size of the organization and comparison across the three data sets further enhanced the understanding of various factors' impact on requirements prioritization under varying nature of software developed. A case study is conducted to analyze factors associated with requirements impacting releases.

A new framework, named ABC Framework has been designed taking in to account parameters of relevance in requirements prioritization to enable multi level decision making. The design enables grouping requirements into 3 classes across 5 levels to reflect the practical development process and parameters. For enabling ease of usage, three innovative methods have been suggested to apply the framework in practice. First method represents the framework in the form of sets and requirements association in the sets. Second one has a unique number representation scheme to allow visual interpretation of the various factors' influence on the requirements prioritization. The third one enables simple use of Excel work sheets to capture classification of requirements based on parameters of importance. Comparison of the ABC framework with four significant Requirements Prioritization methods has been carried out and relative advantages of the proposed framework have been presented.

PREFACE

The research presented in this thesis identifies parameters of influence on requirements prioritization and proposes a new and innovative framework for requirements prioritization for Software products development. The framework encompasses parameters considered in industry and adopts classification into three classes across 5 layers of relevance for product development. Two new schemes of representation and visualization of prioritization based on different parameters have been presented as part of the research.

Objective of the Research Study

Most of the software Organizations currently uses simple methods such as ranking, priority grouping, which do not provide systematic, flexible, scalable methodology for Requirements prioritization in software development in practice. Uncertainties, changes in scope of requirements, multiple parameters to be considered lead to ad hoc handling of requirements prioritization. Systematic methods proposed in research are found to be complex for usage in practice and have not found wide application in practice. In order to address this gap between research and practice, and to address the gap of availability of systematic, simple and easy methods taking into consideration multiple parameters, the following Objectives have been chosen for the research.

Objective 1: To study the factors that influence requirements prioritization and elicit information on order of preference of using these factors.

Objective 2: To compare and analyze data for large, medium and small software organizations.

Objective 3: To propose a new framework – ABC Framework to enable simple and effective methodology for Requirements Prioritization for successive releases.

Objective 4: To Formulate Mathematical models for practical usage of proposed Framework

Objective 5: To Compare the proposed framework with four significant Requirements Prioritization methods.

THESIS LAYOUT

The research Thesis has 11 chapters starting with Chapter 1 introducing Software Product development area. Off-the shelf software product companies and custom product development companies are discussed in this chapter. Processes followed for product development -Waterfall, Iterative, Agile are elaborated.

Chapter 2 presents the survey of literature and description of some of the prominent methods. Recent trends in research are presented. Chapter3 discusses the objective of the research and presents the research methodology followed for the study and analysis as part of the research. Chapter 4 elaborates on the study methodology, data gathering process and nature of data.

Analysis of the study on processes and problem areas is presented in Chapter 5. Chapter 6 explores study on Requirement prioritization methods and factors. Chapter 7 compares the data across 3 datasets grouped based on size – large, medium, small from the gathered data.

As a result of the understanding of the current methods and study results, the design of the proposed framework for Requirements prioritization is discussed in Chapter 8. Advantages of the proposed framework for product development are highlighted.

An innovative mathematical modelling of the framework is presented in Chapter 9. Comparison with four methods is provided in Chapter 10. The study conclusions and further scope of research are presented in Chapter 11.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

1. Software Product Development– An Introduction

1.1 Introduction

Software products development starts with identifying specific needs of customers, and cycles through analyzing the extent of each need, designing how these needs can be met in the product, developing some or all of the needs as feasible based on the available resources in terms of time, knowledgeable teams and capital. Various activities, such as requirements analysis, product features design, development and testing and release to customers are planned either sequentially or in parallel in iterations based on the process methodology followed – water fall, iterative, agile, etc.

Software Products developed for domain specific Applications, such as engineering, finance are complex and are related to specific domains, whose core principles undergo changes less frequently. Typically these Software products start their life with a specific domain application/customer in mind and grow around that application and/or similar applications. And add similar or aligned customers to the products usage.

The products mature in terms of stability and reliability as they are used by different customers for different applications in different environments. At the same time, products under go modifications to meet further requirements of existing customers and new customers. Products also undergo changes to incorporate the benefits of advanced technologies. Providing the customers with ever enhancing products is made possible by successive releases of products at varied intervals. Typically successive releases are planned once or twice a year. In the intervening time there are alpha, beta programs, or even frequent interactions with stakeholders – customers, for fine tuning and refining the release at different intervals.

The fundamental questions that need to be addressed are - What will be available in the next release? How are the requirements evaluated? Can all the requirements we gather be incorporated or we have cost and time implications, which require us to prioritize the requirements? What parameters need to be considered for prioritization? Who makes the decisions? And when?

It is also important to understand that requirements prioritization is not a onetime activity for a release or set of releases, but a continuous activity that starts from understanding and evaluating business value of different sets of requirements and involves decision making in order to maximize the business value of the decision to implement a certain set of requirements. And a decision is likely to undergo changes under changing circumstances, influencing change in the requirement sets decided upon. Hence it becomes important to not only prioritize, make decision appropriately, but also be flexible to incorporate changes with ease, be able to visualize the impacts of changes and reprioritization.

NASSCOM's 2015' Report on Software products states there are more than 4000 product firms in India, with 300 to 400 being both product and services firms. There are 300+ MNCs active in domestic market according to NASSCOM. India's software product segment is dominated by integrated firms and MNCs dominate domestic market in revenue terms. The expertise of software products development, be it off-the –shelf or customized solutions, lies largely with the MNCs and large organizations. Gathering this knowledge from various firms in the important area of requirements prioritization in products development is attempted in this research, in addition to coming up with innovative methods for requirements prioritization. Due to the proprietary nature of work carried out in product development, information gathering posed great difficulties and targeted audience of 106 participants in about 61 organizations could be reached over a period of 2 years with difficulty.

1.1.1 The story of SAP Product Development

SAP – System Analysis and Program Development, a company started by 5 ex-IBM employees in 1972 started with the development of a standard application software for real time data processing, that is processing data when customer asks for it instead of in batch processing at nights and had one customer in Germany. In 43 years, as of 2015, SAP has a customer base of 296,000 World over.

In 1973, SAP completed its Financial accounting system. SAP used IBM servers and DOS operating system. In 1974, SAP converted its Software to be used on OS operating system and had about 40 reference customers. By 1975, SAP enabled companies to integrate their purchase, inventory management and invoice verification with its accounting system. In 1977, SAP expanded customer base to outside Germany. In 1978, SAP added asset accounting for a pilot company. John Deere, a customer of SAP developed French Language version for SAP accounting. In 1979, it started developing mainframe application SAP R/2 with IBM's database. In 1981, it added sales and distribution modules. Collaborating with customers it also added production management module. 250 companies in Europe used SAP by 1980s.

In 1983, HR management was added to SAP stable. By 1988, SAP reached 1000th customer and other countries including USA. In 1989, SAP introduced user friendly SAP R/2. By 1991, SAP R/3 – with client server concept graphic interface, relational databases enabled SAP to address mid size companies and branches and subsidiaries. Between 1992 and 2001 SAP is ported to Japanese language and different hardware platforms. Microsoft Windows version of Sap R/3 is released. In 1996, SAP was made accessible through online applications. In 1999 initiative to combine e-commerce applications with Sap's ER applications – mySAP.com started. 2004 brought in SAP NetWeaver to support end-to-end business processes across systems. By 2011, SAP came up with in-memory computing for real-time data analysis.

SAP story illustrates how the software product development went on from a single module catering for a specific customer to many modules working for customers globally. It throws light on adaption to changing technologies and changing platforms and systems – from mainframe to client server to web to cloud. The processing moved from real time data access to real time data analysis.

1.1.2 Other Software Companies

The Software products companies which started in late 1960s or early 1970s went through similar cycles of changes for their products. Intergraph which started with mapping for the city of Nashville in USA, created both software and hardware for graphic information display. Intergraph came up with several Enterprise products for engineering domains – Plant design, Geographical Information systems, Civil Engineering, Asset management - on clipper based workstations. Intergraph continued to adapt to changing technological advances in both Hardware and software, and increased customer base and demands, from mainframe and dedicated and proprietary databases and formats to client server and open interfaces and then to web world.

The software products developed by SAP and Intergraph are off-the shelf products with customization provided on top of the products for customers. Companies like SAP and Intergraph released newer versions of their products with new features once in a year or more frequently or less frequently based on the nature of the features they incorporated in the subsequent versions. And both the companies continue to release their products newer versions.

1.1.3 Custom Software Development

Software solutions being built for various medium to small size organizations to enable them to leverage software solutions for their businesses are often taken up by startup or small companies. While large companies offer generic solutions as products surviving through years and provide customization for specific business needs, there is a good mix of new customized solution offerings developed anew by companies as well as customized solutions on generalized solutions meeting the needs of IT enablement of business. Similar to off- the-shelf products' initial versions, the development of software starts as a solution development and continues to undergo enhancements and fixes, thereby evolving into business specific products. They are certainly not one time buys and live through versions of modifications till scaling of business demands a new solution or simpler and new technology based solutions are needed. And the cycle of new product solutions begins as illustrated in Fig.1.1. Through these cycles, Requirements are gathered, analyzed, refined, and prioritized as per client's business needs, technology changes and resource needs.

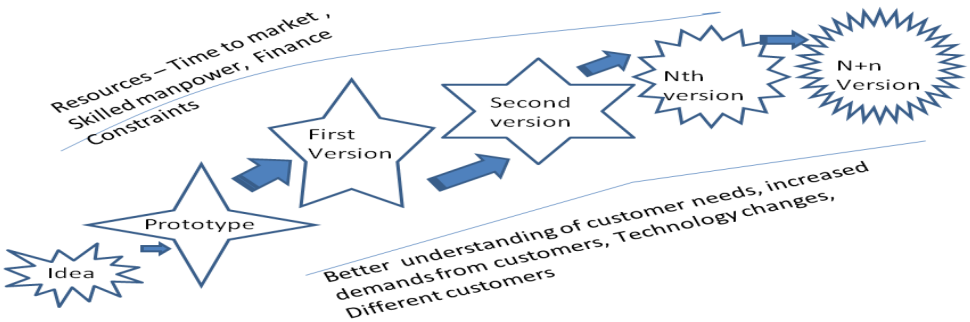

Fig.1.1.Successive Releases - Requirements

1.2 Software Development Methodology

Both Off-the-shelf product development companies and Custom Software development companies also evolved in the processes followed for software development. Traditionally Waterfall process is followed, where in understanding the requirements completely and then designing the product, then developing the product and releasing to customer spanning the complete development cycle. Iterative process emerged to enable visibility and scope for modifications during the development cycle. Agile process is followed currently by many companies to facilitate flexibility in feature release cycles and changes in requirements, by involving customers during the development cycle. The three processes- waterfall, Iterative and Agile are described in the section 2.1 Typical software product development life cycle in practice is depicted in Fig. 1.2 below. The diagram depicts across the processes in general and various actors involved and feedback and action points vary in degree across the processes. The cycle of the development process may vary in duration across processes.

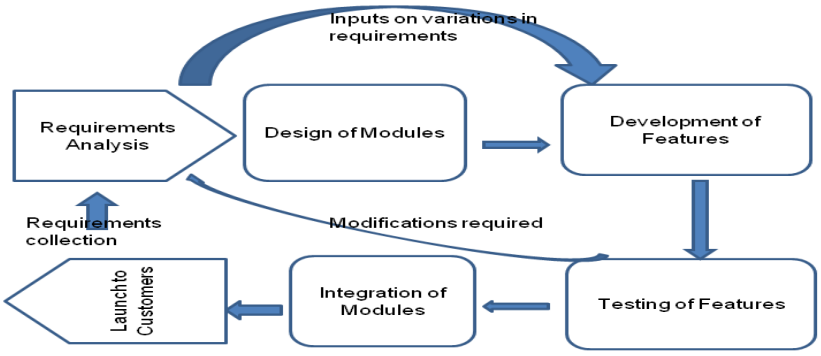

Fig. 1.2.Software Product development life cycle

Typically, Software product development started from addressing specific requirements of specific customers. As the customer's needs expanded or changed there has been a need for adding new features or enhancing the initially developed product. The product thus developed also met other customers' needs in similar industry needs. As the customer base increased, new requirements such as adapting to local languages came in. Customers in different countries had varying needs. In general, the requirements are gathered from various sources as depicted in Fig. 1.3 – the sales, marketing team which is in touch with customers providing a list of needed features and the value these features provide to the customer, the maintenance or customer help desk team with problems faced by the customers with current product version and additional requests from the customers. Executive management provided priority areas and direction for the upcoming releases based on business value. Development team added inputs on new technologies, impacts and efforts involved for new releases.

As seen in the Fig. 1.3 the requirements could be many which, it may not be feasible to get developed into product in the time the customers need them and with the resources available at hand. Hence the need to prioritize the requirements as indicated in Fig. 1.4 to zero in on the set of requirements that can be developed incorporated in the product solution.

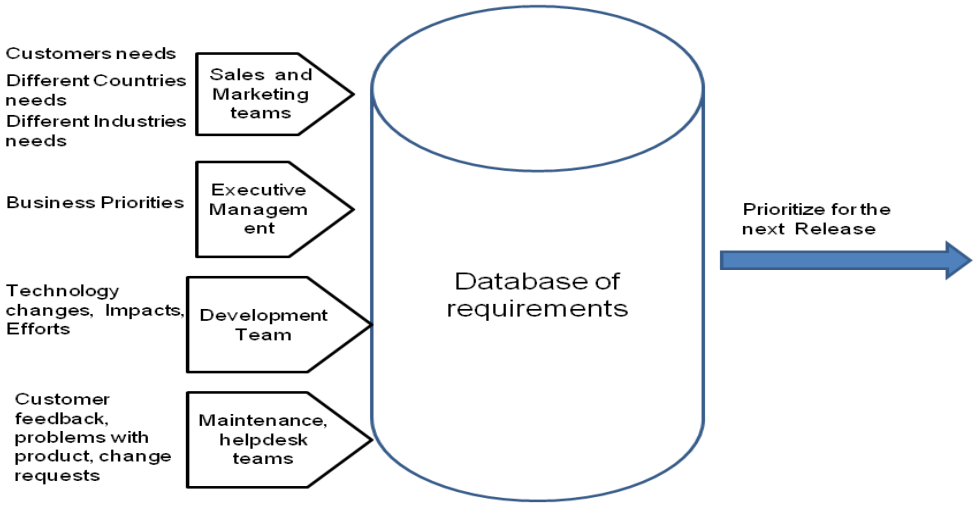

Fig.1.3. Requirements Channels

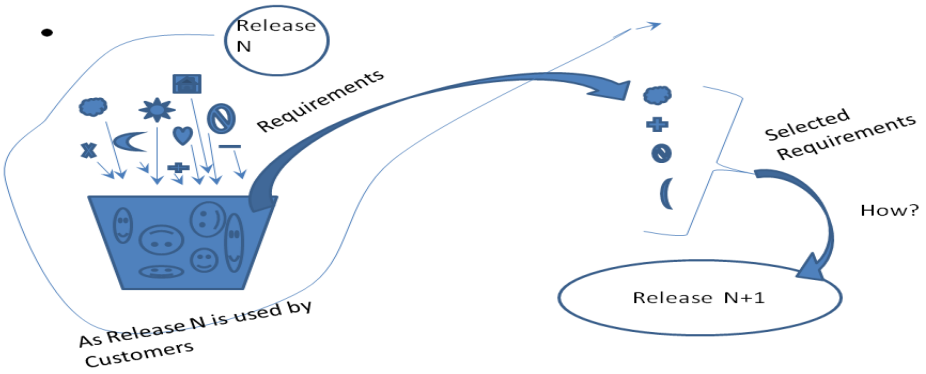

Fig.1.4. Need for Prioritization

1.3 Development Processes

1.3.1 Waterfall

Waterfall has been the traditional process followed for software development and especially for enterprise applications and mission critical applications. This process required detailed requirements analysis before starting to build the software. The time that elapses between requirements collection and analysis and final product being available to clients being considerable, this method has the problem of the inability to reflect changes in requirements into development quickly. Fig. 1.5.

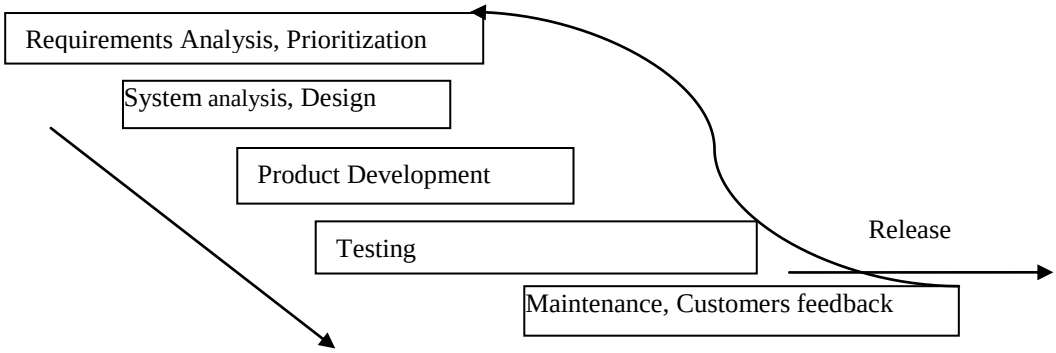

Fig.1.5. Water fall process

1.3.2 Iterative

Iterative software development process enables development in iterations and enables feedback and modification inter-steps. This enables early corrections and module wise development. Iterative process is illustrated in Fig.1.6.

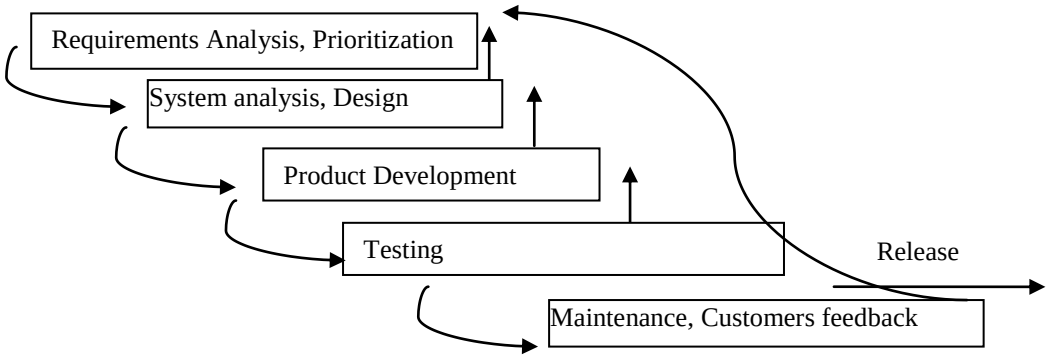

Fig. 1.6. Iterative process

1.3.3 Agile

Agile process facilitates development to start in prototyping mode along with requirements elicitation and analysis as stories, involving stake holders in the process closely. Requirements are captured in to backlog list and are prioritized for each iteration. Iteration can be as short as a week. The process is expected to enable quick development, easy refactoring. This process may suit well for innovative and new conceptual products/solutions development, where requirements are not clear or evolving. The process may end up in developing mismatched systems in case of enterprise systems that need appropriate system analysis. Fig. 1.7 depicts Agile process.

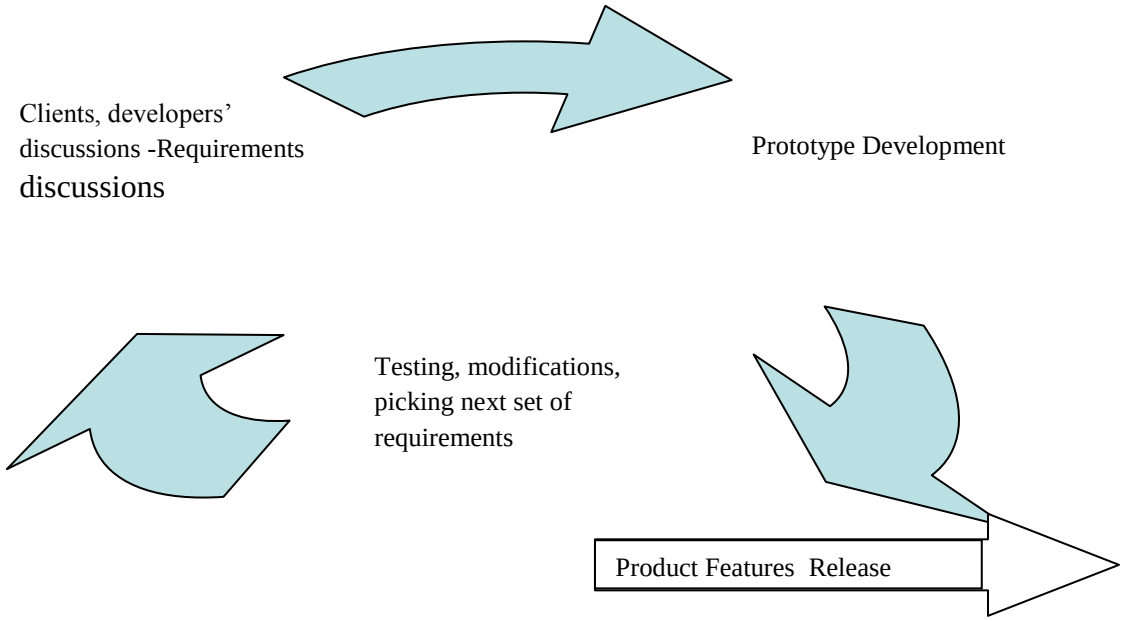

Fig.1.7. Agile Process

1.4 Requirements Prioritization

With any of the process followed, software development needs to put together the requirements, analyze, and prioritize in order to develop tangible product within reasonable time with available resources. If unlimited time is available and unlimited resources are available, there would be no requirement for requirements prioritization, which is not the case in practice. Not only the limited time to develop or the limited resources, Business Value needs to be considered for the requirements in order to maximize the ROI on software development. The requirements take some finite time of resources to get developed, hence development time needs to be considered. If the development is for successive version of the product/solution, the analysis needs to consider how the requirements are going to impact existing customers and also existing components. Factors influencing Requirements prioritization are illustrated in Fig. 1.8.

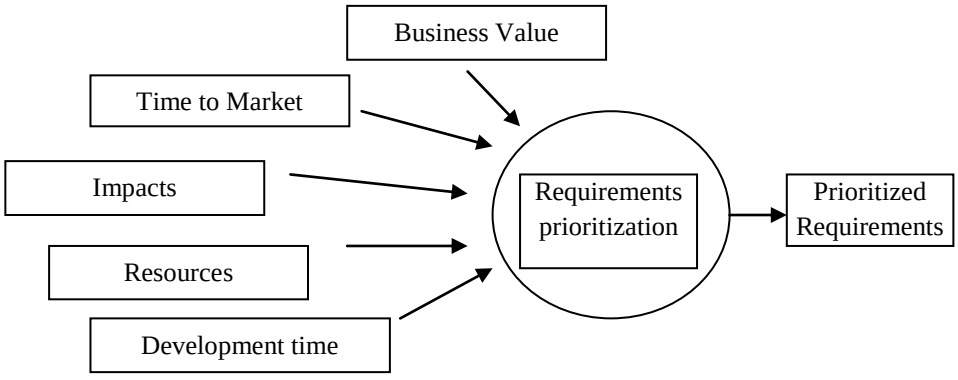

Fig.1.8. Requirements Prioritization

In this cycle of product solutions development, often there is less clarity on requirements in the initial stages and requirements change frequently in nature and scope. Changing business needs during the development phase also results in changes in requirements. Chasing the changes in requirements often results in increased development efforts, over worked teams and extended release dates. In order to understand the requirements handling process during the software development a study is designed to gather current methods, difficulties faced and solutions adopted. Requirements prioritization Methods proposed and discussed in literature are discussed in the next chapter.

1.5 Summary

Software products development as it happens in industry is described in this chapter. Software development of Off-the shelf products development, Custom software development, and Solutions development is detailed. Products and solutions' development methodology, processes followed in general are described. Processes – waterfall, iterative, agile are briefly explained. Factors influencing prioritization of requirements and uncertainties with requirements are touched upon. Requirements prioritization Methods proposed and discussed in literature are discussed in the next chapter.

Chapter 2

2. Software Requirements prioritization – Methods in Literature

2.1 Introduction

Significant Research and empirical studies have taken place in the area of requirements prioritization [27]. Methods have evolved for prioritizing requirements based on different parameters [6] - Value and Cost being prominent among them. The Requirement Prioritization Methods proposed and discussed in literature can be divided in to two groups. The first set being fundamental methods such as Cumulative Voting, AHP, Cost-Value method, Numerical assignment, Priority grouping, Wiegers method, Triage. The second set comprises of methods which are a combination of the above methods and adding influences of different mathematical areas such as Fuzzy Logic, Genetic Algorithms, and Probability Theory. It is difficult to say if the methods of second set have been validated extensively and are in use in practice. In fact, the methods of first set themselves do not have accurate comparison validated. Since applying specific set of methods to similar requirements which are reasonably well analyzed and are at the similar abstraction level in an experiment has been a difficult step in practice for research. The following sections highlight recent methods presented in literature followed by fundamental methods description.

2.2 Fundamental Requirements prioritization Methods in literature

Cumulative Voting method allows different stakeholders to distribute points from a fixed number such as 100 units among requirements according their priorities. This method, ideally, can capture the real needs of stakeholders. Due to the subjective nature, there is possibility of manipulated priorities and difficulty of prioritizing when requirements are many.

Priority Grouping or Numerical Assignment allows requirements to be grouped in to three or more groups based on either a single criteria or a weighted combination of multiple criteria. In its simplest form of grouping in to priority 1(P1), priority 2(P2), priority 3(P3) all the requirements, is the generally used method in software development. P1, P2, P3 can as well be called critical, standard, optional or must have, good to have, need not have or requirements that delight customer, that satisfy customer, that dissatisfy customer. Within a group all requirements will be of same priority. It is possible to group requirements at multiple levels hierarchically and generate finer priorities, with sub grouping.

In Ranking method, all requirements are ranked from 1 to n, prioritizing uniquely each requirement. Here the rank does not indicate relative importance. Ranking is used often due to its simplicity. Combination of the methods is also used for prioritization like Planning game combining grouping and ranking.

Analytical Hierarchy Process- AHP of Saaty [34] is based on pair wise comparison of requirements relative to each other on a scale at successive levels of hierarchy. Cost-Value approach by Karlsson [12] takes the cost of implementation and value of requirements in to consideration in pair wise comparison. Wiegers method [11] proposes risk weighted cost/value ratio for determining priority. Priority Groups method [1] categorizes requirements based on ranking different parameters – mostly importance of requirements and are put in groups.

Davis advises simplifying the process and advises Triage at successive levels, taking into account market realities [1]. Other methods frequently discussed in literature are Planning Game, Planning Game combined with AHP, 100 point method or Cumulative Voting. Comparison of various methods is taken up systematically in [13]. Industry specific studies for products meeting certain specific base parameters seem to have been very few [31] [2]. This makes the conclusions and comparisons difficult to be applicable or reliable. Triaging of requirements, Priority grouping, Cost-Value method, Wiegers method and AHP are described in this chapter. Few other methods like – Win-Win method, Planning Game, Cumulative Voting/100 points, are added in brief.

2.3 Requirements Prioritization Methods for Comparison

Laura Lehtola [16] in his article on Suitability of Requirements Prioritization Methods for Market Driven Software Product Development divides the prioritization approaches roughly into two categories – methods based on giving values to different factors of requirements and negotiation approaches. The methods based category is further subdivided into two subcategories, one with methods which process each requirement uniquely and the other with methods based on comparisons as illustrated in Fig. 2.1. Weigers method, Priority grouping fit in first category and AHP, Karlsson's cost value pair wise comparison falls in to second category. Negotiation approaches determine priorities based on discussions, negotiations among different stakeholders. Win-Win Model is illustrated as an example of negotiation method in this chapter and Cumulative voting or 100 point test is described in brief.

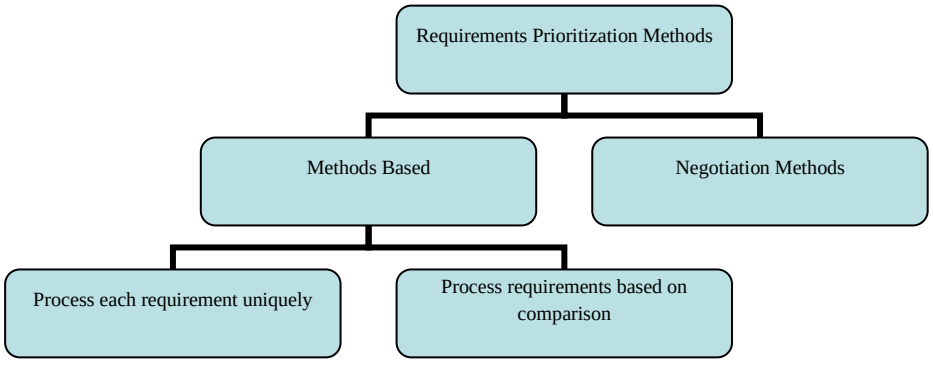

Fig. 2.1. Prioritization methods - Types

2.3.1 AHP

Saaty [34] espouses the usefulness of paired comparisons in combination with hierarchical structures in deriving measurement. Analytical Hierarchy Process (AHP) of Saaty is a multi criteria decision making method. Problem space is expected to be captured as a

hierarchical structure to enable decision maker compare homogeneous elements in each level. According to Saaty, the most creative task in making a decision is to identify the factors important for the decision. Factors relevant to decision making are arranged in the hierarchical structure that flows from overall goal to criteria to sub criteria and alternatives in successive levels. Saaty proposed in his original paper to avoid mere number crunching for decision making and demonstrated how to derive relative scales using judgments given in the form of pair-wise comparisons.

Each requirement is compared to each other requirement in AHP method to determine the importance of the requirement. This leads to pair-wise comparisons in AHP. The number comparisons increases in proportion to the square of number of requirements. Karlsson Adapted AHP in his Cost-Value method with 2 factors – cost and Value. This method is described after AHP.

The scale as defined by Saaty for pair wise comparison of the requirements is described in Table 2.1. Five levels of judgment - 1, 3,5,7,9 corresponding to equal value, slightly more value, strong value, very strong value and extreme value respectively are defined in the scale. 2,4,6,8 provide intermediate values when compromise is needed. In pair wise comparisons, reciprocal of assigned number of one requirement becomes the priority for the pair's other requirement. If there are ratios arising out of comparisons, rationals support the scale.

Table 2.1. Judgement scale

Importance level	Definition of judgments
1	Equal Importance
3	Moderate importance of one over the other
5	Essential or strong importance
7	Very strong importance
9	Extreme Importance
2,4,6,8	Intermediate judgments
Reciprocals	S(j,i) = 1/S(i,j)
Rationals	Ratios arising from the scale

In order to apply AHP for Requirements prioritization 4 steps are involved, that are described briefly here. Requirements are arranged in a matrix form, representing columns and rows as a first step. Pair-wise comparison judgments on the above scale of each requirement with another form the values in the matrix. Step 2 involves comparing each requirement with other one using the scale values. Step 3 involves deriving the priority matrix, which are Eigen values of the matrix arrived at by using averaging over normalized columns. Relative value is assigned to requirements based on the priority in step 4. Illustration in Table 2.2 to 2.4 in the section below provides the workings of AHP.

AHP generates relative ratio scales of measurements from measurements of a set of objects on a standard scale through normalization.

2.3.2 Cost Value

Karlsson and Ryan [12] proposed a two factor method based on AHP. Value that the requirement offers if implemented and Cost of implementation are taken as the factors for decision making. Here the decision is prioritizing the requirements. Requirements are compared based on each factor independently. Both Cost and Value Factors' relative priorities for the requirements are arrived at as illustrated by Karlasson-Ryan - the steps are presented in Table 2.2 to Table 2.4 for the factor - Value. Table 2.2 presents the pair wise comparison and assigned values based on scale of AHP. Table 2.3 presents the step of summing the normalized column values in order to arrive at Eigen values. Table 2.3 derives Eigen values from step2 by averaging the column values. Eigen values represent the Relative Priorities of requirements as shown in Table 2.4.

Table 2.2. Pair wise comparison –assigning Relative priorities

Req.	R1	R2	R3	R4
R1	1	1/3	2	4
R2	3	1	5	3
R3	1/2	1/5	1	1/3
R4	1/4	1/3	3	1

Table 2.3. Eigen-value estimation Process

Req.	R1	R2	R3	R4	normalized column sum
R1	0.21	0.18	0.18	0.48	1.05
R2	0.63	0.54	0.45	0.36	1.98
R3	0.11	0.11	0.09	0.04	.34
R4	0.05	0.18	0.27	0.12	.62

Table 2.4. Relative Value

Requirement	Averaged normalized column sum = Eigen Value	Eigen value	Relative value
R1	1.05/4	.26	26%
R2	1.98/4	.50	50%
R3	.34/4	.09	9%
R4	.62/4	.16	16%

Relative values are plotted in a cost-value diagram in this method, which is used as a conceptual map for identifying requirements to be taken up for implementation. The diagram as illustrated by the authors is presented in Fig. 2.2. This information can also be utilized for strategizing release plan, according to Karlsson and Ryan.

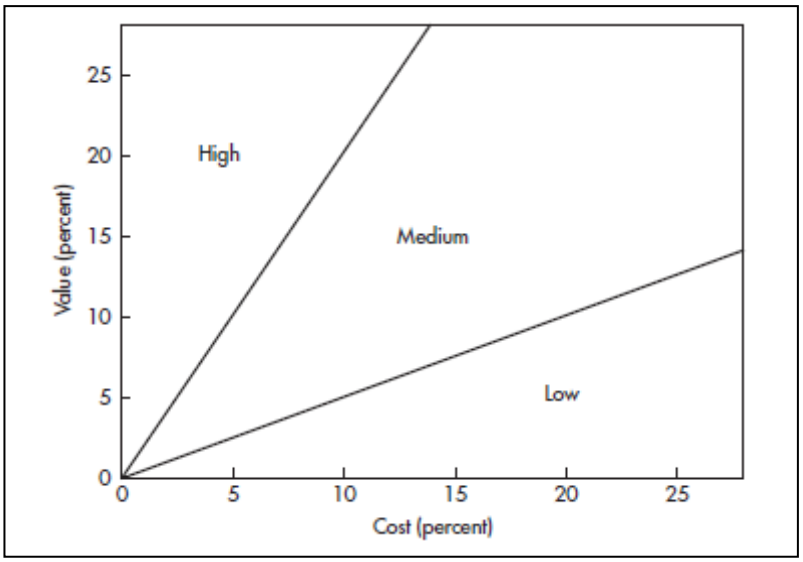

Fig. 2.2. Cost-Value Diagram of Karlsson-Ryan

This method takes into account 2 factors. While it is true that these 2 factors can in turn encompass all other factors implicitly, inter-dependencies of requirements would cause reevaluation of prioritized requirements for feasibility of implementation.

2.3.3 Priority grouping

While a rigorous relative prioritization values are achieved in AHP and AHP based cost-value methods, cost-value diagram indicates to grouping the requirements into high-medium-low priority groups. In Priority grouping method, requirements are not compared to each other based on a criteria, but are grouped into either three – low, medium, high priority groups

or essential/conditional/optional groups or four – most needed, good to have, ok to have and not to have - priority groups based on importance of requirements. Each group can further be grouped within to arrive at finer clusters of requirements. And this sub-classification can extend and form a hierarchy of levels. Whether the criteria at each level will be importance, which can be a combination of different criteria pre-determined or the criteria can be different for sub-grouping is not explicitly discussed in literature.

The decision making in classifying into groups is subjective in this method. Two of the generally used priority grouping scales are given in Table 2.5.

Table 2.5. Priority groups

	Group Levels	Criteria
Scale -1	High	Mission Critical
	Medium	Required, but can wait
	Low	Can live without
Scale - 2	Essential	Requirement is a must
	Conditional	Not unacceptable without
	Optional	May or may not be there.

Discussing the Kano-Analysis for prioritizing requirements under priority grouping with four groups may not be incorrect. Kano analysis tries to identify and contrast essential requirements from incremental requirements as a function of customer satisfaction. Kano's four categories are given in Table 2.6. Catering for Must be features in the initial versions, More is better based on Return of Investment, Surprise and Delight to gain more customers has been suggested in this model.

Table 2.6. Kano groups

Groups	Criteria
Surprise and Delight	Features of differentiation from competition
More is Better	Features offering increasing utility
Must be	Essential features needed by customer
Better not be	Features of dissatisfaction to customer

2.3.4 Wiegers Method

According to Wiegers, customers prioritize initially from the perspective of value and once the cost, technical risk are evaluated, the priorities change. Wiegers suggests balancing of business benefit of requirements against its cost and any implications it has for the products core structures. Wiegers semi quantitative, analytical approach distributes a set of estimated priorities across a continuum rather than grouping them into a few priority levels. Risk adjusted value/cost ratio is used to determine priority in this method. A features attractiveness is directly proportional to the value it provides and inversely proportional to its cost and technical risk of implementation. Weiger suggests applying this method to only negotiable features and not to core business functions or requirements that require compliance with Government regulations. Requirements are listed in a sequence at the same level of abstraction and four parameters – value, penalty, cost, risk are measured for each requirement on a scale of 1 to 9 , 1 being lowest value and 9 representing highest value on the scale. Sample prioritization is shown in Table 2.7. Relative weight is weighted twice in calculating the total value, which includes a combination of benefit of implementing and penalty if the feature is not implemented. While cost indicates to development cost, risk indicates to the risk associated with feature – technical feasibility, resource availability.

Table 2.7. Wiegers Prioritization

Requirement	Relative Benefit	Relative Penalty	Total Value	Value %	Relative Cost	Cost %	Relative Risk
R1	6	4	16	39	3	38	1
R2	9	7	25	61	5	63	3

The Priority is calculated as

Priority = value% /(cost% *cost weight + risk% * risk weight)

Weiger indicates the method is not mathematically rigorous and is limited by the ability to estimate the 4 parameters for each requirement and suggests it should be used as a guideline to make trade-off decisions But this is the same limitation for all the methods using a scale to estimate on different criteria. Wieger points that the method can become unwieldy beyond several dozens of requirements and suggests initial and sub-lists analysis for ease of prioritization.

In this method Value includes the –ve value or penalty for not implementing. Cost is expected to take into account existing modules benefit, risk includes impacts.

2.3.5 Triage Method

In his article “Requirements Management made Easy” Davis emphasizes the need for simplicity in requirements management and states

“Requirements management has been discussed for at least fifteen years. As a discipline and as a practice, it has become more and more complex. We have lost sight of the fact that requirements management was created to *simplify* software development, to reduce its cost,

and reduce the inherent risk associated with building software. Instead, requirements management has become yet one more chore, one more error-prone activity. The purpose of this paper is to distill the common wisdom of requirements management, and to return it to a simple method of ensuring that software development organizations build the right software."

Requirements triage can be as simple as development managers estimating effort and time required to implement the required features and retaining the features that fit in the schedule and budget and removing the rest. Since this simple triage impacts the revenues and profits, triage needs to arrive at a set of features that can be implemented using available resources with acceptable levels of risk which can be sold at an acceptable price to a known market, in sufficient quantities to achieve acceptable levels of revenue and profit. According to Davis, the variables that are at the disposal of the team performing triage are - a feature can be added or deleted and changed; delivery date can be advanced or postponed; resources can be increased or decreased; price of product can be increased or decreased; Triage is the process of altering assumptions about these variables. A product plan is produced then, with features to be included, markets to be addressed, resources needs and revenue generation expectations. Davis defines Triage as "The art of selecting the right features to include in the next release, balancing the requirements with development cost, risk; schedule; market, sales, revenues, pricing, profits, ROI". Triaging process is depicted in Fig. 2.3.

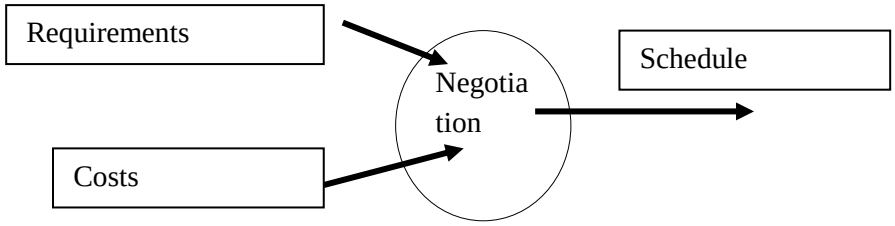

Fig. 2.3. Triage

Literature study points to simple to complex methods being researched and gap between research and practice due to difficulties in implementing the research methods. There is also lack of appropriate bench marking based on nature of product domains. The methods target one time prioritization with accurate information being available. Scaling is an issue with these methods, in that the number of calculations and comparisons becomes cumbersome. Re-planning under changes requires rework.

2.3.6 Other Methods

2.3.6.1 Win-Win Method

Win-win or Theory W method enables every stake holder to be a winner. Related to Requirements prioritization, the stake holders rank the requirements independently at first. Re-ranking is done in successive rounds under the constraints and goals to arrive at conflict free requirements prioritization.

2.3.6.2 Planning Game

Planning Game is used in deciding what to develop for the next release in Agile's Extreme Programming projects. Requirements elicited from customers are prioritized by the

customers in to 1. Must have, 2. Less Essential, but high value 3. Nice to have groups. At the same time development team estimates time required and groups the requirements into 1. Precisely estimated 2.Reasonably estimated 3.Cannot be estimated. Customers prioritize requirements based on these groups and release date needs.

2.3.6.3 Cumulative Voting/100 point method

This method takes into account stakeholders preferences in terms of number of points assigned to each of requirements out of the given total number of imaginary value units, typically, 100 points. User preferences may be skewed in this method and over-weigh some of the requirements. While the method is easy, simple, takes into account stakeholders' preferences, it is difficult to apply to a large number of requirements. Hierarchical cumulative voting can be used to apply cumulative voting of requirements at different levels of hierarchies.

2.4 Recent Requirements prioritization Methods in literature

Ruby and Balakishan in their article on "Role of Fuzzy Logic in Requirements Prioritization" [32] provided rudimentary comparison of some the methods and proposed fuzzy logic for requirements prioritization, though have not specified "how". Bhagyashree Javale and Ashish T Bhole [4] in their paper " Adaptive Fuzzy Hierarchical Cumulative voting" attempted to propose adaption to the combination of Cumulative Voting, fuzzy assignment of priorities, Hierarchy in terms of high level and detailed level requirements. The adaption process is not described in the paper.

Mohammad Dabbagh, etc.[20] looked at handling functional, non functional requirements separately with Integrated prioritization approach(IPA) and Hybrid Assessment Method (HAM) and compared with AHP for time consumption, accuracy of results, ease of use in "Functional and Non-Functional Requirements Prioritization: Empirical Evaluation of IPA, AHP based, HAM-based Approaches". The comparison was done in an experimental set up. In IPA and HAM the basic method appeared to be AHP with a variation of weights application and treating functional and nonfunctional distinctly. In their paper " A systematic literature review of software requirements prioritization Research", Philip Achimugu , etc. [27] have provided review of papers, and papers with comparison studies on requirements Prioritization methods that emphasized the situation of lack of empirical studies and limited utility of comparison in this area.

In "A Machine learning approach to Software requirements prioritization", Perini Anna [24] proposed case based ranking approach to combine stakeholders preferences with approximate preferences computed through machine learning. PersisVoola and A Vinay Babu [26] have proposed Extensive Numerical Assignment method combining Numerical Assignment and Priority Grouping, advocating intervals for taking care of uncertainties. In addition, weights are used for different stakeholders. This method is based on Interval Evidential Reasoning Algorithm according to the authors, with the intervals representing probability distribution. Nupul Kukreja, etc. [21] used the framework Technique of Ordered Preference by Similarity to Ideal Solution (TOPSIS) – a multi criteria decision analysis framework for test cases prioritization as discussed in "Value-Based Requirements Prioritization: Usage Experiences" The value appeared to be based on ranking by stakeholders.

2.5 Literature survey

A total of 90 articles have been studied to understand the nature of research work carried out on requirements prioritization and related areas of software product development. The research works include 70 Research papers; 13 Doctoral Theses, 2 post grad papers, 5 workshop/book chapters of the period - 1996 -2015. Since prioritization is a decision making process, study of decision making literature has formed part of this exploration. Summary of observations from literature study and Research gaps identified from the study are detailed below.

2.5.1 Summary of Observations and Research gaps identified from Literature study

1. There is general agreement on the parameters that influence requirements prioritization for releases in the literature. Cost of Development and Business Value of the feature in their overall sense are used for prioritization across literature except in two methods.

2. Most of the literature covered four methods for Requirements prioritization – AHP, Weigers, Priority Grouping and Cost-Value Method - and combinations of them.

3. Some of the literature has studies on AHP, Cost-Value, Priority grouping, with theoretical comparisons.

4. Saaty's multi level decision making framework , which is applicable to many domains with problem domain specific parameters defined has often formed base for Cost-Value and layered prioritization concept.

5. Methods like Wiegers take into account the risk of not implementing a feature and impacts.

6. Two Tools evolved of the research on requirements prioritization – one being for decision making by Saaty based on AHP and the other being for triaging.

7. Only Four papers have practical studies across projects. Other literature is with sample university projects and not with industry projects. Of the industry related projects, the projects varied widely across domains, nature of development.

8. It is observed form the study that when the methods were compared, the methods did not have a common base line of projects, wherein comparisons could be valid. The nature of projects, level of requirements abstraction varied widely for deducing practical use of comparisons.

9. The methods proposed are complex and do not represent the software development life cycle needs closely. They become tedious to apply and inaccurate to use. They also depend on data being accurate for appropriate prioritization. Recent methods proposed, tried taking cues from other domains, such as fuzzy logic, genetic algorithms, making the process of applying these methods complex.

10. A school of research by Alan Davis encourages simple methods like triaging at multiple levels based on different parameters relevant – such as Cost, Value, Impacts, Time to Market and Resource availability.

11. It is observed that the methods surveyed do not offer the flexibility of re-planning, re-prioritizing in a simple way.

12. Impacts of changes in prioritization and how planning changes under prioritization changes is discussed in one paper. Visualization of planning changes due to prioritization changes during the development cycle is an area under-addressed.

2.5.2 Review and Chronology

Of the 90 literary works covered across requirements prioritization, decision making, release planning, brief summary and gaps of 30 papers are detailed in this section.

Index to Chronological order is presented in the Fig. 2.4

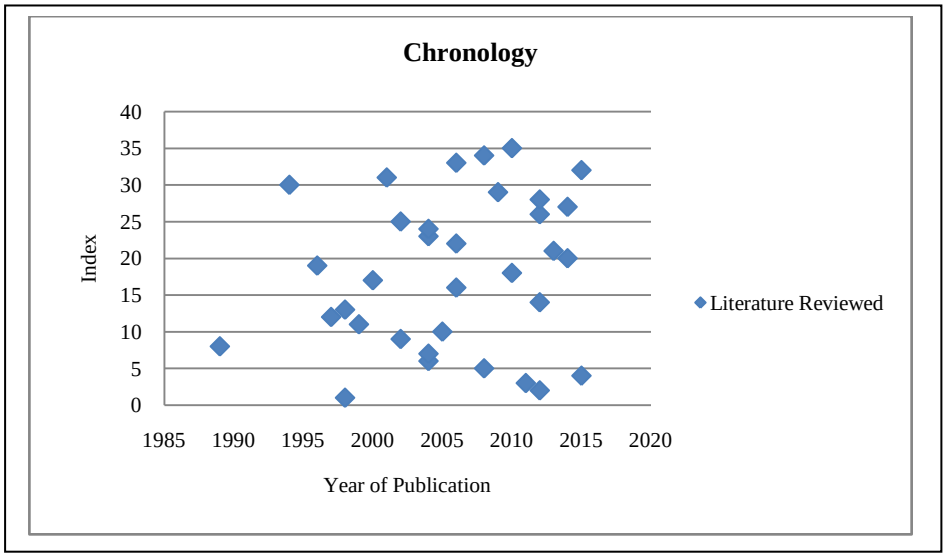

Fig. 2.4. Chronology

Table 2.8. Literature review

No.	**Literature Reviewed**	**Gist of Points gained**
1	Alan M. Davis, Ed Yourdon, Ann S. Zweig. "Requirements Management Made Easy". 39-947, 1998. http://www.omni-vista.com/	Simple triage at multiple levels which is practical and normally used in Industry. Gaps: Scalability, systematic, objective handling.

2	Amir Seyed Danesh and Rodina Ahmad. "Software release planning challenges in software development: An empirical study "African Journal of Business Management Vol. 6(3), pp. 956-970, 25 January, 2012. Available online at http://www. academicjournals.org/AJBM	Empirical study of challenges in release planning for 5 banking projects. Study with 10 participants to identify challenges to release planning. Gaps: The study is limited and identifies generic problem areas for releases of which requirement prioritization is one.
3	Amir Seyed Danesh. " Comparison and Investigation of Re-planning Methods for Software Releases" International Journal of Computer Theory and Engineering, Vol 3, No. 2, April 2011.	2 AHP based re-planning methods are discussed. Gaps: The paper just describes two methods for re-planning
4	Bhagyashri Jawale, Ashish T. Bhole. " Adaptive Fuzzy Hierarchical Cumulative Voting: A Novel Approach Toward Requirement Prioritization", IJRET: International Journal of Research in Engineering and Technology, Vol. 4, Issue 5, May 2015	Combined Cumulative Voting and Fuzzy Logic. Gaps: Applicability in practice needs to be evaluated
5	Carina Alves, Silvia Pereira, George Valenca, etc all. " Preliminary Results from an Empirical Study in Market-Driven Software Companies" Proceeding SAC '08. Proceedings of the 2008 ACM symposium on Applied computing, pp 619-623.	Study of specific factors concerning Requirements Engineering
6	Donald Firesmith. "Prioritizing Requirements", Journal of Object Technology, vol. 3, no. 8, September- October 2004, pp. 35-47. http://www.jot.fm/issues/issue_2004_09/column 4.	Need for prioritization, factors of influence, Different prioritization methods are described. Gaps: Described well standard methods, Evolving trends not included.
7	Dr. Greer, G. Ruhe. "Software Release planning: An evolutionary and iterative approach" Information and Software Technology 46 (2004) 243-253	EVOLVE –incremental method for release planning based on Genetic algorithm taking into consideration stakeholder priority ranking. Gaps: modeled prioritization as optimal set selection from multiple rankings, which is subjective. Does not specify parameters of ranking or constraints suitable. Sample project of 20 requirements is used.

8	Evangelos Triataphyllou, Stuart H. Mann . "An Examination of the Effectiveness of the Multi-Dimensional Decision-Making Methods – A Decision-Making Paradox" Decision Support Systems 5 (1989) 303-312	Four decision making methods – weighted sum, weighted product, AHP, revised AHP are compared indicating contradictions.
9	Frank Moisiadis. "THE FUNDAMENTALS OF PRIORITISING REQUIREMENTS" Systems Engineering, Test & Evaluation Conference, Sydney, Australia, October 2002	When and who should do requirements prioritization is discussed. A framework and a Tool are proposed. Gaps: The framework takes into consideration stakeholders' value and perceptions on importance of requirements, alone and is subjective.
10	G. Ruhe. Software Release Planning. Handbook of Software Engineering and Knowledge Engineering Vol. 3 (S.K. Chang, Ed.), World Scientific 2005, pp 365-394	Release planning with proposed Tool evolve to facilitate what-if scenarios. Gaps: resource capacity, stakeholder value and urgency, dependency constraint are used for quantifying release selection.
11	Karl E Wiegers. "First Things First: Prioritizing Requirements" Software Development Magazine, Vol. 7, No. 9, September 1999, pp. 24-30.	Prioritization based on value, penalty, cost, risk Original contribution. Gaps: Scalability, flexibility problems.
12	Karlsson J, Ryan K. " A cost-value approach for prioritizing requirements" IEEE Software 1997;14(5):67–74.	Cost- Value parameters incorporation in prioritization with AHP. Gaps: taken into account only 2 parameters and used ratio of these for prioritization. Does not talk of other parameters. Less flexible, Problem of scalability.
13	Karlsson J, Wohlin C, Regnell . "An evaluation of methods for prioritizing software requirements", Inform. Software Technol.1998, 39(14-15): 939-947	Three methods evaluation – shows difficulty of standardization. Gaps: has taken high level quality requirements and comparison is subjective.

14	Krzysztof Wnuk. "Visualizing, Analyzing and Managing the Scope of Software Releases in Large-Scale Requirements Engineering" (2012), Dept. of Computer Science, Faculty of Engineering, Lund University. [PhD Thesis]	This PhD thesis studied Large scale requirements Management with Sony Ericsson and suggested linguistic method for consolidating requirements, Provided visualization of scoping dynamics for very large projects. Gaps: While the PhD topic is very relevant, the objectives seem to have spread wide and probably specific to a company . Empirical investigation with 7 participants from one company and 6 participants from 2 other companies to understand challenges in scaling up requirements is done. A survey with 219 participants from 45 companies is done to find how obsolete requirements are managed.
15	L. Karlsson, P. Berander, B. Regnell and C. Wohlin, "Requirements Prioritization: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning", Proceedings 8th Conference on Empirical Assessment in Software Engineering, Edinburgh, UK, 2004.	Comparison of Planning Game and AHP. Gap: Comparison base limited.
16	Laura Lehtola, and Marjo Kauppinen. "Suitability of Requirements Prioritization Methods for Market-driven Software Product Development", Software Process Improvement Practice 2006; 11: 7–19. Published online in Wiley Inter Science (www.interscience.wiley.com). DOI: 10.1002/spip.249	Case study of Two Methods comparison for off the shelf products with two companies. Evaluation of methods used with six companies and 11 participants. Gaps: no common baseline for comparison. Primarily identified the interest of people in prioritization methods.
17	Martin Host, Bjorn Regnell, etc all. "Exploring Bottlenecks in Market-Driven requirements Management Processes with Discrete Event Simulation" Proceedings of PROSIM 2000.	Requirements processing for market-driven software is simulated to study overload situations.
18	MikkoVestola. "A Comparison of Nine basic Techniques for Requirement Prioritization", Seminar in Software Engineering, Aalto University: School of Science, June 2010	Basic description of methods and comparison. Gap: This article briefly discussed 9 methods
19	Ming-June Lee. "Foundations of the Win-Win Requirements Negotiation System" (1996), University of Southern California. [PhD Thesis]	This Thesis introduced formal methods for win-win negotiation method for requirements prioritization and proposed equilibrium model for negotiations.

20	Mohammad Dabbagh, Sai Peck Lee . "An Approach for Integrating the prioritization of Functional and Nonfunctional Requirements", The Scientific World Journal, April 2014, Hindawi Publishing Corporation.	Attempted treating functional and Non functional requirements separately and then integrating them. Gaps: The paper tries to combine the prioritization of functional/non functional requirements process wise, which may not be practical.
21	Nupul Kukreja, Barry Boehm, etc all. "Value-Based Requirements Prioritization: Usage Experiences" Procedia Computer Science 16 (2013) 806-813	Combination of Ranking on multi criteria for test cases selection. Gaps: Focused on test cases prioritization and not on end to end product requirements.
22	P. Berander and P. J¨onsson. "Hierarchical cumulative voting (hcv) prioritization of requirements in hierarchies," *International Journal of Software Engineering & Knowledge Engineering*, vol. 16, pp. 819–849, 2006.	Applying Cumulative Voting Hierarchically to improve on scaling for large requirements. the method tries to group requirements into different levels of abstraction. Gaps: CV/HCV are subjective and are biased by stakeholders.
23	Pablo Trinidad, David Benavides, Antonio Ruiz Cortes. " Improving Decision making in Software Product Line product Management", Decision Support in Software Engineering (ADIS'04), 2004.	Parameters of importance for decision making in product lines
24	Paolo Avesani, Cinzia Bazzanella, Anna Perini, Angelo Susi. "Supporting the Requirements Prioritization Process. A Machine learning Approach" , Proc. of 16th International Conference on Software Engineering and Knowledge engineering (SEKE 2004), June 2004, Banff, Alberta (CAN).	Case based ranking is proposed and ranking approximations through machine learning combining preferences of stakeholders. Gaps: The method claims to take care of scalability compared to AHP. Considers stake holders preferences and pair wise comparisons. Complex.
25	Par Carlshamre. "Release Planning in Market-Driven Software Product Development: Provoking an Understanding" Requirements Engineering (2002) 7:139-151. Springer-Verlag	Proposed a Tool for release planning based on maximizing value of release under the resources constraint and interdependencies. Gaps: Selection of requirements set for a release is encapsulated in to value and cost .
26	PersisVoola, Vinaya Babu. "Requirements Uncertainty Prioritization Approach: A Novel Approach for Requirements Prioritization" Software Engineering: An International Journal (SEIJ), Vol. 2, No. 2, Sept 2012.	Proposed RUPA -Requirements Uncertainty Prioritization Approach Combining numerical assignment, priority grouping and proposed intervals for priority. Gaps: Approach takes into account stakeholders' rankings on importance for decision making. Sample project is used. Different attributes are not discussed.

27	Philip Achimugu , Ali Selamat, Roliana Ibrahim, Mohd Naz'ri Mahrin[2014]. " A systematic literature review of software requirements prioritization Research". Information and Software Technology 56 (2014) 568–585 journal homepage:www.elsevier.com/locate/infsof	73 Primary studies -13 journal articles, 35 conference papers and 8 workshop papers considered. Conclusion drawn is Existing prioritization techniques suffer from a number of limitations - lack of scalability, methods of dealing with rank updates during requirements evolution, coordination among stakeholders and requirements dependency issues and the applicability of existing techniques in complex and real setting has not been reported yet. Gaps: The study is based on search of keywords rather than on practical aspects of software development.
28	Qi Li. "Value-Based, Dependency-Aware Inspection and Test Prioritization", (2012), University of Southern California. [PhD Thesis]	This PhD Thesis introduced Value based, dependency aware test cases prioritization strategy for improving life-cycle cost awareness. Measures for "How much testing" – ROI, "How Quickly" – Risk Reduction Leverage are introduced. Gaps: Test cases prioritization is considered in isolation. Graduate projects are used for evaluating results, other than 2 industrial projects.
29	Quiao Ma. "The effectiveness of Requirements prioritization Techniques for a Medium to large number of Requirements: A Systematic Literature Review". Auckland University of Technology 2009. Post graduate thesis. `	Literature review describing basic terminology. Indicates to lack of evidence on methods for large number of requirements. Methods for medium number of requirements have subjective evaluation based on user perceptions. Gaps: 5 methods in literature are considered and comparison is very basic and does not address comparison on practical aspects.
30	Raymond Joseph Madachy. 'A Software Project Dynamics Model for Process Cost, Schedule and Risk Assessment", (1994), University of Southern California. [PhD Thesis]	This Thesis investigated effects of changes in process on cost, schedule and risk and introduced a dynamic model of inspection based on heuristics for cost estimation and risk assessment. Gaps: Addresses overall process and considers cost estimation, schedule risk mainly.
31	Regnell, B., H¨ost, M., Nattoch Dag, J., Beremark, P., Hjelm, T. " An Industrial Case Study on Distributed Prioritization in Market-Driven Requirements Engineering for Packaged Software", Requirements Engineering 2001, vol 6, no 1, pp 51-62	Case study approach and Industry study more on global requirements collection and sorting. This case study had 8 participants, 18 questions at a broad level of prioritization across market segments. Gaps: Limited study at a broad level.

32	Ruby, Dr. Balakishan. " Role of Fuzzy Logic in Requirement Prioritization", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 4, Issue 6, June 2015	Proposed Fuzzy logic to cater for uncertainties in priorities. Gap: Applicability in practice needs to be evaluated.
33	Thamer AlBourae, G Ruhe, Mahmood Moussavi. "Lightweight Replanning of Software Product Releases" International workshop on Software Product Management (IWSPM'06) 2006 IEEE.	AHP based re-planning method is discussed. Gaps: Selection of requirements for release re-planning is based on AHP and importance of requirements is taken as the attribute. Every time new features need to be added during release, evaluation of new features in comparison with existing features is done taking effort and time as attributes.
34	Thomas L Saaty. "Decision making with analytic hierarchy process" Int. J. Services Sciences, Vol. 1, No.1,2008	Decision making process with AHP Original Contribution Gap: This is a general multi level decision making framework, that needs to be adapted for Software development.
35	Zornitza Racheva, etc. all. " Do we know enough about Requirements in Agile Projects: Insights from a Case Study" , Conference: Requirements Engineering , IEEE International Conference -RE, pp. 147-156, 2010	Survey, Interviews with 11 practitioners, 10 projects, 8 companies on few parameters. Used grounded theory building method – Exploratory case study. Gaps: Limited study. Analyzed only use of value, - ve value, who gets involved in prioritization.

2.6 Summary

Requirements prioritization is treated as an independent problem to be solved in the literature rather than as an integral part of the larger software product development across versions of releases with product life stage considerations and practical aspects of the software development. While release planning is considered in literature, prioritization falls back onto standard methods of AHP, Cost-Value. Cost and Value emerge as the favorite parameters used for prioritization from the literature, while other parameters like Resource capacity, interdependencies, time to market are discussed in theory.

Various methods from literature are examined in this chapter. Methods that have been used often and have original contribution are described in detail and will be used for comparison with the method proposed in this research. Methods of recent times, which are mostly combination of other methods and adapt principles from other streams, such as fuzzy logic and genetic algorithms, are presented, briefly.

Evolving methods adapting requirements prioritization as a continuous activity with parameters of relevance during the software development cycle requires practical knowledge applied to research. Flexibility of incorporating changes in prioritization, visualizing impacts

of changes on planning, understanding effects on quality and delivery of the features are yet to evolve in research.

The present research attempts to bridge this gap, combining industrial experience with research study and analysis. A new framework is proposed with the understanding gained form literature and knowledge assimilated from the study. Value parameter, used across the literature studied, encapsulating business value is considered in this research. Cost , considered in literature, is taken as effort needed to develop the software, taking into account existence of already developed modules. Constraints of time needed to develop, interdependencies, covered in theory in literature are taken in to account. Resource availability discussed in one paper is also considered as a parameter for the proposed prioritization framework. Based on the need for multiple parameters to be considered for prioritization, a multi-level framework is proposed in this research. Objectives of this research and methodology are discussed in the next chapter.

Chapter 3

3. Objectives of Research and Methodology

3.1 Introduction - Relevance of the Topic

Requirements prioritization is the most significant part of software development to enable right products/solutions development in right time with optimal resources. Literature study points to simple to complex methods for requirements prioritization being researched and gap between research and practice due to difficulties in implementing the research methods.

Literature survey indicated to many methods being discussed. Simple methods are used in practice, though insufficient. Many complex methods are discussed in literature, but rarely applied in practice. Evaluation of different methods, though available to some extent in literature, is not on a standardized base set of requirements and not applicable to requirements in general. There is also lack of appropriate bench marking related to factors influencing the prioritization and the relative order of influence of the factors based on nature of product domains. The methods target one time prioritization with accurate information not being available. Scaling is an issue with these methods, in that the number of calculations and comparisons becomes cumbersome. Re-planning under changes requires rework.

Requirements prioritization remains an area handled in ad hoc manner and methods being used for approximate prioritization serving limited purpose. The methods used do not consider all variables in a systematic way. While a combined importance of requirements is normally used to encompass all parameters implicitly, for prioritization most of the times, cost –value ratio or risk weighted cost-value ratio is also used sometimes. These methods do not provide flexibility in prioritization under changes or retain the information as to what are the considerations in prioritization process. Some of the methods can be applied recursively taking each parameter in to consideration, successively. Some methods can be applied hierarchically to prioritize requirements at different detailing levels.

There is agreement on the parameters of relevance for requirements prioritization, though there does not seem to be a detailed study on these parameters and the order in which their importance is relevant in literature. While there has been recent work combining two or three simple methods or suggesting fuzzy logic and genetic algorithms for requirements prioritization to address uncertainty in prioritization, there have not been efforts to study and analyze what happens in practice and what parameters matter and in what order. In addition, methods that are simple, flexible yet comprehensive taking in to account the parameters of relevance are yet to emerge for requirements prioritization.

3.2 Problem Statement/Research Focus

A simple and standardized framework that can be modified for different domains and used for comparison for the purpose of Requirements Prioritization in practice for successive releases of Software Products has not yet materialized to bridge the gap between research and practice in software industry.

The research work in this paper addresses this need precisely. What parameters are of relevance for requirements prioritization for software development across organizations engaged in different software development activities – some developing products, some product based solutions, some developing solutions as products – is studied as part of the research.

This research work attempts to bridge this gap, by proposing a suitable framework from industry experience and studying factors that influence prioritization of requirements

The research focuses on study and understanding of factors influencing requirements prioritization and defining a Framework, that leads to simple requirements prioritization for successive releases of software products and offers scalability, flexibility, visibility and traceability across the development life cycle leading to improved quality and release planning.

The proposed Framework leads to simple requirements prioritization for successive releases of software products and offers scalability, flexibility, visibility and traceability across the development life cycle leading to improved quality and release planning.

The study is accomplished through gathered data with a questionnaire designed for eliciting information related to prioritization of requirements. Principal objectives of the research are to study the parameters influencing requirements prioritization and proposing a simple, scalable, flexible and practical method for prioritizing the requirements. Objectives of the research are presented in Section 3.3. The method is expected to offer advantages across the product development life cycle.

The proposed method is compared with 4 of the methods (AHP, Wiegers, Priority groups, Cost-Value approach) from literature, as part of the research. In addition, a comparison across three data sets grouped from the data gathered, based on size of organization is taken up.

Methodology followed for achieving the objectives of the research is elaborated in section 3.4.

3.3 Objectives

Organizations use simple methods such as ranking, priority grouping, which do not provide systematic, flexible, scalable methodology for Requirements prioritization in software development in practice. Uncertainties, changes in scope of requirements, multiple parameters to be considered lead to ad hoc handling of requirements prioritization. Systematic methods proposed in research are complex for practice usage and have not come into practice. In order to address this gap between research and practice, and to address the gap of availability of systematic, simple and easy methods taking into consideration multiple parameters, the following Objectives have been chosen for the research.

Objective 1: To study the factors that influence requirements prioritization and elicit information on order of preference of using these factors.

Objective 2: To compare and analyze data for large, medium and small software organizations.

Objective 3: To propose a new framework – ABC Framework to enable simple and effective methodology for Requirements Prioritization for successive releases.

Objective 4: To Formulate Mathematical models for practical usage of proposed Framework

Objective 5: To Compare the proposed framework with four significant Requirements Prioritization methods.

3.4 Research Methodology

The Objectives set for research are attempted to be achieved through steps as depicted in Fig. 3.1.

Due to the confidentiality and IPR issues, involved in obtaining relevant information on the topic, it is difficult to collect data from a statistically significant sample of participants. In view of this challenge, data was collected from a broad cross section of professionals, that are actively involved in Requirements Prioritization decisions, using a structured questionnaire, primarily through one-to-one interactions. It was followed by discussions with experienced professionals.

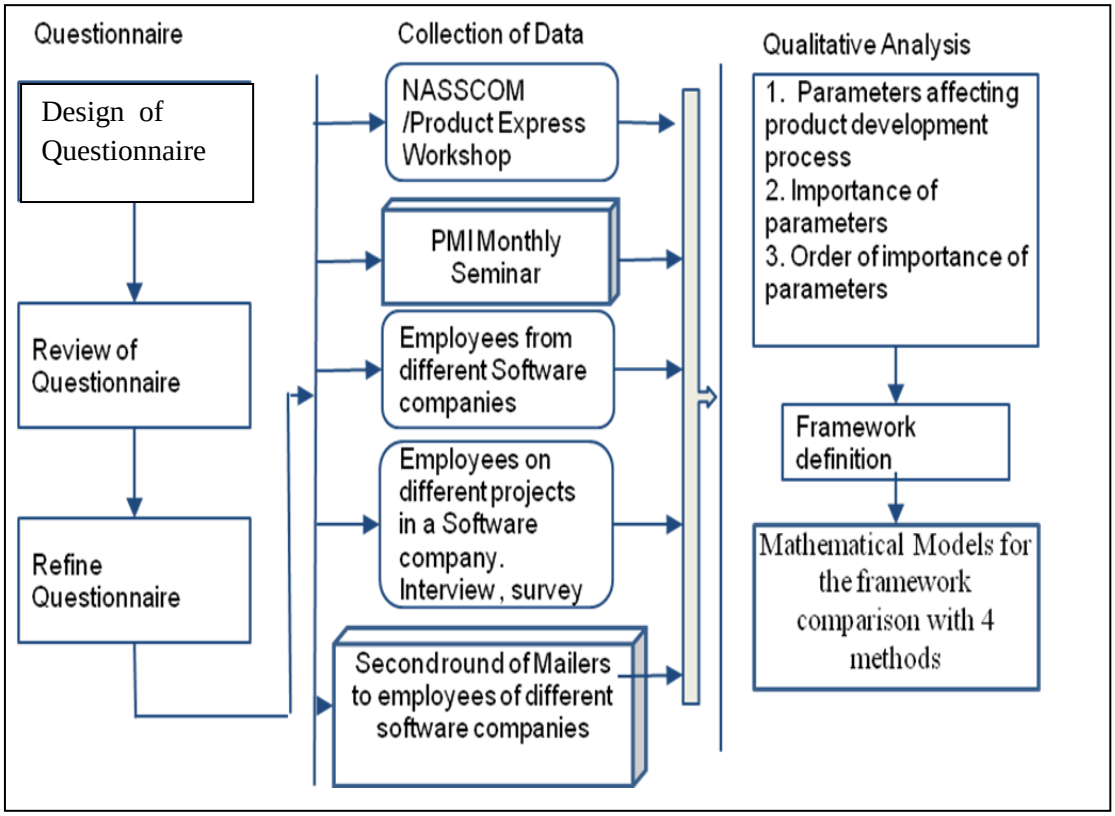

Fig. 3.1. Research Methodology

3.4.1 Questionnaire

The parameters influencing requirements prioritization is well understood from the literature. Working towards **Objective1**, in order to understand the current practices related to software product development, processes followed at different organizations, developing products and solutions for different domains and platforms, a study with a questionnaire is designed. Study has been conducted with targeted organizations and practitioners. The questionnaire has been visualized to be exhaustive and comprehensive to address different stages of product development. The questionnaire has been designed based on the scholar's own experience with multiple products' development companies and with the understanding from the literature study. A review of the questionnaire is taken up with the industry experts in product development. Keeping in mind the data collection ethics, name and organization were not requested in the questionnaire initially. Both these details have been requested for subsequent study based on the feedback received.

The study has been structured to cover breadth of software development organizations to include products/applications in different domains, at different stages of life cycle, following different processes, of varying complexities. The study gathered information on parameters like products domain, maturity of the products, development process variations and requirements handling modes. The objective of the study has been to identify practices related to requirements' prioritization among software development organizations and to understand association of requirements prioritization' affects on software deliveries and resources. The study has been conducted to understand the effectiveness of the current processes and to identify requirements prioritization needs for enabling planned deliverables with reduced uncertainties.

The questionnaire is detailed in the next chapter. The questionnaire has been designed as an on-line form for the participants to respond. Questionnaire has been administered through three channels for information gathering - sending across the questionnaire through email, through discussions and on-line form completion, through workshops conducted.

3.4.2 Study Methodology

Software product development organizations being highly protective about the products information and processes followed, gathering information has posed great difficulty and took around 2 years. While the questionnaire does not seek any confidential information, information gathering from software companies has indeed been a difficult exercise. Around 500 + targeted participants from identified companies have been approached through multiple channels. The questionnaire is administered to

1. NASSCOM/Product Express workshop participants,

2. Employees in Software industry in India, (with few exceptions where participants are from other countries).

3. A Products development Services Company in Pune. Discussion and then on-line form completion.

4. Product Management Institute, Pune chapter monthly seminar participants

106 participants have responded from organizations representing a cross section of companies with varied software products, solutions and domains. The data enabled valid analysis, confirming to the scholar's own experience. Study is described further in the next chapter. The data is gathered over a period of 2 years and the data gathered enabled grouping of responses into three data sets based on the size of the organization. This enabled study and comparison of parameters influencing requirements prioritization as the size of organization and there by nature of software varies. Further details are described in the next chapter.

3.4.3 Analysis

Research on requirements prioritization with empirical studies depended on qualitative analysis for deriving understanding on how the requirements are handled in practice. The same has been adapted in this research. Qualitative analysis based on responses has been taken up to study importance of various factors on requirements' prioritization in software product releases. Analysis has been based on grouping by Organization/Project Size, Product Maturity, and Product Type. This information is elicited from Section I of the questionnaire. The processes followed, difficulties faced with current processes have been analyzed from section II responses. Prioritization methods used, parameters considered is taken from the responses of section III. Data is analyzed to bring out the factors of relevance for prioritization of requirements and also the order of factors of relevance. Analysis has also been carried out across three data sets formed based on the size of the organization. Comparison of responses across the data sets gave view of consistency and variations in responses across these data sets, which enabled strengthening the analysis, forming part of Objective 2. The results of the 3 data sets comparison have been presented in Chapter 7.

3.4.4 Framework

Working towards **Objective3**, with the understanding of the factors and constraints from the study and with the intuitive understanding of the software development practices, a framework has been designed to help prioritize requirements in an easy way. The framework and the considerations taken into account are described in chapter 8.

Three methods for utilizing the framework are proposed extending the objectives. The framework and its benefits have been compared with four other methods. The comparison is presented in Chapter 10.

3.5 Summary

Gaps in the present research –requirements prioritization methods' abilities for simple, flexible, scalable, multi level, amenable for planning visualization - are discussed in this chapter. Research Problem space of research to find a simple, easy, adaptable in practice method for requirements prioritization is detailed. Research focus areas and objectives of the research are identified from the problem space. Research methodology, step wise process, and outline of research work planned are explained in this chapter. Study methodology, process is further elaborated in the next chapter.

Chapter 4

4. Study –Methodology, Participants

4.1 Introduction

The Study questionnaire has been structured around parameters like products domain, maturity of the products, development process variations and requirements handling modes. The objective of the study has been to identify practices related to requirements' prioritization among software development organizations and to understand association of requirements prioritization effects on software deliveries and resources. The study has been conducted to understand the effectiveness of the current processes and to identify requirements prioritization needs for enabling planned deliverables with reduced uncertainties.

The questionnaire was divided in to 3 sections addressing three data collection purposes – section 1 – soliciting information about the nature of the organization, section II – eliciting information about the processes followed for product development, problems faced, alternate solutions used, while section III seeks information on the core of the research – requirements prioritization – how the requirements are handled, prioritized, parameters of importance issues with current methods.

Section I and II enabled grouping of the responses for analysis. Section III provided information on current practices for requirements prioritization and problems associated with current practices.

Participants for questionnaire have been identified from across software industry working on products development and solutions development. Participants came from across companies, across different projects.

Questionnaire has been administered through three channels for Data Collection. Through email, sending across the questionnaire, through discussions and on-line form completion, through workshops conducted.

4.2 The Questionnaire

The questionnaire administered to participants is presented in this section. The questionnaire has 3 sections. Section I has 12 questions. This section mainly gathers information on nature of organization, type of work, activities, participants are involved in. This information is important in grouping responses and understanding depth and breadth of software product development area where requirements' prioritization is required to be studied. Section II has 10 questions enquiring about the processes followed for software development in the participants' organizations. Also information about problem areas and

solutions adapted is solicited in this section. 3 of the questions of this section are descriptive, to enable participants to describe their perceptions. Section III focused on requirements prioritization methods used in the organizations, preference for any methods, problems associated, solutions that could have helped. This section has 20 questions. The questionnaire is presented below.

Study Questionnaire

Requirements Prioritization for Successive Releases of S/W Products

The Questionnaire attempts at understanding your current methods of Requirements Prioritization and Positive aspects and pain points of the process. Section I sets the background with general information about your organization, Products. Section I has about 12 Multiple choice questions Section II queries on Current Processes followed with 10 questions, 3 of which are open ended. and Section III quizzes on Requirements prioritization for successive Release with 20 Multiple choice questions.

Name

Organization

Section I

About Your Organization

1. The size of your organization

    - Small (<25 employees)
    - Medium sized (25 to 200)
    - Large (>200 employees)

2. The Organization has

    - Single product
    - Single Product line (multiple products with single platform)
    - Multiple product lines
    - No Product development

3. What role do you play in your organization?

    - Product owner
    - Project Manager
    - Project Lead
    - Other:

About You

4. You Participate in (for upcoming release)

    - Product Planning
    - Feature evaluation

- Business Value assessment
- Product Development

5. You get involved in Requirements

- Analysis
- Estimation
- Prioritization
- Implementation

About Your Products

6. You work on Products that are in Market for

- < 2 years
- 2 to 5 years
- 5 to 10 years
- >10 years

7. You work on Products in the field of

- CAD/Manufacturing
- CRM/Banking
- Operating Systems/Development Tools
- Other:

8. Your organization releases products' next versions every

- 4 weeks
- Quarterly
- Half yearly
- Yearly
- Other:

9. The development process Your Organization follows

- Waterfall
- Iterative
- Agile
- Other:

10. The products are

- Single tier
- 2 tier
- 3 tier
- n tier

11. The products are used in

- One Country
- One Continent
- All Continents

12. The products can be used on

- Desk tops
- Web
- Mobile
- Cloud

Section II

About Your Current Process

1. How do you choose features/requirements to be implemented for next release?

2. What are the problem areas you see it your current process of feature selection for upcoming release?

3. How do you circumvent the problems with your current process of feature selection ?

4. How often do you have teams working for release under pressure and for long hours in a day?

- Very Often
- Often
- Sometimes
- Rarely

5. Do you have few of the team members over worked during releases?

- Very Often
- Often
- Sometimes
- Rarely

6. How often do you abandon features being implemented for a release and restart on new features?

- Very Often
- Often
- Sometimes
- Rarely

7. Do you feel the right resources availability is an issue for meeting release schedules?

- Very Often
- Often
- Sometimes
- Rarely

8. How often do you abandon features during release due to realized impacts on existing customers?

- Very Often
- Often
- Sometimes
- Rarely

9. Do you analyze the impacts on core structure /architecture/data model, of features to be implemented a priori?

- Very Often
- Often

- Sometimes
- Rarely

10. How often you rework your resource(time, personnel, S/W,H/W)estimates for the features during the development cycle for a release?

- Very Often
- Often
- Sometimes
- Rarely

Section III

About Product Requirements Prioritization

1. Your organization collects requirements through

- Marketing team
- Executive Direction
- Development Team
- Customer Change Requests
- All of the above

2. Requirements Analysis/assessment is done by

- Planning team
- Stake Holders team discussions
- Development team
- Other:

3. Set of requirements for next/successive release is planned by

- Ranking by Value proposition
- Resources availability
- Time availability
- All of the above

4. Requirements evaluating/ prioritizing is done by

- Using a Framework
- Product team discussions
- Comes from Executive Management
- Other:

5. Changes in requirements during the release are managed by

- Reprioritization of complete set for the release
- Removal/Addition of some requirements with no other changes
- Extending the Release date
- Other:

6. The following parameters are considered for requirements prioritization (tick all relevant)

- Business Value (BV)
- Availability of Resources (AR)
- Time to Market (TM)
- Difficulty of Implementation (DI)
- Impact on existing Customers (IC)
- Other:

7. Order of parameters considered for requirements prioritization

- BV, DI, AR, TM, IC
- BV, TM, IC, AR, DI
- TM, DI, IC, AR, BV
- AR, TM, IC, BV, DI
- DI, BV, AR, TM, IC
- Other:

8. Weights are associated with parameters considered for prioritization

- Most Often
- Often
- Not Often
- No Weights

9. A multi stage prioritization scheme is useful for requirements prioritization

- Always
- Most often
- Not often
- Never

10. Working out prioritization exactly for each requirement for product releases

- Most useful
- Often useful
- Not useful often
- Never useful

11. Change in prioritization during release scheme necessitates(with scheme in Q.10 above)

- Complete rework of prioritization
- Minor changes to existing list
- Release date extension
- No Change

12. Classifying requirements in to " 1. Must have 2. Good to have 3. Can live without " groups for product release is

- Sufficient Always
- Sufficient often
- Not sufficient often
- Not sufficient at all

13. Prioritizing requirements using Analytical Hierarchy Process (AHP) for product release is

- Accurate
- Simple
- Complex
- Time taking
- Not used AHP

14. When number of requirements to be handled is large (>20), AHP is

- Accurate
- Simple
- Complex
- Time taking
- Not used AHP

15. It is essential to know how much important each requirement is when compared to other for prioritization

- Always
- Most Often
- Not Always
- Never

16. It is sufficient to know relative importance of requirements for prioritization rather than "how much more important"

- Always
- Most Often
- Not Always
- Never

17. Cost – Value ratio for requirements is the best indicator of priority

- Always
- Most Often
- Not Always
- Never

18. Ranking of requirements(in sequence of priority) based on a parameter is sufficient for prioritization

- Always
- Most Often
- Not Always
- Never

19. Numerical assignment of priority (grouping by assigning priority 1,2,3,..) to requirements is sufficient

- Always
- Most Often
- Not Always
- Never

20. Requirements Prioritization provides traceability along the Product life cycle for improved Quality of the Product.

- Always
- Most Often
- Not Always
- Never

4.3 Study Methodology

The questionnaire presented in section 2 has been created in an online form and link to the form was sent across to participants through mail. For some participants questionnaire was sent as a document. Some of the participants were from a workshop conducted by Product Express and some of the participants were from a PMI seminar. These participants were provided with hard copies of questionnaire. Around 23 participants were from an organization which develops products for many clients in multiple domains. Interview with each domain product team and the gathering information online has been followed with this organization. This particular data also helped as a case study. Different means such as

workshops/seminars, Interviews, form completion online, emails as listed below were utilized for obtaining responses.

1. NASSCOM/Product Express workshop participants,

2. Employees in Software industry in India, (with few exceptions where participants are from other countries).

3. A Products development Services Company in Pune. Interview and then on-line form survey

4. Product Management Institute, Pune chapter monthly seminar participants

The methodology is described in Fig. 4.1.

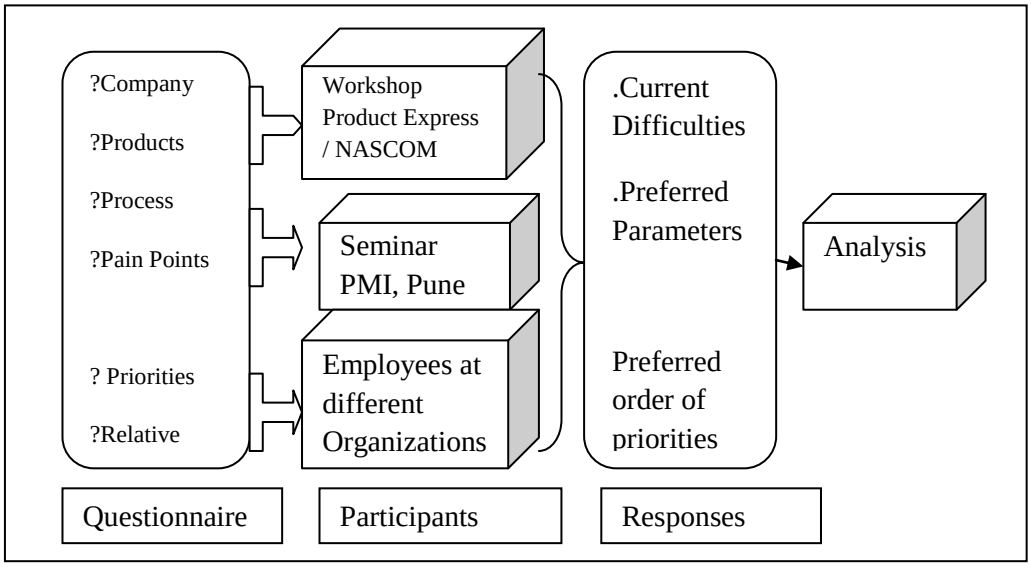

Fig. 4.1. Study Methodology

While the questionnaire has been sent to 500+participants, around 106 responses have been collected. The data represented a cross section of companies and domains and enabled valid analysis, confirming to the scholar's own experience. Responses through different means are elaborated below. Data collection channels are detailed in Table 4.1/Fig.4.2.

Table 4.1. Data Collection Channels

Responses through	**Number of Participants**
email	3
online form filling -multiple companies	57
Product services company - Interview before on-line form filling	21
NASSCOM Workshop/PMI Seminar	25
Total	106

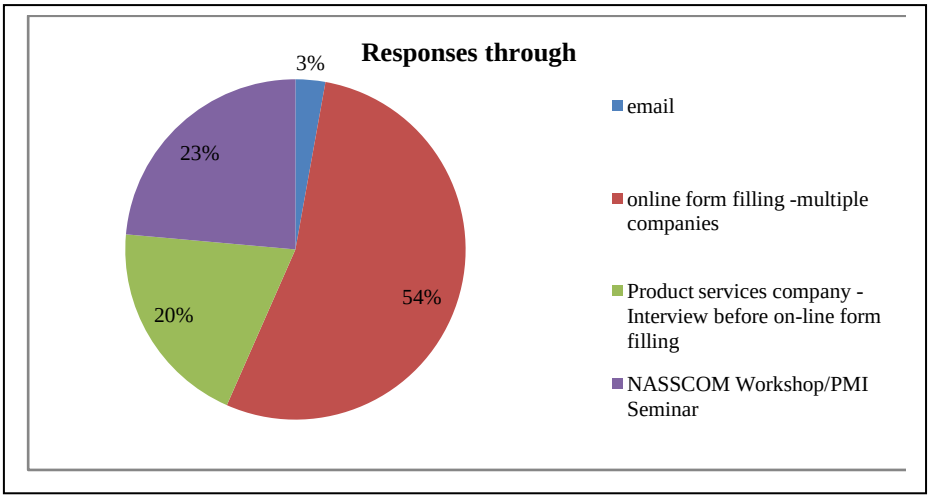

Fig. 4.2. Data collection Channels

Qualitative analysis based on responses has been taken up to study importance of prioritization on product releases. Data is analyzed for over all parameters of influence on requirements across differently characterized software development - domains, maturity of software, complexity of software, release cycles. Grouping based on Organization Size was done and analysis of parameters of influence across these groups on requirements was taken up. This information is elicited from Section I of the questionnaire.

Data is gathered in three stages totaling to 106 participants across 61 organizations over a period of 2 years. In the first stage, responses from 53 participants from 21 organizations are gathered. In the second stage data grew to 65 participants from 40 organizations. By third stage the data expanded to 106 participants from 61 organizations. The first stage has predominantly small companies, whereas by third stage, large companies' proportion increased significantly.

4.4 About The data

The questionnaire has been divided into three parts. The first part elicited data related to the domains of the project, nature of the project, the role played by the respondent, the stage the product is in, and release cycle durations with 12 questions.

The second part focused on the process followed for development and gathered information on what process is followed for development, how the requirements are collected and analyzed, problem areas like over work or overruns on time with 10 questions. Part III focused on how the requirements are handled across the projects and has 20 questions, covering collection of requirements, prioritization methods used, areas of problems and current solutions adopted.

The participants that participated in the study range from organizations that are long term, enterprise products players to relatively new and single product/custom software players. The domains are related to engineering fields to commerce applications to gaming solutions. Some of the products have been under continuous enhancement and maintenance for years.

Different processes - Waterfall, Iterative, Agile are followed across the organizations. The products developed are typically used by large customer base of the clients for specific applications on different platforms and devices. Products undergo modifications to meet further requirements of the clients, often changing requirements as the development progresses. Providing the customers with ever enhancing products is made possible by successive releases of products at varied intervals, ranging from few weeks to few months to 1 or 2 years.

The fundamental questions that are being explored in this research - What is to be made available in the next release? How to manage the requirements under expanding client needs, cost and time implications? Will prioritization of requirements and planning releases help streamline the project deliveries to client's satisfaction without overworking the teams or missing time to market deadlines? – The responses for these questions have been captured in section 3. In general, a simple ranking or grouping is used for the requirements prioritization. Typically working under pressure for completion of changing requirements is the norm during product development across the participants' organizations. The following sections analyze the responses in detail.

4.5 Participants and Organizations – Data from Section I

The organizations of participants varied from large(> 200 employees) to medium (25 to 200 employees) sized to small(<25 employees), with local and global presence of the products. Some of the organizations have multiple product lines, while some have single product lines. The responses have been gathered from 106 participants belonging to 61 organizations. The participants are involved in business analysis, project management and product development.

The data has been collected from participants majorly from Large and Medium organizations, 14% of participants being from Small organizations. Size of organizations is presented in Table 4.2/Fig. 4.3.

Table 4.2. Size of organizations

Size of the Organization	**Number of participants**
Large >200	51
Medium (25 to 200)	40
Small (<25)	15
Total	106

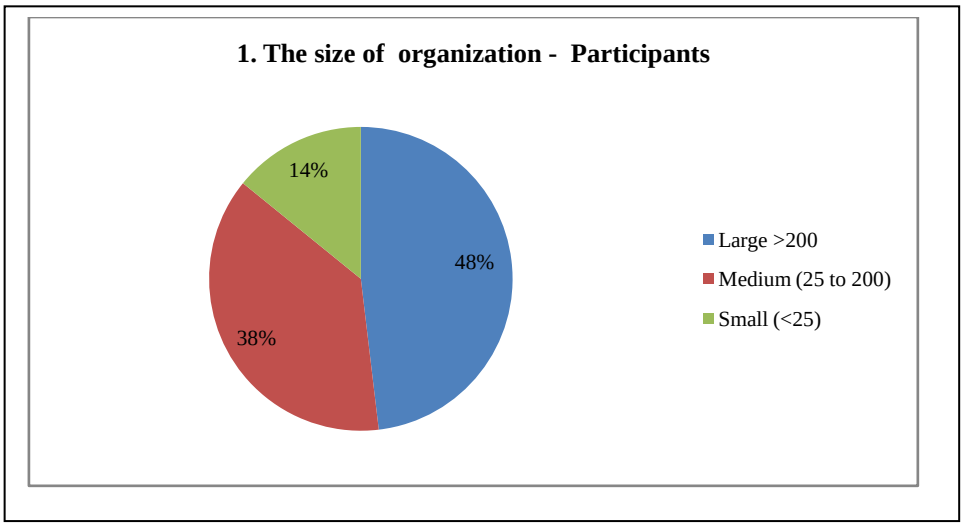

Fig. 4.3. Size of organizations

Large companies have many products of different product lines or a single product line, whereas small companies mainly work on either single product line or single product. The complexity of the products increases as the number of products increases or as the product lines increase. Complexity of the products influences the requirements prioritization in terms of impacts on existing customers, existing core of the product lines. At the same time existing systems enable easy implementation of the requirements on top of the core systems. Due to this reason, complexity of the products is incorporated in the data collected to understand the nature of the products. As is evident from the collected data presented in Table 4.3, 4.4, 4.5/ Fig. 4.4, 4.5, 4.6,4.7.

Large companies worked on multiple product lines indicating the complexity level of the products handled by them. Majority of the participants worked on complex systems as is evident from the data presented.

Table 4.3. Complexity of products–number of product lines

Size of Org.	No. of participants/ Factor	multiple product lines	single product line	No product development
Large >200	51	42	6	3
Medium (25 to 200)	40	22	14	4
Small (<25)	15	4	10	1
Total	106	68	30	8

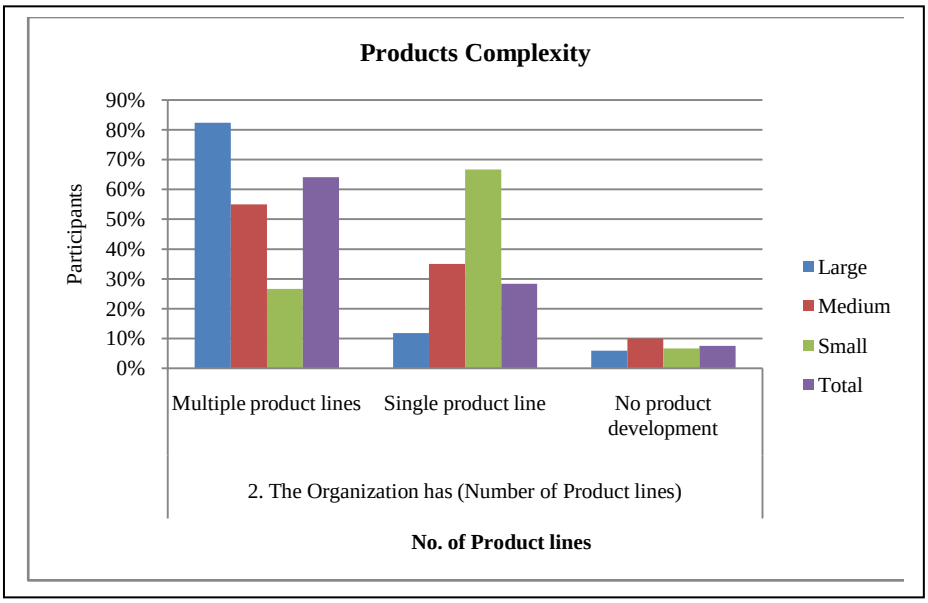

Fig. 4.4. Complexity of Products – number of product lines

Table 4.4. Development Complexity

Size of Org	**Participants working with Products Complexity**		
	3 tier/n tier	2 tier	single tier
Large	21	20	6
Medium	19	16	4
Small	8	2	4
Total	**48**	**38**	**14**

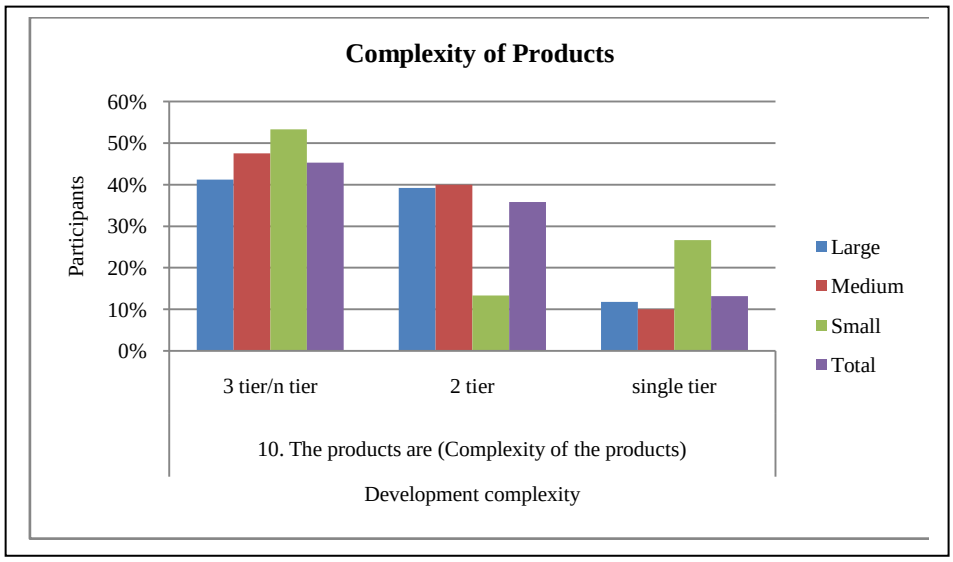

Fig. 4.5.Development Complexity

Table 4.5. Data Collection Details - Usage of products, devices

Spread of Usage – no. of participants				**Devices & Platforms**		
Size of Org.	All continents	one continent	one country	Desktop /mobile/ web/cloud	desktop /web/ mobile	desktop
Large	32	5	9	17	20	4
Medium	20	7	13	3	34	3
Small	6	1	7	3	10	2
Total	**58**	**13**	**29**	**23**	**64**	**9**

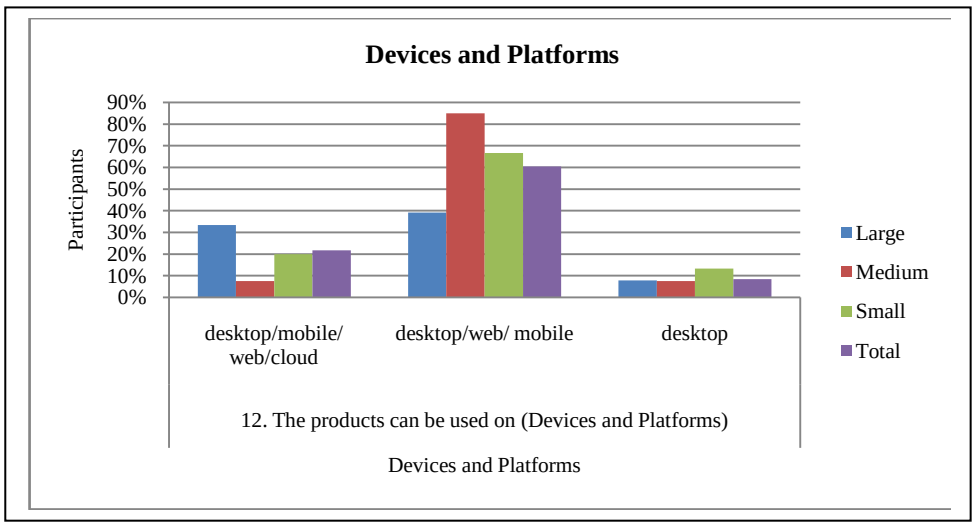

Fig. 4.6. Complexity of products – Devices and platforms

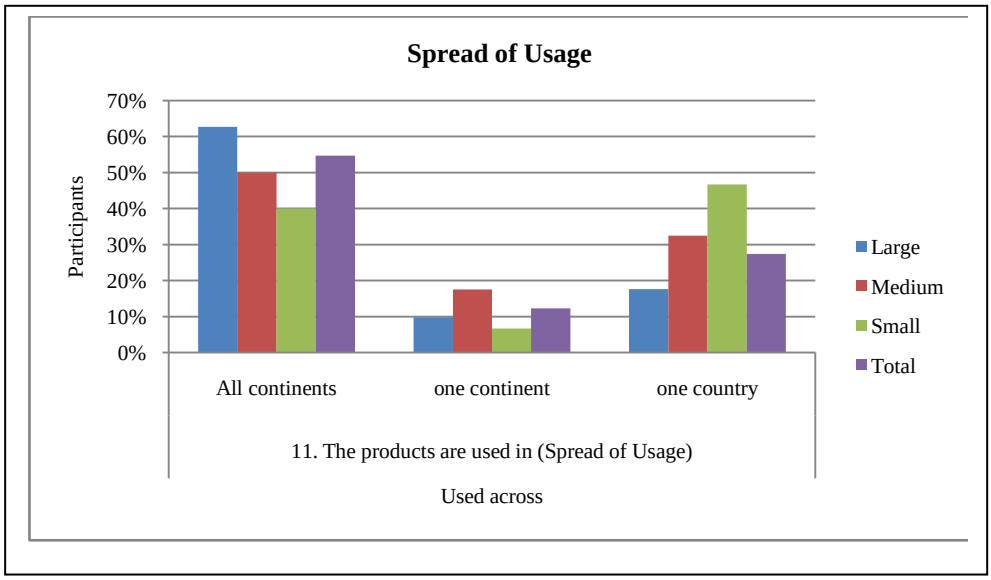

Fig. 4.7. Complexity of products – Spread of usage

Software solutions being developed for Engineering domains require high accuracy, ability to handle large data and stability. They require to be consistent in their results imposing requirements clarity to a great extent. Real time applications such as in Telecom domain, Healthcare require accuracy and speed of response. Consumer applications may not require much of accuracy but would require speed of response. Hence information on the domains of the products, the participants work upon is gathered to understand spread of participants across the domains. The spread of domains is presented in Table 4.6/Fig.4.8. Large companies work involved more of engineering domains where as medium companies' work involved more of consumer applications and involving new Technologies. Responses are distributed evenly across Engineering and Consumer Domains.

Table 4.6. Domains

Size	Engineering Software - CAD/GIS/Telecom/transport/System	Ecommerce/CRM	Web, mobile Technologies
Large	37	9	2
Medium	13	11	16
Small	5	6	4
Total	55	26	22

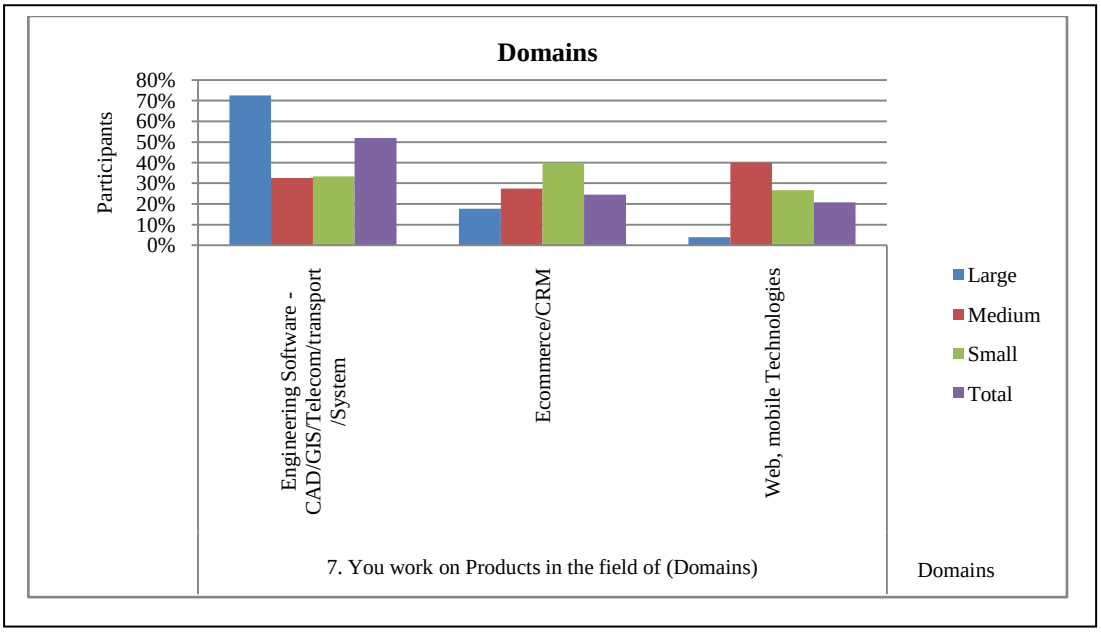

Fig. 4.8. Domains

As the Domains influence how the requirements are handled for the products being developed, The criticality of the products also influences the release cycles at which the products can be released. Consumer applications can enable feature by feature release in shorter durations, where as Engineering applications are released considerably with longer release cycles. The spread of release cycles from a year + to 4 weeks or less is evident from the data gathered representing spread of data across domains, complexity, durations of releases. Release cycles of Large, Medium, small organizations is presented in Table 4.7/ Fig. 4.9.

Table 4.7. Release Cycles

Release Cycles			
Size of org.	yearly, > yearly	half yearly, quarterly	4 weeks
Large	23	17	9
Medium	7	14	20
Small	3	7	6
Total	**33**	**38**	**35**

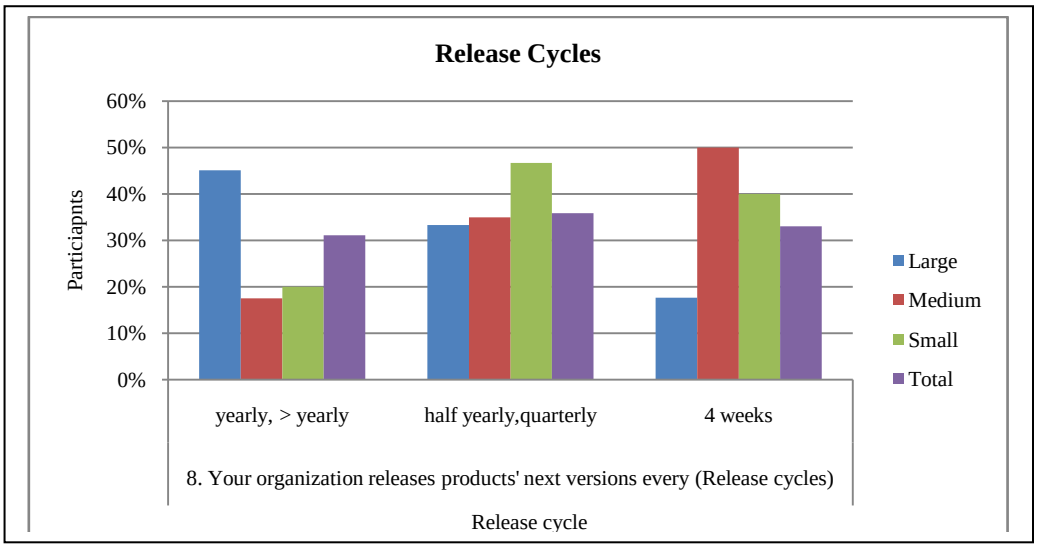

Fig. 4.9. Release Cycles

Table 4.8 below indicates the 4 weeks release cycle predominance with medium size companies working predominantly on consumer applications in Ecommerce, web mobile technologies. Large companies show

the preference for yearly or half yearly release cycle. The responses are evenly spread across engineering applications and consumer applications. And the release cycles vary across year+ to 4 weeks uniformly.

Table 4.8. Data collection details - Domains and release cycles

Domains				**Release cycles**		
Size of Org.	Engineering Software	Ecom merce /CRM	Web, mobile Technologies	yearly , > yearly	half yearly, quarterly	4 weeks
Large	37	9	2	23	17	9
Medium	13	11	16	7	14	20
Small	5	6	4	3	7	6
Total	**55**	**26**	**22**	**33**	**38**	**35**

Typically software products grow in complexity and size over a period and as the products mature, less and fewer requirements are seen for successive releases. Adding features also becomes difficult as the impacts on existing products and customers need to be contained. Unless the technology changes demand major changes, mature products see less and less requirements and requirements that take longer time to implement. Maturity of the products, hence, is again an important factor in requirements understanding. As the data indicates, large companies handled more of mature products of 5 to 10years or beyond 10years. Medium companies handled more of 2 to 5 years young products. The responses are across the products of different maturity with 2 to 5 years products being on higher side. Table 4.9 /Figure 4.10 clearly indicate this aspect.

Table 4.9. Data Collection details - Maturity of the products handled

Maturity of the Products					
Size of org	<2y	2 to 5 y	5 to 10 y	>10 y	Total
Large	2	12	17	17	48
Medium	11	20	8	1	40
Small	7	5		3	15
Total	**20**	**37**	**25**	**21**	103

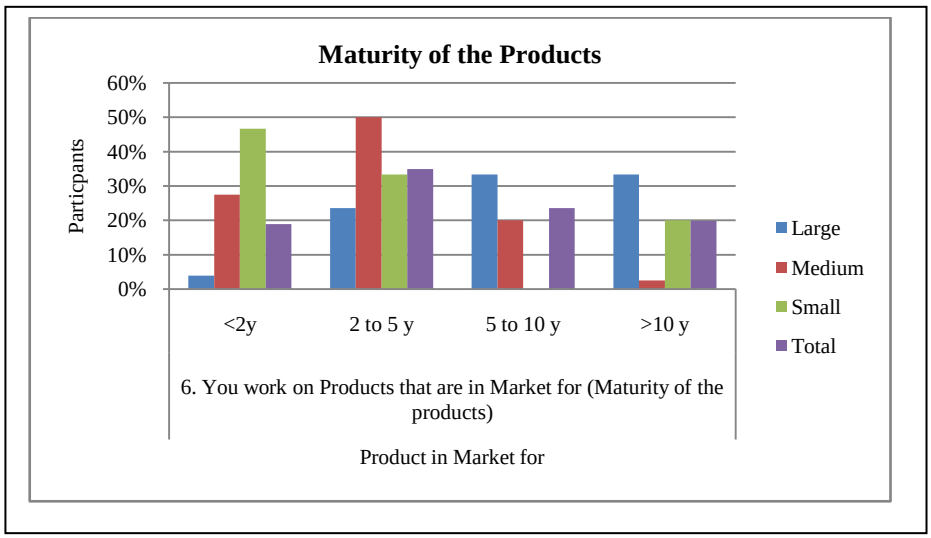

Fig. 4.10. Maturity of the products

Coming to the members that participated in the study, as is common in the software industry, majority of them participated in product planning and are involved in analysis, estimation and prioritization and also to some extent implementation. Many of them played the roles of product owners, project managers, leads. Roles and activities of the participants in detailed in Table 4.10/ Fig. 4.11, 4.12, 4.13.

Table 4.10. Data Collection details - Roles played by the participants

Role played in organization		**Participation in**		**Gets involved in Requirements**	
Role	No. of participants			Activities involved	No. of participants
Business Analyst	10	Product Planning	48	Analysis, Estimation, Implementation	19
CEO/Director/ MD/ Architect	12	product development	54	Analysis, estimation, prioritization	37
product owner/Project Manager/QA Manager/Lead	72	testing	0	Analysis, Estimation, Prioritization, Implementation	48
Lead developer	5	No response	4	No Response	2
Total	**106**	Total	**106**		**106**

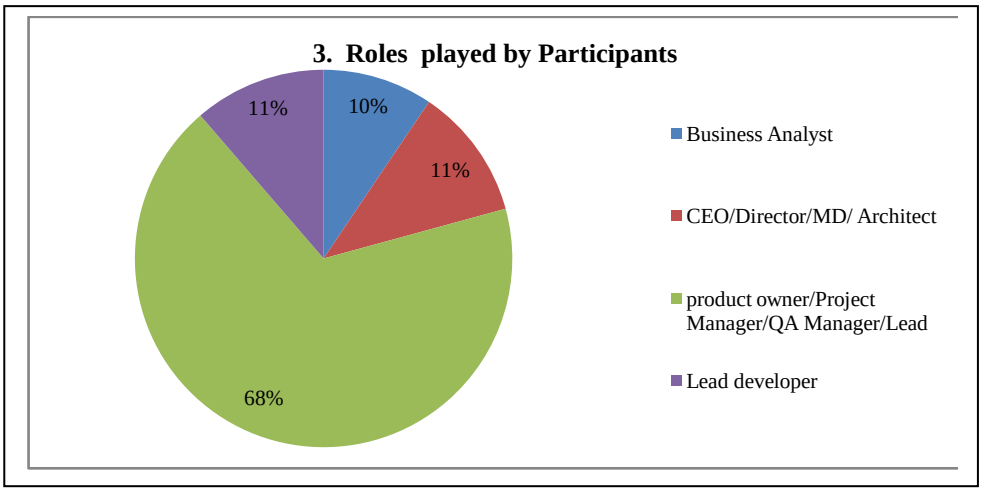

Fig. 4.11. Roles of Participants

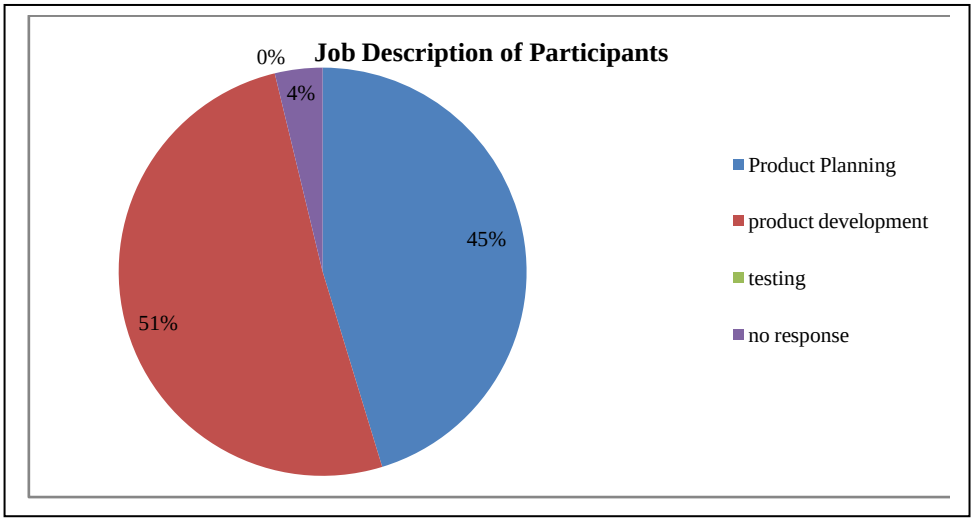

Fig. 4.12. Participants Activities I

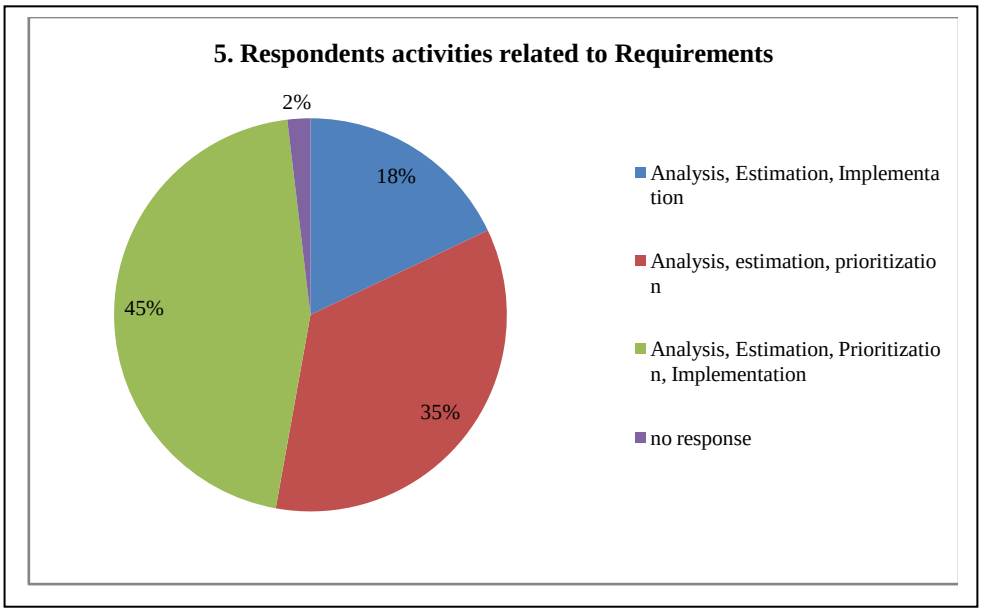

Fig. 4.13. Participants Activities II

4.6 Summary

Study questionnaire to elicit data on requirements prioritization practices in various organizations is detailed in this chapter. This chapter discussed study methodology and data gathered. In addition to the type of organizations and domains of the products, the data gathered encompassed the nature of activities related to software product development, the participants are involved in. The nature of software development that is carried out at the participants' organizations is discussed to demonstrate the spread of the data across domains, complexities of the software in terms of maturity of the products, release cycles adapted and multi –tier architectures. This data is gathered from section I of the questionnaire.

Of the 106 participants from 61 organizations, about 51 (48%) are from Large organizations, about 40 (38%) are from Medium organizations and about 15 (14%) are from small organizations. Large organizations handled multiple product lines and many products, where as medium organization have considerable product lines and products, while small organizations handled single product lines. Complexity of the products appeared similar across the organizations. Large and Medium organizations worked on global products, while small organizations worked on single country products. Large organizations are involved in Engineering and system software, while Medium and small catered for Ecommerce, web/mobile Software as well. Release cycle of the products tended to be longer – year, year+ in large organizations and medium and small have shorter release cycles – half year or less. Large organizations handled mature products compared to medium and small. 45% of the participants participated in products planning across. Participants participated in activities across Software development - analysis, estimation, prioritization and also implementation.

The analysis of section I data represents the breadth of software organizations and participants covered for the research. The data trends with respect to large, medium, small organizations are presented in the Table 4.11 below. Complete data is presented in numbers and percentages in Appendix A.

Table 4.11. Section I data Spread and Trends

Section-I	Factor/Size	Large	Medium	Small	Total
Product lines	multiple product lines	82%	55%	27%	64%
	single product line	12%	35%	67%	28%
	No product development	6%	10%	7%	8%
Products in Market for	<2y	4%	28%	47%	19%
	2 to 5 y	25%	50%	33%	36%
	5 to 10 y	35%	20%	0%	24%
	>10 y	35%	3%	20%	20%
Domains	Engineering Software - CAD/GIS/Telecom/transport/System	77%	33%	33%	53%
	Ecommerce/CRM	19%	28%	40%	25%
	Web, mobile Technologies	4%	40%	27%	21%
Release cycles	yearly, > yearly	47%	17%	19%	31%
	half yearly, quarterly	35%	34%	44%	36%
	4 weeks	18%	49%	38%	33%
Process followed	waterfall/iterative/Agile	48%	17%	21%	33%
	iterative/Agile	27%	17%	29%	23%
	Agile	25%	67%	50%	44%
Type of products	3 tier/n tier	45%	49%	57%	48%
	2 tier	43%	41%	14%	38%
	single tier	13%	10%	29%	14%
Extent of products	desktop/mobile/web/cloud	41%	8%	20%	24%
	desktop/web/mobile	49%	85%	67%	67%
	desktop	10%	8%	13%	9%
Market extent	All continents	70%	50%	43%	58%
	one continent	11%	18%	7%	13%
	one country	20%	33%	50%	29%

Section II, III data elaborating on the processes and problem areas and requirements prioritization methods followed by the organizations is discussed in the next chapter.

Chapter 5

5. Study –Processes and problem areas

5.1 Introduction

Information from Section I of the questionnaire forms the base for understanding the breadth of organizations and different types of products. As a next step information was elicited on processes followed at different organizations and problem areas in product development. Further, Information was gathered on Requirements handling aspects within and across different types of products and processes. Both process related and requirements handling aspects related information is analyzed in this chapter.

Section II of the questionnaire focused on the processes followed for software development and gathered information on what process is followed for development, how the requirements are collected and analyzed, problem areas like over work or overruns on time, with 10 questions. Section III focused on how the requirements are handled across the projects and has 20 questions, covering collection of requirements, prioritization methods used, areas of problems and current solutions adopted.

5.2 Nature of Responses on Processes

The study questionnaire has 3 sections and 42 questions, overall, and notable points from section II are discussed here. As discussed in the previous chapter, the domains of applications developed varied from Engineering applications to consumer applications, across manufacturing, telecom, finance to e-commerce. The products life cycle stage varied from less than 2 years to greater than 10 years. The applications are typically Enterprise applications, Web applications, Mobile applications working across devices and platforms, used in multiple countries and are mostly 3 tier applications.

This section analyses data on the process and requirements management. The processes followed across the organizations are Waterfall, Iterative and Agile, Agile being the predominant process. Whether the development process allows modifications at product level or module level or feature level – typical development process is depicted in Fig 2.3 of Chapter 2. Large companies followed a mix of processes, where as small companies preference has been with Agile. Participants responses on processes followed are presented in Table 5.1/ Figure 5.1. It appeared Agile process does not need to have the customer requirements understanding a priori and that this understanding can be brought in as the development progresses. This assumption, while might lead to some initial quick development, has the problem of lot of rework and wasted resource time and need for rework.

Table 5.1. Development processes

Size	Waterfall/Iterative/Agile	Iterative/Agile	Agile
Large	23	13	12
Medium	6	6	24
Small	3	4	7
Total	32	23	43

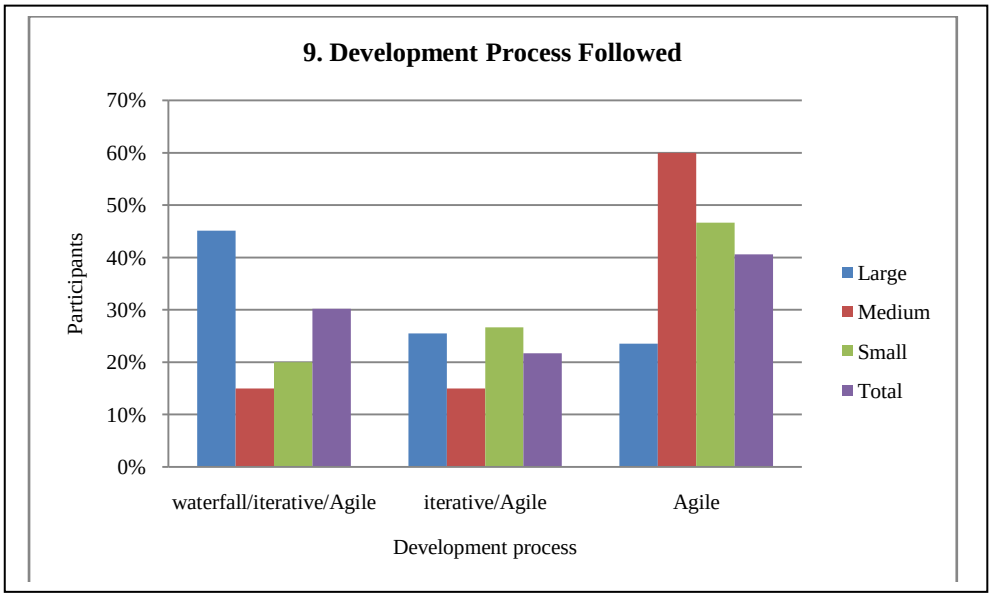

Fig. 5.1. Development Processes

Irrespective of the development process followed, a clear and systematic approach was not apparent from the responses for the question - How does the respondent choose features/requirements to be implemented for next release. It is largely based on customer needs alone. Business analysis is done in hardly 25% of the cases. Table 5.2/Figure 5.2 provides requirements selection methods followed by the participants. Changes in the requirements often resulted in extending the dates for releases.

Table 5.2. Requirements Selection Methods

Requirement Selection	No. of Responses
Based on Customer needs	38
Time to market/Development time needs	7
No preference/from backlog	10
Business Analysis	26
Impact Analysis	2
No Response	23
Total	106

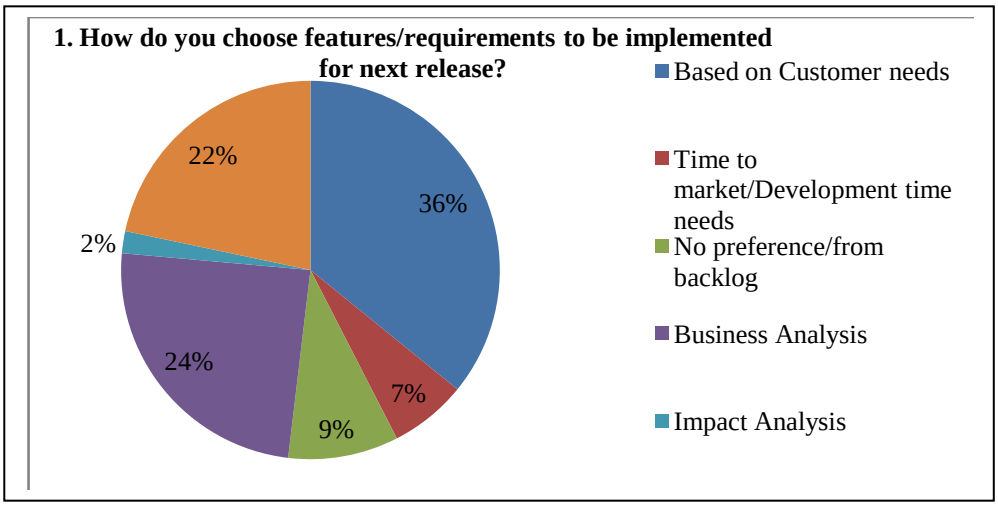

Fig. 5.2. Requirements Selection

Analysis of the responses to the question -"What are the problem areas you see with your current process of feature selection for upcoming release?" narrows down the problem areas with the current process followed, to analysis, estimation, planning. Problem areas as seen by the participants is provided in Table 5.3/Figure 5.3. Significant number of participants did not provide any response to this issue. This perhaps indicates to living with problems through product development.

Table 5.3. Problem Areas

Problem Areas	No. of Responses
Estimation- time resources	22
Lack of prioritization with respect to complexity, time	17
Requirement clarity/change in requirements	16
Dependencies - other modules, new tech	10
Lack of business Analysis	10
No response	31
Total	106

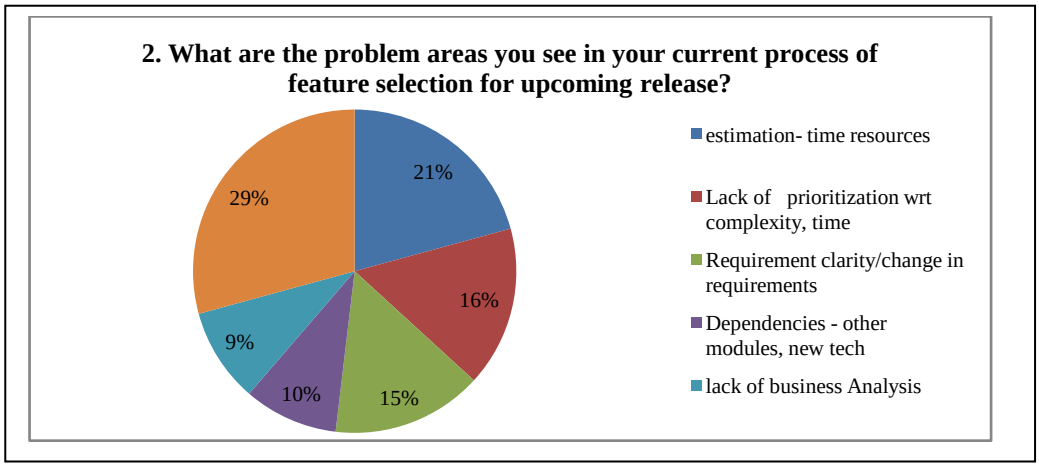

Fig. 5.3. Problem Areas

The question - How do you circumvent the problems with your current process of feature selection? - indicated to typical solutions being followed like over work, extended releases, attempts to convince clients. Often the teams worked under pressure and for long hours in order to meet requirements for release. Typical solutions adapted are listed in Table 5.4/Figure 5.4. Significant number of participants have not provided any response to this query, again indicating continuing with problems during development.

Table 5.4. Solutions adapted

Circumventing problems	No. of Responses
Client management/ meetings	17
discussions with stake holders	25
do nothing	1
Extra time and Hard work	12
Estimate/Extend/Analyze	15
No response	36
Total	106

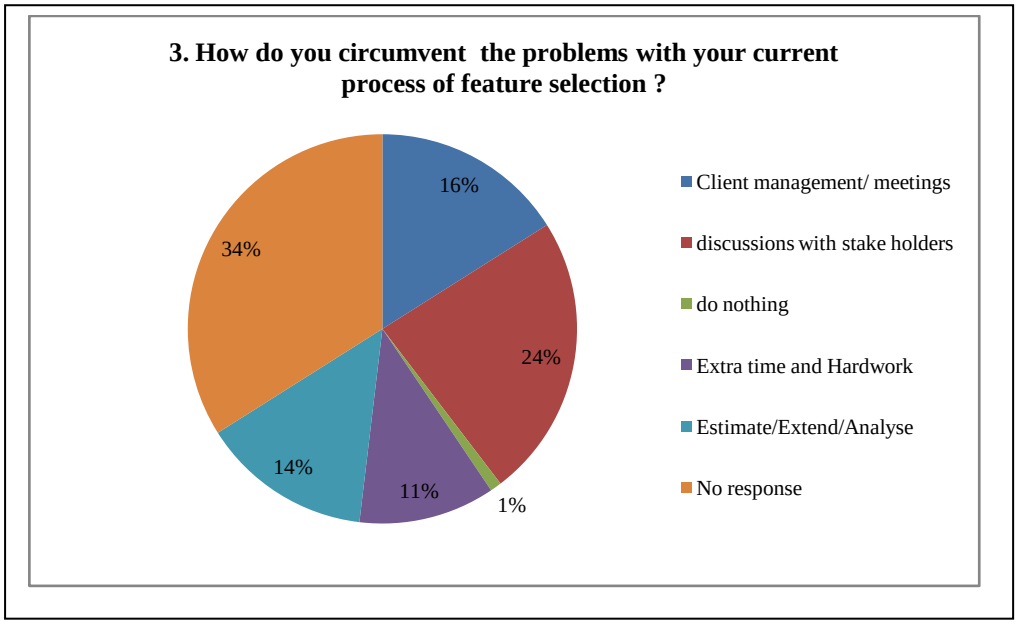

Fig. 5.4. Solutions Adapted

In general, teams worked for additional time often and re-planning the releases by abandoning some features and adding new features is needed. This can be attributed to lack of sufficient analysis and prioritization. Problems faced in software development release cycles are analyzed further based on the responses to questions in section II. Working under pressure and for long hours, some team members getting over worked is often seen due to lack of sufficient analysis of requirements. This is evident from Table 5.5/ Figure 5.5, 5.6.

Table 5.5. Problems in Release cycles – work pressure

Response	Working under pressure & Long Hours	Overworked team members
often	35	32
very often	15	17
sometimes	40	42
rarely	13	12
no response	3	3
Total	106	106

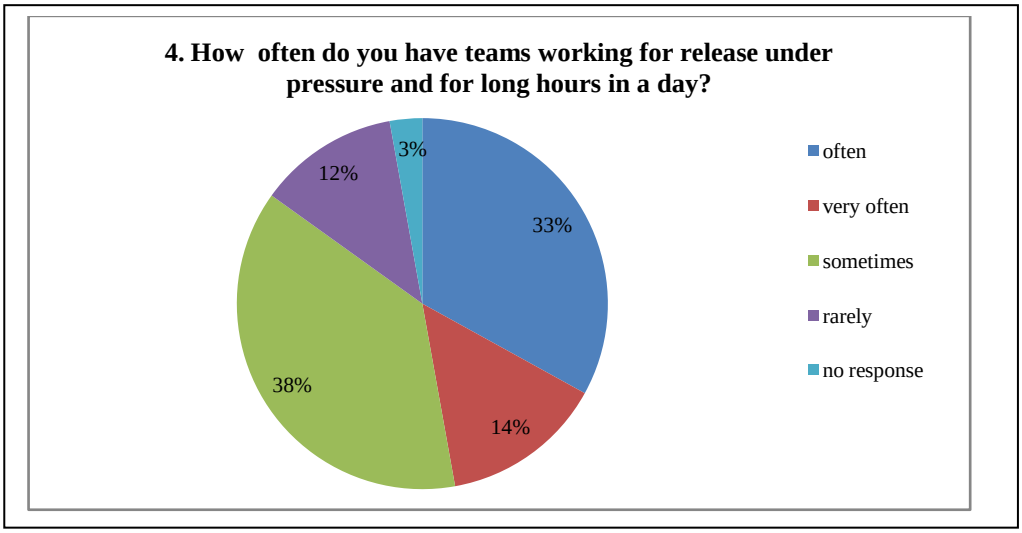

Fig. 5.5. Problems in Release cycles – working under pressure

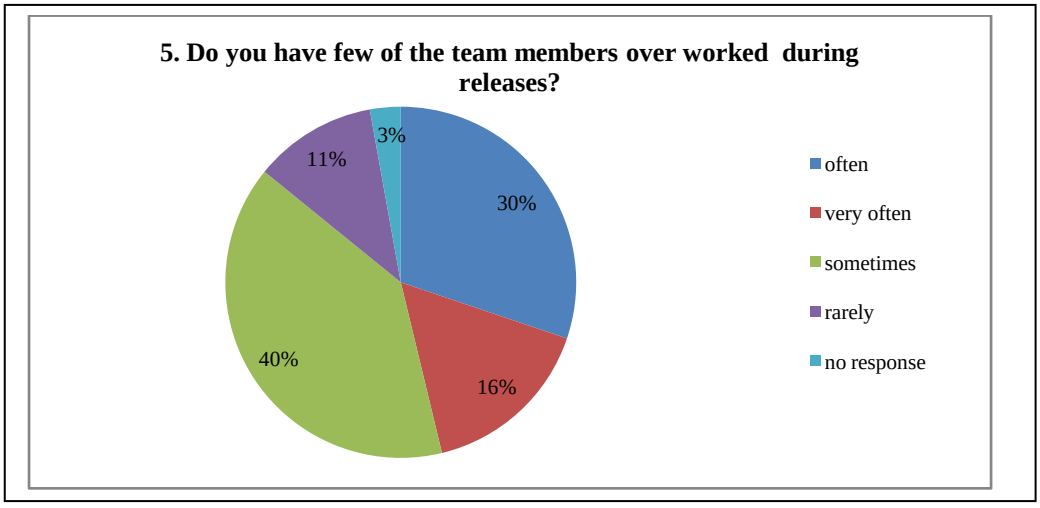

Fig. 5.6. Problems in Release cycles – over worked members

Abandoning features due to changes in requirements either due to customer initiated changes or due to lack of understanding or lack of sufficient analysis and starting on new features during release cycles is generally not an option but is the only solution sometimes. Similar response is seen for impacts of new features on existing customers Table 5.6 / Figure 5.7, 5.8 show this clearly.

Table 5.6. Problems in Release cycles-dropping features

Response	Abandoning features & restarting new features	Abandoning features due to impact on customers
often	10	16
very often	4	7
sometimes	45	34
rarely	44	43
no response	3	6
Total	106	106

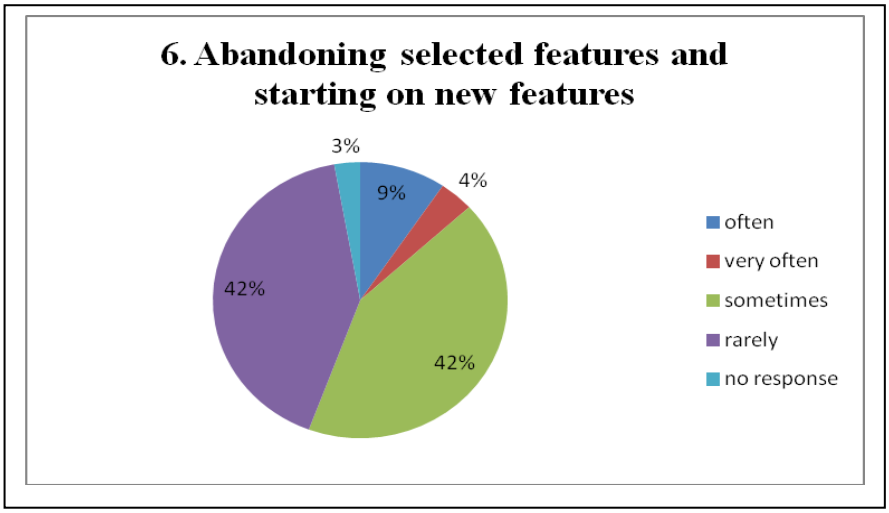

Fig. 5.7. Problems during Release cycles – changing features

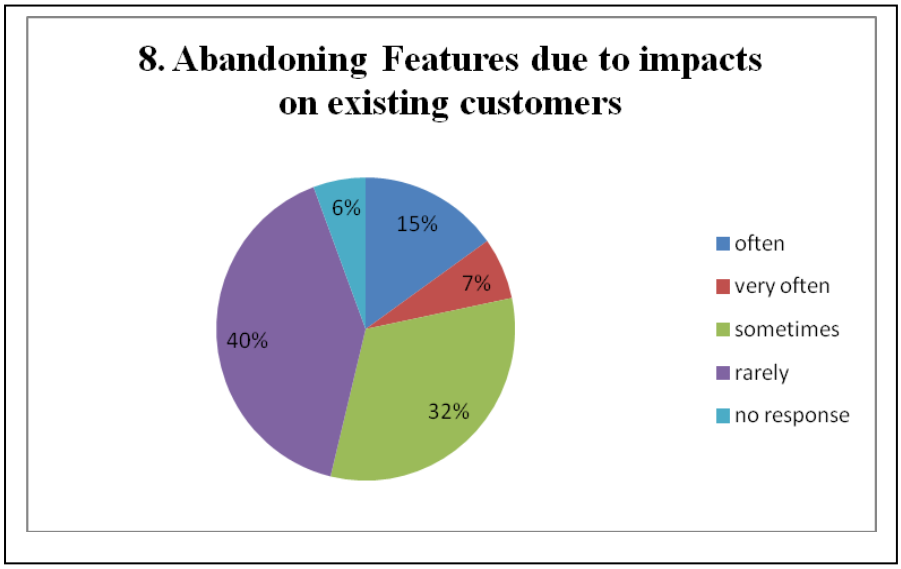

Fig. 5.8. Problems during Release cycles– dropping due to impacts

In addition to additional time needs and features changes, incorrect estimation of resources and time needed for development of the requirements leads to resource related problems during release cycles. Availability of appropriately skilled knowledge resources and

reworking of the resource estimates are significant problems as is evident from Table 5.7/Figure 5.9, 5.10.

Table 5.7. Problems during Releases cycles

Response	Problem of right resources availability	Reworking resource estimates
often	35	43
very often	11	8
sometimes	37	41
rarely	19	11
no response	4	3
Total	106	106

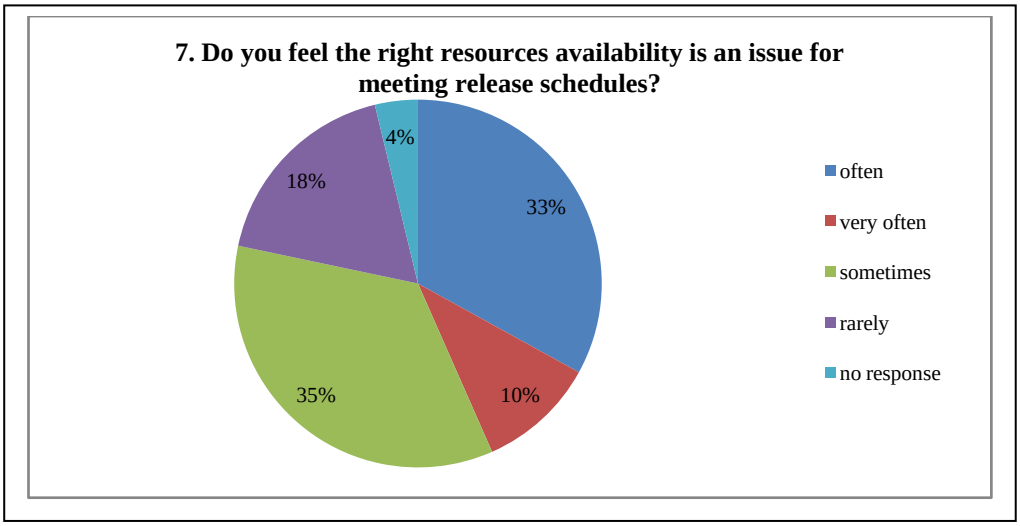

Fig. 5.9. Problems during Release cycle – resources availability

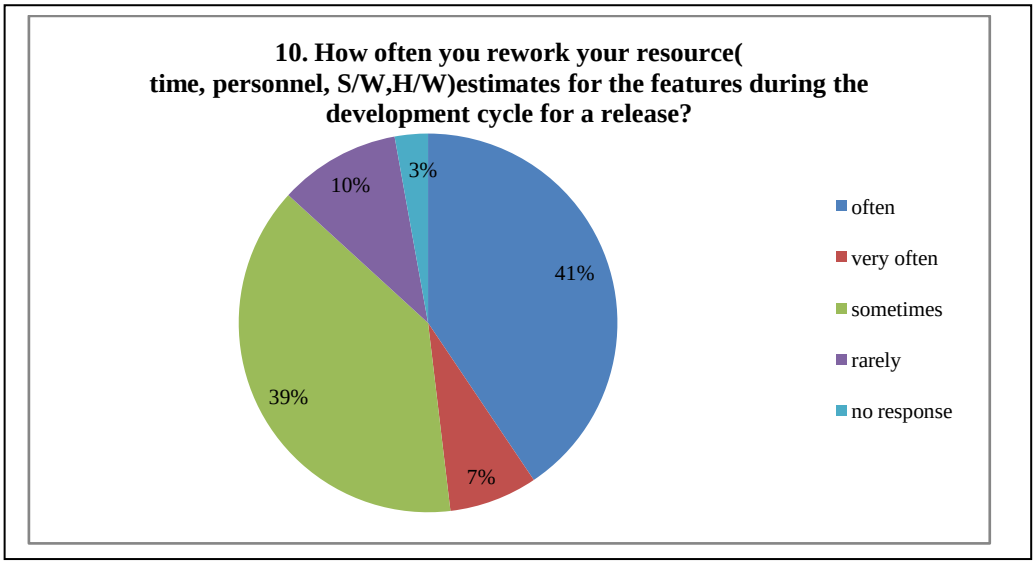

Fig. 5.10. Problems during Release cycle - resources estimates

The impacts of new features on existing components of the products or frameworks that form the core on which the new requirements would be built appears to be analyzed most of

the times, though some responses indicate it is only done sometimes. Responses on this area are provided in Table 5.8/Figure 5.11.

Table 5.8. Impacts Analysis

Response	Analysis of impacts on core
often	40
very often	34
sometimes	22
rarely	7
no response	3
Total	106

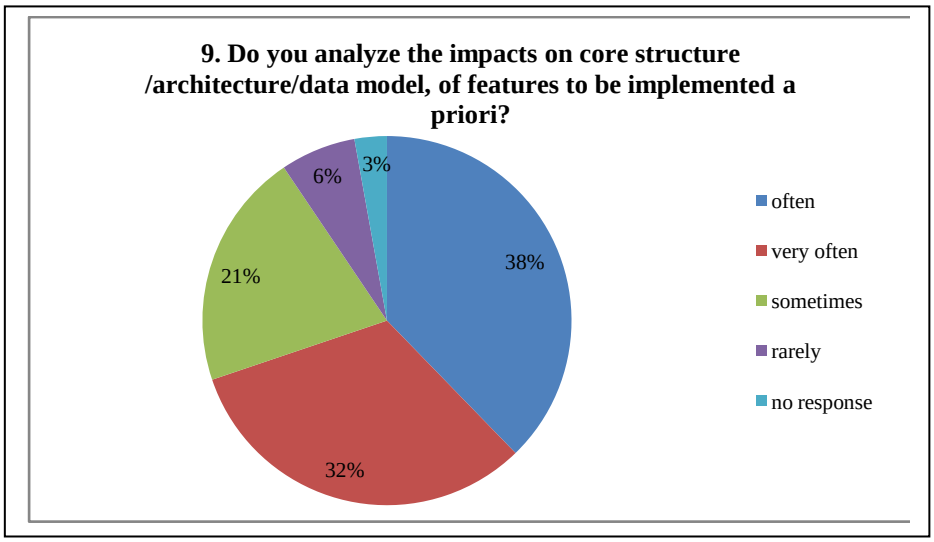

Fig. 5.11. Impacts on core analysis

Summarizing the responses on the processes and problems, it can be said resources availability and type of resources needed, impacts on existing customers are partially considered for prioritizing requirements for upcoming releases. Impacts of new requirements on existing product structures is not taken into account by everyone. Fig. 5.12, 5.13 give classification of data on these three aspects and resulting rework of resource bandwidth for releases. Lack of clear cut requirements analysis prioritization resulted in teams working for additional time often and also in re-planning the releases by abandoning some features and adding new features into the release. Lack of sufficient analysis and prioritization of requirements most often results in extended releases, attempts to convince clients of changes in scope of requirements. Often the teams worked under pressure and for long hours in order to meet requirements for release.

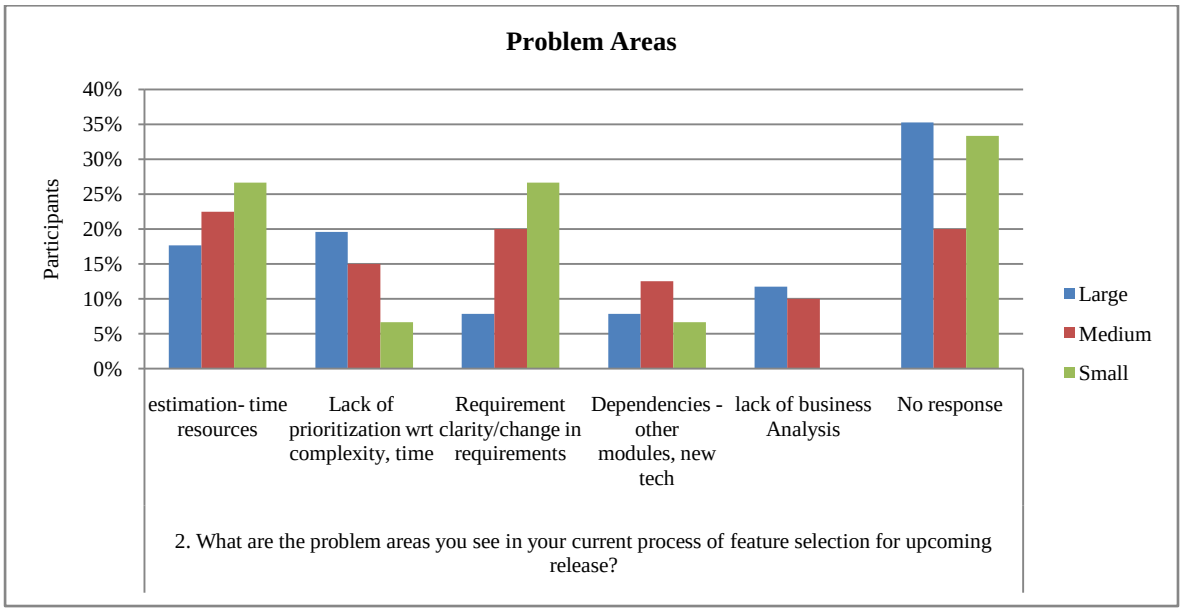

Fig. 5.12.Resources and Impacts Problems during Release cycle

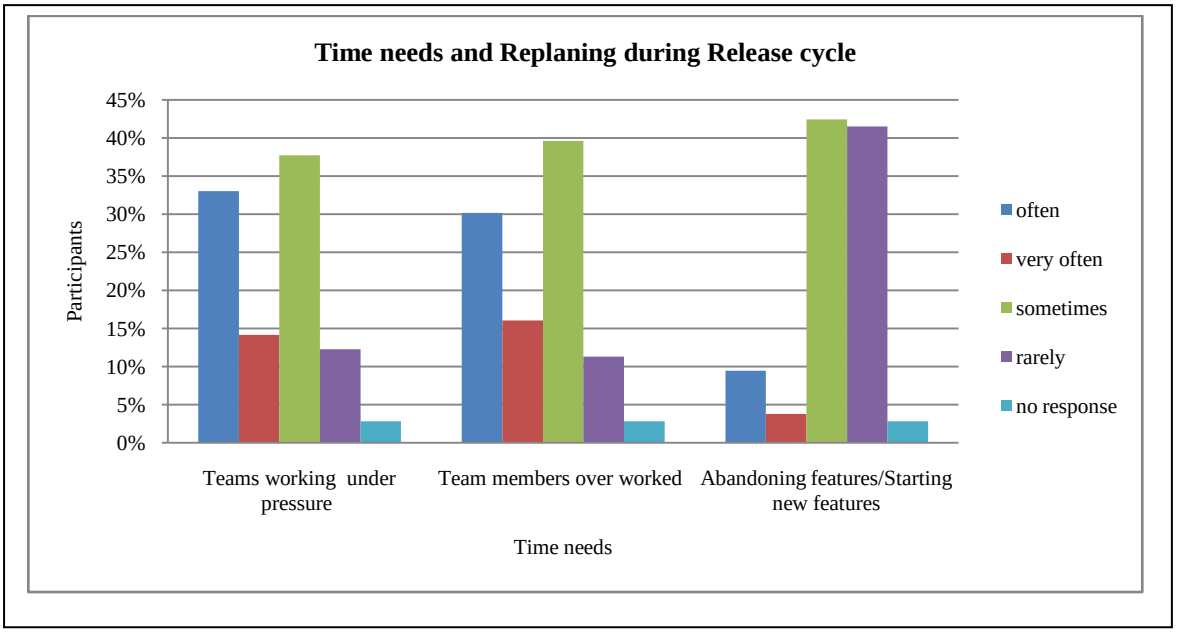

Fig. 5.13. Time needs and Replanning Problems during Release cycle

Lack of clear cut requirements' analysis and prioritization resulted in teams working for additional time often and also in re-planning the releases by abandoning some features and adding new features into the release.

Analyzing and taking into account resources availability, impacts on existing customers are two areas that seem to be only partially considered for defining requirements for upcoming releases. Impacts of new requirements on existing product structures is another area that seem to be not taken into account by everyone.

Any changes in requirements during release cycle are normally handled by extending the release dates, reducing number of features or reprioritizing as indicated in Table 5.9/Figure 5.14.

Table 5.9. Handling Requirements Changes

Changes in requirements handled by	Responses
Extending release date	13
Removal/ addition of some requirements	32
Reprioritization/Extending release date	34
Removal/ addition of some requirements, Extending release date	18
no response	9
Total	106

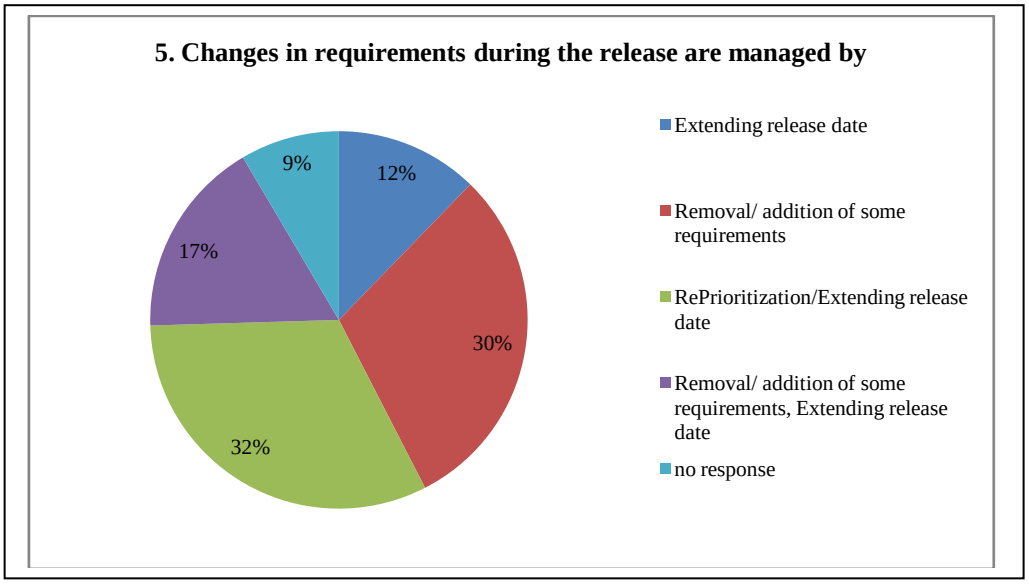

Fig. 5.14. Handling Requirements Changes

5.3 Summary

Data on processes followed in software development organizations is presented in this chapter. The study conducted across organizations developing software products - first version to multiple versions, brings out the lack of systematic methods usage for requirements handling. It also brought out the associated problem areas and difficulties in achieving successful software product deliveries. Problems related to requirements uncertainties, resources availability are identified. The impacts of the problems on the product deliveries are explored.

Large organizations followed a mix of processes – Waterfall, Iterative, Agile, while Medium and small followed Agile process. The choice of process can be linked to the nature of products handled, life cycle of the products and the release cycles followed. Customer needs appeared to influence the requirements selection for the upcoming releases. With all three processes across organizations, dominant problem areas are estimation, lack of prioritization, lack of scope clarity, changes in requirements, dependencies on technology/other modules, in that order. Solutions adapted to address the problems revolved

around negotiating with clients, working hard, extending release dates. Reworking of resources is a problem area during release cycle. While abandoning features from release is rarely done, removing/adding features and reprioritization is done to address changes in requirements during releases. The data trends with respect to large, medium, small organizations are presented in the table 5.10 below.

Table 5.10. Data trends from Section II

Section II	**Factor/Size**	**Large**	**Medium**	**Small**	**Total**
	/Total responses	51	40	15	106
1. How do you choose features/requirements to be implemented for next release?	Based on Customer needs	25%	40%	60%	36%
	Time to market/Development time needs	4%	10%	7%	7%
	No preference/from backlog	20%	0%	0%	9%
	Business Analysis	27%	25%	13%	25%
	Impact Analysis	2%	3%	0%	2%
	No Response	22%	23%	20%	22%
2. What are the problem areas you see in your current process of feature selection for upcoming release?	estimation- time resources	18%	23%	27%	21%
	Lack of prioritization wrt. complexity, time	20%	15%	7%	16%
	Requirement clarity/change in requirements	8%	20%	27%	15%
	Dependencies - other modules, new tech	8%	13%	7%	9%
	lack of business Analysis	12%	10%	0%	9%
	No response	35%	20%	33%	29%
3. How do you circumvent the problems with your current process of feature selection ?	Client management/ meetings	12%	15%	33%	16%
	discussions with stake holders	20%	28%	27%	24%
	do nothing	2%	0%	0%	1%
	Extra time and Hard work	6%	23%	0%	11%

	Estimate/Extend/Analyze	22%	10%	0%	14%
	No response	39%	25%	40%	34%
4. How often do you have teams working for release under pressure and for long hours in a day?	often	24%	43%	40%	33%
	very often	22%	5%	13%	14%
	sometimes	37%	43%	27%	38%
	rarely	14%	8%	20%	12%
	no response	4%	3%	0%	3%
5. Do you have few of the team members over worked during releases?	often	25%	33%	40%	30%
	very often	16%	18%	13%	16%
	sometimes	45%	35%	33%	40%
	rarely	10%	13%	13%	11%
	no response	4%	3%	0%	3%
6. How often do you abandon features being implemented for a release and restart on new features?	often	10%	8%	13%	9%
	very often	8%	0%	0%	4%
	sometimes	39%	43%	53%	42%
	rarely	39%	48%	33%	42%
	no response	4%	3%	0%	3%
7. Do you feel the right resources availability is an issue for meeting release schedules?	often	31%	35%	33%	33%
	very often	10%	10%	13%	10%
	sometimes	35%	30%	47%	35%
	rarely	18%	23%	7%	18%

	no response	6%	3%	0%	4%
8. How often do you abandon features during release due to realized impacts on existing customers?	often	14%	18%	13%	15%
	very often	6%	5%	13%	7%
	sometimes	35%	25%	40%	32%
	rarely	41%	48%	20%	41%
	no response	4%	5%	13%	6%
9. Do you analyze the impacts on core structure /architecture/data model, of features to be implemented a priori?	often	35%	40%	40%	38%
	very often	35%	28%	33%	32%
	sometimes	18%	25%	20%	21%
	rarely	8%	5%	7%	7%
	no response	4%	3%	0%	3%
10. How often you rework your resource(time, personnel, S/W,H/W)estimates for the features during the development cycle for a release?	often	35%	53%	27%	41%
	very often	8%	5%	13%	8%
	sometimes	41%	30%	53%	39%
	rarely	12%	10%	7%	10%
	no response	4%	3%	0%	3%

Next chapter brings out requirements prioritization methods usage and preferences for parameters considered for prioritizing the requirements.

Chapter 6

6. Methods of Requirements Prioritization and Factors influencing Requirements Prioritization

6.1 Introduction

Large companies offering generic solutions as products and providing customization for specific business needs, as well as small to medium companies providing one time solutions and then enhancing and upgrading the solutions strive to meet the needs of different business segments for their IT enablement. Requirements are gathered, analyzed, refined, prioritized as per client's business needs, priorities, technology changes, compatibilities with available components and resource needs. Irrespective of the development process followed, requirements need to be analyzed, prioritized in order to be able to deliver software to the customers within the constraints of time, resources, technological limitations. Lack of this important step in software development leads to problems of dissatisfied customers, burnt out resources, suboptimal software, as observed in the previous chapter.

This chapter analyses the responses gathered in section III of the study for understanding the requirements prioritization methods used in practice, sufficiency of these methods, factors considered for prioritization. It is imperative to understand what factors influence the selection of requirements for the next release under expanding client needs, cost and time implications. Prioritization of requirements and planning releases taking into account these factors helps streamline the project deliveries to client's satisfaction without overworking the teams or missing time to market deadlines. Responses for the 20 questions of section III are presented and analyzed in the following section.

Analysis is focused on factors influencing the selection of requirements for the next release under expanding client needs, cost and time implications. Prioritization of requirements and planning releases taking into account these factors helps streamline the project deliveries to client's satisfaction without overworking the teams or missing time to market deadlines.

6.2 Requirements prioritization

Typically requirements flow from different players involved in software development. While broad direction, evaluating business value, is set by the executive management, Marketing teams, keeping in touch with customers bring in most needed and high returns requirements. Development team analyzes technical aspects and feasibility of development and costs involved. Existing customers' feedback is kept in mind to meet their requirements. Involvement of different players in requirements gathering is indicated in Table 6.1/Figure 6.1.

Table 6.1. Requirements collection methods

Requirements gathering means	Responses
Marketing team, Executive Direction, Development Team, Customer Change Requests	54
Marketing team, Executive Direction, Customer Change Requests	11
Executive Direction, Development Team, Customer Change Requests	26
Marketing team, Development Team, Customer Change Requests	12
No Response	3
Total	106

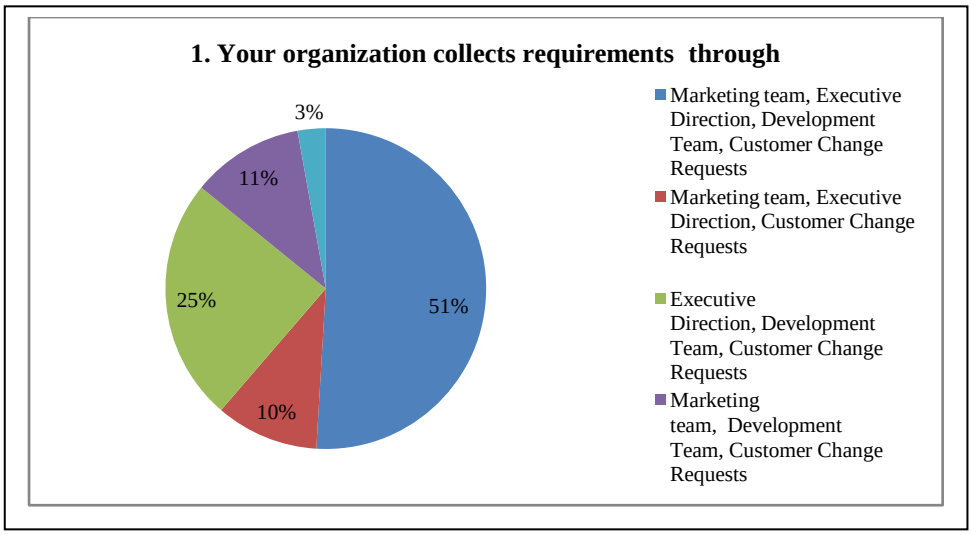

Fig. 6.1. Requirements collection methods

It is evident that all players are involved in general, for requirements gathering. Once the requirements are gathered, they are analyzed and assessed for inclusion into development. Planning team and development team play an important role in assessment in requirements. Responses on - who does assessment of requirements is presented in Table 6.2/ Figure 6.2.

Table 6.2. Requirements assessment

Requirements Analysis done by	Responses
Business Dev, product management team	5
Planning, dev teams	31
Planning, stakeholders, dev, pre sales	30
stake holders , dev	36
no response	4
Total	106

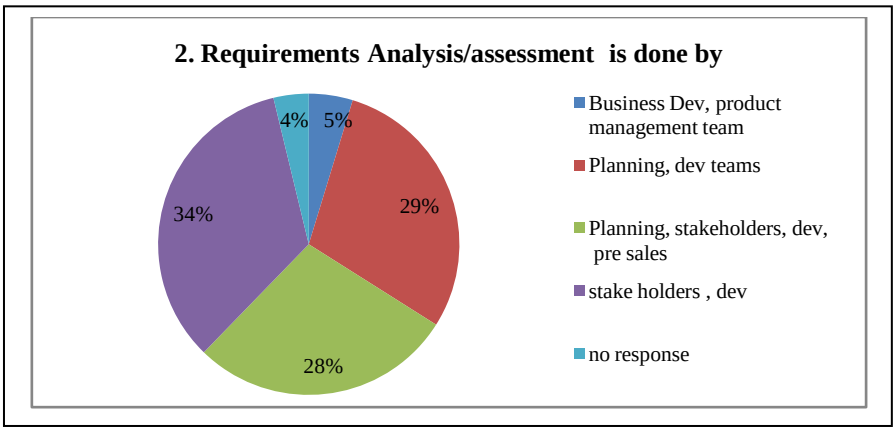

Fig. 6.2. Requirements Assessment teams

Product team discussions and Executive direction seem to be the major influencing factors in evaluating the requirements. It is discussions, rather than a formal framework or methods which are basis for the evaluation and prioritization of requirements. Table 6.3/Figure 6.3 indicates this aspect of requirements prioritization in practice.

Table 6.3. Requirements Evaluation

Evaluation done by	**Responses**
Using a Framework, Product Team discussions, Executive Direction	14
Product team discussions, Executive Direction	32
Product Management, Client discussions	6
Product team discussions	48
no response	6
Total	106

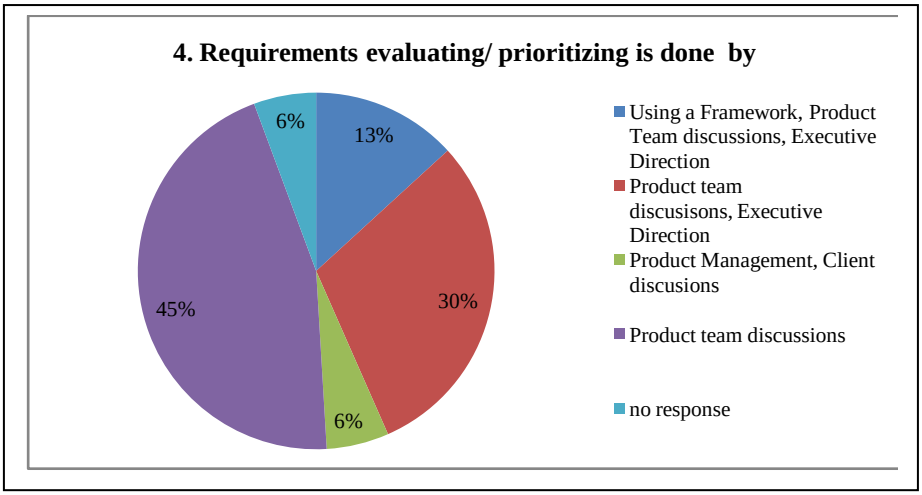

Fig. 6.3. Requirements Evaluation

Typically requirements are ranked based on value proposition, resource availability, time availability. Responses indicate all three factors being considered most of the times. Table 6.4/Figure 6.4 listed the preference for using the parameters in practice.

Table 6.4. Prioritization Factors

Prioritization by	Responses
Ranking by Value proposition, Resource, time availability	55
Ranking by Value proposition	28
Resource, time availability	19
No response	4
Total	106

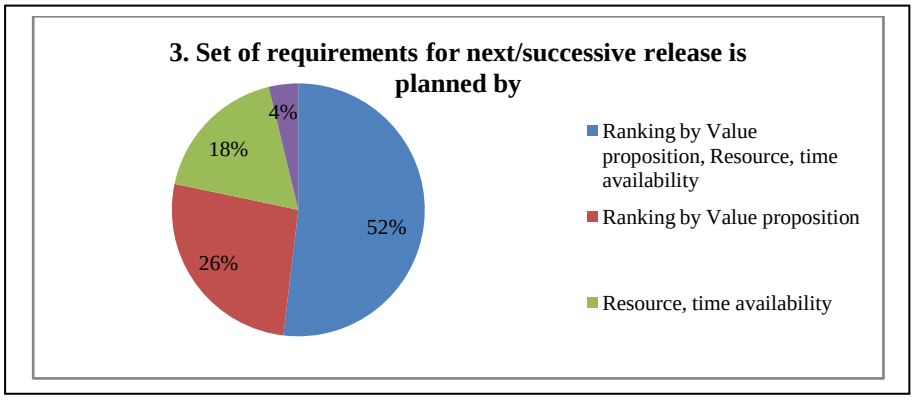

Fig. 6.4. Prioritization Factors

Classifying requirements into three groups of - must have, good to have and need not have – is generally the familiar method followed for requirements prioritization for product releases, though not a precise method. Table 6.5/ Figure 6.5 indicates priority grouping to be a sufficient method often.

Table 6.5. Priority grouping for Prioritization

Priority grouping	Responses
Not sufficient at all	4
Not sufficient often	19
Sufficient always	26
Sufficient often	50
No response	7
Total	106

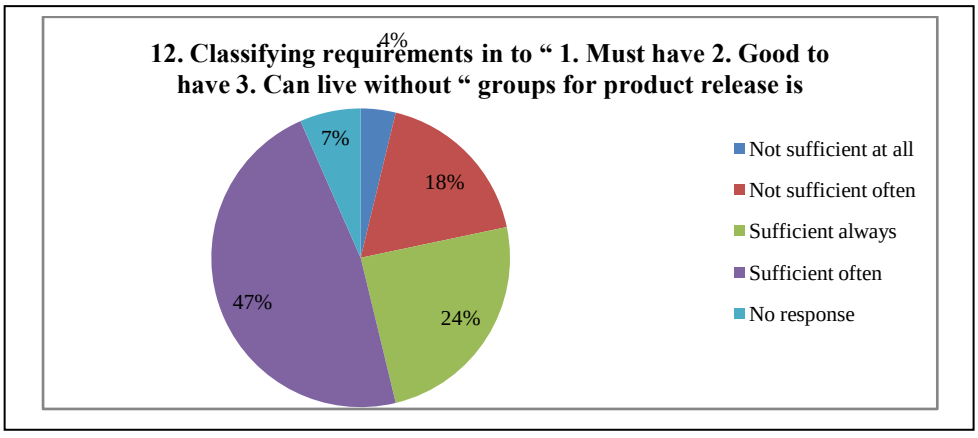

Fig. 6.5. Priority grouping

Ranking of requirements based on a preferred parameter, numerical assignment of priority for prioritization of requirements do find a favor by many, though not by all participants. The two methods are not considered sufficient always. Perception about cost-value ratio for prioritization has similar response. Table 6.6/ Figure 6.6, 6.7, 6.8 show the responses for Numerical assignment, Ranking and Cost –Value methods for prioritization.

Table 6.6. Prioritization methods

	Query responses		
Response\query	Cost – Value ratio for requirements is the best indicator of priority	Ranking of requirements(in sequence of priority) based on a parameter is sufficient for prioritization	Numerical assignment of priority (grouping by assigning priority 1,2,3,..) to requirements is sufficient
Most often	44	48	53
Always	15	16	12
Not always	33	31	35
never	4	3	
No response	10	8	6
Total	106	106	106

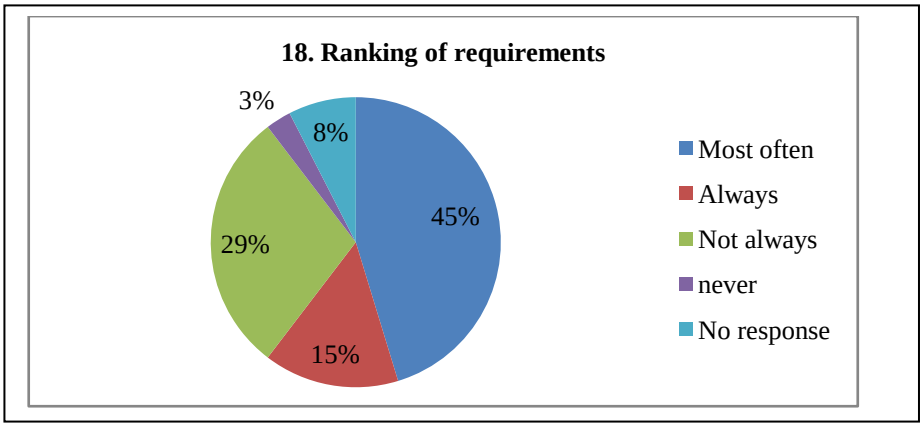

Fig. 6.6. Ranking Preference

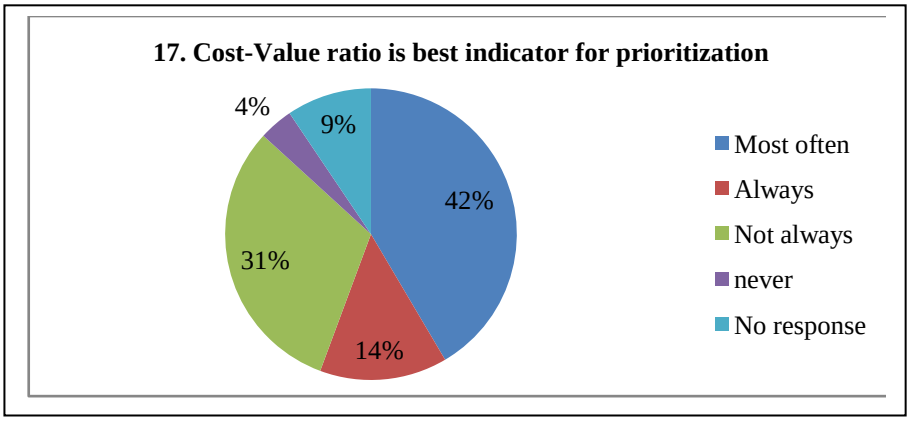

Fig. 6.7. Cost-Value Preference

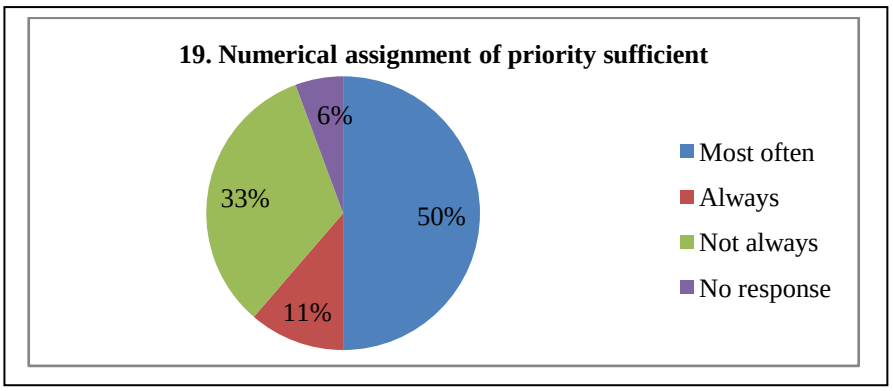

Fig. 6.8. Numerical Assignment Preference

It is often sufficient to just know the ordinal sequence of the priorities where it is known that one requirement comes ahead of another requirement. How much more important or how many times a requirement is important than another requirement, as in ratio scale is often an involved process of determination. Attempt has been made to elicit preference on this aspect and is presented in Figure 6.9, 6.10. The importance of knowing how much more important a requirement is with respect to another is evident from 80% perceiving it to be essential. Though in practice, it is the relative importance that is often used with 67% responses assuming sufficiency of the relative importance.

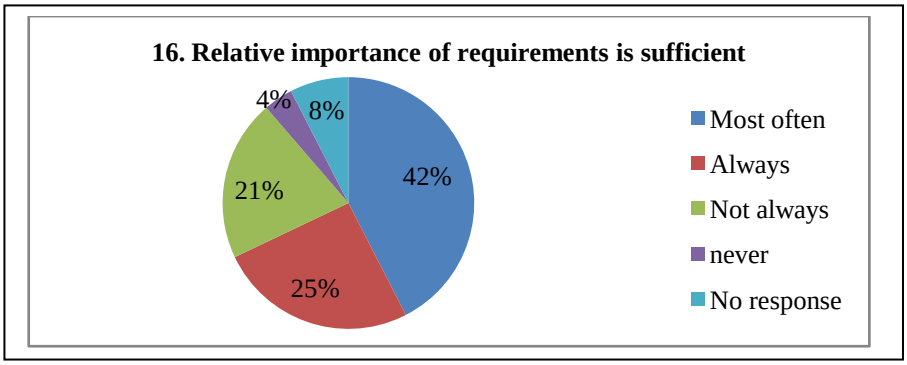

Fig. 6.9. Relative (ordinal) Importance Preference

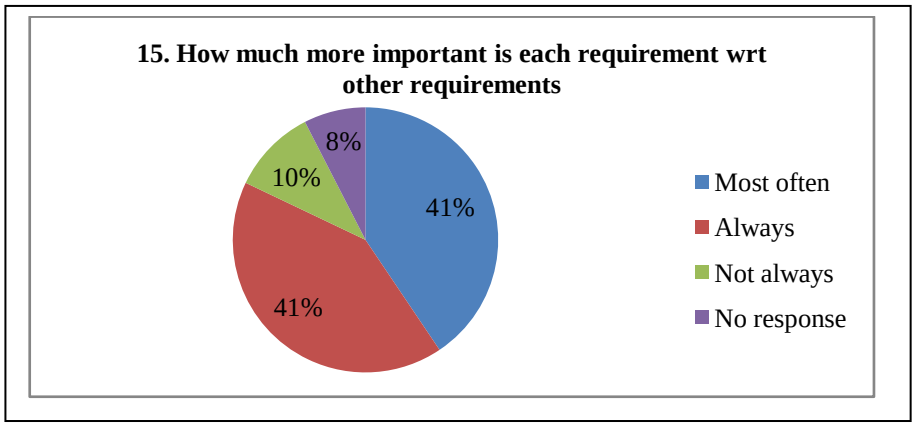

Fig. 6.10. Relative (ratio) Importance Preference

While Analytical Hierarchy process enables deriving exact relative importance of each requirement with respect to each other requirement, the method does not seem to have found awareness or usage in practice. While it is a method involving pair wise comparisons for each criteria that grows in complexity as the number of requirements increase, there exit tools to enable usage of the prioritization method. Cost-Value ratio method, in turn utilizes AHP for each of the parameters- Cost and Value. Responses on AHP are presented in Figure 6.11, 6.12.

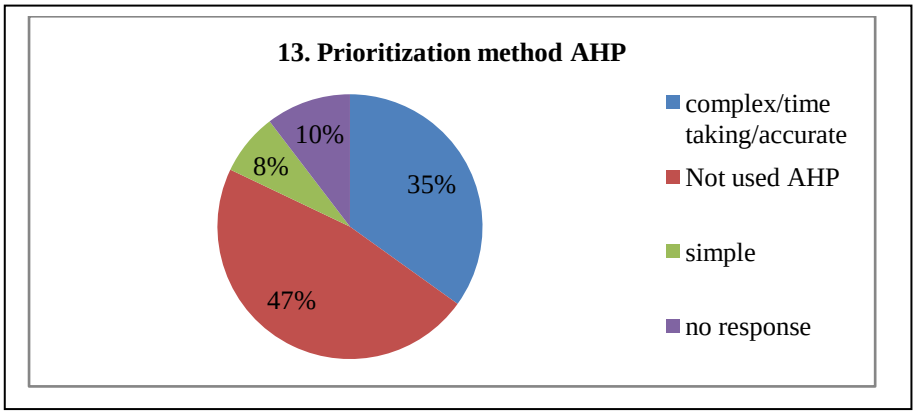

Fig. 6.11. Awareness of AHP

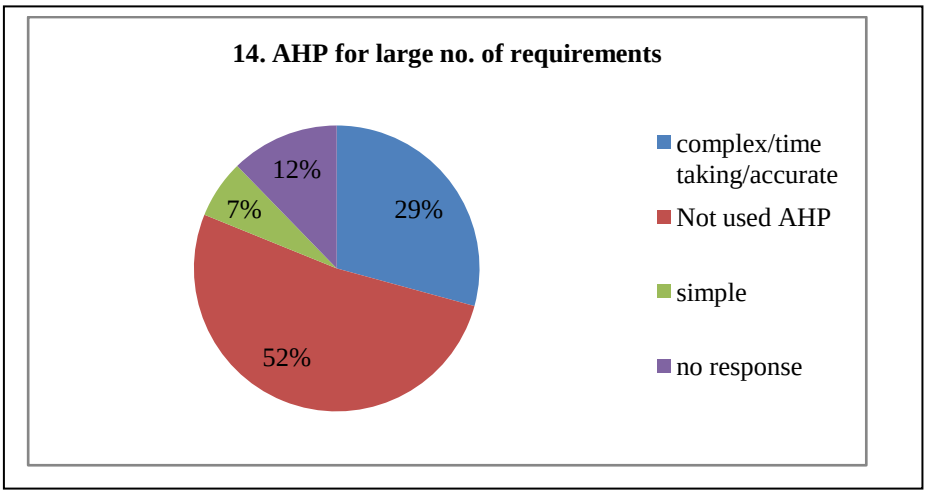

Fig. 6.12. Scaling of AHP

The responses indicate to simple methods usage often, though not sufficient for the important aspect of prioritization of requirements for deciding what goes into the upcoming release. Organizations live with the problems of scope expansions, changes in requirements, impacts discoveries and teams bear the brunt of problems in terms of additional work that needs to be put in through additional hours and working under pressure. The clients bear the consequences in terms of delayed releases, less quality and solutions not meeting their needs. The following section analyzes further various parameters that participants consider for requirements prioritization often.

6.3 Parameters for Requirements Prioritization

From Industrial experience and study of literature, 5 parameters have been considered relevant for prioritization and the study elicited responses on these parameters to understand the usage patterns of the parameters. The parameters are listed below.

BV - Business Value

AR - Availability of Resources

TM - Time to Market

DI - Difficulty of Implementation

IC - Impact on existing Customers/Core modules

61% of the participants considered at least 3 of the factors, for prioritization. Business Value of the requirement appears to be the most used parameter, with 70+ participants indicating use of Business Value for prioritization. while the rest of the four parameters are used to similar extent for prioritization, with 50 to 60 participants indicating use of these parameters. Figure 6.13 shows the extent to which these parameters are considered for prioritization of requirements. Number of responses for preference to each of these parameters is collated in Figure 6.14.

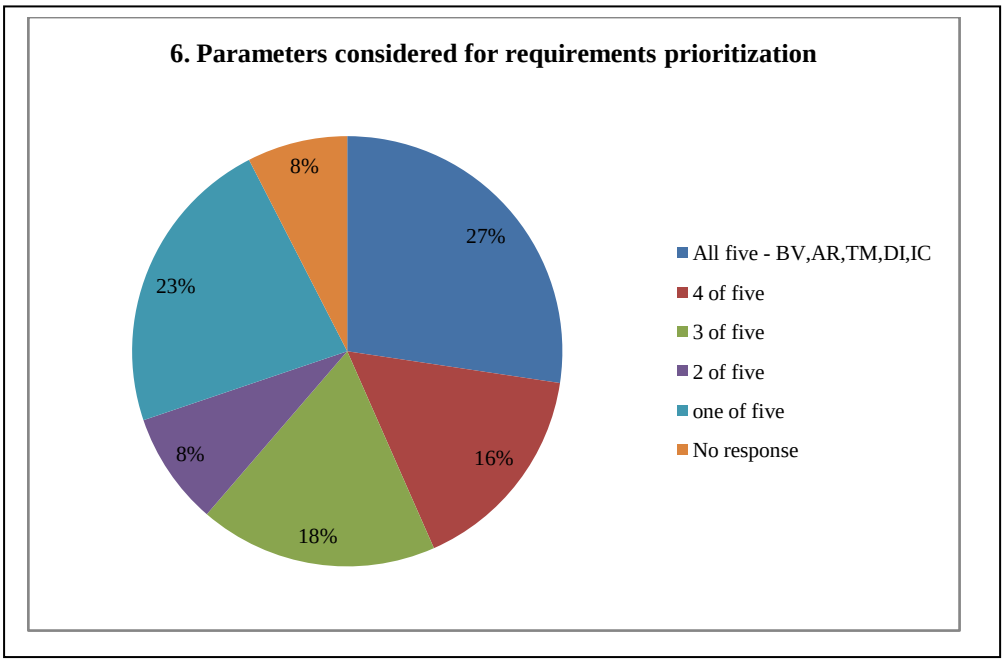

Fig. 6.13. Factors for Prioritization

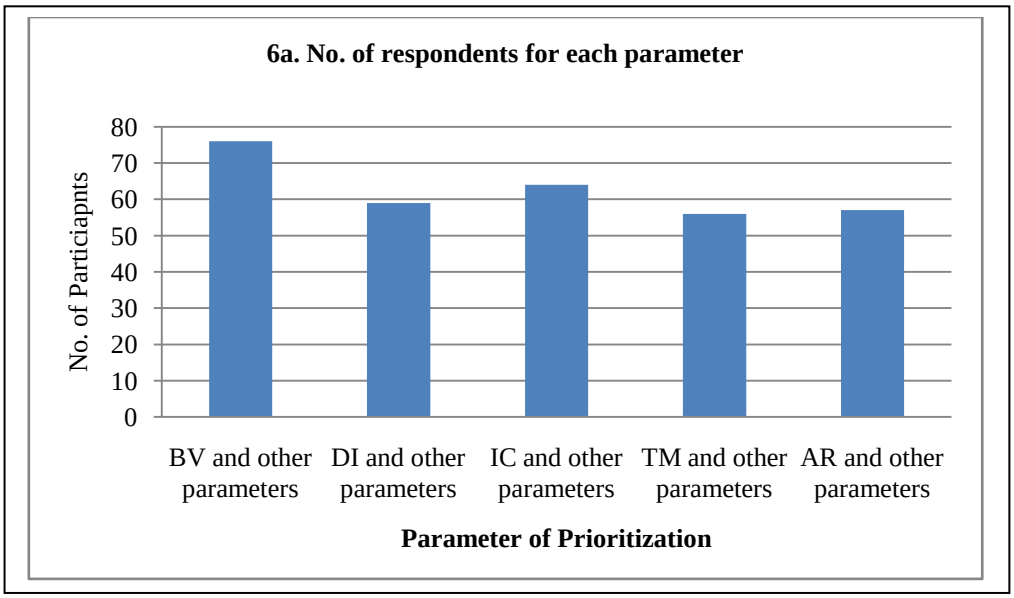

Fig. 6.14. Preference to Factors for Prioritization

Order of applying these parameters in successive stages for prioritizing is indicated in Fig. 6.15. Preference to a specific order of these parameters is plotted in Fig 6.16.

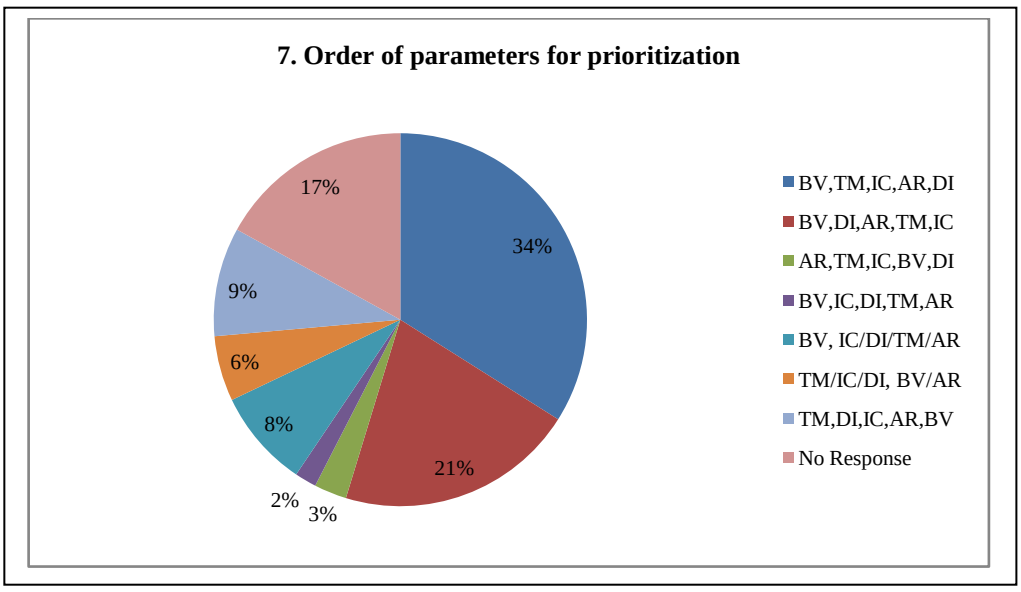

Fig. 6.15. Order of Applying Factors

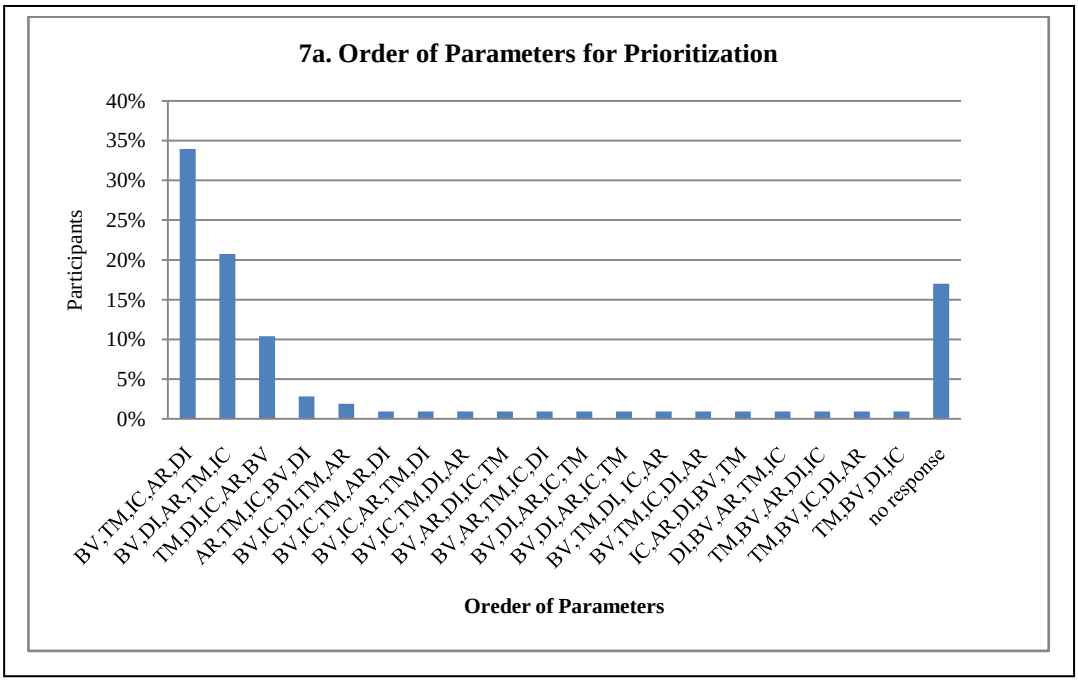

Fig. 6.16. Preference of order of Parameters

Associating weights to the parameters to arrive at appropriate priorities as a combination of weighted parameters is often used as indicated in Fig6.17.

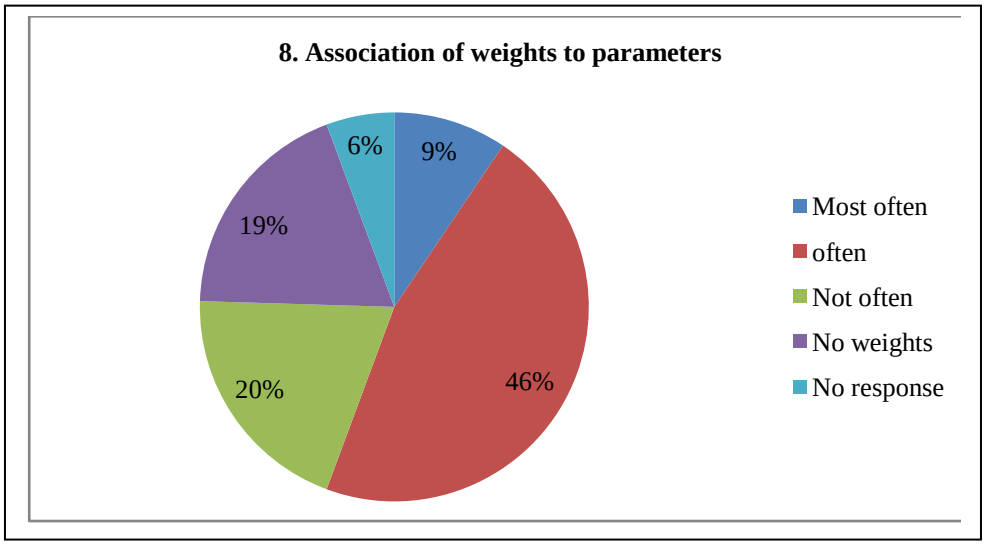

Fig. 6.17. Weights Association Preference

6.4 Summary

It is evident from the study that all players -Marketing team: for customer needs and business value, Executive Management: for strategic direction, Development Team: for efforts estimation and technology impacts, Maintenance team: for customer impacts and customer change requests are involved in prioritization of requirements for the releases. Product teams carry out the evaluation of requirements. 80% of responses indicate to Ranking by value proposition for prioritization. Ranking by value, resource availability, time availability is indicated by 66% of the responses. 70% of the participants consider priority grouping is sufficient for prioritization. 60% consider ranking and numerical assignment sufficient and 56% consider Cost-Value ratio sufficient for prioritization.

Relative importance – how much more – is considered important by 82% participants, while the methods that provide this information – AHP and Cost-Value based on AHP are not known to be used in practice.

Responses to the usage of 5 parameters relevant in requirements prioritization – Business Value(BV), Availability of Resources(AR), Time to Market(TM), Difficulty of Implementation(DI), and Impacts on Customers/Core(IC) – indicated to 61% using at least 3 parameters for prioritization. BV is considered by 70+ participants of the 106 responses. In addition to what parameters are used for prioritization, order of usage of parameters is analyzed. BV,TM,IC,AR,DI comes out to be the preferred order with 34% responses and BV,DI,AR,TM,IC is preferred by 21%. About 55% responses indicated usage of weighted parameters for prioritization.

The study and analysis indicate a need to focus on relevant factors influencing requirements prioritization for planning releases. The methods generally used - relative ranking and grouping into "must have, good to have, need not have" and cost-value - prioritize based on overall importance or aggregated cost and value. Inappropriate requirements prioritization often resulted in teams working under pressure, extended release dates, dropped features.

The purpose of getting a set of requirements implemented for the next release (time bound) is to maximize the business value of the release for the most valued customers. A strict ordering of requirements may not be the need. Need is more for a near optimal sets of requirements. The study brings out the factors – Business Value(BV), Availability of Resources(AR), Time to Market(TM), Difficulty of Implementation(DI) and Impact on existing Customers(IC) relevant to requirements prioritization. The preference for order of considering these factors for prioritization enables a multistage decision framework for prioritization. This analysis has paved way for defining 5-stage framework encompassing the parameters and weights to different parameters. The framework is discussed in Chapter 8. The data trends with respect to large, medium, small organizations are presented in the table 6.7 below.

Table 6.7. Data trends from Section III

Section III	Factor/ Size	Large	Medium	Small	Total
1. Your organization collects requirements through	Marketing team, Executive Direction, Development Team, Customer Change Requests	55%	48%	47%	51%
	Marketing team, Executive Direction, Customer Change Requests	6%	10%	27%	10%
	Executive Direction, Development Team, Customer Change Requests	25%	23%	27%	25%
	Marketing team, Development Team, Customer Change Requests	10%	18%	0%	11%
	no response	4%	3%	0%	3%
2. Requirements Analysis/assessment is done by	Business Dev, product management team	8%	3%	0%	5%
	Planning, dev teams	16%	33%	67%	29%
	Planning, stakeholders, dev, pre sales	31%	30%	13%	28%
	stake holders , dev	41%	30%	20%	34%
	no response	4%	5%	0%	4%
3. Set of requirements for next/successive release is planned by	Ranking by Value proposition, Resource, time availability	53%	50%	53%	52%
	Ranking by Value proposition	25%	25%	33%	26%
	Resource, time availability	18%	20%	13%	18%
	No response	4%	5%	0%	4%
4. Requirements evaluating/ prioritizing is done by	Using a Framework, Product Team discussions, Executive Direction	14%	13%	13%	13%

	Product team discussions, Executive Direction	31%	28%	33%	30%
	Product Management, Client discussions	2%	13%	0%	6%
	Product team discussions	47%	40%	53%	45%
	no response	6%	8%	0%	6%
5. Changes in requirements during the release are managed by	Extending release date	10%	13%	20%	12%
	Removal/ addition of some requirements	35%	23%	33%	30%
	Reprioritization /Extending release date	31%	33%	33%	32%
	Removal/ addition of some requirements, Extending release date	14%	25%	7%	17%
	no response	10%	8%	7%	8%
6. The following parameters are considered for requirements prioritization	All five - BV,AR,TM,DI,IC	25%	28%	33%	27%
	4 of five	24%	10%	7%	16%
	3 of five	10%	28%	20%	18%
	2 of five	8%	5%	20%	8%
	one of five	22%	25%	20%	23%
	No response	12%	5%	0%	8%
7. Order of parameters considered for requirements prioritization	BV,TM,IC,AR,DI	37%	33%	20%	33%
	BV,DI,AR,TM,IC	12%	25%	53%	23%
	AR,TM,IC,BV,DI	2%	5%	0%	3%
	BV,IC,DI,TM,AR	2%	3%	0%	2%
	BV, IC/DI/TM/AR	8%	10%	0%	8%
	TM/IC/DI, BV/AR	4%	8%	0%	5%

	TM,DI,IC,AR,BV	14%	10%	0%	10%
	No Response	22%	8%	27%	17%
8. Weights are associated with parameters considered for prioritization	Most often	14%	5%	7%	9%
	often	43%	48%	53%	46%
	Not often	22%	18%	20%	20%
	No weights	16%	23%	20%	19%
	No response	6%	8%	0%	6%
9. A multi stage prioritization scheme is useful for requirements prioritization	Most Often	37%	45%	53%	42%
	Always	24%	18%	7%	19%
	Not Often	22%	23%	27%	23%
	Not used/never	8%	10%	7%	8%
	No Response	10%	5%	7%	8%
10. Working out prioritization exactly for each requirement for product releases	Most useful	29%	35%	40%	33%
	often useful	53%	30%	47%	43%
	not useful often	10%	25%	13%	16%
	not useful	2%	8%	0%	4%
	no response	6%	3%	0%	4%
11. Change in prioritization during release scheme necessitates	Complete rework of prioritization	16%	18%	13%	16%
	minor changes to existing list	49%	53%	73%	54%
	Release date extension	20%	13%	7%	15%
	no change	6%	13%	7%	8%
	no response	10%	5%	0%	7%

12. Classifying requirements in to " 1. Must have 2. Good to have 3. Can live without " groups for product release is	Not sufficient at all	4%	5%	0%	4%
	Not sufficient often	20%	18%	13%	18%
	Sufficient always	22%	35%	7%	25%
	Sufficient often	45%	38%	80%	47%
	No response	10%	5%	0%	7%
13. Prioritizing requirements using Analytical Hierarchy Process (AHP) for product release is	complex/time taking/accurate	33%	28%	53%	34%
	Not used AHP	41%	58%	40%	47%
	simple	8%	10%	7%	8%
	no response	18%	5%	0%	10%
14. When number of requirements to be handled is large (>20), AHP is	complex/time taking/accurate	31%	23%	40%	29%
	Not used AHP	43%	65%	47%	52%
	simple	6%	8%	7%	7%
	no response	20%	5%	7%	12%
15. It is essential to know how much important each requirement is when compared to other for prioritization	Most often	39%	45%	33%	41%
	Always	43%	35%	53%	42%
	Not always	8%	15%	7%	10%
	No response	10%	5%	7%	8%
16. It is sufficient to know relative importance of requirements for prioritization rather than "how much more important"	Most often	43%	40%	47%	42%

	Always	25%	28%	20%	25%
	Not always	18%	23%	27%	21%
	never	4%	3%	7%	4%
	No response	10%	8%	0%	8%
17. Cost – Value ratio for requirements is the best indicator of priority	Most often	39%	35%	67%	42%
	Always	18%	13%	7%	14%
	Not always	25%	40%	27%	31%
	never	4%	5%	0%	4%
	No response	14%	8%	0%	9%
18. Ranking of requirements(in sequence of priority) based on a parameter is sufficient for prioritization	Most often	41%	48%	53%	45%
	Always	16%	15%	13%	15%
	Not always	27%	30%	33%	29%
	never	4%	3%	0%	3%
	No response	12%	5%	0%	8%
19. Numerical assignment of priority (grouping by assigning priority 1,2,3,..) to requirements is sufficient	Most often	53%	45%	53%	50%
	Always	14%	8%	13%	11%
	Not always	25%	43%	33%	33%
	No response	8%	5%	0%	6%
20. Requirements Prioritization provides traceability along the Product life cycle for improved Quality of the Product.	Most often	43%	50%	27%	43%
	Always	35%	25%	40%	32%

	Not always	12%	20%	27%	17%
	never	4%	0%	7%	3%
	No response	6%	5%	0%	5%

Next chapter discusses the comparison of various aspects related to the data and analysis across three datasets grouped on size of organization from the data gathered for understanding variance and consistency of different factors influence on requirements prioritization as the size of organization varies.

7. Analysis of Requirements Prioritization across Large, Medium, Small organizations

7.1 Introduction

The data gathered based on the study provided a unique opportunity to analyze the influence of size of organization on nature of products handled, problem areas faced, solutions adapted. Requirements handling and prioritization varied on certain parameters and remained consistent on certain aspects. The data has been grouped into three sets based on the size of the organization – large, medium, small and parameters influencing requirements prioritization has been examined across data sets. The analysis is presented in this chapter.

7.2 The data

Section I data for large organizations presented in Fig. 7.1, depicts the nature of work carried out by large organizations – As indicated by the responses, 82% handle multiple product lines which indicates to the complexity of the development carried out. 35% of participants worked on mature products of more than 10 years life span and another 35% worked on products of maturity between 5 to 10 y. This indicates to the amount of impact analysis that will be involved for any change to be implemented. 77% worked on Engineering domains that require thorough analysis and knowledge of domains. 47% worked with release cycles of a year or greater and 35% worked with half yearly or quarterly release cycles. 48% indicated to use of all three processes – Waterfall, Iterative, Agile. 45% and 43% work on 3 tier and 2 tier software respectively indicating to involved nature of development. 90% adapted to multiple platforms including web and cloud. 70% work on global products highlighting the needs of multiple languages translation, distributed requirements collection.

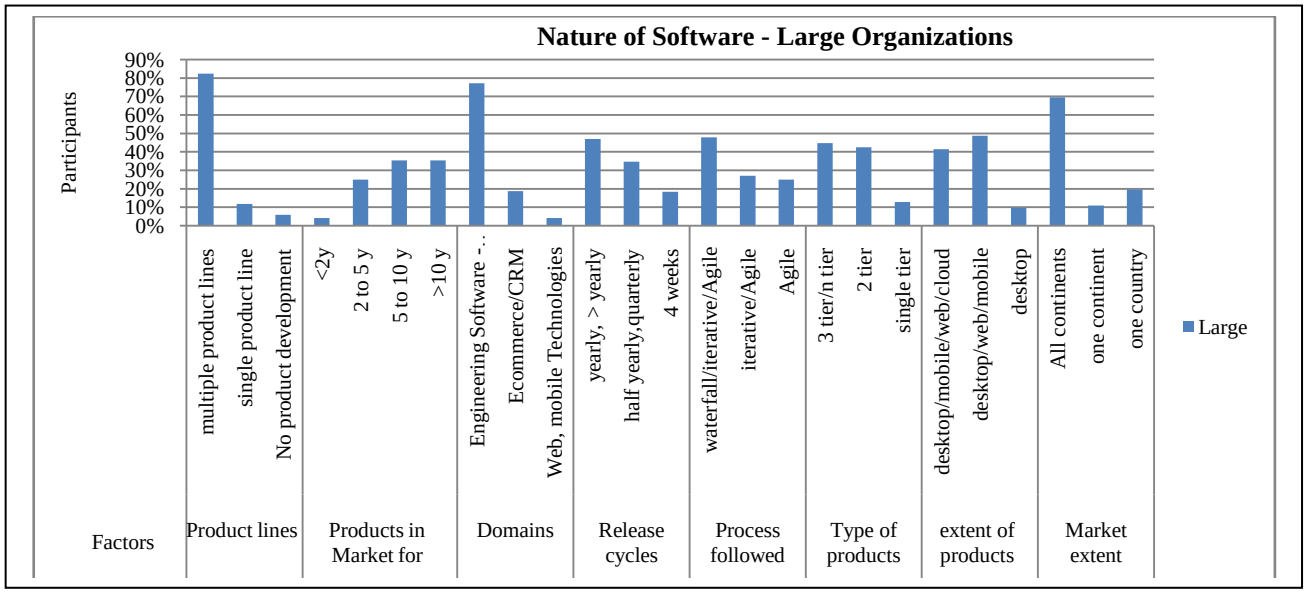

Fig. 7.1. Nature of Software – Large Organizations

Coming to Medium organizations – 55% handle multiple product lines, where as 35% handle single product line software indicating reduced need of analysis when understanding requirements as compared to multiple product lines. 28% work on less than 2 years old software and 50% work on less than 5 years software and only 23% of participants worked on mature products of more than 5 years life span. This indicates to fast changing requirements and need for quick implementation . 40% worked on web, mobile technologies keeping pace with changing technologies, 28% worked on Ecommerce – indicating emerging areas adapting software where requirements clarity may be less. 33% worked on Engineering domains requiring high quality. 49% worked with 4 weeks release cycles and 34% worked with half yearly or quarterly release cycles indicating the need to release software incrementally and quickly. 67% indicated to use Agile process to facilitate quick releases. 49% and 41% work on 3 tier and 2 tier software respectively indicating to involved nature of development as in large organizations. 92% adapted to multiple platforms including web and cloud similar to large organizations. 50% work on global products, where as 33% work on products that are used in single country reducing the need to global requirements analysis. The data is presented in Fig 7.2.

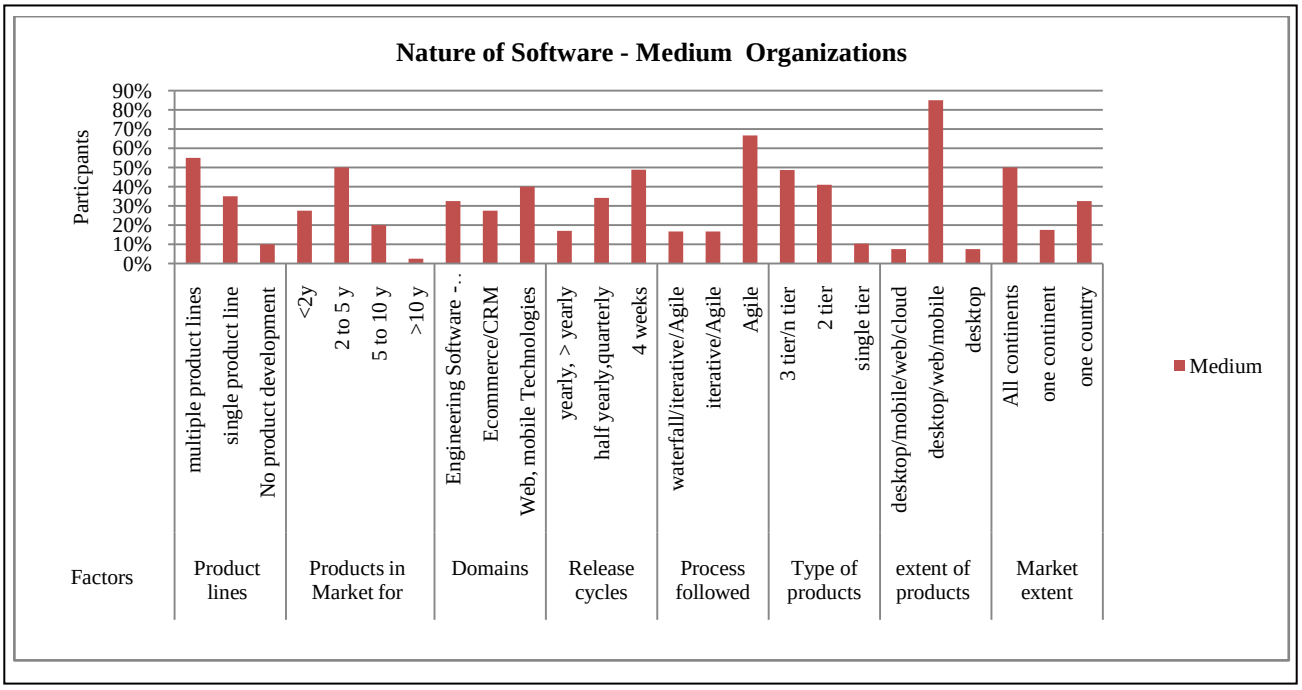

Fig. 7.2. Nature of Software – Medium Organizations

Coming to Small organizations – 27% handle multiple product lines, where as 67% handle single product line software indicating reduced need of analysis when understanding requirements as compared to multiple product lines. 47% work on less than 2 years old software and 33% work on less than 5 years software and 20% of participants worked on mature products of more than 10 years life span. This indicates to fast changing requirements and need for quick implementation. 21% worked on web, mobile technologies, 25% worked on Ecommerce, where as 53% worked on Engineering Software indicating support functioning to large organizations. 33% worked with 4 weeks release cycles and 36% worked with half yearly or quarterly and 31% worked with releases cycles of more than a year confirming support work to large organizations. 57% and 14% work on 3 tier and 2 tier software respectively and 29% work on single tier software. 92% adapted to multiple platforms including web and cloud similar to large organizations. 50% work on global

products, where as 33% work on products that are used in single country reducing the need to global requirements analysis. The data is presented in Fig. 7.3. Comparison is presented in Fig 7.4.

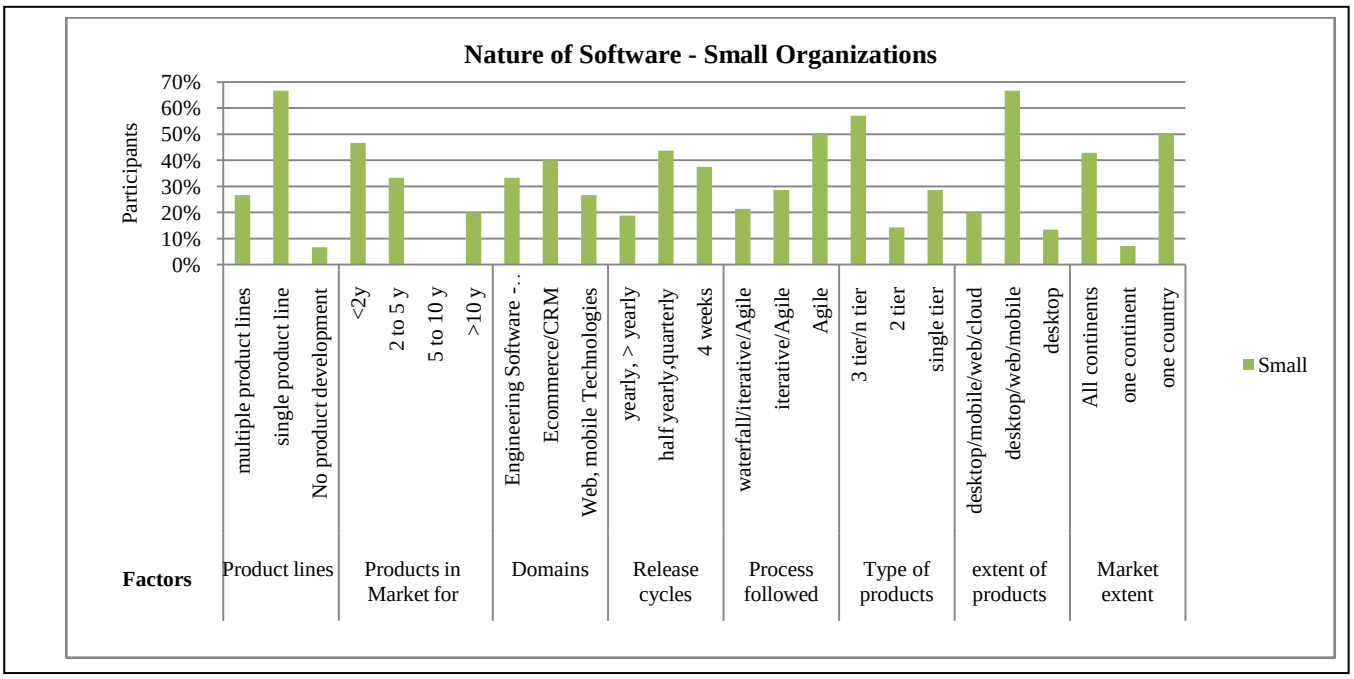

Fig. 7.3.. Nature of Software – Small Organizations

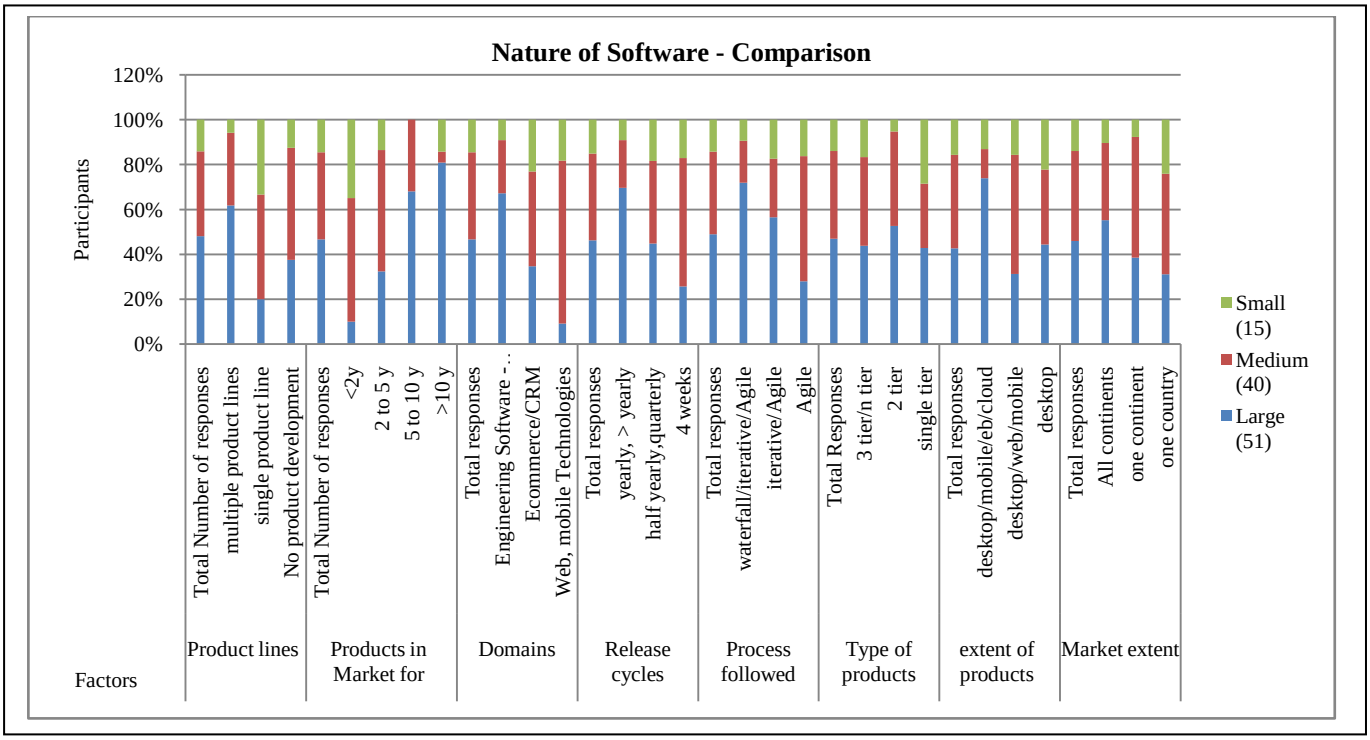

Fig. 7.4. Nature of Software – Comparison – Large, Medium, Small

7.3 Requirements Selection

This section analyses data on requirements Selection. Current requirements selection criteria in large organizations appears to be systematic with customer needs, picking up from ordered backlog and based on business analysis playing important role with 25%, 20% and 27% responses respectively. Medium organizations too appear to take up business analysis with 25% responses while 40% responses indicate to customer needs predominance. 60% of small organizations responses indicate dependence only on customer needs for requirements selection. Requirements Selection criteria with Large, Medium, Small organizations is depicted in Fig. 7.5.

Estimation of required resources and time and prioritization with respect to time and complexity are the problem areas with large organizations, lack of requirements clarity plays equal role in medium and small organizations. This can be attributed to the nature of work as discussed in the section I. Problem areas are indicated in Fig. 7.6. Solutions to the problems with current process of feature selection indicate to over work, extended releases, and attempts to convince clients.

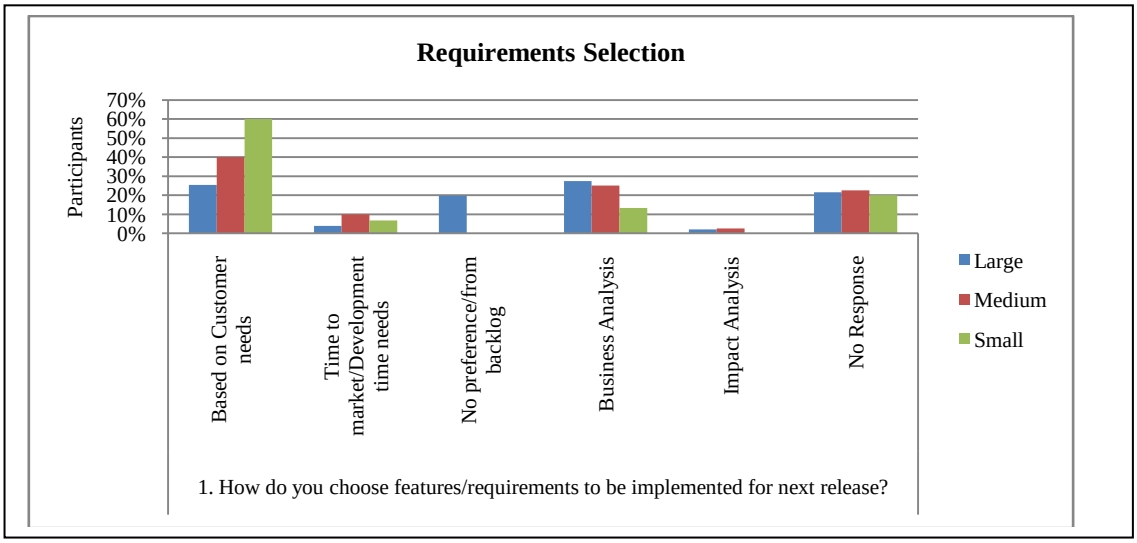

Fig. 7.5. Requirements Selection

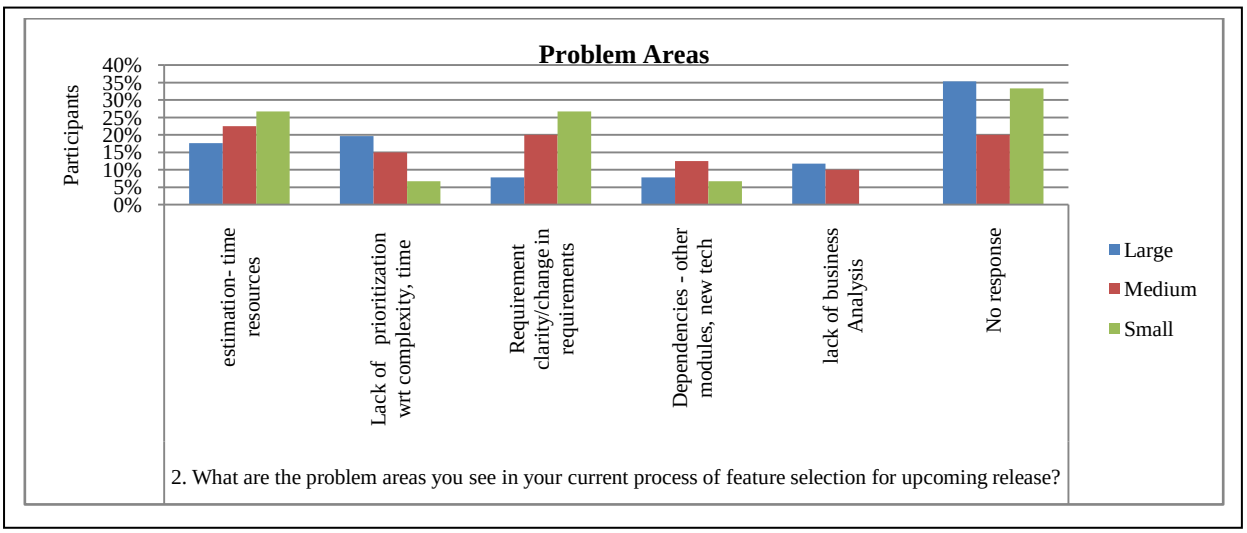

Fig. 7.6 Problem Areas

Large organizations are able to resolve problem areas with discussions among stake holders and further analysis and extensions of release dates. Extended hours and hard work are higher with medium organizations, where as client management dominant with small organizations. Fig. 7.7 demonstrates these trends.

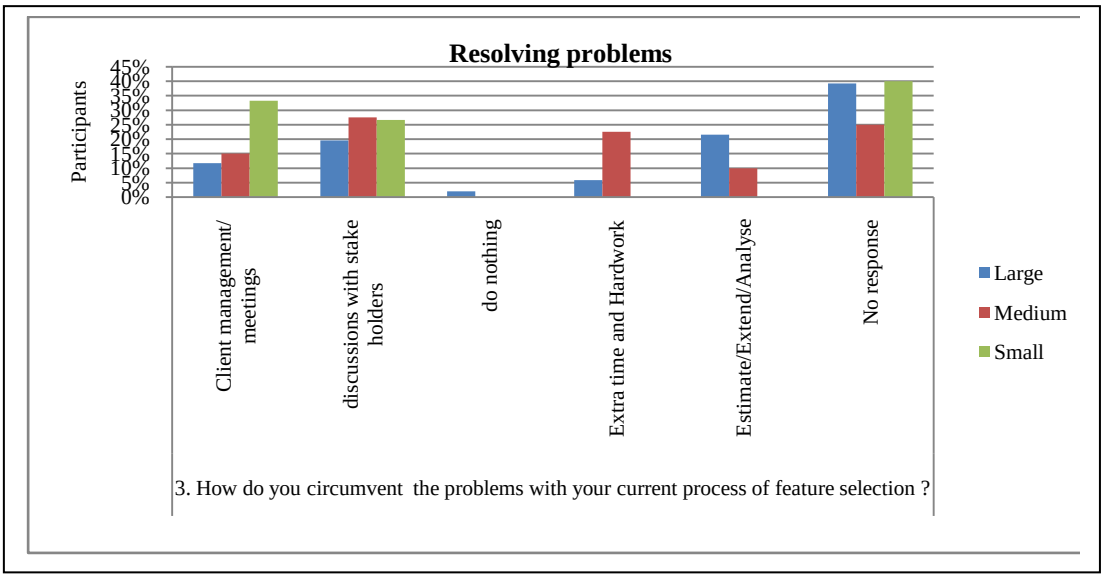

Fig. 7.7.Resolving Problems

7.4 Requirements prioritization

Large, Medium, Small organizations typically perceive Priority Grouping, Ranking, Numerical assignment, Cost-Value method being sufficient for requirements Prioritization. This can be seen from the similar responses across in Fig 7.8.

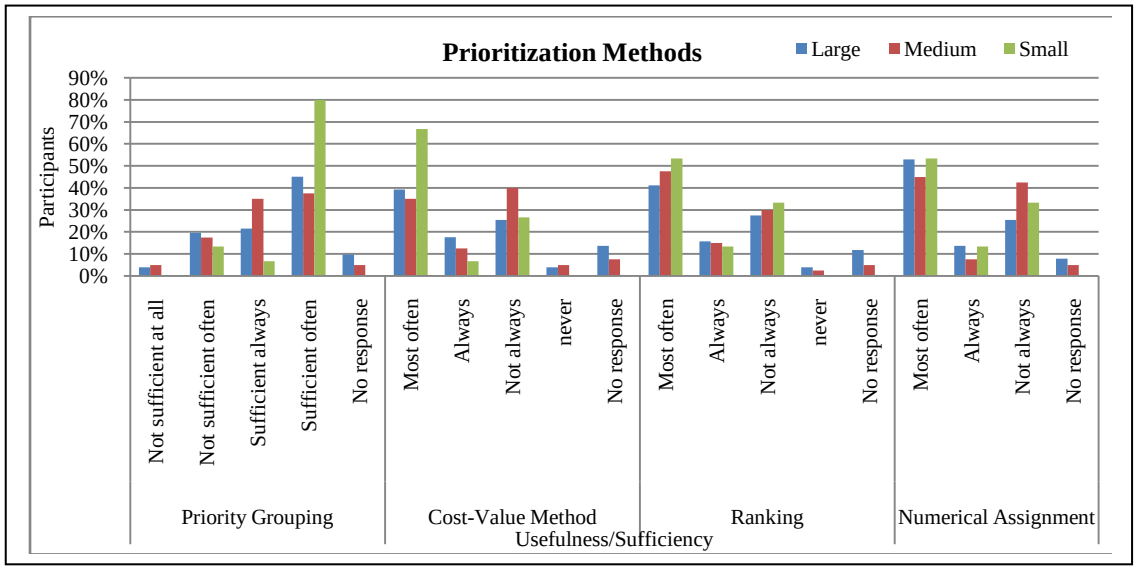

Fig. 7.8. Prioritization Methods

Large Organizations considered multiple parameters for requirements prioritization 25% of the responses indicated use of all five parameters - Business Value (BV), Availability of Resources (AR), Time to Market, Difficulty of Implementation (DI), Impact on Core(IC).24%

of them used four parameters. 22% of them used only one parameter. Medium organization has similar pattern with 28% using all 5 parameters and another 28% using 3 parameters and 25% using one parameter. Small organization also considered all 5 parameters up to 33% and 20% each considered three, two, one parameter. The use of multiple parameters for prioritization indicates to the need of multi layered prioritization approach. Business Value of the requirement appears to be the most used parameter. Fig 7.9 shows the extent to which these parameters are considered for prioritization of requirements.

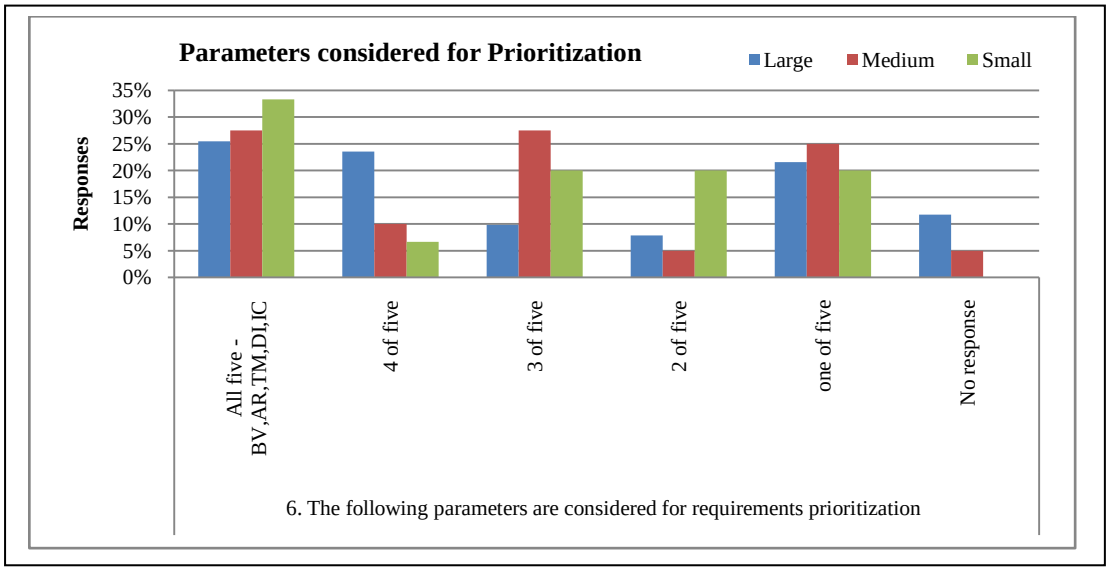

Fig. 7.9 .Factors for Prioritization

Looking at the preference of Order of using the above parameters for prioritizing 37% from Large organizations, 33% from Medium and 20% from small organizations preferred Business Value (BV), Time to Market (TM), Impacts on existing Customers (IC), Availability of Resources (AR), and difficulty of Implementation (DI) in that order. 12% of Large organizations, 25% of Medium and 53% Small organizations preferred the order - BV-DI-AR-TM-IC. Small companies focus on Difficulty of Implementation as the second important factor after Business value contrasts with Large organizations' focus on Time to Market as the second most important factor. Fig. 7.10 shows the order preferences across organizations.

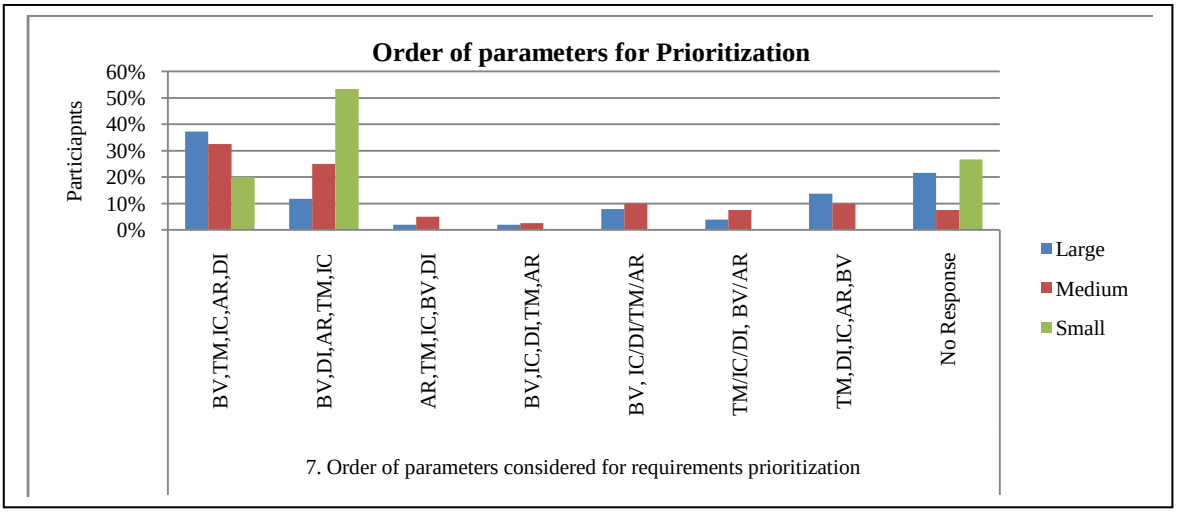

Fig. 7.10.Order of parameters for Prioritization

7.5 Influence of size of Organization - Summary

Requirements prioritization is an important part of the software development. Nature of Software development influences how requirements prioritization is taken up in the organizations. Different aspects – domains, maturity of the products, release cycles, spread of usage, complexity of the software influence how requirements prioritization is handled. The study highlighted the association of the size of the organization to these aspects in section I. While customer needs influence prioritization with small and medium organizations, large organizations tend to deal prioritization systematically with business analysis, backlog planning and stakeholder discussions. Requirement clarity appears to weigh more as a problem area for medium and small organizations. Prioritization methods perceived sufficient preferred across are –Numerical Assignment, Relative ranking, Priority grouping and cost-value. Inappropriate requirement prioritization often resulted in teams working under pressure, extended release dates, dropped features. Large organizations circumvented problems by further analysis, estimation and extension of release dates, while small and medium organizations either managed the clients through discussions or put in additional work.

Business Value (BV), Availability of Resources (AR), Time to Market(TM), Difficulty of Implementation (DI) and Impact on existing Customers(IC) are utilized for prioritization by small, medium, and large organizations for requirements prioritization. While not all five parameters are utilized by all, at least 3 of them are used by more than 70% across. BV being the most and first used factor for prioritization, large organizations focused on TM as second factor, while small organizations considered DI as second factor for prioritization mostly. BV-TM-IC-AR-DI emerged as the preferred order of considering these factors for prioritization for large organizations, whereas small organizations preferred order is BV-DI-TM-AR-IC. The study and analysis indicate the need for multistage framework for requirements prioritization. Tables 4.11, 5.10. 6.7 present the data spread and trends across large, medium, small organizations. Complete data is presented in Appendix A.

In the next chapter , a new framework - ABC Framework is suggested based on the 5 parameters as a multistage decision framework for simple and effective prioritization at multiple levels enabling implicit weights application for relevant parameters for the requirements, which enables flexible planning through the development cycle. The framework provides visualization for the changes in requirements during the release cycle and acts as an easy communicator to the involved stakeholders including testing team members.

8. A New Framework for Requirements prioritization

8.1 Introduction

The study across 61 organizations developing software products brought out the importance of considering different factors relevant in requirements prioritization and their order as relevant to the industrial process of development of products. Prioritization of requirements based on parameters relevant to the product development a priori and during the development cycle facilitates stable and predictable deliveries with less resource allocation uncertainties.

This chapter discusses the proposed ABC Framework reflecting the practical aspects of the software development. The proposed framework takes into account different parameters, elicited from the study – Business Value, Time to Market, Difficulty of Implementation, Availability of Resources and Impact on Customers/Core. The Framework considers the course of software development and links the prioritization to development process, release planning, change management, quality management.

8.2 Considerations and Framework for prioritization of requirements

Analysis of the study responses indicates a need for focus on requirements prioritization for planning releases systematically, with controlled changes during the course of release cycle. The methods being used appear to be relative ranking and grouping into - must have, good to have, need not have. Utilization of weighted parameters for requirements prioritization, adopting multi level prioritization find a place in practice, though not by all. Lack of appropriate requirement prioritization methods, process often appears to have resulted in teams working under pressure, extended release dates, dropped features.

The study covered large companies with mature products releasing successive versions of products with longer release cycle, as well as medium size to small companies working on specific project based product versions with less maturity and shorter duration release cycles. Across this range of organizations, requirements analysis and prioritization for products/projects first versions as well as successive versions is an area that needed attention and systematic methods to be adopted for stable, successful and smooth deliverables in a predictable manner. Taking the nature of products/projects and the process prior to development as constraints, requirements prioritization for the purpose of predictable releases of products is analyzed. Baseline considerations and the new framework defined are discussed in the following sections.

8.2.1 Proposed Framework Considerations

The following assumptions are in order with the premise of the framework proposed -

- The purpose of getting a set of requirements implemented for the next release(time bound) is to maximize the business value of the release.
- A strict ordering of requirements may not be the need. Need is more for a near optimal set of requirements.
- Activities on requirements do not start in serial order, but in parallel in a distributed way.

Three most important factors for determining the release requirements set are listed below and are elaborated in subsequent sections.

- Realizable Business Value (BV)
- Cost of implementation
- Constraints

8.2.1.1 Realizable Business Value

Realizable Business Value is determined based on following understanding -

- Inputs from sales/marketing/executive management/product dev/test/maintenance teams
- A Requirement may satisfy multiple customers
- Specific segments -High Value/medium Value/low Value realizable customers
- Realizable over short/medium/long durations
- Reduces test/maintenance cost
- Marginal development cost
- Opportunity costs

Business value depends on different industries and life cycle stage(new, growing, mature, declining) of the products. It is possible to have many parameters, weighing factors, analysis done to arrive at business value. Methods from any non-software product features' business value determination can be adopted.

Customer Base can be the current or existing customer base requesting for additional features or it could be the new customers that are likely to get added given a set of features implemented.

8.2.1.2 Cost of Implementation

Cost of Implementation needs to take into account the following major factors -

- Nature of requirements - Core model changes/Business Logic changes/UI changes.
- Marginal cost – base model exists/ incremental changes needed to implement the requirements
- Cross impacts & verifications costs of implementing a new requirement.
- Resources – availability of development resources
- Opportunity costs – due to non implementation of other requirements

8.2.1.3 Constraints

Constraints - Since a release is always timed to meet customers expected needs , the following Constraints need to be considered for prioritization of requirements -

- Time /duration – minimum time required for development/ time to market
- Nature of Development needed for the requirements
- Resources – knowledgeable in domain/technology/skill
- Impacts – on existing customers
- Uncertainties – changes imminent due to expanded/extended scope
- Impacts - on existing product modules

8.2.2 Preprocessing of Requirements

It is best to do some preprocessing of the RAW Requirements in terms of

- Broad understanding of the collected requirements
- Removing duplicates
- Merging somewhat similar requirements

As Software products are mostly modularized and can be specified by modules/components, grouping requirements with respect to modules helps in determining marginal costs or values easily. Giving way for some amount of approximation is appropriate in the aggregation of many requirements and evaluating with respect to many other parameters – like time and resources required to implement the requirement, time available, compatibility with current product, feasibility of implementation, etc. , especially, in the initial stages.

8.2.3 Parameters and Considerations

Time to market(TM) parameter that indicates the time available to meet customer needs helps in deciding what requirements are feasible to implement in the time available. Time required for development implicitly puts a constraint on what requirements can be taken up for the release. Nature of Development – whether the requirements needs a user Interface on existing modules or it requires development of a new module or there is a need for changes in core or new development of core indicates to the parameter - Difficulty of Implementation(DI). Availability of Resources(AR) and Impacts on existing customer/core(IC) is covered under the 3rd and 5th constraints. Constraint 4 imposes the need for the Framework to be simple enough to incorporate changes during release cycle and the need for flexible release planning.

Based on the above considerations, a new framework – ABC Framework is defined for simple and effective prioritization at multiple levels enabling implicit weights application for relevant parameters for the requirements, which enables flexible planning through the development cycle. The framework provides visualization for the changes in requirements during the release cycle and acts as an easy communicator to the involved stakeholders including testing team members. The naming of framework as ABC framework is inspired by the classification of requirements in to A, B, C classes followed in defining the framework.

8.3 The ABC Framework

The Framework is defined as 5 sets based on most used parameters in the sequence of priority determination. Each set is defined by three classes defined by % value of the respective set parameters. Requirements are grouped in to the classes in the sets in the process of prioritization. The % bands may vary from industry to industry and organization to organization to some extent.

Prioritization setsS1 to S5 and classes/bins A, B, C within each set are described in Fig. 8.1

- **S1.** Business Value(BV) in conjunction with Customer Base (CB) – with classes -
  - A:20% of CB with 70% BV
  - B:30% of CB with 25% BV
  - C:50% of CB with 5% BV

- **S2.** Requirements Applicability with respect to product – with classes
  - A:70% UW, 30% BI, 0% CP
  - B:50% UW, 40% BI, 10% CP
  - C:30% UW, 50% BI, 20% CP

Where

- UW: User Interface/Workflow Specific/Specific Customer Set ,
- BI : Business Logic/particular Industry vertical,
- CP: Core/data model level /across the portfolios.

- **S3.** Implementation Cost–
  - A: 70% MI,25% NI,5% IR
  - B: 50% MI,40% NI,10% IR
  - C: 30% MI, 50% NI, 20%IR

Where

MI: Marginal Implementation, NI: New Implementation, IR: Impact Recovery

- **S4.** Time Requirement –
  - A: 10% L,20% M, 70% S
  - B: 15% L, 25%M,60%S
  - C: 20% L, 30% M, 50% S

Where

L: 8 to 16 person weeks , M:4 to 8 person weeks, S: 2 to 4 person weeks

- **S5.** Resource Requirement –
  - A:10%RC,20%RI,70%RT
  - B:15%RC,25%RI,60%RT
  - C:20%RC,30%RI,50%RT

Where

- RC: Resources - Core aware (6y) - domain/architecture/design/technology
- RI: Resources - Industry aware (4y) – domain/design/technology
- RT: Resources – Technology aware (2y) – Technology/Skill

Fig. 8.1. Prioritization Sets and Classes

8.3.1 Applying the Framework

The process of applying the Framework is planned to be a layered approach with the following steps-

- Step 1: Evaluate requirements with respect to set 1 and put them in A,B,C bins. Treat each bin separately.
- Step 2: Evaluate requirements in the above bins with respect to set 2 and put them in A,B,C bins. The process can continue only for the bin A also.
- Step 3: Evaluate Next with respect to set 3.
- Step 4: Arrive at time durations with respect to 4
- Step 5: Verify availability of resources with respect to set 5.

The order of preference emerges for the requirements Set through the filtering process. Not all sets may require to be used. And once the selected requirements set is arrived at each step, marginal values of benefits and costs of requirements may get modified and feeding this information back and re-evaluating bins may be required.

When all sets are used for classification, we will arrive at 243 bins of requirements. Based on the constraints and release theme, the bins can be selected in the order of preference.

The parameters analyzed from the study are correlated with the software development process during a release and the sets are not defined as mutually exclusive sets covering the parameters. Rather the sets follow the sequence of considerations during development.

Business Value, in Set 1, is considered in conjunction with customer base for the requirements. This enables implicit weighing in favor of valuable customers. Requirement Applicability of Set 2 looks through modularization aspect of ease and encompasses to some extent Difficulty of Implementation. Implementation cost looks at Difficulty of Implementation from incremental development aspect and covers Difficulty of Implementation further. Time required to develop in the set 4 explicitly indicates to whether a requirement can be accomplished within the Time to Market. Time to Market is also taken into account by the S2 and S3 to some extent. Impact Recovery covers the Impacts on existing customer and also on core. Set 5 focuses on Resources availability.

The most preferred order of parameters BV,TM,IC,AR,DI and the next preferred order BV,DI,AR, TM,IC are amalgamated in the sets of the Framework and resources are considered in set 5 to match the flow of product/solution development considerations. A sample classification template of requirements using an excel sheet is given below in Figure 8.2.

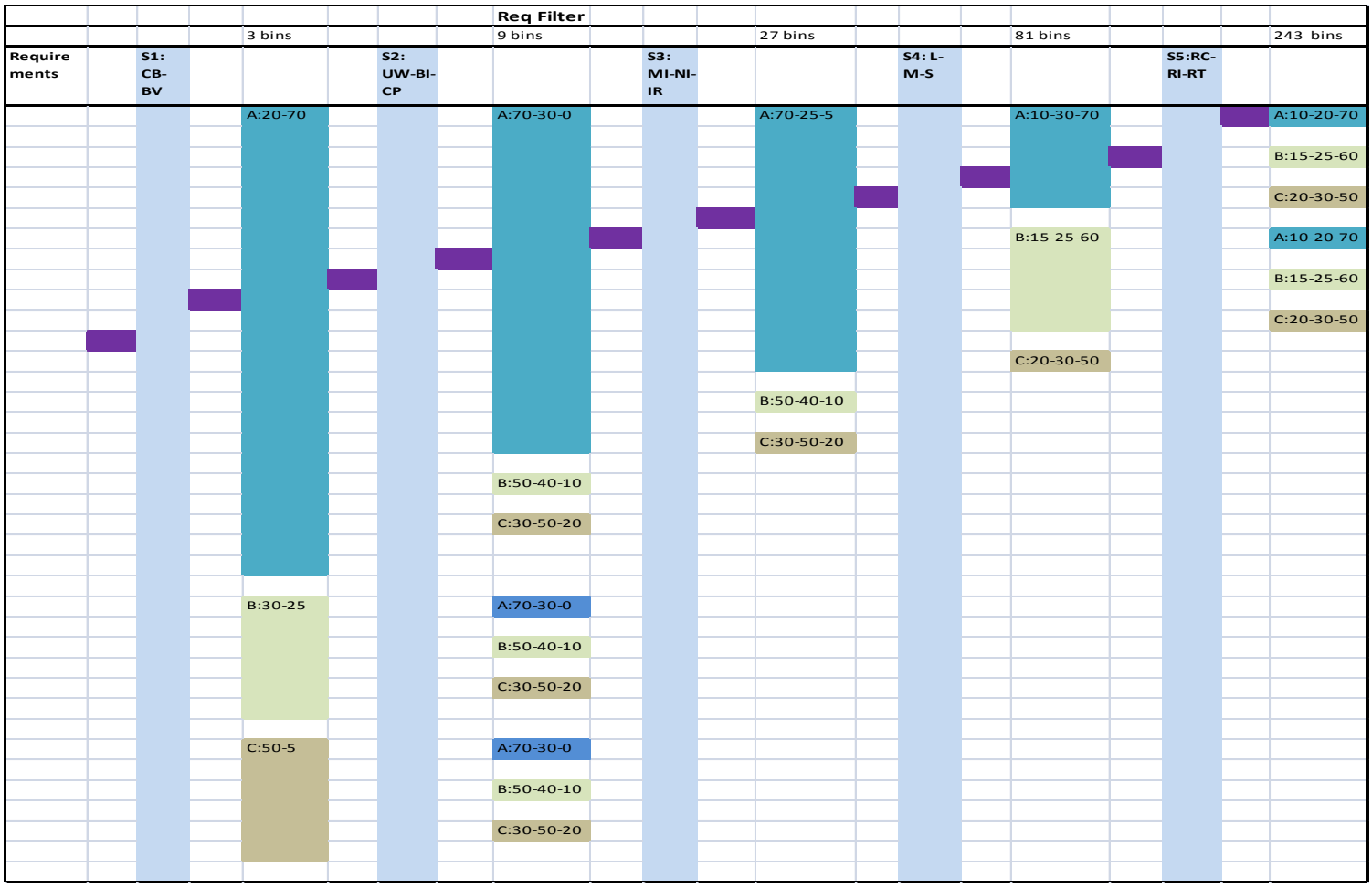

Fig. 8.2. Sample classification template

8.4 Benefits of the Framework

The framework enables simple and effective methodology for Requirements Prioritization for successive releases and leads to better understanding and planning of releases. Suitable process matching to nature of development can be identified for appropriate planning. It helps build traceability and visualize effects of plan changes and helps in informed quality planning. Benefits of the Framework are listed below.

8.4.1 Link to Development process

In addition to arriving at the prioritized requirements set, the suggested process of requirements classification can be extended to determine the development process most suitable for implementation of the requirements as described below, as no one process may be suitable for all requirements.

- Follow Agile for requirements emerging through A across Sets
- Follow Iterative for requirements emerging through AB combinations
- Follow Water fall for requirements emerging through ABC combinations.

8.4.2 Effective Quality Planning

- Requirements classification in to 243 bins enables test planning appropriately.
- Level and importance of testing a requirement can become a function of the bins.
- Nature of testing can be determined based on the bins.
- A priori information available to test teams

8.4.3 Flexible Release Planning

- Effects of Adding/removing requirements during the release cycle will be clearly visible.
- Uncertainties can be accommodated in re-planning easily.
- With the 243 bins available modular release planning becomes feasible.

8.5 Summary

Exploring the need for simple, flexible, scalable, easy to use, amenable to practical software development, a new Framework – ABC Framework is defined in this chapter. Various considerations and parameters taken into account, as a result of research study, for defining the framework are discussed. The framework's multilayered approach with significant parameters considered at each level is detailed. Taking the complexity of multiple parameters in to account and using the simplicity of classifying in to three classes, the framework offers an easy and scalable method for requirements prioritization. The framework's practical application is explained through a simple excel sheet. Advantages of the framework are highlighted. The framework enables simple and effective methodology for Requirements Prioritization for successive releases and leads to better understanding and planning of releases. It helps build traceability and visualize effects of plan changes and helps in informed quality planning. In the next chapter, 2 mathematical models are proposed for the framework usage.

Chapter 9

9. Mathematical Models for ABC Framework

9.1 Introduction

ABC Framework proposed in this research for requirements prioritization takes in to account different aspects encountered in the product development flow in a structured and in a sequence of layers. Requirements prioritization is invariably linked to cost of development and benefit to be achieved in most of the methods proposed for prioritization. In general the cost factor is considered to the extent of time taken to develop or resources cost. Business value is normally understood to the extent of immediate revenue. Considering the "other than software world" projects and cost and benefit analysis done for taking up projects – Business value encompasses present value of future returns, indirect benefits, return on investment periods. The costs involve not just development costs, but also opportunity costs and impact costs.

Typically Software requirements prioritization does not start or stop at one time or in one step. The prioritization of what will finally get into the product release goes through levels of decision making considering different aspects. Trying to club all the aspects into one or two parameters or trying to prioritize at one time considering all aspects generally results in suboptimal or not so well understood prioritization. Don Reinertsen [91] proposed Weighted Shortest Job First, which talks of the economic value in the product development process flow. This model takes into consideration business value, time criticality, risk reduction, future value for determining cost of delay. The model considers job size or time to develop and proposes a ratio of cost of delay to job size as the single weight for prioritization of requirements. Nevertheless, this model comes closest to the ABC Framework in considering Business value aspects for requirements prioritization.

In order to understand- What is to be made available in the next release, How to manage the requirements under expanding client needs, cost and time implications, what set of requirements implementation will increase revenues - a layered approach using ABC Framework paves way. It helps in prioritization of requirements and planning releases, streamlining the project deliveries to client's satisfaction without overworking the teams or missing time to market deadlines.

The framework proposed is conceptualized based on enterprise products' development experience. It enables practical use through its tabular format. This chapter provides mathematical modeling through formation of sets for representing each layer of the framework and association of weights at each layer based on the class, the requirement belongs to, for arriving at a combination of weights for each requirement.

A Unique numbering scheme is proposed for easy interpretation and visualization of the parameters of the prioritization and basis of prioritization in the paper.

This chapter describes the ABC framework briefly in section 2. Interpretation through sets is presented in section 3. Unique weights numbering scheme is presented in section 4.

9.2 ABC framework for prioritization of requirements

The purpose of getting a set of requirements implemented for the next release (time bound) is to maximize the business value of the release for the most valued customers. A strict ordering of requirements may not be the need. Need is more for a near optimal sets of requirements. Since a release is always timed to meet customers expected needs, the following constraints are considered for prioritization of requirements –

1. Time /duration – minimum time required for development
2. Nature of development needed for the requirements.
3. Resources – knowledgeable in domain/technology/skill
4. Uncertainties – changes due to expanded/extended scope
5. Impacts on existing customers and existing product modules

Based on the above considerations, ABC Framework enables simple and effective prioritization at multiple levels enabling implicit weights application for relevant parameters for the requirements, which enables flexible planning through the development cycle. The framework provides visualization for the changes in requirements during the release cycle and acts as an easy communicator to the involved stakeholders including testing team members.

The Framework is defined as 5 sets based on most used parameters in the sequence of priority determination. Each set is defined by three classes/bins defined by % value of the respective set parameters. Requirements are grouped into the classes in the sets in the process of prioritization. The % bands may vary from industry to industry and organization to organization to some extent.

Prioritization sets – S1 to S5 and classes/bins – A, B, C within are described in Table 9.1 below.

Table 9.1. Framework – Sets, Classes

Sets	**Classes/Bins – A,B, C**
S1. Business Value(BV) in conjunction with Customer Base (CB)	A: 20% of CB with 70% BV B: 30% of CB with 25% BV C: 50% of CB with 5% BV

S2. Requirements Applicability with respect to product, where UW: User Interface, BI: Business Logic, CP: Core	A: 70% UW, 30% BI, 0% CP B: 50% UW, 40% BI, 10% CP C: 30% UW, 50% BI, 20% CP
S3. Implementation Cost, where MI: Marginal Implementation, NI: New Implementation, IR: Impact Recovery.	A: 70% MI, 25% NI, 5% IR B: 50% MI, 40% NI, 10% IR C: 30% MI, 50% NI, 20%IR
S4. Time Requirement, where L: 8 to 16 person weeks, M: 4 to 8 person weeks, S: 2 to 4 person weeks	A: 10% L,20% M, 70% S B: 15% L, 25%M,60%S C: 20% L, 30% M, 50% S
S5. Resource Requirement, where RC: Core aware, RI: Industry aware, RT: Technology aware	A: 10%RC, 20%RI, 70%RT B: 15%RC, 25%RI, 60%RT C:20%RC, 30%RI, 50%RT

The Framework is applied in a layered approach through the sets. The order of preference emerges for the requirements Set through the filtering process. Not all sets may be required to be used. When all sets are used for classification, we will arrive at 243 bins of requirements. Based on the constraints and release theme, the bins can be selected in the order of preference for the releases.

9.3 Interpretation through Sets for the ABC framework

Business Value encompasses Value to customer now and repeat value to customer, value to other customers and value possible through being used as a platform component in other products. The assumption is- there is at least one customer for each requirement and all requirements have equal business value of unit 1, if not specified. R1 to R3 are feature level requirements. Table 9.2 provides sample classification in to classes A, B, C for set 1. The table has CB and BV normalized to Total CB and BV, for ease of classification.

Table 9.2. Sample classification - Set 1

Requirements	Normalized CB (CB/∑CB)	Normalized BV in descending order (BV/∑BV)	Class A CB (0..0.2) BV (1..0.7)	Class B CB (0.2.. 0.5) BV(0.05...0.25)	Class C CB(0.5..1) BV(0..0.05)
R1	0.1	0.7	R1		
R2	0.3	0.2		R2	
R3	0.6	0.1			R3

Classifying R1, R2 into class A, class B respectively maps to the default framework suggestions of class boundaries for Customer base and Business Value. R3 is classified in to class C with slightly adjusted class boundaries to suit the requirements at hand.

With the possibility of requirements being incremental on top of an existing product or with development utilizing some of the proprietary frameworks or open-source frameworks or the development involving totally new product from scratch, we can consider each feature will have partly User Interface or data input and output forms, partly business logic implementation for processing the data, industry/ vertical/ domain specific and partly core data model/architecture development. Table 9.3 below details the membership association for set 2. Normalizing UW, BI, CP to Total Effort (TE) allows classification into A, B, C classes based on the normalized values ranges given in the table. Here the effort can be considered in terms of code base to be developed or person weeks or story points required to complete the activities related to UI, BI or CP.

Table 9.3. Membership Association - Set 2

Requirements	UW	BI	CP	Class A UW(0.7.. 1) BI (0..0.3) CP(0)	Class B UW(0.5..0.7) BI (0.3..0.4) CP(0..0.1)	Class C UW(0.3..0.5) BI (0.4..0.5) CP(0.1..0.2)
R1	0.8	0.2	0	R1		
R2	0.5	0.5	0		R2	
R3	0.2	0.5	0.3			R3

In the sample classification above, R1 is classified as per default class boundaries, whereas R2, R3 required a slight adjustment to the class boundaries. The class boundaries can be tuned to the nature of projects and type of development.

Table 9.4 below details the membership association for set 3, which considers whether the feature requires entirely new implementation or marginal implementation is sufficient and if there is going to be impact on existing features and customers due to new requirements. Normalizing MI, NI, IR to Total Effort (TE) allows classification into A, B, C classes based on the normalized values ranges given in the table. Here the effort can be considered in terms of code base to be developed or person weeks or story points required to complete the activities related to UI, BI or CP.

Table 9.4. Membership Assocation - Set 3

Requirements	MI	NI	IR	Class A MI(0.7.. 1) NI (0..0.25) IR(0..0.05)	Class B MI(0.5..0.7) NI (0.25..0.4) IR(0.05..0.1)	Class C MI(0.3..0.5) NI (0.4..0.5) IR(0.1..0.2)
R1	0.6	0.3	0.1		R1	
R2	0.2	0.5	0.3			R2
R3	0.4	0.5	0.1			R3

R2 is classified into Class C with Impact Recovery beyond 20% and marginal implementation being less than 30%.

Set 4 considers the overall effort required to develop the feature or requirement and decides on classes based on the duration – large, Medium or Small as defined in the framework or as practical for a particular organization - required to complete a requirement. In order for a requirement to be completed, some parts of the requirement would need long duration – large and some parts can be completed in shorter duration, while others may take medium durations. The extent of each of these durations influences the classes association. Table 9.5 below describes the associations for set 4. The durations are normalized to Total duration in the table for the classes association.

Table 9.5. Membership Association - Set 4

Requirements	L	M	S	Class A L(0... 0.1) M (0..0.20) S(0.7..1)	Class B L(0.1..0.15) M (0.20..0.25) S(0.6..0.7)	Class C L(0.15..0.2) M (0.25..0.3) S(0.5..0.6)
R1	0	0.3	0.7	R1		
R2	0.2	0.3	0.5			R2
R3	0.1	0.2	0.6		R3	

R2, R3 map to framework default boundaries and R1 is put in class A with Medium class boundary adjustment, based on effort required under Large.

Set 5, the last layer focuses on right resources requirement in order to develop the requirement within the constraints. Here the knowledge needs of resources are emphasized. Table 9.6 provides the classes association for set 5 with resource needs normalized to total resource requirements.

Table 9.6. Membership Assocaition - Set 5

Requirements	RC	RI	RT	Class A RC(0... 0.1) RI (0..0.20) RT(0.7..1)	Class B RC(0.1..0.15) RI (0.20..0.25) RT(0.6..1.0)	Class C RC(0.15..0.2) RI (0.25..0.3) RT(0.5..1.0)
R1	0.1	0.2	0.7	R1		
R2	0.15	0.25	0.6		R2	
R3	0.2	0.3	0.5			R3

Set 5 classification of R1, R2, R3 fits in to default boundaries. Now looking at the classification across the sets S1 to S5 for the three requirements – R1, R2, R3. Assuming weights of 3/3, 2/3, 1/3 for classes A, B, C respectively, macro level priority - Pm can be arrived at for each requirement by multiplying the class weights across sets. The priority can vary from AAAAA resulting in 1 to CCCCC resulting in 0.001372, providing a range of priorities for each of the requirements. The priorities need not necessarily be unique. Same priority requirements can be grouped together for simultaneous development. Table 9.7 indicates Pm calculations for R1, R2, R3.

Table 9.7. Priority Values

Requirements	S1	S2	S3	S4	S5	Pm
R1	A	A	B	A	A	$3^4*2/3^5$ (or 0.66)
R2	B	B	C	C	B	$2^3/3^5$ (or 0.0109)
R3	C	C	C	B	C	$1^4*2/3^5$ (or 0.00274)

9.4 Unique numbering scheme for the framework

As we have seen in section 2 the framework has 5 sets – ranging from S1 to S5 with S1 being the first level and determining the Business Value for the requirement. S2 looks at the existing capabilities in terms of components, products and effort required broadly for the new requirement. S3 goes deeper with effort understanding along with impact insights. S4 attempts to get at time requirements for the job at hand for the requirement, whereas S5 assesses the capabilities in terms of resources.

Section 3 has the discussion on how the requirements can be assigned into classes. Distinct Priority is arrived at by multiplying across the sets the class weights. While a single number may be useful to look at relatively at the requirements, the intelligence of classification into classes is lost from visibility. In order to retain the class information and yet arrive at a weighted priority scheme, the following number sequence is proposed.

Assigning a five digit sequence with each set holding the positional value from S1 to S5 in that order, the sequence will be a number –

S1 S2 S3 S4 S5 - With S1 to S5 holding

S1 (10000ths place)

S2 (1000ths place)

S3 (100ths place)

S4 (10s place)

S5 (unit place)

Where each of S1, S2, S3, S4, S5 can have values - 0 or 1 or 2 based on which class of A, B, C, a requirement falls into.

Each position having three values – 0, 1, 2 and with 5 positions of value, the number of sequences equals to 35, that is 243 sequences.

A requirement falling into class A across sets S1 to S5 will have a sequence 00000.

A requirement falling into class B across sets S1 to S5 will have a sequence11111.

A requirement falling into class C across sets S1 to S5 will have a sequence 22222.

All the 243 values of sequence will range from 00000 to 22222, with each value in each position representing the class and set the requirement belongs to. This enables immediate interpretation of the priority with respect the requirements associated Business value, resources availability, time requirements, cost implications.

While the requirements are being prioritized not all sequences need be used or get used. When new requirements come in to picture during the development cycle, it is easy to insert the requirements once the sequence is determined for the requirement. More than one requirement can

have the same sequence and it becomes easy to group the requirements instantly. Table 9.8 shows sample sequences for requirements R1 to R4.

Table 9.8. Priority Sequences

Requirement	S1	S2	S3	S4	S5
R1	0	0	0	0	0
R2	1	0	0	1	2
R3	2	1	0	2	1
R4	1	2	1	1	1

A new requirement with a priority sequence of 20100 can be placed above R3 and below R2 instantly. A requirement with a priority sequence 10012 can be placed along R2 forming a group, indicating same priority sequence.

9.5 Summary

The ABC framework defined as part of the research work provides a unique representation for prioritization of the requirements. The framework enables understanding and interpreting prioritization in a visual and instant way. In order to understand and utilize the framework in practice, two new innovative mathematical representations are proposed in this research. One based on sets and associations of requirements into the classes within sets and the other based on unique number sequence representation. Both the methods are described in this chapter. The methods simplicity for changes in requirements prioritization is demonstrated. The Framework and both the methods proposed enable simple and effective methodology for Requirements Prioritization for successive releases under dynamic changes and lead to better understanding and planning of releases. Both Methods help build traceability and visualize effects of plan changes and help in informed quality planning.

Comparison of AHP Framework with AHP, Cost-Value, Weiner, Priority grouping methods is discussed in the next chapter.

10. Comparison of ABC Framework with Other Methods

10.1 Introduction

As discussed in chapter 2, Significant Research and empirical studies have taken place in the area of requirements prioritization. Methods have evolved for prioritizing requirements based on different parameters - Value and Cost being prominent among them. Analytical Hierarchy Process- AHP is based on pair wise comparison of requirements relative to each other on a scale at successive levels of hierarchy.

Cost-Value approach by Karlsson takes the cost of implementation and value of requirements in to consideration in pair wise comparison. Wiegers method proposes risk weighted cost/value ratio for determining priority. Priority Groups method categorizes requirements based on ranking different parameters – mostly importance of requirements and are put in groups.

Davis advises simplifying the process and advises Triage at successive levels, taking into account market realities. Industry specific studies for software products meeting certain specific base parameters seem to have been very few. This makes the conclusions and comparisons difficult to be applicable or reliable. Comparison of some of the methods for quality requirements is taken up by Karlsson.

ABC Framework proposed in this research reflects the practical aspects of the software development. The proposed framework takes into account different parameters considered during the course of software development and links the prioritization to development process, release planning, change management, quality management. This chapter looks at Priority grouping, Cost-Value method, Wiegers method and AHP and in comparison analyses the benefits of ABC Framework.

Brief description of the methods – AHP, Cost-Value, Priority Grouping, Wiegers method, with their computational aspects elaborated, is provided in Section 2. Section 3 discusses ABC Framework computational aspects. Comparison basis and merits are discussed in Section 4. Comparison is summarized in Section 5.

10.2 Requirements Prioritization Methods for Comparison

10.2.1 AHP

Analytical Hierarchy Process (AHP) of Saaty is a multi criteria decision making approach in which factors are arranged in a hierarchical structure that flows from overall goal to criteria to sub criteria and alternatives in successive levels. Hierarchy is expected to provide overview of the problem space and enable decision maker compare homogeneous elements in each level. As

illustrated by Karlsson using AHP for decision making involves 4 steps for evaluating requirements using the criterion of value. A scale as defined by Saaty is used for pair wise comparison of the requirements - 1,3,5,7,9 corresponding to equal value, slightly more value, strong value, very strong value and extreme value respectively. 2,4,6,8 provide intermediate values when compromise is needed. In pair wise comparisons, reciprocal of assigned number of one requirement becomes the priority for the pair's other requirement.

For each criterion AHP's pair wise comparisons result in n (n-1)/2 comparisons for n requirements. Assuming 4 requirements R1, R2, R3, R4, Step 1 involves forming 4X4 matrix for pair wise comparison. Step 2 involves comparing each requirement with other one using the scale values. Step 3 involves deriving the priority matrix, which are Eigen values of the matrix arrived at by using averaging over normalized columns. Relative value is assigned to requirements based on the priority. Continuing with Karlsson's illustration, the following matrix indicates pair wise comparison, priorities and relative values of requirements as shown in Table 10.1.

Table 10.1. Pair wise comparison

	R1	R2	R3	R4	Eigen value	Relative value
R1	1	1/3	2	4	.26	26%
R2	3	1	5	3	.50	50%
R3	1/2	1/5	1	1/3	.09	9%
R4	1/4	1/3	3	1	.16	16%

For 4 requirements and one criteria there will be 4*3/2 = 6 comparisons that will be needed. If the number of criteria is 2, the number of comparisons will be 2 * 6 = 12. For n requirements and c criteria the comparisons will be c * n (n-1)/2. Then a step to correlate or combine the priorities across the criteria for a combined priority for each requirement needs to be arrived at. There is a scale and estimating relative importance for each requirement in comparison with another one in the set is done. With different criteria at different levels, relative estimation on these criteria is required.

10.2.2 Cost Value

Karlsson and Ryan proposed using implementation Cost and Value as the high level factors for requirements' pair-wise comparison as in AHP. Both Cost and Value based relative priorities for the requirements are arrived at as illustrated above and are plotted in a cost-value diagram, which can be used as a conceptual map for identifying requirements to be taken up for implementation. This information can also be utilized for strategizing release plan, according to Karlsson and Ryan. Here c is 2, hence the comparisons required for the 4 requirements will be 2*6 = 12. For n requirements the comparisons will be 2* n (n-1)/2 = n (n-1).

10.2.3 Priority grouping

In this method requirements are not compared to each other based on a criteria, but are grouped into either three – low, medium, high priority groups/essential/conditional/optional groups or four – most needed, good to have, ok to have and not to have - priority groups based on importance of requirements. Each group can further be grouped within to arrive at finer clusters of requirements.

And this sub-classification can extend and form a hierarchy of levels. Whether the criteria at each level will be importance, which can be a combination of different criteria pre-determined or the criteria can be different for sub-grouping is not explicitly discussed in literature.

Taking the same 4 requirements, the number of decisions to be made will be 4 – to decide which group the requirement will go to, for a single level grouping. For n requirements the decisions will be n. If successive grouping is done, the decisions would be n*c for c number of successive groupings. The decision making in classifying into groups is subjective in this method.

10.2.4 Wiegers Method

Wiegers semi quantitative, analytical approach distributes a set of estimated priorities across a continuum rather than grouping them into a few priority levels. Risk adjusted value/cost ratio is used to determine priority in this method. A features attractiveness is directly proportional to the value it provides and inversely proportional to its cost and technical risk of implementation. Weiger suggests applying this method to only negotiable features and not to core business functions or requirements that require compliance with Government regulations. Priority is calculated as value% /(cost% *cost weight + risk% * risk weight), where value is a weighted combination of value to customer and penalty of not implementing the requirement.

Since there are 4 criteria – value, penalty, cost, risk, to be estimated on a scale of 1 to 9, for 4 requirements, we will need 4 * 4 = 16 decisions to be made at the initial level. For n requirements, the decisions needed are n * 4. The requirements can be analyzed at subsequent levels for increased granularity. For c levels, the decisions required would be n * c * 4. Weiger indicate the method is not mathematically rigorous and is limited by the ability to estimate the 4 parameters for each requirement and suggests it should be used as a guideline to make trade-off decisions But this is the same limitation for all the methods using a scale to estimate on different criteria. Wieger points that the method can become unwieldy beyond several dozens of requirements and suggests initial and sub-lists analysis for ease of prioritization.

In this method Value includes the –ve value or penalty for not implementing. Cost is expected to take into account existing modules benefit, risk includes impacts.

10.2.5 ABC framework.

The Framework as described in chapter 7 is defined as 5 sets based on most used parameters in the sequence of priority determination. Each set is defined by three classes/bins defined by % value of the respective set parameters. Requirements are grouped into the classes in the sets in the process of prioritization. The % bands may vary from industry to industry and organization to organization to some extent.

Prioritization sets – S1 to S5 and classes/bins – A, B, C within are described briefly in Table 10.2 below.

Table 10.2. Framework - Sets, Classes

Sets	**Classes/Bins - A,B, C**
S1. Business Value(BV) in conjunction with Customer Base (CB)	A: 20% of CB with 70% BV B: 30% of CB with 25% BV C: 50% of CB with 5% BV
S2. Requirements Applicability with respect to product, where UW: User Interface, BI: Business Logic, CP: Core	A: 70% UW, 30% BI, 0% CP B: 50% UW, 40% BI, 10% CP C: 30% UW, 50% BI, 20% CP
S3. Implementation Cost, where MI: Marginal Implementation, NI: New Implementation, IR: Impact Recovery.	A: 70% MI, 25% NI, 5% IR B: 50% MI, 40% NI, 10% IR C: 30% MI, 50% NI, 20%IR
S4. Time Requirement, where L: 8 to 16 person weeks, M: 4 to 8 person weeks, S: 2 to 4 person weeks	A: 10% L,20% M, 70% S B: 15% L, 25%M,60%S C: 20% L, 30% M, 50% S
S5. Resource Requirement, where RC: Core aware, RI: Industry aware, RT: Technology aware	A: 10%RC, 20%RI, 70%RT B: 15%RC, 25%RI, 60%RT C:20%RC, 30%RI, 50%RT

The Framework is applied in a layered approach through the sets. The order of preference emerges for the requirements Set through the filtering process. Not all sets may be required to be used. When all sets are used for classification, we will arrive at 243 bins of requirements. Based on the constraints and release theme, the bins can be selected in the order of preference for the releases. Requirements can be associated with their class membership at each level and a macro priority can be associated as well by associating weights to classes at each level and/or weights to each of the sets]. With the unique numbering scheme, priority sequences can be generated for the requirements, based on class association in each set which help in visualizing basis of prioritization through the development process and visualizing requirements change implications. Table 10.3, 10.4, 10.5 illustrate the macro priorities and number sequences based on ABC framework for 3 requirements.

Table 10.3. Priority Values with Class weights (A- 3/3, B-2/3, and C-1/3)

Requirements	S1	S2	S3	S4	S5	Pm
R1	A	A	B	A	A	$3^4*2/3^5$
						(or 0.66)
R2	B	B	C	C	B	$2^3/3^5$
						(or 0.0109)
R3	C	C	C	B	C	$1^4*2/3^5$
						(or 0.00274)

Table 10.4. Priority Values with Class and Set weights

Requirements	S1-5/5	S2-4/5	S3-3/5	S4-2/5	S5-1/5	Pm
R1	A	A	B	A	A	$(5*4*3*2*1/5^5)*3^4*2/3^5$
						(or 0.0256)
R2	B	B	C	C	B	$(5*4*3*2*1/5^5)*2^3/3^5$
						(or 0.001264)
R3	C	C	C	B	C	$(5*4*3*2*1/5^5)*1^4*2/3^5$
						(or 0.000316)

Table 10.5. Unique Priority Sequences

Requirement	S1	S2	S3	S4	S5	Priority Sequence
R1	0	0	0	0	0	00000
R2	1	0	0	1	2	10012
R3	2	1	0	2	1	21021
R4	1	2	1	1	1	12111

10.3 ABC Framework comparison with other methods

For the 4 requirements ABC framework would require 4 * 5 = 20 decisions to be made, with all 5 sets utilized. For each set the number of decisions is same as in priority grouping that is 4. For n requirements the number of decisions will be n* 5.

ABC Framework adapts to the idea of hierarchical structure of layers of AHP relevant to the problem space of software product development for requirements prioritization... The framework takes in to account different aspects – business value, nature of implementation, and cost of implementation, including impacts, time needs and resource needs- encountered in the product

development flow in a structured way and in a sequence of layers. The class boundaries are defined for intuitive decision making, and are adaptable to specific projects. The criteria encompass short term and long term benefit, cost aspects. Requirements prioritization is invariably linked to cost of development and benefit to be achieved in most of the methods proposed for prioritization. In general the cost factor is considered to the extent of time taken to develop or resources cost. Business value is normally understood to the extent of immediate revenue. Wieger included penalty of not implementing in value. Karlsson's cost –value are to be estimated a priori. Considering the "other than software world" projects and cost and benefit analysis done for taking up projects – Business value encompasses present value of future returns, indirect benefits, return on investment periods. The costs involve not just development costs, but also opportunity costs and impact costs. Wieger included impact costs in risk parameter.

ABC Framework does not pick up the AHP's scale or method of priority calculation. Typically Software requirements prioritization does not start or stop at one time or in one step. The prioritization of what will finally get into the product release goes through levels of decision making considering different aspects. Trying to club all the aspects into one or two parameters or trying to prioritize at one time considering all aspects generally results in suboptimal or not so well understood prioritization. The uncertainties in the input decision making related to determination of values of criteria or related to relative comparison, the author feels mathematical rigor is not warranted for determination of priorities. The classification is more akin to priority grouping at each level. ABC framework can be mapped to priority grouping with different criteria adopted at each level of hierarchy, which are not necessarily sub groups.

In Priority grouping, the grouping of high, medium, low is a subjective judgment. Same is the case with AHP scale, where scale values for comparison are subjective; ABC Framework attempts to define boundaries of subjective decision making, based on problem space of software development. The boundaries are adjustable as per the specific needs of a project. The criteria at each level in the ABC framework are intuitively defined based on practical aspects of software development. The criteria are not mutually exclusive strictly; they reflect the parameters considered as software development progresses.

In cost value method of Karlsson or in value-penalty-cost-risk method of Weigner, the various aspects of software development are expected to be resulting in cost of development, value of requirement, so that decisions can be made on prioritization in terms pair-wise comparison or weighted grouping. ABC framework enables grouping into 3 classes at successive levels based on different criteria faced by the decision makers, without imposing a pair-wise comparison or estimation on a scale, yet resulting in the final outcome of relative priorities.

The framework enables visualization of relative prioritization of requirements at every level and in the final prioritization, instead of criteria getting lost in a mere prioritization number as in other methods. There is implicit cost and implicit value in each of the criteria and there are short term costs and values and long term costs and values with respect to each criteria and determining these is not a formalized science for requirements prioritization so far. Unlike in non-software industry, where project costs and project revenues are determined over projects life periods taking into account present and future revenue flows and costs to be incurred and opportunity costs. Software industry is still seen to be not amenable to this rigorous analysis.

ABC framework has criteria at successive levels which spawn out the development process and attempts to capture cost and value aspects implicitly. The decisions are to be taken based on the

boundary values for the classes, which allows flexibility, adoption, approximation. It enables visualization of short term costs and value and also long term costs and value by virtue of the criteria and classes at successive levels, albeit implicitly through the process and in final prioritization.

Prioritization is somewhat misconstrued concept in software development. It simply means what requirements can be picked up for now for a certain set of customers to provide a solution within a certain time period with the available resources and existing inventory (components/modules). And this scenario is subject to change. Under the changing scenario, it will be imperative to change the development course and it is needed to have as less impact as possible. How do we reconcile the changes to the current decisions on priorities of requirements? What were the parameters considered in the past and how do they change now? Visualization, ease of re-prioritization, impacts visibility on schedules, costs, value are needed. ABC framework provides ease of reprioritization [10], visibility to impacts of change, flexibility for re-planning, which is difficult with other methods.

Requirements are requirements and they need to be implemented at sometime or the other, they need to be spaced out and this spacing out needs to be visible all the time for dynamic decision making, or dynamic choices. The decision map and the criteria of decisions, nature of decisions needs to be visualized throughout the life cycle of the product/project. This is feasible with ABC framework through its unique representation of priorities and unique classification at successive levels with relevant criteria into A, B, C classes whose boundaries are predetermined.

Coming to scalability of the methods, methods based on pair-wise comparison – AHP and Cost-Value tend to be increasingly cumbersome. Weiger indicates to the unwieldiness of the method for large number of requirements due to estimation needs. Priority grouping is still the simplest and easiest, though approximate. ABC framework can be easily used for large number of requirements and number of decision grow only linearly with the number of requirements.

10.4 Summary of comparison

Summarizing the comparative analysis in section 3, ABC Framework offers the ease of Priority grouping method adopts the hierarchical decision making concept of AHP, takes into account different aspects of practical relevance in software development space, which, in effect, are common with cost-value-penalty-risk. Any dynamic changes in priorities of requirements can be easily integrated, visualized and interpreted in ABC framework. The impacts on release plans and coming up with new release plans is similarly simple with ABC framework. Comparison of various aspects of the prioritization methods discussed in section 3 is presented in Table 10.6.

Table 10.6. Comparison of various aspects

Method	AHP	Cost-Value	Weiger	Priority grouping	ABC Framework
Methodology	Pair-wise comparison	Pair-wise comparison	Independent assessment by estimation	Independent assessment	Independent assessment
Criteria	Importance . Can have multiple	Cost, Value	Value, Penalty, Cost, Risk	Importance. Can have multiple criteria	Business Value, Nature of requirement,

	criteria				Implementation costs, Development time , Resources
Scale	1,3,5,7,9 2,4,6,8 reciprocals of above	Same as AHP	1(low) to 9(high)	Grouping into 3 or 4 groups	Classifying into three classes in each set
Levels	As needed for other criteria	As needed for granularity	As needed for granularity	As needed for granularity	5
Number of decisions for n requirements	n(n-1)/2 for each criteria/ level	n(n-1) = 2 * n(n-1)/2	4n for single level	n for single level	5n
Priority representation	Eigen values of comparison matrix	Eigen values	Value% / (cost% *weight + risk% *weight)	Group membership/ranking	Class membership in each set
Visualization of influencing factors in final priority	Relative priority	Cost-value diagram	Relative priority	Ranking in group.	Class/set association sequence
Changes incorporation	Rework the process	Rework the process	Rework the process	Can be added/removed as needed	Can be added/removed as needed
Visualization of change impacts	-	-	-	-	Relative Class sequence, macro priority
Release plan determination, changes in release plan visualization	based on relative priority	Based on cost-value diagram/correlation	Based on relative priority	Based on ranking	Based on release theme relevant class/set sequences

10.5 Summary

ABC framework can be seen as a hierarchy of levels with different criteria representing the software product development space that can be used to classify requirements similar to simple

priority grouping method and taking into account cost and value and risk aspects as in cost-value and Wiegner's methods. In addition, it provides a unique representation for prioritization of the requirements. The framework enables understanding and interpreting prioritization in a visual and instant way. The Framework and priority representation enables simple and effective methodology for Requirements Prioritization for successive releases under dynamic changes and lead to better understanding and planning of releases.

It helps in prioritization of requirements and planning releases, streamlining the project deliveries to client's satisfaction without overworking the teams or missing time to market deadlines, providing dynamic prioritization throughout the process of software development. Advantages of ABC Framework over other methods are listed below.

Relates closely to the Software Development problem space and handles prioritization not as an isolated activity, but as an integrated release planning activity, unlike other methods.

Takes into account parameters relevant to software development process.

Prioritization is handled at 5 levels, reflecting decision making process of prioritization throughout software development process, with flexibility of using less number of levels optionally.

Simple 3 classes' decision making process is effectively used to generate 243 priority groups, which is sufficient to handle large number of requirements, circumventing the scaling problem of other methods.

Considers the uncertain, approximate information on prioritization and does not attempt to attribute preciseness to the priorities. Rather allows final prioritization to emerge easily out of this imprecise information on prioritization.

Provides method – unique numbering scheme - to represent prioritization with visibility to parameters considered

Allows easy re-planning under dynamic changes in prioritization during release cycle and helps view multiple options visually, while other methods need reprioritization a fresh and do not offer visibility into planning of releases.

Research conclusion and Future Scope is laid out in the next chapter.

11. Conclusions and Future Scope

11.1 Conclusions

The research presented in this thesis attempted to understand research gaps in requirements prioritization and propose a solution to bridge the gap. Data on prioritization of requirements in practice is gathered from 106 participants from 61 organizations, in addition to consolidated knowledge form the literature survey. Qualitative analysis of the data led to improved understanding of the parameters influencing requirements prioritization. With the improved understanding of the parameters influencing the prioritization of requirements, a new and innovative framework is proposed for requirements prioritization for Software products development, as a solution to bridge the gap between research and practice. Two new methods are developed for applying the framework practically. Applying the framework and its advantages are demonstrated. The framework encompasses parameters considered in industry and adopts classification into three classes across 5 layers of relevance for product development. Two new schemes of representation and visualization of prioritization based on different parameters are arrived at as part of the research.

The research focused on the factors relevant in requirements prioritization for the software products building and continuing to meet customers' needs. Relevant factors are identified through a study conducted with a questionnaire prepared based on industry experience. Qualitative analysis is carried out grouping the parameters to reflect relevant areas in product development. Analysis across datasets formed based on the size of organization is carried out to understand size and associated parameters impacts on Requirements prioritization. The goals of the research effort in this thesis -to provide effective and simple methods to visualise and prioritize requirements for software products development undergoing continuous changes and releases -.are accomplished by defining the framework and devising innovative mathematical models for using the framework.

The thesis provided improved understanding of requirements prioritization in the context of off-the-shelf products and custom made products, by analyzing qualitatively the factors effecting prioritization of requirements. A case study is conducted to analyse factors associated with requirements impacting releases.

The new framework designed to help in requirements prioritization is based on grouping requirements into 3 classes across 5 levels to reflect the practical development process and parameters. Three different ways are suggested to apply practically the framework. First method represents the framework in the form of sets and requirements association in the sets. Second one has a unique number representation scheme to allow visible interpretation of the various factors influence on the requirements prioritization. The third one enables simple use of excel work sheets to capture classification of requirements based on parameters of importance. Comparison of the framework with four of the generally used requirements prioritization methods brought out the advantages of the proposed framework.

11.2 Research Outcome - Meeting Objectives of the Research

Objective 1: To study the factors that influence requirements prioritization and elicit information on order of preference of using these factors.

Outcome: Objective 1 has been accomplished through study of parameters as discussed and analyzed in chapters5, 6, 7 across 61 Organizations with 106 participants. Analysis across 3 data sets provided additional confidence in bringing out different factors influencing requirements prioritization.

Objective 2: To compare and analyze data for large medium and small software organizations.

Data has been grouped according to size of organization – large, medium, small and data has been compared for understanding the effect of scale of organization on different factors influencing requirement prioritization. The analysis is presented in Chapter 7.

Objective 3: To propose a new framework to enable simple and effective methodology for Requirements Prioritization for successive releases.

Objective 3 has been achieved by proposing a framework – ABC framework – a multi level decision making framework, taking into account parameters of relevance for practical software development with the understanding from the study carried out.

Objective 4: To Formulate Mathematical models for practical usage of proposed Framework.

Two new innovative schemes have been presented for representing priorities and application under changes in priorities. Excel sheets based method has been suggested for applying the framework, in addition.

Objective 5: To Compare of the ABC framework with four significant Requirement Prioritization methods.

ABC Framework proposed in this research has been compared with AHP, Cost-Value Method, Wiegers Method, Priority grouping Method on multiple aspects of ease of use, number of calculations, usage under changes, scalability.

11.3 Limitations of the Study

The research encompassed gathering data from industry on practical aspects of software development and deriving knowledge on practical issues and needs in requirements prioritization. The scope of study currently covered in one group multiple domains, multiple types of development, across regions. The study can further expanded to include domain specific studies, type of development studies and differentiate specific needs.

While a multi level decision making framework specific to software development is developed as part of this research based on the present understanding of the parameters, the framework can be further refined to suit specific domains and can be standardized industry wise. Applying for successive releases and continuous development in industry is required for confirming the advantages of the framework.

11.4 Future Scope for Research

With the framework in place and aided by the understanding of the factors influencing requirements prioritization and the importance of prioritization for release planning under constraints, carrying out case studies at different software organizations through successive releases is planned to be taken up further. The advantages of the defined framework in practice and there by the benefits to the organizations, in terms of smooth and timely, quality and complete deliveries of software, will be studied further to this research.

11.5 Papers Published in Journals as a result of Research

1. Sita Devulapalli, Akhil Khare. "A Framework for Requirement Prioritization for Software Products", IUJ Journal of Management, Vol 2, No.1, May2014, pp:35-41.

11.6 Papers Presented and Published in Conference Proceedings

2. Sita Devulapalli, Akhil Khare, ORS Rao. "Study and analysis of Requirements Prioritization for product development projects - A case study" Proceedings of National Conference on Information Technology NCAIT 2016, MVSR Engineering College, Hyderabad, pp:57-59.

3. Sita Devulapalli, Akhil Khare, ORS Rao. "Requirement Prioritization-Survey and Analysis" Proceedings of the International Congress on Information and Communication Technology ICICT 2015, Volume 2 (Springer Proceedings - AISC), pp: 567 -575.

4. Sita Devulapalli, Akhil Khare, ORS Rao. "Mathematical treatment of ABC Framework for Requirement Prioritization" Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1, Smart Innovation, Systems and Technologies 50, (Springer Proceedings 2016) , pp: 341 – 348.

5. Sita Devulapalli, Akhil Khare, ORS Rao. "Comparison of ABC Framework with AHP, Wiegers method, Cost-value, Priority groups for Requirements Prioritization", paper ID:222,COMNET 2016, International Conference on Communications and Networks, (Springer Proceedings) (to be published)

6. Sita Devulapalli, Akhil Khare, ORS Rao." Study of Factors influencing Requirements prioritization for Software Products Development", ICMIT 2016, 3rd International Conference on Contemporary issues on Management and Information technology, March 2016, Malaysia (AIRCC Journal) (to be published)

7. Sita Devulapalli, Akhil Khare, ORS Rao."Requirements Prioritization – Parameters of Relevance –An empirical Study across 3 datasets", Paper ID:134,CY4, 2016, International Conference on Cyber Security and IT Governance, (to be published in the proceedings of

Second International Conference ICTCS - 2016 by ACM - International Conference Proceedings Series (ICPS) (to be published)

8. Sita Devulapalli, Akhil Khare, ORS Rao."Study of Factors influencing Requirements Prioritization for Software Products Development" (POSTER ID: 75), National Seminar on Science & Technology for Indigenous Development in India, Feb 2016. The Indian Science Congress Association, Hyderabad.

APPENDIX A

Table A1. Section I -Data Spread and Trends across Large, Medium, Small Organizations

Factor	Option/Size	Large	%	Medium	%	Small	%	Total	%
1. The size of your organization		Large >200		Medium (25 to 200)		Small (<25)		Total	
	No. of Responses	51	48%	40	38%	15	14%	106	100%
2. The Organization has (Number of Product lines)	Multiple product lines	42	82%	22	55%	4	27%	68	64%
	Single product line	6	12%	14	35%	10	67%	30	28%
	No product development	3	6%	4	10%	1	7%	8	8%
3. What role do you play in your organization?	Business Analyst	0	0%	10	25%	0	0%	10	9%
	CEO/Director/MD/ Architect	5	10%	4	10%	3	20%	12	11%
	product owner/Project Manager/QA Manager/Lead	39	76%	22	55%	11	73%	72	68%
	Lead developer	4	8%	8	20%	0	0%	12	11%
4. You Participate in (Job Description)	Product Planning	25	49%	17	43%	6	40%	48	45%
	product development	23	45%	22	55%	9	60%	54	51%
	testing		0%		0%		0%	0	0%
	no response	3	6%	0	0%	1	7%	4	4%
5. You get involved in Requirements (Activities)	Analysis, Estimation, Implementation	12	24%	5	13%	2	13%	19	18%
	Analysis, estimation, prioritization	18	35%	14	35%	5	33%	37	35%
	Analysis, Estimation, Prioritization, Implementation	20	39%	21	53%	7	47%	48	45%
	no response	1	2%		0%	1	7%	2	2%
6. You work on Products that are in Market for (Maturity of the products)	<2y	2	4%	11	28%	7	47%	20	19%
	2 to 5 y	12	24%	20	50%	5	33%	37	35%
	5 to 10 y	17	33%	8	20%		0%	25	24%

	>10 y	17	33%	1	3%	3	20%	21	20%
7. You work on Products in the field of (Domains)	Engineering Software - CAD/GIS/Telecom/transport/System	37	73%	13	33%	5	33%	55	52%
	Ecommerce/CRM	9	18%	11	28%	6	40%	26	25%
	Web, mobile Technologies	2	4%	16	40%	4	27%	22	21%
8. Your organization releases products' next versions every (Release cycles)	yearly, > yearly	23	45%	7	18%	3	20%	33	31%
	half yearly, quarterly	17	33%	14	35%	7	47%	38	36%
	4 weeks	9	18%	20	50%	6	40%	35	33%
9. The development process Your Organization follows	waterfall/iterative/Agile	23	45%	6	15%	3	20%	32	30%
	iterative/Agile	13	25%	6	15%	4	27%	23	22%
	Agile	12	24%	24	60%	7	47%	43	41%
10. The products are (Complexity of the products)	3 tier/n tier	21	41%	19	48%	8	53%	48	45%
	2 tier	20	39%	16	40%	2	13%	38	36%
	single tier	6	12%	4	10%	4	27%	14	13%
11. The products are used in (Spread of Usage)	All continents	32	63%	20	50%	6	40%	58	55%
	one continent	5	10%	7	18%	1	7%	13	12%
	one country	9	18%	13	33%	7	47%	29	27%
12. The products can be used on (Devices and Platforms)	desktop/mobile/ web/cloud	17	33%	3	8%	3	20%	23	22%
	desktop/web/ mobile	20	39%	34	85%	10	67%	64	60%
	desktop	4	8%	3	8%	2	13%	9	8%

Table A2. Section II -Data Spread and Trends across Large, Medium, Small Organizations

Factor	Option/Size	Large		Medium		Small		Total	
	/Total responses	51	%	40	%	15	%	106	%
1. How do you choose features/requirements to be implemented for next release?	Based on Customer needs	13	25%	16	40%	9	60%	38	36%
	Time to market/Development time needs	2	4%	4	10%	1	7%	7	7%
	No preference/from backlog	10	20%	0	0%	0	0%	10	9%
	Business Analysis	14	27%	10	25%	2	13%	26	25%
	Impact Analysis	1	2%	1	3%	0	0%	2	2%
	No Response	11	22%	9	23%	3	20%	23	22%
2. What are the problem areas you see in your current process of feature selection for upcoming release?	estimation- time resources	9	18%	9	23%	4	27%	22	21%
	Lack of prioritization wrt. complexity, time	10	20%	6	15%	1	7%	17	16%
	Requirement clarity/change in requirements	4	8%	8	20%	4	27%	16	15%
	Dependencies - other modules, new tech	4	8%	5	13%	1	7%	10	9%
	lack of business Analysis	6	12%	4	10%	0	0%	10	9%
	No response	18	35%	8	20%	5	33%	31	29%
3. How do you circumvent the problems with your current process of feature selection ?	Client management/ meetings	6	12%	6	15%	5	33%	17	16%
	discussions with stake holders	10	20%	11	28%	4	27%	25	24%
	do nothing	1	2%	0	0%	0	0%	1	1%
	Extra time and Hard work	3	6%	9	23%	0	0%	12	11%
	Estimate/Extend/Analyze	11	22%	4	10%	0	0%	15	14%
	No response	20	39%	10	25%	6	40%	36	34%
4. How often do you have teams working for release under pressure and for long hours in a day?	often	12	24%	17	43%	6	40%	35	33%

	very often	11	22%	2	5%	2	13%	15	14%
	sometimes	19	37%	17	43%	4	27%	40	38%
	rarely	7	14%	3	8%	3	20%	13	12%
	no response	2	4%	1	3%	0	0%	3	3%
5. Do you have few of the team members over worked during releases?	often	13	25%	13	33%	6	40%	32	30%
	very often	8	16%	7	18%	2	13%	17	16%
	sometimes	23	45%	14	35%	5	33%	42	40%
	rarely	5	10%	5	13%	2	13%	12	11%
	no response	2	4%	1	3%	0	0%	3	3%
6. How often do you abandon features being implemented for a release and restart on new features?	often	5	10%	3	8%	2	13%	10	9%
	very often	4	8%	0	0%	0	0%	4	4%
	sometimes	20	39%	17	43%	8	53%	45	42%
	rarely	20	39%	19	48%	5	33%	44	42%
	no response	2	4%	1	3%	0	0%	3	3%
7. Do you feel the right resources availability is an issue for meeting release schedules?	often	16	31%	14	35%	5	33%	35	33%
	very often	5	10%	4	10%	2	13%	11	10%
	sometimes	18	35%	12	30%	7	47%	37	35%
	rarely	9	18%	9	23%	1	7%	19	18%
	no response	3	6%	1	3%	0	0%	4	4%
8. How often do you abandon features during release due to realized impacts on existing customers?	often	7	14%	7	18%	2	13%	16	15%
	very often	3	6%	2	5%	2	13%	7	7%
	sometimes	18	35%	10	25%	6	40%	34	32%
	rarely	21	41%	19	48%	3	20%	43	41%

	no response	2	4%	2	5%	2	13%	6	6%
9. Do you analyze the impacts on core structure /architecture/data model, of features to be implemented a priori?	often	18	35%	16	40%	6	40%	40	38%
	very often	18	35%	11	28%	5	33%	34	32%
	sometimes	9	18%	10	25%	3	20%	22	21%
	rarely	4	8%	2	5%	1	7%	7	7%
	no response	2	4%	1	3%	0	0%	3	3%
10. How often you rework your resource(time, personnel, S/W,H/W)estimates for the features during the development cycle for a release?	often	18	35%	21	53%	4	27%	43	41%
	very often	4	8%	2	5%	2	13%	8	8%
	sometimes	21	41%	12	30%	8	53%	41	39%
	rarely	6	12%	4	10%	1	7%	11	10%
	no response	2	4%	1	3%	0	0%	3	3%

Table A3. Section III -Data Spread and Trends across Large, Medium, Small Organizations

Factor	Option/Size	Large		Medium		Small		Total	
	/Total responses	51	%	40	%	15	%	106	%
1. Your organization collects requirements through	Marketing team, Executive Direction, Development Team, Customer Change Requests	28	55%	19	48%	7	47%	54	51%
	Marketing team, Executive Direction, Customer Change Requests	3	6%	4	10%	4	27%	11	10%
	Executive Direction, Development Team, Customer Change Requests	13	25%	9	23%	4	27%	26	25%
	Marketing team, Development Team, Customer Change Requests	5	10%	7	18%	0	0%	12	11%
	no response	2	4%	1	3%	0	0%	3	3%
2. Requirements Analysis/assessment is done by	Business Dev, product management team	4	8%	1	3%	0	0%	5	5%
	Planning, dev teams	8	16%	13	33%	10	67%	31	29%
	Planning, stakeholders, dev, pre sales	16	31%	12	30%	2	13%	30	28%
	stake holders , dev	21	41%	12	30%	3	20%	36	34%
	no response	2	4%	2	5%	0	0%	4	4%
3. Set of requirements for next/successive release is planned by	Ranking by Value proposition, Resource, time availability	27	53%	20	50%	8	53%	55	52%
	Ranking by Value proposition	13	25%	10	25%	5	33%	28	26%
	Resource, time availability	9	18%	8	20%	2	13%	19	18%
	No response	2	4%	2	5%	0	0%	4	4%
4. Requirements evaluating/ prioritizing is done by	Using a Framework, Product Team discussions, Executive Direction	7	14%	5	13%	2	13%	14	13%
	Product team discussions, Executive Direction	16	31%	11	28%	5	33%	32	30%
	Product Management, Client discussions	1	2%	5	13%	0	0%	6	6%
	Product team discussions	24	47%	16	40%	8	53%	48	45%
	no response	3	6%	3	8%	0	0%	6	6%

5. Changes in requirements during the release are managed by	Extending release date	5	10%	5	13%	3	20%	13	12%
	Removal/ addition of some requirements	18	35%	9	23%	5	33%	32	30%
	Reprioritization /Extending release date	16	31%	13	33%	5	33%	34	32%
	Removal/ addition of some requirements, Extending release date	7	14%	10	25%	1	7%	18	17%
	no response	5	10%	3	8%	1	7%	9	8%
6. The following parameters are considered for requirements prioritization	All five - BV,AR,TM,DI,IC	13	25%	11	28%	5	33%	29	27%
	4 of five	12	24%	4	10%	1	7%	17	16%
	3 of five	5	10%	11	28%	3	20%	19	18%
	2 of five	4	8%	2	5%	3	20%	9	8%
	one of five	11	22%	10	25%	3	20%	24	23%
	No response	6	12%	2	5%	0	0%	8	8%
7. Order of parameters considered for requirements prioritization	BV,TM,IC,AR,DI	19	37%	13	33%	3	20%	35	33%
	BV,DI,AR,TM,IC	6	12%	10	25%	8	53%	24	23%
	AR,TM,IC,BV,DI	1	2%	2	5%	0	0%	3	3%
	BV,IC,DI,TM,AR	1	2%	1	3%	0	0%	2	2%
	BV, IC/DI/TM/AR	4	8%	4	10%	0	0%	8	8%
	TM/IC/DI, BV/AR	2	4%	3	8%	0	0%	5	5%
	TM,DI,IC,AR,BV	7	14%	4	10%	0	0%	11	10%
	No Response	11	22%	3	8%	4	27%	18	17%
8. Weights are associated with parameters considered for prioritization	Most often	7	14%	2	5%	1	7%	10	9%
	often	22	43%	19	48%	8	53%	49	46%
	Not often	11	22%	7	18%	3	20%	21	20%
	No weights	8	16%	9	23%	3	20%	20	19%
	No response	3	6%	3	8%	0	0%	6	6%
9. A multi stage prioritization scheme is useful for requirements	Most Often	19	37%	18	45%	8	53%	45	42%

prioritization									
	Always	12	24%	7	18%	1	7%	20	19%
	Not Often	11	22%	9	23%	4	27%	24	23%
	Not used/never	4	8%	4	10%	1	7%	9	8%
	No Response	5	10%	2	5%	1	7%	8	8%
10. Working out prioritization exactly for each requirement for product releases	Most useful	15	29%	14	35%	6	40%	35	33%
	often useful	27	53%	12	30%	7	47%	46	43%
	not useful often	5	10%	10	25%	2	13%	17	16%
	not useful	1	2%	3	8%	0	0%	4	4%
	no response	3	6%	1	3%	0	0%	4	4%
11. Change in prioritization during release scheme necessitates	Complete rework of prioritization	8	16%	7	18%	2	13%	17	16%
	minor changes to existing list	25	49%	21	53%	11	73%	57	54%
	Release date extension	10	20%	5	13%	1	7%	16	15%
	no change	3	6%	5	13%	1	7%	9	8%
	no response	5	10%	2	5%	0	0%	7	7%
12. Classifying requirements in to " 1. Must have 2. Good to have 3. Can live without " groups for product release is	Not sufficient at all	2	4%	2	5%	0	0%	4	4%
	Not sufficient often	10	20%	7	18%	2	13%	19	18%
	Sufficient always	11	22%	14	35%	1	7%	26	25%
	Sufficient often	23	45%	15	38%	12	80%	50	47%
	No response	5	10%	2	5%	0	0%	7	7%
13. Prioritizing requirements using Analytical Hierarchy Process (AHP) for product release is	complex/time taking/accurate	17	33%	11	28%	8	53%	36	34%
	Not used AHP	21	41%	23	58%	6	40%	50	47%
	simple	4	8%	4	10%	1	7%	9	8%
	no response	9	18%	2	5%	0	0%	11	10%

14. When number of requirements to be handled is large (>20), AHP is	complex/time taking/accurate	16	31%	9	23%	6	40%	31	29%
	Not used AHP	22	43%	26	65%	7	47%	55	52%
	simple	3	6%	3	8%	1	7%	7	7%
	no response	10	20%	2	5%	1	7%	13	12%
15. It is essential to know how much important each requirement is when compared to other for prioritization	Most often	20	39%	18	45%	5	33%	43	41%
	Always	22	43%	14	35%	8	53%	44	42%
	Not always	4	8%	6	15%	1	7%	11	10%
	No response	5	10%	2	5%	1	7%	8	8%
16. It is sufficient to know relative importance of requirements for prioritization rather than "how much more important"	Most often	22	43%	16	40%	7	47%	45	42%
	Always	13	25%	11	28%	3	20%	27	25%
	Not always	9	18%	9	23%	4	27%	22	21%
	never	2	4%	1	3%	1	7%	4	4%
	No response	5	10%	3	8%	0	0%	8	8%
17. Cost – Value ratio for requirements is the best indicator of priority	Most often	20	39%	14	35%	10	67%	44	42%
	Always	9	18%	5	13%	1	7%	15	14%
	Not always	13	25%	16	40%	4	27%	33	31%
	never	2	4%	2	5%	0	0%	4	4%
	No response	7	14%	3	8%	0	0%	10	9%
18. Ranking of requirements(in sequence of priority) based on a parameter is sufficient for prioritization	Most often	21	41%	19	48%	8	53%	48	45%
	Always	8	16%	6	15%	2	13%	16	15%
	Not always	14	27%	12	30%	5	33%	31	29%
	never	2	4%	1	3%	0	0%	3	3%
	No response	6	12%	2	5%	0	0%	8	8%

19. Numerical assignment of priority (grouping by assigning priority 1,2,3,..) to requirements is sufficient	Most often	27	53%	18	45%	8	53%	53	50%
	Always	7	14%	3	8%	2	13%	12	11%
	Not always	13	25%	17	43%	5	33%	35	33%
	No response	4	8%	2	5%	0	0%	6	6%
20. Requirements Prioritization provides traceability along the Product life cycle for improved Quality of the Product.	Most often	22	43%	20	50%	4	27%	46	43%
	Always	18	35%	10	25%	6	40%	34	32%
	Not always	6	12%	8	20%	4	27%	18	17%
	never	2	4%	0	0%	1	7%	3	3%
	No response	3	6%	2	5%	0	0%	5	5%

BIBLIOGRAPHY

1 Aaqib Iqbal, Farhan M. Khan, Shahbaz A. Khan. "A Critical Analysis of Techniques for Requirement Prioritization and On Research Issues", International Journal of Reviews in Computing, 2009. www.ijric.org

2 Alan M Davis. " The Art of Requirements Triage", IEEE Computer, March 2003

3 Alan M. Davis, Ed Yourdon, Ann S. Zweig "Requirements Management Made Easy". 39-947. www.omni-vista.com

4 Amir Seyed Danesh. " Comparison and Investigation of Re-planning Methods for Software Releases" International Journal of Computer Theory and Engineering, Vol 3, No. 2, April 2011

5 Amir Seyed Danesh and Rodina Ahmad. "Software release planning challenges in software development: An empirical study "African Journal of Business Management Vol. 6(3), pp. 956- 970, 25 January, 2012. Available online at http://www.academicjournals.org/AJBM

6 Andrea Herrmann, Maya Daneva . "Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research" 16th IEEE International Requirements Engineering Conference, 2008.

7 Anirudha N Joshi. "Integration of Human-Computer Interaction Activities in Software Engineering for Usability Goals Achievement", (2011) Dept. of Computer Science, IIT, Mumbai. [PhD Thesis]

8 B. Regnell, K. Kuchcinski. "Exploring Software Product Management Decision problems with constraint solving -opportunities for prioritization and release planning", Software Product Management (IWSPM), 2011 5th International Workshop on. (IEEE)

9 Barry Boehm. "Value based Software Engineering: Overview and Agenda" Value Based Software Engineering Book, Springer Publication, 2006.

10 Bhagyashri Jawale, Ashish T. Bhole. " Adaptive Fuzzy Hierarchical Cumulative Voting: A Novel Approach Toward Requirement Prioritization, IJRET: International Journal of Research in Engineering and Technology, Vol. 4, Issue 5, May 2015

11 Bryan Lemon, Aaron Riesbeck, Tim Menzies, etc all. "Applications of Simulation and AI Search: Assessing the Relative Merits of Agile vs. Traditional Software Development" 24th IEEE/ACM International Conference on Automated Software Engineering, 2009.

12 Carina Alves, Silvia Pereira, etc all. "Preliminary Results from an Empirical Study in Market-Driven Software Companies", WER 2007, Toronto, Canada.

13 Carina Alves, Silvia Pereira, George Valenca, etc all. "Preliminary Results from an Empirical Study in Market-Driven Software Companies" Proceeding SAC '08. Proceedings of the 2008 ACM symposium on Applied computing, pp 619-623.

14 Carlshamre, P. . "Release Planning in Market-Driven Software Product Development: Provoking an Understanding", Requirements Engineering, Volume 7, Issue 3, Sep 2002, Pages 139 - 151.

15 D A Penny. "An estimation-based management framework for enhancive maintenance in commercial software products", Proceedings of International Conference on Software Maintenance, 2002.

16 D Greer. "Requirements prioritization for Incremental and Iterative Development" Requirements Engineering for Sociotechnical Systems, Information Science Publishing, 2005.

17 Dan Port, Alexy Olkov, Tim Menzies. "Using Simulation to Investigate Requirements Prioritization Strategies", Proceedings of 23rd IEEE/ACM International Conference on Automated Software Engineering, 2008.

18 Davis, A.M. "The Art of Requirements Triage", IEEE Computer 36(3), Mar 2003, pp 42- 49.

19 Deepti Mishra, Alok Mishra. "Managing Requirements in Market-Driven Software Project: Agile Methods View", Technical Gazette, Apr-Jun 2010, Vol. 17, Issue 2.

20 Donald Firesmith. "Prioritizing Requirements", Journal of Object Technology, vol. 3, no. 8, September- October 2004, pp. 35-47. http://www.jot.fm/issues/issue_2004_09/column4.

21 EvangelosTriataphyllou, Stuart H. Mann . "An Examination of the Effectiveness of the Multi- Dimensional Decision-Making Methods - A Decision-Making Paradox" Decision Support Systems 5 (1989) 303-312

22 Fabrício G. Freitas, Daniel P. Coutinho, Jerffeson T. Souza. [2011] "Software Next Release Planning Approach through Exact Optimization", International Journal of Computer Applications, Vol. 22-No. 8, May 2011.

23 Frank Moisiadis. "THE FUNDAMENTALS OF PRIORITISING REQUIREMENTS" Systems Engineering, Test & Evaluation Conference, Sydney, Australia, October 2002

24 G. Ruhe: Software Release Planning. Handbook of Software Engineering and Knowledge Engineering Vol. 3 (S.K. Chang, Ed.), World Scientific 2005, pp 365-394

25 Greg Cohen. "Lean Product Management" First Edition 2011, eBook ISBN: 978?1?60005?216?3, Publisher: 280 Group.

26 Gunter Ruhe, Moshood OmoladeSailu. "The Art and Science of Release Planning", IEEE Software, Vol. 22, Issue 6, Nov 2005. Pg 47-53

27 Hoh Peter In. "Requirements Negotiation Using Multi-Criteria Preference Analysis", Journal of Universal Computer Science, Vol. 10, no. 4 (2004), 306-325.

28 Hubert F. Hofmann, Franz lehner. "Requirements Engineering as a Success Factor in Software Projects", IEEE Software, July/August 2001.

29 Karlsson J, Ryan K. "A cost-value approach for prioritizing requirements" IEEE Software, 1997, 14(5), pp 67-74.

30 Karlsson J, Wohlin C, Regnell B . "An evaluation of methods for prioritizing software requirements", Inform. Software Technol.1998, 39(14-15): 939-947.

31 Krzysztof Wnuk. "Understanding and supporting Large Scale Requirements Management", (2010), Dept. of Computer Science, Lund University.

32 Krzysztof Wnuk. "Visualizing, Analyzing and Managing the Scope of Software Releases in Large-Scale Requirements Engineering" (2012), Dept. of Computer Science, Faculty of Engineering, Lund University [PhD Thesis].

33 Kuda Nageswararao, Praneeth Chakka. "Research and Reflection Report on MDRE Problems", International Journal of Advanced Research in Computer Science and Software Engineering, Vol 2, Issue 12, Dec 2012. www.ijarcsse.com

34 Kyriaki D. Tsilika, Vassilis C. Gerogiannis. "A Simulation Approach to Evaluate the Quality and the Performance of a Software Requirements Engineering process", 5th International Conference on Experiments/Process/System Modeling/ Simulation/ Optimization, Athens, July 2013.

35 L. Karlsson, P. Berander, B. Regnell and C. Wohlin, "Requirements Prioritization: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning", Proceedings 8th Conference on Empirical Assessment in Software Engineering, Edinburgh, UK, 2004.

36 Laura Lehtola and Marjo Kauppinen . "Suitability of Requirements Prioritization Methods for Market-driven Software Product Development", Software Process Improvement Practice 2006, pp 7-19. Published online in Wiley Inter Science (www.interscience.wiley.com). DOI: 10.1002/spip.249

37 Laura Lehtola. " Providing value by prioritizing requirements throughout software product development", (2006), Dept. of Computer Science and Engineering, Hlesinki University of Technology [PhD Thesis]

38 Lipika Bose Goel, Prof. Sanjeev Thakur. "Requirement Prioritization: A Study and Proposed Framework", International Journal of Emerging Technology and Advanced Engineering Volume 3, Issue 6, June 2013. Website: www.ijetae.com

39 Manju Khari, Nikunj Kumar . "COMPARISON OF SIX PRIORITIZATION TECHNIQUES FOR SOFTWARE REQUIREMENTS", Journal of Global Research in Computer Science, Volume 4, No. 1, January 2013. Available Online at www.jgrcs.info

40 Markus Lindgren "Towards a Capability Model for Release Planning of Software Intensive Systems", (2008), School of Innovation, Design and Engineering, Malardalen University [PhD Thesis]

41 Martin Glinz, Patrick Heymans, etc all. "Report on the Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'09)", ACM SIGSOFT Software Engineering Notes, Sept. 2009. Vol. 34, No. 5

42 Martin Host, Bjorn Regnell etc. all. "Exploring Bottlenecks in Market-Driven Requirements Management Processes with Discrete Event Simulation", Proceedings of Software Process Simulation Modeling workshop, London, July 2000. www. prosim.org.

43 Mikael Lindvall. "An Empirical Study of Requirements-Driven Impact Analysis in Object-Oriented Software Evolution", (1997), Dept. of Computer and Information Science, Linkoping University. [PhD Thesis]

44 Mikael Svahnberg, Tony Gorschek, etc. all. "A Systematic Review on Strategic Release Planning Models", Information and Software Technology, Vol 52, Issue 3, March 2010. (Elseveir publication).

45 MikkoVestola . "A Comparison of Nine basic Techniques for Requirement Prioritization", Seminar in Software Engineering, Aalto University: School of Science, June 2010

46 Ming-June Lee. "Foundations of the Win-Win Requirements Negotiation System" (1996), University of Southern California. [PhD Thesis]

47 Mohammad Dabbagh, Sai Peck Lee. "An Approach for Integrating the prioritization of Functional and Nonfunctional Requirements", The Scientific World Journal, April 2014, Hindawi Publishing Corporation.

48 Muhammad Ramzan, M. Arfan Jaffar and Arshad Ali Shahid . "VALUE BASED INTELLIGENT REQUIREMENT PRIORITIZATION (VIRP): EXPERT DRIVEN FUZZY LOGIC BASED PRIORITIZATION TECHNIQUE" International Journal of Innovative Computing, Information and Control, Volume 7, Number 3, March 2011, pp. 1017-1038.

49 N. Kukreja, B. Boehm, etc. all. "Selecting an Appropriate Framework for Value-based Requirements prioritization" 20th IEEE International Requirements Engineering Conference (RE), 2012.

50 Nina Dzamashvili Fogelstrom, Tony Gorschek, Mikael Svahnberg. "Needs Oriented Framework for Producing Requirements Decision Material -Norm" International Conference on Software Product Management, IWSPM 2008.

51 Nupul Kukreja, Barry Boehm, etc all. "Value-Based Requirements Prioritization: Usage Experiences" Procedia Computer Science 16 (2013) 806-813

52 P. Berander and P. J¨onsson. "Hierarchical cumulative voting (hcv) prioritization of requirements in hierarchies," International Journal of Software Engineering & Knowledge Engineering, vol. 16, pp. 819-849, 2006.

53 P. Berander, K. Khan, and L. Lehtola. "Towards a research framework on requirements prioritization," SERPS06: Sixth Conference on Software, Engineering Research and Practice, Sweden, 2006, pp. 39-48.

54 P. Carlshamre, Kristian Sandahl, etc. all. "An Industrial Survey of Requirements Interdependencies in Software Product Release Planning", Proceedings of 5th IEEE International Conference on requirements Engineering, 2001.

55 Pablo Trinidad, David Benavides, Antonio Ruiz Cortes . " Improving Decision making in Software Product Line product Management", Decision Support in Software Engineering (ADIS'04), 2004

56 Panagoita Chatzipetrou, Per Rovegard, Claes Wohlin. "Prioritization of Issues and Requirements by Cumulative Voting: A Compositional Data Analysis Framework", Software Engineering and Advanced Applications (SEAA), 2010, 36th EUROMICRO Conference on. (IEEE)

57 Paolo Avesani, Cinzia Bazzanella, Anna Perini, Angelo Susi. "Supporting the Requirements Prioritization Process. A Machine learning Approach" , Proc. of 16th International Conference on Software Engineering and Knowledge engineering (SEKE 2004), June 2004, Banff, Alberta (CAN).

58 Paolo Avesani, Cinzia Bazzanella, Anna Perini, Angelo Susi. "Facing Scalability Issues in Requirements Prioritization with Machine Learning Techniques", Proceedings of the 2005 13th IEEE Conference on requirements Engineering (RE'05).

59 Paolo Avesani, Sara Ferrari, Angelo Susi. "Case-Based Ranking for Decision Support Systems", IEEE Intelligent Systems, 20(2), 2005. pp 62-72.

60 Par Carlshamre. “Release Planning in Market-Driven Software Product Development: Provoking an Understanding” Requirements Engineering (2002) 7:139-151. Springer-Verlag

61 Patric Berander and Annalisa Andrews. "Requirement Prioritization" in Engineering and Managing Software Requirements, edited by A. Aurum and C. Wohlin, Springer Verlag.

62 Per Rune son, Martin Host. "Guidelines for conducting and reporting case study research in software engineering" Empirical Software Engineering, April, 2009.

63 PersisVoola, Vinaya Babu. "Requirements Uncertainty Prioritization Approach: A Novel Approach for Requirements Prioritization" Software Engineering: An International Journal (SEIJ), Vol. 2, No. 2, Sept 2012.

64 Philip Achimugu , Ali Selamat, Roliana Ibrahim, MohdNaz'ri Mahrin[2014]. " A systematic literature review of software requirements prioritization Research". Information and Software Technology 56 (2014) 568–585 journal homepage:www.elsevier.com/locate/infsof

65 Qi Li. "Value-Based, Dependency-Aware Inspection and Test Prioritization", (2012), University of Southern California. [PhD Thesis]

66 Qiao Ma. "The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements: A Systematic Literature Review", (2009), School of Computing and Mathematical Sciences, Auckland University [Masters Thesis]

67 Raymond Joseph Madachy. "A Software Project Dynamics Model for Process Cost, Schedule and Risk Assessment", (1994), University of Southern California. [PhD Thesis]

68 Regnell, B., H¨ost, M., Nattoch Dag, J., Beremark, P., Hjelm, T. . " An Industrial Case Study on Distributed Prioritization in Market-Driven Requirements Engineering for Packaged Software", Requirements Engineering 2001, vol 6, no 1, pp 51-62.

69 Ricardo Valerdi. "The Constructive Systems Engineering Cost Model (COSYSMO), (2005), University of Southern California. [PhD Thesis]

70 Richard Berntsson Svensson. "Managing Quality Requirements in Software Product Development", (2009), Dept. of Computer Science, Lund University. [PhD Thesis]

71 Ruby, Dr. Balakishan. " Role of Fuzzy Logic in Requirement Prioritization", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 4, Issue 6, June 2015

72 Sairam Vakkalanka, Ranjit Engu. "REQUIREMENTS TRIAGE - CHALLENGES AND SOLUTIONS", International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.2, March 2012.

73 Samer I. Mohamed, Islam A. M. EIMaddah, Ayman M. Wahba. "Criteria-Based Framework for Software Product Management", International Journal of Software Engineering, IJSE Vol. 3, No.1, January 2010.

74 SatishTumuluru, Kambhampati Vijay Kumar, YVR Naga Pavan. "A Machine Learning Approach to Software Requirements Prioritization" International Journal of Scientific Engineering and Technology Research, Vol 3, Issue. 32, Oct 2014. www. Ijsetr.com

75 Thamer AlBourae, Guenther Ruhe, Mahmood Moussavi. "Lightweight Replanning of Software Product Releases", International Workshop on Software Product Management 2006 (IWSPM'06).

76 Thomas Bebensee, Ing Van de Weerd, Sjaak Brinkkemper. "Binary Priority list for Prioritizing Software Requirements", Requirement Engineering: Foundation for Software Quality, Vol. 6182 of the series Lecture notes in Computer Science, pp67-88.

77 Thomas L Saaty. "Decision making with analytic hierarchy process" Int. J. Services Sciences, Vol. 1, No.1, 2008

78 Thomas L Saaty. "How to make a decision: The Analytic Hierarchy Process", European Journal of Operational research 48 (1990), 9-26, North Holland.

79 Tomas Tan. "Domain-Based Effort Distribution Model", (2012), University of Southern California. [PhD Thesis]

80 Vu Nguyen. "Improved Size and Effort Estimation Models for Software Maintenance", (2010), University of Southern California. [PhD Thesis]

81 Vyas Gayathri, Misal Chetan S. "Comparative Study of Different Multi-criteria Decision-making Methods", International Journal on Advanced Computer Theory and Engineering (IJACTE), Vol. 2, issue 4, 2013.

82 Walker Royce. " Improving Software Economics" White paper, May 2009, IBM

83 Wiegers KE. "Software Requirements", Microsoft Press (1999): Redmond, DC.

84 Yoav Freund, Raj Ayer, etc, all. " An Efficient Boosting Algorithm for Combining Preferences", Journal of Machine Learning Research 4 (2003) 933-969

85 Zhihao Chen. "Reduced parameter Modeling for Cost Estimation Models", (2006), University of Southern California. [PhD Thesis]

86 Zornitza Racheva, Maya Daneva, etc, all. "Do we know enough about Requirements in Agile Projects: Insights from a Case Study", Conference: Requirements Engineering, IEEE International Conference -RE, 2010, pp. 147-156.

Additional List

87. Avdic A and Lambrinos T. "Modeling and Illustrating Requirement Prioritization in Public E-Service Development From a Value-Based Perspective" The Electronic Journal of e-Government Volume 13 Issue 1 2015 (pp 3-17).
www.ejeg.com

88. C. Li, J.M. van den Akker, S. Brinkkemper, G. Diepen. "Integrated Requirement Selection and Scheduling for the Release Planning of a Software Product" Department of Information and Computing Sciences, Utrecht University, Technical Report UU-CS-2006-059
www.cs.uu.nl

89. Ishaya P. Gambo,Abimbola H. Soriyan, Rhoda N. Ikono. "A Proposed Process Model for Requirements Engineering using Delphi Techniques for Prioritization" I.J. Information Technology and Computer Science, 2015, 01, 73-80, Published Online December 2014 in MECS.
http://www.mecs-press.org/

90. Karl E Wiegers. "First Things First: Prioritizing Requirements" Software Development Magazine, Vol. 7, No. 9, September 1999, pp. 24-30.

91. Donald G. Reinertsen. " The Principles of Product Development Flow: Second generation Lean Product Development" Celeritas publishing, 2009. http://www.scaledagileframework.com/wsjf/