IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 11, November 2015.
www.ijiset.com

ISSN 2348 – 7968

On Prioritizing and Freezing of Software Requirements

Vikas S. Chomal
[1]Assistant Professor
[2]Research Scholar
[1]The Mandvi Education Society Institute of Computer Studies,
Mandvi, Gujarat, India
[2]Singhania University, Pacheri Bari, District – Jhunjhunu, Rajasthan
vikschomal80@gmail.com

Dr. Jatinderkumar R. Saini
[1]Director I/C & Professor
[2]Research Supervisor
[1]Narmada College of Computer Application,
Bharuch, Gujarat, India
[2]Singhania University, Pacheri Bari, District – Jhunjhunu, Rajasthan
saini_expert@yahoo.com

Abstract: Software requirement is a state or an aptitude necessitated by end user to overcome a problem or to accomplish the needed purpose. Further, it is generally classified into functional and non – functional requirements. Considering this, a well premeditated course of action is used for determining user potential for a new as well as existing product, which is known as Requirement Engineering or Requirement Analysis. Also it is considered as one of the most significant task and serves as a foundation in software development. Requirement analysis procedure consists of various activities such as understanding of domain, collection and classification of requirements, conflict resolution, requirements prioritization and validation. Through this paper, we describe the reviews of requirements engineering and essentially on the most noteworthy task that is prioritizing and freezing of software requirements during Requirement analysis process. The paper is divided into four sections; first section is introductory, followed by literature review. Section three represents finding and analysis from literature review, followed by concluding section.

Keywords: Requirement, Requirement Analysis, Requirement Analysis Process, Requirement Prioritization, Software Engineering

I. INTRODUCTION

An early definition of software engineering was "the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines". Boehm adds the provision of human utility in the definition "software engineering is the application of science and mathematics by which the capabilities of computer equipment are made useful to man via computer programs, procedures and associated documentation" [5]. Lethbridge et al. [30] define software engineering as, software engineering involves real people in real environments. People create software, people maintain software, and people evolve software. Accordingly, to truly understand software engineering, it is imperative to study people - software practitioners as they solve real software engineering problems in real environments. This means conducting studies in field settings. Software engineering is an intensely people-oriented activity, yet too little is known about how designers, maintainers, requirements analysts and all other types of software engineers perform their work. Software Engineering (SE) is not simply concerned with technical activities, but it also involves both managerial aspects, e.g., definition, organization and planning of multiple interrelated activities, and collaborative aspects, e.g., information exchange and coordination of people working to fulfil a common objective [7].

The term requirement may be defined as demand or need. In the world of the software engineering, a requirement is a explanation of what the purposed system should do or perform. A system may have a lot of requirements. Software requirements demand what must be accomplished, shaped or provided. Requirement elicitation is all about knowing the desires of stakeholders. [10]

Requirement Engineering can be defined as a subset of Software Engineering that mainly focuses on the use of various tools and techniques for the development and management of software requirements. The processes involved in Requirement Engineering include domain analysis, elicitation, specification, assessment, negotiation, documentation, and evolution [32, 33].

Requirement Engineering (RE) or Requirement Analysis is concerned with the identification of the goals to be achieved by the envisioned system, the operationalization of such goals into services and constraints, and the assignment of responsibilities for the resulting requirements to agents such as humans, devices, and software. The processes involved in RE include domain analysis, elicitation, specification, assessment, negotiation, documentation, and evolution [9]. Requirement Engineering (RE) is worried about the naming of the goals, achieved by the imagine system. Most difficult and critical to be achieve the better quality of the requirement. Incomplete, inconsistent and ambiguous requirements have the most serious impact on the required software. If requirement errors correction perform late the cost raise up to 200 times as compared the requirement errors correction perform in time. Requirement Engineering deals with a wide range of business domains and tasks like decision, administrative support [3]. Zave[24] provides one of the clearest definitions of requirement engineering: “Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behaviour, and to their evolution over time and across software families”.

During software development, a frequent communication with its users should be maintained so that their specific prospects, decision as well as conflicts on uncertainty of requirements can be eliminated. For this reason, a predefined and well planned requirement analysis process can be followed as specified in figure: 1.

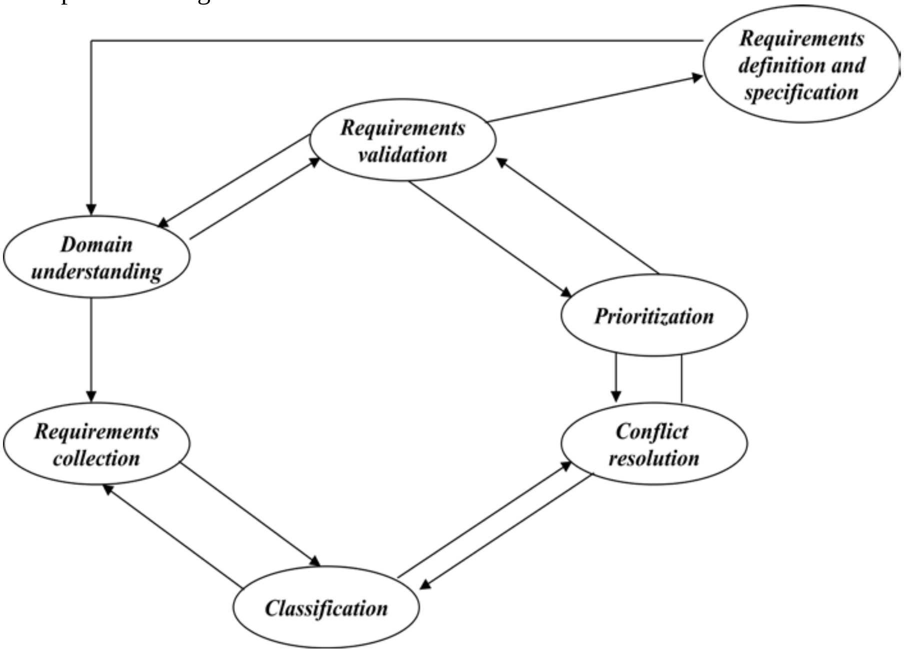

Figure: 1 Requirement Analysis Process [12]

(1) Domain Understanding: Domain understanding means, understanding of application domain to be developed.
(2) Requirement Collections: It is the course of action in which interaction between customers and users is done to discover the requirements of the system.
(3) Requirement Classification: After the requirements are gathered, they are clustered and classified.
(4) Conflict resolution: Many times, when the multiple clients are concerned requirements will conflict. Conflicts need to be determined keeping in mind the overall objectives.
(5) Prioritization: Here, requirements are identified and listed according to their importance.
(6) Requirements Validation: Check and validate the gathered requirements to see if they are complete, correct and sound.

So from Figure: 1, we stipulate that, domain understanding, requirement collections and classifications, conflict resolution, prioritization and requirements validation are major activities done during requirement analysis process.

II. LITERATURE REVIEW

Salman and Awan states that, the use of requirements prioritization helps the anatomy of requirements and isolates the most important requirements. According to them, lot of prioritization techniques, practices and methodologies are used in software requirements. But lack of empirical search program and proficient methodology, was not decide which should be implemented. In their research, the requirement prioritizations for systematically reviews were carried out. Based on systematic review, they introduced a framework further research within requirement prioritization. They presented practical view how to prioritize the requirements. It also reflects the requirements prioritization in the industries needs [29]. According to Joachim and Kevi, the function of prioritization of requirements is very important for an efficient and result oriented product development. Requirement prioritization marks high risk and most significant requirements to be given priority in implementation [15].

Ramzan et al, declare requirement prioritization as an extremely dangerous area of requirement engineering. Without appropriate requirement prioritization, offer by different stakeholders, the necessary objectives of the end product cannot be achieved properly. The product may fails to meet its heart objectives on the basis of several requirements prioritization techniques presented by different researchers [19].

Siddiqi and Shekaran suggested that, when requirements are extracted, it often yields more requirements than can be implemented at once. The requirements need to be prioritized so that the most significant ones are met by the earliest product releases [11]. According to Karlsson and Ryan, during software development utmost care is taken on issues such as release plan, available resources, milestones, risks management, product strategies, budget and so on, but unfortunately, there is a lack of simple and effective techniques for requirement’s prioritization, which could be used for release planning [8]. Khari and Kumar [18] presented a comparison among six techniques for prioritizing software requirements. The chosen techniques were Analytic Hierarchy Process (AHP), Value Oriented Prioritization (VOP), Cumulative Voting (CV), Numerical Assignment Technique (NAT), Binary Search Tree (BST) and Planning Game (PG). To study these techniques, they systematically applied all techniques to prioritize a set of thirteen quality requirements. Also they categorize the techniques from a user’s perspective according to five criteria such as ease of use, total time taken, scalability, accuracy, and total number of comparisons required to make decisions. Chomal and Saini[33,34,35] considered poorly collected and analysed requirements as one of the major cause of software failure.

Ani [4] conducted a survey to investigate the problems encountered and the common practices applied during requirement analysis. The author used requirement decomposition and prioritization as the prominent attributes for research. Along with this attributes, author also provide with a comprehensive list which are identified and stated as, main problems encountered during requirement analysis. They are:

(1) A stakeholder cannot fully define what is needed until there is something tangible to use and critique.
(2) Stakeholders can be unaware of certain aspects of the system being described.
(3) A balance must be achieved between professional responsibilities and developing what is being paid for.
(4) Although complete requirements were stated as being desired, it was also a common policy to exclude details when defining requirements.

Achimugu et al. [22] converse that during requirements engineering, prioritization is performed to grade or rank requirements in their order of importance and subsequent implementation releases. They consider it as one of the major step to be taken while making crucial decisions so as to increase the economic value of a system. They also symbolized prioritization taxonomies and represented it graphically as shown in Figure: 2.

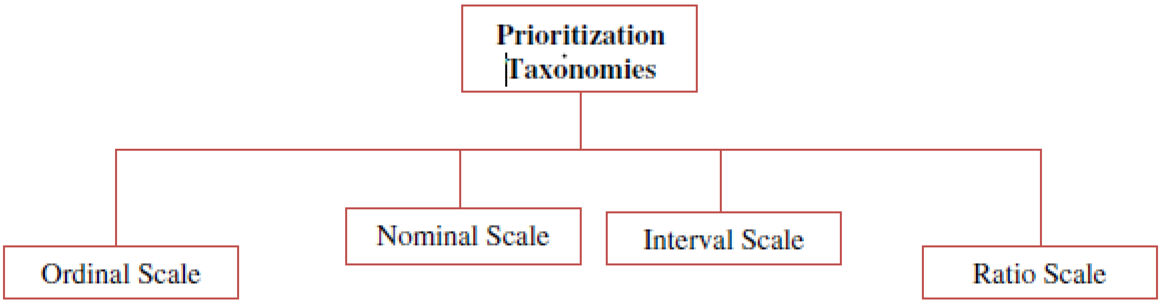

Figure: 2 Prioritization taxonomies [22]

Further they specify that, techniques exhibiting any of these taxonomies order requirements into groups or sub-groups and numbers or inscriptions are assigned across all the requirements to reflect their relative importance. Here, no requirement is expected to belong to more than one group or subgroup at a time. Consequently, the frequencies across all requirements are computed and the requirement with the highest frequency is taken to be the prime requirement. The ordinal scale can be used to complement the performance of the nominal scale since the former present information about the ordering of requirements ; while the latter has the capacity of showing the ranks of various requirements but cannot further indicate the extent at which, one requirement is considered to be more important than the other. The interval scale possesses information about the size of the intervals between the ordered set of requirements so as to enhance the computation of the discrepancy that exist among requirements. An interval scale preserves order, just like the ordinal scale. The ratio scale is considered to be the most viable of all the scales or taxonomies because, it has the capacity of ordering, determining intervals or relative distances.

In direction to prioritization, Thakurta [27] states that, several prioritization techniques are available in order to identify the most important requirements with greater contribution to business value. Herrmann and Daneva (2008) present a comparative review of 15 such techniques with the important ones detailed below. Numerical assignment is based on grouping requirements into different categories, viz. high, medium, and low (Sommerville and Sawyer 1997). A criticism of the technique is that use of categories like high, medium, and low may confuse the stakeholders since different stakeholders may have different views of what, for example, high and medium mean (Wiegers 2003). The pairwise comparison technique (Karlsson 1996) based on the Analytical Hierarchy Process (AHP) (Saaty 1980) performs pairwise comparison of requirements as per their importance. The comparisons provide an understanding of each requirement share of the total value. Cost–value approach (Karlsson and Ryan 1997) is another prioritization technique based on AHP that uses a two dimension graph in order to display the requirements value against its cost. AHP is used from a customer and user perspective to assess the value of each requirement, followed by an assessment of the requirement cost from an implementation perspective. Cumulative voting ($100-Dollar Test, Leffingwell and Widrig 2000) relates to distribution of a fictitious $100 on requirements. At the end of the distribution, the requirements are ranked so that the highest total reflects the most important requirement, the next highest is the Software Quality Journal (2013) 21:573–597 575 next-most important, etc. on a ratio scale. The ranking technique based on an ordinal scale facilitates relative ordering of the requirements from most important to least important. Ranking can be carried out using techniques like bubble sort, binary search tree, binary priority listing, etc. (Karlsson et al. 1998). However, in this case it is not possible to evaluate the relative priority difference among the requirements.

Hassan et al. [28] notified that, the role of prioritization of requirements is imperative to an efficient and result oriented product development. Requirement prioritization marks high risk and most important requirements to be given priority in implementation. Chomal and Saini [31] focuses on requirement as the prime cause for software failure by stating the following: (1) Requirements do not represents the actual needs of the customers (2) Fail to accurately specify requirement in sufficient detail (3) Requirements are incomplete or conflicting (4) Requirements are difficult and expensive to update after they have been agreed upon

Khan [13] in his research work states that, requirements at the earlier stages are vague and evolve and become more specific and clear with the passage of time. There is no specific phase where requirements prioritization takes place; rather it is an iterative process which is performed throughout the product development life cycle (Lehtola et al. 2004) and between different versions of the same product as it is evolved (Kuusela and Savolainen 2000). Berander [23] argues that requirements prioritization is not just asking the different stakeholders about their different priorities related to the system requirements. It is also a social process which requires considering other issue like organizational setup, organizational market value, organizational issues, stakeholders personalities and agendas (Berander 2004a). It also requires tradeoffs during conflicting situations between the different stakeholders to reach to a situation when the candidate requirements have been selected which will eventually form a system.

Goel and Thakur [16] put forward with three consecutive stages of prioritizing, they are:
(1) The preparation stage where a person structures the requirements according to the principle of the prioritizing methods to be used. A team and a team leader for the session is selected and provided all necessary information.
(2) The execution stage where the decision makers do the actual prioritizing of the requirements using the information they were provided with in the previous stage. The evaluation criteria must be agreed upon by the team before the execution stage is initiated.
(3) The presentation stage where the results of the execution are presented for those involved. Some prioritizing methods involve different kinds of calculations that must be carried out before the results can be presented.

Ramzan et al. [20] convey that requirement prioritization is a very critical but often neglected area of requirement engineering. Experience has shown that without proper prioritization of requirements presented by various stakeholders, the end product usually fails to meet its objectives optimally. In fact in many instances, the product is considered a failure because it fails to meet its core objectives. With requirement prioritization, we can identify the focus areas which need most of our attention in order to develop a product which optimally meets the requirements of the stakeholders.Koziolek [14] proposes a new approach to prioritize quality requirements, relying on feedback from architecture evaluation and automated design space exploration. Qiao Ma [25] state that, requirements prioritization is one of many critical steps in the software development process. The context of requirements prioritization is introduced first to present what the role of requirements prioritization is in software engineering and how it relates to other critical steps to produce software systems. Massey et al. [1] reveal that requirements prioritization is used in the early phases of software development to determine the order in which requirements should be implemented. Requirements are not all equally important to the final software system because time constraints, expense, and design can each raise the urgency of implementing some requirements before others. Laws and regulations can make requirements prioritization particularly challenging due to the high costs of noncompliance and the substantial amount of domain knowledge needed to make prioritization decisions. In the context of legal requirements, implementation order ideally should be influenced by the laws and regulations governing a given software system. In their work, they presented a prioritization technique for legal requirements. They applied their proposed technique on a set of 63 functional requirements for an open-source electronic health records system that must comply with the U.S. Health Insurance Portability and Accountability Act.

Wiegers [14] discloses an approach regarding prioritization in which requirements are grouped into three priority categories. Table: 1 shows two typical three-level scales. All such scales are subjective and imprecise, so everyone involved must agree on the meaning of each level in the scale they use. Priority is a key attribute of each requirement that should be included in the SRS or requirements database. Establish a convention for your SRS so the reader knows whether the priority assigned to a higher-level requirement is inherited by all of its subordinate or derived requirements, or whether every individual requirement should have its own priority attribute.

Table: 1 Two Requirements Prioritization Scale [14]

Names	Meanings
High Medium Low	a mission critical requirement; required for next release supports necessary system operations; required eventually but could wait unit a later release if necessary a functional or quality enhancement; would be nice to have someday if resources permit
Essential	the product is not acceptable unless these requirements are satisfied

Conditional	would enhance the product, but the product is not unacceptable if absent
Optional	functions that may or may not be worthwhile

Berander et al [23] wrap up that, there exist a large number of approaches for prioritization of software requirements. Despite of several empirical studies, there is still a lack of evidence of which approaches that are to prefer, since different studies have resulted in different conclusions. Also they strongly considered requirements prioritization as one of the most challenging and complex decision-making activity. An important requirement in one release for a certain customer may not be as important in the next release for another customer. Also, customers may be unconvinced towards prioritizing because they are afraid that only the most important requirements get implemented. At the same time, developers may be sceptical since they do not want to admit that they are not able to implement all the requirements. Vestola [21] compared and illustrated nine basic techniques for requirement prioritization through which will guide in choosing the right prioritization technique and also helps to develop completely new prioritization methods using these prioritization techniques as basic building blocks. In addition, the results of this paper also guide which techniques need more empirical studies.

Otero et al [6] depict that, despite the clear need to prioritize requirements in software projects, finding a practical method for requirements prioritization has proven difficult. Existing requirements prioritization methods that provide the most consistent results are also the most complex, and therefore the most difficult to implement. More informal methods save time and are easier to apply, but may not be suitable for practical scenarios because they lack the structure and consistency required to properly analyze requirements. Zhang [40] elucidates by stating that, test case prioritization is an effective and practical technique in regression testing. It schedules test cases in order of precedence that increases their ability to meet some performance goals, such as code coverage, rate of fault detection. Also Zhang [40] focuses on that previous work carried out on test case prioritization techniques and metrics usually assumed that testing requirement priorities and test case costs are uniform.

Beg et al [26] explicate and focuses on various constraints of software projects such as time, features, budget and resources. The needs and expectations of the customer are dynamic and depend on the various factors which keep changing. In order find out their relevance, to increase the customer satisfaction, planning the release of new prototype or version the requirements prioritization have gained importance and have became an essential characteristic of requirements . This is also important when there are strict constraints on schedule & resources one must negotiate & prioritize the most essential features to be incorporated as to make project successful. They also included explanation regarding industrial projects and found that, there are many requirements which keep on increasing as the projects undergo development, so prioritization is must but often it is dropped because of large number of growing requirements need more number of comparisons. They strongly raised the point as regards to prioritization that, there has been little progress to date, either theoretical or practical, on the mechanisms for prioritizing software requirements. The right requirements are measured as a part and parcel of software quality. Since the materialization of software engineering the perfect requirements have a deeper effect on the overall quality of software systems, stated by Babar et al [17]. They also endeavour to make clear that existing techniques are not providing sufficient automation for such systems due to their certain limitations. Chomal and Saini [36, 37, 38] measured requirement as one of the quantifiable attribute which is used as a criterion to evaluate and score software project documentation.

III. FINDING AND ANALYSIS

This section presents our recognition in Table: 2 which are considered as the major concerns found from the review of related literature. We also present a scrutiny of the recognized learning which proves significance of prioritization and freezing of requirements in software development process.

Table 2: Analysis of Literature

Sr No.	Author(s)	Major Contribution
1.	Hassan and Awan [28]	They described a framework for scrutinize the discussion that take place

		during requirements elicitation and requirements prioritization. They also conducted a survey and presented a practical view how to prioritize the requirements. It also reflects the requirements prioritization in the industries needs. Also provided brief factors of the requirements engineering that affect the requirements prioritization. Their survey study indicated that there are three factors like analyst's qualification, experience and age which affect the prioritization. These factors help to minimize the distance between the stakeholders. The defined three factors resolved the issues like relationships, communication problems between users and software engineers. Any software developer pursuing these attributes can make easily requirements prioritization and implementation of the requirements. Accurate requirements prioritization produced the cost effective solution for the organization.
2.	Khari and Kumar [18]	Outcome of their experiment says that Value Oriented Prioritization (VOP) is supposed to be the best method for prioritizing software requirements. It is an easy method, it gives one of the most accurate results, and it is rather comfortable to handle even if there are many more requirements.
3.	B. Ani [4]	It is common practice to ascribe priority to requirement statements. Several projects, most notably market driven ones, assigned priorities to requirements because certain features are more important than others and some features or properties can be ignored if deadlines dictated.
4.	P. Achimugu et al. [22]	The purpose of their study was to identify and analyze existing prioritization techniques in the context of the formulated research questions. The aim of their research was to identify and examine the status of software requirements prioritization techniques. The method that they utilized in research was a systematic literature review. With this method, some pertinent research questions were formulated based on the aim of the study to identify and ascertain existing prioritization techniques, their limitations, taxonomies and processes. They also developed a review protocol was and some quality evaluation questions or criteria were developed and applied to the studies to ensure relevance and correctness.
5.	R. Thakurta [27]	The author presented a Non – Functional Requirement (NFR) prioritization algorithm in order to facilitate selection of project quality attributes with an aim to satisfy the overall business objectives. Even though the algorithm has been operational based on a subset of NFRs, the method is applicable for all NFRs known in practice. The findings are expected to benefit both the business and the project organizations by identifying and implementing the desired qualities that contribute to business value in a cost effective manner. The results also provide ways to understand design limitations, and offers suggestions for improvement in line with the project's overall objectives. Project organizations not having any criteria for selecting NFRs or resorting to ad hoc methods of prioritization are likely to benefit from the use of the framework.
6.	Awan et al [29]	They conducted the study at three types of industries, to find out the results about requirements prioritization that gives the high level satisfaction of the customer. To achieve the explanation of the actual requirements prioritization, conduct the survey in-depth. The purpose of the survey was to determine how organizations prioritize the requirements. The term priority is the property or attribute of the requirement. In the survey organization, the requirements elicitation and requirements prioritization interacts the domain expert users. The survey study indicated that there are three factors like analyst qualification, experience and age of the analyst, which affect the prioritization. These factors help to minimize the distance between the stakeholders.
7.	Khan [13]	Requirements prioritization involves both quantitative techniques and qualitative methods, which involves a negotiation process. The quantitative techniques also require negotiations between different stakeholders; however they also assign numeric numbers to requirements. Requirements prioritization also helps in detecting requirements ambiguities, incorrectness

		and abstraction. Therefore there is a need for a systematic review within requirements prioritization to see what evidence exists about their effectiveness.
8.	Goel and Thakur [16]	The authors projected a framework to prioritize the software requirements. The proposed framework ranked the requirements by the relative level of value, cost, efforts and threat associated with each requirement. Furthermore they also concluded that, prioritization process should not only considers the crucial factors but also the process must be easy to learn, easy to use, invisible to stakeholders to gain their interest and confidence, understandable to non experts, reliable, efficient, scalable and flexible. Lots of factors can be added to this list. And most of the methods and techniques that are available nowadays are for small development organization. If the scale increases, these techniques get very difficult to implement and are less accurate.
9.	Ramzan et al. [20]	They offered a novel multi-level value based intelligent requirement prioritization technique using fuzzy logic. Along with this, they introduced and applied the concept of requirement value to prioritize requirements. They performed extensive experimentation using our proposed technique along with existing techniques. Results have shown that our technique has achieved superior prioritization results and consistency. The experiments have also shown that proposed technique is capable of delivering impressive prioritization under various circumstances. In this technique, stakeholders, experts and fuzzy logic based system perform separate prioritizations at three different levels. They also presented a comparative analysis based on experimental results. This analysis shows that in almost all different environments, intelligent requirement prioritization is able to exhibit better and impressive results.
10.	A. Koziolek [2]	They illustrated an approach to support quality requirements prioritization by providing feedback from quantitative architecture evaluation and design space exploration. Applying their approach, stakeholders, requirements engineers, and software architects can gain a better understanding of the dependencies of quality attributes and the effects of achieving certain quality values. Thus, it helps them to prioritize quality requirements and decide for an optimal trade - off. However, the approach is currently limited to quantitatively evaluated quality properties.
11.	Q. Ma [25]	Qiao analysed the evaluations of prioritization techniques discussed and came with conclusion that most prioritization techniques generally work well on small numbers of requirements, but many of them have limitations on medium to large numbers of requirements. This leads to a question: are there prioritization methods that are suitable for medium to large numbers of requirements? This question motivated the author to conduct the research. The objective of the research was to investigate the strength of evidence for the effectiveness of different requirements prioritization techniques for medium to large numbers of requirements.
12.	Massey et al. [1]	The authors propose a numeral assignment technique for prioritizing legal requirements. The technique designed and implemented by them can be used by new or existing systems that must comply with laws or regulations to prioritize the requirements relating to those laws and regulations. Their work addresses several research needs as such prioritization addresses requirements refinement and dependency management, previously identified as an under researched area by many research scholars. In addition, they also demonstrated that numeral assignment prioritization techniques may be useful in the legal domain. This numeral assignment technique provides time- and effort-savings when laws, regulations, or the requirements change since only the affected requirements need to have their prioritization scores recalculated.
13.	Wiegers [14]	The author described a semi-quantitative analytical approach to requirements prioritization.
14.	Berander et al. [23]	The authors came with a framework for studies about requirements prioritization, which aims to enable building a more consistent knowledge

		base and stronger evidence. The framework facilitated comparison, replication, and high-level analysis of prioritization approaches by proposing suitable variables to measure. The basis of the framework came from a systematic review conducted on requirements prioritization techniques, and is further refined through literature studies of similar frameworks in related areas, and in a research workshop. Using this framework it will support researchers in conducting and reporting prioritization studies, and supports practitioners in getting information about different approaches.
15.	Vestola [21]	The author in deep presented comparison of nine basic techniques for requirement prioritization and concluded that, it is quite impossible to say, which of the methods the best is. It really depends on the situation. For this reason, the author stated that almost all of the studies used a very small number of requirements (20 or less). None of the studies focused on large sets of requirements. Therefore, it is hard to say how the techniques perform in real-life projects which can have tens or even hundreds of requirements.
16.	Otero et al. [6]	They projected a narrative and practical approach for prioritizing requirements in software projects. Their proposed approach attempts to quantify the quality of requirements to provide a measurement that is representative of all quality criteria identified for a specific software project. The derived quality measurement can be easily computed to serve as the main metric for requirements prioritization.
17.	Zhang et al. [40]	They provided with a new, general test case prioritization technique and an associated metric based on varying testing requirement priorities and test case costs. To explain the approach they illustrated with a case study, that the rate of "units-of testing-requirement-priority-satisfied- per-unit-test-case-cost" can be increased, and then the testing quality and customer satisfaction can be improved.
18.	Beg et al. [26]	Through their research and findings, they encouraged the industries to prioritize requirement, with help of B-Tree method in which the number of comparison required by the proposed method can be kept low.
19.	Babar et al. [17]	In their research they focused on the shortcomings or limitations of existing software requirements prioritization techniques and pave the way for researchers to explore new horizons in software requirements prioritization. They exposed that, most of the techniques are solving the plight of small projects or toy applications. There is not a single evidence of a successful prioritization technique that would solve the problem of large set of requirements. So they concluded that innovative requirements prioritization approaches are required for systems where user requirements increase in hundreds and even in thousands.

IV. CONCLUSION

Thus we conclude that requirement prioritization and freezing are to be considered as an essential task during requirement analysis process so that the developed software has a high probability of acceptance from customer’s side. Also we state that revolutionary requirements prioritization methods and techniques are obligatory for systems where user requirements raise in hundreds and even in thousands.

REFERENCES

1. A. K. Massey, P. N. Otto, A. I. Antón, “ Prioritizing Legal Requirements”, publishes in: Proceeding RELAW 2009 Second International Workshop on Requirement Engineering and Law pages 27 - 32
2. A. Koziolek, “ Prioritizing Quality Requirements based on Software Architecture Evaluation Feedback”, REFSQ 2012, LNCS 7195, pp – 52 -58 2012. Springer – Verlag Berlin Heidelberg 2012
3. A. V. Lamsweerde,” Requirement Engineering in the Year 00: A Research Perspective”, Proceeding in 22nd International Conference on the Software Engineering, University of Limerick, Ireland, pages 5-19, 2000

4. B. Ani, “A Survey of Practices and Problems Associated with Incomplete User Requirements Documents”, Proceedings of AWRE 2001 Australian Workshop on Requirement Engineering.
5. B. Boehm, V. R. Basili, H. D. Rombach, M. V. Zelkowitz ,”Foundations of Empirical Software Engineering: The Legacy of Victor R. Basili”, Springer – Verlag New York, Inc. Secaucs, NJ, 2005.
6. C.E. Otero , A. Qureshi, L. D. Otero, “Framework Using Binary A Quality-Based Requirement Prioritization Inputs”,published in Mathematical/Analytical Modelling and Computer Simulation (AMS), 2010 Fourth Asia International Conference page187 – 192, ISBN: 978-1-4244-7196-6, DOI: 10.1109/AMS.2010.48, Publisher: IEEE
7. E. Bareisa, E. Karciauskas, V. Limanauskienė, R. Marcinkevicius, K. Motiejūnas,”Software Engineering Process And Its Improvement In The Academy”, , Information Technology And Control, 2005, ISSN 1392 – 124, volume.34, No.1.
8. J. Karlsson, K. Ryan, “Prioritizing requirements using a cost-value approach,” IEEE Software 14 (5), pp. 67–74, 1997.
9. J. Karlsson, “Software requirements prioritizing,” in: Proceeding of 2nd IEEE International Conference on Requirements Engineering, pp. 110–116, 1996.
10. J. Karlsson, C. Wohlin, B. Regnell, “An evaluation of methods for prioritizing software requirements,” Information and Software Technology, pages 939-947, 1998.
11. J. Siddiqi, M.C. Shekaran, “Requirements engineering: the emerging wisdom,” IEEE Software 13 (2), pp. 15–19, 1996.
12. J.R.Saini, V.S.Chomal, Y.K.Solanki, “Fundamentals of Software Engineering, First Edition - 2012
13. K. A. Khan , “A Systematic Review of Software Requirements Prioritization “, Master Thesis Software Engineering Thesis no: MSE-2006-18 October 2006 School of Engineering Blekinge Institute of Technology Box 520 SE – 372 25 Ronneby Sweden
14. K. E. Wiegers, “First Things First: Prioritizing Requirements”, Software Development 7(9), 48 – 53, 1999
15. K. Joachim, R. Kevin, “A Cost – Value Approach for Prioritizing Requirements”, published in, Journal IEEE Software Archive, volume 14 issue 5, September 1997, page 67 – 74, IEEE Software, Germany.
16. L. B. Goel, S. Thakur, “Requirement Prioritization: A Study and Proposed Framework”, International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 6, June 2013
17. M. I. Babar, M. Ramzan,” Challenges and future trends in software requirements prioritization”, Published in: Computer Networks and Information Technology (ICCNIT), 2011, page 319 – 324, ISSN : 2223-6317, Print ISBN: 978-1-61284-940-9, DOI: 10.1109/ICCNIT.2011.6020888, Publisher: IEEE
18. M. Khari, N. Kumar , “Comparison Of Six Prioritization Techniques For Software Requirements”, Journal of Global Research in Computer Science, Review Article, volume 4, No. 1, January 2013.
19. M. Ramzan , “Value Based Intelligent Requirements prioritization (VIRP): Expert Driven Fuzzy Logic Based Prioritization Technique”, International Journal. Innovative Computing, Information and Control. Islamabad, Pakisatn. Vol.7, No.3, pp 1017-1038.
20. M. Ramzan, M. A. Jaffar, A. A. Shahid,” Value Based Intelligent Requirement Prioritization (Virp): Expert Driven Fuzzy Logic Based Prioritization Technique”, International Journal of Innovative Computing, Information and Control ICIC International 2011 ISSN 1349-4198 Volume 7, Number 3, March 2011 pp. 1017 - 1038
21. M. Vestola, "A Comparison of Nine Basic Techniques for Requirements Prioritization," 2010, Helsinki University of Technology
22. P. Achimugu, A. Selamat, R. Ibrahim, M. Mahrin, “A systematic literature review of software requirements prioritization research”, Information and Software Technology 56 (2014) 568–585, available at, journal homepage: www.elsevier.com/locate/infsof
23. P. Berander , K. A. Khan, “ Towards a Research Framework on Requirements Prioritization”, Journal SERPS, volume 6, 2010
24. P. Zave, “Classification of Research Efforts in Requirements Engineering”, published in Journal ACM Computing Surveys (CSUR), volume 29, issue 4, 1997, pages 315 – 321
25. Q. Ma , “The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements: A Systematic Literature Review”, A Dissertation Submitted to Auckland University of Technology as a Part of the Requirements for the Degree of Master of Computer and Information Sciences , November 2009 , School of Computing and Mathematical Sciences
26. R. Beg, R, Q. Abbas, R. P. Verma, “ An Approach for Requirement Prioritization Using B-Tree”, Published in: Emerging Trends in Engineering and Technology, 2008. ICETET '08, page 1216 – 1221, E-ISBN : 978-0-7695-3267-7, Print ISBN: 978-0-7695-3267-7, DOI: 10.1109/ICETET.2008.158, Publisher: IEEE

27. R. Thakurta, “A framework for prioritization of quality requirements for inclusion in a software project”, Software Quality Control (Impact Factor: 1.14). 12/2013; 21(4). DOI: 10.1007/s11219-012-9188-5
28. S. Hassan,S. Awan, F. Jaan, H. Akmal, “ Factors affect on Requirement Prioritization”, IOSR Journal of Engineering e-ISSN: 2250-3021, p-ISSN: 2278-8719, Vol. 2, Issue 12 (Dec. 2012) ||V2|| PP 28-33.
29. S. Salman, S. Awan, “Factors Affect on Requirement Prioritization”, Global Journal of Computer Science and Technology Software & Data Engineering Volume 12 Issue 15 Version 1.0 Year 2012, Publisher: Global Journals Inc. (USA) Online ISSN: 0975-4172 & Print ISSN: 0975-4350.
30. T. C. Lethbridge,S. E. Sim, J. Singer, “Studying Software Engineers: Data Collection”, Empirical Software Engineering, 10, 311 – 341, 2005 Springer Science & Business Media, 2005
31. V.S. Chomal, J.R.Saini, “Identification and Analysis of Causes for Software Failures”‖, NATIONAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY] Volume: 04 | Issue: 02 | July – December – 2012
32. V.S. Chomal, J.R. Saini, “Software Quality Improvement by Documentation – Knowledge Management Model”‖, National Journal of System And Information Technology ISSN : 0974 – 3308, Volume 6, Number 1, June 2013, Page Number: 49 – 68
33. V.S. Chomal, J.R. Saini, “Finding Trend of Both Frequency and Type of Errors from Software Project Documentation”‖, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)ISSN 2278-6856, Volume 2, Issue 5, September – October 2013
34. V.S. Chomal, J.R. Saini, “Software Template to Improve Quality of Database Design on basis of Error Identification from Software Project Documentation”‖, International Journal of Engineering and Management Research ISSN No.: 2250-0758,Volume-4, Issue-1, February-2014, Page Number: 168-179
35. V.S. Chomal, J.R. Saini, “ Cataloguing Most Severe Causes that lead Software Projects to Fail”, International Journal on Recent and Innovation Trends in Computing and Communication , May – 2014 ISSN: 2321-8169 Volume: 2 Issue: 5 pages 1143– 1147
36. V.S. Chomal, J.R. Saini, “Significance of Software Documentation in Software Development Process” International Journal of Engineering Innovation & Research, ISSN: 2277 – 5668, Volume 3, Issue 4
37. V.S. Chomal, J.R. Saini, “Identification, Categorization and Weighting of Software Engineering Attributes for Quality Evaluation of Software Project Documentation” , International Journal of Advanced Networking & Applications (Special Issue) IJANA ISSN: 0975 – 0282, November – 2014, pages 53 – 62
38. V.S. Chomal, J.R. Saini, “Software Template for Evaluating and Scoring Software Project Documentations” , International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 1, April 2015, pages 15 – 27
39. V.S. Chomal, J.R. Saini, “Software Project Documentation - An Essence of Software Development” , International Journal of Advanced Networking & Applications Volume: 06 Issue: 06 Pages: 2563-2572 (2015) ISSN: 0975-0290
40. X. Zhang, C. Nie, X. Baowen, “Test Case Prioritization Based on Varying Testing Requirement Priorities and Test Case Costs”, Published in: Quality Software, 2007. QSIC '07. Seventh International Conference, page 15 – 24, ISSN: 1550-6002, ISBN: 978-0-7695-3035-2,DOI: 10.1109/QSIC.2007.4385476, Publisher: IEEE, http://link.springer.com/article/10.1007/s11219-012-9181-z#page-1