International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 12, December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

Requirements Priority Algorithm to Improve the Efficiency of the Requirements Traceability

Swathine. K[1],Dr. N. Sumathi[2]

[1]*Research Scholar, Department of Computer Science, Sri Ramakrishna College of Arts and Science, Coimbatore, Tamil Nadu, (India)*

[2]*Professor and Head, Department of Information Technology, Sri Ramakrishna College of Arts and Science, Coimbatore, Tamil Nadu, (India)*

ABSTRACT

Software are made up off requirements, when developing a software or update the software, requirements plays a key role. If requirements gathered are complete developing or updating the software will be error free. Software goes faulty and failure due to lack of complete and incorrect requirements. A poorly defined requirements can have negative impact on software, leads to loss of money, quality and time. Requirements management (RM) as core discipline for successful delivery of software.In RM requirements traceability solve the problem of software inconsistency. The effective way of doing traceability links between the requirements is with their priority. Requirements prioritization (RP) is important part of requirement traceability. RP is essential activity in software development. Quality software depends on selection of prioritized requirements. The best way of doing RP is based on nature of frequency of requirements, time the requirements is proposed, cost of the requirements, satisfaction obtain from requirements and how much the requirements impact of quality.

Index Terms— Requirements traceability, Software, Requirements Priority, Requirements.

1. INTRODUCTION

Requirements are foundation of the system and software development.Requirement definition and requirement management is recognized as a core discipline for successful delivery of a software. This discipline also required by standards, regulations and quality improvement initiatives such as Capability Maturity Model Integration (CMMI), Software Process Improvement and Capability Determination (SPICE).

A poorly defined requirements can have negative impact, it can have domino effect that could potentially lead to time consuming rework, inadequate deliveries and budget overruns. Creating and managing requirements are challenge for software development team.Requirements definition and management is an activity with great potential to deliver a higher and faster return on investment (ROI).

Therefore project team must understand the various attributes of a good requirements. A well-defined requirements is:

- **Correct:** A requirement is technically accurate and legally appropriate.
- **Complete:** A requirement presents a complete idea.
- **Clear:** A requirement is clearly defined and not ambiguous.
- **Consistent:** A requirement must not be in conflict with other requirement.
- **Verifiable:** A requirements must be verifiable, the application must be suitable to address requirements.
- **Traceable:** Each requirements is distinctively identified and tracked.
- **Feasible:** A requirements must be effectively addressed within the specified cost and schedule constraints.
- **Modular:** A requirements must be easily changed without excessive effort.
- **Design-independent:** A requirements does not pose specific solution on design.

The whole requirements specifies a desired goal or result and contains a success criterion or other measurable indication of quality.

Following ten steps that help developers at organizations for better define and manage requirements.

1. Structure requirements: Controlling and improving the quality of defined requirements. Duplicate requirements can lead to twice the quantum of work. Omitted requirements may lead to missing functionality or cause shortcomings. **Traceability** to higher- and lower-level requirements enables to assess coverage more effectively.

2. Address and link customer needs, requirements and contracts: Focus on design i.e. data structure, software architecture, procedural detail and interface of requirements. These customer needs to undergo an internal translation. A specific contractual documents must be generated from the requirements repository.

3. Manage Constraints: Requirements must not only describes functional behaviour but also non-functional requirements, which are also known as constraints. Typical non-functional requirementsare Performance, Interface, Security, Safety, Reliability, Availability, and Maintainability.

4. Visualize requirements: Most of the requirements analysts find augmenting textual requirements with modelling helpful i.e. representation. These representations must be managed along with requirements to help ensure consistency, traceability, change control and create common understanding between all development team members and stakeholders.

5. Integrate requirements with your quality management plan: An efficient way to manage requirements is to ensure that they are clearly mapped to test case. Requirements are clearly testable early on in the process, they can more effectively improve project success rates and quality.

6. Prioritize the requirements that deliver the greatest business value: The route to better requirements management is to have fewer requirements. Such effective decision making can be achieved by combining value and priority information.

7. Respond more rapidly but accurately to changing requirements: Requirements are subject to continual change. Scripting a perfect first requirements is inadequate, if its evolution is not well-managed, poorly controlled change can lead to inadequate systems and software rework effort and loss of revenue.

8. Capture and track metrics and trends: Capture all requirements and develop a traceability links between requirements to help decision making, overcome inconsistency and change control.

9. Affordmodels of best requirements: By providing models of best requirements and procedures, project teams can boost the quality, stability and completeness of their requirements.

10. Reuse requirements: When a good requirements has been written for a previous project and is applicable to a present situation. A smaller approach to reusing requirements is to maintain a link between the two requirements. By implementing smart requirements reuse, your project teams can improve knowledge sharing across teams and facilitate impact analysis.

II. RELATED WORK

Requirements definitions and management are among the most important activities in any software development methodology and effort in this regards can help improve and accelerate return on investment (ROI). Requirements definitions and management is also first step process improvement area to focus based on "garbage in, garbage out" rule, if the requirements are not clear, it will produce wrong product.

The efficient way of managing the requirements, controlling and monitoring changing requirements is by creating and maintaining traceability links between requirements. Decision making of requirements for traceability links is performed by requirements prioritization [Refer Fig: 1]. A duplicate of requirements can easily identified by requirement priority.A complete requirement priority is obtained after optimizing prioritized requirements.

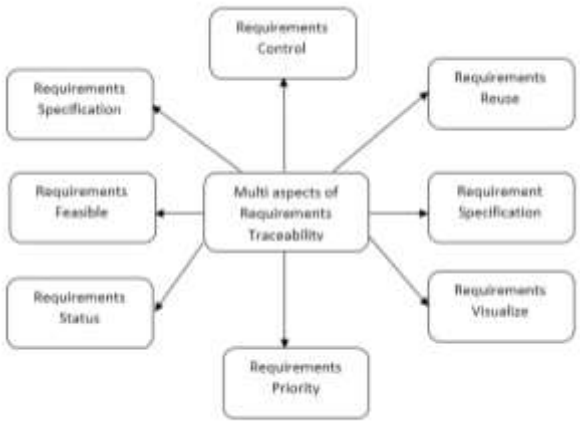

Fig: 1 Multi Aspects of Requirements Traceability

Rather than trying to manage every requirements, project and product managers must able to make decisions related to those requirements that bring the most value to the customer and help improve quality and innovation.

III. RESEARCH METHOD

Requirements prioritization performs an important role in requirement negotiation, software planning and other area of requirement management. The requirement engineers and industry professionals spend a reasonable time in order to justify the requirement prioritization phase.The success of quality software development depends on the right selection of requirements.

During requirements prioritization different factors related to business and technical aspects are considered. These are stakeholder expectations, risk, cost, complexity, time, constraints, dependencies, scalability, sensitivity against errors, contradictory, penalty, volatility, resource, speed, value, effort, approach type, result type, size of requirements granularity, sophistication, perspective, type of technique (manual or algorithmic), number of comparisons, structure, customer importance, strategic importance, expert biases, provision of change of requirements, empirical validation, ease to use, support for consistency index, scales impact, customer satisfaction, marketing, strategic and integrity.

The most essential functionality should be implemented as early as possible. Prioritization techniques should be ease of use, easy of learn, should develop interest in and confidence of the user on the software. Requirements prioritization process is taken as a decision making process as a part of requirement management.

Requirements prioritization approaches can be classified into four different levels, namely: 1.Prioritization activities, 2. Techniques, 3.Methods and 4.Process

1. Prioritization activities: This approach is lowest level. Prioritization activities are used to rank the requirement, separately assign the rank to all requirements, according to the importance of the customer.

2. Prioritization techniques: after ranking the requirements, it's possible to do calculations to order the requirements. Three normal measure (scale) to present the result of ordering. Nominal scale: Produce a lists of categories. Requirements are categorized into groups based on importance. Ratio scale: Produce a rank lists of requirements. Provide relative difference between requirements. Tells how one requirement is important than another, but cannot tell how much important. Ordinal scale: Produce ranked lists. Tell how much one requirement is important than another.

3. Prioritization method: Priority methods are more sophisticated than techniques. While a technique focuses on one aspect (one dimension) to prioritization. Whereas technique focuses on more aspects i.e. more variables. Summary of requirements prioritization techniques with result scale and speed is shown in table 1.

4. Prioritization process: On the highest abstraction level prioritization process refers a description of steps what needs to be done to priority the requirements.

Table 1: Summary of Prioritization Techniques

Techniques	Result scale	speed	Best SuitedForNos.req
Numeral assignment	Nominal	Average	Medium, Large
AHP	Ratio	Slow	Small
Hierarchy AHP	Ratio	Average	Medium, Large
Priority group	Ordinal	Average	Large
Bubble Sort	Ordinal	Average	Small

IV. PRIORITIZATION TECHNIQUE

A finite and complete requirements are taken to do prioritization. Contains both functional and non-functional requirements. To make good decisions we must prioritize requirements both FR and NFR. Prioritization idea is based in that each functional requirements (FR) is related to at least one or more non-functional requirements (NFR). A sample FR of library datasets is listed. Refer Fig.2.

We purpose a simple technique for prioritizing requirements that takes into account (attributes) of frequency of requirements, time the requirements is proposed, cost of the requirements, satisfaction and value. Prioritization works on comparing each requirements. After comparing the all functional requirements, each of functional requirements must compared with non-functional requirements. Before starting FR and NFR must be clearly identified. NFRs gaining more attention, because of mishandling of NFRs have negative impact on FRs. In software development, NFR are not consider then we can expect 60% failure rate or higher on the software product. Refer Fig: 3 Prioritization process.

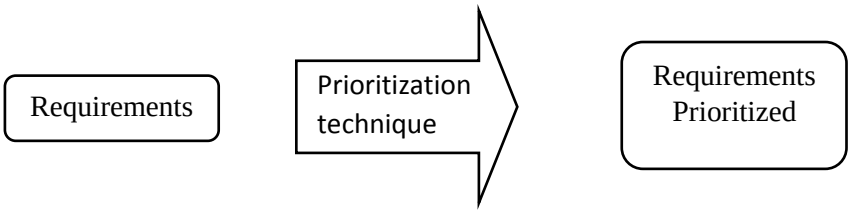

Fig: 3. Prioritization process

The process of prioritization technique supports

- To plan and select an ordered, optimal set of requirements for implementation.
- To estimate expected satisfaction.
- To get technical support and optimize the time.
- To minimize rework and plan stability.
- To provide great value at lowest cost.
-

International Journal of Advance Research in Science and Engineering
Volume No.06, Issue No. 12, December 2017
www.ijarse.com
IJARSE
ISSN: 2319-8354

No	Functions
f1	**Proprietary types of information**: Individual libraries can submit proprietary types of statistical information (e.g. number and types of patrons attending different public activities).
f2	**Manual data entry**: Library staff and other data contributors can enter a limited amount of statistical information via a central web application.
f3	**Shared types of transactions**: National authorities can collect large datasets with shared types of transactions extracted from operational library systems (e.g. Interlibrary Loan requests).
f4	**Proprietary types of transactions**: Libraries can submit large data sets with proprietary types of transactions extracted from operational systems (e.g. anonymized circulation transactions).
f5	**Manual upload of data**: Data contributors can manually upload datasets to a central web application.
f6	**Upload via web service**: Customized local applications can automatically submit data to a central web service.
f7	**Xml documents and schemas**: Submitted datasets must contain xml documents that comply with designated XML schemas (XSD).
f8	**Metadata about datasets** must include information about the origin, ownership, creation date and content plus reference to the types of data objects contained within the datasets.
f9	**Metadata about data objects** must include information about the structure, meaning and validation rules for the data objects.
f10	**Semantic relations**: Metadata must include information about semantic relationships between data elements (e.g. hierarchy and synonyms).
f11	**Metadata maintenance**: Data contributors can maintain their metadata via a central web application.
f12	**Data maintenance**: Data contributors can maintain data objects via a central web application.
f13	**Data Dictionary search**: Library staff and other users may search the data dictionary (metadata) via a central web application.
f14	**Open Access to data**: Access to data objects must generally be unrestricted, but not necessarily gratis.
f15	**Restricted access to data**: Access to certain data objects may be limited to authorized users.
f16	**Data selection**: Users can select data objects by means of a central web application with an intuitive search facility.
f17	**Data reports**: Users can present selected data objects via standard reports produced via a central web application.
f18	**Manual download of data**: Users can download selected data via a central web application.
f19	**Download via web service**: Customized local applications can automatically fetch designated data objects from a central web service, e.g. via the Atom Publishing Protocol.

Fig: 2 A sample FR of library datasets

Requirements Priority Algorithm:

1. f,nf,primax: integer //f-functional nf- non-functional requirements.
2. fr,t,c,s,v: integer // fr-frequency, t-time, c-cost, s-staisification,v-value.
3. Prif[1..n]: integer of FR
4. Prinf[1..n]: integer of NFR
5. Pfnfr[m,n]: integer // two dimensional array contain, priorities provided by comparison of FR and NFR. // n columns contains FR,// priority value, another is NFR.
6. Begin
   //we compare each FR with NFR related to frequency, time, cost, staisification, value.
7. For i=1 to n[f] do // compare of f
8. For j=2 to m[f] do
9. Primax=pfrnfr[1][i]
10. If pfrnfr[j][i]>primax then
11. Primax=pfrnfr[j][i]
12. do
13. For i=1 to n[t] do // compare of t
14. For j=2 to m[t] do
15. Primax=pfrnfr[1][i]
16. If pfrnfr[j][i]>primax then
17. Primax=pfrnfr[j][i]
18. do
19. For i=1 to n[c] do // compare of c
20. For j=2 to m[c] do
21. Primax=pfrnfr[1][i]
22. If pfrnfr[j][i]>primax then
23. Primax=pfrnfr[j][i]
24. do
25. For i=1 to n[s] do // compare of s
26. For j=2 to m[s] do
27. Primax=pfrnfr[1][i]
28. If pfrnfr[j][i]>primax then
29. Primax=pfrnfr[j][i]
30. do
31. For i=1 to n[v] do // compare of v
32. For j=2 to m[v] do
33. Primax=pfrnfr[1][i]
34. If pfrnfr[j][i]>primax then
35. Primax=pfrnfr[j][i]
36. end

V. CONCLUSION

In this paper, we have presented a new requirements prioritization process by considering functional and non-functional requirements. The goal of this work is to aid decision making about requirements prioritization. Requirements prioritization process improves the decision making quality to support the software development.

Thus, this technique permits to the analyst to control and monitor the requirements, to make the good decisions, and to early find out the ambiguities and the conflicts if they are exists.

REFERENCES

[1] Dunhui Yu, Keqing He, Jian Wang, "An algorithm for priority ranking of individualized functional requirements in networked software", IEEE,2008.

[2] Richard R. Maiti, Frank J.Mitropoulos, "Capturing, Eliciting,Predicting and Prioritizing (CEPP) Non-Functional Requirements Metadata during the Early Stages of Agile software Development",IEEE,2015.

[3] Falak Sher, Dayang N.A. Jawa wi, Radziah Mohamad,Muhammad Imran Babar, "Requirements Prioritization Techniques and different Aspects for Prioritization", IEEE, 2014.

[4] Ritu , Dr. Nasib Singh Gill," A Comparison among Various Techniques to Prioritize the Requirements", PP no-601-607,ol no-12,2012.

[5] G. Spanoudakis, A. d'Avila Garcez, A. Zisman," Revising Rules to Capture Requirements Traceability Relations: A Machine Learning Approach", IEEE, 2014.

[6] Wentao Wang, Nan Niu, Hui Liu† and Yuting Wu, "Tagged in Assisted Tracing", IEEE, pp no-8-14, 2015.

[7] Swathine.k and J.KomalaLakshmi,"Requirement Elicitation for Requirement in Software Engineering", International Journal Of Engineering Sciences & Research Technology, pp no- 354-357,2014.

[8] Swathine.k and J.KomalaLakshmi "A Survey of Defect Prediction on Software Leads to Quality Product", International Journal of Computer Science & Engineering Technology,pp no-1038-1040, Vol-5,2014.

[9] Falak sher, Dayang N.A, Jawawi, Radziah Mohamad, Muhammad Imran Babar, "Multi-aspects Based Requirements Prioritization Techniques for Value-Based Software Development", IEEE,2014.

[10] "Ten steps to effective Requirements Management", IBM Omar Badreddin, Omar Badreddin, Timothy C. Lethbridge,"Requirements Traceability: A model driven approach", pp no-87-91,IEEE,2014.

[11] Dunhui Yu, Keqing He and Jian Wang, "An AlgorithmPriority Ranking of Individualize Functional Requirements Is Networked Software", Seventh International Conference on Grid and Cooperative Compting, pp no-701-706,2008.

[12] Lan LIN, Jesse H. POORE,Front. Comput. Sci. China "Pushing requirements changes through to changes In Specifications" pp no-332-343,2008.

[13] Mikko Vestola. "A Comparison of Nine Basic Techniques for Requirements Prioritization".

[14] Rao Muzamal Liaqat, Farooque Azam, Mudassar Adeel Ahmed, Bilal Mehboob,"A Majority Voting Goal Based Technique forRequirement Prioritization"

[15] Masooma, mohammad ubaidullah bokhari, Md zeyauddin "An analysis of software requirement prioritization:
Detailsurvey", , pp no-3966-3970 IEEE 2016.

[16] Abdel Ejnioui, Carlos E. Otero,Abrar A. Qureshi."Software Requirement Prioritization Using Fuzzy Multi-
attribute Decision Making",

[17] "Requirements Prioritization in Software Engineering: A Systematic Mapping Study",Massimiliano Pergher,
BrunoRossi, pp no-40-44 IEEE 2013.

[18] Mohammad Dabbagh, Sai Peck Lee, Reza Meimandi Parizi,"Application of Hybrid Assessment Method For Priority Assessment of Functional and Non-Functional Requirements.

[19] Perini, A., Susi, A., and Avesani, P.: 'A Machine Learning Approach to Software RequirementsPrioritization',
IEEE Trans. on Software Engineering, 2013, 39, (4), pp. 445-461.

[20] Fadoua Fellir, Khalid nafil, Rajaa Touahni,"System Requirements Based Priority: AHP:, IEEE Trans, 2014,
pp-no- 163-167.

[21] Philip Achimugu, Ali Selamat, RolianaIbrahim, Mohd Nazri Mahrin, A Systematic literature review Software requirements Prioritization research,Elsevier Information and software Technolog,2014,pp-no-568-585.

[22] Rubaida Easmin, Alim Ul Gias, shah mostafa," A Partial Order Assimilation Approach for Software Requirements Prioritization", IEEE 3rd international Conference on informatics, electronics & vision 2014.

[23] Jim aza, Randy K.Smith , David Cordes,"SE challenges In small companies"IEEE,2007,pp no-32-37.

[24] Md. Rizwan Beg , Qamar Abbas , Ravi Prakash Verma,An Approach of Requirements Prioritization, First IEEE International Conference on Emerging Trends And Technology,208,p no-1216-1222

[25] Swathine.k, Sumathi.N, "Study on Requirement Engineering and Traceability Techniques in software Artefacts",International Journal of Innovative Research in Computerand Communication Engineering,
vol 5, Issue 1, January, 2017,pp no-114-121.

[26] Mohd, Sadiq, Mohd Shadid, "An approach for Eliciting software requirements priority in AHP", 2009 IEEE
International Conference on Communication and Computing, pp no-790-796.

[27] Balsam A. Mustafa, Azlina Zainuddin, "An Experimental Design to Compare Software Requirements Prioritization Techniques.