American Journal of Computer Technology and Application
Vol. 3, No. 1, April 2015, pp. 1 - 11, ISSN: 2327-2325 (Online)
Available online at www.ajcta.com

Research article

Agile Methodology and Role & Responsibilities process Model Comparison

Saima Tasneem

Faculty of Computer Science, Indus International Institute DG Khan
E-mail: Saima_tasneem@yahoomail.com

Muhammad Ali Farooq

Dept. of Computer Science, university of agriculture, Faisalabad, Pakistan

Nadeem Iqbal

Faculty of Management Science, Ghazi University DG Khan
E-mail: drnadeemiqbal1@gmail.com

Abstract

Agile methodologies, that are most suitable in dealing with volatile business requirements, are likely to face the same challenge as they require developers to drastically change their work habits and acquire new skills. The Role and responsibility Process is one of a number of agile methods for developing software, and a part of the Agile Alliance. It is a disciplined approach to assigning and managing tasks and responsibilities in a development organization. The goal of this process is to produce, within a predictable schedule and budget, high-quality software that will be capable of meeting the needs of its end users. In the local cottage industry of Pakistan, because of the low budget, small team and short time limit, it is almost impossible to follow all standard procedures of model for a successful software application development. The aim of this study is to modify the model according to the needs of local industry of Pakistan.

Keywords: Process Model, Role and Responsibility, Software Engineering,motion control, mobile robot

I. Introduction

There is increasing demands on software development teams to deliver working systems to business in very shorter time periods. Agile software development practices have increasingly been adopted to respond to the challenges of volatile business requirements, where the markets and technologies evolve rapidly and present the unexpected (Pikkarainen*et al*, 2008). Representatives from the agile development movement claim that agile ways of developing software are more fitting to what is actually needed in industrial software development. If this is so, successful industrial software development should already exhibit agile characteristics (Hansson*et al*, 2006).

It is widely believed that Systems Development Methods (SDM) can help improve the software development process. Nevertheless, their deployment often encounters resistance from systems developers. Agile methodologies, the latest batch of SDM that are most suitable in dealing with volatile business requirements, are likely to face the same challenge as they require developers to drastically change their work habits and acquire new skills (Chan and Thong, 2009).

The Role and Responsibility Process is a comprehensive process model that is tailor able, provides templates for the software engineering products, and integrates the use of the Unified Modeling Language (UML). It is rapidly becoming a de facto standard for developing software. The process supports the definition of requirements at multiple levels (Cooper *et al*, 2006). The Role and Responsibility Process captures many of the best practices in modern software development and presents them in a tailor able form that is suitable for a wide range of projects and organizations. The Role and Responsibility Process delivers these best practices to the project team online in a detailed, practical form (Kruchten, 2003)

II. Material and Methods

A survey of existing literature is used to analyze that agile methods are often seen as providing ways to avoid overheads typically perceived as being imposed by traditional software development environments. However, few organizations are psychologically or technically able to take on an agile approach rapidly and effectively (Qumer and Sellers, 2008).

It is widely believed that Systems Development Methods (SDM) can help improve the software development process. Nevertheless, their deployment often encounters resistance from systems developers. Agile methodologies, the latest batch of SDM that are most suitable in dealing with volatile business requirements, are likely to face the same challenge as they require developers to drastically change their work habits and acquire new skills (Chan and Thong, 2009).

The Role and Responsibility Processis a comprehensive process covering almost all aspects of software development projects.However, due to the great level of detail provided by Role and responsibility Process, many professionals do not consider RUP practical for small,fast paced projects. Role and responsibility Process proved to be adaptable to the needs of small projects and was very effective in both projects. One key to the successful application of Role and Responsibility Process in small projects is the careful selection of a proper subset of artifacts and keepingthese artifacts very concise and free from unnecessary formalism.

III. Results and Discusions

In this research focus was to study the exiting requirement management techniques, and try to create a new one for the local industry of Pakistan.

The need of the new requirement management was felt because there are a lot of organizations that are running their own in-house software development, but are facing problem with a lot of requirement management activities.

To build successful complex software systems, developers must collaborate with each other to solve issues. To facilitate this collaboration, specialized tools, such as chat and screen sharing, are being integrated into development environments. Currently, these tools require a developer to maintain a list of other developers with whom they may wish to communicate and to determine who within this list has expertise for a specific situation. For large, dynamic projects, like several successful open-source projects, these requirements place an unreasonable burden on the developer (Minto and Murphy, 2007).

The reasons for requirement management problem with in-house teams was due working without following any requirement management techniques. The major reason of not using any management technique was extra budget and skilled person required. We already have mentioned this issue. So the need was felt that there must be a new management technique which should be constructed keeping in view the environment and resources of the in-house software development. As the end users of the system are the employees of the company itself and every employee is very well aware of his responsibilities and he can explain his requirements regarding system very well. The problem is just to keep it documented in such a way that system have fully traced back and forward it.

IV. REQUEST INITIATOR

End user will initiate the request to accomplish it against one of his responsibilities. User will describe the type of requirement request then he will describe the requirement in detail. Figure blow depicts the detail

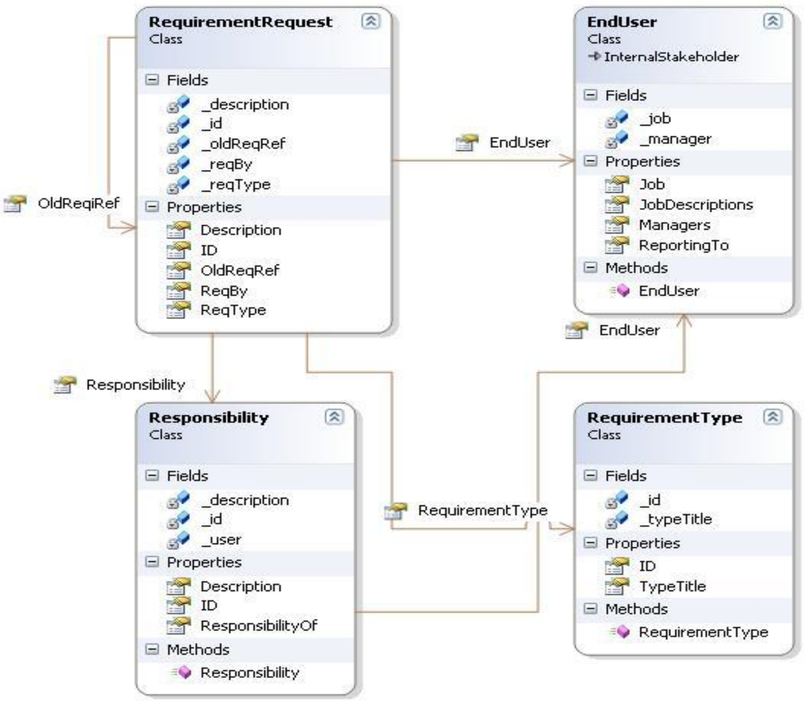


Figure 1. Request Initiator class diagram

V. REQUEST AUTHORIZATION

When requirement request posted by the end user, his manager or authorized person will check it against the responsibility list of the employee. If requirement request fulfill the responsibility check list and requirement is verified by the authorized person then he can forward it to development team lead or manager

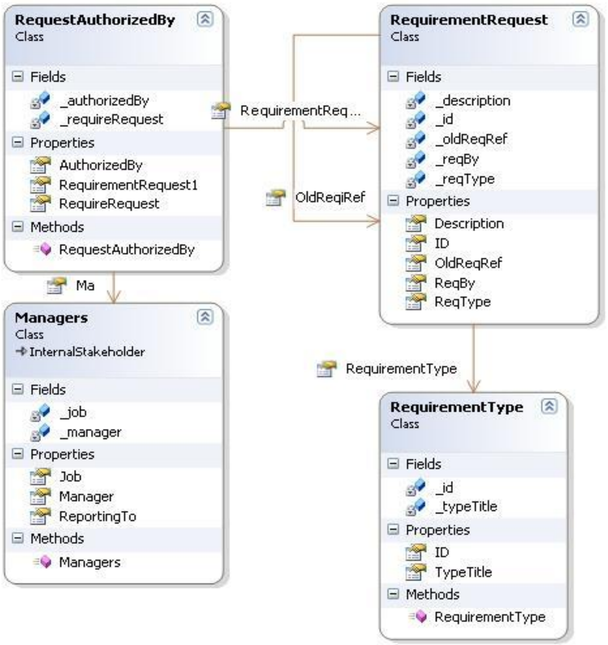


Figure 2. Request Authorization class diagram

VI. REQUIREMENT DEVELOPMENT

Once the requirement request will receive by the development team lead he will cross check it with the responsibility check list of the employee role; He will also check the cross reference of the authorized person of the requirement request. Once everything will check and balance then the development team will start analysis on the requirement request. As before said the customer is actually the employees of the company so there will be no confusion in detail of the requirement request. If some ambiguity find in the requirement then it will be discussed with the end user and the authorized person both

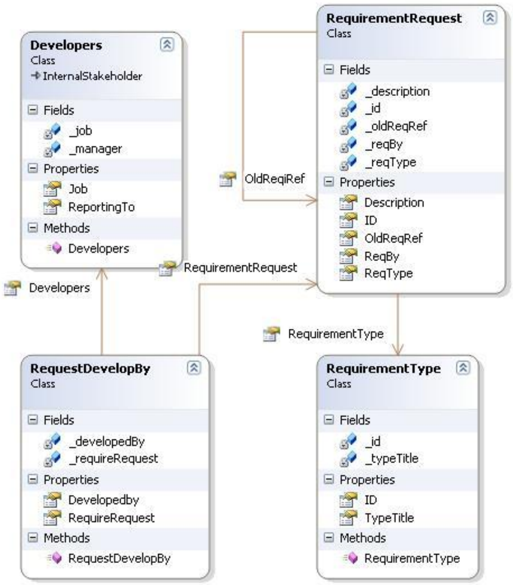


Figure 3. Requirment development class diagram

VII. REQUEST TESTING

Once requirement will developed, it will forward to the end user from where request initiates for the testing the requirement. Again as the employees are on board there will be no confusion if they find any problem with the required request

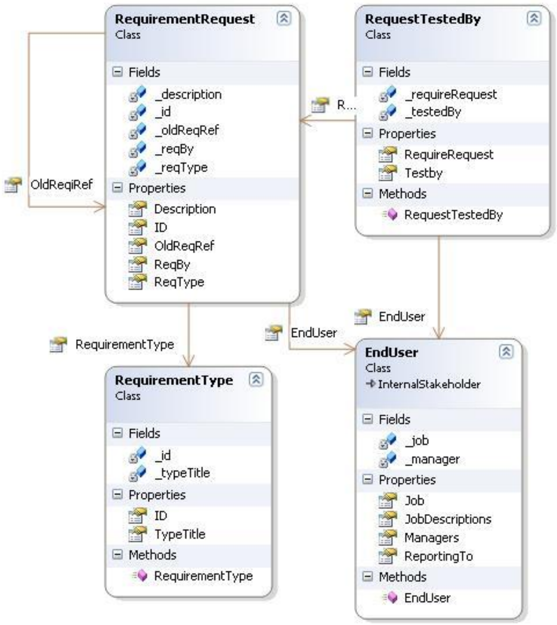


Figure 4. Request Testing class diagram

VIII. REQUIREMENT TRACEABILITY ORBIT

Requirement traceability object will use to trace the requirement at any stage of the software development life cycle. Traceability object will contain the actual requirement request and the end user who initiate the request. Requirement request will contain the role and the responsibility matrix of the end user. Traceability will also contain the authority object and testing object to describe the detail of authorized person and the tester

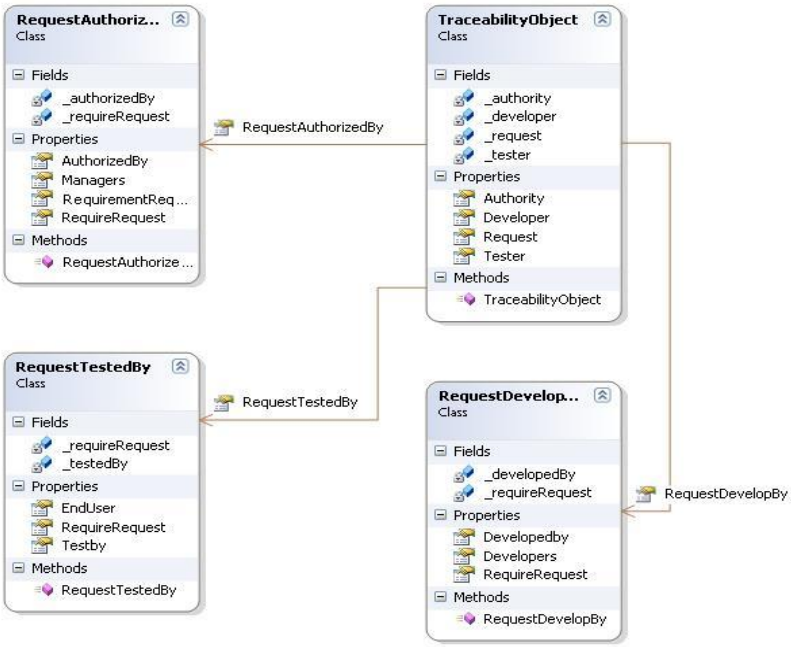


Figure 5. Requirement Tracability Orbit class diagram

Role and Responsibility Model

Every user has some role and particular responsibility. Every end user provides the list of his responsibilities and his or her manager will verify it. Once the list of responsibilities is completed then every user can request requirement against his responsibility. Once the requirement is requested it will pass to reporting manager to verify the requirement against the responsibility. If the request is verified then it will transfer to the development department. After receiving the request, developer will check the authorized check and after verifying developer will start working on the requirement request. Once the requirement request is completed it will transfer to the end user or to tester to test the request against the requirement. When the requirement is completely tested then it will be implemented. During all this process, requirement can be traced at any level. Traceability object is not only used to check the status of the requirement request but also to trace back the requirement if any fault or confliction is found in the system.

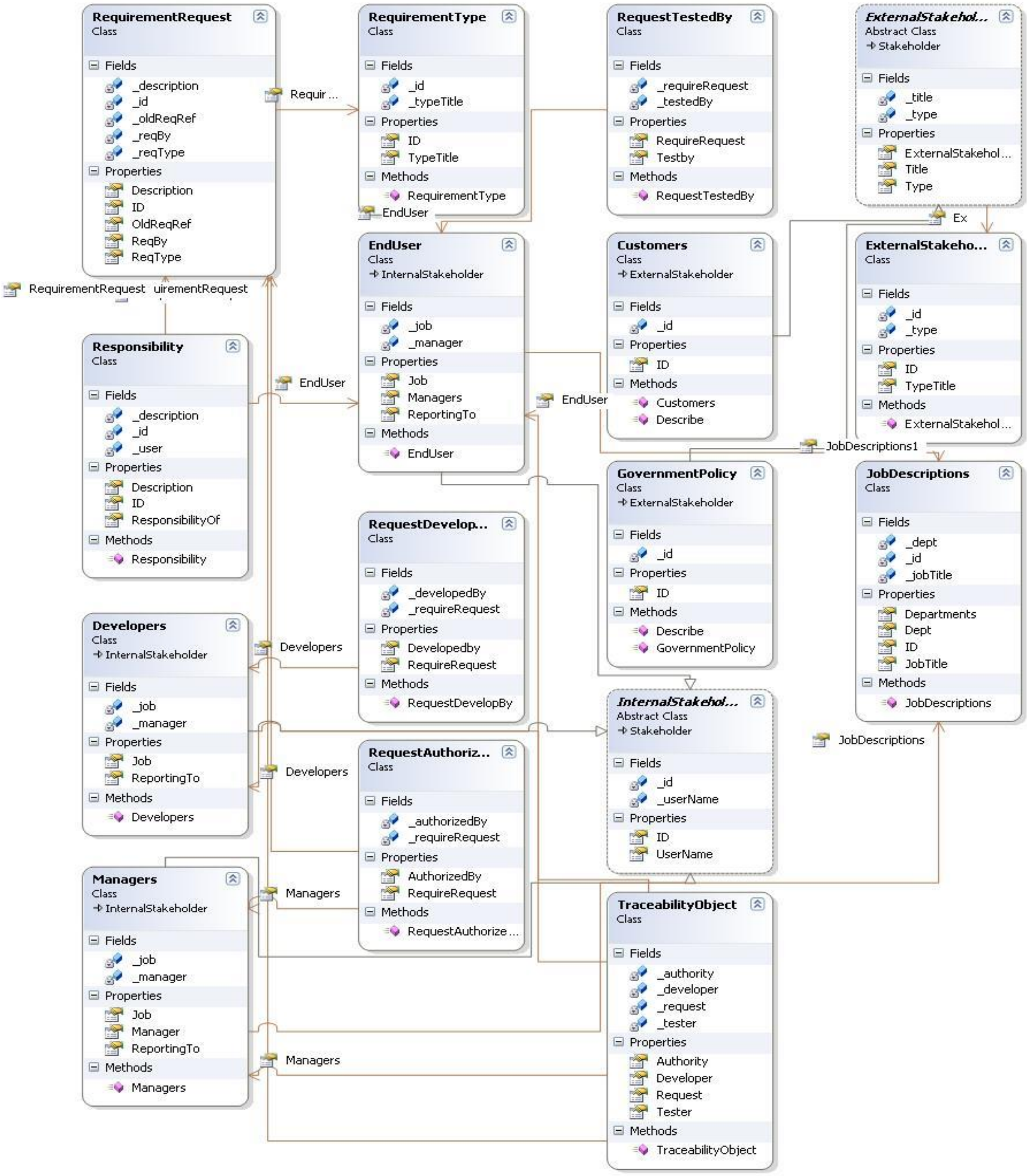


Fig. 6: predicts the whole picture of the model

The technique presented above is implemented on a project for requirement engineering. The technique is tested on different metrics with another project's requirement gathering and management technique. In this research all the metrics discuss one by one and also test its significance with other technique.

Lines of text

First metric is the lines of text; total number of lines of each requirement consists of. Table gives the results.

Measurement of Lines of Text of two projects

Statistical Analysis of Requirements	Lines of Text	
	Project A	Project B
Minimum	12	13
Maximum	80	95
Average	47	39
Stdev	9	14
Total	350	290

If we calculate the confidence intervals of the both projects then we have,

95% Confidence Interval for Project A is 26.03 to 67.07.
95% Confidence Interval for Project B is 6.38 to 71.62.

The results clearly show that technique is very much significant as compare to other. There are 95% chances that average of lines of text will fall between 26 and 67 lines. But the other techniques interval length is very much large then the first one which is clearly less significant than the first technique.
There are some other metrics. Table shows the results

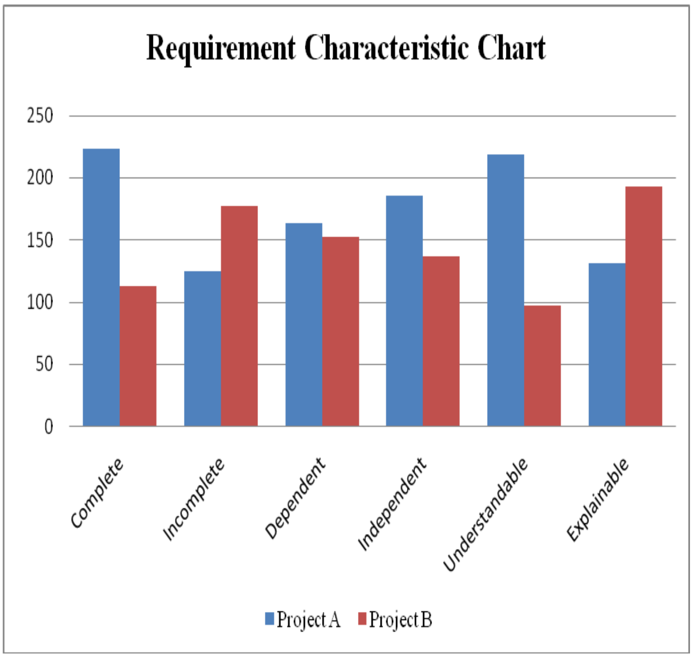
Metrics of Requirements for Both Projects

Metrics	Project A	Project B
Complete	223	113
Incomplete	125	177
Dependent	164	153
Independent	186	137
Understandable	219	97
Explainable	131	193

Table and graph shows the difference between both techniques. If only the completeness is considered here then it is very much clear that technique A gives more complete requirements then technique B. For testing the significance of both techniques, statistics pair sample test is used i.e

$$Z = \frac{p_1 - p_2}{\sqrt{(\frac{p_1 q_1}{n_1} + \frac{p_2 q_2}{n_2})}}$$

According to the values P_1 is 0.64 and P_2 is 0.36 and after testing the values the significance value is got 6.24 which is larger then critical value 2.575 which is very very much significant.



Comparison of Project A and Project B on Metrics

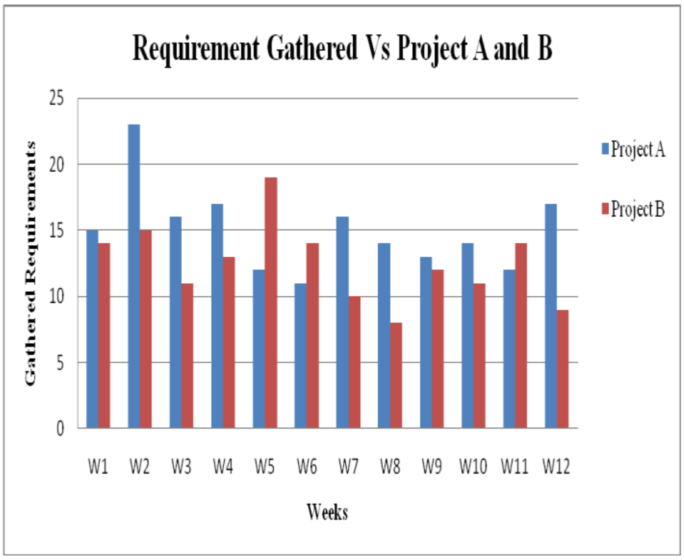
Similarly understandablity of both techinque is tested with the same paired test method and the test showes value of 7.68 which is very very much signficance. Only the dependablity of both techniques is almost equal. This is the only metric which is found less signifcant as compare to others.

Another metric which is noted, number of requirements gethered in a timebox. The term timebox is considered as an interval or interation. Table shows the results.

Values of Project A and Project B on Time Boxes

Time Boxes	Project A	Project B
W1	15	14
W2	23	15
W3	16	11
W4	17	13
W5	12	19
W6	11	14
W7	16	10
W8	14	8
W9	13	12
W10	14	11
W11	12	14
W12	17	9

After twelve week total number of requirements gathered by technique A is 180 and by technique B is 150.



Effect of Project A and Project B with Time intervals

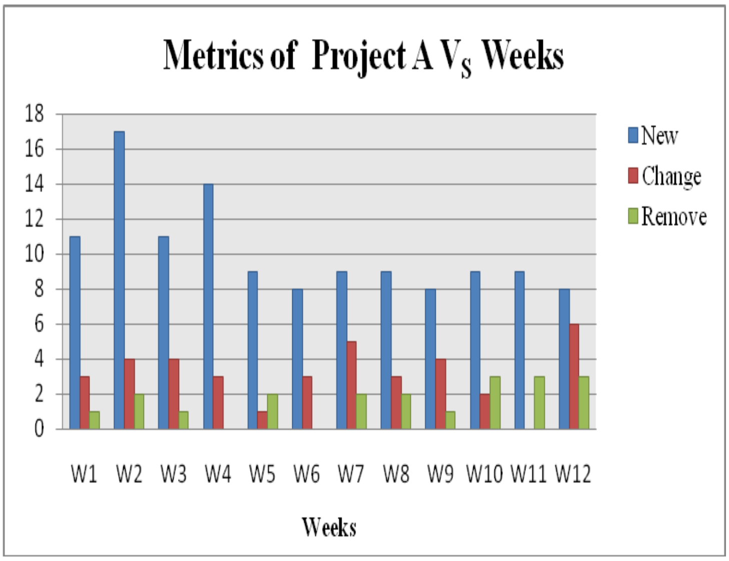
Graph represents the requirement gathering by technique A is much faster then technique B. So technique A is much significance as compare to technique B.
Further this metric is divided into three different metrics based on the requirement type i.e
New: For new requirement and not exists in the system before.
Change: To change the existing requirement.

Remove: To remove the existing requirement.

Different Metrics Based on the Requirement Type

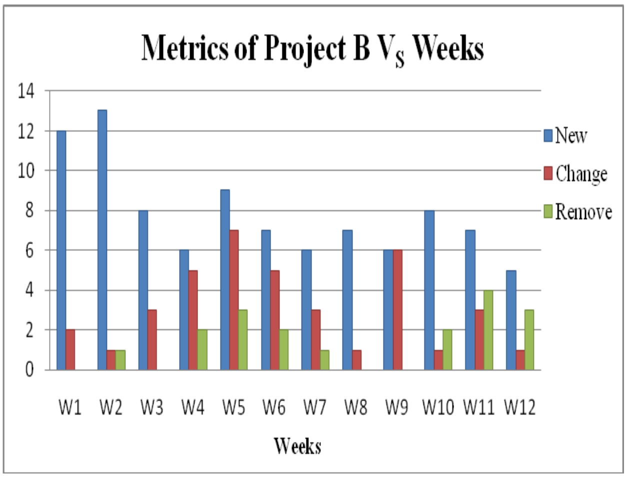
Weeks	Project A			Project B		
	New	Change	Remove	New	Change	Remove
W1	11	3	1	12	2	0
W2	17	4	2	13	1	1
W3	11	4	1	8	3	0
W4	14	3	0	6	5	2
W5	9	1	2	9	7	3
W6	8	3	0	7	5	2
W7	9	5	2	6	3	1
W8	9	3	2	7	1	0
W9	8	4	1	6	6	0
W10	9	2	3	8	1	2
W11	9	0	3	7	3	4
W12	8	6	3	5	1	3
Total	**122**	**38**	**20**	**94**	**38**	**18**

Graph shows the rate of arrival requirement of technique A which very much clears that arrival of new requirement is large enough as compare to change and remove.



Rate of Arrival of New Requirements of Technique A

Graph shows the rate of arrival requirement of technique B which very much clears that arrival of new requirement is large enough as compare to change and remove. But the point is noted here that arrival of new requirements by technique A is looked like greater then technique B.



Rate of Arrival Requirement of Technique B

IX. CONCLUSION

A requirement is an agreed standard of need which is used for a particular product to use for performance. Normally it is used in computer system or software engineering as formal. Each requirement is referred as to necessary attribute, capability, characteristic, or quality of a system. Software plays different functions which are key players of the software engineering. A set of functionalities performed by a software system are called its functional requirements.

There is also vital role of non-functional requirement in engineering which are key elements for software system. Non Functional Requirement in Software engineering presents a systematic and pragmatic approach to building quality into software system. Systems must exhibit software quality attributes such as accuracy, performance, security, safety, availability, maintainability, safety and speed for result and conclusion. 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 behavior, and to their evolution over time and across software families. A framework which is used to plan and control the process of development is known as software development methodology.

REFERENCES

[1] Belani, H., Z. Car and A. Caric. 2009. RUP-based process model for security requirements engineering in value-added service development, IWSESSICSE Workshop on Software Engineering for Secure Systems, pp:54-60.

[2] Chan, F. and Y. K. Thong. 2009. Acceptance of agile methodologies: A critical review and conceptual framework, Decision Support Systems, Elsevier Science Publishers B. V. 46(4): 803-814.

[3] Cooper, K., S. P. Abraham and R. S. Unnithan. 2006. Integrating visual goal models into the Rational Unified Process, Academic Press, Inc, 17(6): 551-583.

[4] Gotterbarn, D. 2004. UML and agile methods: in support of irresponsible development, ACM SIGCSE Bulletin, 36(2): 11 –13.

[5] Hansson, C., Y. Dittrich, B. Gustafsson and S. Zarnak. 2006. How agile are industrial software development practices? Journal of Systems and Software, 79 (9): 1295-1311.

[6] Kruchten, P. 2003. The Rational Unified Process, Addison-Wesley Longman Publishing Co., Inc, Volume 3, pp:280-282.

[7] Louvieris P., J. P. Perry and P. B. Davies. 2004. A/B Dashboard: The Case for a Virtual Information Systems Development Environment to Support a RAD Project, IEEE Computer Society, Volume 8, pp: 8026-8032.

[8] Manzoni, L.V. and R.T. Price. 2003. Identifying Extensions Required by RUP (Rational Unified Process) to Comply with CMM (Capability Maturity Model) Levels 2 and 3, IEEE Transactions on Software Engineering. 29(2): 181 – 192.

[9] Misra, S. C., V. Kumar and U. Kumar. 2009. Identifying some important success factors in adopting agile software development practices, Journal of Systems and Software, Elsevier Science Inc. 82(11): 1869-1890.

[10] Pikkarainen, M., J. Haikara, O. Salo, P. Abrahamsson and J. Still. 2008. The impact of agile practices on communication in software development, The Empirical Software Engineering, Kluwer. 13(3): 303-337.

[11] Qumer, A. and B. H. Sellers. 2008. A framework to support the evaluation, adoption and improvement of agile methods in practice, Journal of Systems and Software, Elsevier Science Inc. 81(11): 1899-1919.