Comparative Study of Traditional Requirement Engineering and Agile Requirement Engineering

Asma Batool , Yasir Hafeez Motla , Bushra Hamid , Sohail Asghar, Muhammad Riaz , Mehwish Mukhtar, Mehmood Ahmed

University Institute of Information Technology, PMAS. Arid Agriculture University, Rawalpindi, Pakistan

asma.sadiq2003@gmail.com, yasir@uaar.edu.pk, bushrakiani@uaar.edu.pk, sohail.asghar@uaar.edu.pk, cyberriaz@yahoo.com, mehwish.mukhtar289@gmail.com, mehmoodahm@yahoo.com

***Abstract*—Traditional RE and Agile RE are two different approaches on the basis of their planning and control mechanism. This Paper distinguishes the Traditional RE and Agile RE. Furthermore it investigates the reasons for which software industries shifted from Traditional RE to Agile RE. Research is carried out by conducting a literature study and finally a case study of software development to evaluate which approach has better success rate than other. With the help of our finding and results we have evaluated that Agile RE performs better than Traditional RE in large organizations where changes evolve throughout the development phase of software life cycle.**

***Keywords*— Requirement Engineering , Traditional Requirement Engineering, Agile Requirement Engineering, Scrum, Extreme Programming, Object Oriented Development, Requirement Elicitation, Requirement Analysis, Requirement Management, Software Requirement Specification.**

I. INTRODUCTION

Requirement Engineering is a branch of software engineering that applies different techniques and methods for the requirement analysis, client's requirements are ensured by requirement engineers, developers while analysts understand the software to be developed through these requirements and they are also fully aware of different features and limitations of the system being developed [4]. Traditional Requirement Engineering follows the documentation for the system development that is constructed by gathering all system requirements from the clients at very early stage. In the traditional waterfall model requirement gathering is completed at very early stage and developers have to achieve all functions of the system in a cycle. This creates a very difficult situation when clients want some changes in given requirements [6].

Traditional Requirement Engineering (TRE) process is a complicated process whereas the real life development in the software industry demands a flexible and speedy process. For the success of projects, an efficient Requirement Engineering (RE) process is always required [3].

Agile Requirement Engineering (ARE) is an emerging area that makes the requirement engineering process more flexible and consequently quicker. It offers an iterative, incremental framework that helps the teams to focus on rapid delivery of products. It provides the benefit of constant communication between customers and developers. As a result the developers can deliver the system in time which conforms to that surely satisfies the customer's expectations, and hence increase the business value [9].

Agile Requirement Engineering has proven the success of projects due to different reasons. Firstly, it makes the system transparent that is visible to the clients. Secondly, it establishes a proper face-to-face communication that makes requirements more clear [1].

In Traditional Requirement Engineering process development team gave great importance to documentation while in the Agile Requirement Engineering it concentrates on as much in face-to-face communication. Moreover customers can interact only in early stage of Traditional Requirement Engineering process but in Agile Requirement Engineering customers can freely communicate with the team throughout the software development life cycle [7].

Requirement engineering is a critical phase in software development. Gathering, understanding and managing the requirements are difficult features of all development methods but since agile practices are based on fully understand the requirements by constant and continue communication with the customers, it prioritizes the requirements, implements them and satisfies the needs and expectations of customers.

II. TRADITIONAL REQUIREMENT ENGINEERING

Traditional Requirement Engineering involves some important features that are elicitation, analysis, documentation and managing of the requirements [3, 5, 7].

A. Requirement Elicitation

In requirements elicitation process requirements and system boundaries are discovered by consulting with stakeholders. System boundaries describe the overall vision of the system. Hence it is necessary to understand the system for its better implementation. For the requirements elicitation different techniques are used. Some of them are prototyping, brainstorming, interviews, reviews of the use cases etc.

B. Requirement Analysis

The goal of Requirements Analysis is to check that elicited requirements are consistent, complete and feasible. Requirements are refined through negotiation process.

1) ***Requirements Prioritization:*** Requirements are required to be prioritized to satisfy some of the limitations, these are time, resources and technical capabilities. When customer expectations are high and timelines are short, you need to make sure that the product delivers the most valuable functionality as early as possible and it function under the pressure and limitation. Requirements should be prioritizes with the help of the stakeholders.

2) ***Modelling:*** Requirements are modelled to make it more clearly for the developers. Models make requirements easy to understand. Different techniques are used for system modelling to describe system requirements. Some of them are data-flow models, semantic data models and object-oriented approaches.

C. Requirements Documentation

In the Requirements Documentation process requirements are written in some consistent, accessible, and reviewable way. Requirement document is the baseline for specifying all types of functional and non-functional requirements. SRS template is used for requirement documentation which provides a consistent structure for specifying the requirements and their related information.

D. Requirements Validation

Validation checks that the requirement statements are consistent, complete and they demonstrate the desired characteristics and satisfy the customer's needs. Test cases are written to reveal the ambiguities and vagueness in the written requirements. Problems identified in the process of validation must be correct for the better and reliable system implementation.

E. Requirements Management

In the requirements management process, all the information about requirements is stored and managed by the requirements management team. Requirements management is concerned with all the changes needed to be done on the agreement of stakeholders. There is a change control board that consists of key stakeholders which decides the proposed changes to incorporate. In addition to this, they evaluated that each proposed requirement change is to determine the effect on the project.

III. AGILE DEVELOPMENT

Agile Methods consists of software development processes that have become popular during the last few years [8]. They play a great role in delivering products faster, with conformance of high quality, and satisfy the high expectations and needs of customer through its flexible principles of the agile software development [11].

A. Agile Principles

Although agile techniques are slight different but they work on the same principles as described in the agile Manifesto [10].

- Delivery of working software is the highest priority.
- Agile processes greet the changes requirements even late for the customer satisfaction.
- Delivery of working software is recurrently in shorter time scale.
- Collaboration and communication of business people and developers must be in the whole time span of the project.
- Projects are built about aggravated individuals those are trust worthy.
- Face-to-face conversation is very proficient and valuable method for transmitting information between customer and development team.
- Working software is the most important measure of progress.
- Simplicity.
- Permanent concentration to technical excellence and good design.
- Self-organizing teams play a vital role in the best architectures, requirements, and designs.
- Team decides how to become more efficient by regulating its behaviour.
- Agile processes endorse prolonged development.

B. Agile values

The Agile Manifesto acknowledges a common set of values and principles for every individual agile methodology [12].

1) ***Individuals and their interactions:*** Individuals and interactions are very important for high performance. For better communication agile rely on frequent inspect and adopt cycles. These cycles consist of pair programming, continues integration. But these cycles work well when team members adopt some key behaviors like respect for the value of the team members. Every communication is based on reality, all data, actions and decisions must be clear, have faith on

every person who supports the team and shows dedication to the team and its goals. If any team members evade truth and transparency due to cultural norms or any other reasons, it produces conflicts and problems in honest communication. For rapid and high performance it is necessary to adopt some key factors like trust, truth, transparency, honesty for the betterment of communication.

2) ***Delivering working software:*** Its aim is to deliver small pieces of working software in the set intervals. Working software is the difference that agile brings. All features of these working software pass through all the tests by team. Some agile teams perform unit test and system level test but some best agile teams also perform integration test and acceptance test. These tests increase the value of products by decreasing their defect rate. The Agile Manifesto emphases on delivering the software in set intervals to bring high performance and increases the level of trust of customer.

3) ***Customer collaboration:*** Delivering the working software allow the customer to provide the feedback actively. Agile manifesto writes and thinks that active participation of customer is very necessary for the success of the product. Customer collaboration is the key factor for the success of the agile processes. So agile methodologies are more successful and agile teams provide a specific environment for the communication of customer and developers.

4) ***Responding to change:*** Responding to change is very important for a product that pleases the customers. As customers never know until they see the working software. Traditional project finishes in budget and time but it leaves the customer unhappy because actually what they have received is not exactly what they required. Agile methodologies change their plans at regular basis according to customer feedback it will enhance the business value and customer satisfaction towards its product.

IV. EVALUATION OF THE EXISTING FRAMEWORKS

We have taken two proposed frameworks, a linear iterative traditional requirement engineering model [16] and an agile collaborative and innovative framework [2]. We have elaborated them to make a comparison between agile RE and traditional RE.

A. Traditional Linear Iterative Requirements Engineering Model

In Figure 1, Loucopoulos and Karakostas have proposed the linear iterative requirements engineering process model.

This model consists of three main phases that are elicitation, specification and validation. Users can only involve in elicitation and validation. Requirements are gathered during elicitation and on the basis of these requirements problem domain of system is developed. This model is best for performing requirements in iteration and is preferred delivery of software acceptable in successive versions [16].

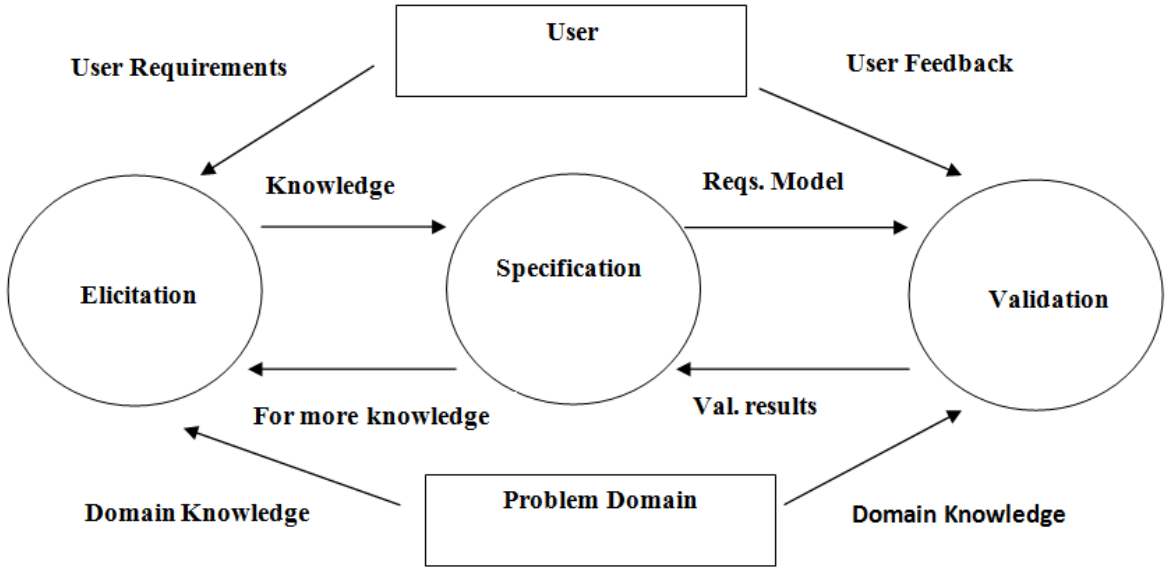

Figure 1. A traditional linear iterative requirements engineering model [16]

B. An Agile Collaborative and Innovative Framework

In figure 2, Veerapaneni Esther presented a Collaborative and innovative framework that uses the features of two agile approaches Scrum and XP. It is recommended that large organizations may adopt this due to its hybrid approach and companies that have important effect on efficiency of the project [2]. All agile methodologies have iterative and flexible frameworks that make the requirement engineering process more effective and efficient. A hybrid approach consists of scrum and XP methodologies. XP and scrum are gradual but different approaches. Scrum is a project management approach but XP is in development side [13]. Agile methods contain the features like refactoring and effective prototyping can also effectively deal with unstable requirements [14]. Scrum experts in management side and XP trust on test-first programming in which minor errors are detected at a very early stage and also at the time of integration [15].

V. MAPPING OF TRADITIONAL RE AND AGILE RE ROLE, ACTIVITIES & ARTEFACTS

In Table I, Traditional RE has mapped to Agile Scrum RE with respect to their roles, activities and artifacts.

- Traditional RE process engages the domain experts, stakeholders and requirement engineers for the requirement elicitation. Their responsibility is to discover the requirements through different techniques. In Agile RE product owner describe requirements to development team and finalizes them in the form of product backlog list.
- Product owner confirms the requirements from customer and ensures that they are complete and consistent. Requirement prioritization and modelling is also done in the requirement analysis phase. Prioritization is done in selected product backlog list in scrum approach through product owner. He has the right to cancel the prioritized, reprioritized all the requirements.
- Requirement document is created by developers for future development in Traditional RE. Agile RE develops only the concise documentation for future. Issues are resolved through meetings.
- In Traditional RE requirements are validated by developers and customers. Agile RE engages the customer throughout the development process and requirements are validated at every stage.
- Requirement management deals with the changing requirements done by the Change Control Board in Traditional RE. Agile handles the evolving requirements at any stage.

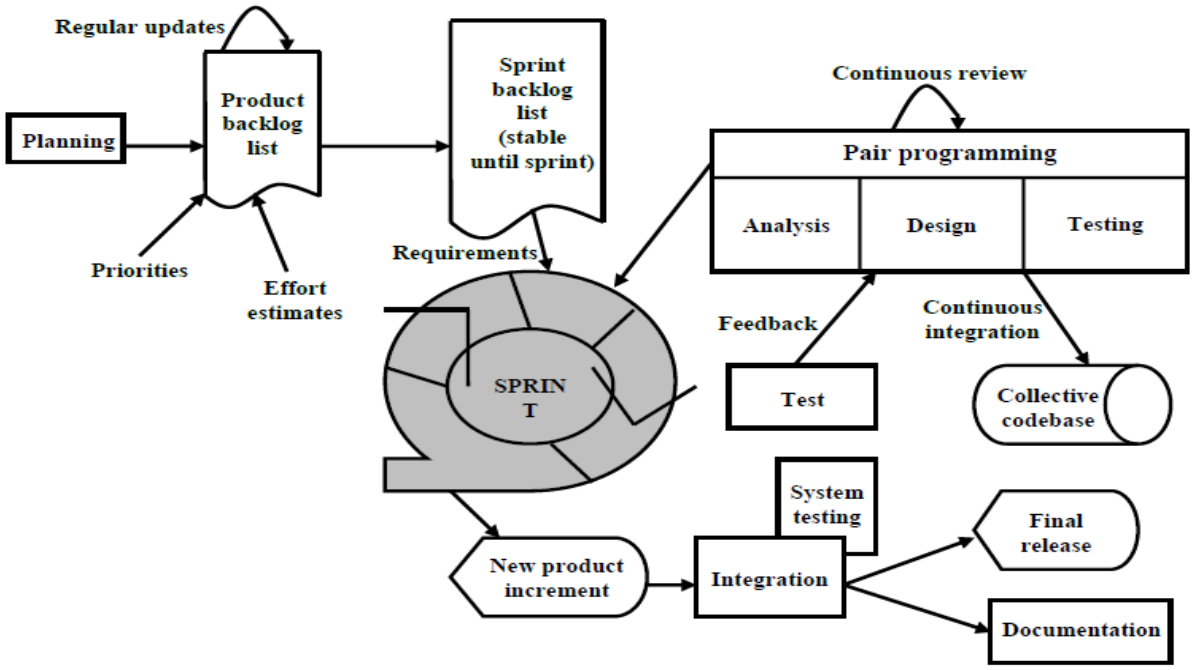

Figure 2. An agile collaborative and innovative framework [2]

TABLE I. MAPPING OF TRADITIONAL RE AND AGILE RE ROLE, ACTIVITIES & ARTEFACTS

Traditional RE	Role		Activities		Artefacts	
	Traditional RE	Agile RE	Traditional RE	Agile RE	Traditional RE	Agile RE
A. Requirement Elicitation	System Analyst, Requirement Engineer, Domain expert.	Stakeholder, Product Owner, Team Members, Visionary.	Discover requirements, System Constraint, Prototyping, Brain Storming, Interviewing	Description of all features, wish list item.	List down requirement	Vision, Product backlog.
B. Requirement Analysis						
1. Requirement Prioritization	Customer, developer, Team leader	Product owner, Scrum Master	Most valuable features, set priority.	Prioritized functions	Prioritized requirements	Prioritized Product backlog List
2. Modelling	Software Engineer, System Analyst, Graphics Designer	Development Team	Bridge between analysis and design	Preliminary Model	Data Flow Model, Semantic Data Model, OOD	Iterative Functional Model
C. Requirement documents	Technical Writer Stakeholder, customer,	Product owner, Development Team	Preparation of Software requirement specification	Preparation of System Architecture	Software Requirement specification document	User stories, product backlog
D. Requirement validation	Development Team, System Analyst	Product owner, Development team	Accepted Requirements as per set standard.	Project status meeting, project demonstration meeting, Retrospective meeting	Validate requirements	Product backlog, Sprint backlog, Sprint
E. Requirement management	Requirement Engineer, Customer, developer	Product owner, Development Team	To capture information, version control management.	Update Sprint Backlog	Baseline Version control document	Product backlog, Sprint backlog

VI. FACTS FOR SHIFTING FROM TRADITIONAL RE TO AGILE RE

- There are traditional approaches like the Waterfall Model. It concentrates on blunt planning while agile methods depend on flexibility and adoptability as per user needs and expectations [17]. Agile methods have emerged as an insurrection against traditional approaches. Traditional approaches are considered as highly technical and unproductive because they are related with complex documentation. [18].
- Due to flexible nature of agile methodologies, level of customer satisfaction increases [7].
- Traditional RE has lack of capability to respond the evolving requirements and insufficient ability to focus on learning. On the other hand, Agile RE can easily grasp the evolving requirements and it welcomes uncertain requirements at any stage of software development life cycle due to its flexible nature. [18].

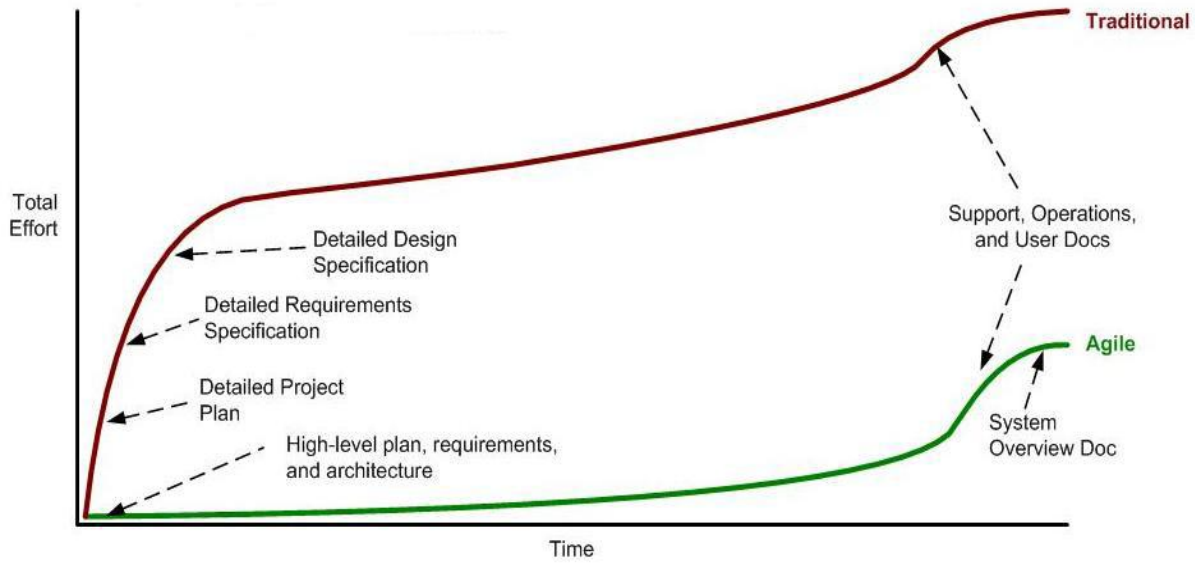

Figure 3. A view of documentation within traditional and agile software developments [19]

In figure 3, it is shown that there is an appropriate level of concise documentation in agile requirements engineering as compared to the complex documentation in Traditional RE.

- In traditional RE developers construct a huge documentation and concentrate on it for the system to be developed. In some cases when changes come, due to shortage of time it is difficult to reorganize the documentation so at the end there is a gap between code and documentation. At that time it seems that this documentation is wastage of time. But Agile RE does not waste time in building huge and complex documentation.
- Agile methods play a key role in solving the problems of constantly evolving requirements and confirm the product delivery in time which will definitely satisfy the client's needs and expectations [5].

VII. CASE STUDY

A case study for public sector hospital has been chosen. This project of Hospital Management Information System (HMIS) has been developed by software industry development team. The experts have applied agile requirement engineering for the development of this project. They have already experience with Traditional Requirement Engineering. We have evaluated the results on the basis of their expert opinions/responses.

HMIS consists of following core modules:

Patient Panel, Admin Panel, Test management,

User management etc.

Opinion of experts has been given in the form of 'Yes' and 'No' and has been tabulated in table II against success factors. Graphical representation of this expert's response is shown in figure 4.

VIII. EVALUATION

The software industry has been planning to develop this project further more assigned this task to project team that consisted of 12 experts that is 2 Data base administrators, 2 Managers, 3 Developers, 3 Technical Writers and 2 QA experts. They were switched to agile software development from traditional software development. We evaluated the results through interviews with the experts that why they moved to agile development. We selected 10 critical success factors from [20], [21]. On the basis of these critical success factors we interviewed these experts and evaluated the results. It was observed that Agile RE was doing better than Traditional RE with the following factors are:

1. Small Duration Project (SDP)
2. Project Team With Expertise (PTWE)
3. Upfront Risk Analysis (URA)
4. Good Customer Relationship (GCR)
5. Face-To-Face Communication (FTFC)
6. Right Amount Of Documentation (RAOD)
7. Flexibility (FLXB)
8. Responsive To Change (RTC)
9. Correct Integration Testing (CIT)
10. Effective Delivery Management Process (EDMP)

TABLE II. COMPARISON FOR RESULTS

Critical Success Factors		SDP	PTWE	URA	GCR	FTFC	RAOD	FLXB	RTC	CIT	EDMP
Traditional RE	Yes	3	5	10	5	4	3	2	4	7	6
	No	9	7	2	7	8	9	10	8	5	6
Agile RE	Yes	8	10	6	10	12	8	12	12	10	9
	No	4	2	6	2	0	4	0	0	2	3
Difference		5	5	4	5	8	6	10	8	3	3

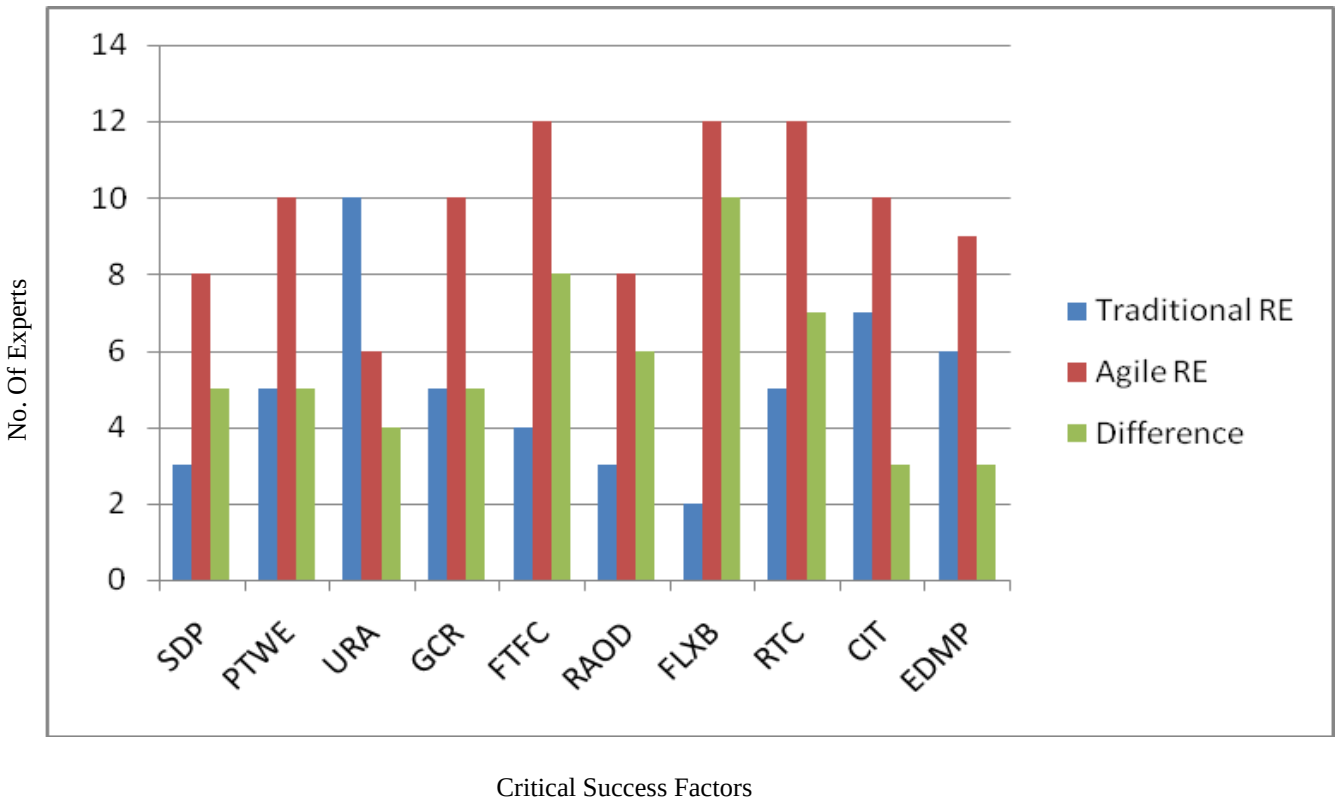

Figure 4. Snapshot of Comparison for Critical Success Factor

Experts have given the opinions on the basis of requirement engineering success factors in the form of "YES" and "NO". On the basis of expert opinions a graph has been prepared. It shows the percentages of RE success factors according to Traditional RE while Agile RE is shown in the Figure 5.

In Traditional RE graph, percentage of "YES" decreases and percentage of "NO" increases. In the graph of Scrum RE the situation is reversed because here the "YES" percentage increases and "NO" decreases.

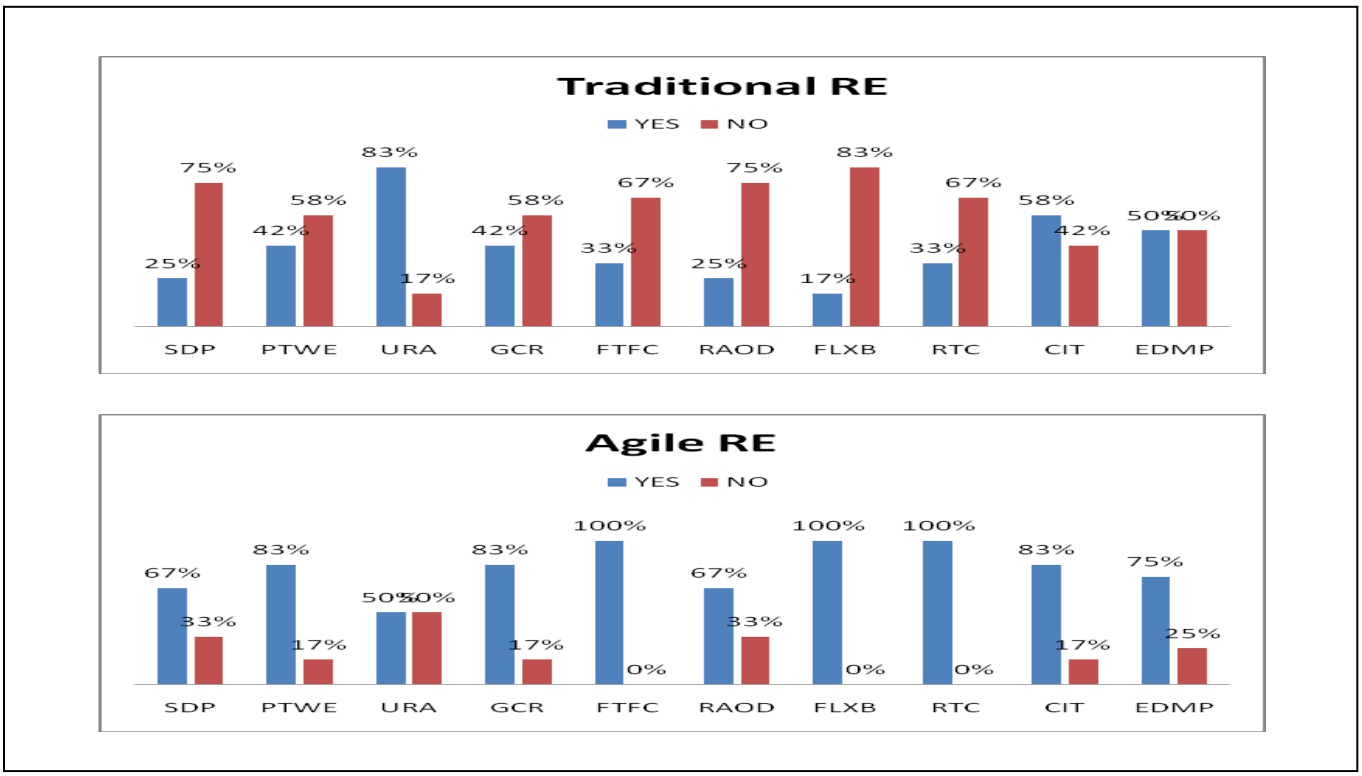

Figure 5. Graphical Representation of Cumulative Results

In Figure 6, shows Graphical Representation of Positive and Negative Responses in Traditional RE and Agile RE in Pie charts.

In the graph of Traditional RE, 59% responses are in NO while 41% in YES but in the graph of Agile RE the situation is different because here 81% responses are in YES and remaining 19% are in NO.

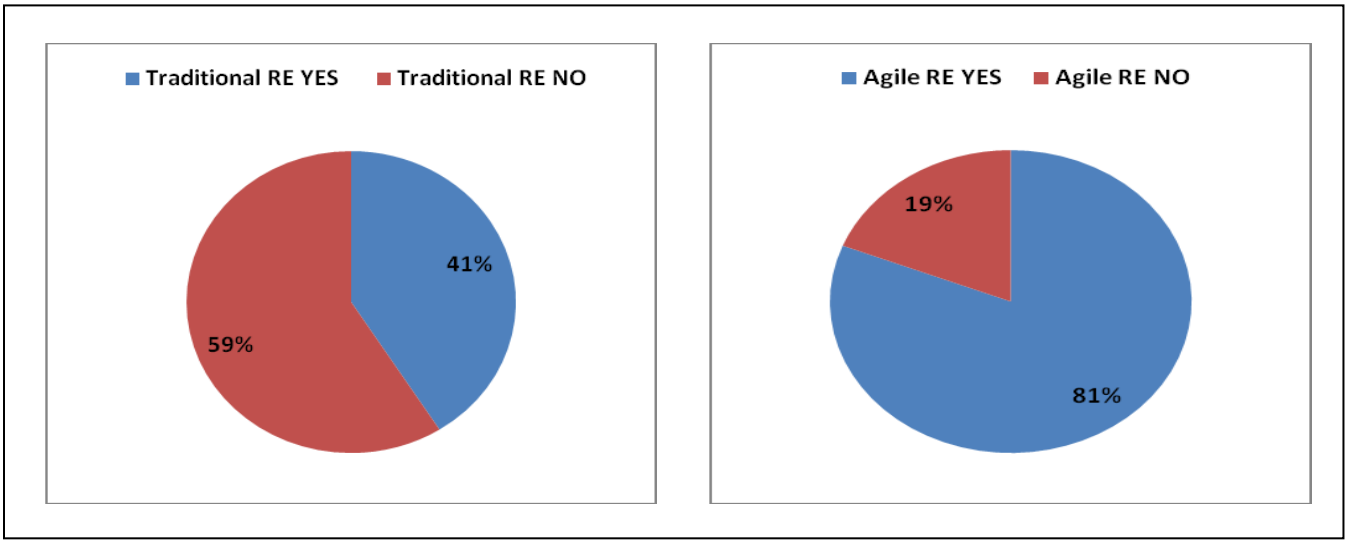

Figure 6. Graphical Representation of Positive and Negative Responses in Traditional RE and Agile RE

XI. CONCLUSIONS AND FUTURE WORK

We have described Traditional RE and Agile RE that how both techniques are different from each other with respect to their roles, artefacts and activities. This comparison revealed that why people have shifted from Traditional RE to agile RE. Our idea was to throw a light on the importance of agile development for effective and flexible requirement engineering process. Result shows that the agile requirement engineering performs well than traditional requirement engineering. This study will also help the practitioners to understand the hurdles face during traditional requirement engineering and how we can add suppleness to it. In future we will apply the Artificial Intelligence techniques like Case Base Reasoning (CBR) and Ontology to make the RE process more effective.

REFERENCES

[1] L. Jun, W. Qiuzhen and G. Lin, "Application of Agile Requirement Engineering in Modest-sized Information Systems Development", 2010 Second WRI World Congress on Software Engineering, 2010.

[2] V. E. Jyothi and K. N. Rao, "Effective Implementation of Agile Practices: Ingenious and Organized Theoretical Framework", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.3, March 2011.

[3] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap", Department of Computing, Imperial College, London, 2000.

[4] H. Saiedian and R. Dale, "Requirements engineering: making the connection between the software developer and customer," Information and Software Technology, 42(6), 2000,:pp.419-428

[5] A. D. Lucia and A. Qusef "Requirements Engineering in Agile Software Development", JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 2, NO. 3, AUGUST 2010.

[6] K.Beck, "Embracing Change with Extreme Programming," IEEE Computer, 32(10),1999,pp. 70–77

[7] F. Paetsch, A. Eberlein, F. Maurer, "Requirements Engineering and Agile Software Development", Twelfth IEEE International workshop, 2003.

[8] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, "Agile software development methods - Review and analysis", VTT Electronics ed.: VTT Publications, 2002.

[9] J. Karbowski, "agile requirement engineering\The Benefits Of Agile Development – VersionOne", Retrieved on 2012-02-10.

[10] K. Beck, A. Cockburn, R. Jeffries, and J. Highsmith, (2001)."Agilemanifesto". http://www.agilemanifesto.org, Retrieved on 2012-02-10.

[11] T. Poppendieck, M. Poppendieck, "*Lean Software Development: An Agile Toolkit for Software Development Managers"*, 2003 Addison-Wesley.

[12] J. Sutherland , "Agile Principles and Values", Retrieved on 2012-01-13

[13] F. Maurer and S. Martel, "Extreme programming: Rapid development for Web-based applications, 2002 IEEE Internet Computing 6(1): 86-90.].

[14] J. E. Tomayko, "Engineering of Unstable Requirements Using Agile Methods", 2002. Agilealliance.org

[15] J. E. Tomayko, "Using Extreme Programming to Develop Software Requirements," *Soft-Ware 2002*. Sprnger-Verlag, 2002, pp. 315-331.

[16] G. Kotonya,I. Sommervile, *"Requirements Engineering PROCESS AND TECNIQUES"*, July 2003

[17] J. Nowicki , B. Furneaux, " Challenges of Transitioning from Traditional to Agile Requirements Engineering: A Case Study", 2011

[18] C. G. Cobb, *"Making Sense of Agile Project Management: Balancing Control and Agility, 2011* New Jersey: John Wiley & Sons

[19] S.W. AMBLER, "AGILE MODELING", 2002. NEW YORK: JOHN WILEY & SONS. RETRIEVED ON 2012-02-10 FROM HTTP://WWW.AGILEMODELING.COM/

[20] T. Chow, D. B. Cao, "A survey study of critical success factors in agile software projects", 2007. The Journal of Systems and Software 81 (2008) 961–971

[21] T.C. Davie, "The ''real'' success factors on projects". International Journal of Project Management 20 (2002) 185–190. 2002.