International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies

http://TuEngr.com

PAPER ID: 11A9A

FACTORS AFFECTING REQUIREMENTS ENGINEERING IN AGILE SOFTWARE DEVELOPMENT: A SYSTEMATIC ANALYSIS

Muhammad Ali Abid [1*], Zia Ud Din [1], Muhammad Ijaz Khan [1], Tariq Naeem [1]

[1] *Institute of Computing and Information Technology, Gomal University, Dera Ismail Khan, PAKISTAN.*

ARTICLEINFO

Article history:
Received 16 January 2020
Received in revised form 05 March 2020
Accepted 10 March 2020
Available online 24 March 2020.

Keywords:
Requirements Engineering (RE); Agile Requirement Engineering Models (AREM); Traditional Models; Agile RE practices; IEEE database; ACM database; Springer database; Elsevier database; Science direct database.

ABSTRACT

Requirements Engineering (RE) has been considered as agile software development initial routes that include system stakeholders in problem analysis iterative process, elicitation requirements, specification, and validation. A successful project is often dominated by the RE process. Numerous techniques about requirement engineering practices are existent to ensure that requests are complete from each dimension. This paper presents an exposition of requirements engineering challenges as well as complexities, beset on large software research division. In this drive, developing requirements is a challenging undertaking that helps in identifying the related issues. In RE, the involved events are defined and facets of an iterative procedure are offered. Removing complexities from AREM is a very tedious and laborious task. We pointed out those complexities who dominate the REM, badly affect the Agile Requirement Engineering Model towards success. If we omit the pointed complexities or have care during the RE model development, then we achieve as we desire accordingly.

Disciplinary: Information Technology, Software Engineering.

1. INTRODUCTION

Many organizations, in the modern era, are working in an agile environment to ensure the provision of high-quality fast software. In this regard, in 2001, seventeen people produced the manifesto of the Agile Software Development (ASD) in the USA (Utah), as representatives from diverse renowned organizations like extreme programming, scrum, the crystal family of the methodologies, Kanban and dynamic development systems method and feature-driven development. ASD focuses upon fast recapitulations, causing slight and recurrent releases in the development process and considered the substantial exit from document-driven heavyweight methodologies for software development. Agile development in no-way taken as a methodology in itself. This “agile development” is considered as “umbrella term” that explains multiple Agile methodologies. Four

*Corresponding author (M.Ali Abid). E-mail: abidawan.gu@gmail.com

core values are provided by agile manifesto those are assorted as follows (Abrahamsson, 2002)

- ✓ Working software through inclusive documentation;
- ✓ Individuals interaction through tools and processes;
- ✓ Retorting to alterations through subsequent planning;
- ✓ Customer cooperation through indenture negotiation.

Requirement engineering is considered to be a cardinal area in software development. A successful project is often dominated by the Requirements Engineering (RE) process. Requirements elicitation, documentation, analysis, management, and validation are vital parts of the traditional software engineering (Sommerville et al., 1998). The requirement stands for and what is to be done, but nothing is mentioned about the implementation. Numerous techniques about RE practices occur to ensure that supplies are consistent, relevant and comprehensively completed (Inayat et al., 2014). The sole aim of RE is to show prior the system development becomes functional. It is intending to stop costly rework. However, upfront requirements specification and gathering exertions are time-consuming and limited scope is left to enlist requirements changing in the cycle of development. The organization of software development deals often with requests that change rapidly and go out of use before the project gets completed. The stakeholder preferences, due to quick alterations in viable threats, market pressures and development technology reduce the pre-specified inappropriate requirements.

Agile methods are very much liked by scholars and practitioners. The regular feedback from the stakeholder requirements come up throughout the development process. This is how the RE process of Agile software differs from that of traditional development. Varied groups of the people hailing from different organizational and individual goals, social positions, hold a stake in a project. The requirements development processes differ vastly in Agile as it depends upon the people who are involved (Pandey et al., 2010). The overall knowledge respecting the requirements development methods is of great importance for the engineers to predict the requirements development process and choose/select the proper method. Some Agile requirement collection techniques are like user stories, prototyping, joint application design, interviews and brainstorming (Grau et al., 2013). There are many good practices of requirement engineering which make rich Agile approaches with benefits and challenges. Communication (face to face) specification (over-written) is advantageous because communication (informal) checks the requisites for the approval processes and time-wasting documentation that are thought to be irrelevant, especially with evolving requirements. But various organizations described that attaining customer representation onsite is problematic in this regard.

When different groups are involved (more than one customer) and each of them concerned with different system aspects then it is challenging to attaining compromise or consensus in diminutive development life-cycle. The iterative RE has two advantages as firstly, it makes the extra satisfactory association with customers and secondly, the requirements are clearer and comprehensible. It has three main defies likewise, as firstly, it is schedule and cost estimated and secondly, the challenge is minimal documentation. While, when breakdown communication happens, the absence of documents may cause a variety of complications. The third challenge is the neglect of requirements (non-functional) like maintainability, scalability, performance, safety, or portability (Sillitti & Succi, 2005). The requirement prioritizations are extreme since consumers are convoluted in the change

process. If continuous reprioritization is not practiced with care, it will lead to unpredictability. Thus, managing changes towards requirements over persistent scheduling minimizes largely the requisite for main changes but redesigning of architecture increases the project cost.

The prototypes are used to attain customer quick response on requirements however in the production model, many organizations highlighted the jeopardies for positioning prototypes. Test-driven Development (TDD) is used to internment wide-ranging requirement and documentation design. A big challenge to TDD's adoption is that adopters are not used to reducing tests in writing before coding. Most of the adopters in the study believe that they do not consistently adopt this repetition for discipline demands. The acceptance tests and use review consultations provide benefits and include opportunities to assess whether the project is according to the target, increasing consumer confidence and trust in teams and recognize difficulties during the process of development. The agile rehearses underline recognition challenging, some organizations find executing like testing arduous and hard because of the access exertion to the clienteles those who mature these assessments. Consequently, numerous organizations use the quality assessment personnel to assist the clientele to mature these prerequisite tests (Kumar et al., 2013).

2. LITERATURE REVIEW

RE is of unique importance in all software development processes and the ASD among the software organizations has been popularly developed. However, the role of RE in Agile development has not been studied in depth. Requirement engineering in Agile consist of five major focuses.

2.1 GENERAL DESCRIPTION OF RE

Few researchers explain Agile approaches to requirements elicitation and management. Regular interactions with customers are focused to add new requirements and discard outdated requirements. Researchers believe that requirements can be created through collection and prioritization activities on the outset of each iteration and Agile systems can effectively manage the requirements in lesser groups instead of large groups (Cao & Ramesh, 2008; Sillitti & Succi, 2005). Certain suggestions are there that these requirements must be collected from various interviews and stakeholders' viewpoints be utilized. The confirmation of initial descriptions of requirements and NFRs included has been proposed (Zhu, 2009). The user stories are the points of the initial requirements gleaning and processes development in Agile systems. The preliminary requirement is starting points and it is probable to enlarge further requirements as others are identified around the products. Likewise, "I will know it when I see it (IKIWISI)" (Bose et al., 2008) has grown in its importance as a required method. An initial form of product develops archetype however; the agile systems are such as that the customers feel inclined towards IKIWISI.

2.2 TRADITIONAL RE vs AGILE RE

Agile methods are considered to be people-oriented instead of process-oriented and also adaptive instead of the projective (Boehm, 2000; Paetsch et al., 2003). Major variance amid traditional and Agile means is client participation in the development procedure. It is concluded that Agile methods in small projects effectively manage requirements however not in outsized projects.

*Corresponding author (M.Ali Abid). E-mail: abidawan.gu@gmail.com Volume 11 No.9 ISSN 2228-9860 eISSN 1906-9642 CODEN: ITJEA8 Paper ID:11A9A http://TUENGR.COM/V11A/11A9A.pdf DOI: 10.14456/ITJEMAST.2020.163

Large projects are managed by traditional methods soundly (Cao & Ramesh, 2008). The latest research-based on case study outcomes revealed that on large Agile adopting context, scattered projects develop worth, agree for the change in the requirements and enhances customer satisfaction as compared to the traditional development methodologies developed for the same purpose (Racheva et al., 2010).

2.3 ISSUES AND BENEFITS OF AGILE RE

An analysis of 16 organizations of software development underlined the advantages and disadvantages 7 Agile requirements engineering practices: communication (face-to-face) over specifications (written), requirements engineering (iterative), prioritization (requirements) drives risky, the requirements managing changes over consistent prototyping, planning, test development determined, reviewing meetings and acceptance tests (Kumar et al., 2013). Agile requirement engineering activities are talked about and difficulties of apiece activities are portrayed, remedy and solution of these difficulties are sorted out. Efficient applications of RE practice in Agile requirements are found out. Likewise, the traceability amid Agile artifacts software is not lucidly described. The balance between agility and stability within the development team and customer remains another challenge (Fowler, 2001; Lucia & Qusef, 2010). A review of challenges and their solutions from 2007-2012 has been presented and it is concluded that the communication, coordination, collaboration, and differences (cultural) are main tasks of Agile distributed development (Svensson et al., 2008).

2.4 RE TECHNIQUES IN AGILE

Various approaches for requirement elicitation in Agile like customers' participation in shared development application sessions, modeling, prioritization, brainstorming, interviews have been proposed (Boehm, 2000). A summary of RE techniques has been provided. UnsTable requirements are dealt in a better way by Agile. Also, two methods for requirement elicitation user stories and prototyping are discussed (Grau et al., 2013; Bjarnason et al., 2011). The Communications serves in Agile an important RE practice between developer and customer (Kumar, 2013). A systematic review of literature during past decades, (2002 & 2013) has recognized seventeen Agile RE practices likewise customer involvement, communication (face-to-face), iterative supplies, user stories, change management, requirements prioritization, prototyping, cross-functional teams-testing afore coding, modeling desires, requirements management, the acceptance tests, and review meetings, shared conceptualizations, code refactoring, continuous and retrospectives planning and requirements analysis pairing (Rizvi, 2013).

2.5 TOOLS AND MODELS FOR RE IN AGILE

An Agile model, with four abstraction levels, has been presented to integrate the traditional engineering requirements in an agile environment. The researchers investigated probability by taking the case company to hold engineering requirements in a scrum situation (Papadopoulos, 2015). In this regard, no real process exists for the agile method suggested to adapt the requirement engineering. To surmount this impediment, it is suggested to adopt the requirements model group to an agile setting so that to construct an agile requirement engineering procedure. The hybrid model namely agile requirements model generation is the lenient organized course that paves way for Agile intents

approach. Thus, this model amalgamates the superlative structures of Agile software development and the requirements generation model (Tomayko, 2002). The important mechanism of Agile is famous as reprioritization which helps and assists in accommodating unstable requirements. The conceptual model based on multiple case studies has also been derived. This model helps us to comprehend the iteration prioritization process this model answers questions as to what are cardinal concepts when requirements prioritization is brought under consideration (Soundararajan et al., 2009).

3. RESEARCH DESIGN

In this connection, the literature systematic review is piloted systematically by succeeding the guidelines set to "collect & analyze" all accessible confirmation around the particular demand in the repeatable and unbiased manner. Therefore, the aim is to meet the most relevant literature to requirement engineering, by viewing out the factors which are the causes of the failure of the Requirement Engineering Models. The following steps will be performed during this SLR.

3.1 RESEARCH QUESTION

This study is mainly focused on the notion that what are the factors affecting the requirement engineering model in ASD?

3.2 SEARCH STRATEGY AND QUERY STRING

For this review, the search standards used involves two measures likewise C1 & C2, which are well-defined with the following descriptions like C1 is the sequence fabricated for keywords towards methods for software agile development like extreme programming, agile, agility, scrum, test-driven development, development feature-driven, Kanban and lean. While, C2 is the string keywords made-up linked with requirements engineering like requirement, user story, requirements engineering as well as features.

3.3 SEARCH TERMS

Table 1 shows the major keywords concerning better validate the concepts through existing research studies. Using the below keywords, the following search query is made. All the keywords and their alternate terms are logically ORed together and then logically ANDed to form the query search. The AND Software (agility OR agile OR ''XP'' OR scrum OR FDD OR extreme programming OR feature-driven OR TDD OR feature-driven development OR test-driven OR development test-driven OR Kanban OR lean) AND (requirements OR requirements engineering OR feature OR user story OR prioritization).

3.4 PRIMARY AND SECONDARY SEARCH STRATEGIES

The above-elaborated query has been applied to well-known databases. Date filter is applied since 2001 as an agile manifesto was introduced in 2001. Search Results from each database were kept in a separate file. At the results of all databases were combined and removed duplicate results. Eventually, we got a total of 211 papers. The detail of the search is listed in Table 2.

*Corresponding author (M.Ali Abid). E-mail: abidawan.gu@gmail.com

Volume 11 No.9 ISSN 2228-9860 eISSN 1906-9642 CODEN: ITJEA8 Paper ID:11A9A http://TUENGR.COM/V11A/11A9A.pdf DOI: 10.14456/ITJEMAST.2020.163

Table 1: The Concepts through Existing Research

Keywords	References
Agile	Sillitti & Succi, 2005; Bose et al., 2008; Soundararajan & Arthur, 2009; Fowler, 2001)
Extreme Programming	Pandey at al., 2010; Bose, 2008; Sillitti & Succi, 2005; Paetsch, 2003; Fowler, 2001)
Scrum	Cao & Ramesh, 2008; Paetsch, 2003; Racheva, 2010; Inayat, 2014; Bose, 2008;)
Feature Driven Development	Grau et al., 2013; Zhu, 2009; Bose et al., 2008; Sommerville & Kotonya, 1998; Pressman, 2005).
Kanban	Cao & Ramesh, 2008; Abrahamsson, 2002; Pandey, 2010; Kumar, 2013; Bose, 2008; Tomayko, 2002).
Dynamic Systems Development Method	Abrahamsson, 2002; Pandey, 2010; Boehm, 2000; Paetsch, 2003; Bose, 2008; Fowler, 2001; Soundararajan & Arthur, 2009).
Lean	Paetsch, 2003; Bjarnason, 2011; Inayat, 2014; Sommerville & Kotonya, 1998; Lucia & Qusef, 2010; Zainab et al., 2017).
Test Driven Development	Kumar et al., 2013; Sillitti & Succi, 2005; Cao & Ramesh, 2008; Bose et al., 2008; Sommerville & Kotonya, 1998; Pressman, 2005).
Requirement	Grau, 2013; Kumar, 2013; Zhu, 2009; Bose et al., 2008; Paetsch et al., 2003; Lucia & Qusef, 2010; Tomayko, 2002; Papadopoulos, 2015; Svensson, 2008; Sajjad, 2010).
Requirement Engineering	Grau, 2013; Sillitti & Succi, 2005; Zhu, 2009; Sommerville & Kotonya, 1998; Pressman, 2005; Sajjad, 2010; Bhatti, 2016; Sehrish, 2010).
Requirement Modeling	Grau, 2013; Zhu, 2009; Bose, 2008; Kumar, 2013; Paetsch, 2003; Svensson, 2008; Lucia & Qusef, 2010; Sajjad, 2010).
User Story	Abrahamsson, 2002; Pandey et al., 2010; Bose et al., 2008; Boehm, 2000; Fowler, 2001; Soundararajan & Arthur, 2009; Lucia & Qusef, 2010; Sajjad, 2010; Swarnalatha, 2014; Davey & Chris, 2008; Lizbeth, 2017).

Table 2: Search Results.

Databases	Before	After
IEEE	120	57
ACM	112	75
Springer	90	42
Elsevier	105	81
Science direct	80	41
Total	507	211

3.5 SELECTION CRITERIA

Keeping in mind the purpose of current SLR the following inclusion/exclusion criteria were defined.

3.5.1 INCLUSION CRITERIA

The inclusion criteria comprised of the papers published; papers English has written; papers presenting approaches and papers under the peer-review process to assimilate the users into processes of agile development; Agile RE papers related; papers related to documentation of agile requirements and definite book sections/chapters.

3.5.2 EXCLUSION CRITERIA

Similarly, the studies those who did not emphasis clearly on agile approaches, however only mention to methods for agile development software as the side-by-side topic (studies which cites agile as adjective) while studies which did not deliberate requirements engineering in agile approaches, likewise the studies which did not come across the criteria for inclusion and viewpoint, opinion, discussions, keynote, comments, editorials, prefaces, tutorials, and papers anecdote and slide presentations formats starved of any related papers.

3.6 STUDY SELECTION PROCESS

This section is divided into the following two stages to cover the criteria required for the systematic study and to achieve the desired information.

3.6.1 TITLE & ABSTRACT LEVEL SCREENING

The exclusive and inclusive standards are pragmatic to abstract and title of 211 primaries studies. At that point, those studies were excluded which are not relevant to the current research, ended up with 30 papers.

3.6.2 FULL-TEXT LEVEL SCREENING

In this stage, all 30 papers were studied in detail. Exclusion/inclusion standards were practical to the contents of all 30 papers. ten papers were excluded in this stage.

3.7 QUALITY ASSESSMENT

The rest of the 20 papers were evaluated against the quality criteria given in (Soundararajan & Arthur, 2009) as shown in Table 3.

Table 3: Checklist of Quality Assessment (adopted by Soundararajan & Arthur (2009)

No.	Question	Score
1	Was the study intended to attain certain objectives?	Y|N|A
2	Are the study specified the research aims?	Y|N|A
3	Are techniques of estimation used selection and described justified?	Y|N|A
4	Are the considered variables by the study have measured suiTable?	Y|N|A
5	Are the data gathering adequately described the methods?	Y|N|A
6	Is the data analysis determination is clear?	Y|N|A
7	Is the data composed described adequately?	Y|N|A
8	Are the negative consequences have been accessible?	Y|N|A
9	Are to analyze data, statistical procedures used justified &described?	Y|N|A
10	Are all the answers to the research questions effectively presented?	Y|N|A
11	Do scholars discourse any issues with reliability/validity of results?	Y|N|A
12	Are the results based on compound projects?	Y|N|A
13	How strong are the connections among data, clarification & decision?	Y|N|A

Using the 3-points scale, each question was answered by Yes (Y=1), No (N=0), Average (A=0.5). Each study could get 0-13 points. Using quartile first (13/4= 3.25) as for including endpoint the study. When the study got equal or more than 3.25 it would be selected otherwise removed.

4. RESULTS OF STUDY

This section shows the overall results of the current SLR. Tables 4 and 5 show the numbers of papers which are evaluated in various phases, detail of accepted and rejected papers are given.

Table 4 Paper Evaluated at |Different Phases

Databases	Search
Search Result	211
After Title & Abstract Selection	30
Papers inaccessible	00
Excepted on inclusive /exclusive measures	10
Study duplicated	00
Low Score Quality Papers	02
Final Paper (b-c-d-f)	18

*Corresponding author (M.Ali Abid). E-mail: abidawan.gu@gmail.com Volume 11 No.9 ISSN 2228-9860 eISSN 1906-9642 CODEN: ITJEA8 Paper ID:11A9A http://TUENGR.COM/V11A/11A9A.pdf DOI: 10.14456/ITJEMAST.2020.163

Table 5: Year wise summary of the research paper in study selection and QA

Year	Before exclusion	After Exclusion
2001	2	1
2002	0	0
2003	0	0
2004	4	4
2005	8	7
2006	12	11
2007	13	12
2008	11	11
2009	12	10
2010	14	12
2011	12	10
2012	15	13
2013	10	9
2014	14	12
2015	18	18
2016	11	10
2017	14	14
2018	19	17
2019	22	22
	211	193

Total Papers = 211, Remaining = 193, Excluded papers = 18

Table 6: RQ: Factors affecting the Requirement Engineering

Factors	Frequencies
Underspecified requirements	4
Revisions of requirements	8
The weak relationship between customer and project lead	5
Lack of executive support	14
Lack of planning	11
freezing of task distribution and redistribution	9
Ambiguous requirements Leeds to ill-spent time and rework.	15
Moving objectives (business processes, changing goals & requirements)	11
Technically unfeasible requirements	9

Table 6 shows the complex issues that mark the process of requirement engineering in ASD.

5. CONCLUSION

The present study aimed at the systematic analysis of the factors affecting the RE in agile development software. The common complexities in software requirement models have been identified. If these complexities are properly addressed, the process of software requirements can be significantly improved. During the groundwork of Agile Requirement Engineering Model if Intention of Model, Design constraints, Selection of appropriate modulus and their dimension concerning the targeted goals be attentive then major complications never are happened and shaped a well versed and versatile Agile Requirement Engineering model.

6. DATA AND MATERIAL AVAILABILITY

Data can be made available by contacting the corresponding author.

7. REFERENCES

Abrahamsson, P. (2002). Agile Software Development Methods: Review and Analysis. VTT.

Bjarnason, E., Wnuk, K., & Regnell, B. (2011). A case study on benefits and side-effects of Agile practices in large-scale requirements engineering. In Proceedings of the 1st Workshop on Agile Requirements Engineering. 3.

Boehm, B. (2000). The requirements that handle IKIWISI, COTS and rapid change. Computer, 33(7), 99-102.

Bose, S., Kurhekar, M., & Ghoshal, J. (2008). Agile Methodology in the Requirements Engineering. SET Labs Briefings Online.

Cao, L., & Ramesh, B. (2008). Agile requirements engineering practices: An empirical study. Software, IEEE, 25(1), 60-67.

Davey, B., & Bendigo, C. (2008). Requirement Elicitation-what´s Missing? Issues in information sciences and information technology, 5, 543-551.

Fowler, M. (2001). The new methodology. Wuhan University Journal of Natural Sciences, 6(1-2), 12-24.

Grau, R., Lauenroth, K., Bereza, B., Veenendaal, E., & Zee, S. (2013). The Requirements Engineering and Agile Development-collaborative, just enough, just in time, sustainable.

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2014). A systematic literature review on Agile requirements engineering practices & challenges. Computers in Human Behavior.

Kumar, M., Shukla, M., & Agarwal, S. (2013). A Hybrid Approach of Requirement Engineering in Agile Software Development., International Conference on Machine Intelligence and Research Advancement (ICMIRA). IEEE. 515-519

Lizbeth, Z., Alfonso, A., Carolina, T., & Sanjay, M. A. (2017). Requirements Engineering Techniques Review in Agile Software Development Methods.

Lucia, A. D., & Qusef, A. (2010). The requirements engineering in Agile software development. Journal of Emerging Technologies in Web Intelligence, 2(3), 212-220.

Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and Agile software development. In 2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. IEEE Computer Society. 308.

Pandey, D., Suman, U., & Ramani, A. K. (2010). An effective requirement engineering process model for software development and requirements management. International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), 287-291.

Papadopoulos, G. (2015). Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects. Procedia-Social and Behavioral Sciences, 175, 455-463.

Pressman, R. S. (2005). *The* Software engineering: a practitioner's approach. Palgrave Macmillan.

Racheva, Z., Daneva, M., & Herrmann, A. (2010). A conceptual model of client-driven Agile requirements prioritization: Results of a case study.

Rizvi, B., (2013). A systematic review of the distributed Agile software engineering. In Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. 39.

Sajjad, U., & Hanif, M. (2010). “Issues and challenges of requirement elicitation in large web projects”, School of computing, Blekinge institute of technology ronne by Sweden.

*Corresponding author (M.Ali Abid). E-mail: abidawan.gu@gmail.com Volume 11 No.9 ISSN 2228-9860 eISSN 1906-9642 CODEN: ITJEA8 Paper ID:11A9A http://TUENGR.COM/V11A/11A9A.pdf DOI: 10.14456/ITJEMAST.2020.163

Sehrish, A., Shahid, N., Shumaila, A. (2016). Analysis of Requirement Engineering Techniques in Agile Development Method. International Journal of Computer Science and Information Security. 14(11), 889-894

Sillitti, A., & Succi, G. (2005). Requirements engineering for Agile methods. In Engineering and Managing Software Requirements. Springer. 309-326

Sommerville, I., & Kotonya, G. (1998). Requirements engineering: processes and techniques. John Wiley & Sons.

Soundararajan, S., & Arthur, J. D. (2009). A soft-structured Agile framework for larger scale systems development. ECBS 2009. 16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems. 187-195.

Svensson, R. B., Birgisson, S. Ö., Hedin, C., & Regnell, B. (2008). The Agile Requirements Abstraction Model–Requirements Engineering in a Scrum Environment. Department of Computer Science, Lund University.

Tomayko, J. E. (2002). The Engineering of unsTable requirements using Agile methods. International Workshop on the Time-Constrained Requirements Engineering (TCRE'02). Essen.

Zainab, S., Rabiya, A., Shahid, N., & Asim, A. (2017). Analytical Review on Test Cases Prioritization Techniques: An Empirical Study. IJACSA, 8(2), 2017.

Zhu, Y. (2009). The Requirements Engineering in an Agile Environment.

Muhammad Ali Abid is a PhD Scholar and an Assistant Director (IT), Institute of Computing & Information Technology, Gomal University, Dera Ismail Khan, Pakistan.

Dr. Zia ud Din is an Associate Professor and Director Institute of Computing & Information Technology, Gomal University, Dera Ismail Khan, Pakistan. His research is Agile Software Development, Wireless Technology, and topics in Software Engineering. He can be reached at ijaz171 @ gmail.com

Muhammad Ijaz Khan is a Lecturer and Director IT, Institute of Computing & Information Technology, Gomal University, Dera Ismail Khan, Khyber Pakhtunkhwa, Pakistan. He can be reached at Ziasahib @ gmail.com

Tariq Naeem is a Lecturer at the Institute of Computing & Information Technology, Gomal University, Dera Ismail Khan, Khyber Pakhtunkhwa, Pakistan. He can be reached at tariqnaeem @ gu.edu.pk