

# Investigate Impacts, Challenges And Solutions Of Requirement Engineering In Agile Methodologies: A Review

Hina Noor, Dr. Babar Hayat, Izaz Shah, Hafiz Wajid Ali, M Asaad Subih

**Abstract:** Requirement Engineering is one of essential phase in software development life cycle. In the last few years many people know much about the success rate of Requirement Engineering in a project's success. The agile approach will produce a high standard product in less budget and time. With the passage of time, consumers will be more appealing as their technology standards increase rapidly every day. Agile software development (ASD) is an iterative and gradual approach to development that relies on the participation of customers in the development process. So, we can conclude that organizations are still struggling with agile evolution and agile values, especially when it comes to user involvement. Although much research has been carried out on the role of requirement engineering in agile methodologies and practices, more studies are still needed. The aim of this research is to analyze and compare researchers ' work, which will help those who are interested in finding interesting research fields in this area. Requirement engineering impacts the agile approach tremendously.

**Index Terms:** ASD (Agile Software Development), Agile Methodologies, Challenges of Requirement Engineering, RE (Requirement Engineering), Solutions

## 1 INTRODUCTION

Agile software development includes plentiful software development systems through which details develop through the aggregate effort of self-arranging and cross-useful groups and their clients or clients. Agile software framework development has become a great deal basic over the recent years. The central matter of new business development is to make fast and steady development procedure of progress that are most appropriate to acknowledge changes and progress. [1]. Agile Methodology has been satisfied as the programming strategy of choice for the quick, insecure universe of Internet and web programming development. The work starts with the elicitation and documentation of a total arrangement of prerequisites, trailed by an engineering and significant level structure, development, and assessment while execution of customary techniques [2]. As necessity building, a conventional software item development approach with the goal of characterizing, assessing, and recording and approving the framework prerequisites to be created is considered. Prerequisite Engineering likewise can be plot in light of the fact that the technique for determinative client desires for a substitution or changed item. These choices, referenced to as prerequisites. Prerequisite Engineering is a major field of Software Engineering. We ought to have a dialog on this field before we push ahead. Prerequisite Engineering (RE) is the procedure through which all framework necessities are resolved.

RE expects exercises to characterize partner needs, think about necessities setting, and configuration, arrange, check, record, and handle those prerequisites. Potential increment in PC frameworks size, improved code reconciliation and condition, more noteworthy code autonomy to change in accordance with its setting. The factors and patterns that make the RE part harder are the expanding financial procedure of code development [3]. RE even have huge amounts of significance in software Project Management (SPM). The Requirement Engineering (RE) is the most significant period of the Software Project the executives (SPM). This stage is utilized to disentangle the fragmented needs and necessities of the end clients of software into an exact type of determinations. In this way, the significance of Requirement Engineering is tremendous to create viable software and in diminishing software mistakes at the beginning time of the development of software. [2]. Agile requirement engineering (ARE) has four core activities, i.e. high-level collaboration between the activity group and also the customer, prioritizing the requirements based on their business value for the customer, and eliciting also the non-functional requirements [2]. Agile methodology consists of multiple techniques and functions. Scrum, Extreme Programming, Adaptive Software Development, Dynamic Systems Development Method, Feature Driven Development and crystal development methods are frequently used, shown in Figure 1. Let's discuss some process in agile development.

Agile Development Model

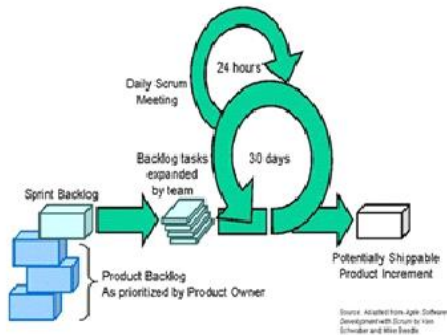


Figure 1 Agile Development Model

- Hina Noor is currently pursuing master's degree program in Software Engineering in University of Lahore (Gujrat Campus), Pakistan, E-mail: hinanoorm006@gmail.com
- Dr. Babar Hayat is a research scholar in University of Lahore(Gujrat Campus), Pakistan, Email: baber.hayat@cs.uol.edu.pk
- Izaz Shah, University of Lahore(Gujrat Campus), Pakistan, Email: izaz.02007@gmail.com
- Hafiz Wajid Ali is currently pursuing master's degree program in Information Technology in University of Lahore (Gujrat Campus), Pakistan E-mail: h.wajidali053@gmail.com
- M Asaad Subih, University of Lahore(Gujrat Campus),Pakistan, Email: asad.subih32@gmail.com

**1.1 SCRUM**

Inside of the Scrum framework individually can address complex versatile issues, while profitably and creatively carrying results of the most exalted credible worth. Scrum itself is a straightforward framework for successful group coordinated effort on complex items. Scrum can be utilized to deal with the creation framework by applying thoughts from the procedure the board framework on expertise, intensity and versatility [3].

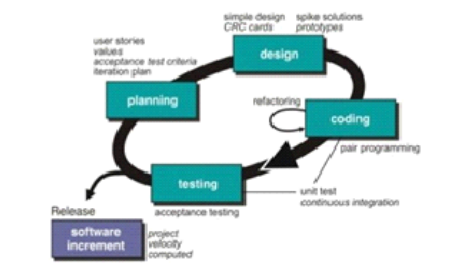


**Figure 2 (1.1 Scrum) iterative process** Products are developed in –one-to four-week sprints that generates working software increments.

**1.2 EXTREME PROGRAMMING (XP)**

Extreme Programming is a system for encoding advancement intended to improve programming quality and its capacity to adjust properly to the changing needs of the client or customer. XP's point is to associate past and tried procedures so as to make them an amazing piece of one another. The legitimization for this methodology is that product improvement is a unique procedure where determinations can't be totally decided from the beginning, yet can be consistently refreshed as tasks progress on.

**Extreme Programming (XP)**



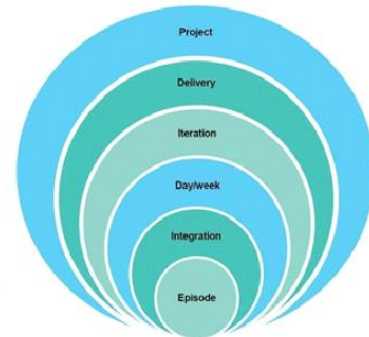
**Figure 3 Extreme Programming (XP)**

Software development needs a procedure that is skilled to adjust to changing prerequisites anytime during the task life [3].

**1.3 CRYSTAL DEVELOPMENT**

In Agile Methodology crystal is a related methodology concentrated on individuals and their commitment, when chipping away at an errand as opposed to on assets and

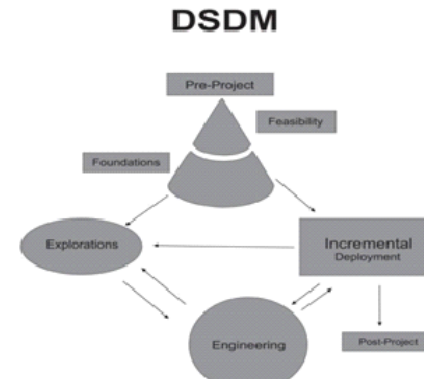
procedures. Crystal procedure has two principle speculations that are gatherings will form their procedures as their work and turn into a streamlined group [3].



**Figure 3 Crystal Development**

**1.4 Dynamic System Development Method (DSDM)**

In agile methodology DSDM is a technique which centers on the full venture lifecycle. It is another method for working in iterative procedure. The DSDM structure incorporates three significant stages, specifically the pre-venture, venture life-cycle and post-venture stages. The undertaking period of DSDM is the most detailed of the three stages [3].



**Figure 4 Dynamic System Development Method**

**1.5 Feature Driven Development (FDD)**

In agile methodology Feature-Driven Development (FDD) is a customer driven, structure driven, and even condemned programming process. The explanation "customer" in FDD is utilized to address what Agile Modeling intimates as experience assistants or extraordinary Programming (XP) calls clients.

**Feature-Driven Development (FDD)**



**Figure 5 Feature Driven Development (FDD)**

### 1.6 Adaptive Software Development

Adaptive software development in agile methodology could be a style rule for software frameworks development appeared in Figure 5. The standard is centered on the fast development and development of software frameworks. The adaptive strategy for development developed from the quick technique for application development. Adaptive Software Development replaces the ordinary procedure of cascades with a redundant arrangement of cycles of investigation, cooperation, and preparing.

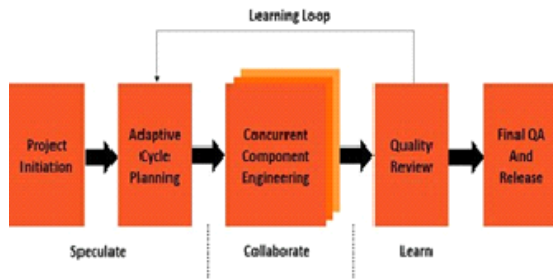


Figure 6 Adaptive Software Development

## 2 LITRATURE REVIEW PROCESS

Sehrish Alam and S Asim Ali Shah planned and proposed a survey investigate. The examination performed in agile methodologies in the territory of detail plan. They directed an audit dependent on necessities, agile practices, distributed year and methodology. The objective is to discover examine pattern around there by characterizing watchwords for scanning for research and diary paper criteria here and furthermore choosing paper criteria for consideration and rejection. They delineate some of difficulties of prerequisite building in agile development and furthermore condense difficulties in every RE practice. Their work is a precise audit of agile methodologies, particular structure downsides and difficulties. They led an orderly audit of the writing to perceive issues [3]. Abdallah Qusef distinguishes issues of testing frameworks in their work. They infer that distinction among conventional and agile development isn't whether to do RE yet the significant contrast is that when to do RE. Agile necessity building applies the significant qualities referenced in the agile system to the procedures of RE [2]. Abdallah Qusef recommended rules for RE for example toward the start, all significant perfect partners must be recognized, the most basic prerequisites are communicated by the partners just as they can toward the start and necessities ought to be actualized in a gradual and iterative way. Their work studies the significant exercises of agile RE including possibility study, elicitation, investigation, documentation, approval, and the executives [2][3]. Frauke Paetsch, Dr. Armin Eberlein and Dr. Straight to the point Maurer look at conventional necessities designing methodologies and agile software development. Their work breaks down similitudes and contrasts between the two methodologies and characterizes potential techniques how agile software development can get advantage from necessities building process [1]. Manoj Kumar and Manish Shukla convey cross breed approach for RE in the agile development by utilizing JAD procedure. They likewise present prioritization of necessity by utilizing perspective. Their works look at these methodologies and examine the advantages of the model [4]. Yasir Hafeez Motla and Bushra Hamid's done

an Agile Requirement Engineering and Traditional Requirement Technology Comparative Research. This investigation demonstrated why individuals have moved from customary RE to agile RE and why agile development is basic for effective, prudent and flexible necessity designing technique. This examination says agile structure innovation is doing great as opposed to traditional prerequisite building. This work will make it simpler for software architects to comprehend the difficulties confronting agile design building and how we can add usefulness to this [5]. EM Schön, J Thomaschewski and MJ Escalona directed an orderly writing audit with a top notch valuation of the included investigations. They perceived 27 important papers and investigate them in detail, at that point start understandings to the accompanying attributes of agile prerequisite designing: partner and client inclusion, information gathering, client viewpoint, incorporated methodologies, shared getting, relics, documentation and Non-Functional Requirements (NFR). The investigation infers that making a mutual comprehension of the client's viewpoint isn't notable in ASD, and this is basically an issue that we face. [6]. S.Thangasamy and N.Ganesh distinguished issues of necessity elicitation in ASD. The principle focal point of study is to pass on mindfulness concerning the difficulties we looked in a few software businesses and it'd permit totally various partners at interims agile principally based setting to rise and see these difficulties in inspiring their prerequisites [7]. T Shah and VS Patel give a survey on prerequisite building difficulties in various software development systems We characterize some need designing difficulties in the plan of use advances, for example Security acknowledgment at RE level, RE framework mix, Accuracy and Performance Measurement of Requirement, Communication Gap, Business Agility and Addressing NFR (Non Functional Requirement). They likewise propose appropriation for appropriateness of issues and difficulties and how to conquer them [8].

### A. Research Questions

1. WHAT ARE THE IMPACT AND CHALLENGES OF RE IN AGILE SOFTWARE DEVELOPMENT?
2. WHAT ARE THE SOLUTIONS TO THE CHALLENGES THAT WE FACED IN AGILE SOFTWARE DEVELOPMENT?

### 2.1 RE practices in Agile

Heikkilä et al. [37] directed a mapping study on necessities building in agile programming development in 2015. Thus, it gives a decent diagram of the theme. They express that "the meaning of agile RE is dubious." This is reflected in the essential examinations that frequently don't determine the solid procedure model utilized. Most the papers examined in their mapping study contained some sort of observational assessment. We allude to their paper for subtleties. Moreover, there is an ongoing methodical writing audit by Inayat et al. [22]. They outline the consequences of 21 essential examinations identifying with agile necessities building. There is just one paper named study explore which thus leads interviews. They arranged that partners should be worried for different motivation behind perspectives for necessities assembling and meetings should be led. We can get constant input from clients by agile interest however breaking points of agile practices don't appear to be very much illustrated. Cao and Ramesh [35] led a subjective investigation of 16 programming development associations

on their agile RE rehearses. They distinguished and evaluated itemized RE rehearses. They found, for instance, that up close and personal correspondence, prototyping and surveys and tests are regular agile RE rehearses. Somewhat tantamount is just the overview by Bustard et al. [24]. They research the development of agile development standards and practices anyway conjointly bit the subject of requirements. They found that their members see a technique benefit in agile necessities social occasion and the board. Paetsch, Eberlein and Maurer (2003) [1] take a gander at the qualifications and correlations between Agile methodologies and conventional RE draws near. Prerequisites building, on the contrary hand, could be a conventional programming designing technique with the objective to spot, break down, report and approve requirements for the framework to be created. Frequently, necessities building and agile methodologies zone unit seen being incongruent RE is typically intensely trusting on documentation for information sharing though agile ways zone unit that represent considerable authority in up close and personal joint effort among clients and designers to prevail in comparable objectives. The point of this original copy is to search out if a few requirements designing strategies are regularly utilized inside agile development and if this may prompt improvements to agile methodologies. They imagined that RE procedure stages elicitation, investigation, and approval are available in every agile procedure. The systems utilized shift in the various methodologies and the stages are not as unmistakably isolated as in the RE procedure - they converge somehow or another (the Planning Game in XP is an elicitation and investigation approach) [37]. The procedures utilized in the agile development forms are once in a while depicted change enigmatically and the real usage is left to the designers. As RE the board depends on archives, isn't very much spoken to in agile programming development approaches. Overall, in key zones (like partner association) agile techniques and RE are seeking after comparative objectives. The significant distinction is the accentuation on the measure of documentation required in a viable undertaking. This incompletely steams from contrasts in center suppositions on the dependability of necessities and exact investigations on the cost viability of the two methodologies are required.

## 2.2 Requirement Prioritization in agile

In Requirement building imperatives of the framework, functionalities of the framework, space, and dangers are characterized [36] [39]. Prioritization of Requirements is a significant stage during the time spent necessity building. It is required considering the way that few out of every odd one of the prerequisites have match the customer's fulfillment. The significant prerequisites may fluctuate for every partner for instance for Project administrator the necessities which are anything but difficult to actualize should execute first and the monetary director is strong for necessities that cost less. In this way, the essential goal of prerequisite prioritization is to pass on functionalities that accomplish the customer's endorsement, appropriate for the task's constraint, and transfer of necessities that are not commendable enough to be executed. Prerequisites prioritization as the Process of sorting out necessities showed by specific criteria, for instance, its essentialness, spending plan, time, hazard, and unconventionality [25]. In this way, Prioritization of necessities

can be performed in numerous Software development forms and is significant because of the restrictions of undertakings and limitations, for example, complete spending plan, time, and partners [34]. This procedure helps in finding irregularity, misevaluating and equivocalness in prerequisites by analyzing the evoked necessities. The normal time and multifaceted nature to organize prerequisites depend upon the venture size, the all-out no. of prerequisites, procedures utilized, and some different components [33]. To initiate an effective prioritization process, the following steps has to be followed:

1. Finding the stakeholders.
2. Select the requirements to prioritize.
3. Define a criterion for prioritization.
4. Choosing a technique for Requirement Prioritization
- 5.

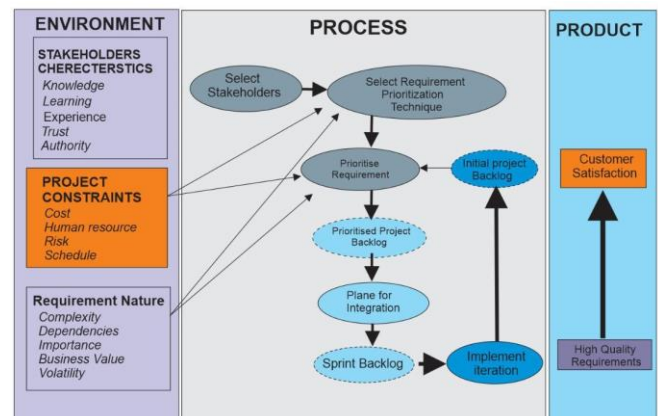


Figure 6 Requirement Prioritization Framework

## 2.3 Environment

In Requirement building limitations of the framework, functionalities of the framework, area, and dangers are characterized [36] [39]. Prioritization of Requirements is a significant stage during the time spent necessity designing. It is required considering the way that few out of every odd one of the prerequisites have matched the customer's fulfillment. The significant prerequisites may differ for every partner for instance for Project supervisor the necessities which are anything but difficult to actualize should execute first and the money related chief is strong for necessities that cost less. In this way, the essential goal of prerequisite prioritization is to pass on functionalities that accomplish the customer's endorsement, reasonable for the task's constraint, and transfer of necessities that are not commendable enough to be executed.

## 2.4 Process

Before prerequisites prioritization mechanism, the correct collection of partners should be settled and selected [33] [37]. Partners ought to have the necessary qualities as referenced before. A limited instances of necessities ranking procedures that can be used join, logical pecking order mechanism, numerical task, top-ten prerequisites or blend of methods [8]. By using the technique, a sorted out errand collection is delivered. Assignment develop an area set of essentials that are noteworthy for the endeavour. In coordinated setting, the requirements with high need will be at first completed in the chief accentuation. The picked requirements to be realized in a cycles are called run overabundance. During re-prioritization

process, the past unimplemented requirements are seen as together with various necessities in the once-over. The consequence of the method is a scramble aggregation for the accompanying accentuation.

### 2.5 Product

In case the Requirement prioritization process is executed reasonably, first class essentials could be made. Sufficiency is portrayed as how much orchestrated activities are recognized and masterminded results are cultivated [32]. The structure shows that experts should concentrate on the prioritization itself just as the accomplices and technique assurance process. To researchers, the proposed hypothetical framework includes some assessment openings in light-footed necessities prioritization. It is essential to devise segments for picking accomplices and requirements prioritization system in nimble improvement. The structure moreover seems to show that the essentials prioritization and re-ordering process in agile development is mind boggling. In this way, a deliberate procedure should be defined.

## 3 REQUIREMENT ENGINEERING PRACTICES

Requirement Engineering has been believed to be one in everything about vital stages inside the development technique. Here are some engineering practices used in agile methodology reflected in survey study.

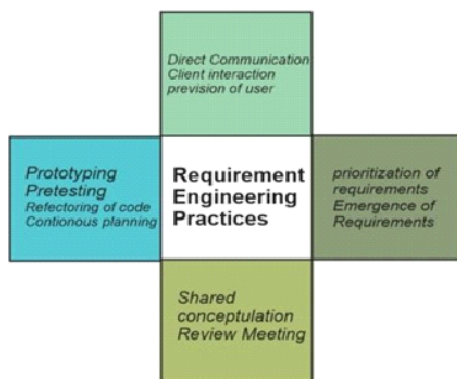


Figure 7. Requirement Engineering Practices

### 3.1 Direct Communication.

Direct correspondences should be energized between all colleagues and partners. Programming experts need to put time in setting up immediate and casual contact with supports [14]. Programming experts need to oblige such a strategy in their venture the board and arranging.

### 3.2 Client interaction and participation

The requirements of the clients are most significant contributions to produce any software product item [11]. These can be recognized with certain occasions and cost necessities and these are fundamental purpose behind the accomplishment of writing computer programs are the most significant contributions to the advancement of any product item [11]. These can be acknowledged with some-time and cost limitations and these are main reason for the success of software.

### 3.3 Prevision of user

User Requirements must be reasonable by end-clients and

clients who don't have a specialized foundation [11]. These prerequisites require client stories and create understanding between partners to changed style from documentation to discourse.

### 3.4 Prioritization of Requirement

Requirements Prioritization is an indispensable piece of basic leadership [12] [15]. Programming item quality is controlled by the capacity to satisfy the requirements of clients and clients. In this manner, evoking the prerequisites and recognizing the right necessities preceding the arrival of the appropriate necessities with legitimate usefulness is the significant advance to item achievement. This prioritization helps in recognizing the urgent prerequisites from the insignificant ones. The upsides of prerequisites prioritization are: it assesses the consumer loyalty's, it lessens modify and plan solidness. Prioritization is an integral part of decision-making [12] [15]. Software items quality is dictated by the capacity to satisfy the requirements of clients and users. Along these lines, inspiring the prerequisites and recognizing the right necessities before the arrival of the appropriate necessities with legitimate usefulness is the significant advance to item achievement. This prioritization helps in recognizing the significant necessities from the insignificant ones. The upsides of prerequisites prioritization are: it appraises the consumer satisfaction, it diminishes improve and plan soundness.

### 3.5 Emergence of requirements

In pursue, needs engineering isn't requested strategy, it's Associate in Nursing monotonous technique [15] inside which exercises are interleaved and prerequisite creates after some time affirmations to correspondence with partners at each progression of development. It removes vulnerability in necessity in a less time. Follow, needs engineering is not ordered method, it's Associate in Nursing repetitive method [15] within which activities are interleaved and requirement develops over time acknowledgments to communication with stakeholders at each step of development. It takes away uncertainty in requirement in a fewer time.

### 3.6 Feature update in change management

Each development cycle emphasis begins with investigation step. This progression incorporates necessity investigation, engineering examination and article examination [13]. The way toward keeping up feature incorporates including new features, adjusting or refreshing existing features and erasing existing features in agile necessities.

### 3.7 Continuous Planning

Persistent administration of necessities is a vital trait [17] so make an arrangement for a nonstop prerequisites engineering process for overseeing prerequisites in huge associations, from approaching necessities to that the prerequisites have be discharged as a piece of the item. Management of requirements is a crucial attribute [17] so make an arrangement for a constant requirements engineering practice for supervision necessities in huge associations, from approaching prerequisites to that the requirements have be discharged as a piece of the item.

### 3.8 Requirement Analysis Pairing

Stakeholders will play out a few jobs by matching of prerequisite examination [15]. Prerequisite investigation is a training at the top degree of the product design which deciphers partner needs and desires into a feasible

arrangement of programming necessities will perform several roles by pairing of requirement analysis [15]. Requirement analysis is a practice at the top degree of the software product design which interprets partner needs and desires into a practical arrangement of programming prerequisites.

### 3.9 Shared conceptualization

To help RE exercises in necessities gathering, prerequisites advancing for agile systems [18]. Refactoring of code Changing and improvement structure of create code in programming development approach that includes altering and tidying up recently composed programming code without changing the capacity of the code at all [14]. The essential reason for code refactoring is to make the code increasingly productive and viable. And improvement structure of develop code in software development approach that includes altering and tidying up recently composed programming code without changing the capacity of the code at all [14]. The fundamental reason for code refactoring is to make the code progressively proficient and viable.

### 3.10 Prototyping

Prototyping in increasingly conventional engineering disciplines is the normal methodology for exhibiting plausibility of the usefulness of a framework right off the bat in the existence cycle [14]. The Survey prerequisite with customers to get input from customers and further more helpful for hazard, appraisal and as a method for approval of end client necessities. In progressively customary designing controls is the normal methodology for showing attainability of the usefulness of a framework right off the bat in the existence cycle [14]. Audit prerequisite with customers to get criticism from customers and furthermore valuable for hazard evaluation promotion as a method for approval of end client necessities.

### 3.11 Pretesting

Error prevention involves great practice in programming engineering. Experiment driven assessment of pre-venture necessities process proposition to compose tests before utilitarian codes, overhauls contribution by tests involves great practice in software engineering. Experiment driven review of pre-venture prerequisites process proposal to compose tests before practical codes upgrades reaction by experiments [19].

## 4 REQUIREMENT ENGINEERING CHALLENGES

Requirement Engineering is a center procedure for Software development life cycle. Bugs in prerequisites don't appear to be known all through development rather they keep on being stowed away till framework gets operational and customer necessities don't appear to be met. Poor prerequisites lead to alterations in necessity particulars as well as require re-planning. These components include:



Figure 8. Requirement Engineering Challenges

### 4.1 Direct communication

The requirements engineering period of programming development ventures is described by the power and significance of correspondence exercises [1]. During this stage, the different stakeholders must have the option to impart their necessities to the examiners, and the investigators should have the option to convey the details they produce back to the stakeholders for approval. It is generally perceived that correspondence issues are a main consideration in the deferral and disappointment of programming ventures. Prerequisites particulars depend on area information be it specialized, useful, managerial or social. In a perfect world, the prerequisites colleagues are specifically enrolled so both the levels and appropriation of information inside the group spread all parts of the area.

### 4.2 Large interaction of client

A flourishing task includes a magnificent impact on educated and prepared stakeholders. Something else, programming may confront crucial dangers [3]. Insufficient specialized aptitudes with prerequisite architects and absence of area data will significantly affect programming. Prerequisites specialists' square measure unfit to enough address issues and finish client's necessities. Plus, some pioneer prerequisite specialists are additionally insensible to early necessity engineering apparatuses. In this manner, ineffectual execution by prerequisite specialists may prompts superannuated and blunder inclined necessities. conceivable

### 4.3 Conceivable Change Requirement

Change of requirement during programming development is one of the one of the huge test to dodge it we need to pursue appropriate systems, strategies and instruments [1]: working together intimately with the mentioning partner, conveying consistently inside the group, refining and organizing persistently the item build-up, preparing on request (Kanban), portraying in detail the necessities in the run build-up, looking into the outcomes normally, talking about the development level of a prerequisite with the group, gathering client stories to sagas, utilizing Kano examination.

### 4.4 Negligence of NFR

Non-functional requirements (NFRs) square measure determinant for the accomplishment of code comes [2]. In any case, they're portrayed as arduous to plot, and in agile programming development (ASD), square measure ordinarily given less need and much of the time not archived. In littler

enterprises, NFRs square measure conveyed through white board and diagram exchanges and designers' understood information is organized over documentation. Be that as it may, loss of detectability of NFRs, the issue in understanding NFRs by new designer's association the group and restrictions of documentation rehearses for NFRs square measure difficulties in requirement engineering.

#### 4.5 Estimation of budget and time

Scheduling is a procedure for arranging and overseeing time and spending plan. Planning them is one among the dominantly inconvenient activity and completely pivotal to programming achievement [3]. Be that as it may, regularly the time required in culmination of assignments all through necessity engineering part is thought little of. Thus, conveyance of achievements gets deferred altogether once assignments are on significant way. Fine difficulties suffer for prerequisite architects to oversee and achieve apparently boundless undertakings. Henceforth, prerequisite designers start to require easy routes or commonly disregard to pressure and target essential perspectives. Thus, needs are inadequately settled or gets postponed. Also, these pointless needs furthermore bring about downstream disappointment of whole programming.

#### 4.6 Next requirement innovation

Requirements engineering is a creative strategy in that partners and creators work along to make thoughts for fresh out of the box new frameworks that territory unit in the end communicated as prerequisites. The advancement gives bits of knowledge in engineer's brain to utilize inventive thoughts in making wanted properties of the since quite a while ago run framework [2]. This requires new ideas, advancements in finding, overseeing and giving expected help to clients. The administration provider ought to rapidly refresh the administration in step with the market request and rivalry. The administration provider will lead bunch activity procedures and workshops to make digressive reference for imagination in order development [1].

#### 4.7 Missing requirement

Missing requirements are significant issue in RE. Fair size frameworks have numerous necessities of huge frameworks can wind up with various thousand separate prerequisites. In this way, it is probability that significant prerequisites might be missed. On the off chance that we give iterative, steady development cycle then these minor slips can't cause a lot of mischief. These missing necessities later distinguished and added to later forms. Truth be told, it is probability that numerous data frameworks have various features which are not utilized by all clients and conceivably that not required by any means. These missing necessities are normally nonfunctional prerequisites. Most generally quality prerequisites express least satisfactory measures of value, for instance availability, interoperability, introduction, transportability, unwavering quality, strength, wellbeing, security, soundness, and ease of use [1].

#### 4.8 Conflicting Requirement

Making a mutual comprehension with respect to the significance of the 10,000-foot view by methods for an item vision, characterizing legends or sub objectives in the first place, dealing with the master plan as an obligation of the item proprietor, giving straightforwardness concerning changes among every, understanding association among client stories by methods for story mapping, envisioning client venture at the outset, including

clients.

**TABLE 1**  
*Challenges under every components*

Components	Challenges
Direct Communication	Minimum documentation and difficult to track requirement changes at different phases
Large interaction of clients	s Client interaction increased work load
Change requirements	Difficult to handle work and repetition in work
Negligence of NFR	Efficiency and usability
Estimation of Budget, time	High cost projects when change in requirements
Innovation in requirement engineering	Creativity issue in agility and in release
Missing Requirements	Repetition in development to find out the missing requirements
Conflicting requirements	Ambiguity in requirements cause large number of iterations

## 5 ASSESSMENTS AND FINDINGS

**TABLE 2**  
*Findings for challenges under every components*

Challenges	Solutions
Minimum documentation and difficult to track requirement changes at different phases	Collocated teams, Face-to-face communication
Client interaction increased work load	Proxy customers
Difficult to handle work and repetition in work	On-site customer
Efficiency and usability	Methods to tackle NFRs in agile methods
High cost projects when change in requirements	Legal measures , Fixed price contracts
Creativity issue in agility and in release	Methods to resolve Creativity issue
Repetition in development to find out the missing requirements	Methods to enable adequate requirements traceability
Ambiguity in requirements cause large number of iterations	Methods for value based requirements prioritization

Bjarnason, Elizabeth and Krzysztof Wnuk [30] had proposed an answer for the difficulties we are looking in direct correspondence they recommended that there ought to be Collocated groups and eye to eye correspondence. For defeat the difficulties of progress prerequisites they recommend arrangement of in the vicinity client. Blast and Tom J [21] proposed an answer of intermediary client to conquer the difficulties of enormous connection of customers. Tarhan, Ayca [27] distinguished strategies to handle nonfunctional prerequisites (NFRs) in agile techniques to defeat the issues of carelessness of NFRs. Yu, Yijun [31] suggested that there ought to be lawful measures and Fixed value agreements to maintain a strategic distance from difficulties in Estimation of spending plan and time. Tarhan [27] talked about difficulties of Innovation in prerequisite engineering then he infer that there ought to be techniques to determine imagination issue. Leffingwell [28] acquainted techniques with empower sufficient prerequisites recognize-ability to beat issues in missing necessities. We are confronting a few difficulties in Conflicting

necessities that reason equivocality in prerequisites and huge number of emphasis happens. Ben Othmane [29] distinguishes techniques for esteem based necessities prioritization. Ganesh [7] proposed an answer for Conflicting Viewpoints among Team. He prescribed that the agile colleagues should utilize steady specialized language in understanding the prerequisites. He also closes a response for plan varieties there should be right instructing of clients to indicate their prerequisites square measure essential assignment to beat these issues. Malik and M. Usman [9] talked about concerning the difficulties of absence of institutionalized RE exercises and contradictory interface to beat these difficulties agile statement and each one the methodologies should have institutionalized and recorded arrangement of RE exercises, and he developing necessities of the customer and front free form will have semantics holes inside the alternatives to downsize these glitches.

## 4 CONCLUSION

In the earlier decade, agile practices have gotten an immense overall acknowledgment related with programming bundle development on account of its inside arrangement of regarding individuals and associations, working system, purchaser participation, and responding to fluctuate. We will in general immovably believe that agile perfectly fits the dynamic methods for RE, on these lines movement like into the universe of RE can ensure an extra critical and a speedier achievement. These appraisal papers speak to a precise survey of agile methodologies, its constraints and difficulties in prerequisite engineering. Deliberate writing survey of associated papers has been made as referenced in A.V.Lamsweerde [12] in 2008. Concerning sixty examination papers were looked, rejection criteria was applied to waitlist applicable investigation papers. B. Ramesh, L. Cao [17] investigate the qualifications and correlations between agile methodologies and antiquated RE approaches and attempt to find what advantages of RE strategies will applied to Agile methodologies RE organizes square measure persistent and coordinated in many emphases and not clear in agile. Suggestions of creators are might want of documentation and prioritization in agile condition for future support and ramifications of unmistakable periods of RE. Finding of this examination is that crafted by analysts which will be useful for individuals who have an enthusiasm for finding entrancing space of investigation during this field because of numerous systems of agile (extraordinary programming, precious stone strategy) prerequisites extra examination and reasonable outcomes as prepared by contemplates. When procedures aren't very much sketched out and handled, clashes in necessities causes drawback. There would resemble of right structure for organizing prerequisites in agile environment. These days, Organizations square measure extra concentrating on giving prerequisites represent considerable authority in cost of clients and association perspective acknowledgments to time confinements, consider cost sparing and fulfillment of purchaser by organizing necessities, business esteem dependent on customer needs are regularly settled. There's eventual a procedure with plot criteria and strategy which will encourage in all around organized excess.

## 5 REFERENCES

- [1] Paetsch, Frauke, Amin Eberlein, and Frank Maurer. "Requirements engineering and agile software

development." WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.. IEEE, 2003.

- [2] De Lucia, Andrea, and Abdallah Qusef. "Requirements engineering in agile software development." *Journal of emerging technologies in web intelligence* 2.3 (2010): 212-220.
- [3] Alam, Sehrish, et al. "Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review." (2017).
- [4] Kumar, Manoj, Manish Shukla, and Sonali Agarwal. "A hybrid approach of requirement engineering in agile software development." 2013 International Conference on Machine Intelligence and Research Advancement. IEEE, 2013.
- [5] Batool, Asma, et al. "Comparative study of traditional requirement engineering and agile requirement engineering." 2013 15th International Conference on Advanced Communications Technology (ICACT). IEEE, 2013.
- [6] Schön, Eva-Maria, Jörg Thomaschewski, and María José Escalona. "Agile Requirements Engineering: A systematic literature review." *Computer Standards & Interfaces* 49 (2017): 79-91.
- [7] Ganesh, N., and S. Thangasamy. "Issues identified in the software process due to barriers found during eliciting requirements on agile software projects: Insights from India." *International Journal of Computer Applications* 16.5 (2011): 7-12.
- [8] Shah, Tejas, and V. S. Patel. "A review of requirement engineering issues and challenges in various software development methods." *International Journal of Computer Applications* 99.15 (2014): 36-45.
- [9] Malik, M. Usman, Nadeem Majeed Chaudhry, and Khurram Shahzad Malik. "Evaluation of efficient requirement engineering techniques in agile software development." *International Journal of Computer Applications* 83.3 (2013).
- [10] Farid, Weam M., and Frank J. Mitropoulos. "NORMATIC: A visual tool for modeling non-functional requirements in agile processes." 2012 Proceedings of IEEE Southeastcon. IEEE, 2012.
- [11] B.Thomas. "Meeting the challenges of Requirement Engineering." *News at Software Engineering Institute*. 2009. Website: <http://www.sei.cmu.edu/library/abstracts/news-at-sei/spotlightmar99pdf.cfm>. Access date: January 2010
- [12] A.V.Lamsweerde. "Requirements engineering in the year 00: a research perspective." *Inproceedings of the 22nd international conference on Software engineering, Limerick, Ireland*. 2008.
- [13] J.Highsmith, "Agile Software Development Ecosystems," Addison-Wesley, Boston, MA, 2002.
- [14] S. Bose, M. Kurhekar, J. Ghoshal, "Agile Methodology in Requirements Engineering," SETLabs Briefings Online. <http://www.infosys.com/research/publications/agilerequirements-engineering.pdf>, February, 2010.
- [15] F. Paetsch, A. Eberlein, F. Maurer, "Requirements engineering and agile software development," Eighth International Workshop on Enterprise Security, Linz, Austria, 9 - 11 June, 2003.
- [16] Sillitti, G. Succi, "Requirements Engineering for Agile



- methods," *Engineering and Managing Software Requirements*, Springer, 2005.
- [17] Ramesh, L. Cao and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Info Systems J*, "doi:10.1111/j.1365-2575.2007.00259.x", 2007.
- [18] S. R. Palmer and J. M. Felsing, "A Practical Guide to Feature-Driven Development," *The Coad Series*, 2002.
- [19] Sommerville and P. Sawyer, "Requirements Engineering A Good Practice Guide," John Wiley & Sons, 2000.
- [20] S. Ambler, "Agile Modeling: Effective Practices for Extreme Programming and the Unified Process," John Wiley & Sons, Inc. New York, 2002.
- [21] Beck, Kent, et al. "Manifesto for agile software development." (2001): 2006.
- [22] Bang, Tom J. "An agile approach to requirement specification." *International Conference on Extreme Programming and Agile Processes in Software Engineering*. Springer, Berlin, Heidelberg, 2007
- [23] Jeffries, Ron. "What is extreme programming" *XP magazine* 11 (2001).
- [24] Stoica, Marian, Marinela Mircea, and Bogdan Ghilic-Micu. "Software Development: Agile vs. Traditional." *Informatica Economica* 17.4 (2013).
- [25] Firesmith, Donald. "Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them." *Journal of Object Technology* 6.1 (2007): 17-33.
- [26] Asghar, Aneesa Rida, et al. "The impact of analytical assessment of requirements prioritization models: an empirical study." (2017).
- [27] Tarhan, Ayca, and Seda Gunes Yilmaz. "Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process." *Information and Software Technology* 56.5 (2014): 477-494.
- [28] Leffingwell, Dean. *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley Professional, 2010.
- [29] Ben Othmane, Lotfi, Pelin Angin, and Bharat Bhargava. "Using assurance cases to develop iteratively security features using scrum." *2014 Ninth International Conference on Availability, Reliability and Security*. IEEE, 2014.
- [30] Bjamason, Elizabeth, Krzysztof Wnuk, and Björn Regnell. "A case study on benefits and side-effects of agile practices in large-scale requirements engineering." *Proceedings of the 1st Workshop on Agile Requirements Engineering*. ACM, 2011.
- [31] Yu, Yijun, and Helen Sharp. "Analysing requirements in a case study of pairing." *Proceedings of the 1st Workshop on Agile Requirements Engineering*. ACM, 2011.
- [32] Bakalova, Z., Daneva, M., Herrmann, A., & Wieringa, R. . Agile requirements prioritization: what happens in practice and what is described in literature. *Requirements Engineering: Foundation for Software Quality*, 2011.p. 181-195
- [33] Ambler, S. *Agile modeling: Effective practices*: John Wiley and Sons; 2002.
- [34] Aurum, A., & Wohlin, C. *Engineering and managing software requirements*: Springer-Verlag New York Inc; 2005.
- [35] Heikkilä, V.T., Damian, D., Lassenius, C., Paasivaara, M.: A mapping study on requirements engineering in agile software development. In: *2015 Proceedings of the 41st Euromicro Conference on Software Engineering and Advanced Applications*, pp. 199–207 (2015)
- [36] Bjamason, E., Wnuk, K., Regnell, B.: 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 - AREW 2011*, pp. 1–5. ACM Press, New York (2011)
- [37] Ramesh, B., Cao, L., Baskerville, R.: Agile requirements engineering practices and challenges: an empirical study. *Inf. Syst. J.* 20, 449–480 (2010).