

System Requirements the Agile Way

M. V. L. N. Venugopal

Abstract: Requirement engineering evolved are not new today but are over decades and has become fundamental instrument in computer science. Requirement engineering is the most important thing in software engineering and development. It is the most important area of software development life cycle. The System Requirement Specification is the base for the software project in classic software methodologies, As Managing requirements is not an easy task; Producing high quality software products meeting the stakeholder's requirements is a major challenge in software engineering. Dynamically changing environments make changes to requirements in the software development process an inevitable task. Poor requirements and changes to requirements are major elements that cause software project failures. Requirements management and Agile developments are current areas of study. Many authors have written on requirements changes; however, most recent literature has not specifically studied an empirical approach to requirements management in an Agile-Scrum development process. This review explores requirements changes in an Agile-Scrum process. Agile methodologies focus on functional requirements This research intends to demonstrate how the application of the Agile-Scrum methodology produces software that meets user needs. If the traditional notions of requirements change are correct, then using the Agile-Scrum should deliver less functionality in a fixed-length development project. Aim of this review paper is also to present the limitations of requirement engineering phases in agile practices and the issues and challenges in implementation of agile practices. Agile methodology allows changes to requirements even late into the project with minimal impact on software functionality and quality of the delivered product

Keywords: system requirements, SRS, agile, usability, requirement management in agile, change management in agile and RE requirement engineering

1. Introduction

Software quality depends on many factors delivery on time, within budget and fulfilling user's needs. Software requirements are the foundations through which quality can be measured. Software quality during requirement engineering process can be maintained through requirement metrics. The System Requirement Specification is the base for the software project in classic software methodologies, The lack of its completeness causes uncertainty of the project foundations. This was one of motivations for agile methodologies if the SRS cannot be easily validated, if it can change in late project phases, then get rid of the SRS. Replace formal requirements with user stories. user stories are mostly functional requirements. As agile methodologies focus on functional requirements, it is easy to forget quality requirements. process and analyzing various requirement metrics, and software quality metrics that are involved during requirement process of a software development life cycle and analyzing impact of requirement metrics in software development environment. impact of quality requirements analysis on functional requirements is also very important. The research so far has shown that the increment of the revealed requirements count may be almost three times greater, compared to the standard requirement specification method.

In the software engineering field, requirements engineering has many definitions. "name given to a structured set of activities that help developers to understand and document system specification for the stakeholders and engineers involved in the system development"(Sommerville, 2001).

Requirement Engineering

Requirements engineering is the branch of software engineering concerned with the real-world goals for functions of, and constraints, on software systems. A high

quality RE procedure often dominates an effective project. Traditional Software Engineering (SE) usually includes the RE process which consist of requirements elicitation, analysis, documentation, management and validation Lot of issues associated to the Requirement engineering are output from the inadequacy or irregularity of specifications and requirements and conflicts between product stakeholders . Effective RE requires better consideration of the domain area, the circumstances in which the framework will run, and requirements of the product's stakeholders

The RE process consists of five main activities:

- Elicitation,
- Analysis and (Negotiation)
- Documentation,
- Validation and
- Management

Requirement Elicitation

Mostly the requirements are gathered from the customers by process know as elicitation. There are many elicitation techniques currently in practice in the software development. The first and foremost thing to choose the elicitation technique which best suites the software, as the best requirement elicitation technique will help getting the good requirements and it will accelerate the development process. Once requirements are collected the changes are inevitable. Changes keep on coming till the end of SDLC. It is the responsibility of the requirement manager or project manager or sometimes customer himself will decide which change to accommodate which change request needed to be ignored. Every software development company usually has the criteria for change management, and committee to decide on the changes I the requirements called steering committee and usually it is not fixed.

Requirements management

Requirements management is the process of documenting, analyzing, tracing, prioritizing, negotiating, agreeing on requirements and then controlling change and communicating to relevant stakeholders. It is a process which continues throughout the software development life cycle. Overseeing requirements in an exceptional case the requirement management model makes it hard to gather requirements with change request and makes it even harder to survey the effect of changes.

Even though where requirements are overseen, it is critical to get the “right” requirements from customer, and to get the requirements “right” in your software development plan, as the success of the software purely based on customer satisfaction. It is hard to survey the effect that a change to requirements has on plan at the beginning.

A recent survey depicts that the 46% of software development teams complains ‘Product does not meet customer needs’ as the reason product launch is either delayed or failed. What the customer wants is the question which seems simple but very difficult to answer. Requirements Management is considered as management of requirements it basically starts when a customer provides his/her needs or a process of software development is started. It consists of managing the requirements definition, explanation and changing requirements during the development cycle and systems development.

Steps in requirements management process.

- 1) Establishment of requirement management plan
- 2) Requirements are elicited.
- 3) The development of the vision document.
- 4) Use cases are developed.
- 5) Describes the supplementary Use case specifications.
- 6) Test cases from the uses cases are developed
- 7) Test cases from the supplementary specification.
- 8) Finally design is created

Requirements traceability

Requirements traceability is apprehensive with documenting the life of a requirement from idea to development and providing bidirectional traceability b/w a variety of requirements associated. It allows users to identify the root of each and every requirement and keep track of each and every change to that particular requirement. For keep track of every change made to the requirement it is required to document each and every change made to the requirement. Without traceability the change management is not possible and it is nearly impossible to comprehend the effect that a configuration change has on the customer requirements.

Requirement traceability steps

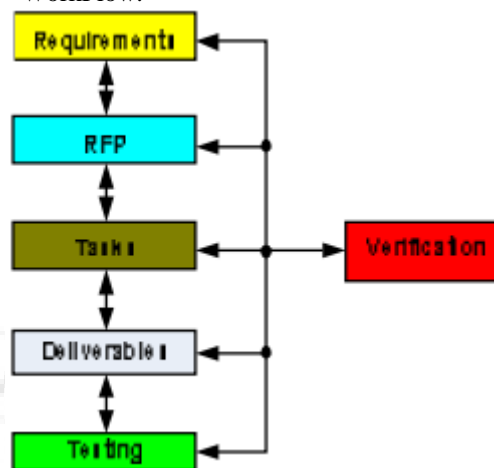
- Following the life of a requirement from idea to implementation, from root to the end of the requirement.
- How requirements impact each other, and how requirements impact other development lifecycle artifacts.

- The decomposition of requirements from high level customer, market needs to system, hardware component requirements and its transformation into design

Requirement traceability Matrix RTM

Requirements tracing, a process of documenting the links between the requirements and the work products developed to implement and verify those requirements. The RTM captures all requirements and their traceability in a single document delivered at the conclusion of the life cycle.

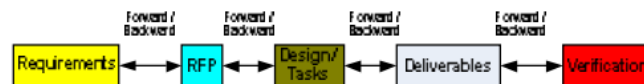
RTM - WorkFlow:



The diagram shows that the RTM can be used during all phases of a project to:

- Track all requirements and whether or not they are being met by the current process and design
- Assist in the creation of the RFP, Project Plan Tasks, Deliverable Documents, and Test Scripts
- Help ensure that all system requirements have been met during the Verification process.

 The Matrix should be created at the very beginning of a project because it forms the basis of the project’s scope and incorporates the specific requirements and deliverables that will be produced. The Matrix is considered to be bi-directional. It tracks the requirement “forward” by examining the output of the deliverables and “backward” by looking at the business requirement that was specified for a particular feature of the product. The RTM is also used to verify that all requirements are met and to identify changes to the scope when they occur



The Matrix is created at the very beginning of a project as it forms the basis of the project's scope and deliverables that will be produced. The Matrix is bi-directional, as it tracks the requirement forward by examining the output of the deliverables and backward by looking at the business requirement that was specified for a particular feature of the product.

Requirement traceability Matrix - Parameters:

- 1) Requirement ID
- 2) Risks Requirement Type
- 3) Requirement Description
- 4) Trace to Design Specification

- 5) Unit Test Cases Integration
- 6) Test Cases System Test Cases
- 7) User Acceptance Test Cases Trace to Test Script

The **Requirements Traceability Matrix (RTM)** is a document that links **requirements** throughout the validation process. The purpose of the **Requirements Traceability Matrix** is to ensure that all **requirements** defined for a system are tested in the test protocols.

In each of the steps shown above, each requirement must be unique and clearly defined. The requirement is then part of each critical component of the project. The references throughout the entire process must be consistent and unique. In order to insure that this occurs, the Matrix traces each requirement and creates a relationship between each of the processes

Req #	Name	RFP #	DDD#	PPT #	TS #	Verification
1	Calculate Interest	CGA 001	DDD 001	5.1.1 6.2.1	TS 001 TS 025	Yes / No
2						
3						

- Req #:** Requirement Number, for each project requirement, begin to list them on the RTM in a numerical order and group them by function.
- Name:** Enter the name and brief description of the requirement
- RFP #:** Request For Proposal (RFP); specify the identification number of the requirement as listed in the RFP.
- DDD #:** Deliverable Definition Document (Also referred to as the Deliverable Expectation Document- DED); use the RFP requirement number as a reference for the DDD that is created for the requirement.
- PPT #:** List the MS Project Subtask and Task numbers that are associated with the requirement.
- TS #:** Test scripts should be prepared for the actual testing process.
- Verification:** Use this field to record completion of the signoff process.

One of the six software development best practices¹ identified by the Rational Unified Process and the first of six key process areas² identified by the Software Engineering Institute's Capability Maturity Model (CMM) Level 2:

Repeatable is requirements management. According to CMM, the purpose of requirements management is to establish and maintain a common understanding between the customer and the project development team regarding the customer's requirements that must be addressed by the software project in order for the project to be successful. Typically, requirements are organized into one of the following two categories:

- Business and functional requirements
- Non-functional requirements

Business and functional requirements are user-specified functions, tasks, or behaviors that the proposed system must support.

Use cases are used to capture business and functional requirements in order to ensure that these requirements drive the software engineering effort through design, implementation, and validation. For this reason, this software engineering process is frequently characterized as a use case driven process and implicit in this process is the understanding that the use cases themselves document the business and functional requirements of the proposed

software system. For purposes of requirements management and traceability, however, business and functional requirements are also documented in a Requirements Traceability Matrix and mapped within this matrix to the use cases that comprise the Use Case Model. The capability to map these requirements increases the likelihood that upon the customer acceptance of a software solution that supports all the use cases within the Use Case Model, the project development team will have delivered a software product that fulfills the customer's needs. In contrast to business and functional requirements, non-functional requirements are system requirements that cannot be readily traced to specific use cases within the Use Case Model but are nonetheless critical to the overall success of the software project. In general, there are two types of non-functional requirements, qualitative and general systems. The first type of non-functional requirements are actually constraints upon the various functions, tasks, or behaviors that constitute the system's business and functional requirements and are, thus, commonly considered to be the qualitative aspects of the proposed software system.

Qualitative systems requirements include considerations like:

- Usability
- Reliability
- Performance
- Supportability
- Performance Measures

The other type of non-functional requirements is general systems requirements typical to most software engineering projects. No specific general systems requirements were gathered during the requirement workshops. However, these topics will be covered in the Software Architecture Document created during the Elaboration Phase of the project.

General systems requirements include considerations like:

- Security
- Relational Database Management System
- Backup and Recovery and Disaster Recovery
- User Documentation and Training

Because non-functional requirements cannot be traced to specific use cases within the use case model or, in some cases like security, apply to all use cases within the use case model, these requirements are typically not documented in the Requirements Traceability Matrix. Rather, non-functional requirements are documented and managed through a separate deliverable called a Supplementary Specification Agile is a more recent software development methodology introduced to help address some of these system development challenges. Agile±Scrum is an iterative development process becoming very popular in industry. However, as in all Agile methodologies, there is a resistance to the development of traditional documents. Instead of a requirements specification, Agile-Scrum follows the principles of the Agile development process. It provides the customer with the view of the product before and as each complete functionality is delivered. Agile Scrum method's main objective is to aim at prevention of common short

falls in the typical traditional development process. The development teams frequently iterate new increments of functionality. Stakeholders/product owners prioritize lists of required systems functionality, cost, timetables, and quality based on emerging business conditions. After the completed iteration, users and development teams collaborate on what to develop next, based on what was just developed and the new business needs.

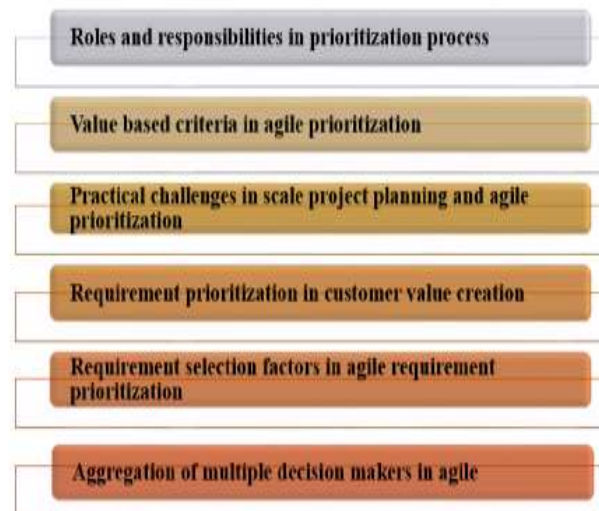
RE practices in Agile

Modifications in requirements arrive at stakeholders should be involved for different point of views for requirements gathering and interviews. There should be verification and validation of early description of functional and non-function requirements involved. Requirements traceability is ensured by adoption of requirement management, and is critical when the requirements are probably going to be changed. continuous feedback from customers is obtained in agile; however limits of agile practices are not well defined.

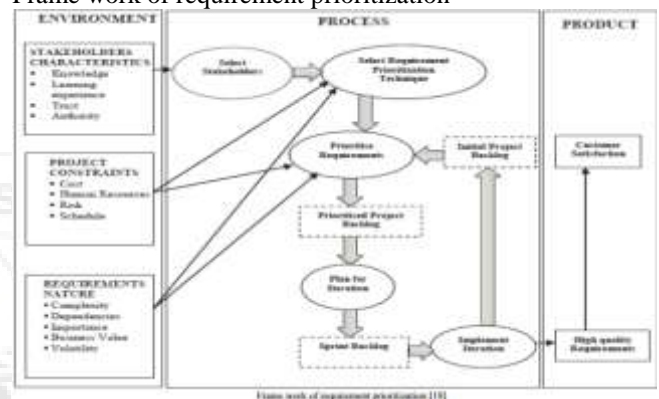
Approaches of agile practice to deal with requirements management and elicitation were started in 2005 . Purpose of these practices is to ignore past requirements and add new by continuous interaction with customers. Interaction between the Agile developers and the clients must be immediate and complete development team should be involved for requirements elicitation and gathering. In beginning of each iteration, requirements can be changed and prioritized so conclusion is that requirements are effectively managed in small teams by agile practices and customer involvement reduces wastage of time in requirements during production. The requirements are ineffectively handled in four agile practices requirements management, validation and verification, nonfunctional requirements and customer interaction but can be ignored in agile domain. Requirement management is basis for managing changes in customer requirements so agile methods must ensure it. For improving quality of agile process requirements validation is applied with requirements verification. Techniques are needed to identify nonfunctional requirements. However, more focus has been given to customer involvement and interaction but different techniques of interviews and context free questions should also be included.

Requirement Prioritization in agile:

The progress of innovation and business needs created client prerequisites to evolve almost each day. Agile practices have been acquainted with relieve such issues by actualizing client necessities incrementally and iteratively.



Frame work of requirement prioritization



Comparisons between Agile approaches and traditional RE

- 1) There is initial stage involvement of customers in traditional RE approach whereas agile involves stakeholders throughout in agile process.
- 2) Agile approaches are less documented and traditional approaches are well documented in detail.
- 3) Traditional RE approach uses diverse levels of abstraction models while Agile uses throw away model.
- 4) Agile methods use index cards for requirement management and change management in traditional approach.
- 5) The RE stages are repeated and merged in most iterations and not clear in agile.
- 6) Recommendations of authors are need of documentation in agile environment for future maintenance and implication of distinct phases of RE. This is not necessary in traditional.

Uses of Agile Requirement Engineering

Agile prototype answer queries about requirements content. Requirements (unstable state) are pre tested for detection of faults in extreme programming

Changes in architecture cause incorrect estimation and extra cost in requirements elicitation in agile environment. Iterative requirement engineering, face-to-face correspondence over specification in written form, utilization of review meetings and acceptance tests, prototyping, test bases development, constant planning for

change requirement management and prioritization of requirements. They dissect both of the advantages and difficulties of these practices and recommend that they are neither beneficial nor useless to establishments and can be used on basis of project specifications and conditions.

RE Issues/challenges in Agile Methodology:

Following are requirement engineering challenges that can be described in Agile

We identified six key challenges industry has to face today in terms of agile RE (see Table 5). In general experts weighted the identified challenges as important [23] and none of them rated one of the six key challenges as unimportant.

RE CHALLENGES IN AGILE

Features	Details	Challenge
Direct Communication	Communication between clients and teams members, minimum documentations in Agile. [20][21]	How to track changes in requirements?
Large interaction of clients.	Client interaction is required for large amount of feedback	Large work load
Change requirements	Handle changes	Work again and again
Negligence of NFR	User stories just store functional requirements	Security and usability
Estimation of Budget, time	Cant estimate due to changes requirements	Overrun and high cost project
Innovation in requirement engineering	Creativity in requirement engineering	Creativity issue in agility and in release
Missing Requirements	How to discover missing requirements?	A large no of iterations for missing requirements
Conflicting requirements	Conflict and ambiguity in requirements	A large no of iterations

2. Conclusion

Since more than decade from now, agile practices has picked up a huge global recognition. Idea of esteeming people and connections, working program, client cooperation, and reacting to change the concept of agile occupied center stage of software development. agile greatly fits the changing of RE, ad will ensure a more noteworthy and a quick achievement.

This paper represent a systematic review of agile methodologies, its limitations and challenges in requirement engineering RE stages are repeated and merged in most iterations and not clear in agile. There is definite need of proper framework for prioritizing requirements in agile environment. Nowadays, Organizations are more focusing on providing requirements focus on value of customers and organization viewpoint because of time limitations, concentrate on cost saving and satisfaction of consumer By prioritizing requirements , business value based on customer needs can be established. There is need of process with

defined criteria and process that can help in well prioritized backlog.

3. Acknowledgement

We thank our colleagues, who provided insight and expertise that greatly assisted this research. we thank all reviewers for their so-called insights

References

- [1] <https://www.omicsonline.org/open-access/an-improved-model-for-requirement-management-system-2165-7866-1000196.pdf> doi.acm.org/10.1145/253228.253258
- [2] https://link.springer.com/chapter/10.1007/978-3-319-57633-6_3
- [3] <http://www.sciencedirect.com/science/article/pii/S2212017313004064>
- [4] <http://ieeexplore.ieee.org/document/8065749/>

Author Profile

M.V.L.N. Venugopal received the M.Tech..in Computer Science Engineering from J..N.T.U. During 1990-1992and. M.Sc. in Electronics from O.U., respectively. During 1985-1987, he worked in TATA Info Tech Ltd. As a Sr Lead System Analyst. He has around 25 He worked in many domains including Telecom Domain. He now with PCI Private Limited