

Enhancing Quality in Agile Requirement Elicitation Process

¹Navneet Kaur, ²Moshiur Bhuiyan, ¹P.W.C. Prasad, ²Farzana Haque and ³A. Elchouemi

¹School of Computing and Mathematics, Charles Sturt University, Sydney, Australia

²Service Consulting, Enterprise Cloud Systems, Sydney, Australia

³Colorado State University Global Campus, USA

Article history

Received: 7-11-2018

Revised: 24-01-2019

Accepted: 25-02-2019

Corresponding Author:

P.W.C. Prasad

School of Computing and
Mathematics, Charles Sturt
University, Sydney, Australia
Email: cwithana@studygroup.com

Abstract: Eliciting requirements is one of the key aspects of the software development life cycle. Incorrect determination of the requirements can lead to the failure of the projects. Traditional waterfall method does not accommodate changing requirements easily while agile methodology welcomes the changing requirements even at the later phases of the development life cycle. The aim of the research is to improve the agile requirement elicitation process to ensure quality product is developed that aligns to the organizational strategy and its high value initiatives. Requirement elicitation process within agile projects should consider organisational actor's strategic dependency and rationale to ensure alignment with organisational strategy. We propose to improve the requirement elicitation process by combining *i**organizational models with standard Agile-Scrum methodology through a real-life health service provider scenario. The aim of our methodology is to ensure the social aspects of the organizational actor's strategic dependency and rationale are considered. We argue the requirement process could be significantly improved by combining these two aspects; Agile-Scrum methodology will address the requirements through user stories, vision document, epics and *i**organizational model will include the social aspects of the organization to reduce the gap between the project goal and organization goal.

Keywords: Requirement Engineering, Requirement Elicitation, Agile, Agile-Scrum, *i**Organizational Modelling, Agent Oriented Conceptual Modelling

Introduction

One of the critical aspects of the software development process is collecting, managing and understanding the requirements. It states the "what" and "why" of the system. Requirement elicitation is one of the most important phases of the software development life cycle because project's initiation is largely dependent on it (Rida *et al.*, 2016). There can be various reasons for the project failure but one of the main reasons for the project failure is due to the various requirement issues like less customer involvement, unclear expectations of the customer or any changes to the requirements. There are many examples where the project failed due to the poor requirement elicitation. The failure of the Therac-25, a computer-driven device for delivering careful bursts of emission to cancer patients (Porrello, 2012). Requirement engineering poses a big challenge to the software industry. To cope up with fast evolving software industry the requirements need to be understood to satisfy the customers and make the project a success.

In software requirement engineering, one of the major considerations is cost that is associated with the changing in requirements. In traditional models like the Waterfall model, Spiral model, Incremental Build Model, V-Model in all of them requirements are applied at the initial stages only because these models are not much flexible to acquire the changes in requirements at later stages as they are based on the sequential flow of activities. Requirements however could evolve throughout all phases from project initiation to closure. Failure of the project can have enormous impact on the organization, hence delivering a project with high quality within budget and satisfying stakeholder needs are one of the major goals of any software developing organization (Yousuf and Asger, 2015).

In traditional requirements elicitation methods it is difficult to make any changes in the latter phases and changing requirements at a later phase have a major impact in terms of budget and customer satisfaction. It can be expensive to fix the errors at the later stages as more cost and time will be involved resulting in the delay of the project. So, to overcome this limitation, the

AGILE-SCRUM methodology is one way for improving the requirement elicitation process as it can acquire changes anytime because every phase is continually revisited throughout the life cycle (Bagal-Kauthalkar *et al.*, 2016).

The substantial dependence on the human factor in agile development process is also a big challenge for the organizations. Incorrect role assignment, improper dependencies, lack of required skills can be some factors for the failure of agile methods. Thus, many types of software process models were introduced which focus on various aspects of software processes like product, activity, resource etc but none of them focuses on social aspect of a project such as motivation for performing a task, what skills are required and what are the dependencies among the team members. *i** modelling framework which is based on the concept of “intentional actor” was developed for analysing and relating the social and human aspects (Rida *et al.*, 2016).

Although agile methods limit the cycle of releases and enhance the user interaction, yet finding the best proposed targeted solution to clients, which are illustrative for their market, is an extreme procedure that might be aced simply after a few cycles/sprints/identifications of releases.

The main aim of the research is to improve the agile requirement elicitation process by combining the Agile-Scrum methodology with *i** modelling framework to develop high quality and high value products for customers. The aim of this paper is to illustrate a methodology to enhance the quality of agile requirement elicitation process. The proposed model is implemented by taking the scenario of the Hospital Management System and provides recommendations to achieve better results.

This research is based on improving the agile requirement elicitation process by discussing the limitations and issues with the traditional models and the current model for gathering the requirements and providing a proposed solution for improving the requirements by combining *i** framework and Agile-Scrum methodology.

Literature Review

Requirements engineering involves processes including identifying, documenting, analysing and validating the requirements for a project. The main purpose is to identify the need of the customers and organize them in such a way that helps in proper understanding, analysis, communication and implementation (Ramadan and Mohamed Abdelwahab, 2016). The RE framework consist of five main activities named Requirement Elicitation, Requirement Analysis, Requirement documentation, Requirement Validation, Requirement Management.

There are many traditional Models for requirement elicitation process like the Waterfall Model, Spiral

Model, Incremental Development Model, Prototyping and Rapid Application Development. Out of these models Waterfall Model is the most common one in which development proceeds to the next phase only when the previous phase is completed, there is no turning back. It is based on the sequential flow of the activities (Lesmana *et al.*, 2016). This model has many limitations like it involves a high amount of risk and security, it is very difficult to go back and change something once a project is in the testing phase, not a good model for complex and object-oriented projects and cannot be used for long and ongoing projects. To overcome these limitations agile methodology came into existence which is incremental and iterative approach, flexible in acquiring the changes even late in the development cycle. The term agile came into existence in 2001, in the Manifesto for Agile Software Development with the aim of delivering value to the customer, rapid feedback, more customer involvement and interaction and to ensure they are satisfied by the project delivered, more face to face communication, welcome the changing requirements and frequent delivery (Denning, 2015), (Alzoubi *et al.*, 2015). The agile methods overcame the limitations of the traditional models like Waterfall model, Spiral model in the way of accepting the changes in the requirements at later stages of development because they are based on “incremental” and “iterative “approach (Lucassen *et al.*, 2016). Every phase of the software development life cycle is divided into iterations and each iteration is revisited again and again to steer the process in another direction if some changes are to be done. The approach of this “inspect-and-adapt” greatly decreases the development cost and time to the organization (Lesmana *et al.*, 2016).

Our literature review identified few interesting works in the area of Agent Oriented Conceptual Modeling technique especially *i** framework combined with UML, BPMN, Goal Oriented Modelling (Zafar *et al.*, 2018; Bhuiyan *et al.*, 2018; Bhuiyan and Krishna, 2010; Sharma *et al.*, 2018). There is however very limited work on the Agile Requirements elicitation process by combing the *i** and Scrum.

There are various methodologies in Agile like Adaptive Software Development (ASD), Extreme Programming (XP), Feature Driven Development (FDD), SCRUM, Agile Unified Process (AUP) but this research focuses on SCRUM (Ince, 2015).

Comparison of Agile with Traditional Approaches

In the traditional methods, all the requirements are gathered before the design phase whereas in agile, requirements keep on changing throughout the development cycle. Pre-defined processes and more documentation are involved in traditional approaches whereas in agile less documentation and more customer interaction is involved. Other than that Agile assumes more

simplicity, embrace change, enable and focus on the next effort, maximize value, incrementally change, rapid feedback to all the stakeholders, quality deliverable and create documentation based on value (Bonner *et al.*, 2016).

Limitations of Agile

Although Agile has gained the popularity in the recent years it often pose challenges in large organizations where there is large number of stakeholder's involved and heavy interaction between the departments. Some constraints involved are general resistance to change, lack of communication and coordination, lack of management support, project complexity, customer collaboration, budget constraints, availability of human resources with required skills, availability to revolutionize the organizations culture, perceived time to transition, agile development teams perceive the user requirements in a wrong way (Saeeda *et al.*, 2015).

Scrum

Scrum is an agile management framework for managing the projects using self-organizing, one or more cross-functional teams. It is lightweight, simple to understand but difficult to master. It follows the principles of the agile development process. In this the customers are provided with a view before and after, functionality is delivered. Scrum uses short iterations called sprints which are no longer than 30 days. After every sprint a new functionality is produced. Until the completion of the project, there can be a change in technology or the customer can change their mind due to the uncertainties of what they want, in that case the Agile-Scrum can adapt to the required changes (Vithana, 2015). Scrum framework is illustrated in Fig. 1.

Terminology used in Scrum

Product Backlog: contains the list of all the requirements and all other business and technical functionality that are to be used while the development process.

Product Owner: the single person responsible for managing the product backlog, increasing the value of the product and work of the development team:

- **Development Team:** consist of experts, developers who deliver the project
- **Scrum Master:** has a leadership role and make sure the Scrum process is performed correctly
- **Sprint:** an iteration after which a new function is introduced
- **Daily Scrum Meetings:** the team plans a sprint before doing each sprint. Agree on the goals to be performed in the next sprint (Sachdeva, 2016)

In the recent years a lot of work has been done in agile like it is combined with various other methods to get the better results.

The author in (Lesmana *et al.*, 2016) combined Agile with Waterfall model to make a hybrid model for prevention information system of dengue viral infection. The numbers of dengue cases were increasing due to the lack of public response of dengue viral infections cases, reporting cases from hospitals, clinics or community health centre and without the valuable information it was difficult to control the ongoing outbreaks. To solve these problems, the author builds the prevention information system by combining Waterfall and Agile to identify the cases which helps the Health Department to decide what actions must be taken. It removes the limitations of both the models by bringing out the advantages of both.



©www.softwaretestingclass.com

Fig. 1: Scrum Framework (Ramadan and Megahed, 2016)

The author in (Elghondakly *et al.*, 2015) combined Waterfall with agile and proposed a novel automated approach for test case generation from requirements. In the proposed approach, SRS documents, non-functional requirements and user stories are parsed using the text mining and symbolic execution methodology for test data and validation in which requirements with various types are covered.

The author in (Younas *et al.*, 2016) provided a framework for agile development in cloud computing. As distributed agile software development poses a number of challenges like the lack of ability to capitalize on business opportunities, complex development processes, lack of communication between users, development teams and other stakeholders. The author proposed a framework to provide a skeletal or structural environment for distributed agile software development in cloud computing environment to solve these problems.

The author in (Jeyasingham, 2014) combined agile with goal oriented modelling to highlight the importance of social modelling in agile as incorrect role assignment, neglected team dependencies and not having enough required skills can be the reason for the failure during the introduction of an agile method. So the author uses the Scrum process for depicting the social aspects of agile methods.

As this paper focuses on improving the requirement elicitation process and for this agile is combined with the *i** organizational framework which is based on the concept of “intentional actor”. As in each organization every actor has some properties such as skills, abilities, goals, beliefs and they depend on each other to achieve the desired goals (Woldeamlak *et al.*, 2016). It consists

of two models Strategic Dependency and Strategic Rationale model. In Strategic Dependency model an actor depends on each other for a goal to be achieved, task to be performed or depends for a resource to be furnished. It highlights four dependencies namely goal dependency where an actor depends on another actor to achieve a goal, softgoal dependency where an actor depends on another actor to achieve a softgoal, resource dependency where an actor depends on another actor for a resource, task dependency where an actor needs another actor to perform a task. Figure 2 represents the SD model for Hospital Management system which involves five actors that is Doctor, Nurse, Patient, Receptionist and Medicare which depend upon each other for a goal, resource and task. For example, patient has goal dependency on doctor to give medicines, nurse has a resource dependency on patient to provide medical reports, patient has a task dependency on receptionist to give appointments and doctor is dependent on the patient to achieve a softgoal to be happy.

Strategic rationale model gives the more detailed information by defining the internal structure of each actor through task decomposition and means-ends link. Mean-ends link states which soft goal, task and/or resource contribute in achieving a goal. Task decomposition link states the necessary sub-tasks of a task. Contribution link represents the negative or positive impact on softgoal by task or other softgoal (Ulbrich and Borman, 2017). Figure 3 represents the SR model for Hospital Management System where actor has its specific boundary and their internal structure is defined in detail. It illustrates the hierarchical decomposition of high level tasks or goal into lower levels e.g.

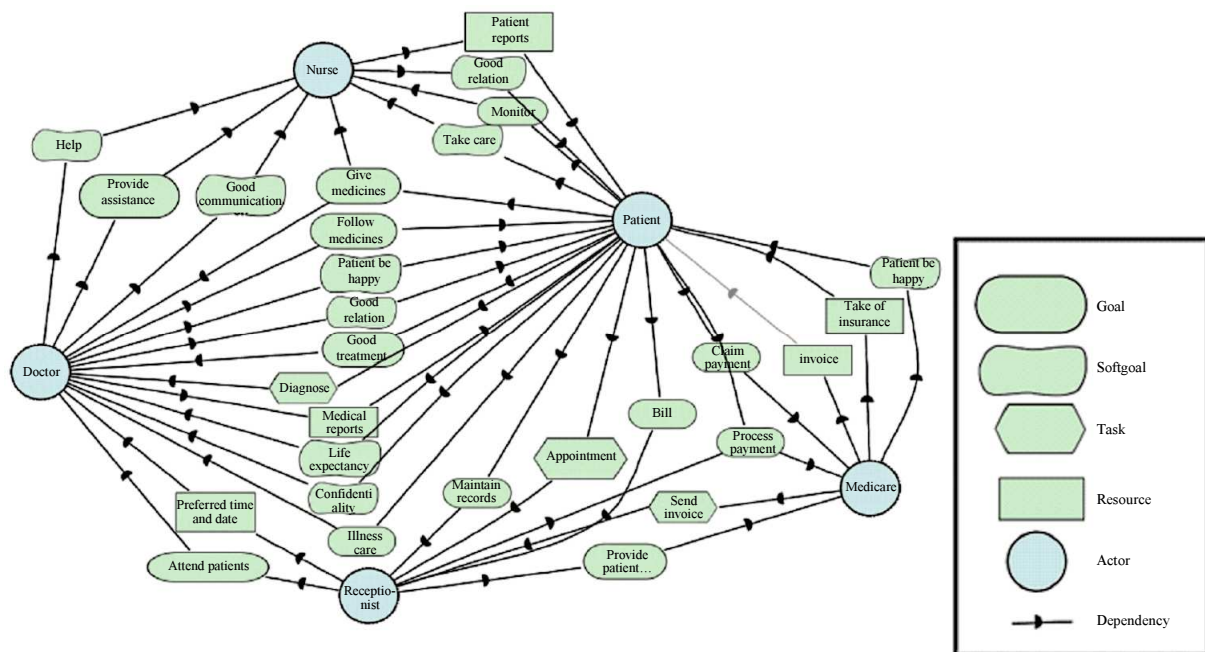


Fig. 2: SD Model for Hospital Management System

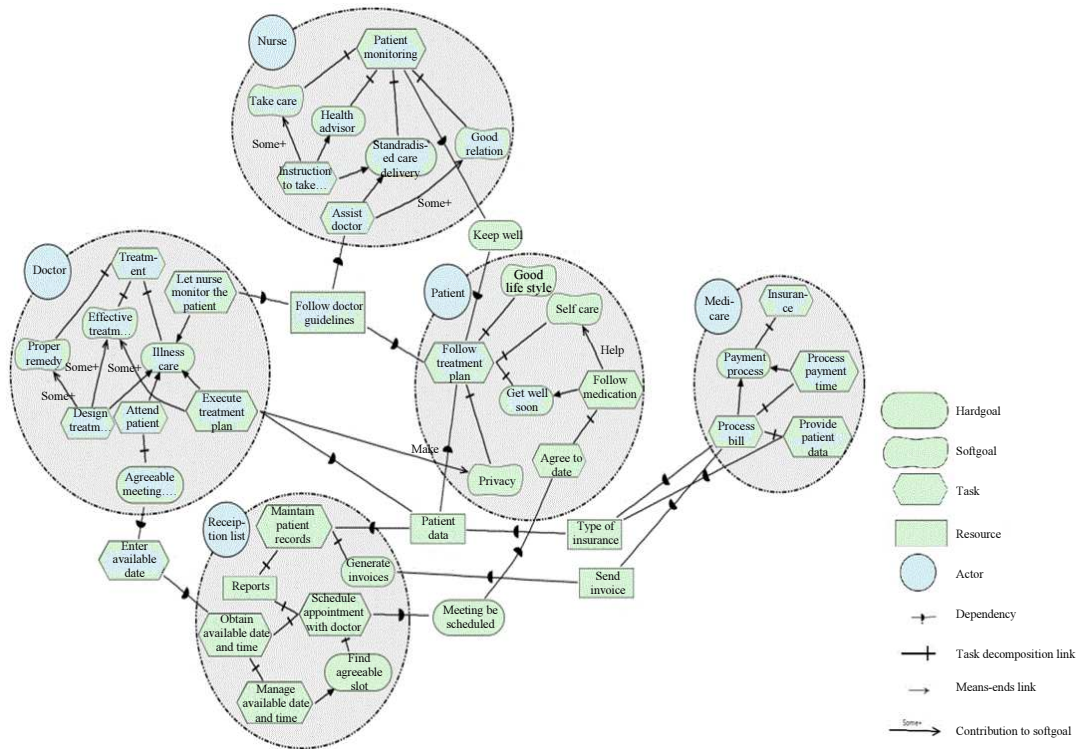


Fig. 3: SR Model for Hospital Management System

Treatment Plan is decomposed into Good Life Style, Self-Care, Get Well Soon), explicit additional skills (e.g. softgoal Good Relation), specifies how the various intentional elements of the actor contribute to achieve each other’s goal (e.g., Follow Medication helps in achieving self-care softgoal).

Proposed Methodology

In the proposed solution, we are combining two methods for Requirement Elicitation Process that is Agile-Scrum methodology and *i** modelling framework to improve the agile requirement elicitation process. Social aspects need to be considered in Agile-Scrum development because every actor involved in the project is depended on another for a resource, to achieve a goal or a task to be performed. Although the project has its own goals and objectives but along with that it is believed that, organizational goals and objectives also need to be met. So it is important to consider the strategic dependency and rationale models to ease the requirement process. By doing so the gap between the project objectives and organizations objective is potentially minimized.

To demonstrate these principles (refer to Fig. 4), certain steps will be followed to improve the requirement elicitation process:

Step 1: Develop Strategic Dependency model using *i** organizational framework.

- Step 1.1: Identify the internal and external actors of the hospital management system.
- Step 1.2: Requirements are identified as goal dependency, softgoal dependency, task dependency, resource dependency between the actors.
- Step 2: Develop Strategic Rationale model using *i** organizational framework.
- Step 2.1: Gives the detailed internal structure of the actors by specifying the boundary
- Step 2.2: Specifies the hierarchical decomposition of high level goal/tasks into lower level ones
- Step 2.3: Shows how the various intentional elements of the actor contribute to each other to achieve the goal.
- Step 3: Develop the project vision (refer to Table 1 for the vision developed for the hospital management system)
- Step 3.1: map the elements FOR, THE, WHO, THAT, UNLIKE in vision document with actors, goal, task or resource in *i** models.
- Step 4: Develop Service Box by mapping its elements with *i** models.
- Step 4.1: Identify the front of the box which includes service name and its selling points and objectives.
- Step 4.2: Identify back of the box which includes feature list and description.

- Step 4.3: Identify the assumptions.
- Step 4.4: Identify the risks and constraints involved.
- Step 4.5: Identify how the success will be measured.
- Step 5: Create the epics and user stories.
- Step 5.1: The goal, tasks and resources in *i** models are elaborated into epics and user stories are elaborated into epics and user stories.

The whole Hospital Management is not developed using this model, just an example is given that this proposed solution can be used in various scenarios to gather the requirements.

Here is the generalized figure of the proposed solution in which firstly the Strategic Dependency and Strategic Rationale models are developed then the agile vision document is made followed by service box, epics and user stories. As our proposed methodology says that social aspects need to be

considered in agile so the elements in vision document, service box, epics and users stories are mapped with *i** organizational models, each and everything is related to each other.

Application of the Proposed Methodology in the Context of Hospital Management System

In this section we will illustrate application of our methodology in the context of a Hospital Management Systems scenario.

Step 1: Strategic Dependency Model

Figure 2 represents the strategic dependency model of the hospital management system which includes five actors named Doctor, Nurse, Patient, Receptionist and Medicare. Each actor is depended on one another for a goal, softgoal, task or resource.

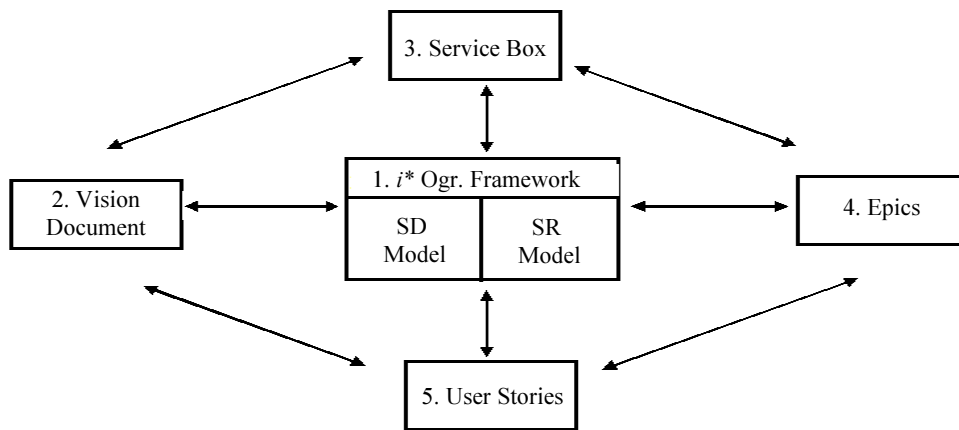


Fig. 4: Process of proposed methodology

Table 1: Vision statement

| Vision statement | | |
|------------------|----------------------------------|--|
| FOR | Target customers or stakeholders | These are the actors like the doctors, nurses, patients or other staff of the hospital who will get benefit. |
| WHO | Statement of need | By considering the actors strategic dependency and rationale the agile requirement elicitation process can be improved like by considering the requirements of patients, nurses, doctors, receptionist in a hospital system can improve the process. |
| THE | Product being developed | A methodology is being proposed by combining Agile-Scrum and <i>i*</i> modelling framework. Strategic dependency and rationale of the actors give a clearer insight of the requirements of what they actually want. |
| THAT | Functional improvement | The overall process will enable the doctors and other medical staff to give the patients the best possible care. |
| UNLIKE | Primary competitive alternative | The current issues can be resolved because this Methodology will give a clearer understanding of what the customers actually want. The things that do not fall under SD and SR should be omitted. |
| OUR PRODUCT | Primary differentiation | Easy to understand the requirements by involving various actors because every actor is depended on another for goal, task or resource and further developing it in the agile gives more understanding of requirements that will help the patients to get them get possible care. |

Step 2: Strategic Rationale Model

Figure 2 represents the strategic rationale model for the hospital management system where each actor (Doctor, Nurse, Patient, Receptionist and Medicare) has its own specific boundary which gives the detailed information of their internal structures.

Step 3: Establish Vision

Every Agile-Scrum project needs a vision statement that acts as the projects true north, sets the goals and objectives and directs the Scrum team. The vision document guides “the goals need to be achieved” statement that the scrum master, product owner, development team and the stakeholders refers to throughout the project. It is basically a clear, concise statement of problem, proposed solution and the features of the product that help in finding expectations and reduce risk.

The Vision Format in Agile is as follows:

For<target customers>
Who<Statement of need/opportunity>
The<product>is a <product category>
That<functional improvement/benefit>
Unlike<primary competitive alternative>

Now this format is applied to the Hospital Management and *i** models are referred and the elements in vision document are mapped with the actors, their goal, tasks and resources.

Step 4: Service Box

After the vision document the next step is to develop the Service Box. The service box includes designing the front and back of the box which contains the detailed description of the services provided, their objectives, risks involved, assumptions and the ways to measure the success.

The general format of the service box in agile is:

- Front of the Box
 - Service name:
 - Selling points and objectives:
- Back of the Box
 - Service description
 - Features list
- Assumptions
- Risks and constraints
- How will we measure success?

Now this format is applied to the example taken that is hospital management system and description of one service that is the goal of the doctor in *i** models is described.

Front of the Box

Service Name: Cure illness of Patients

Selling points and objectives:

- Diagnose the patients
- Treat patients effectively
- Make a treatment plan for the patients
- Reduce the likelihood of a disease
- Doctors are able to communicate with patients properly
- Doctors understand the problems of the patients
- Doctors are able to tell the truth to the patients

Back of the Box

Service description: Patients come to the hospital to cure their illness and for that doctors need to understand their problems and give them treatment accordingly.

Features list:

- Improved quality life of patients
- Good health
- Patients be happy
- Prolongation of life
- Relief from suffering
- Patients feel secured as their data or history is secured.

Assumptions

- Each actor involved performs his duties and responsibility effectively.
- The proposed model is able to cope up with the entire system of the hospital.

Risks and Constraints

- Doctor does not understand the illness of the patient properly.
- Treatment plan does not suit the patient.
- Incorrect prescription can to death.
- The reputation of the hospital can be effected.

How will we Measure Success?

- Feedback from the patients
- More patients are cured
- The model proposed can be used in future for the various business projects.

Step 5: Epics and User Stories

Epic: It is basically a large body of work or a user story that can be split into smaller stories. It can be a user requirement, their request or a feature of the product that needs to be developed. It probably takes more than one sprint for completion and does not specify all the details at this level that the development team require to work upon. These details are defined in user stories.

User stories: They are defined as the smallest piece of work or task in agile framework. They are expressed in few short sentences in simple language which tells the desired goal to be achieved. Each user story has an Acceptance Criteria or the Definition of Done which adds more conversations of what the users want (Hudda *et al.*, 2016).

Format of User Story in Agile

“As a [type of user] I [want/can/able to/need to/etc.] so that [some reason]” (Pandit and Tahiliani, 2015).

Now based on this, some examples of Epic and for that Epic the user story for various actors is given for Hospital Management System by translating the goals, tasks and resources into epics and user stories.

- Epic1: Treatment plan need to be followed to cure the illness the patients.
- UserStory1: As a doctor, I need to design the treatment plan for the patients so that I can cure their illness.

Acceptance Criteria

- AC1: Doctors knows the patient’s problem
- AC2: Design the treatment plan according to his problem
- AC3: Cure the illness of patients
- Epic2: Nurse provides assistance to the doctor to help him to treat the patients.
- UserStory2: As a Nurse, I need to help the doctor so that the patients get the best possible care.

Acceptance Criteria

- AC1: Good Communication between the Nurse and the Doctor.
- AC2: Nurse understands the treatment plan designed by the doctor.
- AC3: Improved quality care of patients.

Epic3: The receptionist maintains the patient records which include their personal information and their medical history.

UserStory3: As a receptionist, I need to make list of patients including their personal information and medical history who are admitted to the hospital so that data is stored in the database.

Acceptance Criteria

- AC1: Receptionist is able to maintain the patient’s record including their personal information and medical history.
- AC2: Save the data to the database.

- AC3: Close the registration page.

Epic4: Patient Registration

UserStory4: As a patient, I need to get appointment from the receptionist so that I can see the doctor.

Acceptance Criteria

- AC1: Patients are able to get the appointment from the receptionist.
- AC2: Receptionist gives the appointment.
- AC3: Patients sees the doctor.

Discussion

By considering the strategic dependency and rationale in agile environment, we could incorporate the actors’ inter dependencies using the notations such as roles, goals, tasks, positions and dependencies which focuses both on social and intentional ontology. It represents the social domain of the software systems. It develops the understanding of intentional desires, maximizing the potential for software systems to solve the real problems by aligning the solution to organisational goals and objectives. Eliciting incorrect or inconsistent requirements can lead to the failure of the projects; hence highlighting the critical roles, by clarifying the actor’s responsibilities and their expectations gives a better understanding of the process. It helps in understanding the requirements in a better way as every project involves actors and by considering the dependencies, goals and tasks of these actors and then mapping those in agile gives a clear understanding of the requirements. The goals, tasks, resources are more detailed in vision document, services, epics and user stories giving a clear view of what the user wants. Organizational goals and objectives need to be met along with project goals; by combining organizational models to requirement elicitation potentially bridges the gap between the project objective and organization objective.

Conclusion and Future Work

Requirement engineering plays a major role in the success or failure of any project. It is important to ensure requirements are understood accurately in order to achieve the desired goal and meet customer expectations. As requirements can potentially change till the end, the traditional methodology need to be adaptable even if they are based on sequential execution. Scrum gained popularity because of the ability to accommodate the changes even at the later stages of the project, able to deliver the product frequently. Our paper focuses to improve the requirement elicitation process by combining Agile-Scrum method with *i** organizational framework by taking the example of the hospital management system. It clearly represents the dependencies between the actors, the responsibilities

assigned to each actor. These dependencies are then mapped with the agile vision document, service box, epics and user stories which gives a clearer understanding of the requirements. This methodology ensures the alignment of project objectives and deliverable to organisational goals and objectives. We plan to apply our methodology to real time industrial projects in multiple domain.

Author's Contributions

Navneet Kaur: Navneet's contribution includes literature review, establishing principles of service design, organisational design and development of the framework.

Moshiur Bhuiyan: Supervised this work including providing guidance, direction and contribution to the methodology.

P.W.C. Prasad: Supervised/worked closely with Navneet during the analysis, design and experiment phases.

Farzana Haque: Contributed to the refinement of the methodology and assisted in final review.

A. Elchouemi: Contributed in the technical content, checked the final review and provide the required funding for the paper.

References

- Alzoubi, Y., A. Qumer Gill and A. Al-Ani, 2015. Distributed agile development communication: an agile architecture driven framework. *J. Software*, 10: 681-694.
- Bagal-Kauthalkar, S., S. Chavan and M.B. Ansari, 2016. Review on agile software development. *Int. J. Trends Eng. Res.*, 3: 150-158. DOI: 10.21884/ijmter.2016.3058.nzrjk
- Bhuiyan, M. and A. Krishna, 2010. Business modeling with the support of multiple notations in requirements engineering. *PACIS*, 2010: 175-180.
- Bhuiyan, M., F. Haque and L. Shabnam, 2018. Integration of organisational models and UML use case. *JCP*, 13: 1-17.
- Bonner, N., N. Kulangara, S. Nerur and J. Teng, 2016. An Empirical Investigation of the Perceived Benefits of Agile Methodologies Using an Innovation-Theoretical model. *J. Database Management*, 27: 38-63.
- Denning, S., 2015. Updating the agile manifesto. *Strategy Leadership*. DOI: 10.1108/sl-07-2015-0058
- Elghondakly, R., S. Moussa and N. Badr, 2015. Waterfall and agile requirements-based model for automated test cases generation. *Proceedings of the IEEE 7th International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp: 607-612.
- Hudda, S., R. Mahajan and S. Chopra, 2016. Prioritization of user-stories in agile environment. *Indian J. Sci. Technology*. DOI: 10.17485/ijst/2016/v9i45/105069
- Ince, C., 2015. Approaches and benefits for adopting agile methods. *INSIGHT*, 18: 18-20. DOI: 10.1002/inst.12042
- Jeyasingham, D., 2014. Open spaces, supple bodies? Considering the impact of agile working on social work office practices. *Child Family Social Work*, 21: 209-217.
- Lesmana, I., R. Karimah and B. Widiawan, 2016. Agile-waterfall hybrid for prevention information system of dengue viral infections: A case study in Health Department of Jember. East Java, Indonesia
- Lucassen, G., F. Dalpiaz, J. van der Werf and S. Brinkkemper, 2016. Improving agile requirements: The Quality User Story framework and tool. *Requirements Eng.*, 21: 383-403. DOI: 10.1007/s00766-016-0250-x
- Pandit, P. and S. Tahiliani, 2015. AgileUAT: A framework for user acceptance testing based on user stories and acceptance criteria. *Int. J. Computer Applications*, 120: 16-21. DOI: 10.5120/21262-3533
- Porrello, A.M., 2012. Death and denial: The failure of the THERAC-25. *A Medical Linear Accelerator*.
- Ramadan, N. and I. Mohamed Abdelwahab, 2016. A security testing framework for scrum based projects. *Int. J. Computer Applications*, 138: 12-17. DOI: 10.5120/ijca2016908928
- Ramadan, N. and S. Megahed, 2016. Requirements engineering in scrum framework. *Int. J. Computer Applications*, 149: 24-29.
- Rida, A., S. Nazir, A. Tabassum, Z. Sultan and R. Abbas, 2016. Role of requirements elicitation and prioritization to optimize quality in scrum agile development. *Int. J. Advanced Computer Science Applications*. DOI: 10.14569/ijacsa.2016.071239
- Sachdeva, S., 2016. Scrum methodology. *Int. J. Engineering Computer Science*. DOI: 10.18535/ijecs/v5i6.11
- Saeeda, H., F. Arif, N. Mehmood Minhas and M. Humayun, 2015. Agile scalability for large scale projects: Lessons learned. *J. Software*, 10: 893-903. DOI: 10.17706/jsw.10.7.893-903
- Sharma, D.P., F. Haque, M. Bhuiyan and P.W.C. Prasad, 2018. Service optimization using goal-oriented modelling in judicial domain. *J. Software*, 13: 287-299.
- Ulbrich, F. and M. Borman, 2017. Extended dependency network diagrams: Adding a strategic dimension. *J. Global Operations Strategic Sourcing*, 10: 42-66. DOI: 10.1108/jgoss-05-2016-0019
- Vithana, V., 2015. Scrum requirements engineering practices and challenges in offshore software development. *Int. J. Computer Applications*, 116: 43-49. DOI: 10.5120/20472-2649

- Woldeamlak, S., A. Diabat and D. Svetinovic, 2016. Goal-oriented requirements engineering for research-intensive complex systems: A case study. *Systems Engineering*, 19: 322-333.
DOI: 10.1002/sys.21355
- Younas, M., I. Ghani, D. Jawawi and M. Khan, 2016. A framework for agile development in cloud computing environment. *J. Internet Computing Services*, 17: 67-74.
DOI: 10.7472/jksii.2016.17.5.67
- Yousuf, M. and M. Asger, 2015. Comparison of various requirements elicitation techniques. *Int. J. Computer Applications*.
DOI: 10.5120/20322-2408
- Zafar, U., M. Bhuiyan, P.W.C. Prasad and F. Haque, 2018. Integration of use case models and BPMN Using goal-oriented requirements engineering. *JCP*, 13: 212-221.