

# Integration of Requirement Engineering and Artificial Intelligence: Agile Practices and Case Based Reasoning

Mehwish Mukhtar<sup>1\*</sup> and Yasir Hafeez<sup>2</sup>

<sup>1</sup>COMSATS, Wah Campus, University Institute of Information Technology, Pakistan

<sup>2</sup>PMAS Arid Agriculture University, Rawalpindi, Pakistan

## Abstract

Guidelines to the Requirement engineering along with the Agile team regarding improvement of Agile practices incorporated with Case based reasoning (CBR).

**Context:** Applying Artificial Intelligence (AI) techniques on Agile practices to software development have acknowledged slight attention until now.

**Objective:** In this research, we present a model which has been introduced for evolution of the Agile Software Development practices by using AI technique.

**Method:** In this research, we have used published reports, articles, and existing case studies. Expert's review method is used to accomplish the appraisal of this model.

**Result:** This model has been resulted in improving the agile practices by using the concept of CBR. An addition to the SCRUM process it is beneficial to cope with reusability of requirements solutions in an Agile Development environment.

**Conclusion:** The framework provides guidelines to the requirement engineering and also to the agile team regarding improvement of agile practices incorporated with CBR.

**Keywords:** Requirement engineering; Software development life cycle; Hybrid software engineering process; Rational unified process model; Agile; Scrum model; Extreme programming; Software engineering practices; Case based reasoning

## Introduction

Requirement Engineering (RE) is the most important part and primary phase of software development lifecycle. Requirement Engineering (RE) definition provided by Zave has clearly explained the concept of RE: RE is the subdivision of Software Engineering (SE) compacts with the real world objectives for, tasks of, and restrictions on software systems. It also deals with the dependency of these factors to detailed specifications of software performance and to their development over time and software areas. Adopting RE practice provides assurance that the development process follows the standard. The processes used for RE mostly differ due to the difference in application domain, the people involved and the organization developing the requirements [1]. Instead, RE is a traditional software engineering process with the aim to recognize, examine, document and authenticate requirements for the system to be developed. On the other hand, RE and Agile approaches are seen being unsuitable because of following reasons: RE often severely depends on documentation for information sharing while agile methods concentrating on head-on association between customers and developers to reach the same objectives.

Relationship between Agile development and Requirements Engineering is very composite. RE offers practices for understanding user requirements, analyzing the actual needs, evaluating feasibility, reaching a logical solution [2]. This depicts that the ultimate goal of requirements engineering which is to specify actual requirements clearly and present them in a more applicable way for further phases. Agile is an iterative process are handled in which requirements evolve throughout the process and handle by Agile Practices because they are flexible and reliable process in which all development phases are completed in shorter sprints and change is easy to manage. Agile

Software Engineering represents alternative to conventional Software Engineering (SE) for sessions of software and definite type of software projects. It has been verified to deliver successful systems rapidly.

In existing researches it has observed and also mentioned by the developers that there are reduced documentations in Agile practices because every team members has the same level of skill and information about the systems. So if any of the team members leaves the organization or quits the project for a short span of time, still there but exists a lot of shared knowledge among other team members. However, in actual working environment that equal level of knowledge is not achievable [3-7].

Agile Framework is based on values i.e., Active communication between Team and Users, Iterative and Incremental, Continuous Delivery and Flexible Approach to Development [8]. Each module is built as an independent working module and integrated after the completion of a release. The agile models provide processes that are light weight and able to accommodate rapidly changing requirements [9,10]. Agile software development (ASD) methodologies focus on iteration that must not consume more than two weeks. The vital function of ASD is that testing is performed and also the integration is accommodated very quickly [8].

**\*Corresponding author:** Mehwish Mukhtar, COMSATS, Wah Campus, University Institute of Information Technology, GT Road, Quaid Avenue, Pakistan, Tel: +9251931438283; E-mail: [mehwish.mukhtar289@gmail.com](mailto:mehwish.mukhtar289@gmail.com)

**Received** July 13, 2017; **Accepted** August 31, 2017; **Published** September 04, 2017

**Citation:** Mukhtar M, Hafeez Y (2017) Integration of Requirement Engineering and Artificial Intelligence: Agile Practices and Case Based Reasoning. J Comput Sci Syst Biol 10: 070-078. doi:10.4172/jcsb.1000252

**Copyright:** © 2017 Mukhtar M, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

It is also highlighted that the interaction between both fields i.e., AI and SE is so speedy on the features that provide advantage in future. In this paper, the areas of RE interaction and the factors of interaction between both fields are identified [11].

One of the long-lasting Object-oriented (OO) software development processes: Rational Unified Process (RUP). A great number of frequently used superlative practices of modern software development highlight its adoption. RUP intertwines these approaches into the following terms: Roles, Disciplines, Activities, and Artifacts. RUP is a repeatable process that is classified into four steps of software development for any project namely: Inception, Elaboration, Construction, and Transition phases. The XP model is similar to that of RUP but it differs in number of cycles at each stage and it has only 4 phases - Design, Code, Test and Listen [12]. The lightweight OO Software Development process named as Extreme Programming (XP) is based on four morals namely: Communication, Simplicity, Feedback, and Courage.

In traditional RE, many approaches have been recommended to manage and assimilate different RE activities and products and issues including: unnatural language and incomplete requirements, developing information based systems and listing of requirements. RE is the initial and vital part in the software development lifecycle. The quality of software building depends upon this phase which is directly proportional to the quality of RE. Good quality of RE does not the guarantee success in software product but a low quality definitely brings failure.

The intersection scenario of AI and SE is so vast that it requires updating of framework. So integration of Agile Practices with (AI) techniques like Case Base Reasoning (CBR) is used to make the RE process more effective by extended expansion and ratification of the requirements of Agile Software Product Management (SPM) processes and inquiry of the tasks that are relevant to an Agile SPM process. Also more enhancements are be done in existing implementations of agile SPM processes, and their integration with agile development by using AI techniques. Applying AI techniques on Agile practices to software product management has acknowledged little consideration until now. The significance and integration of both fields also caught attention recently.

In this research, we present a framework which has been introduced for evolution of the agile software development practices by integrating AI technique. We have selected Scrum framework for this reason as it easily manages the change and fill the gaps of communication among developers and customers. This keeps facilitate the product management in agile practices by using AI techniques.

We have used published reports, articles, and existing case studies. It is a conceptual framework that encourages constant communication during the course of the development cycle. This framework has resulted in improving the agile practices by using the concept of CBR. Expert's Review Method is used to accomplish the appraisal of this model. The framework provides guidelines to the agile team regarding improvement of agile practices incorporated with CBR.

In Figure 1, the sub division of the research paper is shown. Section 1 is presents the introduction of the paper, section 2 shows the literature review of the paper, section 3 is the proposed methodology which consists of three parts: preliminary model, abstract level model and the detailed framework. Section 4 presents the research method and the section 5 presents the limitation of our work. Finally, section 4 shows the references (Figure 2).

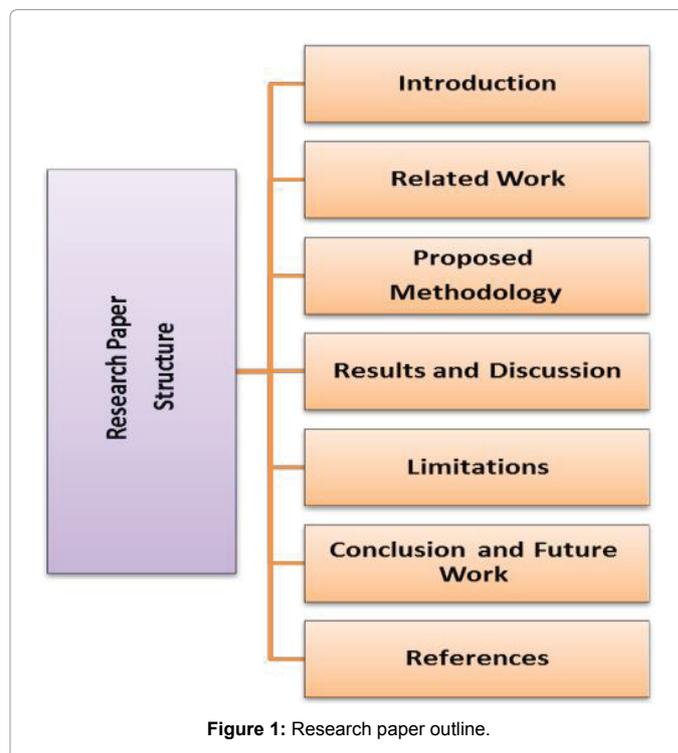


Figure 1: Research paper outline.

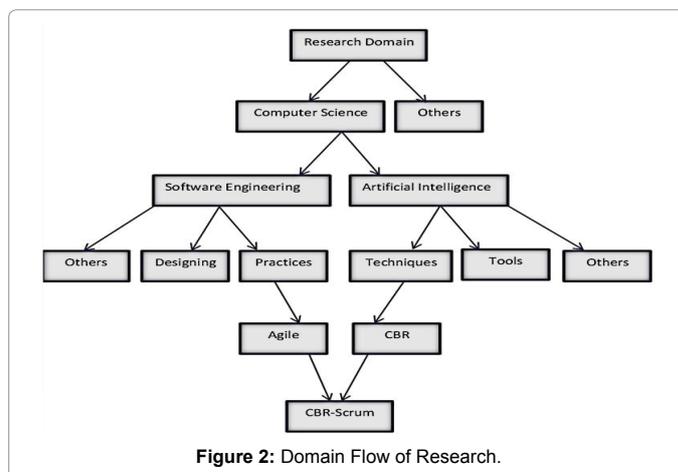


Figure 2: Domain Flow of Research.

## Literature Review

Numerous papers have been written on software development processes and on Agile Software Development process; a few them are discussed below. Abrahamsson et al. [3] states that the Agile methods consist of a software development process that has become popular during the last few years. They play a great role in delivering products faster, with conformate of high quality, and satisfy the high expectations and needs of customer through its flexible principles of the lean production to Software Development [13].

The Agile methods consist of a Software Development process that has become popular during the last few years [13]. They play a great role in delivering products faster, with conformate of high quality, and satisfy the high expectations and needs of customer through its flexible principles of the lean production to software development [14]. Agile Requirement Engineering (ARE) is an emerging area that makes the

Requirement Engineering process more flexible. It provides the benefit of constant communication between customers and developers. As a result the developers can deliver the system in time that surely satisfies the customer's expectations and also increase the business value [15].

Agile Requirement Engineering has proven the success of projects due to different reasons. Firstly, it makes the system transparent i.e., visible to the clients. Secondly, it establishes a proper face-to-face communication that makes requirements more clear and thirdly, team work adds confidence to the work and confirms the success of project [16].

Both processes are based on similar objectives but the main difference between these two: in Traditional Requirement Engineering, documentation is very much important for the system is being developed but it is not so in the case of Agile Requirement engineering because concentrates on face-to-face communication [17].

Scrum and XP have been adopted in software industry for the last many years. Several benefits of Scrum and XP have been explored but few limitations are as under:

- The key emphasis of Scrum is on project management but it remained silent about the software engineer.
- Scrum stresses heavily on skilled professionals to build Scrum team.
- XP's short comings are in project management practices [18].

In the previous studies, the agile approaches have been also considered to be successful in many cases. Companies that have practiced the agile practices i.e., Scrum [19] which varies from small scale companies as mentioned by [6] to large corporations [6]. Studies have shown that by the use of scrum significant benefits have been achieved within the organization [8], which shows that its usage is not limited only to small/local projects [9].

Scrum software development methodologies in small organizations. Project estimation can be challenging due to the customer involvement in the project. This is due to the fact that rapidly changing requirements increase overall cost of the project. Scrum methodology boosts the development time and it also welcomes change at any phase of development. It is challenging for an organization to estimate development time because change request can comes at any stage of the development [20].

Scrum model for distributed project. The most important factor in Scrum is the communication and regular feedback of the customer to the scrum team. The success of the project depends upon proper communication between the product owner and scrum team. For a quality product the distributed project requires continuous unit testing [19].

As day-by-day small scale software development is becoming more popular and this fashion is expected to keep on going and keep on flourishing in future. Long duration projects are mostly targeted by OPEN and RUP, as there is great difficulty in managing a large number of people. On the other hand, XP has some shortages relating to some valuable XP process components [12].

As much of the development has taken separately in both disciplines i.e., AI and SE and very limited research knowledge is exchanged in the form of results and concepts. Nowadays researchers are working to merge both fields and applying the methods of AI to SE and vice versa. The prerequisite before performing interaction is: accurate information, discussion and understanding of factors. In this paper a

framework is presented in which both fields are interacting [21].

The main theme of CBR is based on a theory of reconstructive memory. According to this theory it has been revealed that humans are unable to recall things as they really occurred. Although, "remembering" is a recalling process, characterized by Bartlett as: combination of knowledge contained in specific hints prearranged at the time it occurred, plus (retrieval time) inferences based on information, beliefs, theories and arrogances derived from other sources [22-26].

Survey recommends that the use of AI techniques in SE would be of great advancement and is needed for larger scale assessment of more research. It is required to understand the worth of different approaches. In many studies it is proposed that the use of Case Based Reasoning (CBR) for planning of software development is very useful [21].

The fusion between AI and SE is becoming more common but slowly it is becoming more common. Joint points of both disciplines are emerged from the solicitation of techniques from one discipline to the other. Approaches and practices from both disciplines establish the research respectively [27,28]. Figure 3 shows contrast between traditional and agile methods [4] shows the list of gaps in existing literature.

### Proposed Methodology

In the start the vision document may be imprecise but with the passage of time it will become more specific with the passage of time as the project moves forward. As the most responsible person is Product Owner for getting initial investment, delivering the vision and presenting the customers views and creating the Product Backlog [21]. Through the Sprint Planning Meeting the selected preferable arranged items in the Product Backlog are distributed into small functionalities and retained in the Sprint Backlog. Each item in the Product Backlog is explained by the Product Owner by keeping in view the content, purpose, meaning, and intentions of every item. Questions related to the items in the product Backlog can be asked by the team members

	Conventional Methods	Agile Methods
Characteristics	<ul style="list-style-type: none"> <li>• Extensive planning</li> <li>• Codified process</li> <li>• Rigorous reuse</li> <li>• Heavy documentation</li> <li>• Big design up front</li> </ul>	<ul style="list-style-type: none"> <li>• Iterative and incremental development</li> <li>• Customer collaboration</li> <li>• Frequent delivery</li> <li>• Light and fast development cycle</li> <li>• Tacit knowledge within a team</li> <li>• Light documentation</li> </ul>
Strengths	<ul style="list-style-type: none"> <li>• Straightforward, methodical, and structured nature</li> <li>• Predictability, stability, and high assurance</li> </ul>	<ul style="list-style-type: none"> <li>• Short development cycle</li> <li>• High customer satisfaction</li> <li>• Low bug rate</li> <li>• Quick adaptation to rapidly changing business requirements</li> </ul>
Weaknesses	<ul style="list-style-type: none"> <li>• A slow adaptation to rapidly changing business requirements</li> <li>• A tendency to be over budget</li> <li>• A tendency to be behind schedule</li> <li>• Difficult to create a complete set of requirements up front</li> </ul>	<ul style="list-style-type: none"> <li>• Significant document reduction and heavy dependence on tacit knowledge</li> <li>• Not sufficiently tested for mission/safety-critical projects</li> <li>• Not adequate for highly stable projects</li> <li>• Can be successful only with talented individuals who favor many degrees of freedom</li> <li>• Not appropriate for large-scale projects</li> </ul>

Figure 3: Contrast between traditional and agile methods.

in the Sprint Planning Meeting. All the sub divided modules in the Sprint Backlog are completed through the iteration of the Sprint which comprises the Daily Scrum Meetings [3].

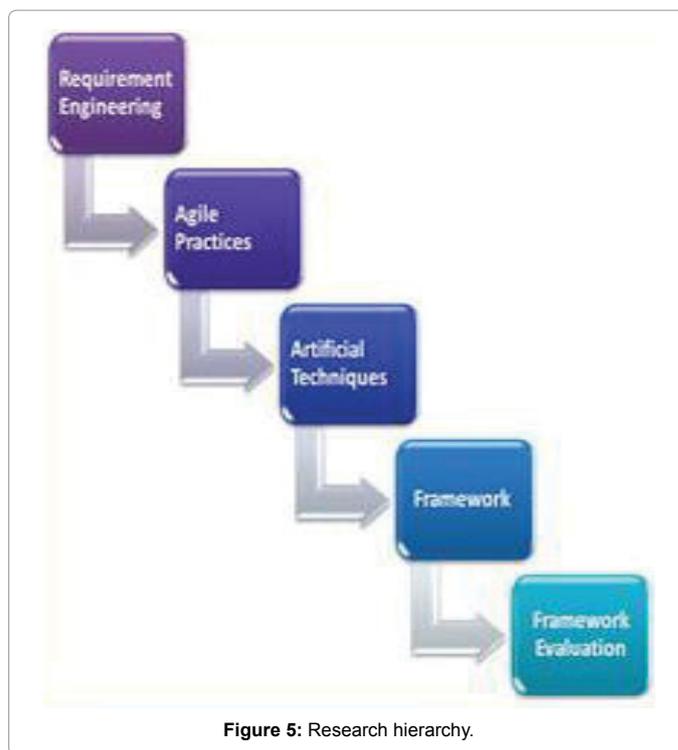
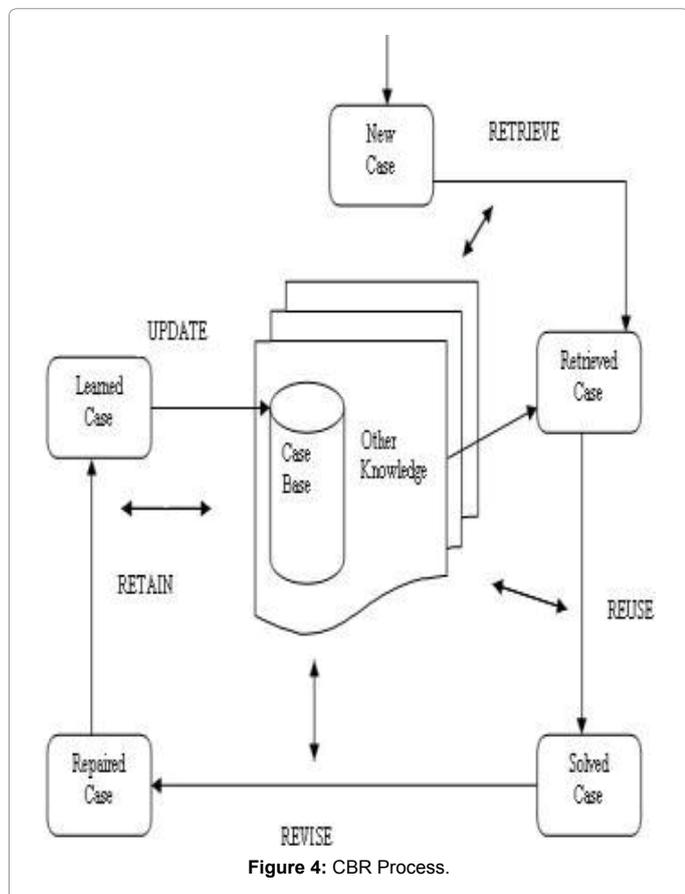
Product owner as the client plays a vital role among scrum and other Agile Practices is responsible for the triumph or failure of the product. Major responsibilities of the product owner include conserving and prioritizing the product backlog which includes identifying and gathering individual user stories and their standards of approval [21].

CBR is an AI technique that motives on memorizing previously experienced cases. The CBR technique is used to appraise the development process by referring to previously stored cases (past practices). CBR is based on four major steps (CBR cycle) shown in Figure 4.

The modification that is presented in this research is to expand the agile software development process by applying Case-Based Reasoning (CBR) which provides the most suitable set of solutions. Within this research study, the CBR is used to evaluate the information in the form of cases (quality attributes and indicators) provided by the user and to give the corresponding quality results with a proposed most suitable solution related to any problems. As in many researches it is suggested that AI techniques should be used to improve the development life cycle. In this paper, a hybrid model which integrates the agile practices with RUP and Scrum has been proposed along with the CBR shown in Figure 5.

### Preliminary model

The proposed hybrid model is presented in this model consists of



three layers: RUP layer, Integrated Scrum layer and CBR layer shown in Figure 6.

### Abstract level framework

The model presented below is the abstract level in which the technical solution is provided for the proof of the concept. Elicited requirements are needed by the team of developers for the development of the software. Resources are wasted at the large scale if the requirements are not properly elicited because this will cause the delay in delivering the software at the exact time. It will also affect the quality of the product as the code is based on the poor specified and elicited requirements. To keep away these kinds of problems there is need of active product managers in every project so that during development it coordinates with the team members so that the requirements are properly elicited according to the needs of the customer [27].

In Agile Software Product Management (SPM) Scrum allows the flow of requirements which is later goes to sprint backlog. Better requirements are defined by the agile SPM as it allows the preliminary procedure for the management requirement lifecycle. Concurrently along with the management of requirements definition other requirements are implemented in agile SPM. A Figure 7 shows the flow of work within the Agile SPM process and it is based on the traditional SCRUM development, described in the previous section, and is appended with SPM-specific adaptations. In the Figure 7 the product sprint backlog is introduced.

### Extended framework

Overall structure of our proposed framework is consisted of 5 phases as shown in Figure 8. Basically these steps show the working. Each of these steps, perform a valid functionality in proposed framework shown in Figure 9. We will discuss each and every step one by one in detail below.

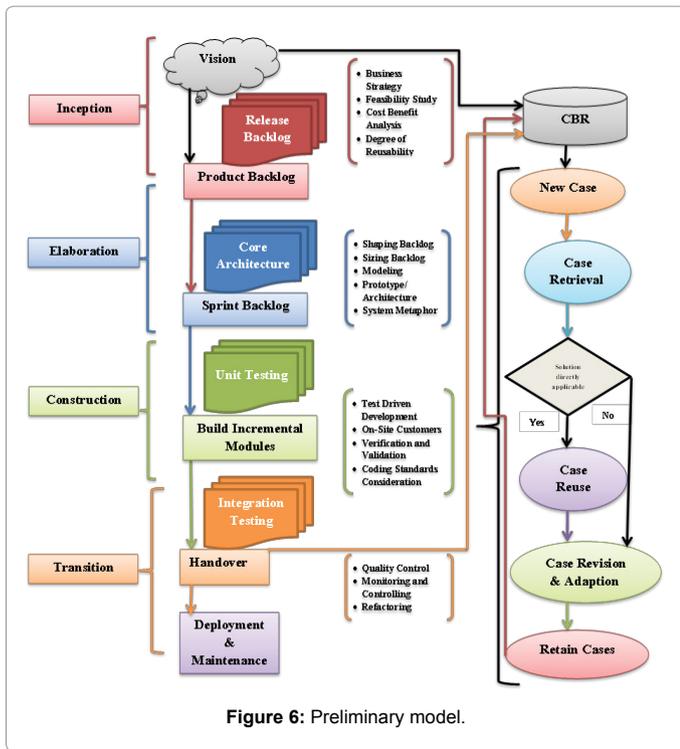


Figure 6: Preliminary model.

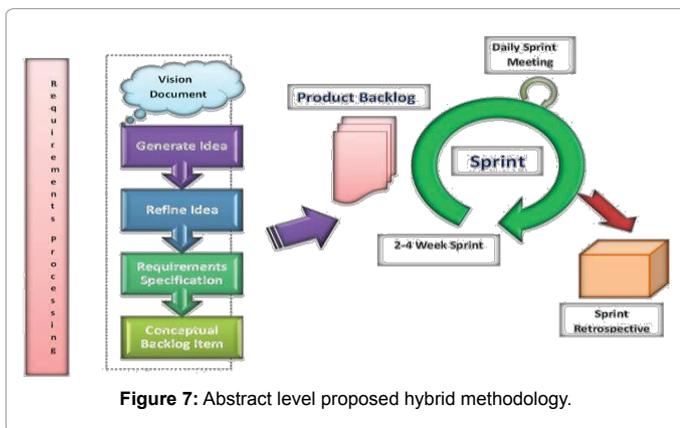


Figure 7: Abstract level proposed hybrid methodology.

- Vision: It is the vision that will guide towards the system.
- User stories: User stories are the description of requirements expressed by customers that consists of sufficient information needed to the developers for the effort estimation and implementation. Product owners elicit the requirements by consulting with the stakeholders in the form of 'user stories'. The requirements are taken on index cards/story cards. All requirements are not gathered at this stage; they are flexible and can be changed at any given stage. After gathering the requirements in the form of user stories these raw requirements enter into the first step that is planning.

**Phase I (Initial Phase):** In the initial phase of framework we started by checking the selected backlog item from a pool of requirements that are in the form of user stories which is done by the product owner. As with the agile point of view Product owner prioritizes the requirements and then estimate the efforts of these prioritized requirements. This leads towards the tracing the business issues which will further moved

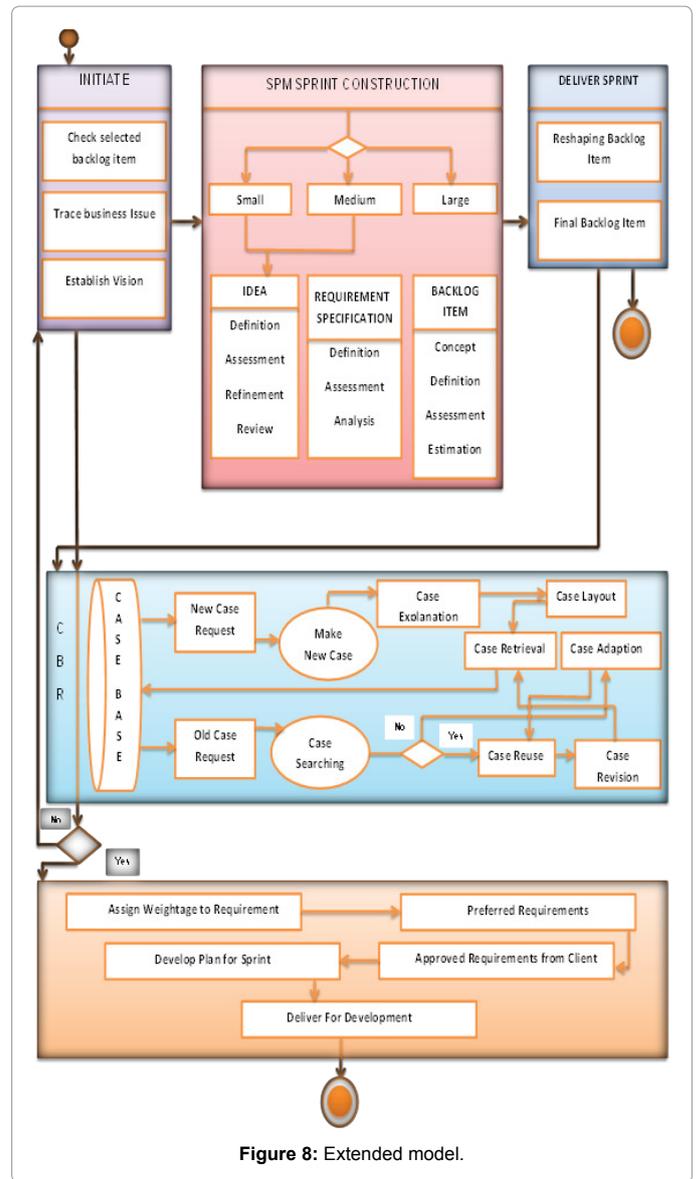


Figure 8: Extended model.

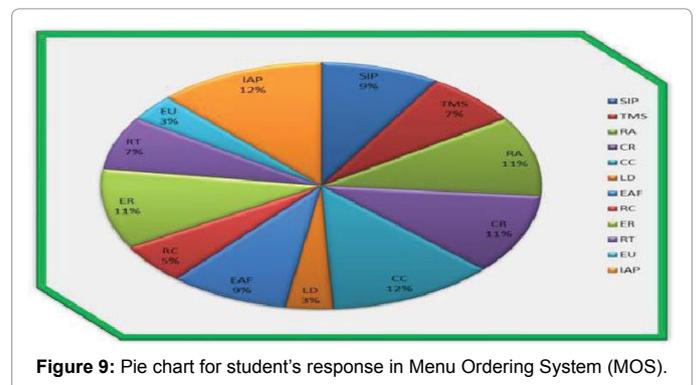


Figure 9: Pie chart for student's response in Menu Ordering System (MOS).

forward establishing the vision document. The input of this phase is the raw requirements and output is the vision document. In the initial phase Product owner engages the stakeholders in every step so that the requirements and their selection for the construction of vision.

**Phase II (SPM sprint construction):** In this phase which is the software product management (SPM) construction, vision will go through different stages, during which it is polished. The stages are:

- **Vision:** A vision is the starting point for the lifecycle scrum. It is an impression, which is fetched up by a customer or any other stakeholder or by any person from the company, and it is in well-defined because they are explained in basic terms. After that the vision comes to a product manager, which is then converted into a (set of) idea.
- **Idea:** An idea is the formal plan for action which is the expansion and description of vision in more detail. The product manager describes the definition means that what is the purpose of the new functionality, assessment is done in terms of the business value of the idea, and the involved stakeholders. It provides the clarity about the idea. An idea should be concise description of the problems related to the business, its point of origin and the issues related to the scope of an idea.
- **Requirement specification:** In it, requirements are approved to moved further towards the next section in which they are designed, coded and tested by a cross functional team. The approved requirements are designed by drawing their use case diagram to make them more specific and understandable for programmer as well as by the customer. The definition of requirements is accomplished in different steps. Software product management (SPM) team converts an idea into a list of requirement definitions and the details are omitted. Requirement definitions contain the depiction, a logic and limitation related to the specific criterion that occurs during the implementation of the requirements. The cross checking is done by architects (use case diagram) or by lead developers to check the probability and compatibility with other requirements. After that the requirement definitions are re-checked by the software development team and the detail is provided by them and assessment is done either these requirements are in the scope which means that they are practicable, reliable and comprehensible or not.
- **Backlog item:** The Backlog maintains all the items that are required to be completed within the sprint by each product manager. The backlog decomposed items into concepts that form into product backlog (PB), it contains the complete list of concepts, ideas and vital requirements list for a system. The concept in the form of PB items definition is provided and further assessment and the requirement estimation is calculated before they can enter the sprint backlog (SB).

**Phase III (Deliver sprint):** At the start of each sprint, each SPM team has to manage product and to formulate sprint backlog in the form of product management sprint backlog. After that the team members selected the vital PB items at first hand, those are considered to be completed in the first approaching sprint. Similar to the traditional sprint preparation which is achieved by the development team members. Then proceeds towards the next step, which is to refine and re-shape the sprint backlog items or announcing new designs which come through support from the customer end, by holding the meetings with business experts, from industry specialists and by the participation of the active market dealers from different types of forums. In the course of a sprint, refinement is done on every sprint item from its existing stage to the next aspect level, i.e., from vision to idea or from concept to requirement specification definition.

**Phase IV (CBR phase):** In this phase, Case based reasoning (CBR)

offers solutions to the upcoming problems by reusing and adapting the previous solutions those were used to solve old alike problems. According to the CBR cycle as an input it takes new problem detailed description. On the basis of provided problem descriptions, CBR looks into the case repository, where the old solutions are kept, for the past related cases. By recalling the previous old cases solution is re-applied or adapted if it is suitable to the new scenario. After that the new attained solution is retained to the CBR repository if it matches so the case is reused then revision on the previous case is done according to the new situation then it is retrieved. If the retrieval is unproductive then it will move back to phase one and after then when solution is suggested it will be saved in CBR in the form of new case.

**Phase V (Requirement phase):** In the last phase the preferred requirements from the customer which are placed in the sprint backlog are developed one by one by the development team members. The main objective of this phase is to verify the product that it meets the specified requirements and assess the response of customers. Product owner, scrum master, team and stakeholders gather to test the deployed software. In sprint retrospective, team demonstrates the performance of working software. Preliminary testing is done by product owner and stockholders. If the product meets its specified requirements and raise the level of satisfaction than the product will deliver to the stockholders otherwise pointed issues resolved in the planning phase in the next cycle.

## Results and Discussion

This model is evaluated by many experts from industry in the form of questionnaires and by using case study. As mentioned in the limitations later we will provide detailed evaluation of this model. Now this model is evaluated against 12 success factors of RE mentioned below. Case study provides an empirical inquiry that investigates a phenomenon within its real-life context. In case studies we have made statistical analysis to evaluate the proposed integration of agile practices with AI technique. Then we took some success factors and evaluated the results on the basis of expert opinions. On the basis of these cumulative results it is clear that scrum integrated framework is more flexible than traditional scrum process which increases our form of work worth and proved to be a big step ahead. The main theme of our framework is to find out the applicability in practical life and in requirement engineering process. All of this process is to minimize the human effort.

In this case study, there are two project groups, each having three group members and one supervisor. In this case study they have worked with the proposed framework. Table 1 shows their opinions in the form of yes and no. MOS was developed for an academic research project at university by a team consisting of a faculty supervisor, three graduate students. Management was very ad hoc, although weekly research meetings were well attended and served as motivation for continued progress for each team member. Tasks tended to be assigned by matching project needs with the individual capabilities of team member.

### Case study controlled experiment from academia

As a very first step, we need to identify the validation criteria for our results. In order to validate the results; we have selected 12 requirement engineering success factors, those are: Small Interval Project (SIP), Team Members Skill (TMS), Risk Analysis (RA), Customer Relationship (CR), Constant Communication (CC), Less Documentation (LD), Efficiency and Flexibility (EAF), Reduce Project

Student Response		SIP	TMS	RA	CR	CC	LD	EAF	RC	ER	RT	EU	IAP
Integrated Framework	Yes	5	4	6	6	7	2	5	3	6	4	2	7
	No	3	4	2	2	1	6	3	5	2	4	6	1

Table 1: Students Response in Menu Ordering System (MOS).

Cost (RC), Enhance Reusability (ER), Reduce Project Time (RT), Ease of Usability (EU), Improved Agile Practices (IAP).

### Menu ordering system

This part describes an example of a Menu Ordering System that is taken to explain the idea proposed in this thesis that requirement change management can be improved by using Case based reasoning technique. The Menu Ordering System (MOS) is a system to replace manual and telephone process to order lunches by the employees of organization to which the Cafeteria belongs. Registered persons who are given Login ID can order meals online on this system. Payment of lunch will be deducted from their salary. Menu manager will manage the system, cafeteria staff will prepare the meals and deliverers will deliver the meals on the location as desired by the registered user. Few functional requirements of this system (MOS) are:

- **Placing order by registered user:** A registered user (Patron) can place an order for one or more meals.
- **Registration confirm by system:** The MOS confirms that a logged in user is registered Patron to place order for meals and for payroll deduction of meal's payment.
- **Placing order by unregistered User:** If an unregistered user asks for placing an order, which is the employee of the organization, MOS gives option for registration or to pick up the meal from the Cafeteria instead of delivering meal on the location of user. Else COS asks the user to exit from system.
- **Menu display:** MOS shall display the menu of that day.
- **Items available-menu display:** MOS shall display the food items available along with their prices and current date on the menu.
- **Units of Food items:** Patron will indicate in the order, the number of food items he or she wishes to have.
- **Order Limit:** MOS will mention in the menu the maximum number of units of all food items available to order.
- **Order Confirm:** MOS asks for the confirmation of order from the Patron.

The graphical representation of the results is shown in the Figure 8. In this Figure 9, No. of students satisfied in the vertical axis shows the positive responses of the experts and horizontal axis shows the success factors. Experts have given the opinions on the basis of success factors in the form of "YES" and "NO". On the basis of expert opinions we have made a percentage pie graphs that shows the percentages of success factors shown in the Figure 8. In the Figure 10 the no. of experts satisfied against the success factors is shown. From the results that we gathered from the expert's response are satisfying because the majority of the response is positive to the maximum no. of success factors. So in the next case study expert review as well as the control experiment is performed to get more reliable results against over framework.

Case study provides an empirical inquiry that investigates a phenomenon within its real-life context. In case studies we have made statistical analysis to evaluate the proposed integration of agile practices with AI technique. Then we took some success factors and

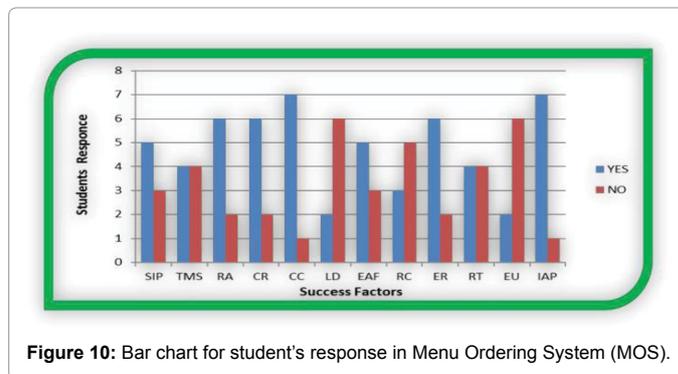


Figure 10: Bar chart for student's response in Menu Ordering System (MOS).

evaluated the results on the basis of expert opinions. On the basis of these cumulative results it is clear that scrum integrated framework is more flexible than traditional scrum process which increases our form of work worth and proved to be a big step ahead. The main theme of our framework is to find out the applicability in practical life and in requirement engineering process. All of this process is to minimize the human effort.

### Questionnaire evaluation

The graphical representation of the results is shown in the Figures 11-13. From the results that we gathered from the no. of responses that are satisfied against the success factors. As the majority of the response is positive towards the maximum number of success factors.

### Limitations

The main limitations of our study are the single-case use, small sample size of experts and the possibility of preconception in data collection and analysis from questionnaire. The fact that we used a single-case holistic design makes us more susceptible to bias and eliminates the possibility of direct replication or the analysis of contrasting situations. Therefore, the general criticisms about single-case studies, such as uniqueness and special access to key informants, may also apply to our study. Our goal was not to provide statistical generalizations about a population on the basis of data collected from a sample of that population. Another limitation is that a part of our evaluation is based on semi-structured questionnaire. The practical evaluation from industry is also lacks behind for the possible evidence of our framework.

### Conclusion and Future Work

This study provides a primary overall integrating model that illustrates the fundamental concept of coordination between AI techniques and agile software development projects. It has to be mentioned that the model presented in this paper is still improved in on-going research studies and is still subject to further refinement. In this thesis an integrated framework an Agile practice with AI technique is proposed. The major area of contribution in this framework is to expand and enhance the agile software development life cycle. This framework is more feasible for the projects in which requirements and its solution are reused throughout the development cycle. It has the

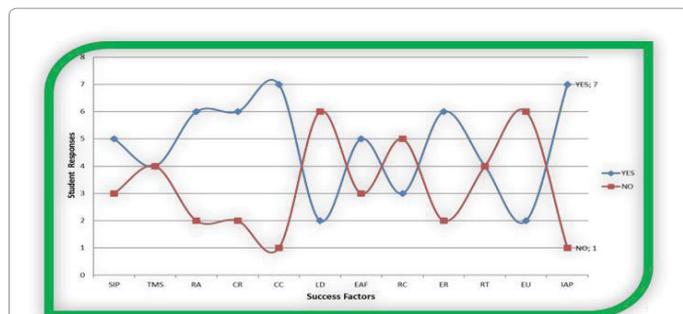


Figure 11: Response against success factors.

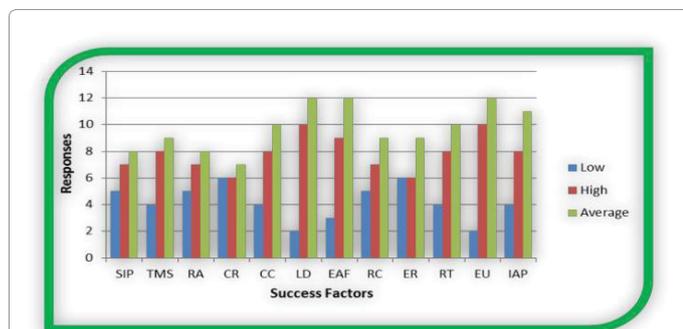


Figure 12: Bar chart graphical representation of responses.

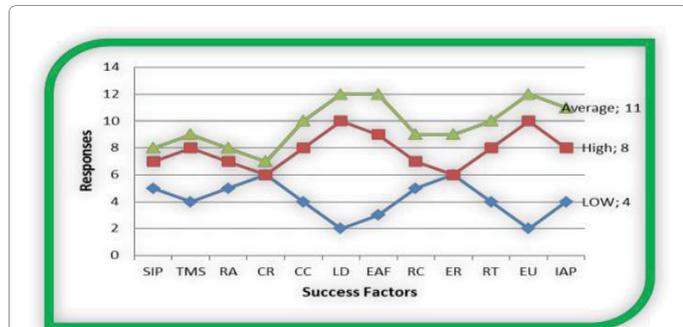


Figure 13: Graphical representation of responses.

capability to deal efficiently with almost every kind and size of project i.e., small, medium and large size projects. Main contributions are:

- This will help the developers and stakeholders to have clear vision of scenarios and views of user requirements.
- This will stick the developers and customers throughout the development cycle and this will increase the confidence of customers.
- Stakeholders and specially users can get clear pictures of what kind of product these requirements will form, so they can change at any stage.
- This will focus more on people and communication against process and documentation.

Future research will focus on more specific to hybrid models to obtain in-depth understanding and provide complete framework. Furthermore, a survey may be conducted to elicit critical information

about the ways in which industry tailors software practices and methodologies, and the compatibility and effectiveness of hybrid software methodologies. In addition to this, AI techniques along with the intercommunity between, classical SE and agile methodologies can be valuable research direction.

## References

1. Zave P (1997) Classification of Research Efforts in Requirements Engineering. ACM Computing Surveys 29: 315-321.
2. Pressman RS (2001) Software Engineering. McGraw Hill, USA.
3. Cho J (2008) Issues and Challenges of Agile Software Development with Scrum. Issues in Information Systems 9: 188-195.
4. Cho J (2009) A Hybrid Software Development Method For Large-Scale Projects: Rational Unified Process With Scrum. Issues in Information Systems 10: 340-348.
5. Rachel H (2012) Report from the first international workshop on realizing artificial intelligence synergies in software engineering. ACM SIGSOFT Software Engineering Notes 35: 34-35.
6. Beck K, Beedle M, van Bennekum A, Cockburn A, Cunningham W, et al. (2001) Manifesto for Agile Software Development. United States of America.
7. Cockburn A (2002) Agile software development. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA.
8. Pressman RS (2010) Software Engineering. McGraw Hill, USA.
9. Jiang L, Eberlein A (2008) Towards a framework for understanding the relationships between classical software engineering and agile methodologies. International Conference on Software Engineering, Leipzig, Germany.
10. Nuseibeh B, Easterbrook S (2000) Requirements Engineering: A Roadmap. ICSE '00 Proceedings of the Conference on the Future of Software Engineering, Limerick, NY, USA.
11. Prince J (2011) Interaction between Software Engineering and Artificial Intelligence - A Review. International Journal on Computer Science and Engineering 3: 3774-3779.
12. Al-masum SM, Morshed AM, Mitsuru I (2010) Object Oriented Hybrid Software Engineering Process (SEP) model for Small Scale Software Development Firms.
13. Abrahamsson P, Warsta J, Siponen MT (2003) New Directions on Agile Methods: A Comparative Analysis. Proceedings of the 25th International Conference on Software Engineering, Portland, OR, USA.
14. Poppendieck T, Poppendieck M (2003) Lean Software Development: An Agile Toolkit. Addison-Wesley, Boston, Massachusetts, United States.
15. Jun L, Qiuzhen W, Lin G (2010) Application of Agile Requirement Engineering in Modest-sized Information Systems Development. Second WRI World Congress on Software Engineering, Hubei, Wuhan, China.
16. Paetsch F, Eberlein A, Maurer F (2003) Requirements Engineering and Agile Software Development. Twelfth IEEE International workshop, IEEE Computer Society Washington, DC, USA.
17. Zaigham M, Qureshi MRJ (2012) Novel Hybrid Model: Integrating Scrum and XP. International Journal of Information Technology and Computer Science 6: 39-44.
18. Marchesi M, Mannaro K, Uras S, Locci M (2007) Distributed Scrum in Research Project Management, Agile Processes in Software Engineering and Extreme Programming, 8th International Conference, Italy.
19. Dingsøyr T, Hanssen G, Dyba T, Anker G, Nygaard J (2006) Developing software with scrum in a small cross-organizational project. European Conference on Software Process Improvement, 13th European Conference, Finland.
20. Meziane F, Vadera S (2012) Artificial Intelligence in Software Engineering: Current Developments and Future Prospects. IGI Global Disseminator of Knowledge.
21. <http://www.controlchaos.com/>
22. Rech J, Althoff KD (2004) Artificial intelligence (AI) and software Engineering (SE): Status and future trends. KI 18: 5-11.

23. Jeff S, Anton V, Jack B, Nikolai N (2015) Distributed Scrum: Agile Project Management with Outsourced Development Teams. 40th Annual Hawaii International Conference, Waikoloa, HI, USA.
24. Shi Y, Wang H, Tang J (2011) A case-based reasoning system for fault management in CDMA network. In Control and Decision Conference, China.
25. Ammar H, Abdelmoez W, Hamdi MS (2012) Software Engineering Using Artificial Intelligence Techniques: Current State and Open Problems. Institute of Communication, Culture, Information and Technology, University of Toronto Mississauga.
26. Batool A, Hafeez YM, Hamid B (2013) Comparative Study of Traditional Requirement Engineering and Agile Requirement Engineering. 15th International Conference on Advanced Communications Technology, Sungnam Kyunggi-Do, Korea.
27. Vlaanderen K, Jansen S, Brinkkemper S, Jaspers E (2011) The agile requirements refinery: Applying SCRUM principles management. *Information Technology* 53: 58-70.
28. Hayata T, Han J (2009) A hybrid model for IT project with Scrum. *IEEE International Conference on Service Operations, Logistics, and Informatics (SOLI)*.