HYBRID OF AGILE PROCESS AND USABILITY EVALUATION METHOD

Saad Masood Butt
Department of Computer and Information Sciences
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia
saadmasoodbutt668@yahoo.com

Wan Fatimah Wan Ahmad
Department of Computer and Information Sciences
UniversitiTeknologi PETRONAS
Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia
fatimhd@petronas.com.my

ABSTRACT:*Agile software development is one of the most widely used methodologies for software development. It plans to improve software quality and responsiveness to modifying client requirements. Agile software development is based on an iterative cycle with the goal of faster distribution of solutions with on-going user participation. To improve the market value of software, usability evaluation techniques in software model helps to gain user satisfaction and increase product market value. Producing rapid solutions by agile software methods leads to the deficiency of good design and architecture is regarded to be very expensive if followed. As a remedy, a proposed life cycle for agile software development has been designed. The proposed life cycle outlined in this paper integrate usability evaluation concepts and agile software methodologies for the development of interactive software. To achieve the result, a survey form was designed and conducted worldwide. The results show a greater interest of usability at the early stage of development along with the user participation at every stage. To evaluate the proposed life cycle and IEEE Std 12207-2008, ISO 9241:210 was used to validate the proposed software model.*

Key words: Software Development, Usability, Agile Software Models

1. INTRODUCTION

In the designing of the software interface, experts of the SE and HCI need to understand the user's behavior, user's familiarity with different features of a software interface and user's expertise while working with other software interfaces. The HCI deals with social, cognitive and interaction phenomena. Where the social layer is focused on how people interact with each other as well as with technology based on the surroundings. Therefore a high level of usability is identified as an important function of the software products. Those products which are having poor and inefficient design are common causes, among other things, results in the failure of a software product. Thus, one of the complications involved in software development is to observe users and analyse its participation in the design and development stages, their behavior and effective, then to collect information idea for the ensuing development.

It is quite apparent that the development of better systems needs the collaboration of different professionals like Human Computer Interaction (HCI), Software Engineering (SE), Stakeholder, Usability experts and User experience experts etc. Software Engineers, HCI experts and users are the most important key players of the software development. Involving users in the software development is an important thought-provoking task. Neglecting HCI approaches from software development will affect the role of usability in the software and make the software difficult to learn and to use which results in the dissatisfaction of the users. Software engineers, HCI experts and users need to cooperate with each other to create a software product that is usable and useful for the target audience. Unfortunately, in reality, these three key players (Software Engineers, HCI experts and Users) do not cooperate as smoothly as they should.

By the time this lack of cooperation is released, big software projects may fail to deliver what they promised; therefore, such projects often fall short [1]. Their failure can be in various forms; such as (a) the delivered system isn't able to offer considerable specifications to the conventional system which is top quality that would make it possible to actually set up the system,(b) involvement of the user is introduced too late which produces a high impact on the software efficiency , (c) the cost, time and resources are exceeded by large factors and (d) the product is difficult to understand due to the lack of usability role in the development. The agreed upon factors for which software projects fail are many and various [2]. In this paper, a new agile software process is proposed where usability evaluation has been integrated intoan agile software development method.

2. Literature Review

Many of the standard development methodologies are based on technical dimensions. But, they fall apart when trying to meet up with the project strategies. Therefore, the researchers have come up with a model that is standardized by ISO/IEC12207. It uses balanced score card to fulfil missing dimensions in the project strategies. Agile software development model is mapped with ISO/IEC12207 using balanced score card to create the appropriate action plan. A model [1] is developed that helps project managers to

evaluate effectiveness of the Agile Software Development model using balanced score card.

Balanced score card would ensure that the goals and objectives of the project are being met. Balanced score card aligns business strategies with the action plans. It helps project managers to align his/her decisions with the business strategies from the score card aligning people and activities to it.

ISO/IEC12207 is a standard for software life cycle processes for systems and software engineering to use its elements as an established set of Life Cycle Processes. It establishes conformance of the project to the established environment. It contains processes, tasks and activities that would be applied during the acquisition, supply, development, operations, maintenance and disposal of software.

Agile Software Development method was developed by Agile Manifesto in 2001. It is based on customer's collaboration sharing requirements and values to be incorporated in the product. Highest priority is given to customer satisfaction. The scrum is adapted by agile software development methodology. It introduces iterations called sprints. Each sprint refers to a short term plan in which developers should meet on daily basis for discussions. Tasks are backlogged into scrum and each sprint is allocated a task to be done within a time frame.

So, the agile software development is mapped on to the ISO/IEC12207:2008. It applies 27 processes to achieve the set targets. These processes are further mapped on to the balanced score card using Analytical Hierarchy Process (AHP). Hierarchy of decisions is developed from the measures of related processes. Firstly, software response time is measured, then decision criteria is formed that is mapped on to the goal achievement processes. And, other processes are prioritized accordingly.

Large scale software development companies employ waterfall model that is plan driven and takes long time to complete. This approach is susceptible to failures in rapidly changing environments. Thus, another approach is introduced that understands the dynamics of software development projects known as Agile Software Development Life Cycle. 'Agile genome' defines seven characteristics of agile projects. Systems Dynamics Model is constructed for agile software projects called Agile Project Dynamics (APD) model [2] that takes care of all seven aspects as a major component of the model. Many of the commercial softwares used agile methodology to deal with the pressures that could occur in traditional development in terms of requirements changes, schedule delays, defects that result in endless delays and redesign.

Software development team when adopts agile development actually they practice various agile methods. It reduces coordination cost and focus teams to come up with reliable product iterations which are released in increments. Agile methodology has marked its presence in the commercial world and now moving towards aerospace and defence areas. Here the APD model is compared with waterfall model integrating its various methods with the management policies to come up with the best project performance. It is common practice that software development processes integrate with product development processes that include business planning and requirements analysis (figure 2). Here the researcher was only concerned with software development so his team used SCRUM but the other stakeholders that are System Integrators (SIs), System Engineers, and testing team still operated in the traditional manner. System Engineers handed the requirements to development team and in turn they passed the code to SIs.Here the software development team was under supervision of system engineers and software quality assurance team so they had to produce same designs as in the requirements design documentation, and other specifications even if they were using SCRUM. Yet SCRUM energized the development process and speed up the experience gained by new engineers.The research further dived into agile processes and understood it by integrating with CMMI level 5 software engineering environments.In turn the APD model produced is complex using Vensim PLE system dynamics model. It is developed in several views that helped in formation of different subcomponents in isolation and linked them using shadow variables. Agile Rework Cycle forms the core of this model. It has an iterative-incremental development style in a waterfall approach.

Agile development methodologies are getting wide acceptance as they address many Software development risks. Faster delivery of software is made possible and it is flexible towards changes introduced in the software with time. Organizations adopting agile are inclined towards adding features that increase user interest in the system in terms of value and usability [3]. Usability engineering explores Human Computer Interaction (HCI) focusing on how people interact with the systems. But, it was difficult to integrate user interactive process in traditional agile methodologies as used in practice. This lead to the evolution of eXtreme Scenario-based Design (XSBD) [3] process that integrated agile usability approach. XSBD maps well on the established Scenario-Based Design (SBD) process already part of usability engineering fundamentals and is also in compliance with the agile development model using XP and Scrum. XSBD keeps large softwares on track by ensuring quality by system usability measure. Central Design Record (CDR) forms the core of XSBD, which provides sharing of design that guides usability process. Thus, usability and agile development work practices closely coordinate and communicate in XSBD. The usability evaluation results are coupled with the design and high level project goals adhering to the key benefits of SBD and links to the agile work process. XSBD has been developed and tested by partnering with several Software Development Companies and results gathered through practitioners who used XSBD in their development process. The results of this research demonstrate a broad scope of continued research in adopting it in practice and linking it with HCI methodologies and reusability of knowledge gained.

Here software development and usability design runs in parallel. Personals in usability and development team closely collaborate to come up with the quality system

keeping to the quality standards with a focus on increased user interactivity. CDR tightly couples evaluation results generated through usability development process with the design features and goals of large systems making it possible for the usability engineer to embed key benefits of SBD while remaining within agile incremental development cycle. With partnering with companies like; Meridium, Inc. etc. helped refining XSBD approach with actual implementation of it in practice. In the analysis, the different divergent aspects were addressed between usability and agile methodology. Based on the case studies and analysis, the principles were formed for the practitioners who would follow agile usability approach.

The challenges that are present in XSBD are:

1. Comparison of core principles – it compares the convergence and divergence points where the SBD is incorporated in agile practice.
2. Balance of Power – when instantiating the combination of two approaches that are usability and agile working process there needs to be a balance in both approaches running in parallel.
3. Checks and balances – to deal with divergence existing in implementing two approaches in parallel the tradeoffs are considered. How these tradeoffs be made? There can be a tradeoff that has to sacrifice development speed to validate usability characteristics of the system. The integrated approach should be balanced enough to help personals in these tradeoffs.
4. Synchronicity – this establishes a measure to keep usability and agile methodology in practice synchronized.

It has remained a challenging task to develop an engineering model that meets time and scope within an iterative cyclical design process. The new interface is designed keeping in view the user requirements – what it would be used for and who the users would be? The requirements specify the tasks that would be performed by users in a sequence. These tasks form the basis for early interface design process. Paper prototypes of the interface get developed then these prototypes are converted to early interface designs with limited functionality. With time and usability testing the final and fully operational interface is created [4]. Such tools are thus developed that help early phases of design process. High emphasis is given on cognitive approach to design that map on to the rapid, cyclical agile process.

The rapid, X-treme programming methodology and usage centered design forms the foundation of agile design process. The process is described as 'adaptive' rather than 'predictive' and is more 'user focused' than process oriented. The planning needs to follow agile methodology and higher interaction is required of design team members with a focus on HCI principles. These design principles and interactive collaboration makes the agile process iterative.

The agile design process consists of several steps that span from 1 month to 6 months period [4]:

-Stating the mission: defining the purpose of the system to be built

-Domain model: describing the domain with respect to organizational structure, customers, users and locations

-Role model: defining the actors that would be present in the system

Essential stories: narrating the system usage

-Task descriptions: setting tasks and goals remaining technology independent

-Activity diagrams: description of how users would interact with the system

-Test case design: measures adopted to test the system

-Content model: narrations that best describe the requirements of each task

-Wire frame/Canonical prototype: format of HCI and content design

Prototype HCI design: designing layout, content, and functionality irrespective to widget design

-Usability evaluation plan: various prompts, questions and storyboards to test

-Prototype: iterations of HCI working as the system is built

-Task flow diagram: user goes through a walkthrough to test HCI

-Usability evaluation and report: at each development phase in a spiral results would be generated

-Incremental feature map: each spiral would be responsible for time-based features

Usability testing does not require long time span or a higher budget to be more effective. 'Discount usability' allows engineers work in team by thinking aloud, card sorting, scenario-based, walkthroughs and using heuristic approach – making the process much cheaper, fast and easy. These techniques can be applied early in the life cycle and during implementation phase for evaluating major/minor usability issues. In this framework, discount usability model is used within agile setting to be iterative and be more effective [5]. The software not only has to be useful but it also needs to be usable these days. Agile development model follow an iterative approach and has a very strict time frame where daily scrum meeting is held to update the team with the happenings. Researchers came up with possibility of merging usability methods with the agile model.

Adopting an agile approach while focusing on usability centered design, lead to an awesome experience that resulted in timely delivery of a highly usable product. The motivation for using agile with usability measures was because agile environment had a tested procedure at any point in the development phase. It was made for effective by inviting the User Experience Team to play as a customer to evaluate the usability of the product.

The cycles shown in figure 5, could use any technique to gather user data like; scenario-based, or walkthroughs, etc. at the end of each cycle a working deliverable is expected featuring customer's expectations.

The success of discount usability model is that it is [5]:

1. Easy to use, teach and comprehend. In just a half hour meeting the heuristics of usability techniques could be laid down to reveal issues that could be present in the product at hand.

2. The discount usability model is very cheap to adopt as no expensive tools or equipment is required.
3. No usability experts need to be hired to perform evaluation. Evaluating is a very flexible process in discount usability.
4. Using techniques like; card sorting can get early feedback in the design process before reaching to a working system.

Discount usability also as its limitations that should not be ignored [5]:

1. Over simplification of this method adds some distortion to it that is introduced in it while making it easy, and fast. When making discount usability out of traditional usability approach, only the key principles were adopted out of thousands of entries. Thus, it became more generalized and could confuse developers.
2. Although it is understood that evaluation process does not need to recruit usability experts or end-users making the procedure flexible, there is an opinion that this could lead to misinterpreted changes that are actually not required in real. And, that would degrade the usability of the system.

3. Current Gaps in Software Models

At present various software projects are being developed and still developing using different software models. However, there are some loopholes which persist in the software developmentsuch as documentation, total cost of the project, time to complete the project, resources required for the project and usability of the software. Research and market report shows 70 to 80% effort is done for the development of feasibility report and SRS (software requirements specification) document of the software project. The requirement of the software keeps on changing with time which gives great impact on the development of software [13].

Hedbergdiscussed about the Integration of HCI specialist in Open Source Software (OSS) Development[6]. Typical OSS development projects are organized around developers whose interaction is based on specialized technicalaspects and source code. It seems very difficult to communicate with end users who have no technical knowledge.Hedberg proposed a model (as shown in figure 1) that intergrade HCI in OSS and makes the existence of HCI professionals noticeable in the projects, and encourages connections between designers and the HCI specialists in the course of a project.

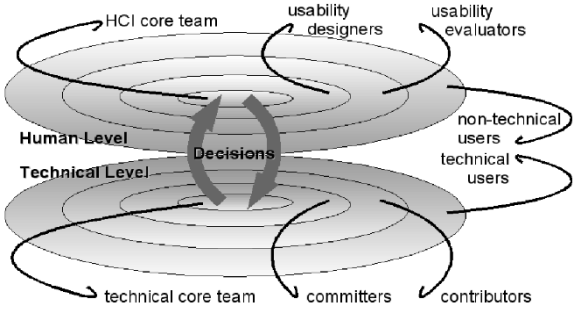

Figure 1: The proposed model with the technical level and the human level roles

In 2007, Memmel et al. from Human-Computer Interaction Lab Germany proposed a software development life cycle in his paper "Agile Human Centered Software Engineering" later published by British Computer Society [7]. The authors of this paper confess that they did not implement this development lifecycle practically. But the proposed lifecycle is based on the facts of industry and research experience. His proposed lifecycle CRUISER as shown in figure 2 helps to bridge HCI and SE based on common features of both fields. CRUISER is very close to XP but don't have agile aspects. The author tried to integrate the important discipline in one lifecycle. Increase the involvement of user and stakeholder by using prototype and scenarios.

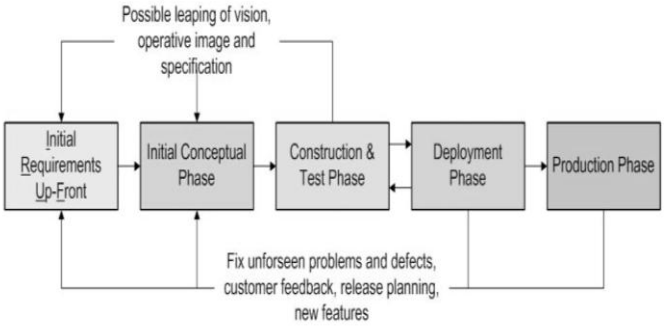

Figure 2: Phases of CRUISER

Pardha discussed in [17] about the proposed model that provides a development infrastructure which accommodateUsability Engineering and SDLC. But the proposed model have some potential downfall like resource overhead, need expert for documentation entry into the design representation model.

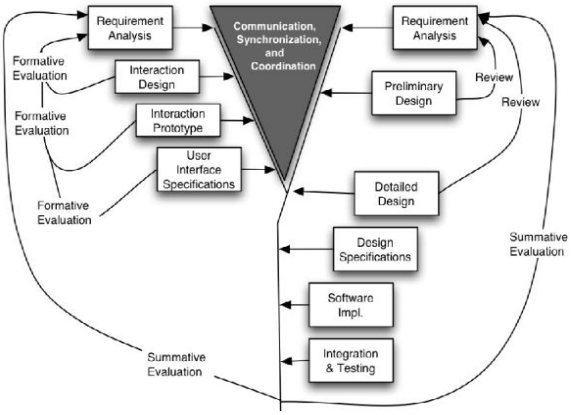

Figure 3: Proposed Model-Based Framework for Integrating Usability and Software Engineering Life Cycles

4. Proposed Life Cycle

Figure 4shows the agile life cycle starts from Flex REQ [8] and ends at a final product. Passing through various processes helps agile software experts, usability experts to work together. Flex REQ [8] is a process to develop product feasibility documents in a small amount of time unlike traditional soft model spends a considerable amount of time in documentation to achieve product quality at the end [11].

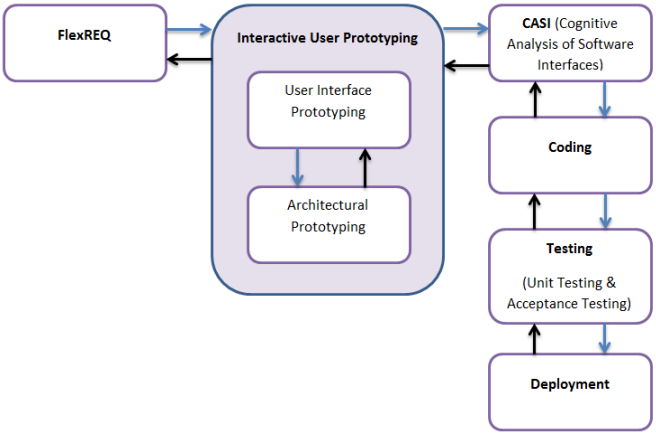

Figure 4: Agile Usability Software Engineering Lifecycle

Interactive User Prototyping consists of two further process user interface prototyping and architectural prototyping as shown in Figure 2. The final requirements gathered from the Flex REQ phase are now in the phase of designing (IUP) where interface and architecture prototypes are refined according to the specified requirement. The resulted prototypes from the IUP phase will further tested for the Usability by using CASI [9,14]. CASI is a Usability evaluation method helps to improve the usability of software interfaces. Involvement of user and usability expert is highly important in this phase to find the usability defects. CASI keeps on the evaluating the interface until user fully satisfied. After CASI phase the coding starts and later unit and acceptance testing will be conducted to check final product satisfies all specifications and useful for the customer [15,17].

5. Survey

For this analysis, a survey has been conducted 45 randomly selected IT professionals from the Information Technology domain have participated in the survey. The purpose of the study is to create an "Agile Usability Software Engineering Life Cycle" that could comprehend the influence of the Users in the software development process. The purpose is to make software development process reliable and finally integrate the Usability Evaluation to make the software more usable. The distribution of the survey targeted IT experts, researchers, software users and stakeholders. The questionnaire is divided into four sections. Section A is about demographic information of all those people who will answer the survey questions.

Section B is on the software process particularly focusing agile process in software industry. Section C looks into the Usability Evaluation in the agile software process. Section D is focused on developing a process that able to do the things that normally require human intelligence to perform that task in software development. All questions mentioned in every section and was rated using the scale of 1 to 5 (1= strongly disagree, 2= disagree, 3= fair, 4= agree, and 5= strongly agree).

Figure 5 shows 81% respondents agreed on the active participation of users in software development. Also software interfaces play magnificent role in product success and failure but also prefer less documentation in software development process. Rest 5% disagree with the active participation of users in software development, software interfaces plays role in product quality and also disagree on less document in software development.

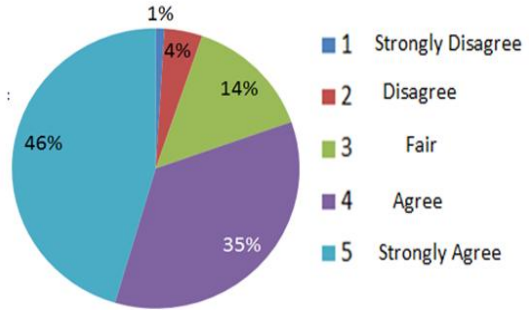

Figure 5: Active Participation

From Figure 6 it shows 78% respondents agreed on the Evaluation methods should be considered in software development. Prefer to consider in evaluation methods in agile software development. Remaining 6% disagree with the points mentioned above.

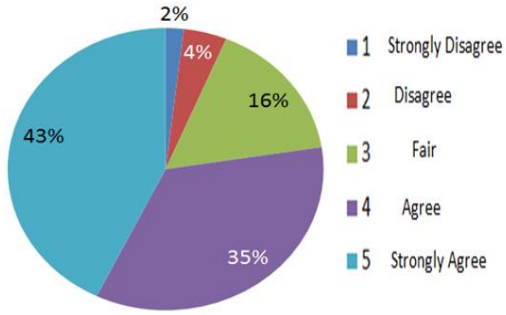

Figure 6: Usability Evaluation in the agile software process

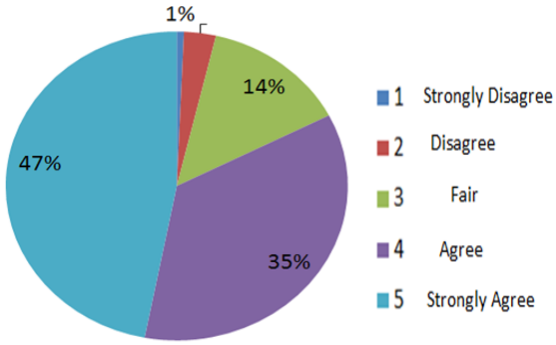

Figure 7: Use variousprocesses in an Agile Software Model for faster development

From Figure 7 it shows 82% respondents agreed on a development such process that is the part of the software model to make development faster. The other 4% disagrees with such model used in software process.

The focus of this paper mainly is to analyse the role of usability and users in the software model. From the research, it has been discovered that in the software

development, the parts played by HCI and users are important. In addition role of helping process in software model that makes development faster and effective.

After analysis of various software models through the numerous factors and keeping survey analysis report that discussed, it has been found that all models expect agile model are expensive to use (in term of cost, time and resources), used for big projects and having lack of Usability approaches. Whereas agile software model is a renowned model and is followed by many companies for small medium and large project. Hence introducing usability approaches in agile model increase the efficiency and usability of software.

The same project subsequently was developed using the most popular methodologies such as Scrum, Waterfall Model and OSSD and was rated using the scale of 0 to 4 (0= No, 1= fair, 2= Good, 4= Excellent). The results shown in figure 8 are based on three important features; Usability, User Involvement in software development and Time taken to meet the deadline of the project deployment. From the outcomes it indicates that Usability concentrated more in AUSL as compared to other models. On the other hand, User Involvements was observed more in AUSL and Scrum. The time for completion outcome shows that by using OSSD the time of completion will be lesser as compared to other software models.

There are some essential features that mostly considered in validation of the software model. In the proposed AUSElifecycle, Industry Standards should be followed to validate every process and make the processes of AUSE standardize.

To validate AUSE life cycle, ISO 9241:210 [10,16,,17] (Usability standards) and IEEE Std 12207-2008 (System Context Processes)will be followed. The International Standards (ISO 9241:210 and IEEE Std 12207-2008)

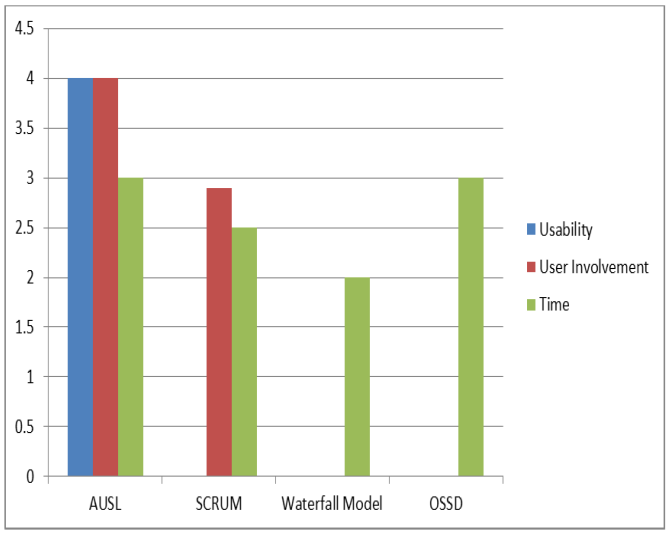

Figure8: Comparison with other Models

6. Validation

determine a common model for software life cycle process, having a well-defined terminology that can be recommended by the software industry [11].Figure 9 shows the most common processes of system set by the (International Standards Group) that may be performed during the lifecycle of software system. The outcome mentioned in each process need to be achieved to standardize the process.11 standard process are use in order to validate and standardize the AUSE life cycle then random survey was taken globally in which 49 respondent filled the survey form. Later SPSS tool was used to perform different analysis.

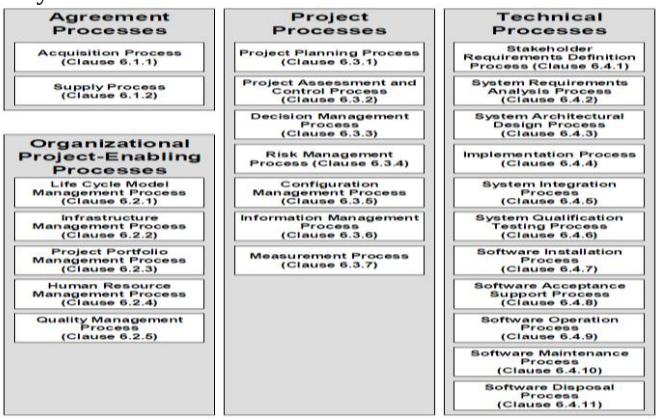

Figure 9: System Context Processes [12]

System Requirements Analysis Process

System Requirements Analysis Process is to convert the described stakeholder requirements into a set of preferred system specialized requirements that will monitor the style of system. The reliability score of this section checked by SPSS was 0.7579.

Stakeholder Requirements Definition Process

The Stakeholder Requirements Definition Process is to define those requirements need by the stakeholders.It examines and converts these into a typical set of stakeholder requirements that show the designed connections the program will have with its functional atmosphere and that are the referrals against which each causing functional support is verified to be able to validate that the program satisfies needs.The reliability score of this section checked by SPSS is 0.7701.

System Qualification Testing Process

The purpose of the Systems Qualification Testing Process is to ensure that the requirement specified by the user is tested and ready for deployment.

Usability Standards

The standard describes 6 key principles of human centered design act as a manifesto for the field of user experience. This process standard is responsible for managing design processes and gives an overview of the activities that are recommended for human centered design. The reliability score of this section is 0.829.

Software Acceptance Support Process

The purpose of the Software Acceptance Support Process is to give confidence to the user that end product will meet user's requirements. The reliability score of this section checked by SPSS is 0.7728.

Table 1: Descriptive Analysis of System Requirements Analysis Process

	N	Minimum	Maximum	Mean	Std. Deviation
A defined set of system functional and non-functional requirements describing the problem to be solved are established	48	2.00	5.00	3.5000	.82514
The appropriate techniques are performed to optimize the preferred project solution	48	1.00	5.00	3.4167	1.02798
Do system requirements analyze for correctness and testability in AUSEL	48	2.00	5.00	3.1458	.79866
Understanding the impact of the system requirements on the operating environment done by AUSEL	48	1.00	5.00	3.4583	.87418
Do AUSEL prioritized, approved and updated the requirements as needed	48	1.00	5.00	3.1875	1.02431
Consistency and traceability are established between the system requirements and the customer's requirements baseline and are observed in AUSEL	48	2.00	5.00	3.6042	.76463
Changes to the baseline are evaluated for cost, schedule and technical impact	48	1.00	5.00	3.5000	.89917
The system requirements are communicated to all affected parties and baselined	48	2.00	5.00	3.4792	.79866
Valid N (listwise)	48				

Table 2: Descriptive Analysis of Stakeholder Requirements Definition Process

	N	Minimum	Maximum	Mean	Std. Deviation
Is the required characteristics and context of use of services are specified	48	2.00	5.00	3.4583	.79783
Does the constraints on a system solution are defined	48	1.00	5.00	3.2292	.77842
Traceability of stakeholder requirements to stakeholders and their needs is achieved	48	1.00	5.00	3.4583	.96664
The basis for defining the system requirements is described	48	1.00	5.00	3.6667	1.03827
The basis for validating the conformance of the services is defined	48	1.00	5.00	3.5000	.92253
A basis for negotiating and agreeing to supply a service or product is provided	48	1.00	5.00	3.5208	1.01036
Valid N (listwise)	48				

Table 3: Descriptive Analysis of System Qualification Testing Process

	N	Minimum	Maximum	Mean	Std. Deviation
Criteria for evaluating compliance with system requirements are developed	47	1.00	5.00	3.3404	.73059
The integrated system is tested using the defined criteria	48	1.00	5.00	3.3750	.95928
Test results are recorded	47	1.00	5.00	3.1277	.89969
Readiness of the system for delivery is assured	48	1.00	5.00	3.3750	.89025
Valid N (listwise)	46				

Table 4: Descriptive Analysis of Usability Standard

	N	Minimum	Maximum	Mean	Std. Deviation
The design phase mentioned in IUP and CASI is based upon an explicit understanding of users, tasks and environments	48	2.00	4.00	3.4375	.61562
Is the design driven and refined by user-centered evaluation in IUP and CASI process	47	2.00	5.00	3.4043	.90071
Are users involved throughout the design and development process in AUSE lifecycle	48	2.00	5.00	3.7917	.87418
The processes in AUSL are iterative	48	1.00	5.00	3.6250	.89025
Is design phase (IUP and CASI) addressing the whole user experience	48	1.00	5.00	3.4167	1.10768
Is the design team included multidisciplinary skills and perspectives	48	1.00	5.00	3.6458	1.06170
Valid N (listwise)	47				

Table 5: Descriptive Analysis of Software Acceptance Support Process

	N	Minimum	Maximum	Mean	Std. Deviation
The product is completed and delivered to the user	48	2.00	5.00	3.3125	.77614
User acceptance tests and reviews are supported	48	1.00	5.00	3.1875	.86679
The product is put into operation in the users' environment	48	1.00	5.00	3.2917	.94437
Problems detected during acceptance are identified and communicated to those responsible for resolution	48	1.00	5.00	3.3125	.97099
Valid N (listwise)	48				

Software Configuration Management Process

The purpose of the Software Configuration Management Process is to integrateall process of software and make them available to concerned parties. The reliability score of this section is 0.8196.

Table 6: Descriptive Analysis of Software Configuration Management Process

	N	Minimum	Maximum	Mean	Std. Deviation
A software configuration management strategy is developed	48	1.00	5.00	3.5208	.92229
Items generated by the process or project are identified, defined and baselined	48	2.00	5.00	3.2708	.81839
Modifications and releases of the items are controlled	48	1.00	5.00	3.2292	.90482
Modifications and releases are made available to affected parties	48	1.00	5.00	3.1667	.88326
The status of the items and modifications are recorded and reported	46	1.00	5.00	3.2174	1.00914
The completeness and consistency of the items is ensured	48	1.00	5.00	3.5625	1.12810
The storage, handling and delivery of the items are controlled	48	1.00	5.00	3.2083	1.03056
Valid N (listwise)	46				

Software Verification Process

The purpose of this Process is to ensure that the specified requirements must be available in the each process of the software product. Reliability score of this section is 0.8081.

Table 7: Descriptive Analysis of Software Verification Process

	N	Minimum	Maximum	Mean	Std. Deviation
A verification strategy is developed and implemented	48	1.00	5.00	3.3333	.97486
A criterion for verification of all required software work products is identified	48	1.00	5.00	3.5000	.87519
Required verification activities are performed	48	1.00	5.00	3.3333	.78098
Defects are identified and recorded	48	1.00	5.00	3.3750	1.00266
Results of the verification activities are made available to the customer and other involved parties	48	1.00	5.00	3.1458	1.03121
Valid N (listwise)	48				

Software Validation Process

The purpose this Process is to ensure that the requirements for a specific intended use of the software work product are fulfilled.Reliability score of this section is 0.8551.

Table 8: Descriptive Analysis of Software Validation Process

	N	Minimum	Maximum	Mean	Std. Deviation
A validation strategy is developed and implemented	48	1.00	5.00	3.3958	.89299
A criterion for verification of all required software work products is identified	48	1.00	5.00	3.2708	1.00508
Required validation activities are performed	48	1.00	5.00	3.3333	.85883
Problems are identified and recorded	48	1.00	5.00	3.3333	.88326
Evidence is provided that the software work products as developed are suitable for their intended use	48	1.00	5.00	3.2083	.92157
Results of the validation activities are made available to the customer and other involved parties	48	1.00	5.00	3.4375	1.00861
Valid N (listwise)	48				

7. CONCLUSION

Agile development methods are the most flexible approach for software development where the development team keeps on improving the software with ongoing involvement of user.Inspite of its flexibility approach in software development, agile methods are not integrated with usability approaches.Whereas it is crucial to integrate in order to achieve software usability.It is of essential quality to incorporate usability process in the agile software method. This paper has produced a variety of contributions: literature review, current gapsin software models, survey, proposed agile software model, experiments and results.From the literature and proposed life cycle we derived that there are many benefits that can be achieved by integrating usability in the agile software model. A few major loopholes were succinctly explained under the heading of current gaps in software models.A survey was conducted among IT professionals to analyse Usability Evaluation in agile software development. After getting the survey results, a proposed agile mode i.e. Agile Usability Software Engineering Lifecycle is proposed.Meanwhilethe AUSE life cycle was validated following the IEEE Std 12207-2008 and ISO 9241-210 (Usability standards).

REFERENCES

[1] Atiwithayaporn S. &Rivepiboon W., A development of the effectiveness evaluation model for agile software development using the balanced scorecard, (2013), *IMECS*, Hong Kong

[2] Glaiel F., Agile Project Dynamics: A Strategic Project Management Approach to the study of Large-Scale Software *Development using System Dynamics*, (2012), [Massachusetts Institute of Technology]

[3] Lee J. C., Integrating Scenario-based usability engineering and agile software development, (2010), [Virginia Polytechnic Institute, State University],

[4] Divita J., Morris R., &Osga G., Rapid, Agile Modeling Support for Human—*Computer Interface Conceptual Design*, (2008), [SPAWAR Systems Center PACIFIC, California]

[5] Ghanam Y., *Discount Usability Testing*, (2007), [University of Calgary]

[6] Akbari, F. , "A Review to the Usage of Concepts of Software Architecture in Agile methods", International Symposium on Instrumentation & measurement, Sensor Network and Automation, *IEEE Software*,vol**2**, pp : 389 –392, 2012.

[7] Ahmad, W. F. W., Butt, S. M., & Rahim, L. Usability Evaluation of the Agile Software Process. In Advances in Visual Informatics (pp. 640-651, 2013). Springer International Publishing

[8] M Singh, "U-SCRUM: An agile methodology for promoting usability". *Proc, AGILE 2009 conference (Agile '09), IEEE* Press,,pp, 555-560,2009.

[9] P, Mcinerney and F, Maurer, "UCD in agile projects: dream team or odd couple?" Interactions, vol. 12, issue 6, ACM Press, pp, 19-23,.2007.

[10] Hedberg, H., &Iivari, N.. Integrating HCI Specialists into Open Source Software Development Projects. In Open Source Ecosystems: Diverse Communities Interacting (pp.51-263,2011). Springer Berlin Heidelberg

[11] Butt, S. M., Ahmad, W. F. W., & Fatimah, WAn Overview of Software Models with Regard to the Users Involvement. *International Journal of Computer Science Issues (IJCSI*), **9**(3 (1)), 107-112, 2012,.

[12] Thomas Memmel, Fredrik Gundelsweiler , HaraldReiterer, CRUISER: A Cross-Discipline User Interface and Software Engineering Lifecycle, Human *Computer Interaction*, Part I, HCII, LNCS 4550,2007, pp.174-183, 2007, Springer, Heidelberg (2007)

[13] Pyla, P. S., Pérez-Quiñones, M. A., Arthur, J. D., &Hartson, H. R. (2008). Towards a model-based framework for integrating usability and software engineering life cycles. arXiv preprint cs/0402036

[14] Masood Butt, S., & Ahmad, W. F. W. (2012, June). Handling requirements using FlexREQ model. In Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on (pp. 661-664). IEEE.

[15] Butt, Saad Masood, and Wan Fatimah Wan Ahmad. "ANALYSIS AND EVALUATION OF COGNITIVE BEHAVIOR IN SOFTWARE INTERFACES USING AN EXPERT SYSTEM." International Journal 5 (2012).

[16] IEEE STANDARDS ASSOCIATION, 12207-2008-IEEE System and Software Engineering- Software life cycle processes, "https://standards.ieee.org/findstds/standard/12207-2008.html

[17] ISO 9241-210: Ergonomics of human-system interaction -Part 210: Human-centred design for interactive systems 2010 http://www.iso.org/iso/catalogue_detail.htm?csnumber=52075

[18] Ahmad, W. F. W., Butt, S. M., & Rahim, L. Usability Evaluation of the Agile Software Process. In Advances in Visual Informatics (pp. 640-651). Springer International Publishing, (2013)
.