

Available online at <http://www.mecs-press.net/ijem>

Scrum-Based Methodology for Product Maintenance and Support

Riffat Naz, M. N. A. Khan, Muhammad Aamir

Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan

Abstract

Product maintenance techniques have significant importance because they are much cost effective and less time consuming to maintain a product or software rather to change it. There are different product maintenance and support techniques. These previous techniques do not solve user/clients bugs, issues and enhancements effectively and efficiently. Scrum is being used now a day as a quick, flexible and holistic methodology to develop software. In Scrum projects there is the much customer involvement is included which help to develop a user oriented product. Users can change their requirements in Scrum. Many techniques have been proposed for product maintenance and support. However, in this paper, there have been a detailed literature review of existing product maintenance techniques and also presented a new proposed model and technique for the product maintenance by using Scrum methodology. This Scrum based model for maintenance is designed and based on the analysis of client request types and severity (priority). In our approach, The Session attendees (Scrum Master, Product Owner and Team) choose that bug, issue or enhancement first which has an urgent type or higher priority request and resolves it then select low priority request or non urgent requests and facilitates the clients in timely manner. In this way this proposed model works effectively and defiantly to meet the customer's demand. A comprehensive study on product maintenance and support has been carried out which adds to the current practices in the scrum. We found that maintenance phase of the scrum has been given less attention in the existing literature. In view of this, we have made an attempt to propose a novel model that focuses on the maintenance phase of scrum.

Index Terms: Scrum, Product Backlog, Rapid Application Development, Agile Software Development, Extreme Programming, Software Quality Assurance.

© 2016 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science.

1. Introduction

Scrum is a framework used for software development and managing software product development. It is an iterative and incremental agile software development methodology. Scrum describes a product development

* Corresponding author.

E-mail address: riffatnaz704@gmail.com, mnak2010@gmail.com, amir_amirkhan94@yahoo.com

stratagem which is flexible and holistic where the development team works as a one entity to attain common goal and challenges. It facilitates with a self-organizing team in which team members are encouraged by physical co-location; or may be close online association and group effort of all the team members. It can be in the form of daily face-to-face communication among the entire members of the team and disciplines within the project. During the Scrum project the customers can change their requirements. The client can ask for adding or removing some features and these unexpected changes cannot be easily tackled in a traditional software development approach. Whereas, scrum implements an empirical approach which focuses on make the most of the team's capability to hand over rapidly and take action to up-and-coming the requirements.

Scrum was first defined as a flexible product development stratagem in 1986 by Hirotaka Takeuchi and Ikujiro Nonaka in the "New Product Development Game". Soon after they elaborated that scrum was a sort of organizational knowledge creation that was mainly fine at getting about novelty constantly and incrementally. In 1990s Ken Schwaber named the technique as "Scrum". In 2001, Schwaber illustrated Scrum technique in his book named "Agile Software Development with Scrum". Soon after, Schwaber established the Scrum Alliance. Scrum word is not an acronym. Scrum is derived from agile development and agile is derived from rapid application development (RAD).

1.1. Scrum process

In scrum process there are roles, events and artifacts and each of them is divided into sub group. There are three main roles in scrum: Product Owner, Development team, Scrum master.

- a. **Product Owner:** The product owner is the representation of the stakeholders, voice of the customers. He/she is just like a bridge between scrum team and their clients. Product owner has the duty to write the customer-centric things such as user stories frequently. There is one product owner in a Scrum team.
- b. **Development Team:** In scrum, the development team consists of three to nine members. They are liable for delivering a complete and deliverable product on the completion of every sprint. Development team analyses the project, makes design of it then starts development on it. After that they test and maintain documents of it. They are very self-organizing.
- c. **Scrum Master:** In Scrum process Scrum Master is responsible for removing obstructions and hindrances in the development of a quality product. Scrum Master cannot be called Project Manager, but he performs like safeguard or shock absorber among the development team and every distracting influences. The Scrum Master makes compulsory that the team members follow the scrum rules. He makes sure the leads vital meetings. He also makes sure the development challenges make better the team. He acts like a referee who monitors the others and emphasize these double views.

1.2. Events

Events are of three types: Sprint, Meetings and Extensions. These events are discussed below one by one.

1.2.1 Sprint

Sprint is a fundamental and basic element in Scrum development. It is like a time-boxed effort. Every sprint has certain limited or fixed time duration. This time duration is fixed for every sprint in advance. Generally time duration for Sprint can be one week to one month. By making a planned meeting sprint is started. A record named sprint backlog is maintained where all the details related to sprint are identified. After this sprint target is completed. Each and every sprint is finished by making an evaluation, assessment and demonstration meeting on it. In the last meeting sprint progress is analyzed. Meetings are of three types: Sprint planning meeting, Daily scrum meetings and End meetings.

Sprint planning meeting: This type of meetings is held at the start of the sprint cycle. This sprint cycle

duration is 7–30 days. In this meeting, the strategy to finish the selected work is devised. The meeting aims at arranging the backlog for sprint and all important aspects that how much time will be consumed for this task and with how much effort.

Daily scrum meeting: These types of meetings are held on daily basis. When sprint is started, every day this meeting occurs that is why it is known as daily scrum or also known as stand up meeting. This meeting has following features:

- All team members show their work updates and status.
- It begins particularly and fixed on time regardless any member is failing to attend.
- Time and location is fixed for every day.
- Time duration is fifteen minutes.

Obstructions which are recognized in this meeting are noticed and written by the Scrum Master. He finds the solutions for this.

End meetings: When sprint cycle is completed two types of meeting is arranged: Sprint Review Meeting and Sprint Retrospective. In Sprint Review Meeting, the entire planned act is analyzed. Time duration is four hours. In the Sprint Retrospective what has went before sprint is discussed. In this meeting two most important questions are raised: What left good for the duration of the sprint? What possibly will get better in the next sprint? Its time duration is three hours.

1.2.2 Extensions

Extensions include Backlog refinement and Scrum of scrums. Backlog refinement is a continuing procedure in which product backlog is evaluated. Things are tested and verified so that they are properly arranged and prioritized. Scrum of scrums is a method to mount and climb up scrum to considerable development grouping. It permits groups of teams to talk about their work. Each and every day scrum inside a sub-team terminates by specifying one associate like an "ambassador". This ambassador takes part in the every day meeting with ambassadors from the other teams. This is known as the Scrum of Scrums. The schema will be similar to a Daily Scrum.

1.3. Artifacts

1.3.1. Product backlog (PB)

PB is a structured catalog of requirements. It is maintained in development process intended for every product. PB contains all non functional requirements, features and bug fixes. It is arranged by the Product Owner. Things which are inserted in PB are generally written in the format of a story. PB contains the order in which a product is when and how delivered. Anyone can change it but main in charge of PB is Product Owner. PB not only contains user stories but simply items because scrum is unbiased on requirements. Items in PB can be in any form that can be user stories or use cases etc.

PB maintains requests related to a product modification. New features are affixed in this way. Containing properly maintained rules regarding how work is added, eliminated and arranged makes easier the entire team formulate improved decisions concerning the amendment in the product. Items are selected from backlog and make user stories and then worked on it.

1.3.2. Sprint backlog

The sprint backlog is the catalog of work that is essential for the Development Team to deal with it for the

duration of the next sprint. List is obtained by choosing user stories from product backlog. Development team select user stories from PB for a certain sprint by inquiring the question such as “Can we also do this?” In scrum working process every team member select tasks by himself according to his skills. No one assigns a specific job in sprint backlog. This is called as self-organization. An additional task board is bringing into played for mentioning the status of the tasks. In front of every task, there is a status mentioned by writing “in progress” for tasks which are under process or “done” for the completed tasks.

1.3.3. Product Increment

It is the total of the entire PB items finished for the duration of all prints. When a sprint is completed, increment is required to be done. This increment follows to the criteria defined by the scrum team called Definition of Done.

1.3.4. Sprint Burn Down Chart

The sprint burn down chart is an open presented chart. It gives you an idea about work that is remaining in the sprint backlog. It displays an outlook about the progress of sprint. It in addition, it provides a fast and rapid visualizations and image in support of reference. In the meeting arranged to plane sprint, the idyllic burn down chart is designed.

1.3.5. Release Burn Down Chart

It is the means for the team to follow and check progress. It also gives visibility. Scrum Master updates it when sprint ends. The X-axis shows the sprints and the Y-axis shows remaining work. The release burn down chart gives an idea about completed work, pending work and work has been done.

1.4. Software Maintenance

Software goes into maintenance phase when it is delivered to clients. Software Maintenance method is used to make available cost-effective sustainability to a consigned software product. It is also known as software servicing or software maintenance. This phase consists of repeated alterations and modifications to software. Objective is to maintain software working in a cost effective manner. It does not support to new aspiration to add new features/functionalities. Main changes which developers commence are generally bug fixing (issue identified by client), change (business rules are changed) or enhancement (improvement or new features).

According to Rajlich [1] throughout the maintenance phase there are as a minimum two versions of software in existence, first that is in production and is serviced and second is in corresponding the developer team develops for the upcoming releases. Maintenance may possibly persist provided that software is in use.

1.5. Agile maintenance

Agile maintenance is the most excellent option to stay pace for customer requirements Omanovic and Buza [2]. This is answer to customer requests for a change. When a client’s behavior changes it may cause amendment in the release plan. Secure and steady development rate is the significant process sign in agile maintenance.

1.6. Scrum-ban

It is a model used for software fabrication. It is established at Scrum and Kanban. Scrum-ban is mainly set for the maintenance of the projects. Mostly used in those projects those have with recurrent, sudden and

unforeseen programming errors. These set of circumstances have time bounded sprints are of no significant use, so daily meetings and further attempts can be applied. By using it the workflow of team is give directions in a line of attack which permits least amount of time for completion of programming error.

2. Literature Review

The basic purpose of the literature review is to analyze the work of other that what type of work they have done and how they have done. In this section we have analyze different product maintenance techniques. Also we did comparison of different papers based on its efficiency and cost effective approach for solving the problem.

Frakes and Kang [3] discuss research on software reuse, major contributions and unsolved problems. Software reuse is constructing new software by using the existing software. Its idea is to get better and advance software quality as well as its yield/output. Previous methods used in software engineering were insufficient and poor. Most software systems were designed according to a specific domain. With the passage of time as domains changed system also need to be changed. It is hard to replace a system by designing it again. That is why software reuse is a better approach to meet the new domain requirements. Quality and productivity can be enhanced. It also makes betterment in the process software production.

Lin et al. [4] presents a method which is layer-based and it is for development of software rapidly. This layer-based method follows the course of action of Extreme Programming (XP). XP involve extremely communicative programming languages which can be Java and some other CASE tools. It is cleared that these method in XP focus on software development rapidly but only good for tiny projects or else average sized projects. So for the valuable assistance upon the better development, it enforces a concept which is architecture based but it is layered based design. Author mentioned that, for modeling tool this technique uses UML and other technique which is Petri nets. Further for explanation author presented an example of an application in this paper. Author described suitable software features in this document.

Sohaib and Khan [5] discuss Extreme Programming (XP) which is the form of agile software development, Usability and Discount usability. Authors proposed an integrated approach discount usability engineering in XP process. Authors describe the way in which usability can be incorporated with the XP for an effective usable software system. Authors mentioned in the document discount usability which includes scenarios, card sorting, heuristic evaluation. Discount usability can be used with least and fewer set of support. It can be incorporated into lifecycle of XP project. Issue discussed by authors is that XP does not consider usability because it is a heavy process, for this issue authors proposed the discount usability technique which it lightweight. This technique has four phases:

- It uses scenarios together in the company of user stories in initial stage.
- Sorting of cards play the role of Release Planning, in planning stage.
- Evaluation based on heuristic is used at some stage in the approval of testing.

Lin et al. [6] discussed a tool, used in the development of software and makes it rapid. Proposed tool require a reduced amount of design effort. It enhances the development speed of software. It leads the successful management by enforcing a layer of specification on system parts. Petri net and some other RAD method like scrum functionalities are used. The discussed tool has a layer on system parts. All the management actions are added in a proper way. It improved efficiency of development team. A friendly collaboration is developed among team members. Strengths of the techniques:

- Less design work.
- Rapid software development.
- Speeds up components construction.
- Effective management

- Less effort needed.
- Productivity increased
- Team collaboration developed.

Gerber et al. [7] discuss Rapid Application Development (RAD) methodologies and their implications at some stage in the development. They discussed both of large systems and complex systems.

The research approach described in study discussed is derived from a case study. The system discussed in this case study is the achievement of a development team that was small but that was much competent. They adopted one of a RAD methodology that was agile development approach. Gerber et al. [7] selected this approach because it is lightweight and it offers reduced cost and development time. After the case study they found some practical implications as given below:

- Confusion among Team Members related to positions. Other confusion are related to task/responsibilities:
- In the team of development roles assigned to the members are very confusing.
- Roles and responsibilities in traditional methodologies:

Rousseaux and Lhoste [8] described that with the faster evolution in IT tools rapid software prototyping is a best way to meet the demands of users instantly. With a classical software development cycle it is difficult to develop an application so fast which normally takes several months for development. By using RAD functionalities and interfaces can be revised with the passage of time. Author mentioned some past difficulties, the rapid prototyping approach corresponding to the web tools. Ajax is a latest way for the applications of web said by Jesse James Garrett in 2005 for the first time. Garrett named this phrase for the architecture and makes it equal like the new production group of web Applications. These new productions can be Google maps. Ajax cannot be said as an actual language of programming. This is an advanced design model. It is made up of a lot of connected technologies and thoughts. With the use of Ajax in applications of web we can get back data to the server asynchronously.

Maschotta et al. [9] says that in the development of a simulation-based system design analysis and validation is not a simple job. Author in this paper presents a framework which allows the rapid development of reusable simulation-based applications. The proposed framework supports agile, evolutionary, iterative, and architecture-centric software development process. For explanation author described an example which is designed and works according to the proposed model. In the proposed model development of each component, implementation of simulation application and optimization are separated. That is why development of these parts is possible in parallel and this is agile software development process. Author mentioned some following advantages of this technique

- It easily allows reusability of existing components
- It decreases the programming effort for system
- It is user-friendly system design tools.

In addition to the advantages of proposed framework besides of agile software development process the system expert can work directly with the simulation-based system design tool. So in this way software errors will be declared soon. Developers can fix bugs earlier. At early stages of product development all additional changes and desired enhancements can be integrated into the tool.

Chehili et al. [10] discussed a Service Oriented Architectures framework for agile development. This framework named as FASOAD. It employs the agile method for SOA development as a component assembly model. As it uses the foundation of Service Component Architecture (SCA) model, so it permits developing SOA projects by implementing agile technique.

Loffler et al. [11] discussed a method of agile which is scrum. This method has much customer participation. It gives ease of techniques that are used in the analysis of requirements and testing. This model is consists of

different useful techniques which are important in maintaining the record of users requirements. These techniques are in the nature models that are already tested. This model can be used by both testers and developers. It is easy to adopt and language used in it is simple. Test scripts derived from this model for fitNesse / selenium. Scrum focal point is implementation. New and users preferred features are included in previous design by using iterative steps. They are known as sprints. When these sprints are included, they caused deprived and deficient requirements. As the consequences user feel trouble to using software. At some stage in sprints the next problem is continuous addition of new functions and their implementation. These problems are like latest functions are not harmful for existing ones.

Vlaanderen et al. [12] discussed the gaps in agile development process and introduced a novel method named as agile requirements refinery to fill it. Author used case study method to illustrate and assess the method. This Agile requirements refinery is an extension to the Scrum practice which facilitates product managers to deal with complex requirements in an agile development environment. Related to maintenance, author described in this paper that the maintenance planning idea is persuaded all over its development. This idea illustrates functionality related to the maintenance. It was firstly commenced on the PB for the analysis. The whole SPM lifecycle idea persists for 7 months.

Schramm et al. [13] discussed issues and alleviations stratagem when agile is used in student projects. For this purpose a group of 4 to 5 students were made in a semester and assigned scrum bases project. Every student of each group is asked to maintain a record or written document for whole project. During the project student was asked the answer of questionnaires throughout the project. At the end of the project when student maintained record was analyzed some frequent problems and mitigation stratagem are disclosed which are as given below:

- Issues in organizing the meetings
- Communication issues and problems
- Problems in assigning roles of projects
- Some requirements engineering problems
- Continuous checks and test issues
- Issues in adherence of process.

Omanovic and Buza [14] discussed the significance of agile maintenance. As authors mentioned in this document that agile maintenance is the most excellent preference to meet the client demands effectively. This way deals with client's change requests in very short time. Management process can be affected by changing the client's behavior. This may cause delay in product release. Authors mentioned here some problems in accordance with this and presented some solution to put a stop to them. Client's liking is the mainly significant part in the software development. By knowing the cause of change in but customer's behavior developer can make it stronger and user oriented. An un-balanced relationship between client and developer is identified by developer first by means of the velocity instability. Steady development velocity is mainly significant method sign. It is important for the developers in agile maintenance. According to author a balanced relationship among developer and customer should be always in limits.

Khalane and Tanner [15] find and introduce the affects of project stakeholders related to Software Quality Assurance (SQA) in a Scrum environment. Authors mentioned that due to the absence of concrete guidance the Scrum needs to be adoptive in nature. This way consists of some practices those are copied or learned from different methodologies and cautiously and thoroughly making the design architecture. It gives a mutual / shared ownership. This document further recommends that Scrum requires to be considered as a framework of 'empty buckets' which have to be fill up according to client's demands and some definite SQA practices. A case study was proposed and discussed here. To find the way for attaining software quality requirements by using Scrum. It also gives an idea about scrum working in SQA related environment. Still needs to be more exploration in depth. It also makes known that a greater part of the regular Agile SQA practices mainly have a discussion regarding it in the literature are XP practices. It is for that reason because researchers believe that

Scrum does not set down SQA practices like agile SQA practices such as TDD.

Santos et.al [16] discussed Scrum and its implementation in this document. According to authors Scrum is a knowledge-intensive activity and depends on learning to evolve. Authors discussed Scrum with respect to the organization learning. A qualitative research has been conducted on it by involving key members from UOL (the most significant Brazilian company). The core and foremost results are as given below:

- The procedure of Organizational Learning may be confirmed all the way through the learning of individual member.
- Changes and other thing can be also measured in same way.
- It also created a link between Organizational Learning and Scrum implementation.
- It helps to find changes and issues which are related to practical field.
- These are still issues in agile like knowledge management.

Authors refine a number of variations which an organization may experience when they implements Scrum. Rajlich [17] discussed software maintenance and evolution in this document. With the passage of time, technology change, trend change and due to many other factors, software has need of regular maintenance. This becomes software evolution. Author also mentioned some processes of evolutionary software development. These are agile, iterative, open source, inner source and other processes. He encourages analysis in these above processes. A model phased model of software change (PMSC) is presented in it. This model shows fundamental task of software evolution and software change. It also gives tracks for enhancement. Author also made a comparison in software evolution with software maintenance.

References [21-54] reviewed different techniques in different domains and reported their critical evaluations along with a workable framework where necessary.

3. Proposed Solution

Study of the contemporary literature delineates that scrum provides good support in the development of small and medium size software projects. Furthermore, the existing literature and the industrial survey conducted during this study explain that it is a tedious job to maintain a product using state of the art scrum model. Therefore there is a strong need of research in order to resolve the problems and issues arisen during the product maintenance and support phase while using scrum. The key questions are:

Q1: How to use the scrum process effectively during the software maintenance phase?

Q2: How to write "User Stories" during the software maintenance phase?

Software goes into a maintenance phase when it is delivered to the client. The maintenance phase consists of repeated alterations and modifications to the software. The objective is to keep software working and updated in line with the business processes in a cost effective manner. The main changes which developers generally take account of consist of the following three activities:

- Bug fixing (problems or malfunctioning identified by the users)
- Changes requested by the client (e.g., business rules of the organization are changed)
- Feature enhancement (including adding new features into the software)

Bugs/issues are identified by a bug tacking tool or may be reported from the client side. Whenever a change, issue or enhancement request is reported by a client, the development team places it in the product backlog. Every bug/change request has its unique identification number (ID). As we know that product owner maintains the product backlog so IDs of bug/change requests are brought into the product backlog during the maintenance phase. User stories are selected from the product backlog for each sprint. Some of the problems are:

- What will be the criteria to select a bug/change request for the sprint?
- On what basis these requests can be selected for the next sprint?
- On what basis these requests can be immediately handled?

These are the key challenges when we use scrum methodology in the maintenance phase.

3.1. Proposed Solution

In this section, we endeavor to propose a viable solution for the above mentioned problems. To address the three research questions raised in the previous section, we suggest using a two-pronged approach to handle the change requests and bug fixing. The proposed approach is as follows:

- Assign severity and priority level for bug fixing tasks
- Categorize the change request into urgent or normal

When bug/change IDs are brought into the product backlog, there should be an indicator field (or column) for every change request which shows its status. In the status field, priority and severity level of the “bug fixing” should be mentioned or category (or type) should be mentioned in case of a change request. The product owner assigns priority to a certain backlog and describes which backlog is mainly required. Then the development team decides modus of undertaking the work which can be finished in the upcoming sprint. In this case, product owner can update the status field whether it is an urgent or non-urgent request so that the development team can pick it from the product backlog by checking its priority and type. Development team will decide to choose it for the current sprint backlog or postpone it by looking at its status. In this way, bug fixing and change requests can be handled timely and efficiently.

In order to provide a solution for the given problem we propose an enhanced Scrum Model for the Software Product Maintenance.

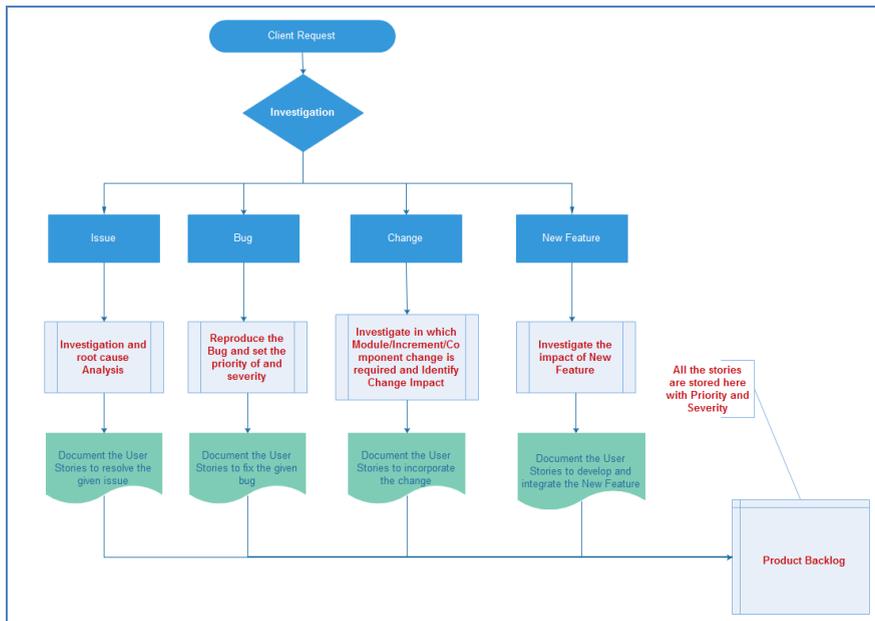


Fig.1. The Proposed Scrum-Based Software Maintenance Model.

The investigation decisions in the proposed model are of four types as described below:

Issue: Issue is a conflict in requirement understanding and development.

Bug: Bug can be programming or logical error or any deviation from the requirements.

Change: Change is the modification in the current system or module due to business roles changes.

New Feature: New Feature is a new functionality need to be incorporated in the given system.

Whenever a client request is generated, first of all an investigation session will be held in which the nature of the request will be identified. The session attendees include Scrum Master, Product Owner and Development Team. There are four types of requests:

- Issue
- Bug
- Change
- New Feature

If nature of the request is an issue, then it is the responsibility of the product owner to investigate root cause of the issue. Once the root cause is identified then product owner will document the user stories for resolution of the issues.

If the nature of the request is a Bug, then this is the responsibility of the team to identify the flow of the bug and investigate the severity and priority of bug. The resolution of bug will be then discussed with the Product Owner who will then start writing and documentation of the user stories for the bug fixing.

If nature of the request is a Change, then the Product Owner will start analysis to investigate purpose of the Change, nature of change (Urgent/Non-Urgent), identifying the modules that needs to be changed, assessing impact of the change on the other modules etc. Once the Product Owner finds answer for these questions, then he/she will start documentation of the User Stories for the new change.

If the nature of the request is New Feature, then the Product Owner will start collaboration with the Client or Client Representative in order to understand the feature to be added. The Product Owner will write and document the user stories for the New Feature.

Note that all the user stories should be maintained in the Product Backlog. Once all the above decisions carried out then normal Scrum flow will be started.

4. Data Analysis and Findings

It is difficult to evaluate a software development process because accurate data collection is hard to find. It can be done in many ways but we used a survey based approach to validate our proposed solution. For this purpose, we prepared a questionnaire and conducted a survey to find out the opinions of the practitioners on various aspects of the proposed solution.

4.1. Survey questionnaire

Our survey questionnaire was completed by many software developers etc at industrial level. Most of these practitioners are using scrum practically. Survey questionnaire include the pertinent, specific and core questions related to the scrum practices to validate our proposed solution.

Table 1. Survey Questionnaire.

Q. No.	Survey Question
1	Scrum methodology cannot be used effectively during software maintenance and support:- a. Strongly agree b. Agree c. Do not have any knowledge d. Disagree e. Strongly disagree
2	In maintenance phase while using scrum, it would be good to carry out an investigation session once client request is generated:- a. Strongly agree b. Agree c. Do not have any knowledge d. Disagree e. Strongly disagree
3	While using scrum, it would be better that we have to investigate that whether the client is requesting for the resolution of an issue or fixing of a bug or adding new feature to the system:- a. Strongly agree b. Agree c. Do not have any knowledge d. Disagree e. Strongly disagree
4	It would be better to reproduce a bug raised by the client:- a. Strongly agree b. Agree c. Do not have any knowledge d. Disagree e. Strongly disagree
5	Root cause analysis of an issue is too much important while using scrum or any other software process model:- a. Strongly agree b. Agree c. Do not have any knowledge d. Disagree e. Strongly disagree
6	While using scrum, it would be better to write and document the user stories in the product backlog for the resolution of bug/issue: a. Strongly agree b. Agree c. Do not have any knowledge d. Disagree e. Strongly disagree
7	While using scrum, It would be better to write and document the user stories in the product backlog for the modification in the current system or adding new feature to the existing system:- a. Strongly agree b. Agree c. Do not have any knowledge d. Disagree e. Strongly disagree

4.2. Survey Results

By putting these questions to industrial sector, we got results against every question. We discuss these results one by one.

Q1. Scrum methodology cannot be used effectively during software maintenance and support

When we analyzed this question, we found that almost 80% people agreed on this. It is hard to use scrum methodology effectively in software maintenance and support. Our proposed solution is helpful to use scrum methodology in maintenance effectively.

Q2. In maintenance phase while using scrum, it would be good to carry out an investigation session once client request is generated

Second question is related to maintenance when we use scrum. In our proposed model, firstly the client request is investigated to find whether it is a bug, issue, change or new feature. For this approach we found that more than 75% people agreed upon.

Q3. While using scrum, it would be better that we have to investigate that whether the client is requesting for the resolution of an issue or fixing of a bug or adding new feature to the system

According to our third step of the proposed model, it is better to investigate that whether client is requesting for bug, change, issue or new feature. 80% people agreed to the issue raised in the question.

Q4. It would be better to reproduce a bug raised by the client

This question is related to quality of software. We got 95% agreed and strongly agreed views on it.

Q5. Root cause analysis of an issue is too much important while using scrum or any other software process model

Again this is a core question related to our proposed model validation. We can validate our proposed model on the basis of responses we got. Again for this question, we got more than 97% agreed upon.

Q6. While using scrum, It would be better to write and document the user stories in the product backlog for the resolution of bug/issue

We have user stories in the product backlog. Developers pick user stories from product backlog. In our model, we proposed a step in which user stories related to bug, issue are added in product backlog once again. So the developer can get it from product backlog and respond it quickly and effectively. Related to this question we got 85% views which are in favor of this proposed step.

Q7. While using scrum, it would be better to write and document the user stories in the product backlog for the modification in the current system or adding new feature to the existing system

In product backlog we have user stories. Developers pick user stories from product backlog so in our proposed model we proposed a step in which user stories related to change or new features are added in product backlog once again. So the developer can get it from product backlog and respond it quickly and effectively. Related to this question we got 90% views which are in favor of this proposed step.

In this section, we have compared the proposed model with industry survey results. The 90% of views were in favor of Scrum product maintenance and support methodology model. We make an aggregate on every

question and find that it is better to use scrum methodology in product maintenance and support according to our proposed model. It will be more effective and efficient. It will respond the client requests timely and effectively.

4.3. Framework validation

In addition to the survey conducted for getting feedback from the practitioners on Scrum Maintenance, we also compared our proposed model with work of the other researchers published in the contemporary literature. The proposed framework has also been validated through a comparative study based on the previous work already carried out by the researchers in this area. Moreover, the scrum methodology in product maintenance and support is a new area of study in the field of Information Technology and is growing day by day. Tabular comparison model is presented in Table 2 to evaluate the research work carried out in this study *viz-à-viz* the previously published research. For comparison purposes, the work of Vlaanderen et al. [18], Omanovic and Buza [19] and Khan et al. [20] have been studied and evaluated *viz-à-viz* our proposed model. The comparison has mainly been drawn for specific attributes including: maintenance, requirements, support, time efficiency, analyzing client requests, product backlog and bug fixing. Our observations are reported in Table II.

Table 2. Comparative Table for Model Validation.

Sr #	Attributes/ Challenges	[18]	[19]	[20]	Our Proposed Framework
1	Maintenance	Yes	Yes	No	Yes
2	Requirements	Yes	Yes	Yes	Yes
3	Support	No	No	No	Yes
4	Time efficiency	No	No	No	Yes
5	Client request investigation	No	No	No	Yes
6	Request priority	No	Yes	No	Yes
7	Product backlog	No	Yes	No	Yes
8	Theme elaboration	No	Yes	No	Yes
9	Complex requirements	Yes	No	Yes	Yes
10	Bug fixing	No	No	No	Yes

On the basis of the comparison of the previously proposed methodologies with our proposed framework as shown in Table 2, we noticed that support, bug fixing and time efficiency have not been addressed by any previous model. Rest of the attributes are partially provided by some models. Whereas, our model accounts for the attributes which can be helpful for smooth and effective execution of the scrum-based software development tasks.

5. Discussion

After analyzing the proposed solution/model it is clear that to use scrum methodology in product maintenance and support we need to investigate client request first as in our proposed model (step no.1). Whenever a client request is generated, first of all an investigation session will be held in which the nature of the request will be identified. The session attendees include Scrum Master, Product Owner and Development Team. By investigating client request, categorize it into four different groups (issue, bug, change, new feature) as in our proposed model (step no. 2). Once the root cause is identified then product owner will document the user stories for resolution of the issues. If the nature of the request is a Bug, then this is the responsibility of the team to identify the flow of the bug and investigate the severity and priority of bug. If nature of the request is a

Change, then the Product Owner will start analysis to investigate purpose of the Change, nature of change (Urgent/Non-Urgent), identifying the modules that needs to be changed, assessing impact of the change on the other modules etc. Once the Product Owner finds answer for these questions, then he/she will start documentation of the User Stories for the new change. If the nature of the request is New Feature, then the Product Owner will start collaboration with the Client or Client Representative in order to understand the feature to be added. The Product Owner will write and document the user stories for the New Feature. All the user stories should be maintained in the Product Backlog. Once all the above decisions carried out then normal Scrum flow will be started.

When we evaluate our solution we find that 85% to 90% people is facing problem to use scrum in product maintenance and support. Our proposed model will be help software developers etc with efficiency. Our proposed technique is an effective technique to handle the maintenance or support request in a short span of time. To validate our proposed model we made a survey. We posted the survey online and requested the practitioners to provide their feedback about the key questions raised in the questionnaire. Results are very encouraging. By analyzing the results we found that product maintenance was a difficult and time consuming but by using Scrum in product maintenance and support now it is very efficient and customer satisfactory. We have compared the proposed model with industry survey results. The 90% of views were in favor of Scrum product maintenance and support methodology model. We make an aggregate on every question and find that it is better to use scrum methodology in product maintenance and support according to our proposed model. It will be more effective and efficient. It will respond the client requests timely and effectively. On the basis of the comparison of the previously proposed methodologies with our proposed framework as shown in Table II, we noticed that support, bug fixing and time efficiency have not been addressed by any previous model. Rest of the attributes are partially provided by some models. Whereas, our model accounts for the attributes which can be helpful for smooth and effective execution of the scrum-based software development tasks.

6. Conclusion and Future Work

In this paper, we have proposed a new technique and model for product maintenance and support phase by using Scrum software development methodology. This is an effective technique to handle the maintenance or support request in a short span of time. It provides an in depth knowledge about Scrum methodology working in maintenance phase. Applying Scrum in maintenance phase needs to be more efficient. It responses to urgent problems quickly compared to other maintenance techniques. The maintenance phase consists of repeated alterations and modifications to the software. These alterations and modification can be bug, change or enhancement. If there is a bug it needs to be solved in minimum time duration. In this case our proposed technique works effectively and efficiently. Different maintenance techniques have been compared and their performances have been analyzed. These techniques are not better because it is difficult to use keep using Scrum in maintenance. These are not well for medium size projects. The objective is to keep software working and updated in line with the business processes in a cost effective manner. Furthermore, the existing literature and the industrial survey conducted during this study explain that it is a tedious job to maintain a product using state of the art Scrum model. Therefore there is a strong need of research in order to resolve the problems and issues arisen during the product maintenance and support phase while using scrum. Moreover, technique is mentioned in order to cope up these issues. Techniques for product maintenance and support phase while using scrum are mentioned in this report. This technique is efficient and comparatively better than other product maintenance techniques. This technique is also works well in medium sized projects.

The technique has a very simple design model. When bug/change IDs are brought into the product backlog, there should be an indicator field (or column) for every change request which shows its status. In the status field, priority and severity level of the “bug fixing” should be mentioned or category (or type) should be mentioned in case of a change request. The product owner assigns priority to a certain backlog and describes which backlog is mainly required. Then the development team decides modus of undertaking the work which can be finished in the upcoming sprint. In this case, product owner can update the status field whether it is an

urgent or non-urgent request, so that the development team will pick it from the product backlog by checking its priority and type. In this way, bug fixing and change requests can be handled timely and efficiently.

The model has been extensively tested using various industrial surveys. The obtained results are very simple, efficient and encouraging. The model is also very simple to implement and can be used for the practical applications that demand a quick and quality solution/support. To validate our proposed model, we conducted a survey. For this purpose we prepared a questionnaire consisting of 13 questions. The questions primarily addressed the core functionality of Scrum in maintenance. We posted the survey online and requested the practitioners to provide their feedback about the key questions raised in the questionnaire. Results are very encouraging. By analyzing the results we found that product maintenance was a difficult and time consuming but by using Scrum in product maintenance and support now it is very efficient and customer satisfactory.

In future work we are intended to work on Scrum maintenance model for large size projects. We will use different testing techniques in order to get more accurate results which will help us to use this technique in software industry.

References

- [1] V. Rajlich (2014, May). Software evolution and maintenance. In *Proceedings of the on Future of Software Engineering* (pp. 133-144). ACM.
- [2] S. Omanovic & E. Buza (2013, October). Importance of stable velocity in agile maintenance. In *Information, Communication and Automation Technologies (ICAT), 2013 XXIV International Symposium on* (pp. 1-8). IEEE.
- [3] B. Frakes, & K. Kang, (2005). Software reuse research: Status and future. *IEEE transactions on Software Engineering*, 31(7), 529-536.
- [4] L. Lin, W. Yang, & J. Lin, (2012). A layer-based method for rapid software development. *Computers & Mathematics with Applications*, 64(5), 1364-1375.
- [5] O. Sohaib, & K. Khan, (2011). Incorporating discount usability in extreme programming. *International Journal on Software Engineering and its Applications*, 5(1), 51-61.
- [6] J.Lin, C.Lin, & W.Yang, (2013, December). A Method-Based Tool Support for Rapid Software Development and Effective Process Management. In *Software Engineering (WCSE), 2013 Fourth World Congress on* (pp. 115-118). IEEE.
- [7] A. Gerber, A. Van Der, & R. Alberts, (2007). Practical implications of rapid development methodologies. Computer Science and Information Technology Education Conference.
- [8] F. Rousseaux, & K. Lhoste, (2009, February). Rapid Software Prototyping Using Ajax and Google Map API. In *Advances in Computer-Human Interactions, 2009. ACHI'09. Second International Conferences on* (pp. 317-323). IEEE.
- [9] R. Maschotta, S. Jager, T. Jungebloud, & A. Zimmermann, (2013, April). A framework for agile development of simulation-based system design tools. In *Systems Conference (SysCon), 2013 IEEE International* (pp. 861-866). IEEE.
- [10] H. Chehili, L. Seinturier, & M. Boufaida, (2013, August). FASOAD: A framework for agile service-oriented architectures development. In *Database and Expert Systems Applications (DEXA), 2013 24th International Workshop on* (pp. 222-226). IEEE.
- [11] R. Loffler, B. Guldali, & S. Geisen, (2010). Towards Model-based Acceptance Testing for Scrum. *Softwaretechnik-Trends, GI*.
- [12] K. Vlaanderen, S. Jansen, S. Brinkkemper & E. Jaspers (2011). The agile requirements refinery: Applying SCRUM principles to software product management. *Information and Software Technology*, 53(1), 58-70.
- [13] W.Schramm, C. Draeger, & T. Grechenig (2011, September). Issues and mitigation strategies when using agile industrial software development processes in student software engineering projects. In *AFRICON, 2011* (pp. 1-4). IEEE.

- [14] S. Omanovic & E. Buza (2013, October). Importance of stable velocity in agile maintenance. In *Information, Communication and Automation Technologies (ICAT), 2013 XXIV International Symposium on* (pp. 1-8). IEEE.
- [15] T. Khalane & M. Tanner (2013, November). Software quality assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations. In *Adaptive Science and Technology (ICAST), 2013 International Conference on* (pp. 1-6). IEEE.
- [16] V.A. Santos, A. Goldman, A. C. M Shinoda, and A. L. Fischer, (2011, July). A view towards Organizational Learning: An empirical study on Scrum implementation. In *SEKE* (pp. 583-589).
- [17] V. Rajlich (2014, May). Software evolution and maintenance. In *Proceedings of the on Future of Software Engineering* (pp. 133-144). ACM.
- [18] K. Vlaanderen, S. Jansen, S. Brinkkemper & E. Jaspers (2011). The agile requirements refinery: Applying SCRUM principles to software product management. *Information and Software Technology*, 53(1), 58-70.
- [19] S. Omanovic & E. Buza (2013, October). Importance of stable velocity in agile maintenance. In *Information, Communication and Automation Technologies (ICAT), 2013 XXIV International Symposium on* (pp. 1-8). IEEE.
- [20] U.Z. Khan, F. Wahab, and S. Saeed, (2014). Integration of Scrum with Win-Win Requirements Negotiation Model. *Middle-East Journal of Scientific Research*, 19(1), 101-104.
- [21] S. Iqbal, M. Khalid, M. N. A. Khan, "A Distinctive Suite of Performance Metrics for Software Design", *International Journal of Software Engineering & Its Applications*, vol. 7, no. 5, (2013).
- [22] S. Iqbal and M. N. A. Khan, "Yet another Set of Requirement Metrics for Software Projects", *International Journal of Software Engineering & Its Applications*, vol. 6, no. 1, (2012).
- [23] M. Faizan, S. Ulhaq, M. N. A. Khan, "Defect Prevention and Process Improvement Methodology for Outsourced Software Projects", *Middle-East Journal of Scientific Research*, vol. 19, no. 5, (2014), pp. 674-682.
- [24] M. Faizan, M. N.A. Khan, S. Ulhaq, "Contemporary Trends in Defect Prevention", A Survey Report. *International Journal of Modern Education & Computer Science*, vol. 4, no. 3, (2012).
- [25] K. Khan, A. Khan, M. Aamir and M. N. A. Khan, "Quality Assurance Assessment in Global Software Development" *World Applied Sciences Journal*, vol. 24, no. 11, (2013).
- [26] M. Amir, K. Khan, A. Khan, M. N. A. Khan, "An Appraisal of Agile Software Development Process", *International Journal of Advanced Science & Technology*, vol. 58, (2013).
- [27] M. Khan and M. N. A. Khan, "Exploring Query Optimization Techniques in Relational Databases", *International Journal of Database Theory & Application*, vol. 6, no. 3, (2013).
- [28] M. N. A. Khan, M. Khalid and S. UIHaq, "Review of Requirements Management Issues in Software Development", *International Journal of Modern Education & Computer Science*, vol. 5, no. (1), (2013).
- [29] M. Umar and M. N. A. Khan, "A Framework to Separate Non Functional Requirements for System Maintainability", *Kuwait Journal of Science & Engineering*, vol. 39, no. 1 B, (2012), pp. 211- 231.
- [30] M. Umar and M. N. A. Khan, Analyzing Non-Functional Requirements (NFRs) for software development. In *IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS)*, (2011), pp. 675-678).
- [31] M. N. A. Khan, C. R. Chatwin and R. C. Young, "A framework for post-event timeline reconstruction using neural networks", *digital investigation*, vol. 4, no. 3, (2007), pp. 146-157.
- [32] M. N. A. Khan, C. R. Chatwin and R. C. Young, "Extracting Evidence from File system Activity using Bayesian Networks", *International journal of Forensic computer science*, vol. 1, (2007), pp. 50-63.
- [33] M. N. A. Khan, "Performance analysis of Bayesian networks and neural networks in classification of file system activities", *Computers & Security*, vol. 31, no. 4, (2012), pp. 391-401.
- [34] M. Rafique and M. N. A. Khan, "Exploring Static and Live Digital Forensics: Methods, Practices and Tools", *International Journal of Scientific & Engineering Research*, vol. 4, no. 10, (2013), pp. 1048-1056.
- [35] M. S. Bashir and M. N. A. Khan, "Triage in Live Digital Forensic Analysis", *International journal of Forensic Computer Science*, vol. 1, (2013), pp. 35-44.

- [36] A. Sarwar and M. N. A. Khan, "A Review of Trust Aspects in Cloud Computing Security", *International Journal of Cloud Computing and Services Science (IJCLOSER)*, vol. 2, no. 2, (2013), pp. 116-122.
- [37] A. H. Gondal and M. N. A. Khan, "A review of fully automated techniques for brain tumor detection from MR images", *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 5, no. 2, (2013), pp. 55.
- [38] A. Zia and M. N. A. Khan, "Identifying key challenges in performance issues in cloud computing", *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 4, no. 10, (2012), pp. 59.
- [39] K. U. Rehman and M. N. A. Khan, "The Foremost Guidelines for Achieving Higher Ranking in Search Results through Search Engine Optimization", *International Journal of Advanced Science and Technology*, vol. 52, (2013), pp. 101-110.
- [40] M. Khan and M. N. A. Khan, "Exploring query optimization techniques in relational databases", *International Journal of Database Theory & Application*, vol. 6, no. 3, (2013).
- [41] R. Shehzad, M. N. KHAN and M. Naeem, "Integrating knowledge management with business intelligence processes for enhanced organizational learning", *International Journal of Software Engineering and Its Applications*, vol. 7, no. 2, (2013), pp. 83-91.
- [42] S. U. Haq, M. Raza, A. Zia and M. N. A. Khan, "Issues in global software development: A critical review", *An Appraisal of Off-line Signature Verification Techniques 75 Copyright © 2015 MECS I.J. Modern Education and Computer Science*, vol. 4, (2015), pp. 67-75 *Journal of Software Engineering and Applications*, vol. 4, no. 10, (2015), pp. 590.
- [43] A. Zia and M. N. A. Khan, "A Scheme to Reduce Response Time in Cloud Computing Environment", *International Journal of Modern Education and Computer Science (IJMECS)*, vol. 5, no. 6, (2013), pp. 56.
- [44] M. Tariq and M. N. A. Khan, "The Context of Global Software Development: Challenges, Best Practices and Benefits", *Information Management & Business Review*, vol. 3, no. 4, (2011).
- [45] A. Shahzad, M. Hussain and M. N. A. Khan, "Protecting from Zero-Day Malware Attacks", *Middle-East Journal of Scientific Research*, vol. 17, no. 4, (2013), pp. 455-464.
- [46] A. A. Khan and M. Khan, "Internet content regulation framework", *International Journal of U-& EService, Science & Technology*, vol. 4, no. 3, (2011).
- [47] K. Ullah and M. N. A. Khan, "Security and Privacy Issues in Cloud Computing Environment: A Survey Paper", *International Journal of Grid and Distributed Computing*, vol. 7, no. 2, (2014), pp. 89-98.
- [48] A. A. Abbasi, M. N. A. Khan and S. A. Khan, "A Critical Survey of Iris Based Recognition Systems", *Middle-East Journal of Scientific Research*, vol. 15, no. 5, (2013), pp. 663- 668.
- [49] M. N. A. Khan, S. A. Qureshi and N. Riaz, "Gender classification with decision trees", *Int. J. Signal Process. Image Process. Patt. Recog*, vol. 6, (2013), pp. 165-176.
- [50] S. S. Ali and M. N. A. Khan, "ICT Infrastructure Framework for Microfinance Institutions and Banks in Pakistan: An Optimized Approach", *International Journal of Online Marketing (IJOM)*, vol. 3, no. 2, (2013), pp. 75-86.
- [51] A. Mahmood, M. Ibrahim and M. N. A. Khan, "Service Composition in the Context of Service Oriented Architecture", *Middle East Journal of Scientific Research*, vol. 15, no. 11, (2013).
- [52] M. A. Masood and M. N. A. Khan, "Clustering Techniques in Bioinformatics", *I. J. Modern Education and Computer Science*, vol. 1, (2015), pp. 38-46.
- [53] Ur Rehman, T., Khan, M. N. A., & Riaz, N. (2013). Analysis of Requirement Engineering Processes, Tools/Techniques and Methodologies. *International Journal of Information Technology and Computer Science (IJITCS)*, 5(3), 40.
- [54] Ahmed, R., & Khan, M. N. A. (2013). An Analytical Review of Stereovision Techniques to Reconstruct 3D Coordinates. *International Journal of Information Technology and Computer Science (IJITCS)*, 5(7), 80.

Authors' Profiles



Riffat Naz is pursuing for MS degree in Software Engineering at Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan. Her research areas include software engineering, Rapid Application Development, and data mining techniques.



M.N.A. Khan obtained D.Phil. degree from the University of Sussex, UK. His research interests are in the fields of software engineering, cyber administration, digital forensic analysis and machine learning techniques.



Muhammad Aamir obtained MS degree in Software Engineering from SZABIST, Islamabad, Pakistan. He has over seven years experience of working in the IT Industry. His research interests are in the fields of Software Engineering particularly Software Testing, Rapid Application Development, Agile Software Development Methodologies and Scrum.

How to cite this paper: Riffat Naz, M. N. A. Khan, Muhammad Aamir, "Scrum-Based Methodology for Product Maintenance and Support", *International Journal of Engineering and Manufacturing (IJEM)*, Vol.6, No.1, pp.10-27, 2016. DOI: 10.5815/ijem.2016.01.02