

Requirement Elicitation for Open Source Software By using SCRUM and Feature Driven Development

Umber ISMAIL^{1*}Salman QADRI¹Muhammad FAHAD²¹Department of Computer Science and Information Technology, The Islamia University of Bahawalpur, Pakistan²Department of Computer Science, Government College of Commerce, Multan, Pakistan

*Corresponding author:

Email: umber.ismaiel555@yahoo.com

Received: February 12, 2015

Accepted: March 23, 2015

Abstract

During previous years, some movements have been widely discussed in the software development community: Agile and open source development. Both have faced some of the same criticism, and both claim some of the same benefits. This study discusses the question, whether open source software development is in accordance with agile software development principles and therefore well within the planning spectrum? A comparative study is discussed about small scale open source projects by implementing Scrum and Feature Driven Development (FDD). Scrum is useful when requirements were not only incomplete at the start, but also could change rapidly during development phase. Scrum is concerned with functionality, but when we see that there is a functionality collapse; an alternative way is adopted, which is FDD which work upon domain base. In this study we compare that for Open Source Software (OSS), which methodology is good (SCRUM or FDD) for requirement gathering of OSS projects. We will also compare and conclude that which of the two methodologies (SCRUM or FDD) is good for requirements gathering of OSS projects.

INTRODUCTION

Open Source Software (OSS) is a software having its source code open for every person, it work under an open source license GNU/GPL. The process that is used to develop the Open Source Software is called Open Source Software Development (OSSD). OSS have many characteristics like it support dynamic requirement changes. The distributed teams are also a part of the OSSD [1]. The primary focal point of this work is to "relate OSS with SE and use agile methodologies like SCRUM and Feature Driven Development (FDD) to compare the requirement gathering of OSS by using SCRUM and FDD" and check which methodology is more suitable for OSS for small tasks and which approach is having less difficulties for team members during requirement gathering.

This study contains some problems that are faced in both methodologies and give some recommendation to improve these difficulties. SCRUM and Feature Driven Development are types of agile methodologies that support developers to produce a flexible as well as good product that acquire user satisfaction. But these methodologies have not been implemented for OSS because OSS have its own umbrella activities in which requirement gathering is not separately performed because requirements are collected from old release and some new features are added according to the need of current users.

This study will undertake Open Source software projects to indulge with agile methodology with proper requirement gathering phase belonging to Small Scale Projects and observe that Requirements Efficiency is better in which of the two methodologies i.e. SCRUM or FDD, and also which methodology is more useful and cost effective for requirement gathering of Open Source small scale projects and also make OSS to become a part of Software Engineering.

Literature Survey

According to *Budwig et al. (2009)*, comparative study is performed on a case study using User eXperience (UX) and SCRUM, and concluded that SCRUM team take less time to collect requirements than UX team, and SCRUM teams works than UX to solve emergent requirements. *Sunitha (2011)* discuss about the requirement gathering for small projects using agile methods like (SCRUM, FDD, DSDM, XP and ASD) [16]. The main focus on this paper is the use of user stories and INVEST model for requirements gathering. They concluded that SCRUM is better approach for small scale projects than other approaches like XP.

Koch (2004), in his paper discusses about SCRUM and OSS and conducted a comparative study on SCRUM and OSS [10]. He found the results that SCRUM and OSS have many similarities like: teams in both methodologies have highly skilled as well as self-organizing teams, shorter feedback loops and frequently release of applications, close integration and are collaborative with customers and users. Both methods also have some differences like: co-location and personal interaction is main point of SCRUM but not in OSS.

Major issues and challenges faced by Open Source Software are discussed by *Fitzgerald (2004)* in his paper [9]. The important points are as follow:

a. First, the absence of face-to-face communication and interaction in OSS, but the organization or stakeholders want to communicate with development team directly.

b. Increase the risks of code quality is second issue because some negative programmer involved in the process generate harmful code that effect the market values of OSS. This makes negative effect on the efficiency of software as well as hardware.

c. Modularity increases the risks of common coupling that makes it difficult to maintain the software that's why it is impossible to long run the OSS.

d. Documentation, testing, internationalization/localization and field supports are

tedious task in OSS. Because non-technical Free and Open Source Software (FOSS) contributor and user may not fill the gap to the extent originally predicted.

e. Continued support is not provided in FOSS.

f. Interoperability and compatibility problems arise due to many different commercial version of same Open Source product.

Goyal (2007) discussed the usefulness as well as importance of Feature Driven Development and its practices and process model [6]. Key roles of FDD are described those are: Project Manager, Chief Architect, Development Manager, Chief Programmer, Class Owner, Domain Experts, and also some additional roles that may or may not be a part of FDD.

Gillani *et al.* (2014) in their paper introduced the requirement engineering process to precede the software development process in order to act as a decision support system in in Pakistan's budding software industry which usually develops small to medium user centered software [21]. The most appropriate methodology they proposed for such type of projects is the Agile scrum.

MATERIALS AND METHODS

What is OSS?

According to Hertel *et al.*, in the 1960s, the term of open source is used for sharing code because of the absence of viable software in market [14]. In September 1983, Stallman launched a GNU project that was a free UNIX-like platform and this was the initiation of the first free software movement [19]. Nowadays mostly Open Source projects are working under the GNU General Public Licenses that is developed by Stallman [2]. After that Open Source is the place for programmers who make software without any cost [3].

Free software is a social movement whereas open source software development (OSSD) is a software development methodology [18]. According to free software advocates, like Stallman[1] and the Free Software Foundation, "Open Source" Software can be interchangeably used as "Free Software" but there is a little bit change in both terminologies like: Free Software is the software that does not require the permission of licensor to the access, modification, use, copy and distribute of source code and not any payment is paid to licensor for distribution [2,3]. New sites and new kinds of software processes are emerging within OSSD projects, as well as new appearances for development project success, when related to those found in outdated industrial software projects and those showed in software engineering textbooks. Free (as in freedom) software and open source software (OSS) are often considered as the same thing [1]. We can say that, all Open Source Software (OSS) is not cost free and not all cost free software is open source. Instead, F/OSSD (Free / Open Source Software Development) is a different [4], somewhat orthogonal approach to the evolution of software systems where much of the development activity is openly visible, development artifacts are publicly available over the Web, and generally there is no formal project management rule, budget or schedule. "The software, whose source code is freely available to be used, copied and modified also" is called Open Source Software. It is hard to run an open-source project following a more traditional software development method like the waterfall model, because in these traditional methods it is not allowed to go back to a previous phase. In open-source software development, the

requirements are rarely gathered before the start of the project; instead they are based on early releases of the software product [8].

Capturing Requirement in SCRUM

When agile scrum is chosen to handle a task, there are certain techniques to apply at the start of an agile project to meet requirements for the project, set the product release contents, and plan the product releases. The best place to start and to gather the requirements for the product and to group them together into releases is to create a product backlog of user stories [15, 16]. Then product backlog is maintained using that user story. After creating product backlog, the release of the product is planned. Initially, there are so many features and functions that the product could include. So it is best to start with writing informal user stories and then update them with more formalized user stories. Foremost of all, an initial or informal user story would be written that simply trace a product feature or functionality. According to Khramtchenko (2004) [15] a user story is as: "A user story is a small piece of the product functionality written in easy to read sentences". Any project stakeholder can easily compose a user story, including clients, users, sales team members, designers, developers, or testers. A well written user story will describe what the desired functionality is [6], who it is for, and why it is useful? User story is a type of technique that is used to capture the requirements. Meetings can take place for collecting requirement. A typical format for a user story template is: "As I want to so that I can". Agile methods can be very useful for the projects nowadays.

Capturing Requirement in Feature Driven Development (FDD)

Feature Driven Development is an agile software development methodology developed by Jeff De Luca and Peter Code. This methodology got famous in 1997. FDD followers discuss the methodology and processes in the FDD community web site [15]. FDD claims that it achieves the more success in software projects than other agile methodologies. It has more formal requirements and steps than SCRUM, but adds a precise tracking of progress. Feature can be defined as: "a feature can be considered from two aspects that is intension and extension". In intension a feature is: "a cohesive set of individual requirements" and in extension a feature: "It is a user or customer visible capability of a software system" [12]. So in FDD feature is main element of requirement gathering phase.

FDD development consists of the two main phases:

- Discovering list of features of project;
- Implementing each features.

Discovering List of Features

The critical process of FDD is discovering features lists. This process requires all time participation of customers and stakeholders [7]. UML diagrams are developed as aim the result of this process. If the two-way development tool is used, than UML diagrams are assisted by the compile-able code in the target programming language.

Using UML diagrams the features list of projects are extracted in FDD [13]. The features are writings in the language which are both understandable for development community and customers. The UML diagrams may include classes like Employees, Student Information, and Administration. The resulting list of features may include:

- Create new User accounts of Employee.
- Insert the Record of Student.
- List all Students Enrolled.
- Calculate the total Student Strength that appear in Exam;

This list of features presents a value to a sponsor, since it directly mirrors the functionality that will be available in an Open Source College Management System project. Work package contains the group of people working and following the below steps in each iteration.

- Kick-off meeting: this is the first step that holds the details of included features are explained [15].

Building problem domain (Gathering Requirements)

Major Tool that is used for requirement gathering is called UML or color UML version. This phase is called process one. Following steps/iteration are performed in this phase [11]:

User story is starting point of each iteration. That is the resultant of meetings. The major team is sub divided into further sub teams, each team have domain expert as well as developers. UML model of each story is then developed by groups. The teams present and discuss the presented models. When the iteration is ended the best model is chosen or merged. The modeling iterations continues until the full problem domain model satisfies all the stories or requirements mentioned in user stories. Fahad *et al.* (2014) in their paper also discussed the same process of implementing user stories on a hybrid DXPRUM model [20].

RESULTS AND DISCUSSIONS

A presentation of an empirical work in which the software engineering class of computer science department in “The Islamia University Bahawalpur” and some remote people that belong to different areas were interviewed for the research. The study was conducted by collecting data from students developing a Web Based Open Source Software application. There were 2 teams of developers, with each team consisting of few members. Each team received the same problem description from the product owner to build an Open Source College Management Information System (OSCMS). Disciplined Agile-Scrum and Feature Driven Development are the development processes used to carry out the project. FDD and Scrum provide the artifacts for requirement gathering and completing the development phases.

Following Diagram Describe the working methodology of case study execution followed by development teams involved in collecting requirements. Our proposed model is given above. From start, customer and stakeholders communicated with different teams via online community groups. There are two teams that worked on same case study but with different stakeholders, that is OSCMS (Open Source College Management System).Both teams collect requirements to follow the different methodologies. Team one followed the Feature Driven Development Methodology and team 2 worked on Scrum Methodology. In last phase all collected properties that is best is called as OSS Product Properties which is the desired result of our research.

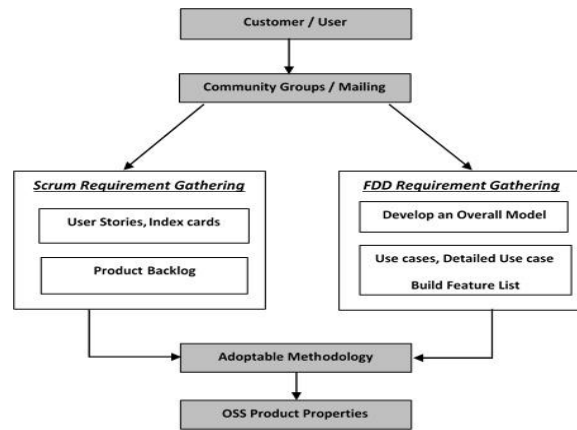


Figure 1. Proposed Model

FDD Requirement gathering

Team that used FDD methodology collects requirements in two sub-phases.

Develop an Overall Model

In this phase, user stories are written in an informal style, then these reports are formed as Use case diagram and detailed use case diagram by team 2 that is adopting the FDD as a growth model.

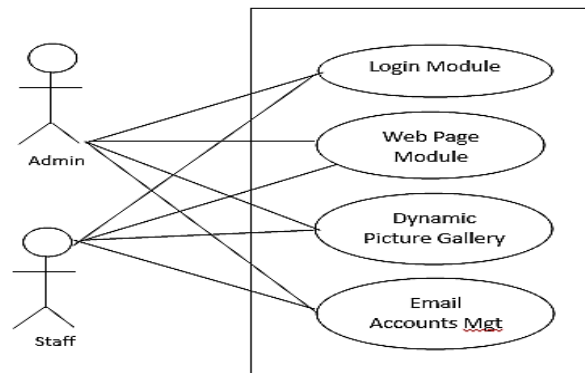


Figure 2. Use Case Diagram of OSCMS

Build Feature list

After producing an overall model of OSCMS, features are extracted from utilizing stories and use cases and a list is produced which contain more than 35 features for above four modules. Some of these features are as, registration/sign up, delete account and insert account, Table 1 Contain a list of some of these features.

SCRUM Requirement gathering

Team 2 follow the SCRUM methodology, first of all meetings are conducted for requirement gathering and planning. SCRUM requirement methodology is divided into two phase that is as follow:

- User Stories, Index Cards Preparing

During first phase, the analyst collect requirements for the four modules of the system. These modules are: Login module, Web Page Module, Dynamic Picture Gallery and email accounts management.

Table 1. Features List in FDD

Sr. No.	Module	Features	Sr. No.	Module	Features
1.	Web page Management Section	Admin Add web page for publishing data	7.	Dynamic Picture Gallery	Admin create category for easy exploring
2.		Admin Edit web page for modification	8.		Admin Upload Pictures for sharing with staff, student
3.		Admin Delete Web page	9.		Admin Delete Pictures from gallery
4.		Add menus for connection of web page	10.		Admin Delete Category
5.		Delete Menus that is not usable	11.		Admin Manage Category for sequencing
6.		Edit Menus for link modification	12.		User navigate pictures by direct click

a. User Stories are collected in following format.
As a faculty member to College, I want to register myself so I require a username and password so that I can use OSCMS.

b. Index Cards, are the template that capture the user stories in some specific form [16],

- Product Backlog Maintenance

User stories capture the features and also have high level functionality of the software. The Product Backlog records all functional requirements and also have the information about all activities performed during the project. Table 2 contains some of the requirement list that are collected by Scrum Team along with their priorities.

Table 2. Requirements specification List in SCRUM

REQ ID	Requirement Description	Priority
1	Sign In Functionality must be provided to Employee of the college to access campus management module	Medium
2	The software and web module should be viewed on mobile devices.	Medium
3.	A website should be contain the Integrated module of Campus Management Solution	High
4	The website should be lightweight and easy to load	High
5	There should be separate sections for displaying information	High
6	UX norms should be followed for website module.	Medium

FINDINGS AND RESULTS

Table 3 shows that both teams have figured out parallel and utilize 4 weeks (1 month) time for requirement gathering. After getting requirements following results of both methodologies are produced. During first week team 1 collected more user stories or properties than team 2, Team 1 collected 18 properties in comparison of 12 properties of team 2.

Table 3. Product Properties Collected in 1 month

Time	Properties	
	FDD	SCRUM
week1	12	18
week2	7	7
week3	4	3
week4	3	4

Property



Figure 3. Product Properties Collected in 1 month

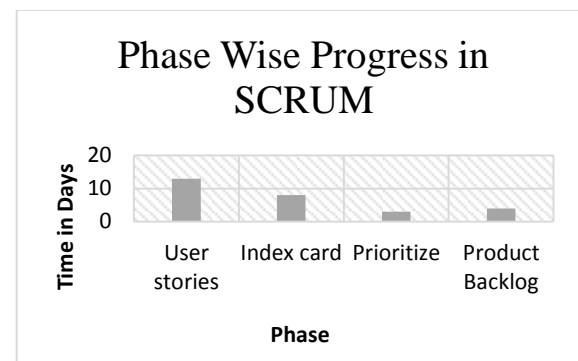


Figure 4. Phase wise progress in SCRUM

Some interview sessions were conducted with the groups for getting results. That questions asked and their responses are given below:

Table 4. Responses of Questions

Sr. No.	Question Asked	Response	
		SCRUM	Feature Driven Development
1.	What is the size of team that work on the project?	4-6	5-10
2.	What is the size of the Project?	Small-Medium	Small-Medium
3.	Which Design Tools you have used to collect Requirements of OSCMS?	Community forum, User Stories, Index cards, Prioritize List	Community Groups, Use Cases, UML Diagrams, Features, Features List Module wise.
4.	Do you think the phase shifting in the methodology is easy?	Phase shifting is easy in it and if any emergent requirement is arise then it may deal it in every stage like Sprint planning to sprint backlog.	Phase shifting is not easy in it and if any emergent requirement is arise then it may deal it according to the its severity. if emergent properties arise during plan by Feature phase then if it is more severe then phase may be shifted back to Develop an Overall Model (DOM).
5.	How you consider the efficiency of team member in forum based interaction?	Team member work efficiently via online forums. user stories, and index cards are used as a good graphical and communication tool between stake holders than other tools like use cases.	Team member coordinate with each other through forums and work efficiently, but, during user stories phase and DOM phase in use case diagrams, its performance is not good as a communication tool.
6.	What is the resultant document that your team produced?	Product Backlog and then sprint backlog	Final Feature List Document

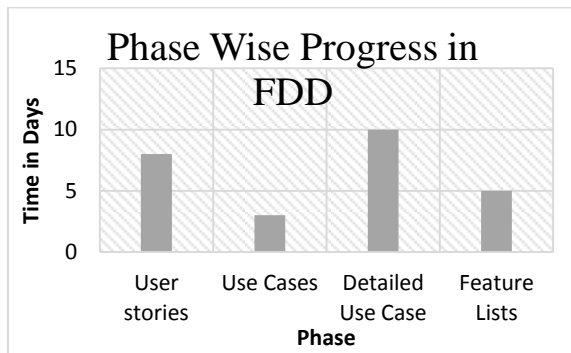


Figure 5. Phase wise progress in FDD

Observation

After observing Table 4 we can say that SCRUM is more feasible than FDD and requires less team member, short term iterations and is more supportive for community groups. Emergent properties are adopted at any stage. Ratio of requirement is much in time slice. FDD will produce good result and quality product, if stakeholders involved have more time, sufficient resources, Domain experts and with size of the project is medium to large.

CONCLUSION

The conclusion of this study is as follow:

SCRUM and FDD belongs to agile spectrum, so the major focusing point of both methodologies are people, if we use agile methodology to develop Open Source Software then it's a good way to collect requirements to use Agile Methodology especially SCRUM, because it has many processes which are used to extract requirement more precisely for small scale projects because small teams are needed in this methodology [16].

FDD and SCRUM have more ways of collaboration, many meetings were conducted for collaboration purpose.

Due to this communication, information exchange is very easy. So for this purpose community groups can be created or to provide this functionality, mailing lists is also an effective tool for OSS [17].

SCRUM accepts changes in requirement at any stage of development process while FDD divide changes according to its severity. Due to community groups and mailing lists functionality SCRUM accepts changes in requirement at any stage but FDD Team get more time than SCRUM

Tasks priorities are set in both approaches but different roles perform this task. In FDD domain experts perform task/feature prioritization and in SCRUM, scrum master perform this task [16, 17].

In both methodologies changes are handled at different phases In Scrum changes are handled at the time of backlog maintenance in sprints, while FDD categorize changes in two phases. At first phase minor changes are handled by creating feature lists, while major changes are handled at domain modeling phase in FDD.

Customer interaction is allowed in both methodologies. In Scrum, at every phase the customer interacts with teams through mailing lists like: product backlog maintenance and in Sprint backlog, while on other hand interaction is done at initial phase of FDD [18, 17]. In SCRUM customer interact and ask about emergent change in requirements of OSS at any phase but in FDD team member first checked the nature of change then changes will be made in Develop an Overall Model and so on.

Future Work

Agile Methodologies have many phases. Requirement gathering is effective for OSS development and is a new addition in OSS traditional development methodologies. Future work should be based on designing of OSS using SCRUM and FDD methodologies. And to test which methodology (SCRUM or FDD) is best for designing of OSS.

REFERENCES

- [1] G.F. Murray, M.A. Duncheon, & R. Llp. 2006. "Free and Open Source Software : An Introduction".
- [2] H. Works, "Introduction to Open Source Software : Introduction to Open Source Software."
- [3] A. Khanjani and R. Sulaiman. March 2011. "The aspects of choosing open source versus closed source," In 2011 IEEE Symposium on Computers & Informatics (ISCI), pp. 646–649.
- [4] E. Y. Nakagawa, E. P. M. de Sousa, K. de Brito Murata, G. de Faria Andery, L. B. Morelli, and J. C. Maldonado. 2008. "Software Architecture Relevance in Open Source Software Evolution: A Case Study," In 32nd Annual IEEE Int. Conf. on Computer Software and Application 2008 (COMPSAC'08), pp. 1234–1239.
- [5] M. N. Huda. 2011. "Comparison between Adaptive Software Development and Feature Driven Development". In Proc. of IEEE Int. Conf. on Computer Science and Network Technology (ICCSNT'11), 1: 363–367.
- [6] S. Goyal. 2008. "Major Seminar on Feature Driven Development". Jennifer Schiller Chair of Applied Software Engineering.
- [7] J. Wusteman. 1997. "OJAX : A Case Study in Agile Web 2. 0 Open Source Development OJAX : A Case Study in Agile Web 2. 0 Open Source Development". pp. 1–24.
- [8] B. Paech and B. Reuschenbach. 2006. "Open Source Requirements Engineering" In Proc. of 14th IEEE Int. Conf. on Requirements Engineering, pp. 257–262.
- [9] B. Fitzgerald. 2001. "IT SYSTEMS PERSPECTIVES A Critical Look". 37(7): 92–94.
- [10] S. Koch. 2004. "Agile Principles and Open Source Software Development: A Theoretical and Empirical Discussion". In Extreme Programming and Agile Processes in Software Engineering, Springer Berlin Heidelberg, pp. 85–93.
- [11] P. Coad, J. de Luca and E. Lefebvre. 1999. "Java Modeling In Color With UML: Enterprise Components and Process".
- [12] W. Zhang, H. Mei, and H. ZhaoJun. 2006. "Feature-driven requirement dependency analysis and high-level software design". Requirements Engineering, 11(3): 205–220.
- [13] M. T. Sletholt, J. Hannay, D. Pfahl, H. C. Benestad, and H. P. Langtangen. 2011. "A literature review of agile practices and their effects in scientific software development". In Proc. of 4th ACM Int. Workshop on Software Engineering for Computational Science and Engineering (SECSE '11), pp. 1-9.
- [14] G. Hertel, S. Niedner, and S. Herrmann. 2003. "Motivation of software developers in open source projects: An Internet-based survey of contributors to the Linux kernel". Research Policy, 32(7): 1159–1177.
- [15] S. Khramtchenko. 2004. "Comparing eXtreme Programming and Feature Driven Development in academic and regulated environments". Feature Driven Development.
- [16] C. R. Kavitha and S. M. Thomas. 2011. "Requirement Gathering for small Projects using Agile Methods". In IJCA Special Issue on Computational Science - New Dimensions & Perspectives (NCCSE'11), pp. 122–128.
- [17] X. Ge, R. F. Paige, F. A. C. Polack, H. Chivers and P. J. Brooke. 2006. "Agile Development of Secure Web Applications". In Proc. of the 6th ACM Int. conf. on Web engineering, pp. 305–312.
- [18] W. Scacchi. 2006. "Understanding Open Source Software Evolution". Software Evolution and Feedback: Theory and Practice, pp. 181-206.
- [19] Richard Stallman - Wikipedia, the free Encyclopedia. "Richard Stallman". wikipedia.com. [Online]. Available: http://en.wikipedia.org/wiki/Richard_Stallman [Accessed: Mar. 27, 2013].
- [20] M. Fahad, S. Qadri, S. S. Muhammad, M. Husnain. 2014. "Software Quality Assurance of Medium Scale Projects by using DXPRUM Methodology". In International Journal of Natural and Engineering Sciences (IJNES'14). (Turkey), 8(1): 42-48. ISSN: 1307-1149.
- [21] S. M. Gillani, S. Qadri, M. Fahad. 2014. "Customer Oriented Requirement Engineering by using Scrum Methodology". In International Journal of Natural and Engineering Sciences (IJNES'14). (Turkey), 8(3): 07-11. ISSN: 1307-1149.