ISSN No. 0976-5697

Volume 4, No. 3, March 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Collaborative Model of SCM in Scrum

Ritu Sharma
M.Tech Scholar, Software Engineering
Pacific Institute of Technology, Pacific University
Udaipur, India
riitusharma@gmail.com

Shilpi Sharma
M.Tech Scholar, Software Engineering
Pacific Institute of Technology, Pacific University
Udaipur, India
ershilpisharma22@gmail.com

*Abstract***:** Software configuration management (SCM) is known as a method of bringing control to the software development process, and thus, proper application of SCM is a key component in the development of quality software. Scrum is becoming a popular project management method for agile software development. Scrum is more like a framework and says nothing about how to develop software; instead it's about how to manage the development process. In development of software SCM is mentioned in Scrum explicitly. Hence, in this paper proposing SCM implicitly in Scrum and to see how much is it possible feasible.

Keywords: Scrum, SCM, agile, implicitly, explicitly

I. INTRODUCTION

SCM also defined as "the art of identifying, organizing, and controlling modifications to the software being built by a programming team. It maximizes productivity by minimizing mistakes."

Software configuration management (SCM) is the practice of controlling and managing software components throughout the software development lifecycle. The goal of SCM is to provide effective control over the development products that comprise the software application and to provide the ability to accurately reproduce the state of the software at any point in time."

Scrum is an iterative and incremental agile software development method for managing software projects and product or application development. Scrum has not only reinforced the interest in project management, also challenged the conventional ideas about such management. Scrum focuses on project management institutions where it is difficult to plan ahead. Mechanisms of empirical process control, where feedback loops that constitute the core management technique are used as opposed to traditional command-and-control oriented management.

II. SOFTWARE CONFIGURATION MANAGEMENT

A. *SCM activities:*

a. Configuration Identification:

Configuration Identification is the activity where a system is divided into uniquely identifiable components, called Configuration Items (CI), for the purpose of software configuration management.

b. Configuration Control:

Software requirements can change often and quickly, and the development team must be able to respond to these changes in order to satisfy the customer. However, if changes are implemented in an uncontrolled manner, it will sooner or later lead to chaos

c. Configuration Status Accounting:

The aim of Configuration Status Accounting (CSA) is to keep managers, users, developers, and other project stakeholders informed about the various configuration stages and their evolution.

d. Configuration Audit:

The objective of the Configuration Verification and Audits is to verify that the software system matches the configuration item description and that the system is complete.

B. *SCM Roles:*

a. Configuration Manager:

Responsible for identifying configuration items. Also often responsible for defining the procedures for creating promotions and releases.

b. Change Control Board Member:

Responsible for approving or rejecting change requests.

c. Developer:

Creates promotions triggered by change requests or the normal activities of development. The developer checks in changes and resolves conflicts.

d. Auditor:

Responsible for the selection and evaluation of promotions for release and for ensuring the consistency and completeness of this release.

III. SCRUM

A. Sprint:

A Sprint is a time boxed of 30 sequential calendar days during which a team works to turn the Selected Product Backlog into an increment of potentially shippable product.

B. Three core roles:

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files.

a. Product Owner:

The Product Owner represents the stakeholders and is the voice of the customer. He or she is accountable for ensuring that the team delivers value to the business. The Product Owner writes customer-centric items (typically user stories), prioritizes them, and adds them to the product backlog.

b. Development Team:

The Development Team is responsible for delivering potentially shippable product increments at the end of each Sprint.

c. Scrum Master:

Scrum is facilitated by a Scrum Master, sometimes written as Scrum Master, who is accountable for removing impediments to the ability of the team to deliver the sprint goal/deliverables.

C. Scrum meetings:

a. Sprint Planning:

Sprint Planning refines the Product Backlog items assigned to the sprint into more granular Sprint Backlog items.

b. Daily Scrum:

Each work day the Scrum Team gathers in the same place at the same time. The Scrum Master asks each person 3 questions, in turn:

a) What user stories need to be done (to do)?
b) What user stories are in progress (in progress)?
c) What user stories are done (done)?

c. Sprint Review:

Sprint Review meeting is done at the end of the Sprint, usually on the last sprint day. During this meeting, all committed user stories are presented to Product Owner. Team should "Show" software working. It could be managed by one person or by different team members. Each user story is shown at a time, product owner decides if the story is meeting the expectations or it's not.

At the end of the Showcase, team is able to get velocity from the stories accepted by product. For the stories not accepted they will pass once again to Product Backlog to be reconsidered in a further sprint.

d. Sprint Retrospective

The sprint retrospective meeting is held at the end of every sprint after the sprint review meeting. The team and Scrum Master meet to discuss what went well and what to improve in the next sprint. The product owner does not attend this meeting.

D. Impediments:

Anything that prevents a team member from performing work as efficiently as possible is an impediment. Each team member has an opportunity to announce impediments during the daily Scrum meeting.

E. Product Burndown Chart:

In Scrum, the product burn down chart is a "big picture" view of a project's progress. It shows how much work was left to do at the beginning of each sprint. The scope of this chart spans releases; however, a release burn down chart is limited to a single release.

F. Sprint Burndown Charts:

A sprint burn down chart depicts the total task hours remaining per day. This shows you where your team stands regarding completing the tasks that comprise the product backlog items that achieve the goals of the sprint. .

G. Release:

The transition of an increment of potentially shippable product from the development team into routine use by customers. Releases typically happen when one or more sprints has resulted in the product having enough value to outweigh the cost to deploy it.

H. Release Burndown Chart:

In Scrum, the release burn down chart is a "big picture" view of a release's progress. It shows how much work was left to do at the beginning of each sprint comprising a single release. The scope of this chart is a single release; however, a product burn down chart spans all releases.

I. Sprint Metrics:

Following metrics can be used for Scrum:

a. Annual Stories completed vs Committed Stories
b. Retrospective process improvement
c. Team velocity
d. Technical debt management and few other metrics

J. Velocity:

In Scrum, velocity is how much product backlog effort a team can handle in one sprint. This can be estimated by viewing previous sprints, assuming the team composition and sprint duration are kept constant. It can also be established on a sprint-by-sprint basis, using commitment-based planning.

Once established, velocity can be used to plan projects and forecast release and product completion dates.

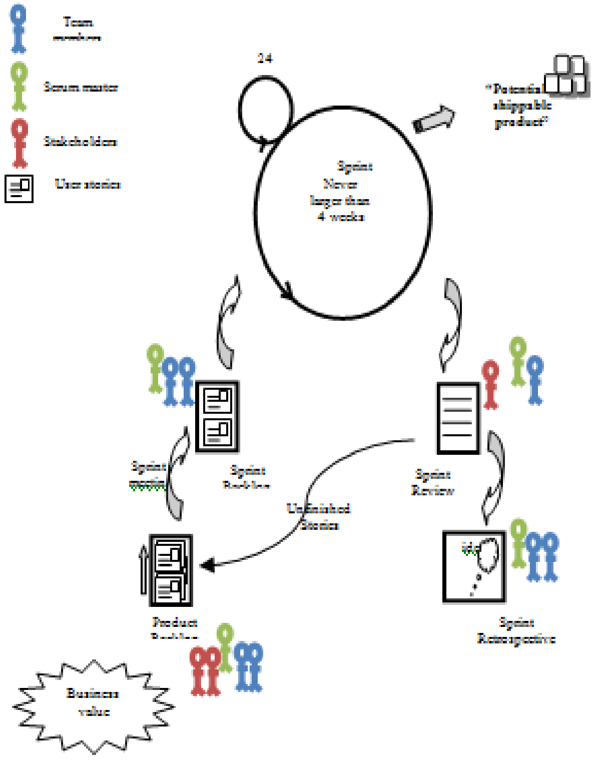

Figure 1: Scrum framework

K. *Product Backlog:*

The Product Backlog lists any required deliverables. Contents are ordered by business value. Backlog Item priority might change, requirements can be added and removed - thus the Product Backlog is a continuously maintained plan towards a growing business value.

L. *Sprint Backlog:*

The Sprint Backlog consists of the necessary tasks to realize the final accepted a version of the Selected Product Backlog. The team uses it to synchronize its activities. The tasks turn the selected product backlog into working product functionality.

M. *Increment:*

The increment is the sum of all the Product Backlog Items completed during a sprint and all previous sprints. At the end of a sprint, the Increment must be done according to the Scrum Team's definition of done. The increment must be in useable condition regardless of whether the Product Owner decides to actually release it.

IV. COLLABORATIVE MODEL

A. *Need of Collaborative Model:*

a. There is a lot of code that need to be managed.
b. It is difficult to manage the conflicts on the revisions.
c. Developers find it too formal to work with SCM practices.
d. Sometimes, developers see management practices a waste of time, the time in which a lot of code can be written.
e. It is not always possible to have a configuration manager in every project.
f. Respond to change rather than trying to prevent it.

B. *SCM Activities in Scrum:*

a. Configuration Identification

The Configuration Identification serves as the starting SCM function in a Scrum project. It will not be as heavy as it is in waterfall and other models. The artifacts that add business value are made Configuration Items.

a) Configuration Control:

The change requests can be placed in the Product Backlog where these are discussed, evaluated and prioritized. Sometimes, the sprint retrospective also ends with the change requests.

b) Configuration Status Accounting:

The status accounting involves answering of the following questions:

(a). List of the versions of CIs?
(b). What and who made the changes in CIs?
(c). How many CIs are incomplete?

c) Configuration Audit:

Both the physical- and functional configuration audits can be done in an informal and face-to-face manner.

C. *SCM roles in Scrum:*

a. Product Owner acts as Configuration Manager: In SCM Configuration Item Owner, Configuration Librarian and Configuration Auditor perform their individual roles in project development, but these roles are not disturbed to different persons in scrum as it is short duration methodology so the roles can be performed by the product owner.
b. Scrum Master as partial Project Manager: The role of the project manager encompasses many activities which sometimes it played by scrum master and sometimes product owner. The Planning and Defining Scope, developing schedules, time estimating, creating charts, etc.

- The Scrum team is a cross functional team and indulges the roles of the developers, testers etc.

D. *Version Control System in Scrum:*

Implementation of this system in Scrum makes the development less confusing. Thus, an approach can be development where with the task board a similar board or board view can be adopted. The board can be developed showing which Configuration items are in use.

a. Release Management in Scrum:

Alignment of development sprints with release management sprints would be ahead off 1 sprint.

b. Workspace Management in Scrum:

Develop without disturbing other developers.

c. Build Management in Scrum:

The Scrum Team should be cross-functional and everyone should know how to perform private build, stable builds, integration builds, formal release builds etc.

d. Change Management in Scrum:

Keep traceability between Sprints and User Stories, i.e. release-notes.

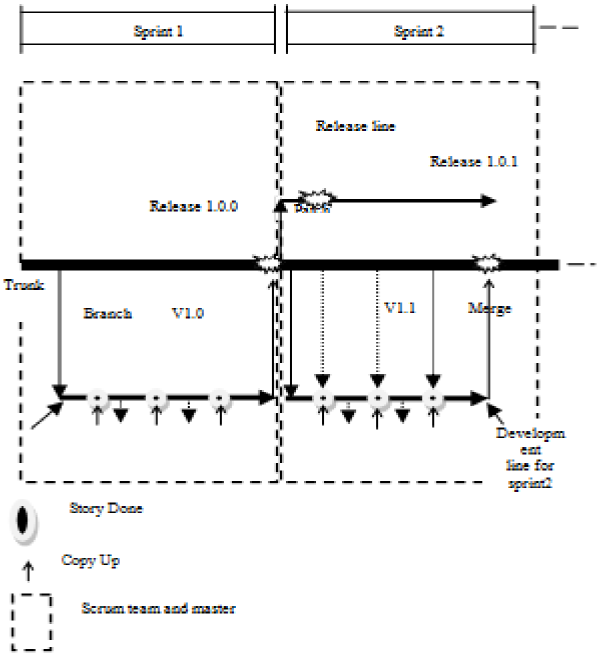

Figure 2: Explanatory model

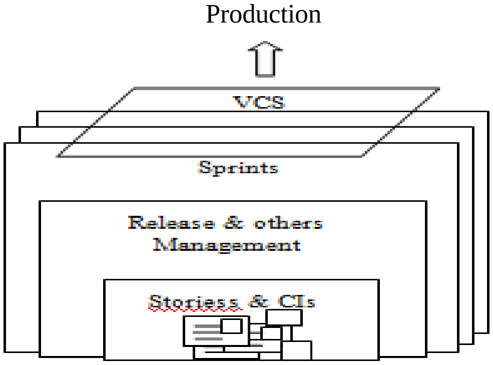

Figure 3: Collaborative model

V. SCRUM WITH XP PRACTICES

Scrum is a project management methodologies which does not focus on any development methodologies. Among many development methodologies extreme programming (XP) can be jelled closely with Scrum. The traces of XP practices can polish the Scrum to a little extend.

a) Pair Programming fits the Scrum team size constraint
b) In XP system design is made simple, Scrum demands fast and simple backlogs and sprints.
c) Continuous integration by the SCM and development toolsets.
d) Refactoring makes incremental improvement for new functionalities.

VI. CONCLUSION

This paper has depicted the enhancing of the overall productivity of Scrum by collaborating software configuration management with Scrum. SCM is not explicitly mentioned in the Scrum. Too much integrating of either SCM or XP can distort the Scrum purpose of existence but a little trace of these can be beneficial. This paper presents a raw and rough basic level model on implicitly mentioning a very few SCM benefits in Scrum without overburdening Scrum with unnecessary SCM activities.

VII. REFERENCES

[1] Stephen P. Berczuk, Brad Appleton,” Software Configuration Management Patterns: Effective Teamwork, Practical Integration”, Addison-Wesley Professional, 2003
[2] Juha Koskela, ”Software configuration management in agile methods”
[3] Andreas Bergstrom,”Software Configuration in SCM projects”, June 2008
[4] Andreas Back, Ola Bodin: Agile Software Configuration Management, Spring 2006.
[5] Kenneth S. Rubin,” Essential Scrum: A Practical Guide to the Most Popular Agile Process”, Addison-Wesley Professional, 26-Jul-2012
[6] Mario E. Moreira,” Adapting Configuration Management for Agile Teams Balancing Sustainability and Speed, John Wiley & Sons, 15-Apr-2010
[7] Henrik Kniberg on Mar 31, 2008, Version Control for Multiple Agile Teams
[8] Bill heys,Branching for Scrum,18 Jan 2011
[9] Jeff Sutherland, The Scrum Papers:Nut, Bolts, and Origins of an Agile Framework. Version 1.1 – 2 Apr 2012
[10] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=EDP242&contentType=Education+%26+Learning&queryText%3Dsoftware+configuration+management
[11] Jessica Keyes, Software Configuration Management, CRC Press, 01-Feb-2004
[12] Jeff Sutherland, Scrum Handbook
[13] K. Schwaber and J. Sutherland. The Scrum Guide. Scrum.org, 2010.
[14] http://c2.com/cgi/wiki?ScrumProcess
[15] www.scrumalliance.org

CONFERENCE PAPER
II International Conference on
“Advance Computing and Creating Entrepreneurs (ACCE2013)”
On 19-20 Feb 2013
Organized by
2nd SIG-WNs, Div IV & Udaipur Chapter , CSI , IEEE Computer Society Chapter India Council ,
IEEE Student Chapter Geetanjali Institute of Technical Studies, Udaipur, Rajasthan, India