International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 4

A Modern Review on Scrum: Advance Project Management Method

Suvarna Shinde
PG Student, Affiliated Department of (MCA), Modern college of Engineering, Pune, India

Pratibha Adkar
Professor, Affiliated Department of (MCA), Modern college of Engineering, Pune, India

ABSTRACT

Scrum method is an agile management method approach toward software development as it progresses incrementally and repetitively. The scrum method involves constant evaluation and revaluation of the progression of the project, to insure it is completed on time, while meeting the specific needs as directed by the product owner. Scrum unique to other agile methods in that it provides an empirical chart to track a product's progression through all stages of its development. This paper includes methods in agile testing, Traditional methods in project management. After this Why to use Scrum and Scrum Framework discussed in detail. Brief description of Tools and Techniques of Scrum is given and last advantages and disadvantages of Scrum are explained.

Keywords: *Scrum; agile; framework; roles; principles; techniques*

I. INTRODUCTION

A method is a model, which project managers employ for the design, planning, implementation and achievement of their project objectives. There are different project management methods to benefit different projects.

A. Agile

Agile software development refers to a group of software development methods based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. Agile methods or agile processes generally promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals. .Agile is a software development method to build software incrementally using short iterations of 1 to 4 weeks so that the development is aligned with the changing business needs. There are various methods present in agile testing, and those are listed below:

- ***Scrum****:* SCRUM is an agile development method which concentrates specifically on how to manage tasks within a team-based development environment. Basically, Scrum is derived from activity that occurs during a rugby match. Scrum believes in empowering the development team and advocates working in small teams (say- 7 to 9 members).

- ***Extreme Programming (XP):*** Extreme Programming technique is very helpful when there is constantly changing demands or requirements from the customers or when they are not sure about the functionality of the system. It advocates frequent "releases" of the product in short development cycles, which inherently improves the productivity of the system and also introduces a checkpoint where any customer requirements can be easily implemented. The XP develops software keeping customer in the target.

- ***Dynamic Software Development Method (DSDM):*** DSDM is a Rapid Application Development (RAD) approach to software development and provides an agile project delivery framework. The important aspect of DSDM is that the users are required to be involved actively, and the teams are given the power to make decisions. Frequent delivery of product becomes the active focus with DSDM.

- *Kanban:*Kanban originally emerged from Japanese word that means, a card containing all the information needed to be done on the product at each stage along its path to completion. This framework or method is quite adopted in software testing method especially in agile testing.

B. Scrum

Scrum is one of the most popular agile methods. It is an adaptive, iterative, fast, flexible, and effective method designed to deliver significant value quickly and throughout a project. Scrum ensures transparency in communication and creates an environment of collective accountability and continuous progress. The Scrum framework is structured in such a way that it supports product and service development in all types of industries and in any type of project, irrespective of its complexity. A Scrum project involves a collaborative effort to create a new product, service, or other result as defined in the Project Vision Statement. Projects are impacted by constraints of time, cost, scope, quality, resources, organizational capabilities, and other limitations that make them difficult to plan, execute, manage, and ultimately succeed. However, successful implementation of the results of a finished project provides significant business benefits to an organization. It is therefore important for organizations to select and practice an appropriate project management method.

II. LITERATURE SURVEY

A. Traditional methods in project management

- ***Waterfall****:* The Waterfall Method, on the other hand, is a traditional approach to project management and more commonly used in the manufacturing or construction sectors. A lot of experts believe that it was the first model to have been adopted in software engineering. The model takes a linear approach towards project management with the project being broken down into sequences with the kick-off of a phase dependent on the completion of the preceding one.

- ***V- Model****:* It is an extension of the waterfall model, Instead of moving down in a linear way, the process steps are bent upwards after the implementation and coding phase, to form the typical V shape. The major difference betweenV-model and waterfall model is the early test planning in the V- model.

- ***Prototyping Model****:* It refers to the activity of creating prototypes of software applications, for example, incomplete versions of the software program being developed. It is an activity that can occur in software development. It used to visualize some component of the software to limit the gap of misunderstanding the customer requirements by the development team. This also will reduce the iterations may occur in waterfall approach and hard to be implemented due to the inflexibility of the waterfall approach. So, when the final prototype is developed, the requirement is considered to be frozen.

- ***Spiral Model (SDM)****:* It is combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. This model of development combines the features of the prototyping model and the waterfall model. The spiral model is favored for large, expensive, and complicated projects. This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk assessment, and the building of prototypes and simulations.

B. Why used Scrum?

The Traditional Waterfall method reveals a more lengthy process where planning alone could take a couple of months before moving to the next stage – design. The design phase could also take a couple of months; this could lead to the launch of a product that could be termed obsolete in the current marketplace.[1]With Scrum, however, the planning is just enough to kick off the project as it is based on the

Agile framework. It's a great way to prevent delays in product launch because the entire process focuses on team collaboration. The Scrum master facilitates the scrum sessions (sprints) which occur within a time frame of 1-3 weeks. The result is an iterative process that significantly saves the company a lot of time and money.

Table 1

Points	**Scrum**	**Traditional Project Management**
Emphasis is on	People	Processes
Documentation	Minimal—only as required	Comprehensive
Process style	Iterative	Linear
Upfront planning	Low	High
Prioritization of Requirements	Based on business value and regularly updated	Fixed in the Project Plan
Quality assurance	Customer centric	Process centric
Organization	Self-Organized	Managed
Management style	Decentralized	Centralized
Change	Updates to Productized Product Backlog	Formal Change Management System
Leadership	Collaborative, Servant Leadership	Command and control
Performance measurement	Business value	Plan conformity
Return on Investment (ROI)	Early/throughout project life	End of project life
Customer involvement	High throughout the project	Varies depending on the project Lifecycle

Scrum vs. Traditional Project Management

III. SCRUM FRAMEWORK

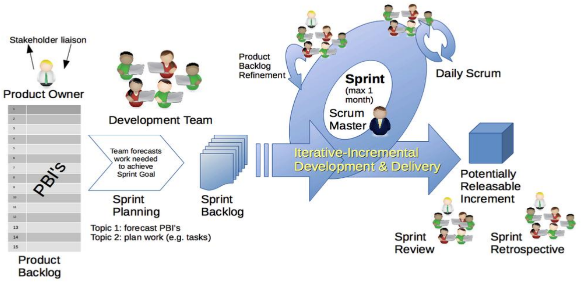


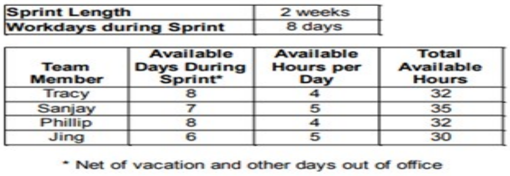
Fig. 1: Framework of scrum method

A. Sprint

Sprint (or iteration) is the basic unit of development in Scrum. The sprint is a time boxed effort; that is, it is restricted to a specific duration. The duration is fixed in advance for each sprint and is normally between one week and one month, with two weeks being the most common.

B. Sprint Planning

At the beginning of each Sprint, the Sprint Planning Meeting takes place. It is divided into two distinct sub-meetings, the first of which is called Sprint Planning Part One The Sprint Planning Meeting will often last a number of hours, but no more than eight hours for a four-week Sprint – the Team is making a serious commitment to complete the work, and this commitment requires careful thought to be successful. The Team will probably begin the Sprint Planning Part Two by estimating how much time each member has for Sprint-related work – in other words, their average workday minus the time they spend attending meetings, doing email, taking lunch breaks, and so on. For most people this works out to 4-6 hours of time per day available for Sprint-related work. This is the team's capacity for the upcoming Sprint.



Sprint Length	2 weeks
Workdays during Sprint	8 days

Team Member	Available Days During Sprint*	Available Hours per Day	Total Available Hours
Tracy	8	4	32
Sanjay	7	5	35
Phillip	8	4	32
Jing	6	5	30

* Net of vacation and other days out of office

Fig. 2: Estimating available hours

C. Daily scrum

Fig. 3: Daily Stand-up meeting

A daily scrum is in the computing room. This centralized location helps the team start on time. Each day during a sprint, the team holds a daily scrum (or stand-up) with specific guidelines. During the daily scrum, each team member typically answers three questions:

- What did I complete yesterday that contributed to the team meeting our sprint goal?
- What do I plan to complete today to contribute to the team meeting our sprint goal?

D. Sprint review and retrospective

At the end of a sprint, the team holds two events: the sprint review and the sprint retrospective.

Guidelines for sprint reviews:

- Incomplete work cannot be demonstrated
- The recommended duration is two hours for a two-week sprint (pro-rata for other sprint durations)

Guidelines for sprint retrospectives:

- Two main questions are asked in the sprint retrospective: What went well during the sprint? What could be improved in the next sprint?
- The recommended duration is one-and-a-half hours for a two-week sprint (pro-rata for other sprint durations)
- This event is facilitated by the scrum master

E. Extensions

The following activities are commonly done, although not considered by all as a core part of Scrum:

- *Backlog refinement:* Backlog refinement (once called backlog grooming) is the on-going process of reviewing product backlog items and checking that they are appropriately prioritised and prepared in a way that makes them clear and executable for teams once they enter sprints via the sprint planning activity. Product backlog items may be broken into multiple smaller ones; acceptance criteria may be clarified; and dependencies, investigation, and preparatory work may be identified and agreed as technical spikes.
- *Cancelling a sprint:* The product owner can cancel a sprint if necessary the product owner may do so with input from the team, scrum master or management. For instance, management may wish the product owner to cancel a sprint if external circumstances negate the value of the sprint goal. If a sprint is abnormally terminated, the next step is to conduct a new sprint planning, where the reason for the termination is reviewed.

F. Product backlog

The product backlog comprises an ordered list of requirements that a scrum team maintains for a product. It consists of features, bug fixes, non-functional requirements, etc.—whatever must be done to successfully deliver a viable product. The product owner prioritizes those product backlog items (PBIs) based onconsiderations such as risk, business value, dependencies, size, and date needed.

G. Sprint backlog

The sprint backlog is the list of work the development team must address during the next sprint The list is derived by the scrum team progressively selecting product backlog items in priority order from the top of the product backlog until they feel they have enough work to fill the sprint. The development team should keep in mind its past performance assessing its capacity for the new sprint, and use this as a guide line of how much 'effort' they can complete.

H. Product increment

The increment (or potentially shippable increment, PSI) is the sum of all the product backlog items completed during a sprint, integrated with the work of all previous sprints. At the end of a sprint, the increment must be complete, according to the scrum team's definition of done (DoD), fully functioning, and in a usable condition regardless of whether the product owner decides to actually release it.

IV. SCRUM ROLES

Scrum development efforts consist of one or more Scrum teams, each made up of three Scrum roles: Product owner, Scrum Master, and the Development team.

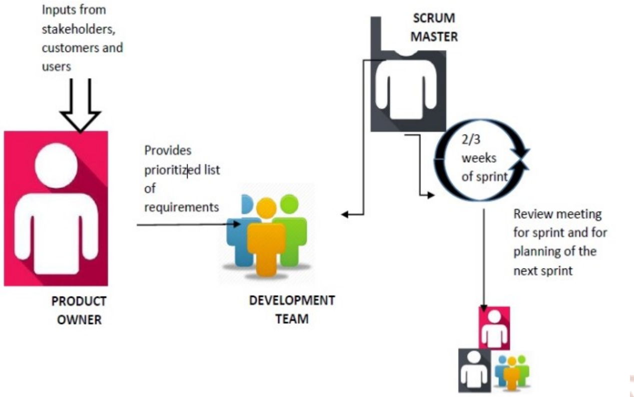

Inputs from stakeholders, customers and users
SCRUM MASTER
Provides prioritized list of requirements
2/3 weeks of sprint
Review meeting for sprint and for planning of the next sprint
PRODUCT OWNER
DEVELOPMENT TEAM

Fig. 4: Scrum roles and their relationship

A. Product Owner

- is the empowered central point of product leadership
- is the only authority responsible for what will be developed and in what order
- he maintsains and communicates to all other participants a clear vision of what the Scrum team is trying to achieve
- as such, the product owner is responsible for the overall success of the solution being developed or maintained.

B. Scrum Master

- helps everyone involved understand and embrace the Scrum values, principles, and practices
- acts as a coach, providing development process leadership
- as a facilitator, Scrum Master helps the team resolve issues and make improvements to its use of Scrum
- is responsible for protecting the team from outside interference
- takes a leadership role in removing impediments that inhibit team productivity
- Has no authority to exert control over the team, so this role is not the same as the traditional role of project manager or development manager. The Scrum-Master functions as a leader, not a manager.

C. Development Team

Traditional software development consists of various job types, such as architect, programmer, tester, database administrator, UI designer, etc. Scrum defines a development team as a diverse, cross-functional collection of people who are responsible for designing, building, and testing the desired product.

- the development team self-organizes to determine the best way to accomplish the goal set out by the product owner
- is typically five to nine people in size; its members must collectively have all skills needed to produce good quality, working software.

V. PRINCIPLES OF SCUM METHOD

The principles of Scrum can be applied to any type of project or organization, and they must be adhered to in order to ensure appropriate application of Scrum.

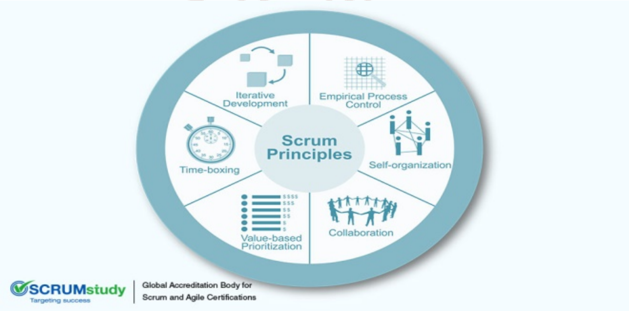

Scrum Principles
Iterative Development
Empirical Process Control
Self-organization
Collaboration
Value-based Prioritization
Time-boxing
SCRUMstudy
Targeting success
Global Accreditation Body for Scrum and Agile Certifications

Fig 5: Principles of scum method

The aspects and processes of Scrum can be modified to meet the requirements of the project, or the organization using it, but Scrum principles are non-negotiable and must be applied as described in the framework presented in A Guide to the Scrum Body of Knowledge the SBOK™ Guide. Keeping the principles intact and using them appropriately instils confidence in the Scrum framework with regard to attaining the objectives of the project.

A. Empirical Process Control

This principle emphasizes the core philosophy of Scrum based on the three main ideas of transparency, inspection, and adaptation.

B. Self-organization

This principle focuses on today's workers, who deliver significantly greater value when self-organized, and these results in better team buy-in and shared ownership; and an innovative and creative environment which is more conducive for growth.

C. Collaboration

This principle focuses on the three core dimensions related to collaborative work: awareness, articulation, and appropriation. It also advocates project management as a shared value-creation process with teams working and interacting together to deliver the greatest value.

D. Value-based Prioritization

This principle highlights the focus of Scrum to deliver maximum business value, from early in the project and continuing throughout.

E. Time-boxing

This principle describes how time is considered a limiting constraint in Scrum, and used to help effectively manage project planning and execution. Time-boxed elements in Scrum include Sprints, Daily Stand-up Meetings, Sprint Planning Meetings, and Sprint Review Meetings.

F. Iterative Development

This principle defines iterative development and emphasizes how to better manage changes and build products that satisfy customer needs. It also delineates the Product Owner's and organization's responsibilities related to iterative development.

Scrum principles are the core guidelines for applying the Scrum framework and should mandatorily be used in all Scrum projects. The Scrum aspects and processes, however, can be modified to meet the requirements of the project or the organization.

VI. TOOLS AND TECHNIQUES IN SCRUM

A. Burndown chart

A burndown chart is a graphical representation of work left to do versus time. The outstanding work (or backlog) is often on the vertical axis, with time along the horizontal. That is, it is a run chart of outstanding work. It is useful for predicting when all of the work will be completed. It is often used in agile software development methods such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

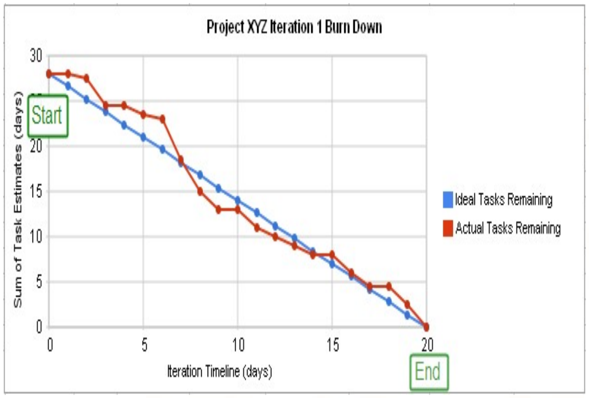


Fig 6: Burndown chart

Burndown chart shows the following information in scrum method

- *Total Estimate:* It is the sum of efforts in hours of all the user-stories, tickets, and issues, basically its the total number of works in hours to which the team is committed to.
- *Amount of work Remaining or Effort Remaining:* This is what burn down shows and this is how this graph get its name, in literal meaning its the "effort burndown chart". The Team will burndown some effort each day so that on last day of sprint or release there is no work effort remains
- *Working Days:* Since team need to calculate and carefully work on the commit item each day, so that is the reason total days of commitment of work are shown in graph. This is the total working days in a sprint (excluding holidays, weekend, etc.). This is actually your sprint duration**.**
- *Ideal Effort:* The ideal effort is drawn as a guide for a team,it's drawn by calculating the exact amount of effort remaining which team need to burndown. That is the reason you see a very straight line from the top of Y-axis to X-axis, which is the last day of your sprint.
- *Real Effort:* Effort remaining line varies from team to team and day to day. It depends on how much effort remaining is added or reduced each day. If more items (user storiesand issues) are added after the sprint started, this show as an upward spike.

B. Scrum task board

A ScrumBoard is a tool that helps Teams makes Sprint Backlog items visible. The board can take many physical and virtual forms but it performs the same function regardless of how it looks. The board is updated by the Team and shows all items that need to be completed for the current Sprint.Either during or before the daily scrum, estimates are changed (up or down), and cards are moved around the board.As an example, the Scrumboard looks like this:

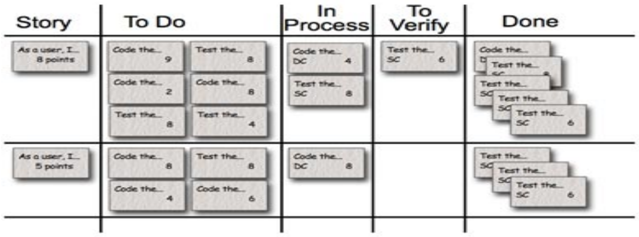


Fig 7: Scrum task board

The columns we generally use on a taskboard are:

- *Story*: The story description ("As a user we want to…") shown on that row.
- *To do:* Place for all cards that are not in the "Done" or "In Process" columns for the current sprint.
- *Work in process:* Any card being worked on goes here. The programmer who chooses to work on it moves it over when she's ready to start the task. Often, this happens during the daily scrum when someone says, "I'm going to work on the boojum today."
- *To verify:* A lot of tasks have corresponding test task cards. So, if there's a "Code the boojum class" card, there is likely one or more task cards related to testing: "Test the boojum", "Write FitNesse tests for the boojum," "Write FitNesse fixture for the boojum," etc. Some task cards don't get corresponding test cards ("Fix Bug No. 321 in Bugzilla") so those are placed in the "To Verify" column.
- *Done:* Cards pile up over here when they're done. They're removed at the end of the sprint. Sometimes we remove some or all during a sprint if there are a lot of cards.

C. Velocity Charts

While burndown charts measure the work completed in each sprint, velocity charts measure the work completed over the course of an entire project. The vertical y-axis represents the number of story points, and the horizontal x-axis represents the number of sprints (see screenshot below).Velocity is calculated by adding up the story points completed by the team in each iteration. The average velocity is simply the average of all sprints completed to date in a project.

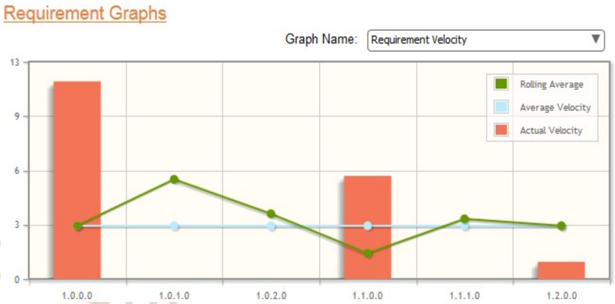


Fig 8: Velocity Chart

With velocity charts, teams can:

- Schedule reports to run concurrently with a project, automatically re-calculating average velocity as teams complete each sprint.
- Use built-in search functionality to easily recall previous project data.

VII. ADVANTAGES

Scrum is a highly prescriptive framework with specific roles and ceremonies. While it can be a lot to learn, these rules have a lot of advantages. The benefits of Scrum include:

A. More transparency and project visibility

With daily stand-up meetings, the whole team knows who is doing what, eliminating many misunderstandings and confusion. Issues are identified in advance, allowing the team to resolve them before they get out of hand.

B. Increased team accountability

There is no project manager telling the Scrum Team what to do and when. Instead, the team collectively decides what work they can complete in each sprint. They all work together and help each other, improving collaboration and empowering each team member to be independent.

C. Easy to accommodate changes

With short sprints and constant feedback, it's easier to cope with and accommodate changes. For example, if the team discovers a new user story during one sprint, they can easily add that feature to the next sprint during the backlog refinement meeting.

D. Increased cost savings

Constant communication ensures the team is aware of all issues and changes as soon as they arise, helping to lower expenses and increase quality. By coding and testing features in smaller chunks, there is continuous feedback and mistakes can be corrected early on, before they get too expensive to fix.

VIII. DISADVANTAGES

While Scrum offers some concrete benefits, it also has some downsides. Scrum requires a high level of experience and commitment from the team and projects can be at risk of scope creep.

Here are the disadvantages of Scrum:

A. Risk of scope creep

Some Scrum projects can experience scope creep due to a lack of specific end date. With no completion date, stakeholders may be tempted to keep requesting additional functionality.

B. Team requires experience and commitment

With defined roles and responsibilities, the team needs to be familiar with Scrum principles to succeed. Because there are no defined roles in the Scrum Team (everyone does everything), it requires team members with technical experience. The team also needs to commit to the daily Scrum meetings and to stay on the team for the duration of the project.

C. The wrong Scrum Master can ruin everything

The Scrum Master is very different from a project manager. The Scrum Master does not have authority over the team; he or she needs to trust the team they are managing and never tell them what to do. If the ScrumMaster tries to control the team, the project will fail.

D. Poorly defined tasks can lead to inaccuracies

Project costs and timelines won't be accurate if tasks are not well defined. If the initial goals are unclear, planning becomes difficult and sprints can take more time than originally estimated.

CONCLUSION

Scrum encourages data-based, iterative decision making in which the primary focus is on delivering products that satisfy customer requirements. To deliver the greatest amount of value in the shortest amount of time, Scrum promotes prioritization and Time-boxing over fixing the scope, cost and schedule of a project. An important feature of Scrum is self-organization, which allows the individuals who are actually doing the work to estimate and take ownership of tasks. Scrum requires a high level of experience and commitment from the team. Some of the key benefits of using Scrum in any project are adaptability, more transparency and project visibility, increased team accountability, .Increased cost savings, faster problem resolution and also continuous delivery of value.

ACKNOWLEGEMENT

I take this opportunity to sincerely thank my teacher, Prof. Mrs. Pratibha Adkar for her timely suggestion and invaluable guidance during preparation of this paper. She is the one who gave me confidence for publishing this paper.

REFERENCES

1. Igor Ribeiro Lima, Tiago de Castro Freire, Heitor Augustus Xavier Costa "Adapting and Using Scrum in a Software Research and Development Laboratory" in journal of " Revista de Sistemas de Informação da FSMA n. " published in Revista de Sistemas de Informação da FSMA n. 9 (2012) pp. 16-23
2. Bernardo Vasconcelos de Carvalhoa , Carlos Henrique Pereira Mellob," Scrum agile product development method - literature review, analysis and classification",injournal of "a University Center of Itajubá b Federal University at Itajubá" published in Vol. 9 nº 1 June 2011.
3. AsthaSinghal, Divya Gupta "Scrum: An Agile Method",in journal of "International Journal of Engineering Technology, Management and Applied Sciences" published in November 2014, Volume 2 Issue 6, ISSN 2349-4476.
4. Sutherland J., Viktorov A., Blount J., Puntikov N., 2007 "Distributed Scrum: Agile Project Management with Outsourced Development Teams." Proceedings of the 40th Hawaii International Conference on System Sciences 2007.IEEE Computer Society.

5. Eliza S. F. Cardozo," SCRUM and Productivity in Software Projects: A Systematic Literature Review" published by Cardozo A. NetoBarza França Da Silva
6. Bates C. D., Yates S., 2008 Scrum down: a software engineer and a sociologist explore the implementation of an agile method. International Conference on Software Engineering, Leipzig, Germany.ACM.
7. Abrahamsson, P.; Salo, O.; Ronkainen, J.; Warsta, J. "Agile Software Development Methods – Review and Analysis" published by VTT Publication 478. 107 p. 2002

Book reference

8. TridibeshSatpathy,"A Guide to the SCRUM BODY OF KNOWLEDGE (SBOK™ Guide) 2016 Edition",Published by: "SCRUMstudy™, a brand of VMEdu, Inc. 410 N 44th Street, Suite 240 Phoenix, Arizona 85008 USA.