Journal of Theoretical and Applied Information Technology
15 June 2012. Vol. 40 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

BE AGILE: PROJECT DEVELOPMENT WITH SCRUM FRAMEWORK

[1]SOUMYADIPTA PAUL, [2]K. JOHN SINGH

[1]Final Year M.Tech (IT), School of Information Technology and Engineering

VIT University, Vellore, Tamil Nadu, India

[2]Assistant Professor (Senior), School of Information Technology and Engineering

VIT University, Vellore, Tamil Nadu, India

E-mail: [1]soumyadipta_paul@yahoo.co.in, [2]johnsinghaj@yahoo.com

ABSTRACT

Today IT companies are facing the toughest competition to deliver their products with high quality within time. The rapid changes of the behavior of the industry and all new inventions of technologies in different fields, it becomes really challenging to develop and deliver a large software product within time with full customer satisfaction. Over all the industry is getting very spry in nature. On other hand software is something we cannot see unless it is fully working or providing some output. Now human cannot predict the future or plan for uncertain changes in distant future. So to support the flexibility of the market the development process of such project also needs to be more flexible, more agile. Therefore agile software development methodology helps to develop a project in iterative and incremental way and scrum is one of the frameworks which help to implement agile methodology in a software project development.

Keywords: *Agile Methodology, Scrum, Iterative and Incremental Development*

1. INTRODUCTION

Software development is an innovative process rather than predictive manufacturing process. In other industries like civil, automobile we can estimate the flow of work, we can visualize the growth of project. But software development needs great deals of continuous innovation and interaction. Here ethical project development approaches do not work out. So for obvious reason software development needs a different approach for being treated. The Waterfall model is a very well-known software development methodology.

In 1970 Winston Royce describes a serial method of developing software [1]. After that the methodology was widely accepted and used to develop software. The phases of this methodology are gathering specifications to define how the software will look like, detailed analysis and design work before start of development, develop the software according the plan, testing the software and deliver the software. Adoption of this model helps to develop software but 70 per cent software project using this model fails to meet one or many objectives of the project [3]. Here the development is divided into specified phases, it is assumed that specifications hold sufficient details of the product and can be perfectly understood. Now software industry is an emerging and young industry, here every day requirements, standards are changing which leads a fixed phased iteration of project development to a failure. That is why we need some methodology which is lightweight and complete. Lightweight helps to adapt the change, according need and complete relies on strong communication. Therefore people start to develop with fewer boundaries of documentation, detailed design and planning and they start to get success too. The determining factor of the project success seemed to be moved from documentation to the people connected to the project. When people start to realise the benefits of lightweight process development, they start to develop in different lighter approach of software development, they named them Extreme Programming, SCRUM, Crystal, Adaptive, etc. Now these frameworks may focus on different aspect of a successful software development but they create the focal point for the community to accept iterative and adaptive way for software development. On the verge of this change agile methodology comes with the promise to fulfill these needs. It encourages to use rapid adaptive planning and iterative development to provide high

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

delivery rate with flexibility to change while maintaining the quality.

2. AGILE METHODOLOGY

In February 2001, 17 software developers with their new vision of software development gather to discuss their approaches, to explore a lightweight development method. The result was "Manifesto for Agile Software Development" [20]. It aggregates the core principles and values of agile methodology.

Agile Manifesto:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more [18]

Principles behind the Agile Manifesto:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals.
- Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development.
- The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity-the art of maximizing the amount of work not done-is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly [18].

In this methodology the whole project is broken down into small iterations. Each iteration are small time framed approach where one cross platform team performs the planning, requirements analysis, design, coding and unit testing for the iteration. At the end of the iteration the developed product is demonstrated to the customer and the feedback is taken. This approach helps to adapt any changes at any stage of development. Now, in a single iteration all features of the product is not expected but some features are developed. Therefore one project becomes the result of multiple iterations. This helps to minimize the risk factors because you can see and identify the development of the product step by step.

In agile methodology teams are typically small in size (5-9) people [16]. It helps to increase the team collaboration. To complete the project there may be possibility that multiple teams work for a common or separate goals. Everyday team organize a standing meeting to discuss the last day progress, current issues and plan the work list for the rest of the day.

Agile methodology frees the project manager from being an operational controller and allow to an adaptive leader. He helps to keep the spotlight on the vision, inspire the team, promote teamwork and collaboration, removes obstacles to progress. He also regularly communicates with the customer and partly works as a customer representative for the team.

The numbers of software development frameworks which follow agile methodology are increasing. Among them the widely used

Journal of Theoretical and Applied Information Technology
15 June 2012. Vol. 40 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

frameworks are SCRUM, Extreme Programming, Crystal, Feature Driven Development, Lean Development etc. This paper will focus on the scrum framework to conclude a software project with agile methodology.

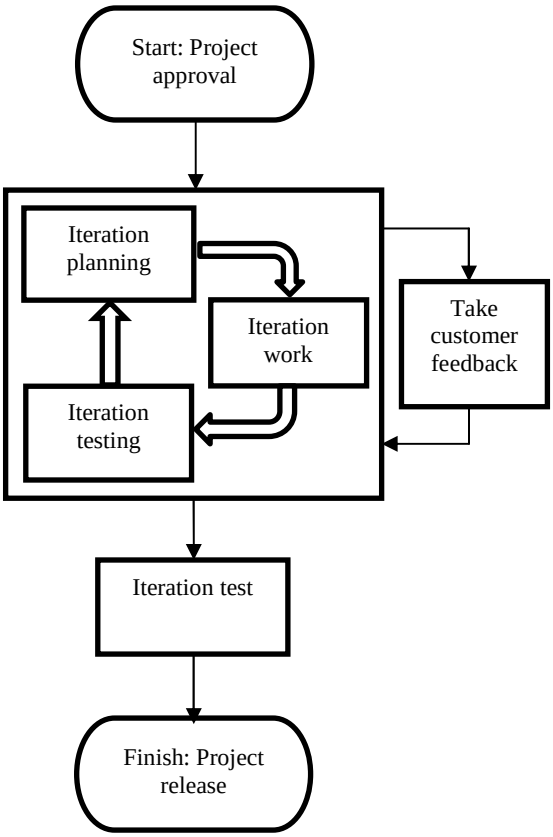

Fig.1. A generic agile project development work flow.

3. SCRUM AT A GLANCE

Scrum is an iterative and incremental framework for project or application development. The framework was introduced by Ken Schwaber and Jeff Sutherland in 1995 in their joint paper. They also held an implementation workshop in OOPSLA '95 in Austin, Texas [20]. In this framework the full development is done in small multiple iterations. These are known as sprint. Sprints are time-boxed approach, each sprints are of maximum one month at the end of time the sprint is stopped, no matter the planned jobs are completed or not. At the start of each sprint a cross platform team is organized and the team selects the items to be done in that sprint from a prioritized list of tasks. When a sprint gets started, every day the team meet in a discussion to briefly inspect the progress, to plan the next steps and to arouse any issues if there. At the end of the sprint the team follows up with the customer or customer representative on the progress of the project. In this follow up meeting the team collect the feedback, any new features if required and on basis of that next sprint is planned.

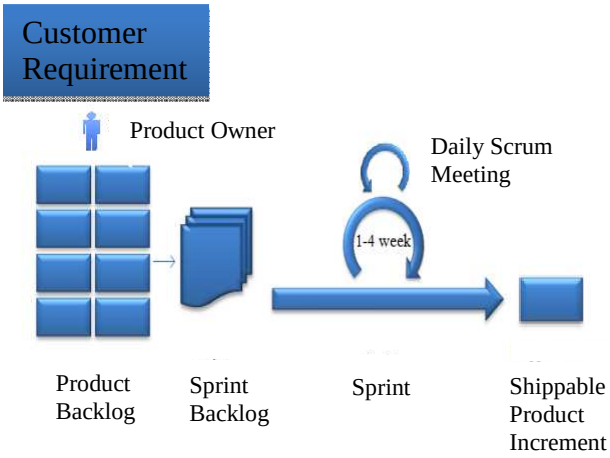

Fig.2 Scrum Workflow

Currently scrum is used in companies like Yahoo, Microsoft, Google, Siemens, Nokia, Alcatel-Lucent etc.

4. THE THEORY BEHIND SCRUM

The theories working behind the scrum are transparency, inspection and adaptation [14]. Scrum encourages in taking baby steps in development. It helps for inspecting the present scenario and the resulting product, adapting the product goal and process practice.

The process must be visible to the people who are responsible for the successful end of that. Transparency is required in every aspect so that everyone share a common understanding on everything. For this a common language and common definition of "Done" [15] must be shared. The project related people must inspect the project artifacts and the progress in some intervals. This helps to detect undesirable discrepancy.

Now if any aspect of the process divert from the acceptable limit of the product then the process must be adjusted at the point of identification of the diversion. This ability of adaption in project development helps to minimize the deviation at the end of product development.

5. SCRUM ROLES

Before dive into the framework let us see the different roles played in scrum framework. The

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

people involved into a scrum development are together known as scrum team. Scrum team consists of three roles and they are Product owner, scrum master and the team [15].

Product Owner:

Product owner is a person who is responsible for maximizing the value of the product in the sense of return on investment (ROI) [16]. Product owner identify the features of the product and build a prioritized list of tasks and assist the scrum team to select the next group of tasks to be completed in next sprint. Product owner also need to reprioritize and refine the list in continues time span. In some cases product owner can be visualized as customer or a representative of customer, in that case, he or she has to look into the profit and loss of the product. For internal project product owner is responsible for maximize the ROI in sense of high business value and low cost. Product owner may represent the customer or the internal company committee but product owner must be always one person. The role for product owner is quite different from the traditional project manager because product owner stays in contact with the team personally, meet them in every sprint planning meeting and help to prioritize the task personally rather than taking development decisions. For the scrum team the product owner's list of task is final and respected most, no one can change or add task in that accept product owner.

Scrum Master:

Scrum master is the person who helps the team to follow the scrum framework in way to achieve the goal. The scrum master makes sure that the team is following the scrum rules and moving forward to achieve the goal for the sprint. For product owner's help purpose, scrum master understands the tasks, priorities, long term goals, agility of the project and facilitates all scrum events. On the other hand scrum master coach the team in self-organization, cross functionality, remove issues in team's progress and lead them to create high value product. Scrum master protect the team from any outside disturbance, so if needed scrum master organize meeting with other teams to collect sprint related data from them and provide them to the team. Scrum master does not assign task like a project manager but support the team to organize and manage itself.

The Team:

The team is built with expertise from different areas to produce potential releasable increment of product at the end of each sprint. The team only builds the product which is going to be used by the customer. In scrum, teams are cross functional and self-organized. So if it is a software product, then the team will have programmer, interface developer, tester etc. but all of them known as developer. There are no sub-teams exist in a scrum team. Every one concentrates on their part of work and correlate with each other daily. The size of the team always kept small, it varies from five to ten. Since the team take care of every features of the product so some time teams are also known as feature team. As there is no project manager concept in scrum so the team only finalize which task to be taken in a sprint and how much time should be taken for each task in a sprint.

6. SCRUM FRAMWORK

Scrum framework is very easy to understand and follow if everyone follows their roll properly. To understand the framework let us go through different scrum artifacts and scrum events.

Product Backlog:

The first step in scrum is for product owner to visualize the product and generate a refined, prioritized list of task which defines the product. Product owner meet the customer and get the features needed in the product. Then he or she build more technical oriented list of tasks. It contains required features of the customer, updated features proposed by the team or scrum master for improvement of product, features to provide security, research work. These tasks are written in story format. The structure of a story is: "As a <user type> I want to <do some action> so that <desired result>". In this way team can easily identify the user and understand the actual requirement. Product backlog stays till the end of the project. Product owner regularly update or reprioritize the list according to the need of customer, insights or new ideas of team, market need, technical issues and so on.

In every product backlog grooming product owner and the team collaborate on the development of the product. They add details, review and order the details in product backlog. The team tell about the amount of effort is needed in every task and product owner is responsible to assign business value to those task. According to the effort needed every task is assigned with some points and in each sprint the target points to be achieved are 26. The tasks in product backlog may vary in size or in needed effort so bigger tasks are broken into small

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

task list. This is performed in Backlog Refinement workshop or in the Sprint Planning Meeting.

Table 1: Product Backlog Outline

Item	Priority	Estimate of Value	Estimate of effort
As a buyer I want to place a book in shopping cart	1	7	6
As an owner I want to improve transaction process	2	6	7
…..	….	….	….
As a shopper I want to add, delete items from my wish list	10	4	2

The Sprint:

The heart of scrum framework is sprint. It is a time-boxed iteration of some scrum practice. At the end of each sprint we get a usable, shippable and potentially increment of the product. The duration of sprint is fixed from one week to one month, but throughout one development effort the length of every sprint is fixed and same. Each sprint consists of Sprint Planning Meeting, Daily Scrums, the development work, the Sprint Review Meeting, and the Sprint Retrospective. In each sprint a list of task is taken to complete and a definition of "Done" guide the developer to finish the task properly. No matter the task is completed or not by the definition of "Done" but the time limit of a sprint is not stretched. If it is not finished then the whole task is again put to the product backlog. Due to small iteration, maximum one month, sprints help to predict the progress by inspection and adaptation in development by once in a month at least. So the risk also comes down to one month's effort minimum.

Sprint Planning Meeting:

Before the start of every sprint a meeting is held to plan the sprint. The whole scrum team participates in this meeting. The time duration of the meeting is proportional to the time limit of sprint. The meeting has two parts. The first part is to decide why should be done in next sprint and the next part is to decide how the chosen works should be done.

In the part one of this meeting product owner introduce the product backlog to the team with new prioritized tasks and new tasks if any is added later. Then product owner define the definition of "Done" for each task. This definition provides the standard of the outcomes or result of each task. This helps the developers to understand the requirement. Now product owner always try to complete the high priority task first but choosing tasks for a sprint totally depends on the team. In the second part team decide how to materialize the product backlog tasks. They make a plan for the sprint; if any product backlog task is big enough then those are broken down into small story points. The tasks decided for a sprint are documented and this is known as sprint backlog. Then the teem design a work system and list the works to be done to build working increment for the product. Product owner may present in this part of planning for further clarification of product backlog. At the end of the meeting the team should be able to describe their involvement, commitment towards the development and how they are planning to accomplish the goal.

Daily Scrum:

Daily scrum is a daily time-boxed meeting of the team and scrum master. In this meeting the team discusses about the efforts implemented from the last daily scrum meeting, plan the works to be completed till next daily scrum meeting and discuss about any issues if occurred. The team uses this as assessment for moving towards the goal of the sprint. Here every member took part and perform their part of the meeting. Therefore this meeting serves the purpose of more self-organize rather than current status report. The scrum master ensures that the meeting takes place daily, help to improve the communication between team members and enforce different scrum rules. To keep track of the work progress scrum master maintain a burn-down chart. This chart is a graphical representation of work to be done on daily basis and the work is done on daily basis. This helps the team to visualize the remaining work to be done. In fig4, the blue line indicates the remaining work till that date of sprint.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

And the red line indicates actual work remaining till that date of sprint.

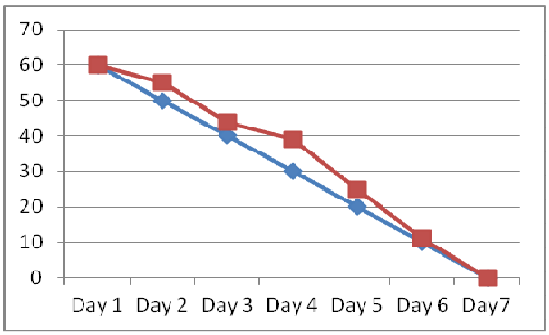

Fig.3 Sample Burn-Down Chart (with remaining work vs. no of days in sprint)

Sprint Review:

At the end of each sprint a sprint review meeting is taken place. Here the team discuss with the product owner about what is "Done" and what is not "Done". After that the team demonstrates the product backlog tasks what are "Done". Stakeholders provide their view towards the progress of the product. Product owner discuss the product backlog as where it stands and reorganize it according need. The team discusses about what went well and what went bad in the last sprint to adapt with the current scenario. This meeting is a time-boxed effort which varies proportionally with the time span of sprint. The purpose of this meeting is not only giving 'demo' but to inspect and adapt. This helps everyone from the scrum team to get an idea about what going on with product, team etc. and as a result of that it increase the collaboration between the scrum team.

Sprint Retrospective:

Sprint retrospective is a practice of self-organize, self-evaluation and inspection of areas where improvement is needed. This is a three hour time-boxed meeting and proportionally less is allotted for small sprints. Scrum master always should encourage for this kind of meeting. Because the backbone of scrum framework depends on inspect and adaptation. So after every sprint before the next sprint the scrum team must analyses what is going well and what is not and where scope of improvement is present.

7. CHALLENGES

Every changes have its own consequences and so does scrum framework. I have worked in a multinational company in two different projects of different size. There I have used agile framework and scrum methodology for development purpose. During my work I realize that implementation of scrum in different scenario has its own set of challenges. Before implementing the framework first the organization must be prepare to adapt it. Some time workers fail to understand the need and scope of new changes and organization also fail to explain the proper advantages of new changes. For a large organization teams should be connected with each other. Sometimes different development teams are situated in geographically different places. So it become difficult to build a cross functional team to work on a project. I have also seen that cross functional teams take time to bring integrity between team members, people hide their faults, do not share their opinions and these leads delay in decision making process. Instead of tacking decision by themselves team members think that scrum master will take decisions. Now as scrum highly depends on transparency so large organizations often fail to get the proper essence of scrum or agile methodology. Sometimes handling some issues are beyond the capability of scrum master, at that time he or she needs organization level help. Still follow a total new methodology just because the hype of the methodology in market cannot be justified. So organization itself first realizes the nee of he changes.

8. OVERCOME THE CHALLENGES AND THE FUTURE:

The philosophy behind agile or the theory behind the scrum is easy to realize, but implementation of them in large scale needs proper planning, time and organizational support. Main reason behind the failure of scrum is seen to be the misinterpretation of the scrum rules. Therefore the best way for scrum practice is to have a scrum coach to assist the teams in initial days. The scrum masters, product owners and team members should be properly trained by scrum specialist to carry on their tasks properly. The organization also needs to be bent to support the new development methodology and framework. To support scrum master in organizational level meta-scrum masters should be trained also who will solve cross development or higher level issues for a scrum master. On top of that feasibility of scrum implementation in one organization must be checked also.

Now following a new methodology does not mean to follow that only. Scrum is a simple framework with complex behavior. Something

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

works for one organization does not mean that will work for others. I have been trained to develop project in scrum framework and during my work I have realized the benefits of the methodology but I have also realized that, we should not throw away all good practices for just the sake of welcoming a new methodology. Scrum is a framework only to follow agile methodology in software development. We can add our own rules to customize the framework for better result according to the need of organization. That is why we can see Scrum-Ban which is nothing but the combination of the best features of scrum and kanban [19]. Similarly we can notice the emergences of ScrumP in some organizations. There people are developing with the hybrid methodology of Extreme Programing and Scrum [12]. For a large scale of project we can use pipeline of sprints. In this framework independent sprints may run in parallel and later they are merged [8]. Large organization can also use multilevel architecture of scrum. In this kind of architecture one higher level sprint consist of more than one sprint as a unit task. The result of lower level sprint is send to the higher level scrum master and so on.

In my projects we used scrum with some other tools also. We have used a work progress board. It consists four columns, planned, in progress, blocked and done. We write the sprint tasks in sticky notes and paste them in that board in there proper places according their current progress status. The board was visible to all team members and accessible to all. We kept the board near our daily scrum meeting place and changed the positions of tasks if needed. Now the board concept we took from Kanban methodology but still it served us well in scrum methodology.

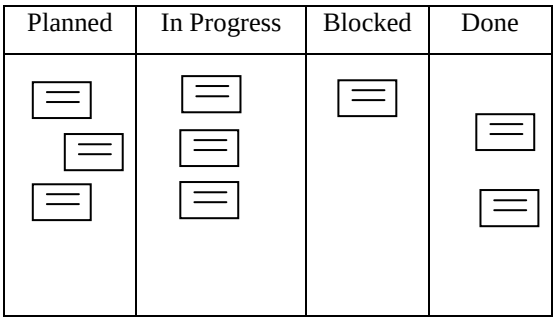

Fig.4 Diagram of our work board with sprint task and four statuses

My second project was smaller than the previous one. Therefore we changed the feedback loop and we contacted to product owner according our need not only at the end of sprint. We know that no tool is perfect or complete they just provide us some guide line to solve different problem. Now it is up to us that how we use those tools or how to combine them to get our own complete tool to solve our specific problem.

9. CONCLUSION:

Over the last decade scrum emerges from a humble beginning to a movement in software industry. Hundreds of leading companies are getting success in their thousands of project by using scrum framework. Now it is true that there are still some issues to rectify. Still scrum allows teams to work closely to the edge of chaos to promote development evolution. It enforces some simple rules to get the teams to be self-organized and help them to produce proper solutions within time. Now the pace of changes in software industry is at high rate. People are pushing every boundary. So scrum also needed to be pushed to its boundary by exploring all its hidden essences, so that we can develop an ultimate software development solution.

REFERENCES:

[1] http://en.wikipedia.org/wiki/Waterfall_model

[2]Victor Szalvay, “An Introduction to Agile Software Development”, Danube Technologies, Inc, 2004.

[3] An Introduction to Agile Software Development, Serena Software, 2007

[4] Posted by Todd Charron, “Organizational Culture and Agile: Does it fit?”, InfoQ, 2011

[5] Gabrielle Benefield, “Rolling out Agile at a large Enterprise”, HICSS '08 Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences, IEEE Computer Society Washington, DC, USA, 2008

[6] Jeff Sutherland, PatientKeeper, “Agile Can Scale: Inventing and Reinventing Scrum in Five Companies”, Inc., 2001

[7] Jeff Sutherland, Anton Viktorov, Jack Blount, Nikolai Puntikov, “Distributed Scrum: Agile Project Management with Outsourced Development Teams”, Agile 2006 International Conference, Minneapolis, Minnesota, USA, 2006

[8] Jeff Sutherland, Guido Schoonheim, Maurits Rijk, “Fully Distributed Scrum: Replicating

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Local Productivity and Quality with Offshore Teams", System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference, 2009.

[9] Brent Barton, Ken Schwaber, Dan Rawsthorne, "Reporting Scrum Project Progress to Executive Management through Metrics", Scrum Alliance, 2005

[10] Venkat Subramaniam, "Agile Methodologies", Presented in Boston, MA, August, 2005

[11] "Agile Project Management", CC Pace Systems, 2011

[12] Lowell Lindstrom, Ron Jeffries, "Extreme Programming and Agile Software Development Methodologies", Auerbach IT Journal, 8th Edition, 2003

[13] Pat Elwer, "Agile Project Development at Intel: A Scrum Odyssey", Danube Case Study, Intel Corporation, 2008

[14] Ken Schwaber and Jeff Sutherland, "The Definitive Guide to Scrum: The Rules of the Game", The Scrum Guide, July 2011.

[15] Jeff Sutherland, "Jeff Sutherland's Scrum Handbook", Scrum Training Institute,2011

[16] K. Schwaber and J. Sutherland, "The Scrum Guide", Scrum.org, 2010.

[17] J. Sutherland, "Future of Scrum: Parallel Pipelining of Sprints in Complex Projects", AGILE 2005 Conference Denver, CO: IEEE, 2005.

[18] http://agilemanifesto.org/

[19] Henrik Kniberg and Mattias Skarin, "Kanban and Scrum - making the most of both", InfoQ.com, 2009

[20] http://en.wikipedia.org/wiki/Scrum_(development)