

# SCRUM: AN AGILE FRAMEWORK

Rashmi Popli<sup>1</sup> & Naresh Chauhan<sup>2</sup>

---

During the 90s a number of different people realized that things had somehow changed. These people became interested in developing software methodologies that were a better fit with the new business. Although the details of these methodologies differ, they all share certain underlying principles, to the extent that these methodologies are often now grouped under the title "agile methodologies". The Scrum Agile development process was invented to rapidly drive new product to market. This paper presents an overview of Scrum Project Management & Development Methodology. The purpose of this paper is to represent the benefits of Scrum over Traditional Methods by comparing cost, quality, productivity and Return on investment.  
Keywords: Scrum, ROI, Sprint, Iteration

---

## 1. SCRUM INTRODUCTION

Scrum name derived from an activity that occurs during Rugby Match (A group of players forms around the ball and team-mates work together, sometimes violently, to move the ball downfield.) Scrum is a framework that enables iterative and incremental product development, allows getting things done at the right time, maximizing the value of what is delivered. Tasks are performed faster and with higher quality by self-organizing teams. High levels of self-motivation are achieved and are the reason why Scrum allows teams to reach higher productivity faster [1]. Customer requirements are constantly prioritized according to Business Value and integrated into the Product at regular intervals, allowing the customer to promptly provide feedback to the Team and thus improving the quality of the product on time.

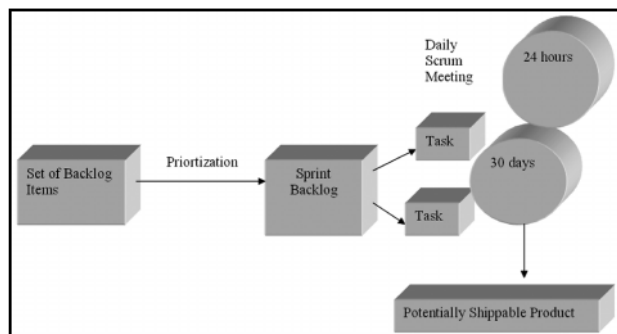


Fig. 1: Sprint Process Overview

The Scrum methodology emphasizes communication and collaboration, functioning software, and the flexibility to adapt to emerging business. Scrum's goal is to deliver as much quality software as possible within a series (3-8), of

short time-boxes called Sprints that typically last about a month. Fig 1 shows the Sprint Process Overview. Each stage in the development cycle (Requirements, Analysis, Design, Evolution, and Delivery) is now mapped to a Sprint or series of Sprints.

The traditional software development stages are retained for convenience primarily for tracking milestones [2]. So, for example, the Requirements stage may use one Sprint, including the delivery of a prototype. The Analysis and Design stages may take one Sprint each while the evolution stage may take anywhere from 3 to 5 Sprints.

## 2. SCRUM PHASES

### 2.1. Pregame Phase or Planning Phase

This phase includes definition of a new release based on currently known backlog, along with an estimate of its schedule and cost. If a new system is being developed, this phase consists of both conceptualization and analysis. If an existing system is being enhanced, this phase consists of limited analysis. It also includes system architecture modification and high level design. Fig. 2 shows Pregame Phase.



Fig. 2: Pregame Phase

### 2.2. Game or Development Sprints Phase

This phase contains development of new release functionality, with constant respect to the variables of time, requirements, quality, cost, and competition. Interaction with these variables defines the end of this phase. There are multiple, iterative development sprints, or cycles, that are used to evolve the system.

### 2.3. Postgame Phase or Closure Phase

Closure phase includes preparation for release, including final documentation, pre-release staged testing, and release.

## 3. WORKING MODEL

Scrum is iterative working method. The biggest difference to normal process based method is that it doesn't take any effort on how things are done and in what order [3]. Scrum only takes actions that in a sprint scrum team will deliver new working and runnable version of software. Scrum relies in that the team know which way is the best to get the job done.

### 3.1. Product Backlog

Product backlog is a document which lists all the characteristic of software in prioritized order [7]. Product backlog lives the same life cycle of a product itself. When some feature is implemented it is removed from a list. The product backlog items are prioritized and then transferred to the sprint backlog.

### 3.2. Sprint Backlog

Sprint backlog works as a task list in one sprint. It contains items from product backlog which are planned to be implemented in a one sprint. One product characteristic is divided in to smaller tasks. New tasks cannot be added into spring backlog when the sprint has started.

### 3.3. Sprint

Scrum work phase is based on iteration. One work unit is called sprint. And the goal of sprint is to develop deployable unit of software, or extend existing ones. Scrum sprint is 20 – 60 days long in calendar time. At this time team members only duty is to give all their contribution to tasks which are in sprint backlog. Sprint starts having a meeting with product owner, and then team decides together with product owner which characteristics are taken to sprint backlog [4].

## 4. SCRUM BURN-DOWN CHART

The Burn-down Chart is used as a tool to guide the development team to successful completion of a Sprint on time with working code that is potentially shippable as a product. On a Agile project, the team tracks its progress against a release plan by updating a release burn-down chart [5] at the end of each sprint.

The horizontal axis of the release burn-down chart shows the sprints as in figure 3; the vertical axis shows the amount of work remaining at the start of each sprint. The work remaining can be shown in whatever unit the team prefers—story points, ideal days, team days, and so on.

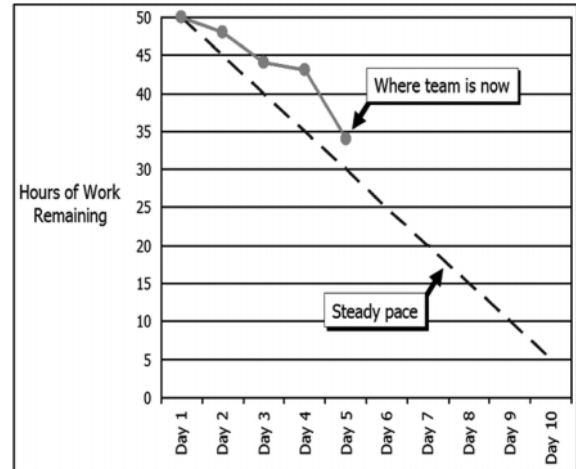


Fig. 3: Scrum Burn-down Chart

## 5. SCRUM METHODS COSTS AND BENEFITS

Scrum emphasize teams, working software, customer collaboration, and responding to change, while Traditional Methods focus on contracts, plans, processes, documents, and tools. The SEI study identified 99 data points on cost, schedule, productivity, quality, satisfaction, and ROI gains from 25 organizations as reported by CMMI®-related literature from SEI conferences. In Table 1 and Table 2, the category represents the benefits of Agile and Traditional Methods, while the low, median, and high represent the range of reported benefits within each category. Table 1 shows the benefits in terms of cost, schedule, productivity, Quality and ROI.

Table 1  
Scrum Methods Costs and Benefits

No.	Category	Low	Average	High	Points
1.	Cost	10%	26%	70%	9
2.	Schedule	11%	71%	700%	19
3.	Productivity	14%	122%	712%	27
4.	Quality	10%	70%	1000%	50
5.	Satisfaction	70%	70%	70%	1

Table 2  
Traditional Methods Costs and Benefits

No.	Category	Low	Average	High	Points
1.	Cost	3%	20%	87%	21
2.	Schedule	2%	37%	90%	19
3.	Productivity	9%	62%	255%	17
4.	Quality	7%	50%	132%	20
5.	Satisfaction	- 4%	14%	55%	6
6.	ROI	200%	470%	2,770%	16

## 7. CONCLUSION

There is no panacea for the complexities of software development. Scrum is devised specifically to wrest usable products from complex problems. It has been used successfully on thousands of projects in hundreds of organizations over the last sixteen years. Scrum is not for those who seek easy answers and simple solutions to complex problems; it is for those who understand that complex problems can only be met head on with determination and wit.

## REFERENCES

- [1] Pekka Abrahamson, Outi Salo, Jussi Ronkainen, Juhani Warsta "Agile Software Development Methods", VTT Publications 478, ESPOO 2002.
- [2] Victor Szalvay, "An Introduction to Agile Software Development", Danube Technologies, November 2004.
- [3] Ken Schwaber, Mike Beedle, "Agile Software Development with Scrum", Prentice Hall, 2001, pp. 100-101.
- [4] Mike Beedle, Martine Devos, Yonat Sharon, Ken Schwaber, Jeff Sutherland, "SCRUM: An Extension Pattern Language for Hyper Productive Software Development", 2000.
- [5] Ken Schwaber, "Agile Project Management with Scrum", Microsoft Press, 2004.
- [6] Miller, Steve, "Agile Scrum-An Overview", Pragmatic Software Newsletter, Feb. 2008, Pragmatic Software, 17 Jun 2009.
- [7] Ken Schwaber, Mike Beedle, "Agile Software Development with Scrum", Prentice Hall, 2001.

