

SCRUM

Maria Belkina
Jennifer Schiller
Maxim Masunov
Vyacheslav Filippov

May 23, 2006

Contents

1	Introduction	3
2	Agile Project Management/ software development	3
3	What is Scrum?	4
4	History of Scrum	4
5	Functionality of Scrum	5
6	Components of Scrum	5
6.1	Scrum Roles	5
6.1.1	Product Owner	6
6.1.2	Scrum Master	6
6.1.3	Scrum Team	6
6.2	The Process	7
6.2.1	Sprint Planning Meeting	7
6.2.2	Daily Scrum	7
6.2.3	Sprint Review Meeting	8
6.3	Scrum Artefacts	8
6.3.1	Product Backlog	8
6.3.2	Sprint Backlog	10
6.3.3	Burndown Charts	11
7	Scaling Scrum	13
8	XP@Scrum	14
9	Conclusion	15

1 Introduction

Classical methods of software development act on the assumption that corresponds to the procedure of engineers. However, this has several disadvantages:

- huge effort during the planning phase
- Adjustments during the realization is just with great effort possible a thereby poor requirements conversion in a rapid changing environment
- Treatment of staff as a factor of production

Agile methods attempt to counter these problems. I.e. there basic properties are:

- Incremental working in short cycles
- Permanent adjustments to new requirements
- Special consideration of the human component/ constituent

2 Agile Project Management/ software development

In the late 1990's several methodologies began to get increasing public attention. Each had a different combination of old ideas and new ideas. But they all emphasized close collaboration between the programmer team and business experts, face-to-face communication (as more efficient than written documentation), frequent delivery of new deployable business value, tight, self-organizing teams, and ways to craft the code and the team such that the inevitable requirements churn was not a crisis.

Early 2001 various originators and practitioners (Ken Schwaber & Jeff Sutherland amongst other) of these methodologies met to figure out just what it was they had in common. They picked the word "agile" for an umbrella term and crafted the Manifesto for Agile Software Development, whose most important part was a statement of shared development values:

"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." [2]

Hence, clearly can be seen that agile methods are a family of development processes not single approaches to software development. Most agile methods attempt to minimize risk by developing software in short iterations (each is like a miniature software project (requirements analysis, design, coding, testing É) of its own). Agile methods emphasize real-time communication, preferable face-to-face. That is agile methods produce very little written documentation. However, this can turn over to an enormous drawback, when all knowing team members leave the company, there knowledge goes with them and canÖt be reproduced.

Agile methods are indicated for small teams (j 10 developers) facing unpredictable or rapidly changing requirements. Organizations and individuals, who assure oneself of the potentiality of an agile method, can contact the *Agile Alliance*. This non-profit organization supports them during the execution of one of the different agile approaches to develop software (like Scrum, Crystal Clear, Extreme Programming, Adaptive Software Development ASD, Feature Driven Development, Dynamic System Development Method (DSDM) É).

3 What is Scrum?

Definition from rugby football:

A scrum is a way to restart the game after an interruption, where the forwards of each side come together in a tight formation and struggle to gain possession of the ball when it is tossed in among them.

Scrum - an agile process

SCRUM is an agile process for managing and controlling software and product development in rapidly changing environments. However, it can theoretically be applied to any context where a group of people need to work together to achieve a common goal. SCRUM naturally focuses on entire organizations on building successful products. Within 30 days teams are building useful, demonstrable product functionality. Scrum is a set of interrelated practices and rules that optimize the development environment, reduce organizational overhead, and closely synchronize market requirements with iterative prototypes. Based in modern process control theory, SCRUM causes the best possible software to be constructed given the available resources, acceptable quality, and required release dates. SCRUM generates the benefit of agile development with the advantage of a simple implementation. Hence, it is ideally suited for projects with rapidly changing or highly emergent requirements.

In contrast to classical development methods, Scrum intensely focuses individual members and the team, who achieve much responsibility, delegated by the management. This accentuation causes a significant increase in productivity. In addition Scrum improves communication and cooperation between the developers among themselves, to the management and to customers. Conflictive interests are reconciled. Furthermore one principle of Scrum is that the development process should be not disrupted, impediments must be removed and the team should be protected from everything outside the Sprint Goal to reach a high productivity.

Scrum is a lightweight process, which means that it doesn't contain any instruction for software development. Thereby it is possible to combine Scrum with real agile software development methods like Extreme Programming.

Over fifty organizations have successfully used SCRUM in thousands of projects to manage and control work, always with significant productivity improvements.

4 History of Scrum

Scrum emerged from combined endeavors of Advanced Development Methods (ADM) and VMARK Software, to enhance the problems of classical processes of development. In 1995 ADM intended to optimize their development processes. Therefore scientists from DuPont Experimental Station were assigned to analyze common software development methods. They concluded that the most development methods act on the assumption that software is developed in repeatable, clear-cut defined and predictable processes. However, software development processes are mostly unpredictable and non-repeatable, they are empirical processes.

Awareness of this status brought Jeff Sutherland (VMARK) and Ken Schwaber (ADM) (included the scientist from DuPont) to design their own, new method. Especially ideas from Ikujiro Nonaka and Hirotaka Takeuchi ("The Knowledge Creating Company", Oxford University Press, 1995) flow into the new approach. Also Takeuchi and Nonaka were the first who used the term SCRUM in "The New New Product Development Game" (Harvard Business Review 86116:137-146, Jan-Feb 1986).

Scrum owes Mike Beedle a lot of enhancements. He was one of the first who applied this new method and suggested several ideas for enhancement and was responsible for the combination of

Scrum with Extreme Programming. In 1996 Scrum was introduced on the occasion of the OOPSLA conference. (The Eleventh Annual ACM Conference on Object-Oriented Programming Systems, Languages and Applications [4]). Since then, Scrum was applied successfully for the software development in over 50 companies. With the publication of "Agile Software Development with Scrum" (Prentice Hall) followed a broad description of Scrum by Schwaber and Beedle in 2001. The inventors of Scrum are members of the Agile Alliance [3].

5 Functionality of Scrum

Scrum is ideally suited for projects with rapidly changing or highly emergent requirements. The work to be done on a Scrum project is listed in the Product Backlog, which is a list of all desired changes to the product. At the start of each Sprint a Sprint Planning Meeting is held during which the Product Owner prioritizes the Product Backlog and the Scrum Team selects the tasks they can complete during the coming Sprint. The tasks are then moved from the Product Backlog to the Sprint Backlog. During the Sprint the developers keep up-to-date by holding brief daily meetings. At the end of each Sprint the team demonstrates the completed functionality at a Sprint Review Meeting.

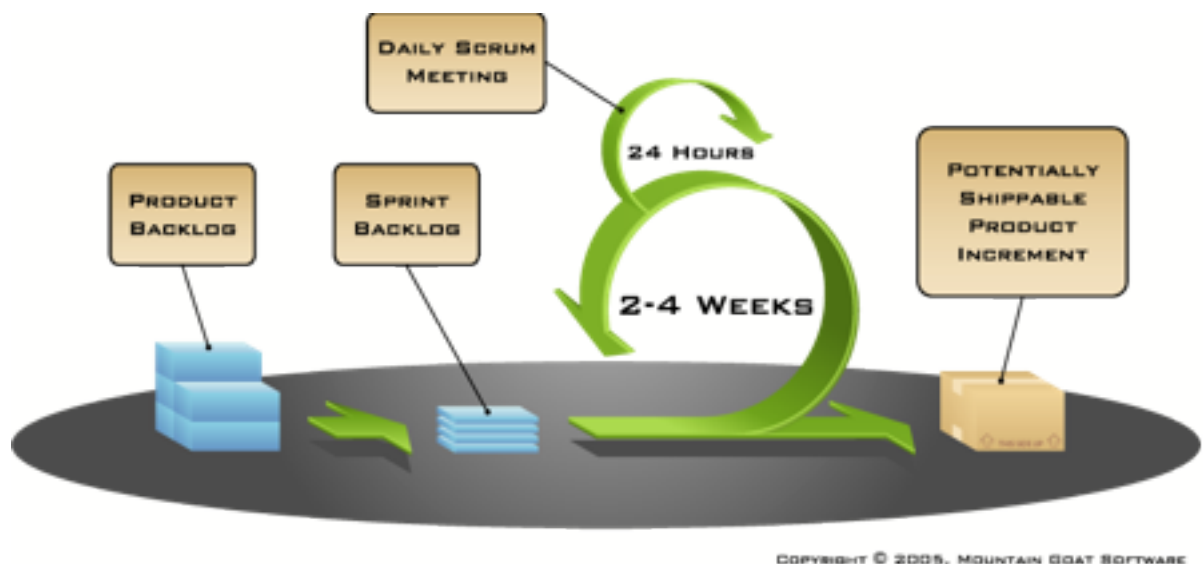


Figure 1: Scrum Overview

Scrum is facilitated by a Scrum Master, whose primary job is to remove impediments to the ability of the team to deliver the sprint goal. The Scrum Master is not the leader of the team (as they are self-organizing) but acts as a productivity buffer between the team and any destabilizing influences. [7]

6 Components of Scrum

6.1 Scrum Roles

Scrum uses three roles. These are Product Owner, Scrum Master and Scrum Team (also known as Project Team).

6.1.1 Product Owner

Product Owner is one of the product stakeholders, representing all of the customers [11]. This person must be available to the team at any time, but especially during the Sprint Planning Meeting and the Sprint Review Meeting. Focus of the PO (because he is a stakeholder of the product) is getting ROI (return on investment) as soon as possible, that is why he is strongly interested in the progress of the project. Main function of the Product Owner is to throw direct requirements at the Project Team before the sprint and to prioritize Product Backlog on the Sprint Planning Meetings. He can't change his requirements to the Team during the sprint. The PO owns the business plan, functional specification and product backlog of the product. PO can become not only owner, but the member of the team (to a greater or lesser extent), for example, when he visits all or most of the Scrum Meetings (but he still can't make changes in the selected part of the product backlog during these sessions).

Challenges of being a Product Owner are:

1. Resisting the temptation to "manage" the team. The team may not self-organize in the way he would expect it to.
2. Resisting the temptation to add more important work after a Sprint is already in progress.
3. Being willing to make hard choices during the Sprint Planning Meeting.
4. Balancing the interests of competing stakeholders.[13]

It is necessary for the Product Owner to work in consultation with the Scrum Master. Linkage between them can be very tight.

6.1.2 Scrum Master

Scrum Master is the key of the project, because he coordinates Team members, helps to plan every sprint and chairs each Daily Scrum Meeting. In spite of this Scrum Master is not the leader of the team (as they are self-organizing) but acts as a productivity buffer between the team and any destabilizing influences. [7]

The Scrum Master is the "Agile" Project Manager, but unlike the traditional Project Management role they aren't solely accountable for the success or failure of the project [11]. Mostly the Scrum Master is a part of the team: he can be designer, developer etc., but it is not the right way of being Scrum Master, because in that case he has not enough free time for solving Team's issues and coordinating them.

Scrum Master is like a bridge between the development and the product owner, helping the PO to drive the development. Scrum Master is a "nurse" of the Scrum Team: he makes most of the final decisions in the development process, tracks the progress of the team and removes impediments on their way. That is why he must be a guru not only in solving of possible problems, but also in developing, design and whatever is important for the team. But main function of the Scrum master is to keep his Team productive.

6.1.3 Scrum Team

Like in some of the other agile practices a development team in Scrum is cross-functional. That means it doesn't include any of the traditional software engineering roles such as Programmer, Designer, Tester, or Architect. Everyone on the project works together to complete the set of work they have collectively committed to complete within a sprint. [7]

Scrum team is self-organizing: They divide the tasks up among themselves, and are responsible for helping each other out beyond their "specialties". A Scrum team is collaborative and cooperative, and does not retreat behind job descriptions when work needs to be done [9]. So, the main function of the Team is meeting the sprint goal, doing it the way that they have chosen independently. Size of the Scrum Team is usually 6-10 people, it is recommended, that they work in one Team Room. But size of the Team can be much greater (up to 1000 people) when using "Metascrum" technology.

All the roles in Scrum are tightly linked between each other. If we imagine, that developing the project is analogous to driving a car in a rally race, the Product Owner would be the navigator and the Scrum Master would be the driver. The team is the engine, the chassis, the drive train, and the wheels. The Scrum Master follows the navigational directions of the Product Owner precisely and drives the car adroitly. The car and its occupants are totally responsible for winning the race. At the end of every Sprint, other players move in and can make modifications to improve the timing of the next Sprint. [14]

6.2 The Process

6.2.1 Sprint Planning Meeting

Each Sprint starts with the Sprint Planning Meeting. It is a collaborative meeting between the Product Owner, the Scrum Master and the Scrum Team. This typically takes one day (8 hours) and consists of two parts. During the first part of the meeting (before lunch) is created a Product Backlog and determined the Sprint Goal which is a short description of what the sprint will attempt to achieve. Product Owner describes the highest priority features to the Team, members of the team should ask him questions, so that that they can go off after the meeting and determine which tasks they will move from the Product Backlog to the Sprint Backlog.[7]

The second part of the Sprint Planning Meeting is dedicated to the Sprint Backlog. Typically Product Owner has not to be there. During these 4 hours Scrum Master and Scrum Team select items from the Product Backlog list and translate them into tasks for implementing the release, so that Team members can more effectively share work during the Sprint. As a result they create Sprint Backlog, which consists of "frozen" items from a Product Backlog.

Inputs of this meeting are previous Sprint Backlog, previous version of Product backlog (if they exist) and requirements of the Product Owner and (possible) of the Team. Outputs are new version of Product Backlog and Sprint Backlog for the next Sprint.

There is a special form of Sprint Planning Meeting: Pre-project/Kickoff Meeting. This is a meeting before the begin of a project, during which an initial Product Backlog is created, prioritized and estimated. [13] This drives the initial Sprint of the project while more Product Backlog items are developed. Sprint Planning Meeting is like the ignition key in the Scrum car. Success of the campaign is dependent of this Meeting.

6.2.2 Daily Scrum

Every day there is a short (15 minute) Daily Scrum meeting. This meeting is usually held in the morning, before the Team starts working. Chairman of this meeting is the Scrum Master. In the meeting he asks three questions, which should be answered by every team member. The questions are:

1. What did you do since the last Scrum?
2. What are you doing until the next Scrum?

3. What prevented you from doing work?

In first two questions is reflected the productivity of the team. Hence, the Scrum Master is able to quickly address and solve problems that occur during the sprint. The third question gives the Scrum Master all the information he needs to help the team work better. Sometimes it is as simple as "free coffee." Sometimes it can be as complex as organizational change. [12] Characteristic property of this meeting is that it is not a problem solving session. The Daily Scrum is also not a status update meeting where information is collected about who is behind schedule. It is a meeting in which team members make commitments to each other (not only to a Scrum Master!). By focusing on what each team member accomplished yesterday and will accomplish today the Scrum Team gains an understanding of what tasks have been done and what tasks remain.

Each participant in the Daily Scrum is known as either a chicken or a pig, depending on his involvement in the project. The reference is to the old joke that in a ham-and-eggs restaurant the pig is committed while the chicken is only involved. Only pigs are allowed to talk or ask questions during a Daily Scrum meeting. [7] Chickens cannot participate, because they are not actually in the team. According to our "rally conception" the Daily Scrum is a checkpoint during the rally.

6.2.3 Sprint Review Meeting

Sprint Review Meeting is held at the end of each Sprint. Production-quality, demonstrable business functionality, which is accomplished during the Sprint is demonstrated to the Product Owner (and other stakeholders). [13] Typically this is a demo of the new features. Ideally the Team should accomplish each task in the Sprint Backlog, but it is more important that they achieve the overall goal of the sprint determined during the Sprint Planning Meeting. Sprint Review Meeting is informal and should not distract Team members of doing their work. It is a natural completion of the Sprint.

6.3 Scrum Artefacts

6.3.1 Product Backlog

A product backlog is a prioritised list of things that a customer wants a development team or development teams to produce.[10] These includes both functional and non-functional customer requirements, as well as technical team-generated requirements.[7] Product Backlog is managed and owned by the Product owner. Usually it is a spreadsheet, but other good way of writing Product Backlog items is derived from XP: this is writing User Stories.

User Stories are brief one or two paragraph descriptions, in plain language, that define how the application should solve a problem or support a business process. They are most often written on 4 x 6 cue cards to keep them to-the-point and to avoid the hang-ups that using a tool can engender. [14]

Product Owner can write his own User Stories, but it is more likely to write them with the team members during the Sprint Planning Meeting, because the team can also add requirements. For example, the team requests to re-factor the code to eliminate duplicate code and increase maintainability. The Product Owner and stakeholders prioritize 'Eliminate Refresh Inconsistencies By Using the Same Code Base' as any other requirement. [7]

After creating of items is competed Product Owner should add a priority to each item, e.g. "low", "medium", "high" or "must have", "nice to have", "have if possible" ([14]). The team begins with working on the items of the highest priority, because they deem important to the Product Owner and ensure, that working on them meets his expectations. The Product Backlog is not frozen, it's re-prioritized before each Planning Meeting. New items in the Backlog may be

portions of features that were not completed in the Sprint, or new ideas surfaced by reaction to the work produced. During the Sprint some of the Backlog items are "locked" and cannot be changed, but the rest of the items is "unlocked" and can be changed any time the Product Owner (or anyone else) wants it.

	Item #	Description	Est	By
Very High				
	1	Finish database versioning	16	KH
	2	Get rid of unneeded shared Java in database	8	KH
		- Add licensing	-	-
	3	Concurrent user licensing	16	TG
	4	Demo / Eval licensing	16	TG
		Analysis Manager		
	5	File formats we support are out of date	160	TG
	6	Round-trip Analyses	250	MC
High				
		- Enforce unique names	-	-
	7	In main application	24	KH
	8	In import	24	AM
		- Admin Program	-	-
	9	Delete users	4	JM
		- Analysis Manager	-	-
	10	When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab	8	TG
		- Query	-	-
	11	Support for wildcards when searching	16	T&A
	12	Sorting of number attributes to handle negative numbers	16	T&A
	13	Horizontal scrolling	12	T&A
		- Population Genetics	-	-
	14	Frequency Manager	400	T&M
	15	Query Tool	400	T&M
	16	Additional Editors (which ones)	240	T&M
	17	Study Variable Manager	240	T&M
	18	Haplotypes	320	T&M
	19	Add icons for v1.1 or 2.0	-	-
		- Pedigree Manager	-	-
	20	Validate Derived kindred	4	KH
Medium				
		- Explorer	-	-
	21	Launch tab synchronization (only show queries/analyses for logged in users)	8	T&A
	22	Delete settings (?)	4	T&A

Figure 2: Example of Product Backlog spreadsheet (from [7]).

Finally, the team makes rough estimate assessments of how long each story will take to implement. The primary goal of estimating backlog items is to establish the team's "velocity", which is how much Product Backlog Effort a team can handle in one Sprint. [13] Product Backlog estimates provide a relative measure of the complexity of each different feature in the backlog. There are several methods of providing these estimates, but each has two components: the units of complexity employed and the method for analyzing a feature for its complexity [10]. Some common units of complexity are:

- Size category ("T-shirt size" estimate: large, small, x-large etc. The simplest way to estimate complexity of the item. The units of measure are categorical, not quantitative, so they can not be added up.)
- Story points (Relative, non-dimensional units of complexity. There is a one-point feature, two-point feature etc. These units are quantitative and form a linear scale of complexity.)
- Work days/work hours

Methods of estimation are:

- Expert review
- Creating a Work Breakdown Structure (WBS)(hierarchically decomposing of the Backlog items)[10]

But everyone should always remember: these estimates are actually more like long-range weather forecasts: not exact and are never intended to be (especially first estimations, that are made before the beginning of the first Sprint).

6.3.2 Sprint Backlog

Sprint Backlog represents a subset of Product Backlog Items, which are converted into tasks. It defines the work for a Sprint, represented by the set of Tasks that must be completed to realize the Sprint’s goals. Items on the Sprint Backlog are drawn from the Product Backlog by the team based on the priorities set by the Product Owner and the team’s perception of the time it will take to complete the various features. None but team members are allowed to create the Sprint Backlog, because they are the ones committing to completing the tasks. The team defines the Sprint goal that they will meet during the Sprint by developing these backlog items (such as ”demonstrate personalization capability”). The goal should be broad enough that the team can meet it in a variety of ways: it should specify what, not how. [9]

		Days Left in Sprint				
		15	13	10	8	
Who	Description	7/22/2002	7/24/2002	7/26/2002	7/31/2002	
Total Estimated Hours:		554	458	362	270	0
-	User's Guide	-	-	-	-	-
SM	Start on Study Variable chapter first draft	16	16	16	16	
SM	Import chapter first draft	40	24	6	6	
SM	Export chapter first draft	24	24	24	6	
Misc. Small Bugs						
JM	Fix connection leak	40				
JM	Delete queries	8	8			
JM	Delete analysis	8	8			
TG	Fix tear-off messaging bug	8	8			
JM	View pedigree for kindred column in a result set	2	2	2	2	
AM	Derived kindred validation	8				
Environment						
TG	Install CVS	16	16			
TBD	Move code into CVS	40	40	40	40	
TBD	Move to JDK 1.4	8	8	8	8	
Database						
KH	Killing Oracle sessions	8	8	8	8	
KH	Finish 2.206 database patch	8	2			
KH	Make a 2.207 database patch	8	8	8	8	
KH	Figure out why 461 indexes are created	4				

Figure 3: Sprint Backlog [7]

Sprint Backlog list (see the picture above) is a list of all the tasks/activities needed to develop the Product Backlog selected for this Sprint. Each of Sprint Backlog Items has its own status (usually estimated time (in hours)), which should be changed by its owner as the item is worked

on. Keeping the Sprint Backlog updated is very important, because it is an early warning indicator. Here are some notes about creating Sprint Backlog:

- There should be no more than 300 tasks in the sprint backlog list.
- Teams can add or subtract to sprint backlog as they learn more about the work that needs to be performed. Customers and management are not allowed to add or subtract to sprint backlog. [5]
- As each task listed, an estimate of 4-16 hours is assessed. If a task requires more than 16 hours, it should be broken down.
- If the team needs more information to make a decision during the Sprint Planning Meeting, it should be able to reach the Product Owner quickly.
- As the team is assembling their estimates, the Scrum Master is keeping a tally of the total estimate hours to see if the tasks can fit the sprint period. If the amount is greater than allows the Product Backlog, the last one either needs to be revisited with the Product Owner and pared down, or the task reviewed to see if there is a less-costly way to accomplish it.
- Sprint Backlog is only a forecast that is made by the developers. So its estimates can be incorrect. If there is too much work, tasks can be deleted from the backlog during the sprint and if there is not enough work, tasks can be added to the backlog. Team members make final decision only after they have consulted with the Scrum Master in both cases. Each Sprint Backlog item is used to generate a burn down of all work in the sprint.

6.3.3 Burndown Charts

Scrum uses Burn down Charts to represent "work done". [13] There are three main problems, which interest the Project management:

- Progress of the Sprint (how is the team meeting their Sprint goal?)
- Progress of the release (will the release be done right on time?)
- Progress of the product (is the product exactly what we need?)

And there are three types of charts according to these problems: Sprint Burn down Chart, Release Burn down Chart and Product burn down chart. To motivate the team, all the charts should be displayed prominently. These are effective Information Radiators: "Two characteristics are key to a good information radiator. The first is that the information changes over time. This makes it worth a person's while to look at the display... The other characteristic is that it takes very little energy to view the display." [14]

X-Axis of all the charts represent time (usually in days), Y-Axis remaining effort. Charts need to be updated continuously.

Sprint Burn down Chart

Sprint Burn down Chart (or "Sprint Burn down Graph") depicts the total Sprint Backlog task hours remaining per day. Each team member is responsible for estimating the number of hours remaining to complete all assigned tasks during a Sprint.

Ideally the chart burns down to zero by the end of the Sprint. So this is an early warning indicator. If the line on the chart is parallel to the X-axis, there are problems (not enough work has been done during long time). However, burn down graph cannot be a perfect straight line. If the team members are reporting their remaining task hours realistically, it should bump up and down chaotically, because tasks can be added into the sprint backlog during the sprint (if team members and Scrum Master think, there is not enough work for the sprint). If there is too much work during the sprint tasks can be deleted from the backlog, so there can be big drops on the chart (see figure 4).

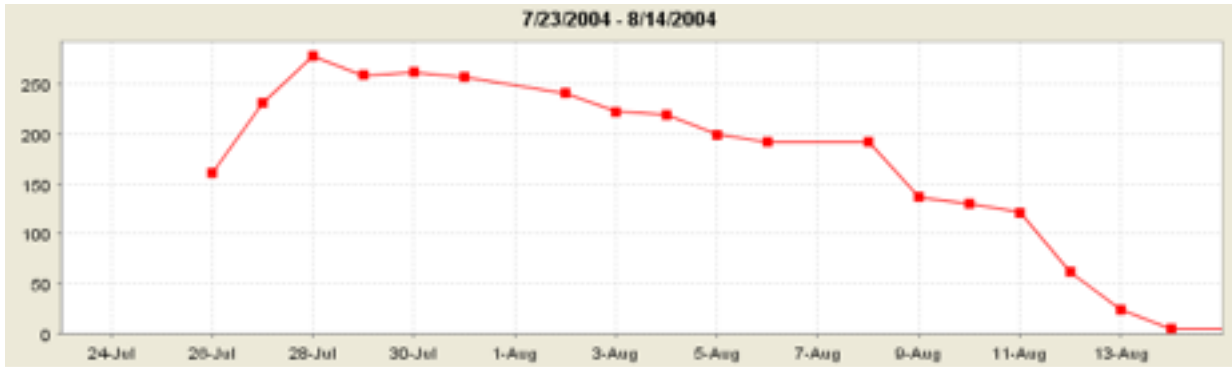


Figure 4: Sprint Burn down Chart

Release Burn down Chart

The X-axis of the release burn down chart (see figure 5) shows the sprints; the vertical axis shows the amount of work remaining at the start of each sprint (in story points or in working days/hours). [5]

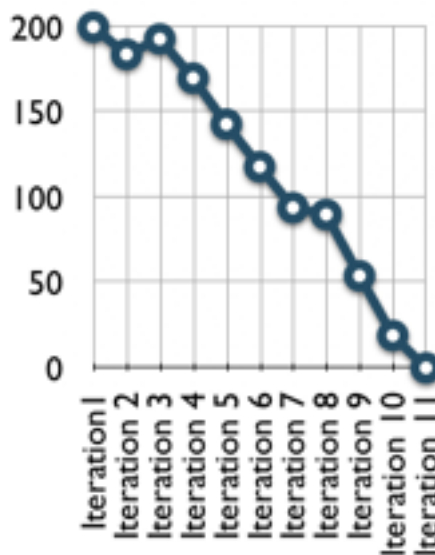


Figure 5: Release Burn down Chart [7]

The estimated work remaining can also burn up, when new tasks are added to the project. That’s why on projects with lots of changing requirements is useful another alternative release burn down chart, which consists of bars (one bar for each sprint). Values on the Y-axis can be either positive or negative. Negative part of the bar represents the work that has been added to the project. Height of the bar represents the work that should be done. So, Release Burn down charts provide a visual, visceral indication of the progress of a Scrum team from a long-term business perspective (see figure 6). [12]

Product Burn down Chart

Product Burn down Chart spans all the release charts. It is a “big picture” view of a project’s progress (see figure 7).

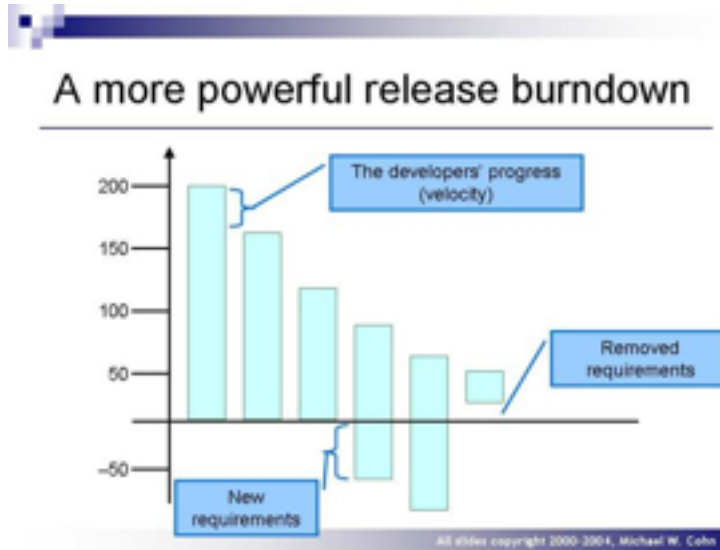


Figure 6: Alternative release burn down chart [7]

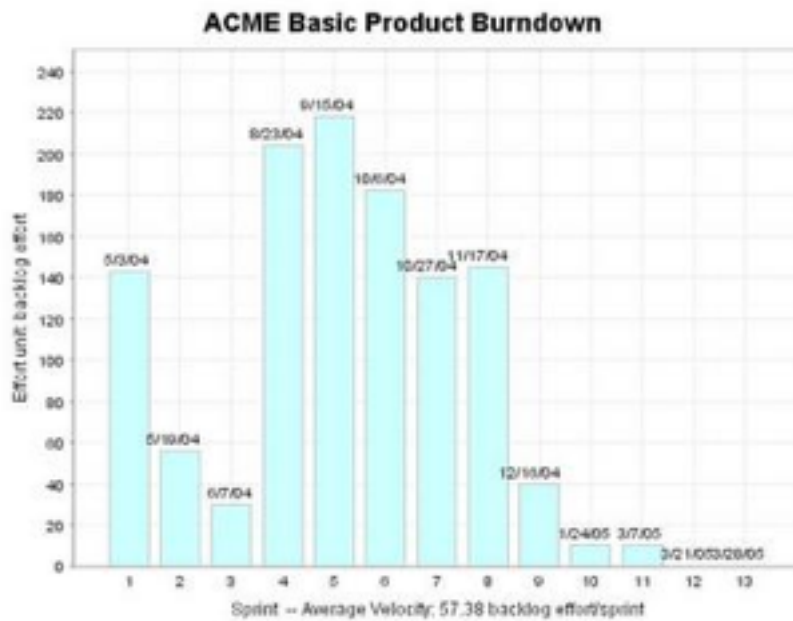


Figure 7: Product Burn down Chart [13]

7 Scaling Scrum

A typical Scrum team consists of 6-10 people but Jeff Sutherland has scaled Scrum up to over 800 people. The primary way of scaling Scrum to work with large teams is to coordinate a "Scrum-of-Scrums" or a so called "Meta-Scrum". With this approach each Scrum team proceeds as normal but each team also contributes one person who attends Scrum-of-Scrum-meetings to coordinate the work of multiple Scrum teams. These meetings are analogous to the Daily Scrum meeting but do tend to happen weekly rather than daily.

Figure 8 shows how a Scrum-of-Scrums approach allows Scrum to scale up (in this case to

243 people). Each cell represents one person on a Scrum team. The bottom of this illustration shows teams with nine developers on them. One person from each team (the differently coloured cell) also participates in a Scrum-of-Scrum to coordinate work above that team. Then from those nine-person teams another person is selected (this time shown with diagonal lines) to participate in what might be called a Scrum-of-Scrums-of-Scrums.[7]

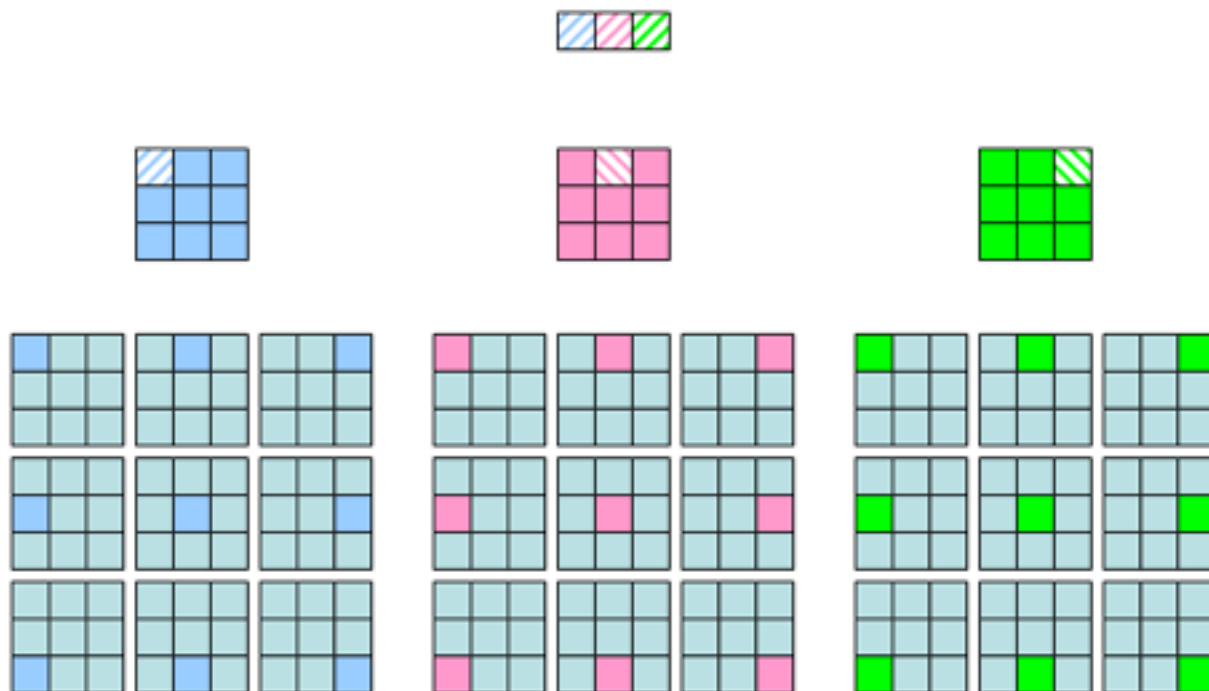


Figure 8: Scaling Scrum

8 XP@Scrum

Scrum has been employed successfully as a management wrapper for Extreme Programming engineering practices. Scrum provides the agile management mechanisms; Extreme Programming provides the integrated engineering practices (see figure 9). An article written by Ken Schwaber and Kane Mar on one implementation is at the Prentice Hall InformIT website.

Benefits of XP@Scrum include:

- The agile management and control mechanisms of Scrum are applicable for any type of project, including business initiatives that consist of multiple, simultaneous software development, business development, re-engineering, marketing, support, and implementation projects. XP@Scrum projects fit within the overall management framework of these initiatives.
- XP@Scrum projects realize the full benefits of self-organization; teams are iteration (or Sprint) goal directed, rather than story directed.
- When Extreme Programming projects are wrapped by Scrum, they become scalable and can be run simultaneously by non-located teams.
- Scrum implements in a day; Extreme Programming can be gradually implemented within the Scrum framework.

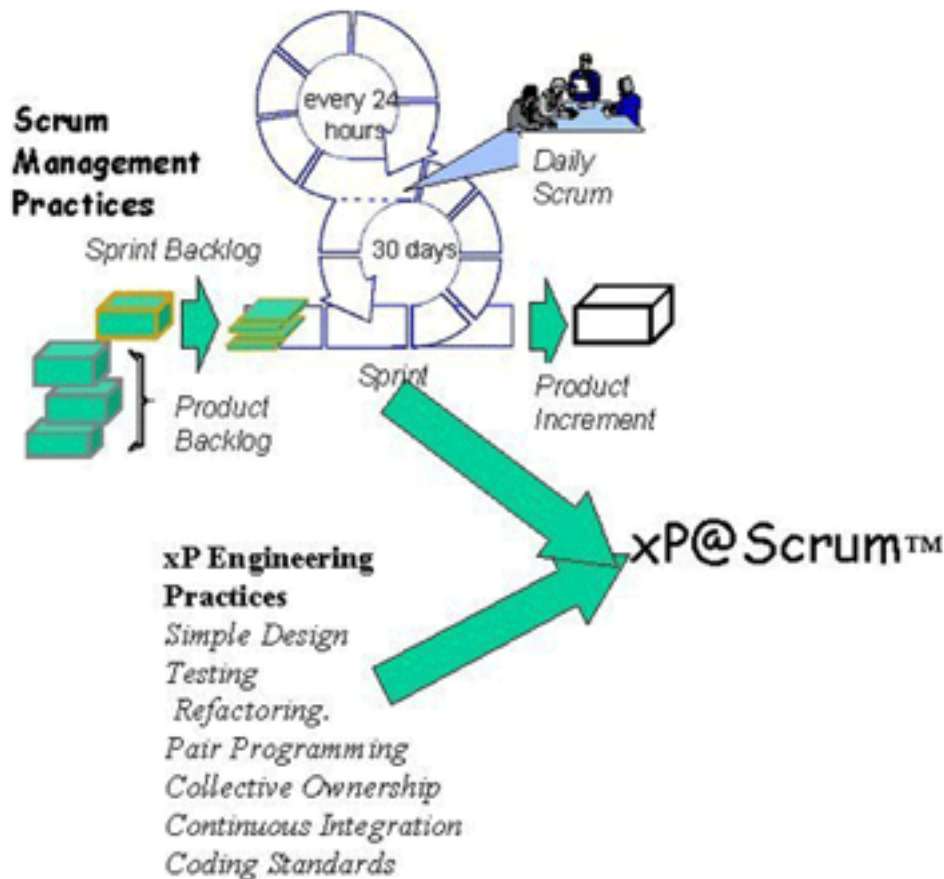


Figure 9: Scrum Management Practices

- XP@Scrum projects benefit from ADM's business value metrics process for measuring and managing initiative ROI.

9 Conclusion

SCRUM is an agile, lightweight process for managing and controlling software and product development in rapidly changing environments, which can be applied for projects with different sizes. The basic principles of Scrum are:

- Small teams, to increase communication and exchange of information
- Flexibility and adaptability concerning changes in technology and customer requirements
- Frequent development of executable product version, which can be examined, adjusted, tested and documented
- Partition of work into small, independent subtasks
- The possibility to declare anytime a project as finished (for reasons of time, finance, competition, ...)

The power of this method is due to its simplicity and its clearly defined rules. Scrum and its main features can be understood by everyone already after a short time. Assignment of responsibility to everyone in the team results in significant productivity increase. Auspicious sounds the combination of Scrum with Extreme Programming.

Currently Scrum is mainly carried by its inventors. To what extent this method can persevere after their retirement, can not be answered definitely today. However the simple composition, and the organization of Scrum Masters in a community with clearly defined rules, provides the necessary conditions for a widely spread and the survival of Scrum.

References

- [10] <http://www.scrumalliance.org/index.php/content/download/>.
- [11] "Scrum FAQ", Ken Schwaber, 2006.
- [12] <http://blogs.imason.com/chris.chapman/archive/2006/02/09/3000.aspx>.
- [13] http://www.scrumalliance.org/index.php/scrum_alliance/for_everyone/learn/glossary_of_scrum_terms.
- [14] "Future of Scrum" Jeff Sutherland.
- [2] <http://www.agilemanifesto.org/>.
- [3] <http://www.agilealliance.org>.
- [4] <http://www.acm.org/sigs/sigplan/oopsla/oopsla96/oopsla96.html>.
- [5] <http://www.controlchaos.com/>.
- [7] <http://www.mountaingoatsoftware.com/scrum/index.php>.
- [9] <http://blogs.imason.com/chris.chapman/archive/2006/02/06/2971.aspx>.