HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

h_da

Agile Software Development

Part 8: Scrum

Prof. Dr. A. del Pino

Scrum

The New New Product Development Game: From relay race ...

High quality at low cost is not the only desireable goal for todays software product development. In competitive markets, speed and flexibility start play an increasingly more important role too.

> This new emphasis on speed and flexibility calls for a different approach for managing new product development. The traditional sequential or „relay race“ approach to product development [...] may conflict with the goals of maximum speed and flexibility.
>
> Source: H. Takeuchi, I. Nonaka: *The New New Product Development Game*. Harvard Business Review, January-February 1986

- What is the relationship between software development and a relay race ?
- What drives the factors speed and flexibility ?

Scrum

...To Rugby

Image source: www.scrum.com

Instead, a holistic or „rugby“ approach – where a team tries to go the distance as a unit, passing the ball back and forth – may better serve today's competitive requirements.

Source: H. Takeuchi, I. Nonaka: *The New New Product Development Game*. Harvard Business Review, January-February 1986

Scrum

What is the rugby approach ?

Under the rugby appoach, the product development process emerges from the constant interaction of a hand-picked, multidisciplinary team whose members work together from start to finish. Rather than moving in defined, highly structured stages, the process is born out of the team members' interplay.

Source: H. Takeuchi, I. Nonaka: *The New New Product Development Game*. Harvard Business Review, January-February 1986

Scrum

What is Scrum ?

A variation on Sashimi, an "all-at-once" approach to software engineering. Both Scrum and Sashimi are suited best to new product development rather than extended development.

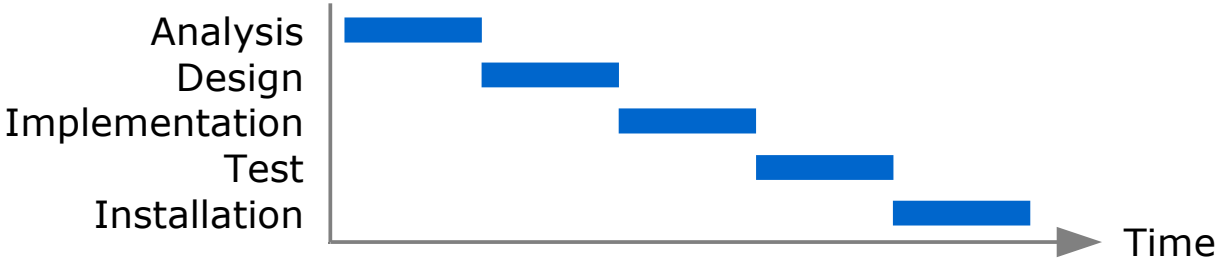
Sashimi originated with the Japanese and their experiences with the Waterfall model. They had the same problems with the Waterfall model as everybody else, so they adapted it to suit their own style. [...]

Other companies took Sashimi one step further, reducing the phases to one and calling it Scrum.

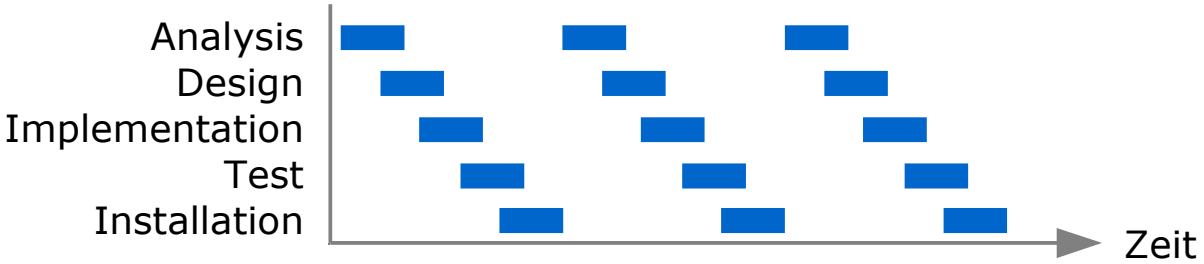
Source: http://www.controlchaos.com/download/Scrum%20Explanation.pdf

Scrum

Waterfall, Sashimi, and Scrum



Serial development
„Waterfall“



Overlapping, or concurrent development
„Sashimi“

All-at-Once
„Scrum“

Scrum

Scrum

Noise in systems development

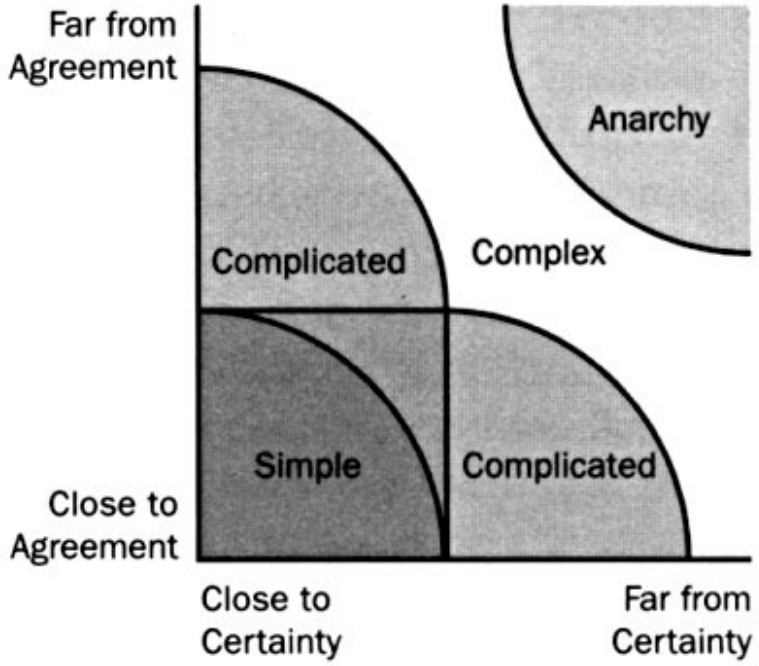


Image source: K. Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2004

Noise in systems development is a function of the three vectors of requirements, technology and people.

Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002

Scrum

Empirical process control

It is typical to adopt the defined (theoretical) modeling approach when the underlying mechanisms by which a process operates are reasonably well understood. When the process is too complicated for the defined approach, the empirical approach is the appropriate choice.

Source: B. A. Ogunnaike, W. H. Ray. *Process Dynamics, Modeling, and Control*. Oxford University Press, 1994

Is software development too complicated for a defined approach ? If yes, why ?

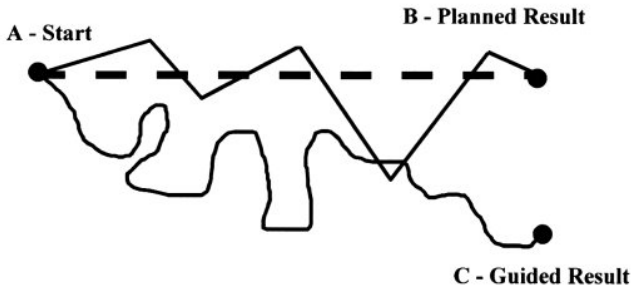


Image source: J. Highsmith. *Adaptive Software Development*. Dorset House Publishing, 2000, pg. 43

Scrum

Excursion: What are impediments and backlogs ?

Over time, an own vocabulary, Agilese, has evolved in agile projects. Impediments (ger.: Hindernisse) are in more traditional projects known as issues, problems, or risks. A backlog is essentially a to do list, and so on.

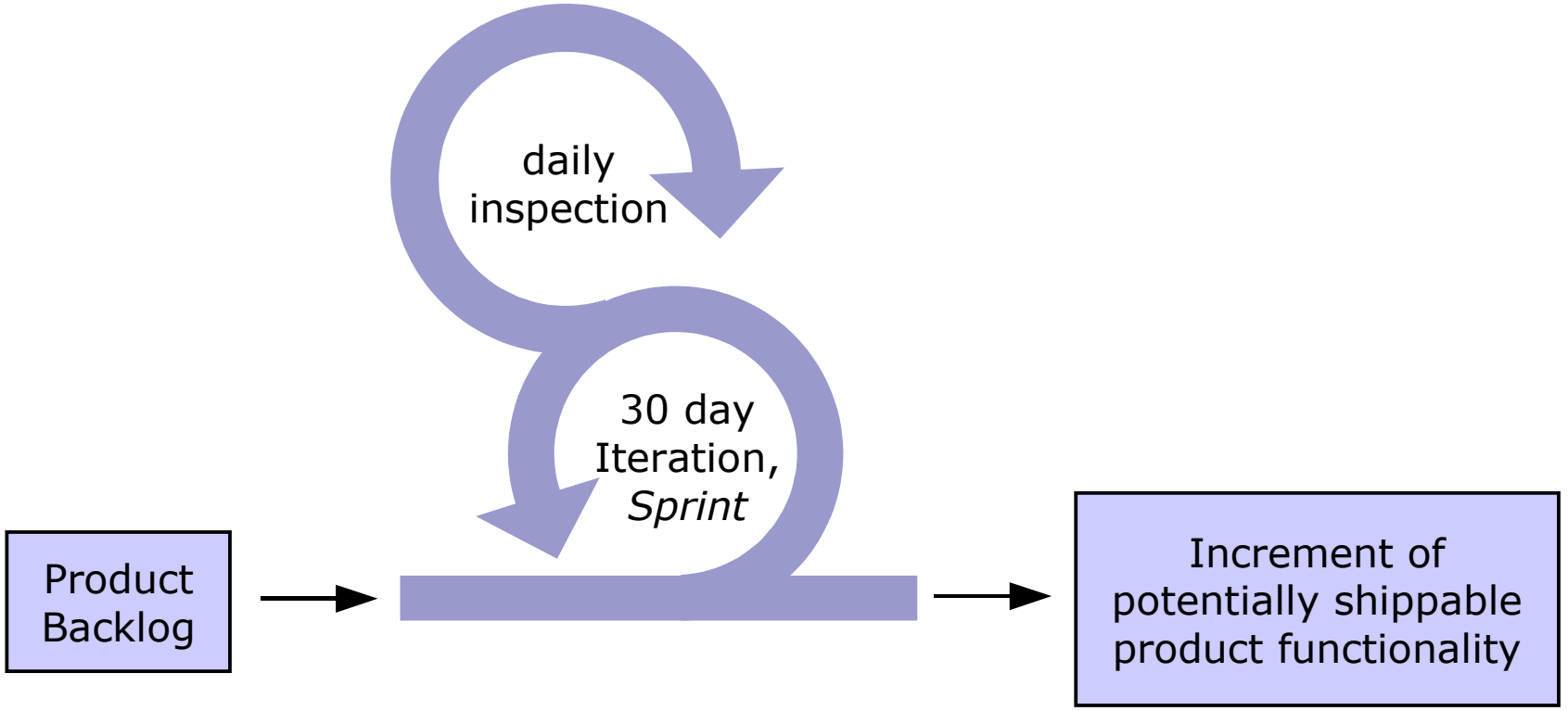
P. Kruchten compiled a list of common Agilese terms.

Vanilla	Agilese
Need, requirement, use case, scenario	User story, saga, epic
Phase, iteration	Sprint
Unscheduled work items	Backlog
Rework, redesign, modification, scrap	Refactoring
Postmortem, debriefing, end-of-project review	Retrospective
Progress meeting	Scrum, standup meeting, huddle
Productivity	Velocity
Risk, issue, constraint	Impediment
Team	Pod, cell

Source: P. Kruchten. *Voyage in the Agile Memeplex*. ACM Queue, Vol. 5, No. 5, July/August 2007

Scrum

Principle



Scrum

Scrum Roles – The Scrum Master

Scrum knows three roles. The Scrum Master, the Product Owner, and the Team.

> The Scrum Master is responsible for the success of Scrum.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 31

Responsiblities include the following:

> The ScrumMaster is responsible for the Scrum process, for teaching Scrum to everyone involved in the project, for implementing Scrum so that it fits within an organizations culture and still delivers the expected benefits, and for ensuring that everyone follows Scrum rules and practices.
>
> Source: Ken Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2004, pg. 7

Assume a side discussion starts between two developers at the Daily Scrum Meeting. What is expected to happen next ?

Scrum

The Scrum Master

An important responsibility of the Scrum Master is to make decisions.

> When decisions need to be made in the Daily Scrum, the Scrum Master is responsible for making the decisions immediately, even with incomplete information. I've found that it's usually better to proceed with some decision than no decision. The decision can always be reversed later, but in the meantime, the team can continue working.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 32

Scrum

The Scrum Master

The Scrum Master is also responsible to remove impediments, either personally, or through someone else.

> If it's likely that many impediments will have to be initially removed, this position may need to be filled by a senior manager or a Scrum consultant. [...] Removing impediments requires determination and stubbornness.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 32

Further responsibilities of the Scrum Master include:

- To represent the project team to the management and vice versa.
- To work with management and customers to identify a product owner.

Scrum

The Product Owner

> The Product Owner is responsible for representing the interests of everyone with a stake in the project and its resulting system. The Product Owner achieves initial and ongoing funding for the project by creating the projects initial overall requirements, return on investment (ROI) objectives, and release plans.
>
> Source: Ken Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2004, pg. 6

In other words:

> This is the person who is officially responsible for the project.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 34

What is the difference to the Scrum Master. Isn't the Scrum Master responsible for the success of the project ?

Scrum

The Product Owner manages the Product Backlog

> Only one person is responsible for managing and controlling the Product Backlog. This person is referred to as the Product Owner.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 34

In particular, *managing and controlling the product backlog* means:

> The Product Owner is responsible for using the Product Backlog to ensure that the most valuable functionality is produced first and built upon; this is achieved by frequently prioritizing the Product Backlog to queue up the most valuable requirements for the next iteration.
>
> Source: Ken Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2004, pg. 7

Scrum

Prioritizing the product backlog (Kano model)

Classification of features into 3 classes:

- Must-have features
- Linear features
- Attractive, exciting features

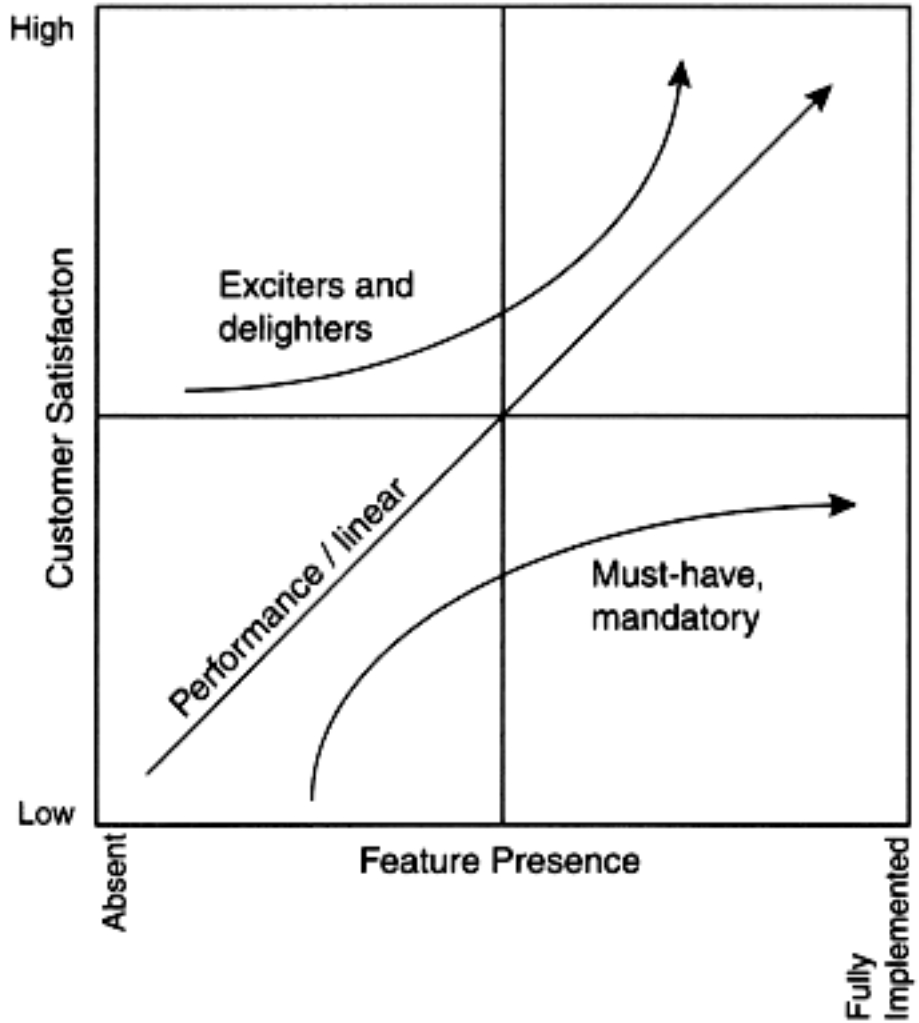


Image Source: M. Cohn. *Agile Estimating and Planning*. Prentice-Hall, 2006, pg. 113

Scrum

The Product Owner estimates items in the Product Backlog

> As backlog is created, the Product Owner works with others to estimate how long it will take to develop. To reach the estimate, he or she talks to the developers, technical writers, quality control staff, and other people who understand the product and technology.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 35

Estimatation is iterative in its nature.

> Estimates change as more information emerges about the backlog item and the item becomes better understood.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 35

To which degree is such an estimate binding to the team ?

Scrum

Can the Product Owner be a Committee ?

> The Product Owner is one person, not a committee. Committees may exist that advise or influence this person, but any person or body of people wanting an item's priority changed has to convince the Product Owner to make the change.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 34

What can happen if this is not the case ?

> [O]nly one person is responsible for maintaining and sustaining the content and priority of a single Product Backlog. Otherwise, multiple conflicting lists flourish and the Scrum teams don't know which list to listen to.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 34

Scrum

The Team

> The Team is responsible for developing functionality.
>
> Source: Ken Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2004, pg. 7

In particular, this means:

> The Scrum Team commits to turn a selected set of Product Backlog into a working product. The Scrum team makes this commitment every Sprint. [...] The team is responsible for meeting the goal to which it commits at a Sprint planning meeting. The amount of backlog it will address is solely up to the team.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 35

Scrum

Teams are self-managing and self-organized

> The team is accorded full authority to do whatever it decides is necessary to achieve the goal. [...] It is only constrained by organizational standards and conventions. Show it what to do, and it will figure out how to do it.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 35

A known side effect:

> A team often goes through a short period during which it doesn't understand that it has full authority. It too is shocked and incredulous to find out that nobody else is going to tell it what to do. This surprise quickly disappears and the productivity of self-organization takes hold.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 38

Scrum

Teams are cross-functional

A Scrum Team should include people with all of the skills necessary to meet the Sprint goal. Scrum eschews vertical teams of analysts, designers, quality control, and coding engineers. A Scrum Team self-organizes so that everyone contributes to the outcome. Each team member applies his or her expertise to all of the problems. The resultant synergy from a tester helping a designer construct code improves code quality and raises productivity.

Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 37

Scrum

Knowledge creation in a Scrum team

How does learning take place in a self-managed, self-organizing and cross-functional team ?

> The fact that we gather requirements for an application strongly indicates that they are in tacit form, and as we make design decisions we acquire knowledge that we eventually capture in the code or in an executable model. From this perspective, software is nothing else than codified and explicit knowledge.
>
> Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 111

I. Nonaka and *H. Takeuchi* proposed a model of knowledge creation by four modes of knowledge conversion between tacit and explicit knowledge. These modes are socialization, externalization, combination, and internalization.

Scrum

Knowledge creation – Socialization (tacit to tacit)

Socialization is a process of sharing experiences and thereby creating tacit knowledge such as shared mental models and technical skills. An individual can acquire tacit knowledge directly from others without using language. Apprentices work with their masters and learn craftsmanship not through language but through observation, imitation, and practice. In the business setting, on-the-job training uses basically the same principle.

Source: I. Nonaka, H. Takeuchi. *The Knowledge-Creating Company*. Oxford University Press, 1995, pg. 62

Which agile practices support this ?

Scrum

Knowledge creation – Externalization (tacit to explicit)

Externalization is a process of articulating tacit knowledge into explicit concepts. It is a quintessential knowledge-creation process in that tacit knowledge becomes explicit [...].

When we attempt to conceptualize an image, we express its essence mostly in language [...]

The externalization mode of knowledge conversion is typically seen in the process of concept creation and is triggered by dialogue or collective reflection.

Source: I. Nonaka, H. Takeuchi. *The Knowledge-Creating Company*. Oxford University Press, 1995, pg. 64

How does externalization take place in agile development projects ?

Scrum

Knowledge creation – Combination (explicit to explicit)

Combination is a process of systemizing concepts into a knowledge system. This mode of knowledge conversion involves combining different bodies of explicit knowledge. Individuals exchange and combine knowledge through such media as documents, meetings, telephone conversations, or computerized communication networks. Reconfiguration of existing information through sorting, adding, combining, and categorizing of explicit knowledge [...] can lead to new knowledge.

Source: I. Nonaka, H. Takeuchi. *The Knowledge-Creating Company*. Oxford University Press, 1995, pg. 67

Examples for combination in agile development projects ?

Scrum

Knowledge creation – Internalization (explicit to tacit)

Internalization is a process of embodying explicit knowledge into tacit knowledge. It is closely related to "learning by doing." When experiences through socialization, externalization, and combination are internalized into individuals' tacit knowledge bases in the form of shared mental models or technical know-how, they become valuable assets. [...] For explicit knowledge to become tacit, it helps if the knowledge is verbalized or diagrammed into documents, manuals, or oral stories.

Source: I. Nonaka, H. Takeuchi. *The Knowledge-Creating Company*. Oxford University Press, 1995, pg. 69

When does internalization take place in agile projects ?

Scrum

Knowledge creation – The Knowledge Spiral

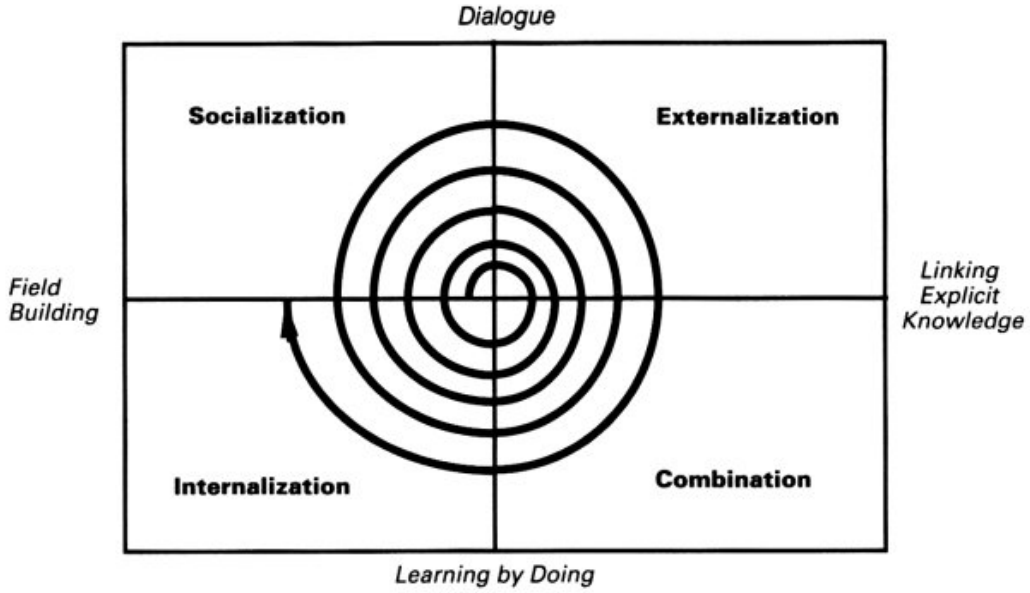


Image source: I. Nonaka, H. Takeuchi. *The Knowledge-Creating Company*. Oxford University Press, 1995, pg. 71

For organizational knowledge creation to take place, however, the tacit knowledge accumulated at the individual level needs to be socialized with other organizational members, thereby starting a new spiral of knowledge creation.

Source: I. Nonaka, H. Takeuchi. *The Knowledge-Creating Company*. Oxford University Press, 1995, pg. 69

Scrum

Team members

Scrum avoids people who refuse to code because they are systems architects, or designers. Everyone chips in and does his or her best, doing or learning how to do what is needed. Scrum Team members don't have job descriptions other than doing the best possible.

Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 38

Agile methods require a critical mass of highly motivated, knowledgeable team members. Since documentation and design are kept to a minimum, the ability for team members to retain and act on tacit knowledge is crucial.

Source: B. Boehm, R. Turner. *Balancing Agility and Discipline*. Addison-Wesley, 2004, pg. 20

Scrum

The chicken and the pig

Team members commit to a goal and do the work that is required to meet it. They are called pigs because they, like the pigs in the joke, are committed to the project. Everyone else is a chicken. Chickens can attend Daily Scrums, but they have to stand on the periphery.

Source: K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002, pg. 42

Scrum

Team members

This distinction is important in Scrum and is relevant to Scrum's insistance upon total visibility. It should always be clear who is on the hook and who is just a kibitzer. Who is responsible for the ROI, and who has a stake in the ROI, but isn't accountable. Who has to turn difficult technology into functionality, and who is a troublesome „devil's advocate“?

The rules of Scrum distinguish between the chickens and the pigs to increase productivity, create momentum, and put an end to floundering.

Source: K. Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2004, pg. 7

Scrum

The Sprint Planning Meeting

Each *Sprint* (30 calendar days) starts with a *Sprint planning meeting* which lasts eight hours.

Part 1 - first 4 hours

- The product owner presents to the team the items in the *Product Backlog* with the highest priority.
- During this presentation the team can ask questions to get a better understanding.
- Before the first part of the meeting is over, the team selects as much items from the *Product Backlog* as it deems possible to transform into added product functionality during the next sprint.

Part 2 - last four hours

- The team creates an initial *Sprint Backlog* from the selected items which it commited to implement.

Scrum

The Sprint Planning Meeting

- Story cards vs. computerized system: Who has control over the keyboard ?
- Selecting stories, but not allocating them to individual team members
- Stories are broken down into tasks, which are estimated by ideal hours.
- Try to identify all tasks including testing, etc.
- Add a spike for tasks which are difficult to estimate

> - Determine what's affected-two hours.
> - Make the changes-ten hours.
>
> This first task is called a spike. A spike is a task included in an iteration plan that is being undertaken specifically to gain knowledge or answer a question.
>
> Source: M. Cohn. *Agile Estimating and Planning*. Prentice-Hall, 2006, pg. 156

Scrum

Velocity- vs. Commitment-driven Iteration Planning

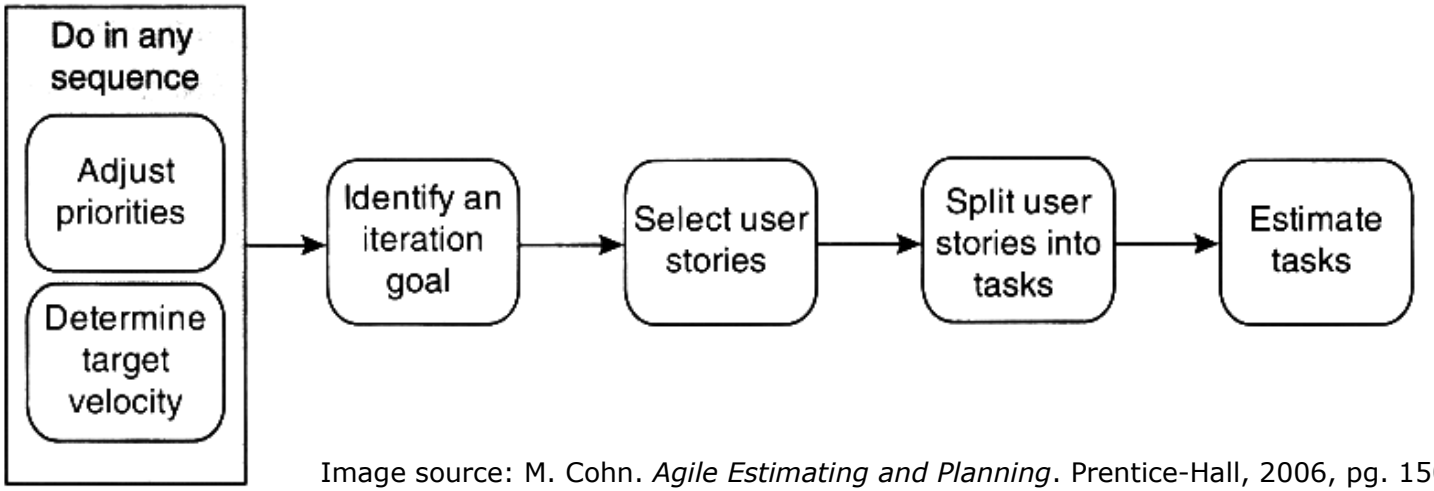


Image source: M. Cohn. *Agile Estimating and Planning*. Prentice-Hall, 2006, pg. 150

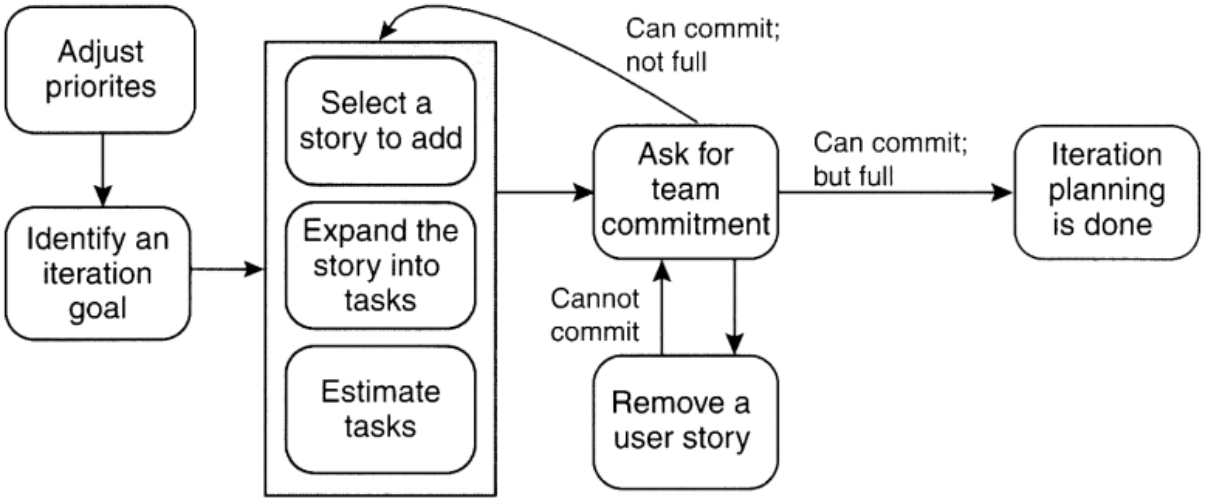


Image source: M. Cohn. *Agile Estimating and Planning*. Prentice-Hall, 2006, pg. 159

Scrum

The Daily Scrum Meeting

The *Daily Scrum meeting* takes 15 minutes and its objectives are to synchronize the team and to identify any issues which need to be discussed in a separate meeting.

- Every day, same place, same time.
- Only one person talks – No side discussions.
- All team members must attend the Daily Scrum.
- Chickens are not allowed to talk during or disturb the Daily Scrum, otherwise they will be excluded from it.

During the Daily Scrum, each team member answers the following three questions in a brief and concise manner.

- What have you done for this project since the last Daily Scrum meeting ?
- What will you do for this project until the next Daily Scrum meeting ?
- Which stumbling blocks might prevent you from reaching your commitments to this Sprint and to this project ?

Scrum

The Sprint Review Meeting

The *Sprint review meeting* is a four hours time-boxed meeting and takes place at the end of a Sprint to collectively determine what the team should do next.

- During this meeting the team presents to the product owner and other stakeholders the functionality which had been added to the product during the last Sprint.

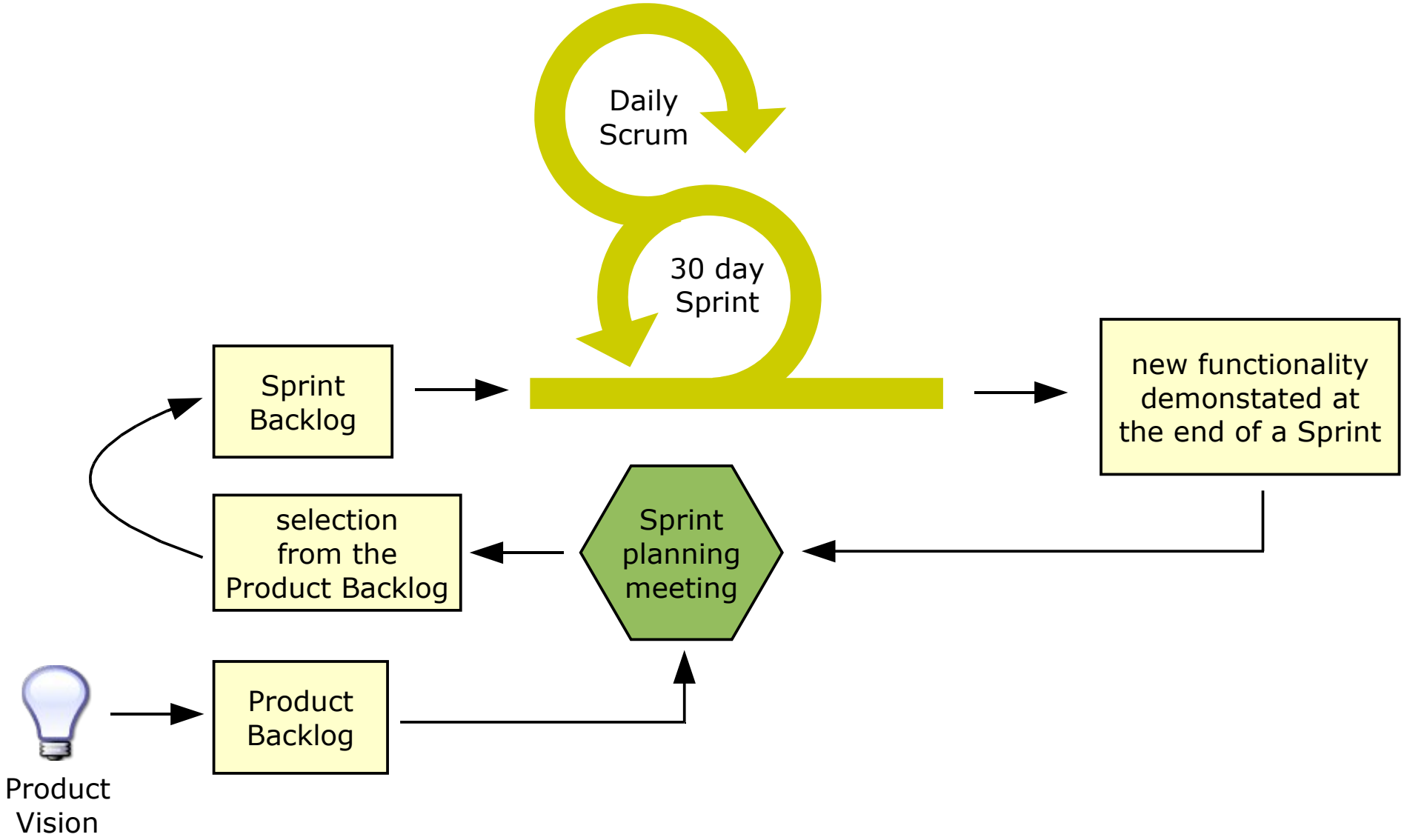
The Sprint Retrospective Meeting

After the Sprint review meeting and before the next Sprint planning meeting, the ScrumMaster holds a three hours time-boxed *Sprint retrospective meeting* with the team.

Its purpose is to identify opportunities for improvement for the next sprint, to make the work more effective and enjoyable.

Scrum

The Scrum Process



Scrum

The Product Backlog

The *Product Backlog* is a list of prioritized requirements which is developed and maintained by the product owner. It is an *as-is* document and therefore never complete.

Requirements for Sprint 1,

(First entry has the highest priority)

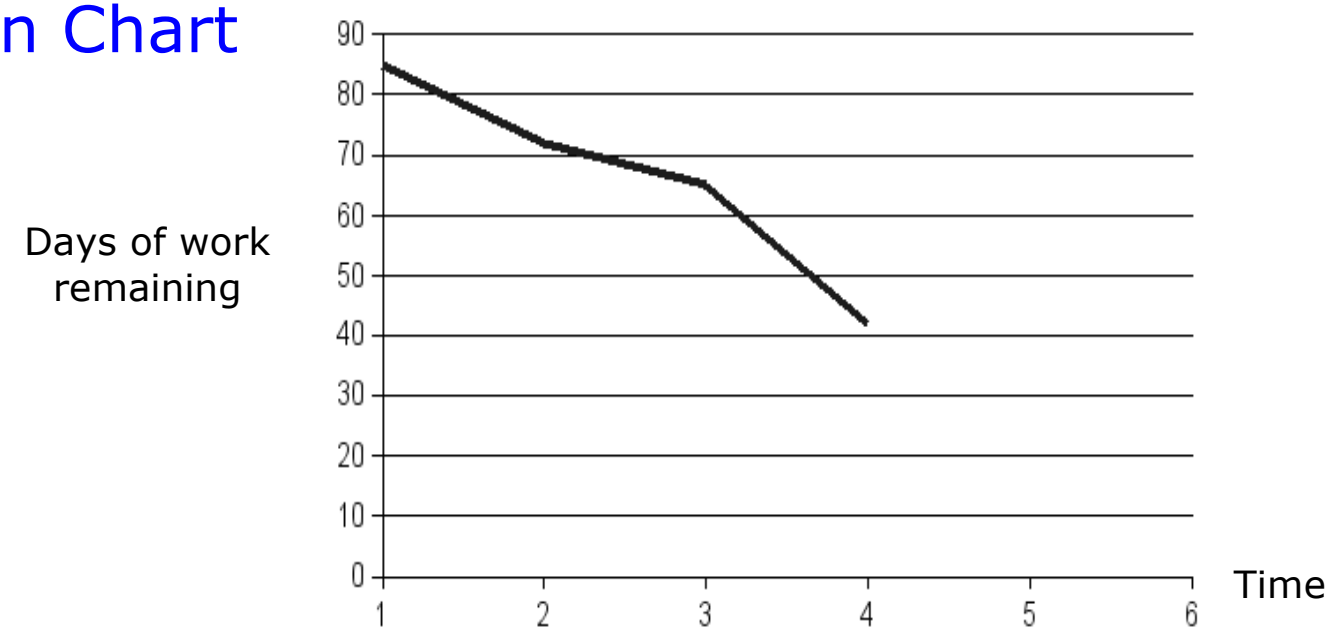
Requirements for Sprint 2

Item Title	Initial Estimate	Adjustment Factor	Adjusted Estimate	Remaining Work until completion of Sprint#				
				1	2	3	4	5
Display tree	3	0,2	3,6	3,6				
Recalculate values	2	0,2	2,4	2,4				
…								
SPRINT 1	**5**		**6**	**6**				
Save tree to file	1,5		1,5		1,5			
Recalculate values	1		1		1			
…								
SPRINT 2	**2,5**		**2,5**	**0**	**2,5**			
…								

Not completed in Sprint #1, continued in Sprint#2.

Scrum

The Burndown Chart



The *burndown chart* shows how much work still remains to be done over a time axis. It is maintained by the ScrumMaster. The intersection of a trend line with the time axis gives an estimate when the work will approximately be done.

Ken Schwaber:

> *The burndown chart is the collision of reality (work done and how fast it's being done) with what is planned, or hoped for.*
>
> Source: K. Schwaber: „Agile Project Management with Scrum", Microsoft Press, 2004

Scrum

The Sprint Backlog

The *Sprint Backlog* defines the necessary tasks to transform selected entries from the Product Backlog into shippable product functionality.

- It is initially created in the second part of the Sprint planning meeting and maintained by the team.
- Tasks should have a granularity between a half and two working days.
- Tasks with a longer duration can temporarily be added to the Sprint Backlog, but must be refined at a later time.

Example

			Remaining hours of work in this Sprint																			
Task Description	**Responsible**	**Status**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Save as PDF file		In Progress	16	16	16	16	8	8	8	8												
Export as XML		Done	8	8	8	8	8	0	0	0												
Switch to JDK 1.5		Not yet started	64	64	64	64	64	64	64	64												
...																						

Scrum

Literature

Books (both books are available at our library)

- K. Schwaber, M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002
- K. Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2004

Papers

- J. Sutherland. *Agile Can Scale: Inventing and Reinventing SCRUM in Five Co*mpanies. Cutter IT Journal, Vol. 14, No. 12, 2001

  http://www.jeffsutherland.com/papers/scrum/Sutherland2001AgileCanScaleCutter.pdf

Web sites

- http://www.controlchaos.com
- http://www.jeffsutherland.com