

# Lecture on Scrum

Alessandro Bocci

Advanced Software  
Engineering

29/09/2022

# Motivations

- Software company managers need information that will help them understand how much it costs to develop a software product, how long it will take and when the product can be brought to market.
- Plan-driven development provides this information through long-term development plans that identify deliverables - items the team will deliver and when these will be delivered.
- Plans always change so anything apart from short-term plans are unreliable.

# Scrum

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

The Scrum Framework contains a set of principles and rules to following order to achieve a common goal.

# Scrum Theory

Scrum is founded on

- ***Empiricism***: it asserts that knowledge comes from experience and making decisions based on what is observed
- ***Lean thinking***: it reduces waste and focuses on the essentials.

Scrum employs an iterative, incremental approach to optimize predictability and to control risk.

# Scrum Theory - Pillars

- ***Transparency:***

The emergent process and work must be visible to those performing the work as well as those receiving the work.

- ***Inspection***

Scrum artifacts and the progress toward agreed goals must be inspected frequently and diligently to detect potentially undesirable variances or problems.

- ***Adaptation***

If any aspects of a process deviate outside acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced must be adjusted. The adjustment must be made as soon as possible to minimize further deviation.

# Scrum Theory - Values

Successful use of Scrum depends on people becoming more proficient in living five values:

- Commitment
- Focus
- Openness
- Respect
- Courage

# Scrum in practice

- ***Scrum team***: Product owner, Scrum Master, Developers
- ***Artifacts***: Product backlog, Sprint backlog
- ***Scrum events***: Sprint planning, execution and review

# Key roles in Scrum

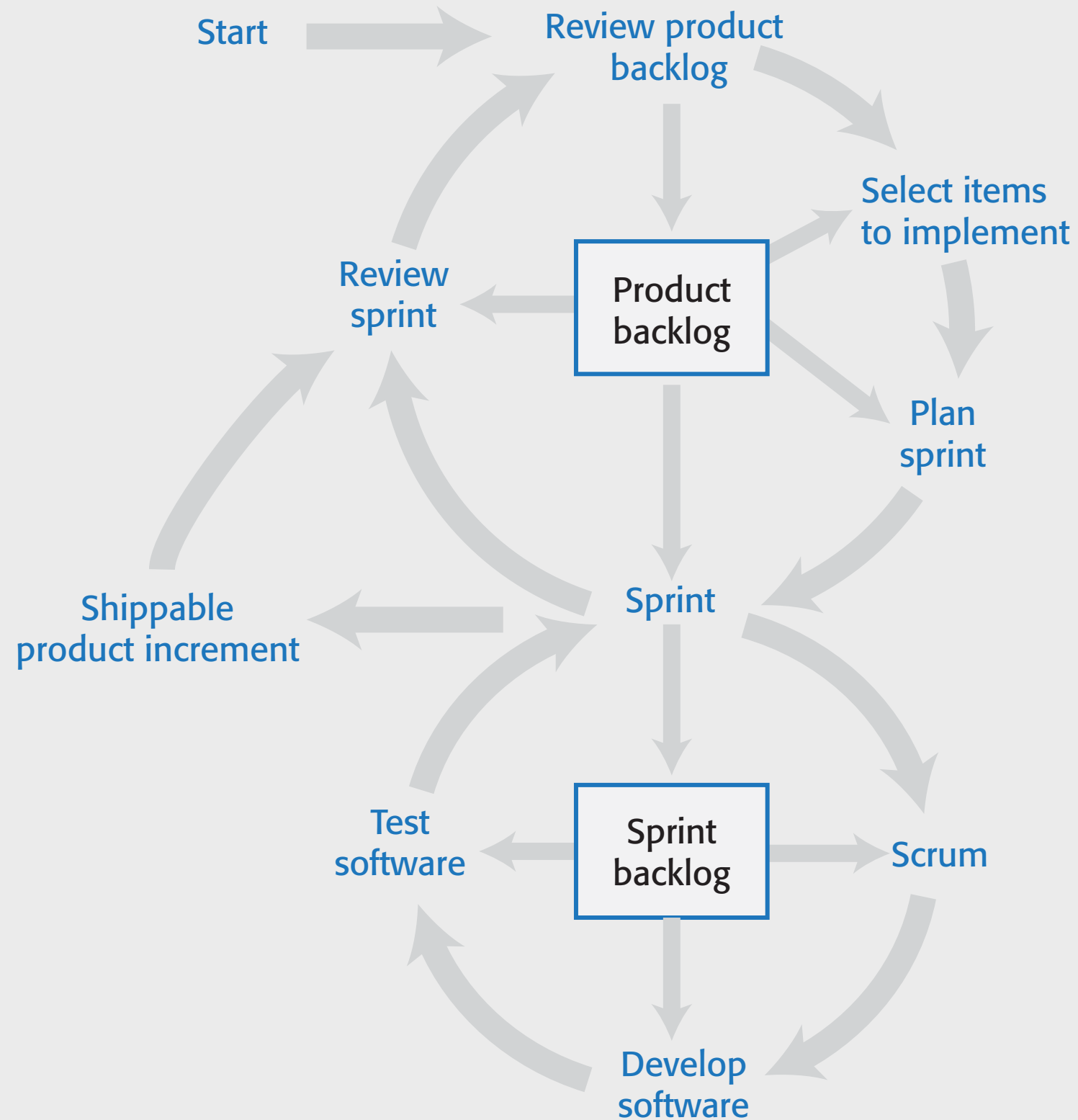
- ***The Product Owner*** is responsible for ensuring that the development team are always focused on the product they are building rather than diverted into technically interesting but less relevant work.
  - In product development, the product manager should normally take on the Product Owner role.
- **The ScrumMaster** is a Scrum expert whose job is to guide the team in the effective use of the Scrum method. The developers of Scrum emphasize that the ScrumMaster is not a conventional project manager but is a coach for the team. They have authority within the team on how Scrum is used.
  - In many companies that use Scrum, the ScrumMaster also has some project management responsibilities.



# Scrum and sprints

- In Scrum, software is developed in sprints, which are fixed-length periods (2 - 4 weeks) in which software features are developed and delivered.
- During a sprint, the team has daily meetings (Scrums) to review progress and to update the list of work items that are incomplete.
- Sprints should produce a 'shippable product increment'. This means that the developed software should be complete and ready to deploy.

**Figure 2.3 Scrum cycles**



# Key Scrum practices

- ***Product backlog***

This is a to-do list of items to be implemented that is reviewed and updated before each sprint.

- ***Timeboxed sprints***

Fixed-time (2-4 week) periods in which items from the product backlog are implemented,

- ***Self-organizing teams***

Self-organizing teams make their own decisions and work by discussing issues and making decisions by consensus.

# Product backlogs

- The product backlog is a list of what needs to be done to complete the development of the product.
- The items on this list are called product backlog items (PBIs).
- The product backlog may include a variety of different items such as product features to be implemented, user requests, essential development activities and desirable engineering improvements.
- The product backlog should always be prioritized so that the items that be implemented first are at the top of the list.

**Table 2.6 Examples of product backlog items**

1. As a teacher, I want to be able to configure the group of tools that are available to individual classes. (feature)
2. As a parent, I want to be able to view my childrens' work and the assessments made by their teachers. (feature)
3. As a teacher of young children, I want a pictorial interface for children with limited reading ability. (user request)
4. Establish criteria for the assessment of open source software that might be used as a basis for parts of this system. (development activity)
5. Refactor user interface code to improve understandability and performance. (engineering improvement)
6. Implement encryption for all personal user data. (engineering improvement)

## Table 2.7 Product backlog item states

### ***Ready for consideration***

These are high-level ideas and feature descriptions that will be considered for inclusion in the product. They are tentative so may radically change or may not be included in the final product.

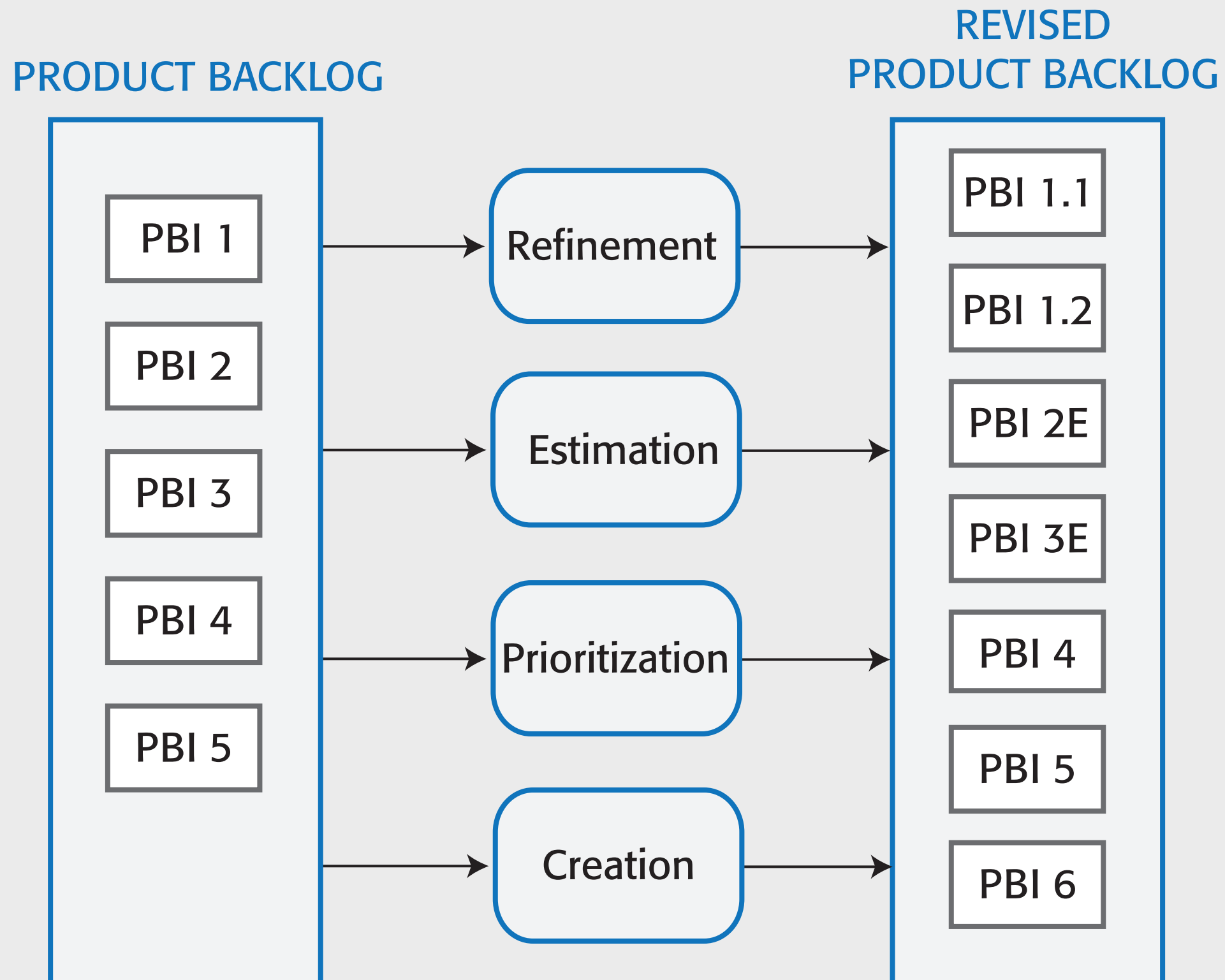
### ***Ready for refinement***

The team has agreed that this is an important item that should be implemented as part of the current development. There is a reasonably clear definition of what is required. However, work is needed to understand and refine the item.

### ***Ready for implementation***

The PBI has enough detail for the team to estimate the effort involved and to implement the item. Dependencies on other items have been identified.

**Figure 2.5 Product backlog activities**



# Product backlog activities

- ***Refinement***

Existing PBIs are analysed and refined to create more detailed PBIs. This may lead to the creation of new product backlog items.

- **Estimation**

The team estimate the amount of work required to implement a PBI and add this assessment to each analysed PBI.

- **Creation**

New items are added to the backlog. These may be new features suggested by the product manager, required feature changes, engineering improvements, or process activities such as the assessment of development tools that might be used.

- **Prioritization**

The product backlog items are reordered to take new information and changed circumstances into account.



# PBI estimation metrics

- Effort required
  - This may be expressed in person-hours or person-days i.e. the number of hours or days it would take one person to implement that PBI. This is not the same as calendar time. Several people may work on an item, which may shorten the calendar time required.
- Story points
  - Story points are an arbitrary estimate of the effort involved in implementing a PBI, taking into account the size of the task, its complexity, the technology that may be required and the 'unknown' characteristics of the work.
  - They were derived originally by comparing user stories, but they can be used for estimating any kind of PBI.
  - Story points are estimated relatively. The team agree on the story points for a baseline task and other tasks are estimated by comparison with this e.g. more/less complex, larger/smaller etc.

# Timeboxed sprints

- Products are developed in a series of sprints, each of which delivers an increment of the product or supporting software.
- Sprints are short duration activities (2-4 weeks) and take place between a defined start and end date. Sprints are timeboxed, which means that development stops at the end of a sprint whether or not the work has been completed.
- During a sprint, the team work on the items from the product backlog.

# Sprint activities

- ***Sprint planning***

Work items to be completed in that sprint are selected and, if necessary, refined to create a sprint backlog. This should not last more than a day at the beginning of the sprint.

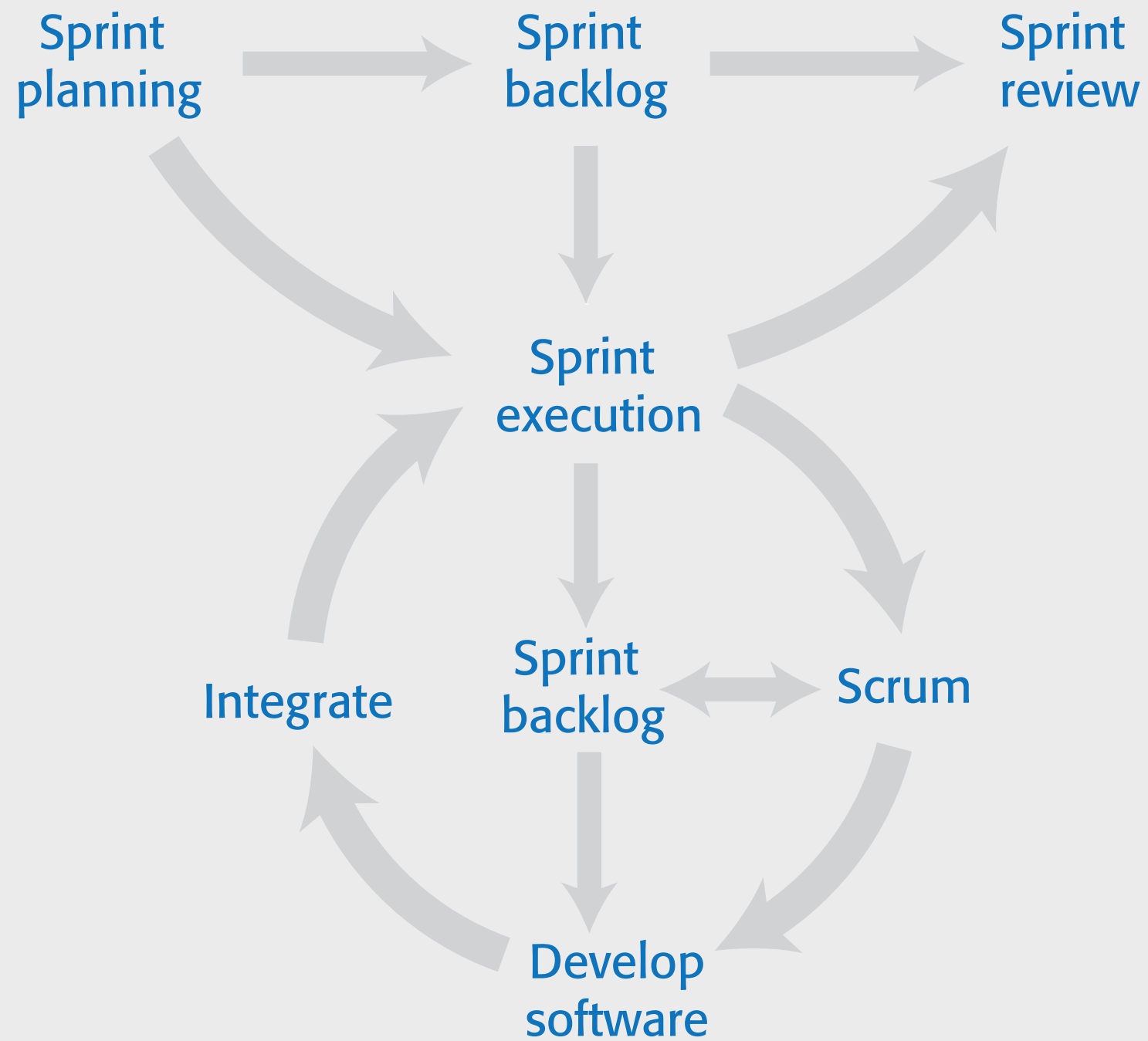
- ***Sprint execution***

The team work to implement the sprint backlog items that have been chosen for that sprint. If it is impossible to complete all of the sprint backlog items, the sprint is not extended. The unfinished items are returned to the product backlog and queued for a future sprint.

- ***Sprint reviewing***

The work done in the sprint is reviewed by the team and (possibly) external stakeholders. The team reflect on what went well and what went wrong during the sprint with a view to improving their work process.

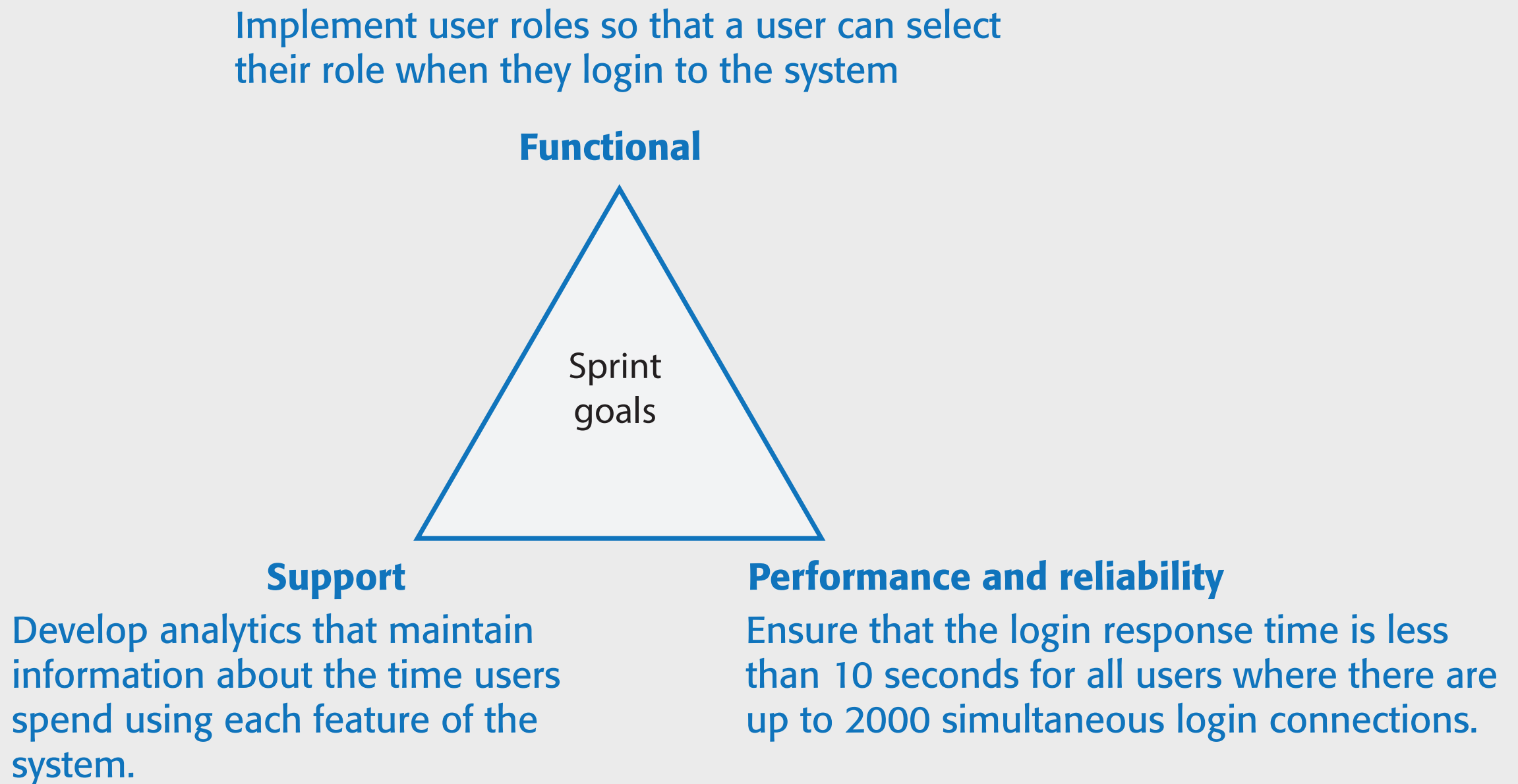
**Figure 2.7 Sprint activities**



# Sprint planning

- Establish an agreed sprint goal
  - Sprint goals may be focused on software functionality, support or performance and reliability,.
- Decide on the list of items from the product backlog that should be implemented
- Create a sprint backlog.
  - This is a more detailed version of the product backlog that records the work to be done during the sprint

**Figure 2.8 Sprint goals**



# Sprint planning

- In a sprint plan, the team decides which items in the product backlog should be implemented during that sprint.
  - Key inputs are the effort estimates associated with PBIs and the team's velocity
- The output of the sprint planning process is a sprint backlog.
  - The sprint backlog is a breakdown of PBIs to show the what is involved in implementing the PBIs chosen for that sprint.
- During a sprint, the team have daily meetings (scrums) to coordinate their work.

## Table 2.8 Scrums

A scrum is a short, daily meeting that is usually held at the beginning of the day. During a scrum, all team members share information, describe their progress since the previous day's scrum, problems that have arisen and plans for the coming day. This means that everyone on the team knows what is going on and, if problems arise, can re-plan short-term work to cope with them.

Scrum meetings should be short and focused. To dissuade team members from getting involved in long discussions, they are sometimes organized as 'stand-up' meetings where there are no chairs in the meeting room.

During a scrum, the sprint backlog is reviewed. Completed items are removed from it. New items may be added to the backlog as new information emerges. The team then decide who should work on sprint backlog items that day.



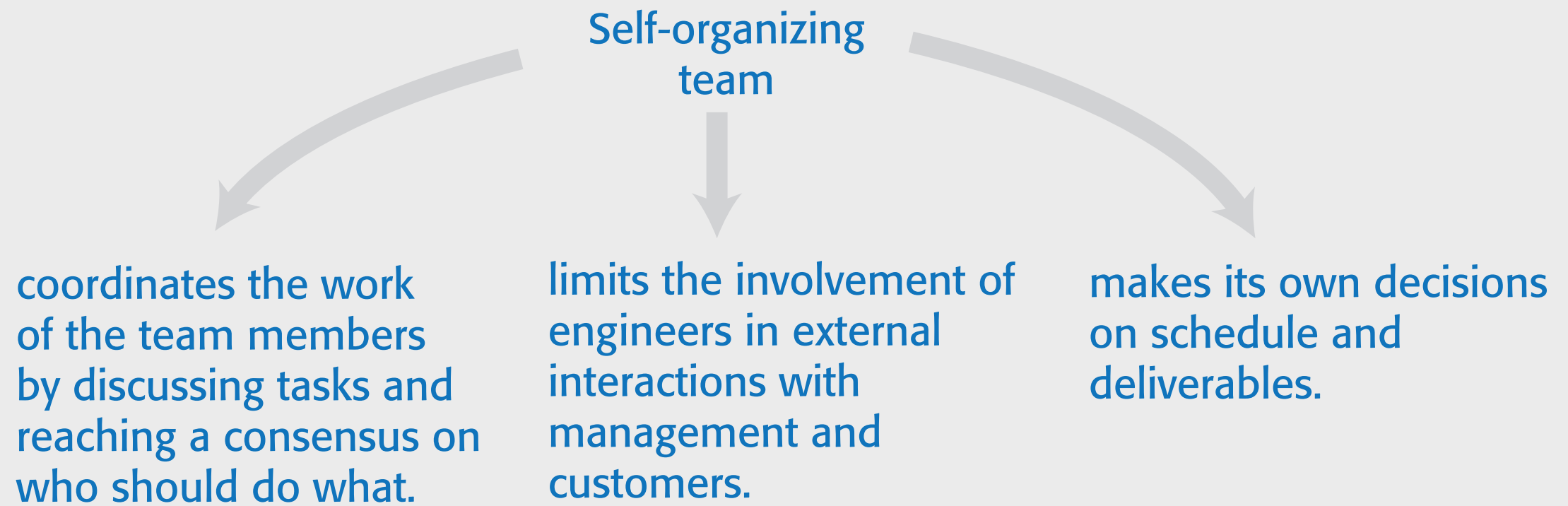
# Agile activities

- Scrum does not suggest the technical agile activities that should be used. However, there are two practices that are strongly suggested by agile software engineers to be used in a sprint.
- ***Test automation***  
As far as possible, product testing should be automated. You should develop a suite of executable tests that can be run at any time.
- ***Continuous integration***  
Whenever anyone makes changes to the software components they are developing, these components should be immediately integrated with other components to create a system. This system should then be tested to check for unanticipated component interaction problems.

# Sprint reviews

- At the end of each sprint, there is a review meeting, which involves the whole team. This meeting:
  - reviews whether or not the sprint has met its goal.
  - sets out any new problems and issues that have emerged during the sprint.
  - is a way for a team to reflect on how they can improve the way they work.
- The product owner has the ultimate authority to decide whether or not the goal of the print has been achieved. They should confirm that the implementation of the selected product backlog items is complete.
- The sprint review should include a process review, in which the team reflects on its own way of working and how Scrum has been used.
  - The aim is to identify ways to improve and to discuss how to use Scrum more productively.

**Figure 2.9 Self-organizing teams**



# Team size and composition

- The ideal Scrum team size is between 5 and 8 people.
  - Teams have to tackle diverse tasks and so usually require people with different skills, such as networking, user experience, database design and so on.
  - They usually involve people with different levels of experience.
  - A team of 5-8 people is large enough to be diverse yet small enough to communicate informally and effectively and to agree on the priorities of the team.
- The advantage of a self-organizing team is that it can be a cohesive team that can adapt to change.
  - Because the team rather than individuals take responsibility for the work, they can cope with people leaving and joining the team.
  - Good team communication means that team members inevitably learn something about each other's areas

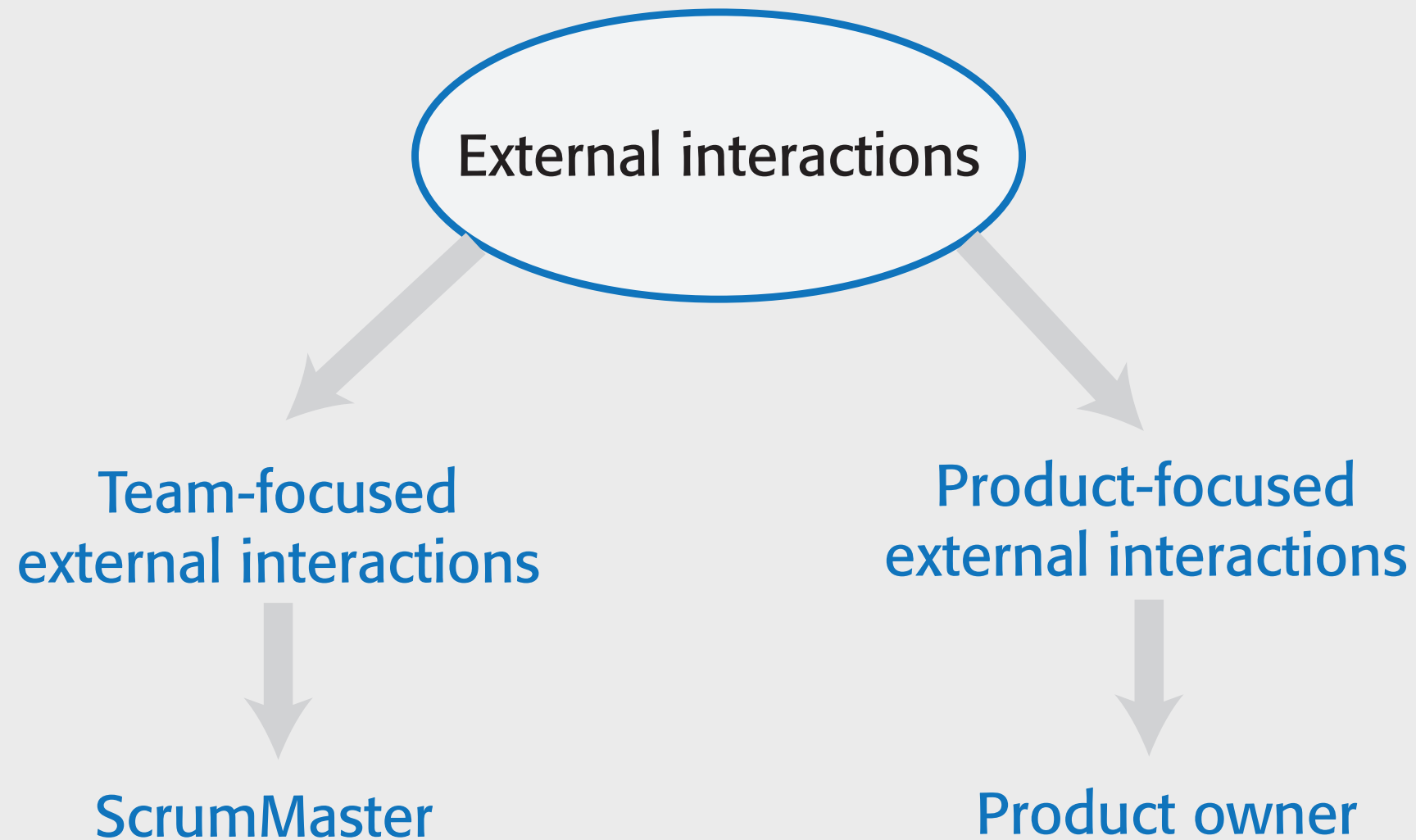
# Team coordination

- The developers of Scrum assumed that teams would be co-located. They would work in the same room and could communicate informally.
  - Daily scrums mean that the team members know what's been done and what others are doing.
- However, the use of daily scrums as a coordination mechanism is based on two assumptions that are not always correct:
  - Scrum assumes that the team will be made up of full-time workers who share a workspace. In reality, team members may be part-time and may work in different places. For a student project team, the team members may take different classes at different times.
  - Scrum assumes that all team members can attend a morning meeting to coordinate the work for the day. However, some team members may work flexible hours (e.g. because of childcare responsibilities) or may work on several projects at the same time.

# External interactions

- External interactions are interactions that team members have with people outside of the team.
- In Scrum, the idea is that developers should focus on development and only the ScrumMaster and Product Owner should be involved in external interactions.
- The intention is that the team should be able to work on software development without external interference or distractions.

**Figure 2.10 Managing external interactions**

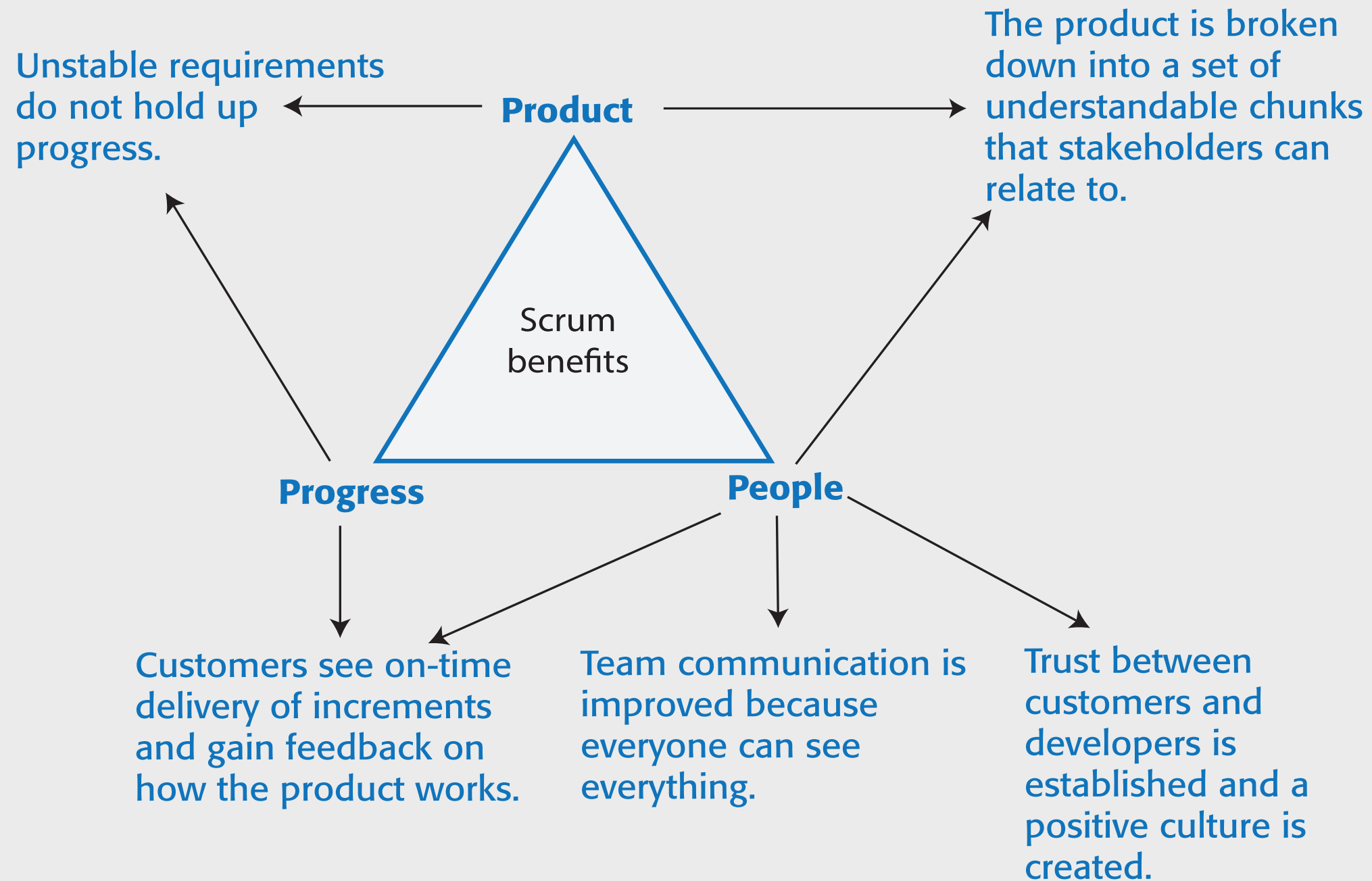


# Project management

- In all but the smallest product development companies, there is a need for development teams to report on progress to company management.
- A self-organizing team has to appoint someone to take on these responsibilities.
  - Because of the need to maintain continuity of communication with people outside of the group, rotating these activities around team members is not a viable approach.
- The developers of Scrum did not envisage that the ScrumMaster should also have project management responsibilities.
  - In many companies, however, the ScrumMaster has to take on project management responsibilities.
  - They know the work going on and are in the best position to provide accurate information and project plans and progress.



**Figure 2.4 The top five benefits of using Scrum**



# Key points

- Scrum is an agile method that focuses on agile planning and management. The development team may use any technical practices that they believe are appropriate for the product being developed.
- In Scrum, work to be done is maintained in a product backlog – a list of work items to be completed. Each increment of the software implements some of the work items from the product backlog.
- Sprints are fixed-time activities (usually 2–4 weeks) where a product increment is developed. Increments should be ‘potentially shippable’ i.e. they should not need further work before they are delivered.
- A self-organizing team is a development team that organizes the work to be done by discussion and agreement amongst team members.
- Scrum practices such as the product backlog, sprints and self-organizing teams can be used in any agile development process, even if other aspects of Scrum are not used.

# References

*'Scrum guide'*, Ken Schwaber & Jeff Sutherland, 2010-2020

*'Engineering Software Products: An Introduction to Modern Software Engineering'*, Ian Sommerville, 2018