FACULDADE ENGENHARIA
Departamento de Informática

Engenharia de Software (14341, 16230, 15386)

Agile Software Engineering

(adapted from *Software Engineering: International Version,* Ian Sommerville, Pearson, 2015, and *Engineering Software Products: An Introduction to Modern Software Engineering*, Ian Sommerville, Pearson, 2020)

Topics covered

- ✧ Agile methods
- ✧ Agile software engineering

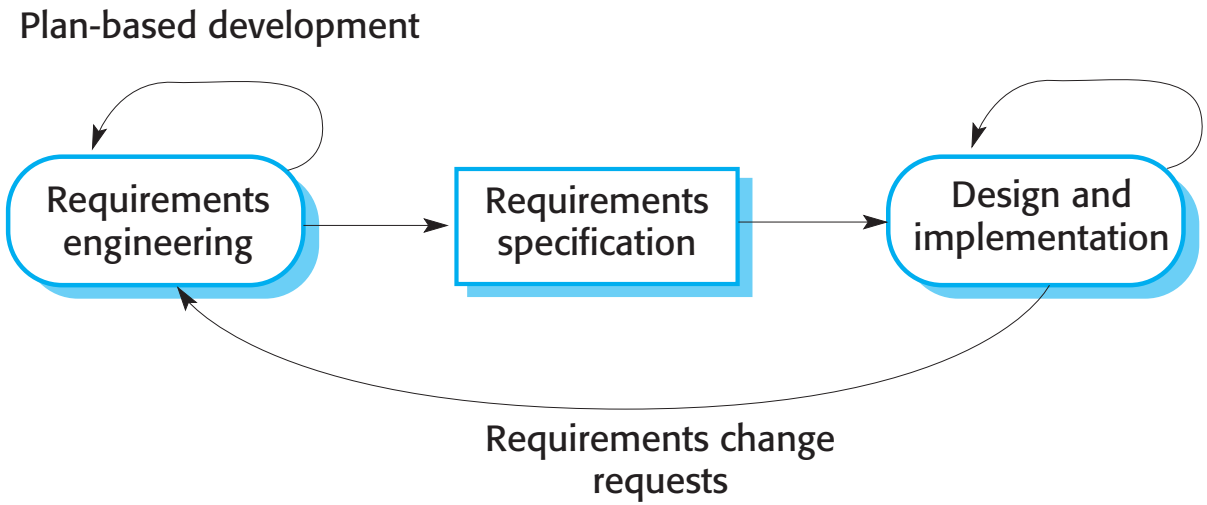
Rapid software development

- Rapid development and delivery is now often the most important requirement for software systems
  - Businesses operate in a fast–changing requirement and it is practically impossible to produce a set of stable software requirements
  - Software has to evolve quickly to reflect changing business needs.
- Plan-driven development is essential for some types of system but does not meet these business needs.
- Agile development methods emerged in the late 1990s whose aim was to radically reduce the delivery time for working software systems

Agile development

- Program specification, design and implementation are inter-leaved
- The system is developed as a series of versions or increments with stakeholders involved in version specification and evaluation
- Frequent delivery of new versions for evaluation
- Extensive tool support (e.g. automated testing tools) used to support development.
- Minimal documentation – focus on working code

Plan-driven and agile development



Plan-driven and agile development

✧ Plan-driven development

- A plan-driven approach to software engineering is based around separate development stages with the outputs to be produced at each of these stages planned in advance.
- Not necessarily waterfall model – plan-driven, incremental development is possible
- Iteration occurs within activities.

✧ Agile development

- Specification, design, implementation and testing are inter-leaved and the outputs from the development process are decided through a process of negotiation during the software development process.

Agile methods

Agile methods

- Dissatisfaction with the overheads involved in software design methods of the 1980s and 1990s led to the creation of agile methods. These methods:
  - Focus on the code rather than the design
  - Are based on an iterative approach to software development
  - Are intended to deliver working software quickly and evolve this quickly to meet changing requirements.
- The aim of agile methods is to reduce overheads in the software process (e.g. by limiting documentation) and to be able to respond quickly to changing requirements without excessive rework.

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.*

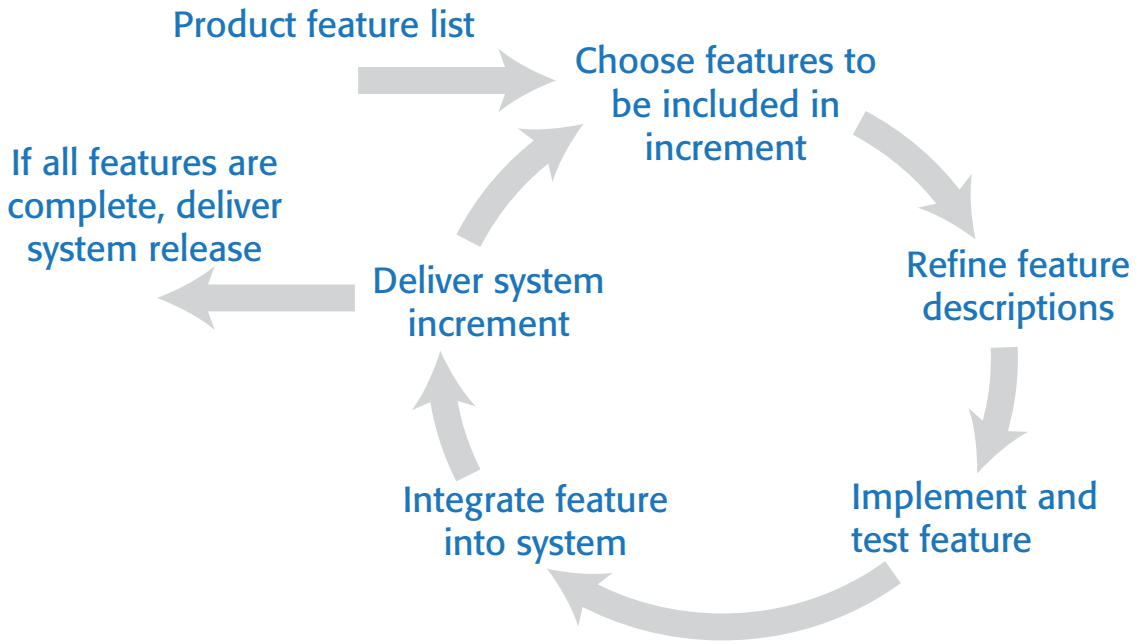
The principles of agile methods

Principle	Description
Customer involvement	Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the system.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.
Embrace change	Expect the system requirements to change and so design the system to accommodate these changes.
Maintain simplicity	Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

Incremental development

- All agile methods are based around incremental development and delivery.
- Product development focuses on the software features, where a feature does something for the software user.
- With incremental development, you start by prioritizing the features so that the most important features are implemented first.
  - You only define the details of the feature being implemented in an increment.
  - That feature is then implemented and delivered.
- Users or surrogate users can try it out and provide feedback to the development team. You then go on to define and implement the next feature of the system.

Incremental development



Incremental development activities (1 of 2)

Activity	Description
Choose features to be included in an Increment	Using the list of features in the planned product, select those features that can be implemented in the next product increment.
Refine feature descriptions	Add detail to the feature descriptions so that the team members have a common understanding of each feature and there is sufficient detail to begin implementation.
Implement and test	Implement the feature and develop automated tests for that feature that show that its behavior is consistent with its description.

Incremental development activities (2 of 2)

Activity	Description
Integrate feature and test	Integrate the developed feature with the existing system and test it to check that it works in conjunction with other features.
Deliver system increment	Deliver the system increment to the customer or product manager for checking and comments. If enough features have been implemented, release a version of the system for customer use.

Agile software engineering

Why Agile methods in software engineering?

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
TGSJKI

Agile software engineering (1 of 2)

- Software products must be brought to market quickly so rapid software development and delivery is essential.
- Virtually all software products are now developed using an agile approach.
- Agile software engineering focuses on delivering functionality quickly, responding to changing product specifications and minimizing development overheads.

Agile software engineering (2 of 2)

- A large number of ‘agile methods’ have been developed.
  - There is no ‘best’ agile method or technique.
  - It depends on who is using the technique, the development team and the type of product being developed

Scrum

- 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 is an agile method that provides a framework for agile project organization and planning. It does not mandate any specific technical practices.

Scrum

- Scrum is an agile method that focuses on managing iterative development rather than specific agile practices.
- There are three phases in Scrum.
  - The initial phase is an outline planning phase where you establish the general objectives for the project and design the software architecture.
  - This is followed by a series of sprint cycles, where each cycle develops an increment of the system.
  - The project closure phase wraps up the project, completes required documentation such as system help frames and user manuals and assesses the lessons learned from the project.

Scrum terminology (1 of 2)

Scrum term	Explanation
Product	The software product that is being developed by the Scrum team.
Product Owner	A team member who is responsible for identifying product features and attributes. The Product Owner reviews work done and helps to test the product.
Product backlog	A to-do list of items such as bugs, features, and product improvements that the Scrum team has not yet completed.
Development team	A small self-organizing team of five to eight people who are responsible for developing the product.
Sprint	A short period, typically two to four weeks, when a product increment is developed.
Scrum	A daily team meeting where progress is reviewed and work to be done that day is discussed and agreed.

Scrum terminology (2 of 2)

Scrum term	Explanation
ScrumMaster	A team coach who guides the team in the effective use of Scrum.
Potentially shippable product increment	The output of a sprint that is of high enough quality to be deployed for customer use.
Velocity	An estimate of how much work a team can do in a single sprint.

Key roles in Scrum (1 of 2)

- ***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.

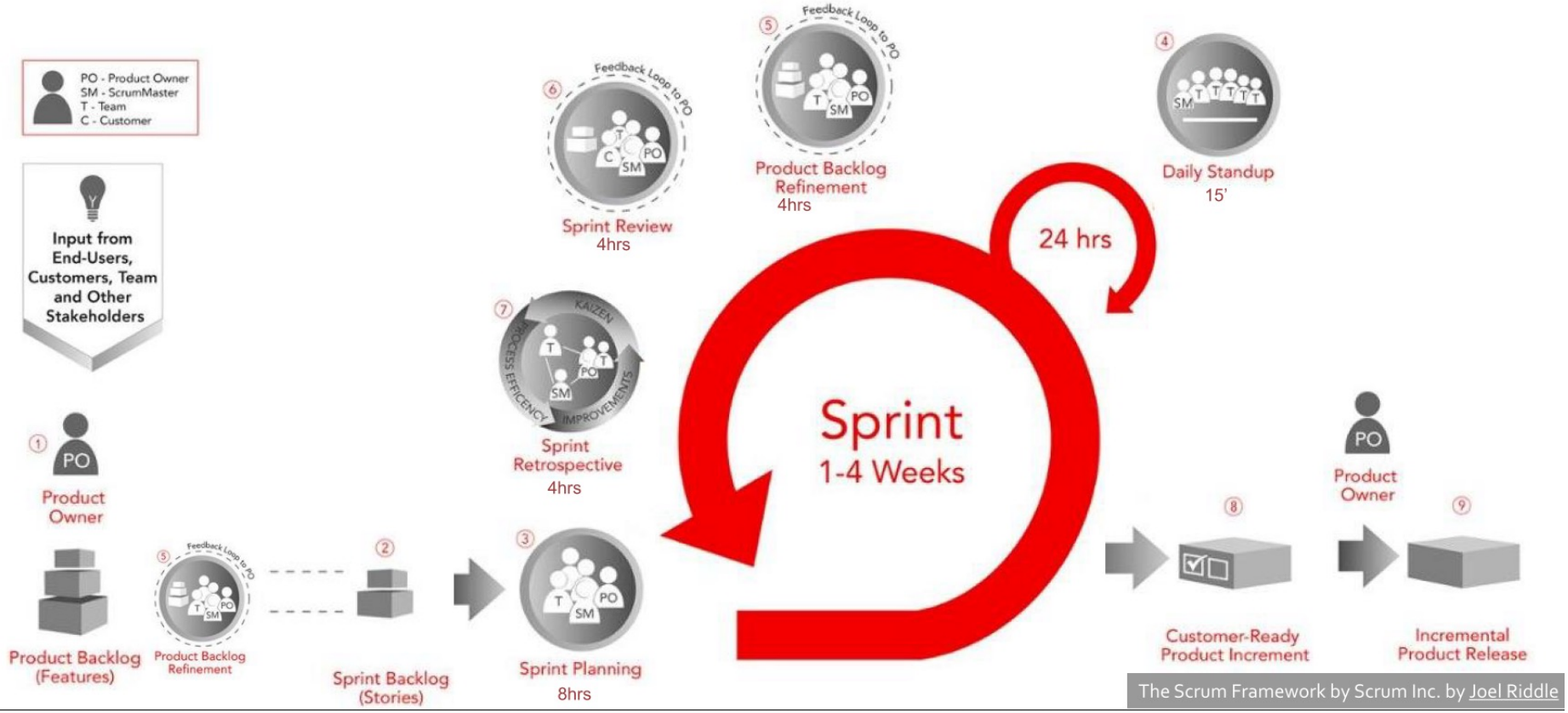
Key roles in Scrum (2 of 2)

- ***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

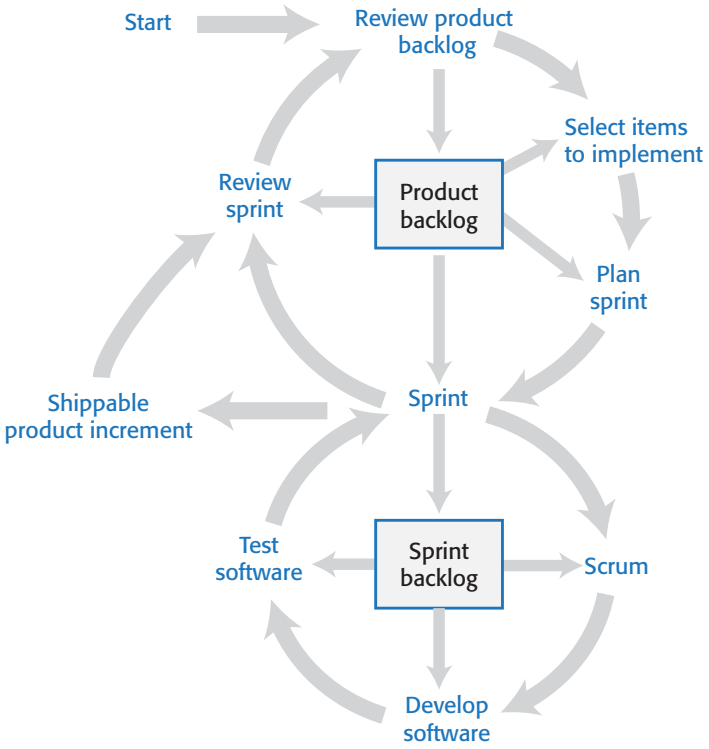
- In Scrum, software is developed in sprints, which are fixed-length periods (2-4 weeks) in which software features are developed and delivered.
- During a sprint, the team has daily meetings (Scrums) to review progress and to update the list of work items that are incomplete.
- Sprints should produce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

Scrum cycle

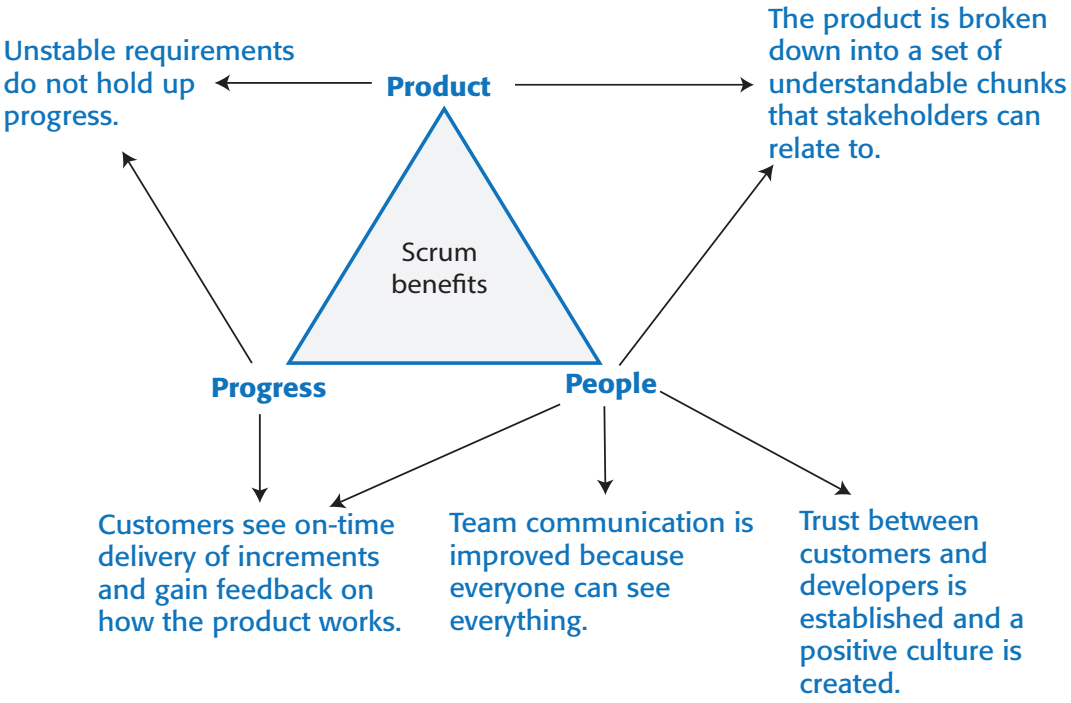


The Scrum Framework by Scrum Inc. by Joel Riddle

Scrum cycle


Start
Review product backlog
Select items to implement
Product backlog
Review sprint
Plan sprint
Shippable product increment
Sprint
Test software
Sprint backlog
Scrum
Develop software

The top five benefits of using Scrum



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.

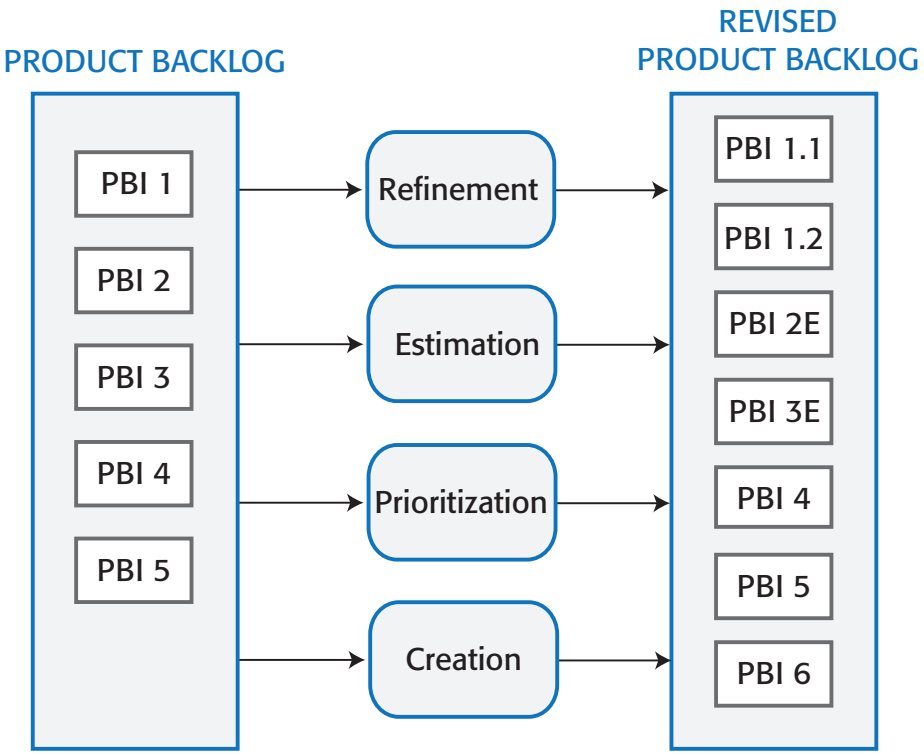
Examples of product backlog items

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 children's 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)

Product backlog item states

Heading	Description
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.

Product backlog activities



Product backlog activities (1 of 2)

- ***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.

Product backlog activities (2 of 2)

- ***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 (1 of 2)

- ***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.

PBI estimation metrics (2 of 2)

- ***Story points***
  - 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.

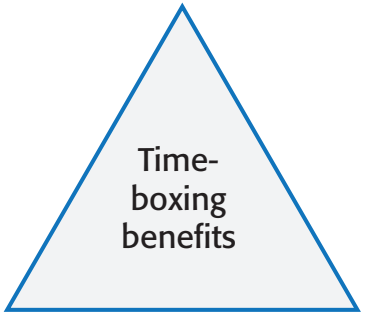
Timeboxed sprints

- 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 (1-4 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.

Benefits of using timeboxed sprints

There is a tangible output (usually a software demonstrator) that can be delivered at the end of every sprint.

Demonstrable progress



Problem discovery

If errors and omissions are discovered the rework required is limited to the duration of a sprint.

Work planning

The team develops an understanding of how much work they can do in a fixed time period.

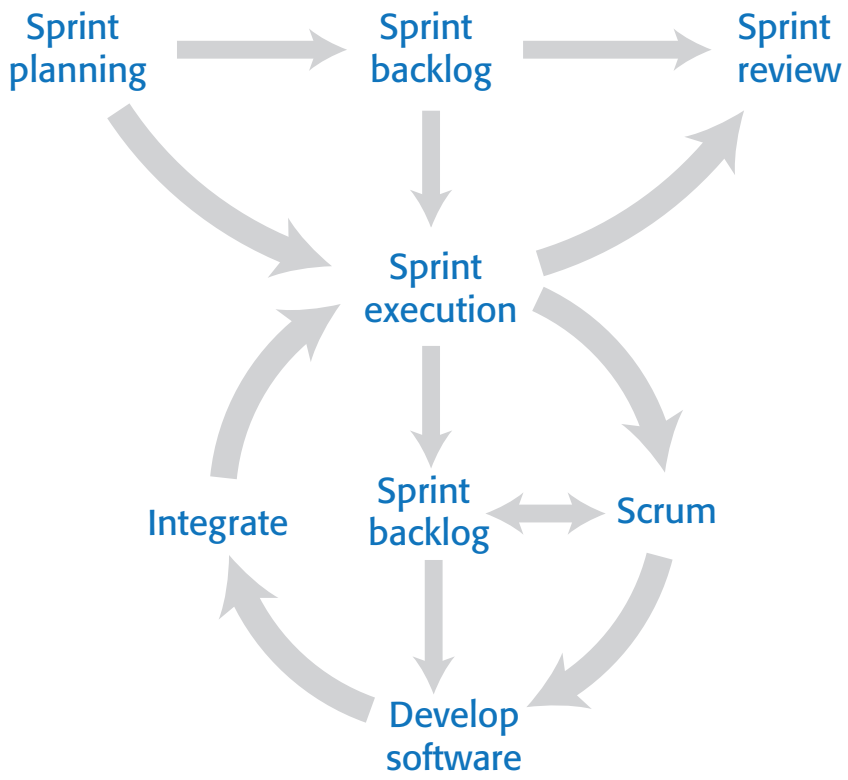
Sprint activities (1 of 2)

- ***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 activities (2 of 2)

- ***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.

Sprint activities



Sprint planning (1 of 2)

- 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.

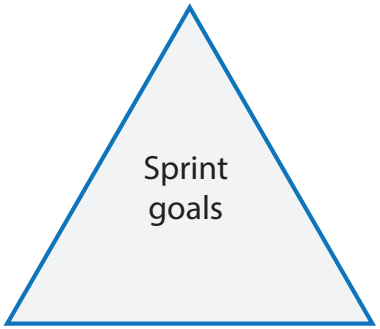
Sprint planning (2 of 2)

- 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.

Sprint goals

Implement user roles so that a user can select their role when they login to the system

Functional



Support

Develop analytics that maintain information about the time users spend using each feature of the system

Performance and reliability

Ensure that the login response time is less than 10 seconds for all users where there are up to 2000 simultaneous login connections.

Questions about SCRUM

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
GCERRX

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, and present 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, scrums 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 decides who should work on sprint backlog items that day.

Agile activities

- Scrum does not suggest the technical agile activities that should be used. However, I think there are two practices that should always 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.

Code completeness checklist

State	Description
Reviewed	The code has been reviewed by another team member who has checked that it meets agreed coding standards, is understandable, includes appropriate comments, and has been refactored if necessary.
Unit tested	All unit tests have been run automatically and all tests have executed successfully.
Integrated	The code has been integrated with the project codebase and no integration errors have been reported.
Integration tested	All integration tests have been run automatically and all tests have been executed successfully.
Accepted	Acceptance tests have been run if appropriate and the Product Owner or the development team has confirmed that the product backlog item has been completed.

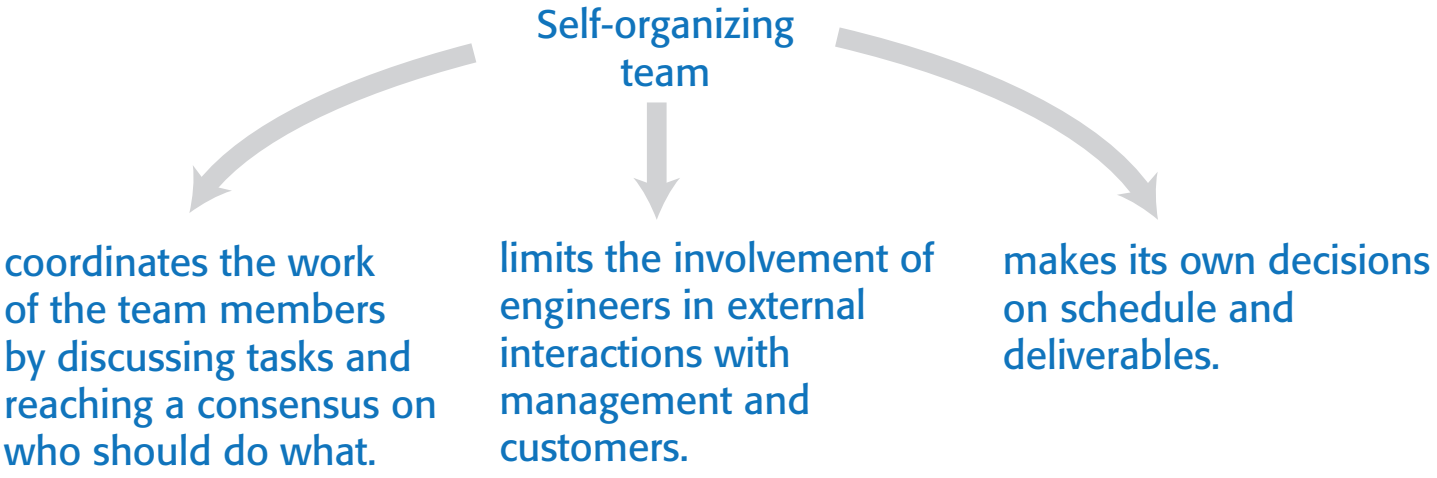
Sprint reviews (1 of 2)

- 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.

Sprint reviews (2 of 2)

- 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.

Self-organizing teams



Team size and composition (1 of 2)

✧ 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.

Team size and composition (2 of 2)

- 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 (1 of 2)

- 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.

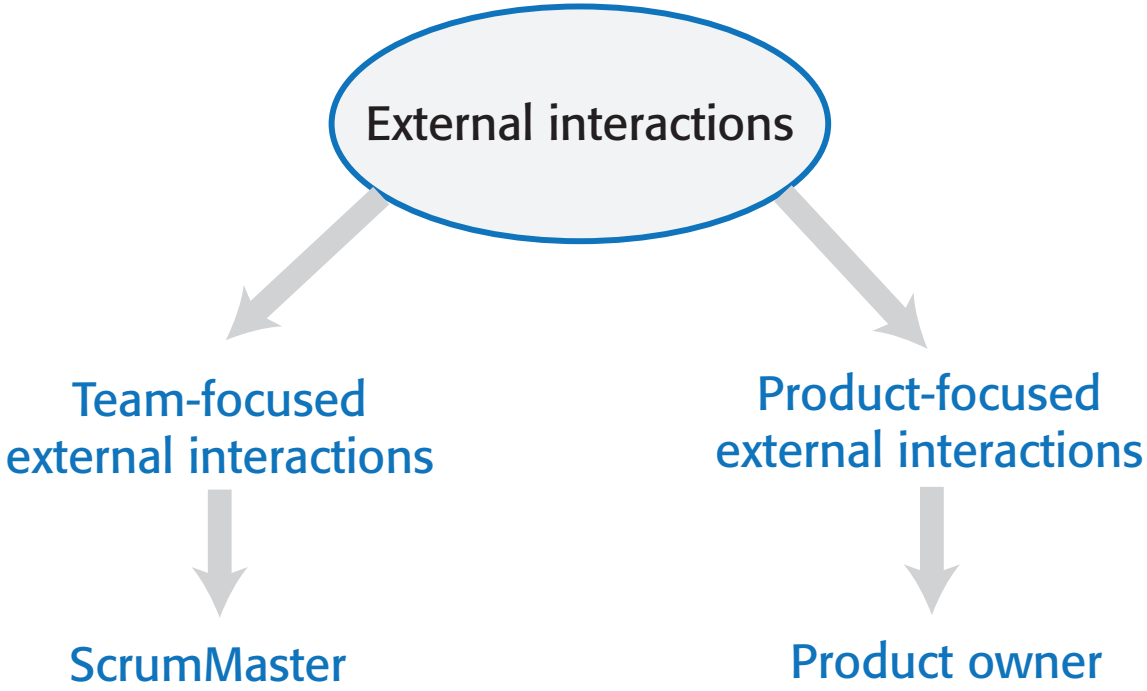
Team coordination (2 of 2)

- 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.

Managing external interactions



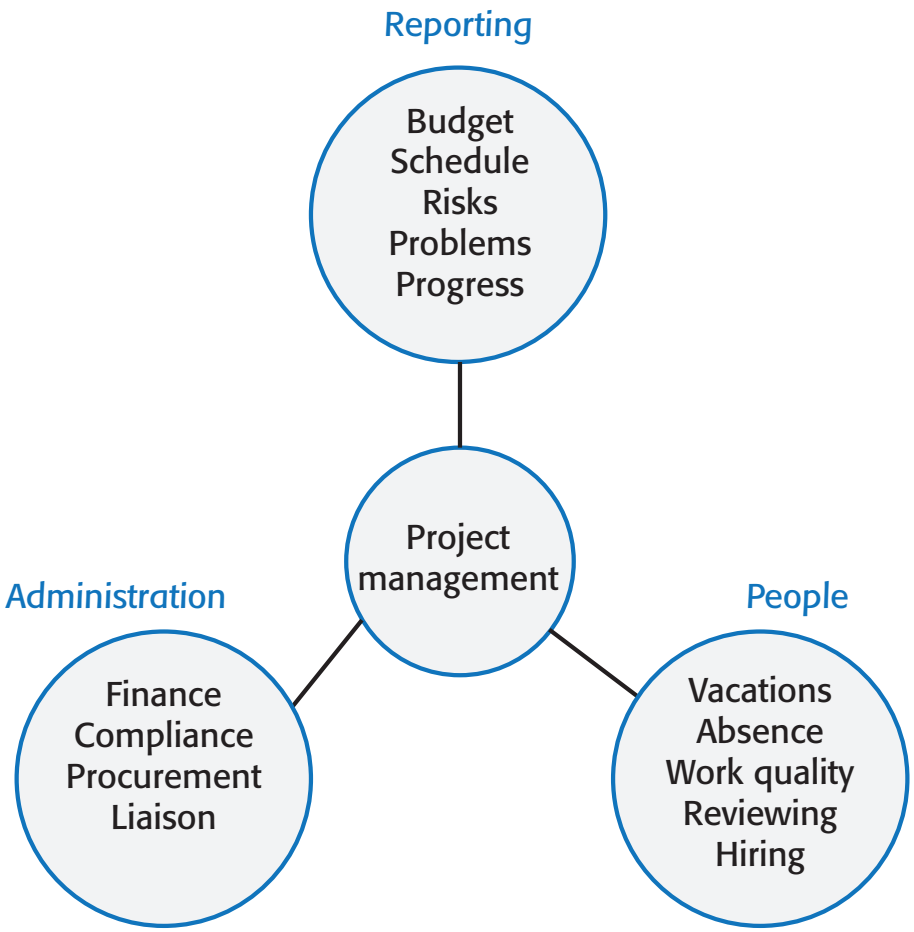
Project management (1 of 2)

- 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.

Project management (2 of 2)

- 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.

Project management responsibilities



Questions about SCRUM

1 Go to wooclap.com

2 Enter the event code in the top banner

Event code
OICBPF

Key points (1 of 2)

- Agile methods are incremental development methods that focus on rapid software development, frequent releases of the software, reducing process overheads by minimizing documentation and producing high-quality code.
- Agile development practices include
  - User stories for system specification
  - Frequent releases of the software,
  - Continuous software improvement
  - Test-first development
  - Customer participation in the development team.

Key points (2 of 2)

- Scrum is an agile method that provides a project management framework.
  - It is centred round a set of sprints, which are fixed time periods when a system increment is developed.
- Many practical development methods are a mixture of plan-based and agile development.
- 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.


HOW
WHY
WHERE
WHEN
WHERE
WHY
WHERE
WHO
WHEN
WHAT
WHAT
WHY
WHAT
WHEN
HOW
WHEN
WHO
WHERE
HOW
WHY