CS30500:
Introduction to Software Engineering

Chapter 3. Agility and Process 1/2

Prof. In-Young Ko

School of Computing

Today's Plan

- Agility

- Popular agile methods
  - Scrum

  - Extreme Programming (XP)

  - Kanban

As a Software Engineer…

- **Your task** is to design and develop a smart home system where a butler robot understands the user's needs and seamlessly controls and coordinates home appliances.

- What <u>development activities</u> do you plan to carry out?

[Generated by ChatGPT 4o]

What would be the most effective development process for this project?

KAIST School of Computing

What would be the most effective development process?

ChatGPT's Answer:

For a **normal development situation** (not a fast-track approach), an **Agile methodology**—specifically **Scrum or Kanban**—would be the best software process. This ensures **continuous iteration, flexibility, and adaptability** while developing a complex smart home system like the LG butler robot.

AGILITY

Manifesto for Agile Software Development

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.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

"In February 2001, seventeen software developers met at the Snowbird resort in Utah to discuss lightweight development methods."

http://agilemanifesto.org/

What is Agility?

- **Effective** (rapid and adaptive) response to change

- **Effective communication** among all stakeholders

- Drawing the customer onto the team

- Organizing a team so that it is in control of the work performed

- Rapid, incremental delivery of software

Development cost

Cost of change using conventional software process

Cost of change using agile process

Idealized cost of change using agile process

Development schedule progress
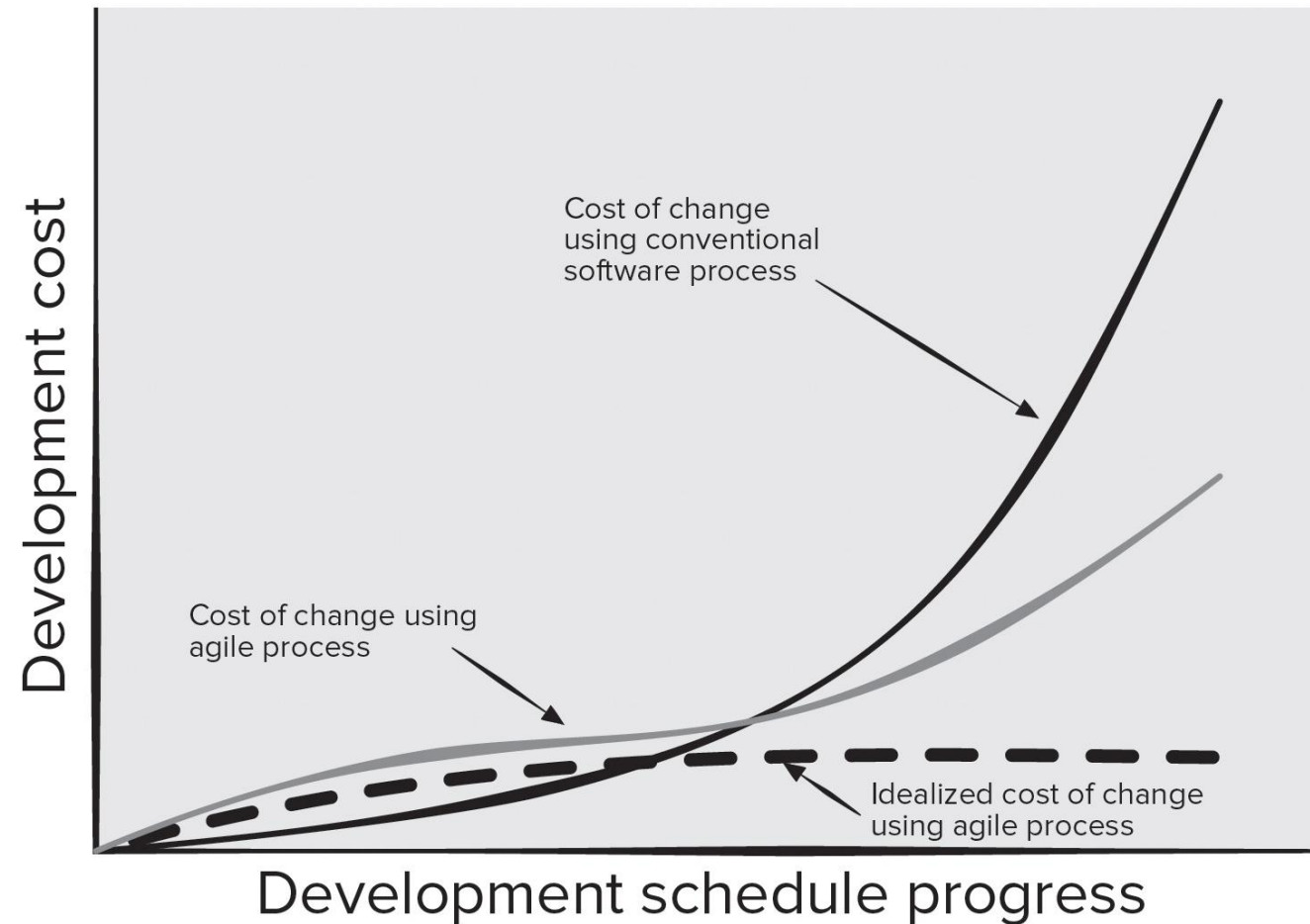
Agility and Cost of Change

[PrMa20]

Principles Behind the Agile Manifesto

1. Early and continuous delivery of valuable software
2. Welcome changing requirements (even late in development)
3. Deliver working software frequently (from two weeks to two months)
4. Business people and developers must work together daily
5. Build projects around motivated individuals
6. Face-to-face conversation (within a team)
7. Working software is the primary measure of progress
8. Sustainable development (maintain a <u>constant pace</u> indefinitely)
9. Continuous attention to technical excellence and good design
10. Simplicity
11. Self-organizing teams
12. The team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

Agile Practices

- More than 70% of organizations employ agile approaches

- Core concept:
  - Iterative, intensive communication, early customer feedback

- Several weeks per iteration

- Each iteration delivers a complete <u>working software</u>

- Pay less attention to detailed documentations

- <u>Close collaboration</u> across <u>the team</u> and with <u>customers</u>

- Frequent releases allow <u>continuous improvement</u>

- Cons of Agile Methods
  - Make the maintenance more complicated
  - Difficult to accurately estimate budget, time and people required for the project

Self-organizing Team

- Does not depend on or wait for a manager to assign work
  - <u>Finds their own work and manage</u> the associated responsibilities and timelines

- Takes on the responsibility of <u>choosing the most effective and efficient way</u> to complete their work
  - Regularly looks for ways to <u>improve</u> through experimentation

- Must have a high sense of ownership and responsibility

- Needs to communicate often and trust in the capabilities of everyone on the team

When is Agile Recommended?
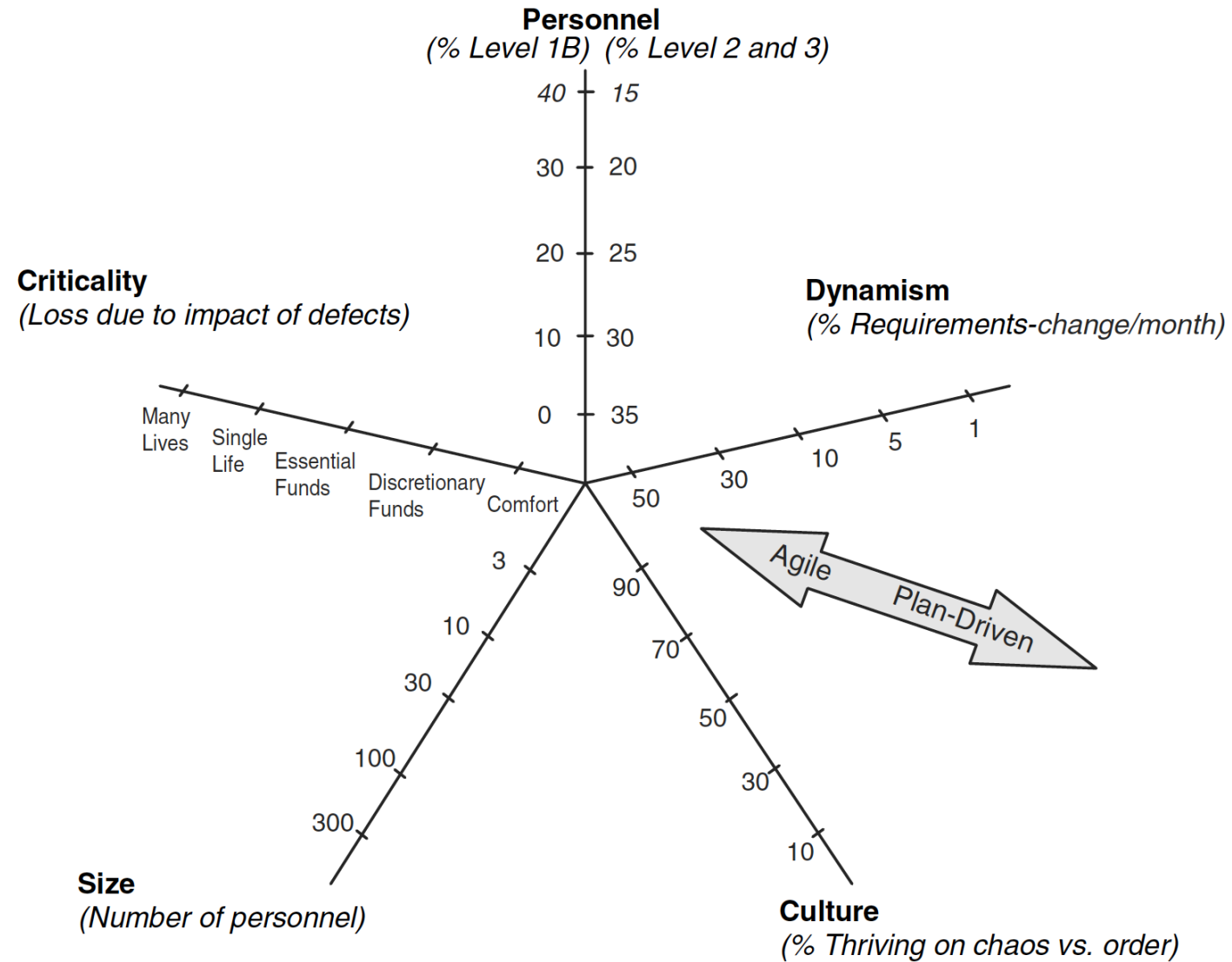
Boehm and Richard Turner. 2003. Balancing Agility and Discipline: A Guide for the
Perplexed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Use Cases of Agile Processes

- <u>Startup initiatives</u>, when end users' early feedback is required

- Most of mid-sized projects where business requirements cannot be confidently translated to detailed software requirements

- Large projects that are easy to <u>divide into small functional parts</u> and can be developed incrementally over each iteration
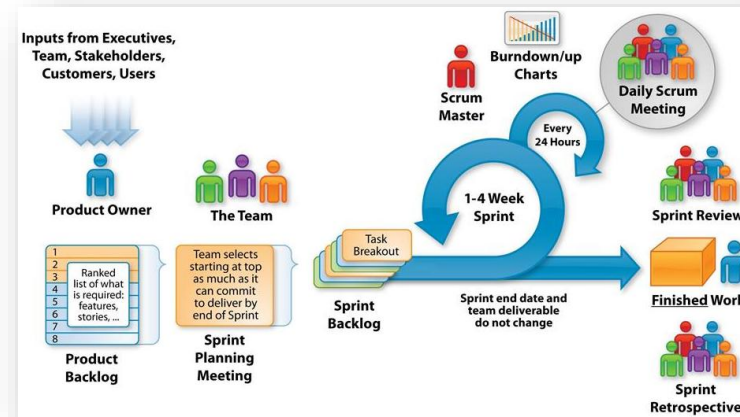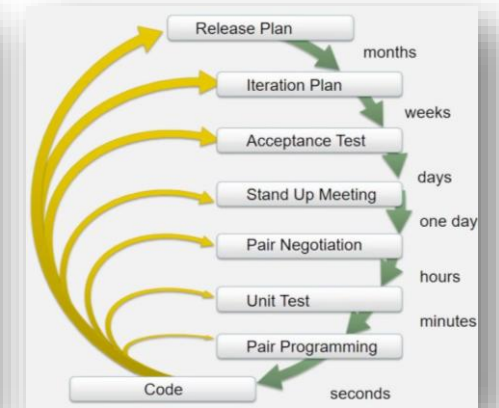
[Shi19]

POPULAR AGILE METHODS

Popular Agile Methods

- Scrum

- Extreme Programming (XP)

- Kanban

- DevOps

- ….

https://www.c-sharpcorner.com/UploadFile/d9c992/the-agile-scrum-framework/

https://www.youtube.com/watch?v=FufRFdFFu4k

https://www.bocasay.com/kanban-method-it-development-projects/

https://ko.m.wikipedia.org/wiki/%ED%8C%8C%EC%9D%BC:Devops-toolchain.svg
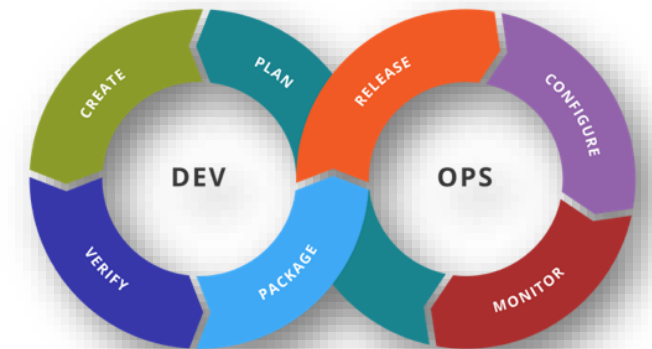
What is Scrum?

- The name is derived from a rugby activity (<u>working together</u> with the teammates)

- Developed by Jeff Sutherland and his team in the early 1990s

- **Sprint**

  [Shi19]

  – <u>An iteration</u> of a framework activity (requirements, analysis, design, …)

  – <u>2-4 weeks</u> long

  – Preceded with thorough <u>planning</u> and <u>previous sprint assessment</u>

  – No changes are allowed after the sprint activities have been defined

- The work tasks to be done within a sprint are defined by the Scrum team

  – Adapted to the <u>problem</u> to solve

  – Defined and modified in <u>real time</u>

- <u>Number of sprints</u> needs to be decided based on product complexity and size
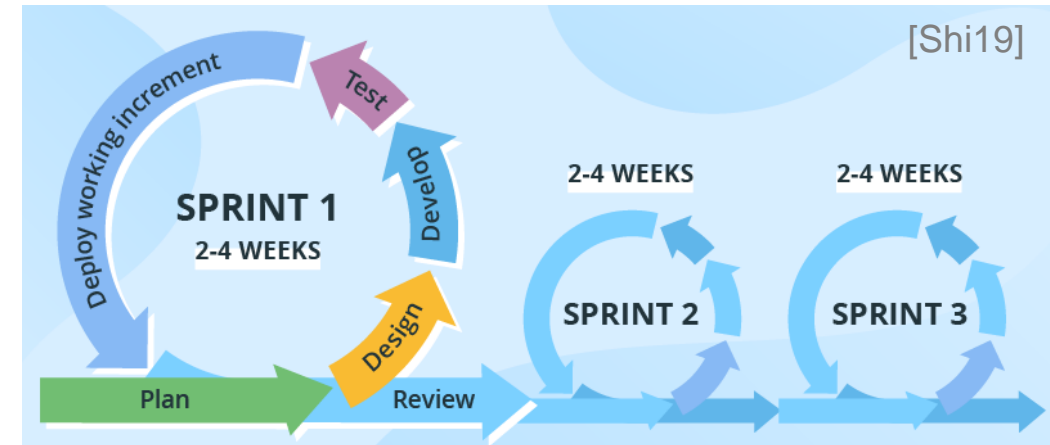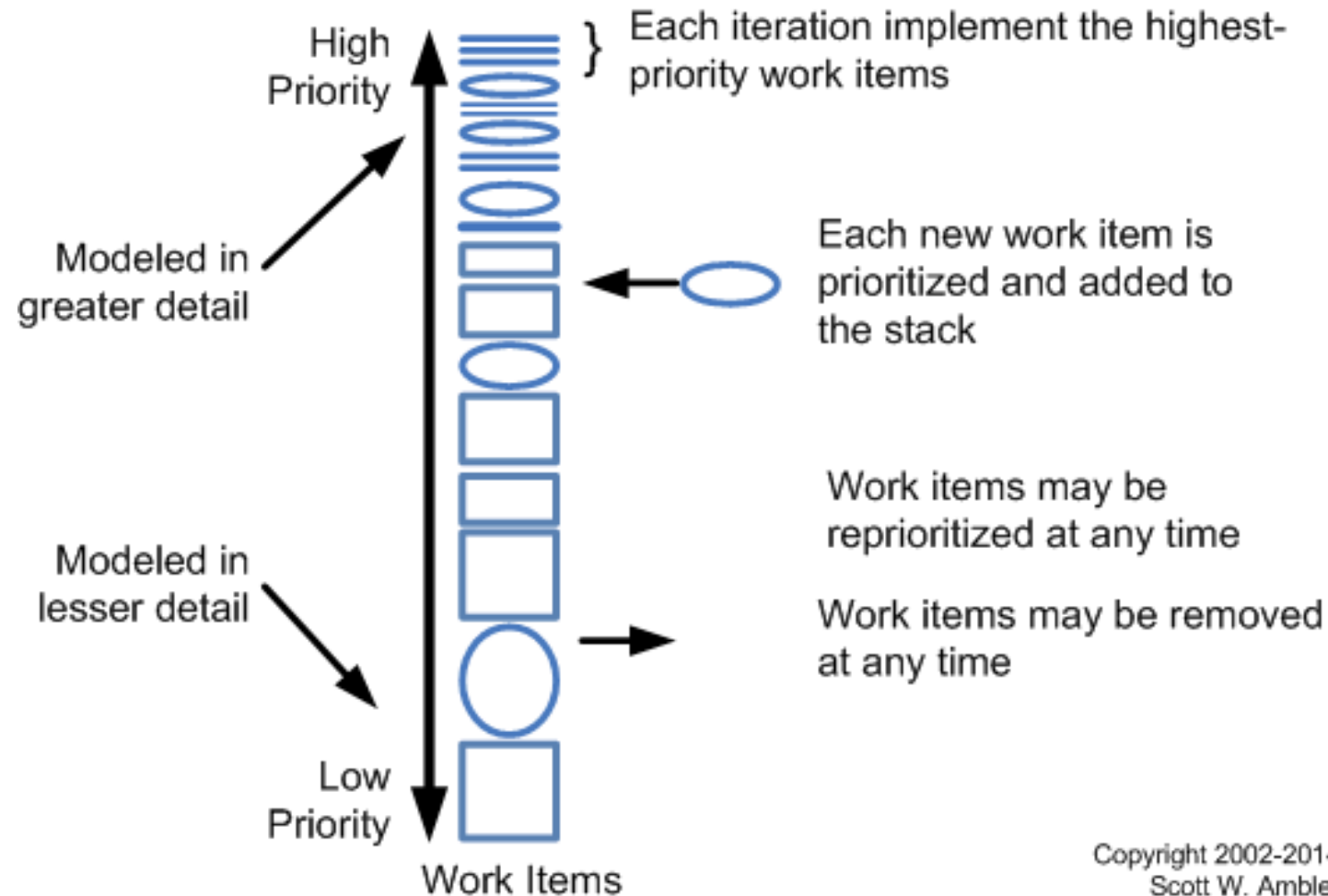
[PrMa20]

Scrum Teams and Artifacts

- Scrum team: A <u>self-organizing</u> <u>interdisciplinary</u> team consisting of a
  - *Product owner*: the person who orders the product features and decides whether to end a sprint
  - *Scrum master*: a <u>facilitator</u> who runs the daily Scrum meeting, <u>coaches</u> the team members, and helps the product owner to <u>manage</u> the product backlog
  - *Development team*: a <u>small (3-6 people)</u>

- Scrum artifacts:
  - *Product backlog*: a prioritized list of <u>product requirements</u> (a requirement can be added at any time)
  - *Sprint backlog*: the subset of product backlog items selected by the team <u>to be completed during the current active sprint</u> (cannot be changed during a sprint)
  - *Code increment*: all backlog items completed in previous sprints + sprint backlog items

- Development proceeds by breaking the project into a series of <u>incremental prototype development</u> (*sprints*, 2-4 weeks each)

Agile Requirements Change Management

Scrum Meetings

- **Backlog Refinement Meeting**: developers work with stakeholders to <u>create the product backlog</u>

- **Sprint Planning Meeting**: backlog partitioned into "sprints" derived from the backlog and <u>next sprint defined</u> (sprint backlog items are prioritized)

- **Daily Scrum Meeting**: team members <u>synchronize their activities</u> and <u>plan a work day</u> (15 minutes max.)

- **Sprint Review**: <u>prototype "demos"</u> are delivered to the stakeholders at the end of a sprint

- **Sprint Retrospective**: after sprint is complete, team considers <u>what went well</u> and <u>what needs improvement</u> (3 hours)
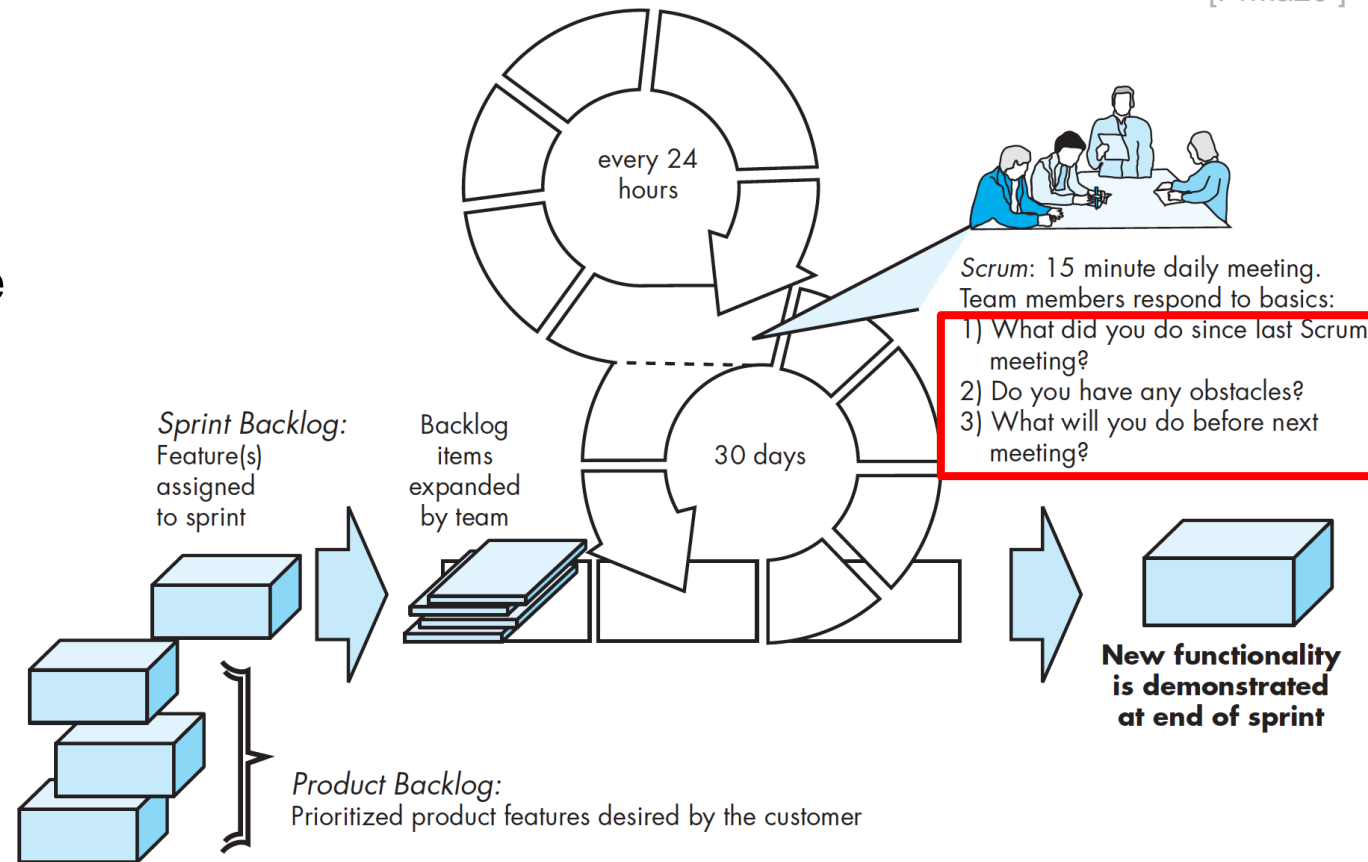
- Example of a daily scrum meeting:
  - https://www.youtube.com/watch?v=q_R9wQY4G5I (3:40-5:40-8:30)

Pros and Cons of Scrum

- Pros
  - Product owner sets priorities
  - Team owns decision making
  - Documentation is lightweight
  - Supports <u>frequent updating</u>

- Cons
  - Difficult to control the <u>cost of changes</u>
  - May not be suitable for large teams
  - Requires <u>expert team members</u>