Projecte d'Enginyeria del Software: Metodologies àgils

Xavier Franch

En col.laboració amb M.J. Casany, S. Martínez i J. Piguillem

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Agile Manifesto

Individuals and interactions

over processes and tools

Working software

over comprehensive documentation

Customer collaboration

over contract negotiation

Responding to change

over following a plan

http://www.agilemanifesto.org/

Agile Principles (summary)

- Early and continuous delivery of valuable software
- Changing requirements
- Business people and developers work together
- Motivated individuals
- Face-to-face conversation
- Sustainable development
- Technical excellence
- Simplicity (maximizing the amount of work not done)
- Self-organizing teams
- Regular reflection and adjustment

Agile vs. plan-driven development

- Plan-driven development processes typically have many rules, practices, and documents and need high disciplined process performers
- Agile approaches have less rules and practices and are easier to follow

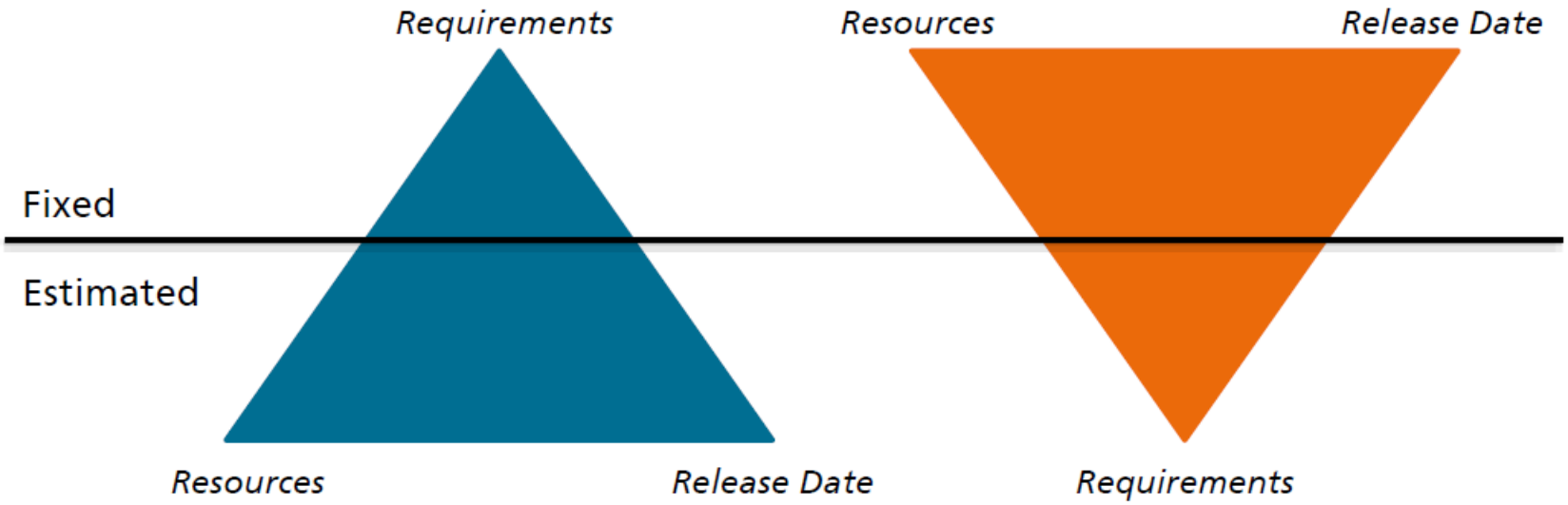
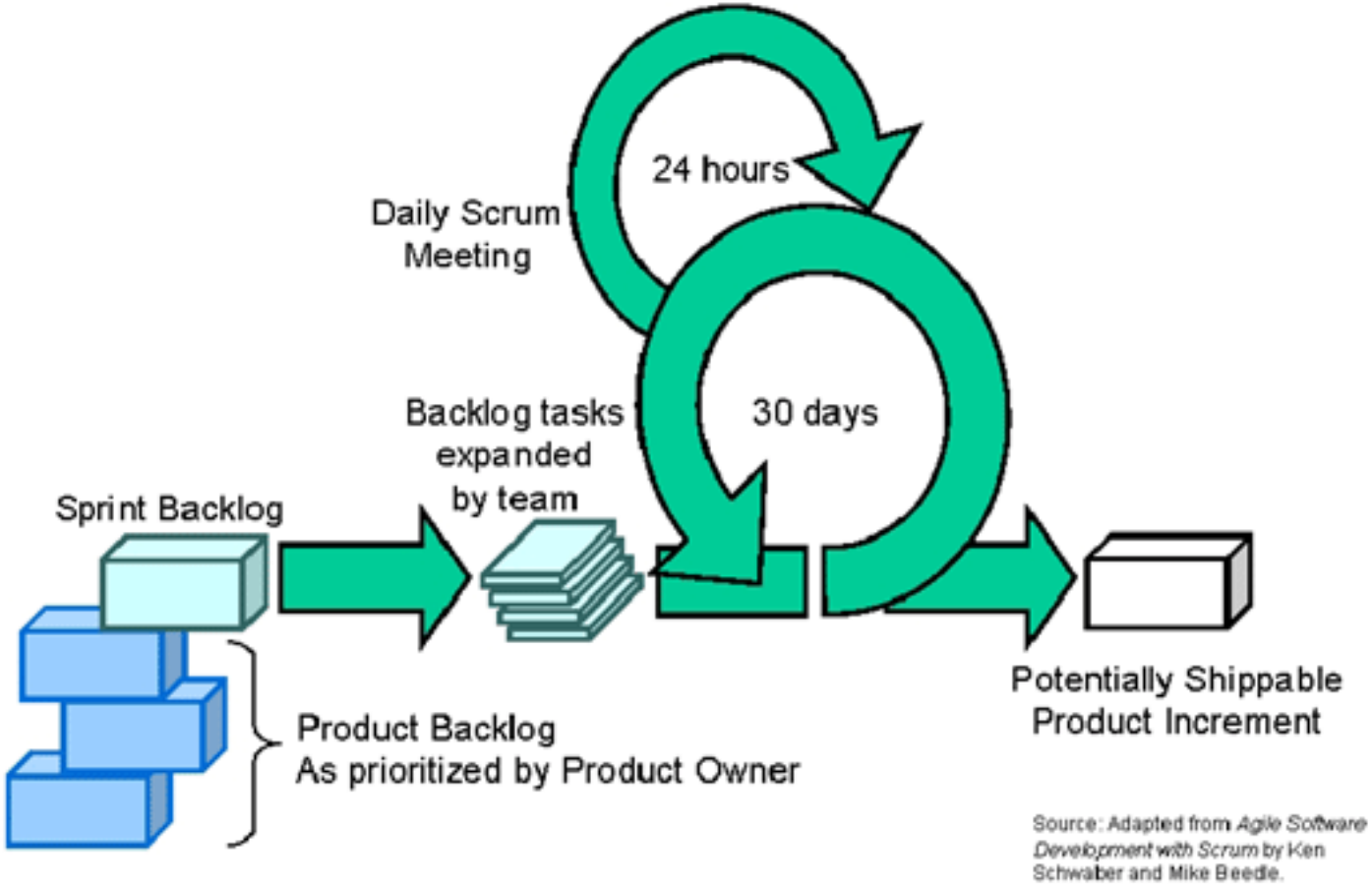


Image source: FhG IESE, Process division, based on SAP, BITKOM AK QM Meeting, 2014/03/27

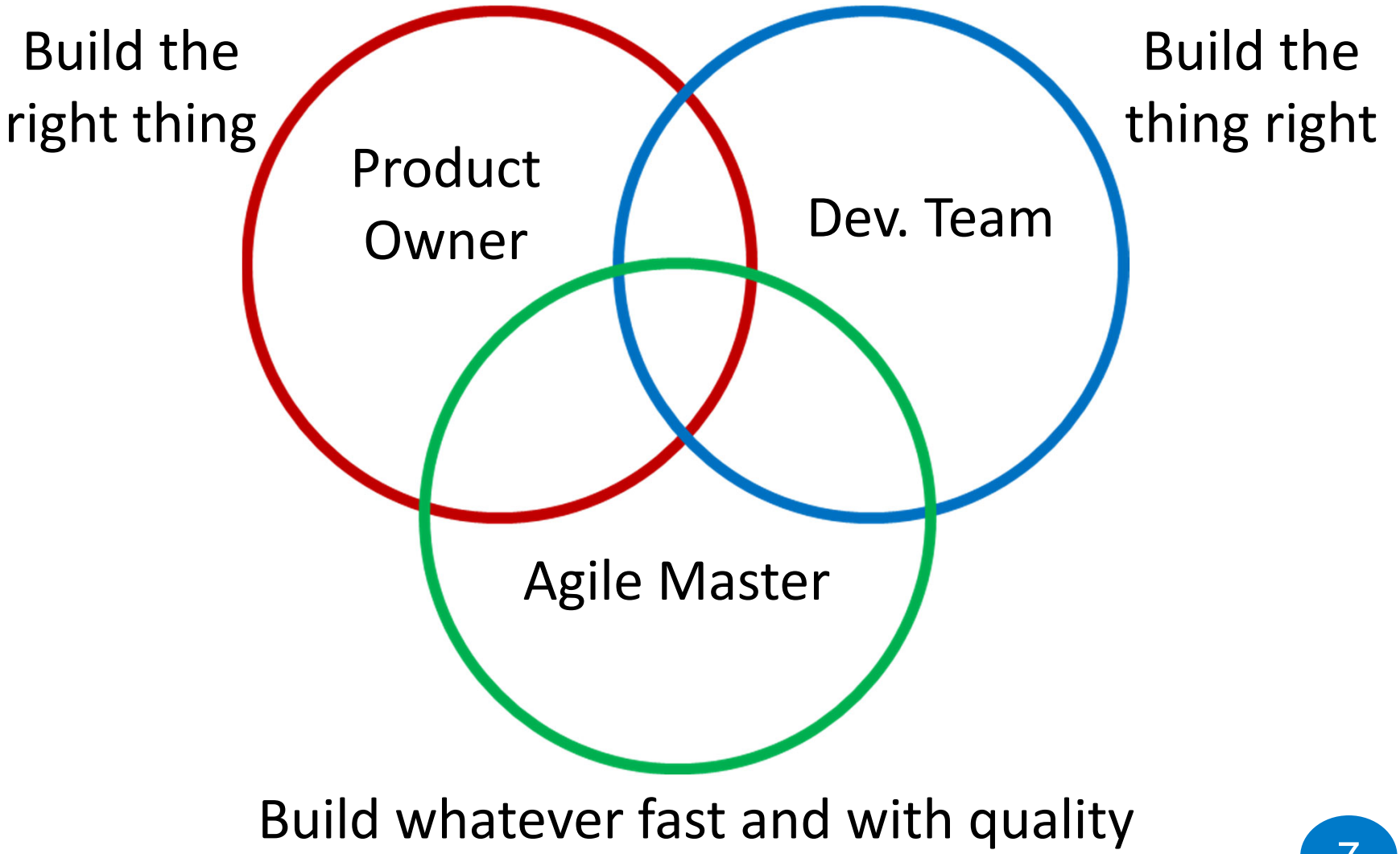
The agile process: e.g., SCRUM



Roles

- Product Owner
  - brings his/her vision to the team
  - outlines the work in the product backlog
  - prioritizes user stories based in business value
  - usually a client representative, who reconciles interests and expectations of all stakeholders
- Agile Master
  - facilitates the agile process
- Development team
  - cross-functional, self-organizing, self-managing

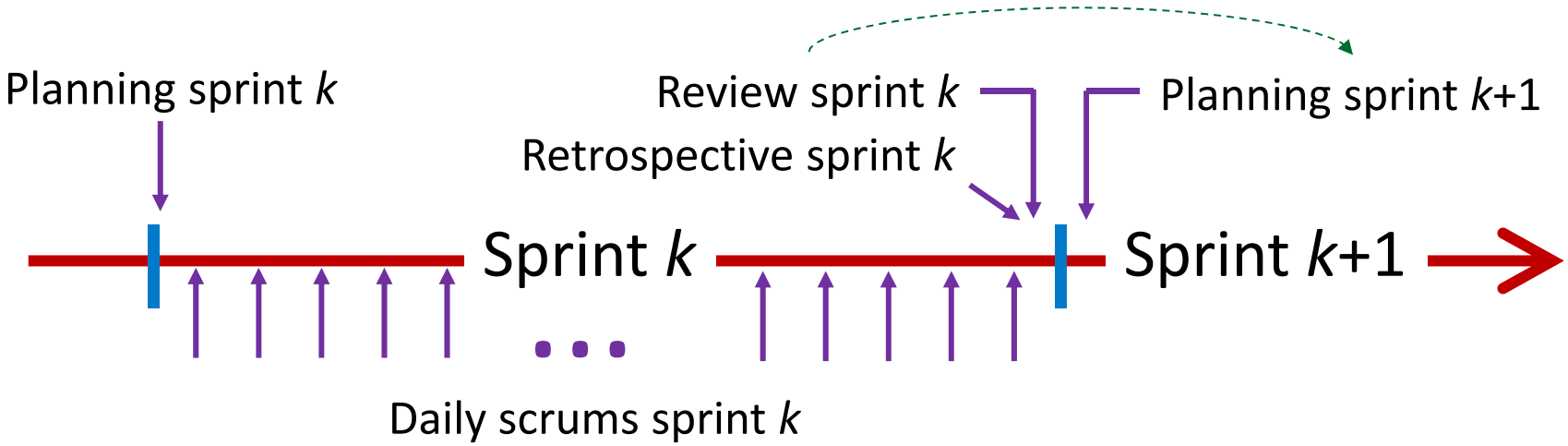
Roles


Build the right thing
Product Owner
Dev. Team
Build the thing right
Agile Master
Build whatever fast and with quality

Phases

- Inception: create a product plan
  - establish the project scope
  - determine the initial product backlog
  - identify possible risks
  - formulate a candidate architecture
  - prepare the working environment
- Development: through different sprints
  - organize the work in every sprint
  - develop and test the software
  - finalize every sprint with a potentially shippable product

Agile ceremony



- Sprint planning
  - Defining a sprint goal and creating the sprint backlog
- Daily scrum (a.k.a. stand-up)
  - The team synchronizes activities and create a plan for the next 24 hours

Agile ceremony: Sprint Review

The development team:

- discusses what went well during the sprint, what problems it ran into, and how those problems were solved
- demonstrates the work that it has "Done" and answers questions about the Increment
- the entire group collaborates on what to do next, so that the Sprint Review provides valuable input to the next Sprint Planning

The result of the Sprint Review is a revised Product Backlog that defines the probable Product Backlog items for the next Sprint

From: https://www.scrum.org/resources/what-is-a-sprint-review

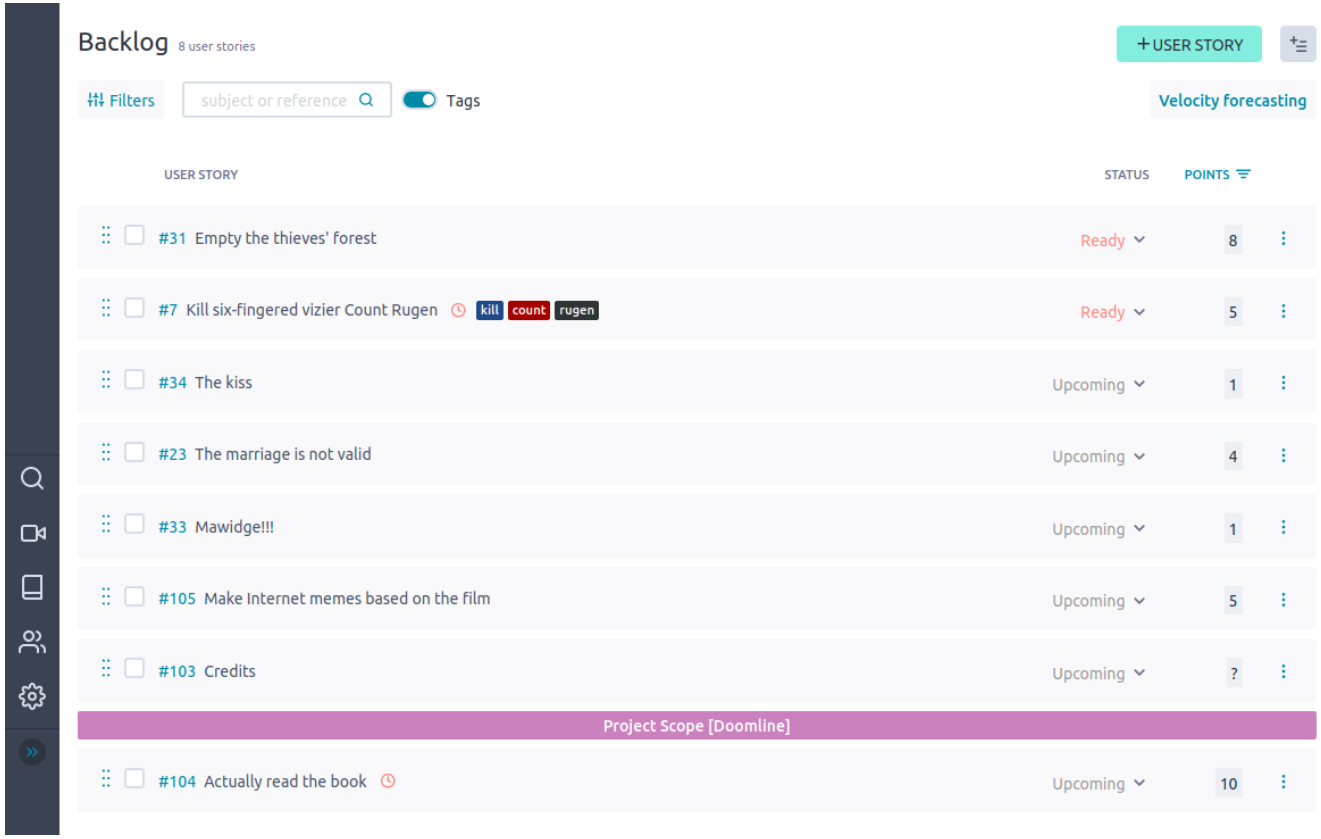
Agile ceremony: Sprint Retrospective

- The Sprint Retrospective occurs after the Sprint Review
- During the Sprint Retrospective, the team discusses:
  - What went well in the sprint
  - What could be improved
  - What will they commit to improve in the next sprint

From: https://www.scrum.org/resources/what-is-a-sprint-retrospective

Backlog

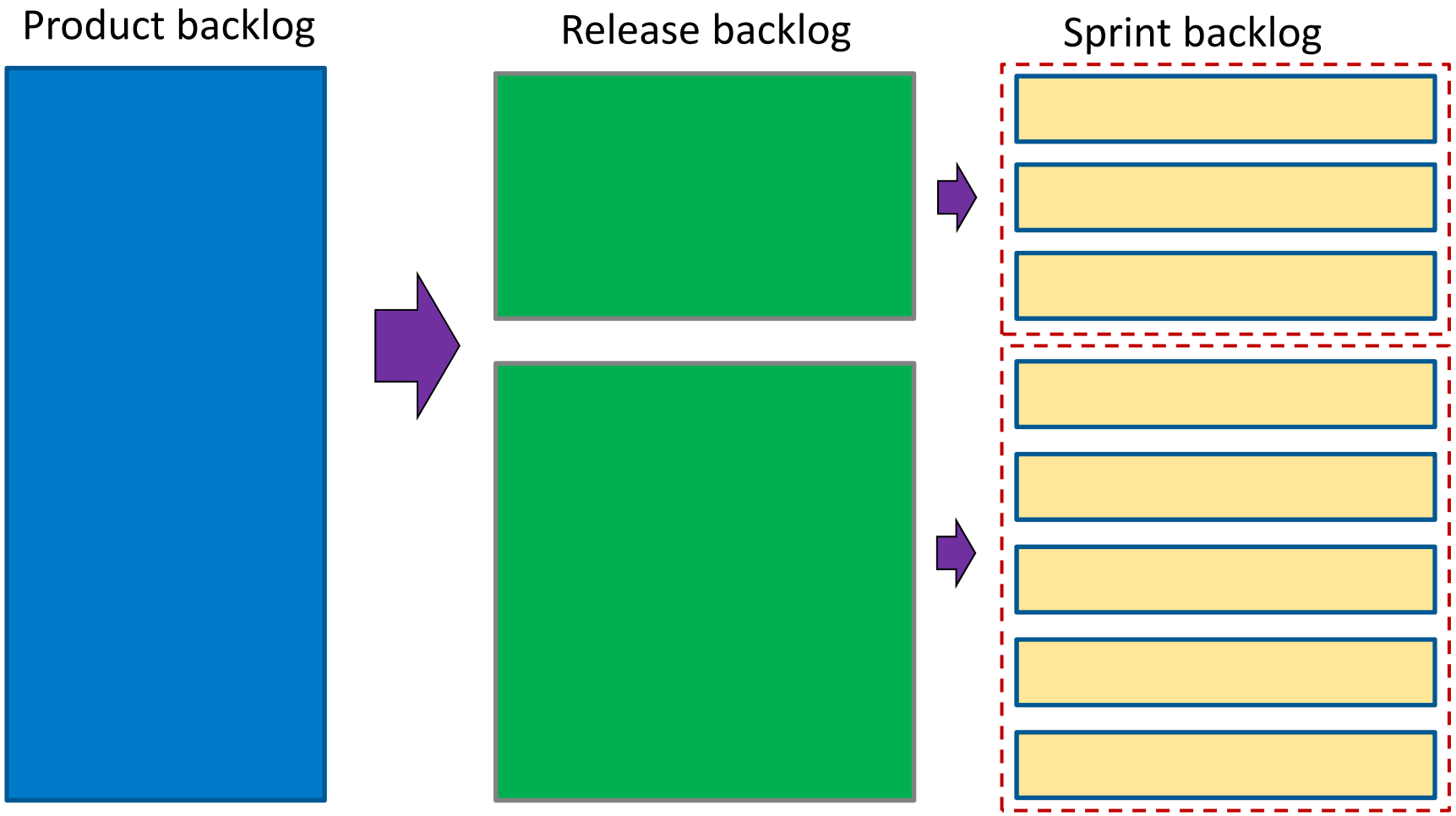
Maintains a list of requirements with associated attributes



Types of backlog

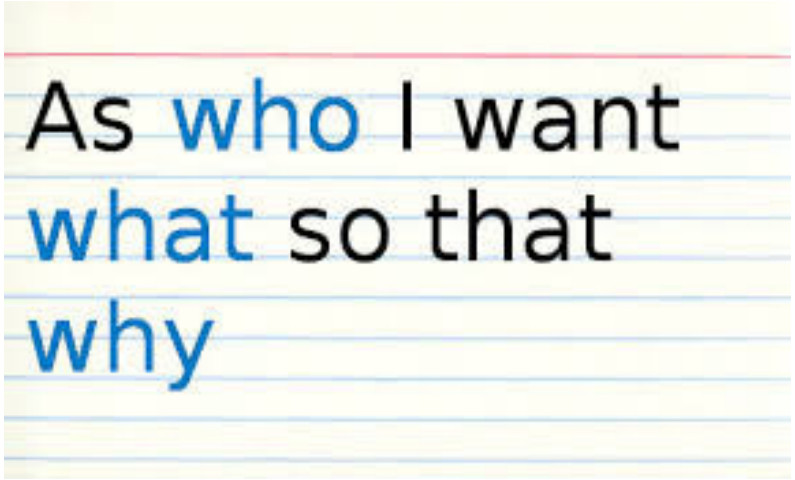
- Product backlog
  - Contains an ordered list of product requirements that an agile team maintains for a product
- Release backlog
  - Requirements to be implemented in the next release
- Sprint backlog
  - The list of work the development team must address during the next sprint
  - The (potentially releasable) output of the sprint that meets the sprint goal is called *increment*

Types of backlog



User stories

Requirements are usually expressed as user stories



Acceptance criteria

Useful for:

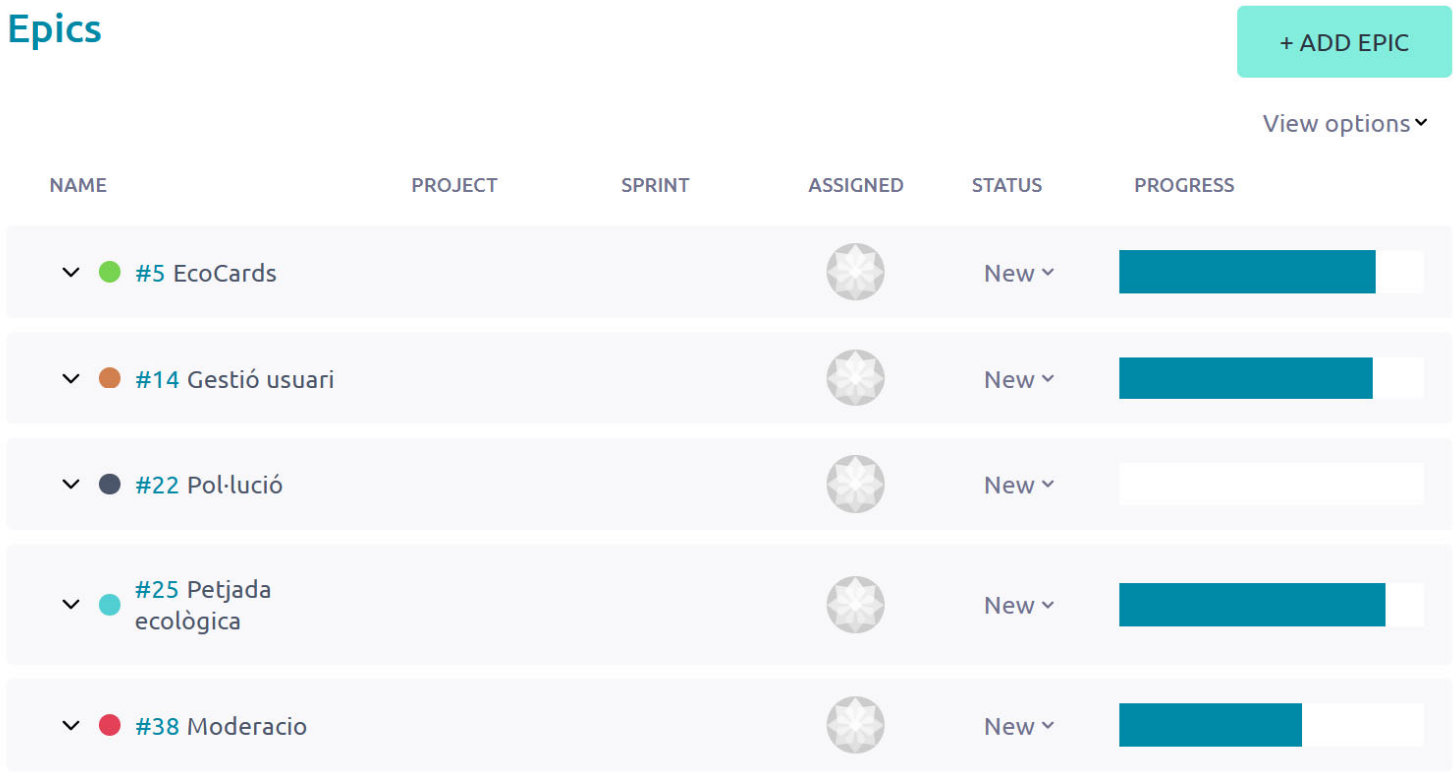
- adding details to user stories
- to know exactly when a user story is fulfilled

Validated through acceptance tests:

- ideally implemented through automated tests

Epics

Group of interrelated user stories



Other mechanisms exist (themes, features, …), with sometimes confusing meaning

Non-Functional Requirements

- System-wide
  - Create user stories (and eventually, themes)

    E.g. As final user, I want the system to respond fast so that…
  - Identify as much acceptance criteria as possible

    E.g. Test that the system responds to all non search requests within 1 second of receiving the request
    - These are checked in those user stories where they apply
  - Others to be added directly as acceptance criteria in the user stories that apply
- Local
  - Example: privacy requirement in a login
  - Include directly as acceptance criteria

In any case, be sure that they can be measured and are realistic!

http://www.scrumcrazy.com/Handling+Non+Functional+Requirements+in+User+Stories

Story Points

Abstract measure of complexity

- numbers (e.g., 1-10; 1-2-3-5-8-13)
- labels (e.g., X-Small, Small, Medium, Large, X-Large)

Every user story in the backlog is estimated

- analogy
- expert assessment
- disaggregation

Planning poker combines all of them

Splitting user stories

Split a user story when

- it is still too large (i.e., too many story points) to fit in an iteration
- even if it would fit, there is not enough room

Some criteria to split

- directed by data or by functionality
- remove cross-cutting concerns
- consider chunks of different priorities

Iteration tasks

Make user stories concrete in an iteration:

- mix of detail, engineering nature and management
  - knowing more about acceptance rules
  - design UI, code, specify acceptance tests, …
  - project meetings (reviews, planning, …)
- they need to be quantified in terms of effort hours
  - roughly matching the story points estimation

Tasks are assigned to team members dynamically:

- considering both skills and availability

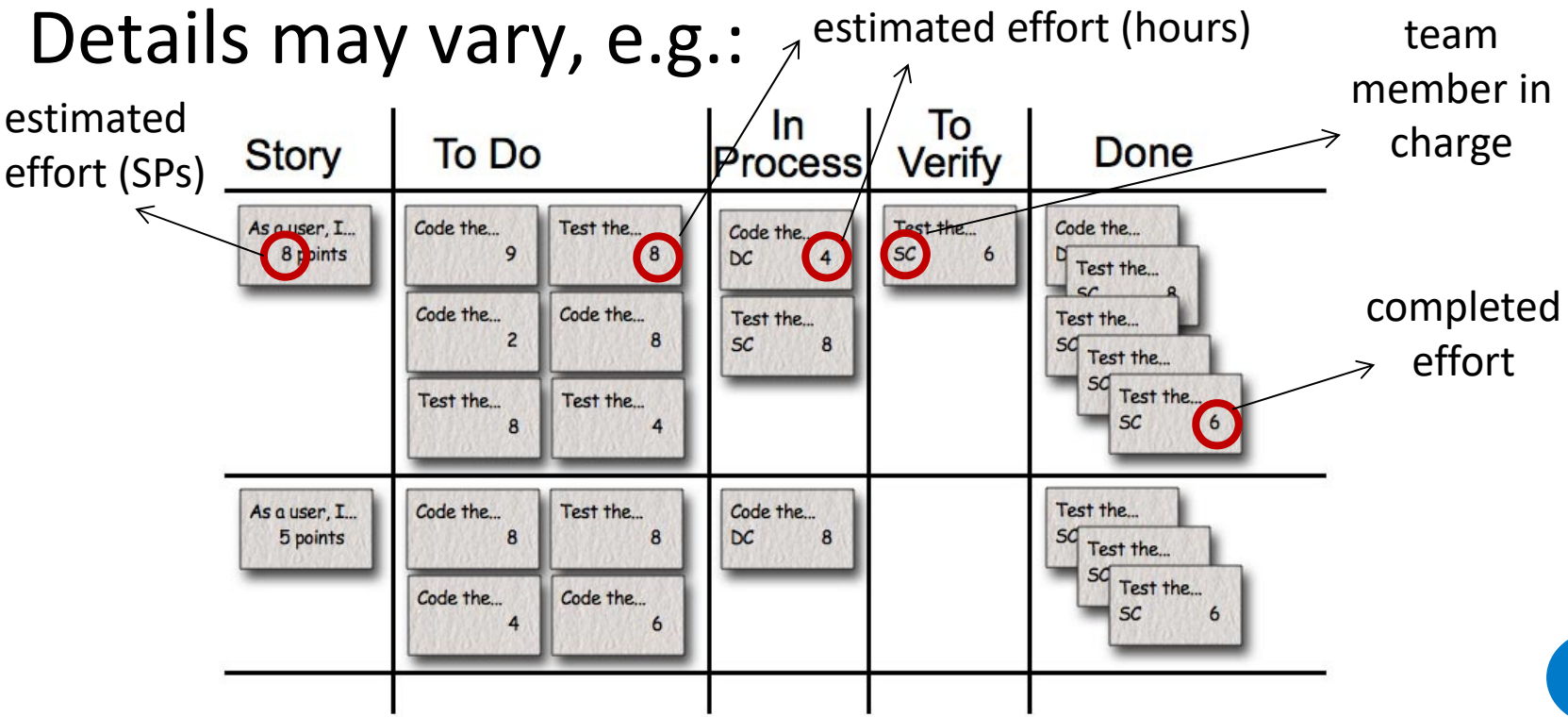
Tasks need to add value to the project

The task board

Two main purposes

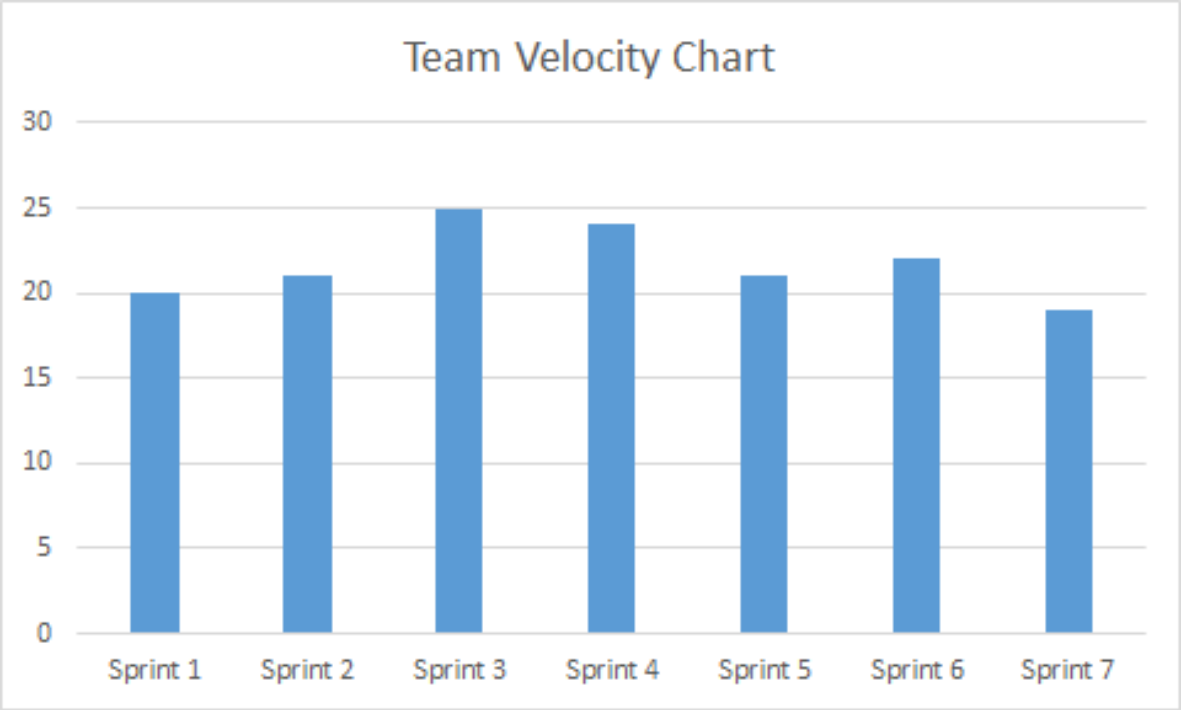
- self-organization
- monitoring (what is left, what is currently being done)

Details may vary, e.g.:



Velocity

Amount of work completed in each iteration



It is the main instrument in planning dates and scope

Prioritization

Factors to be considered:

- financial value
- development costs
- new knowledge
  - about the product
  - about the project
- amount of risk removed

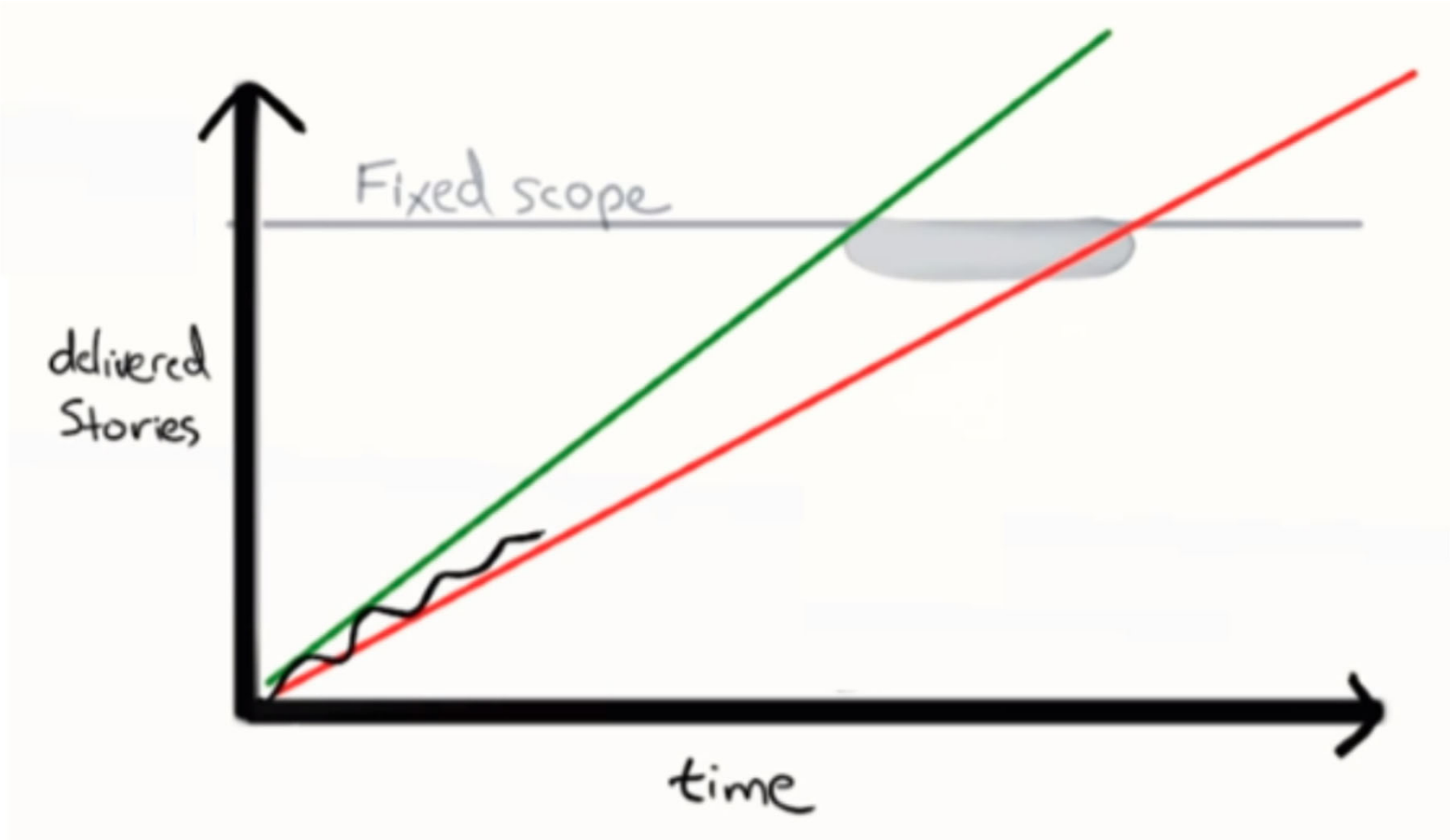
Dealing with bugs

- Bugs may appear at any moment
- In user stories still open, do not have important effects
- In user stories already closed, a decision is needed
  - If bug is not much important, simply fix
  - If bug can be costly to fix, reopen the user story
    - Beware with the story points

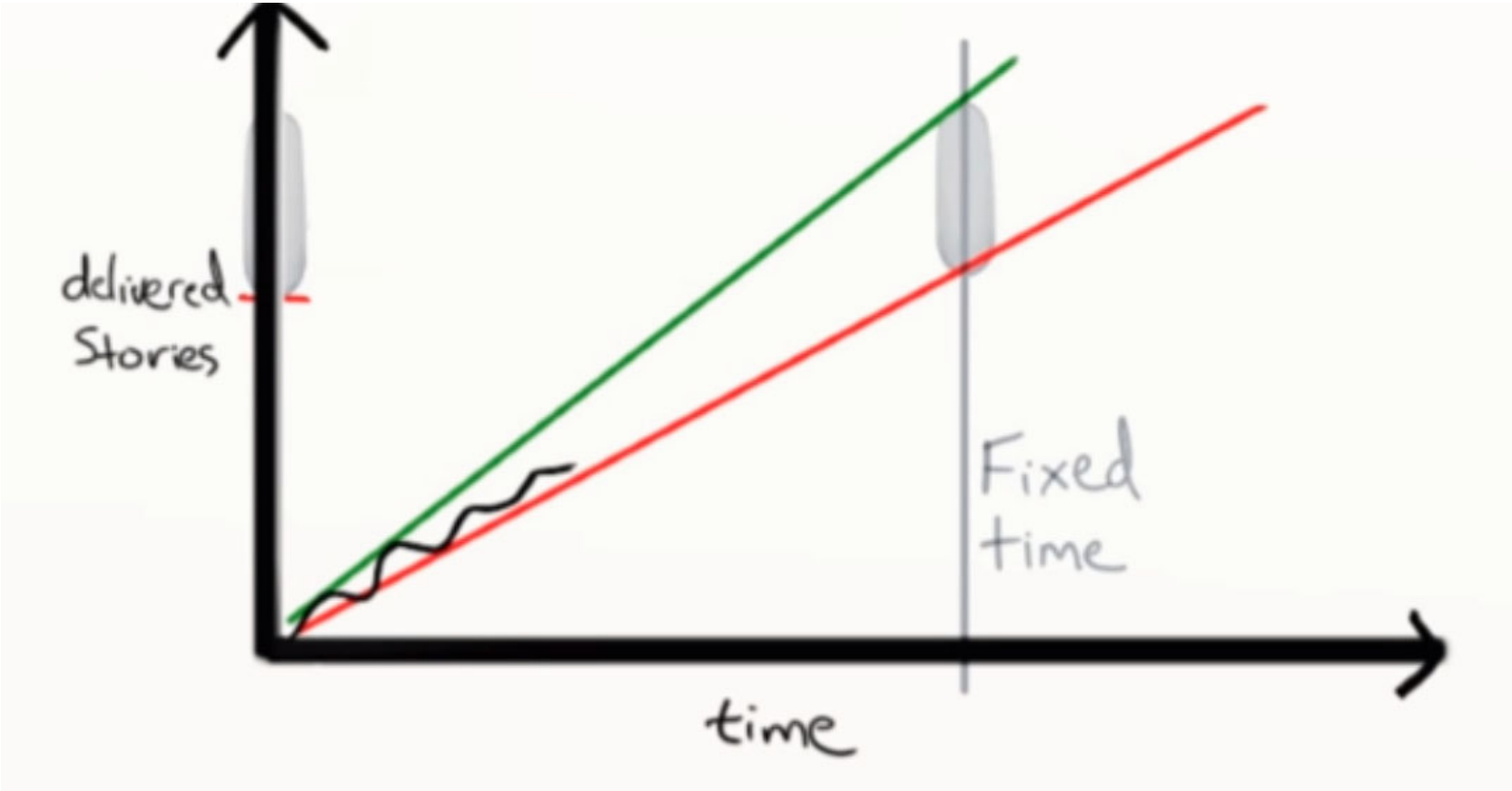
The planning onion

M. Cohn, *Agile Estimating and Planning.* Prentice Hall PTR, 2006.

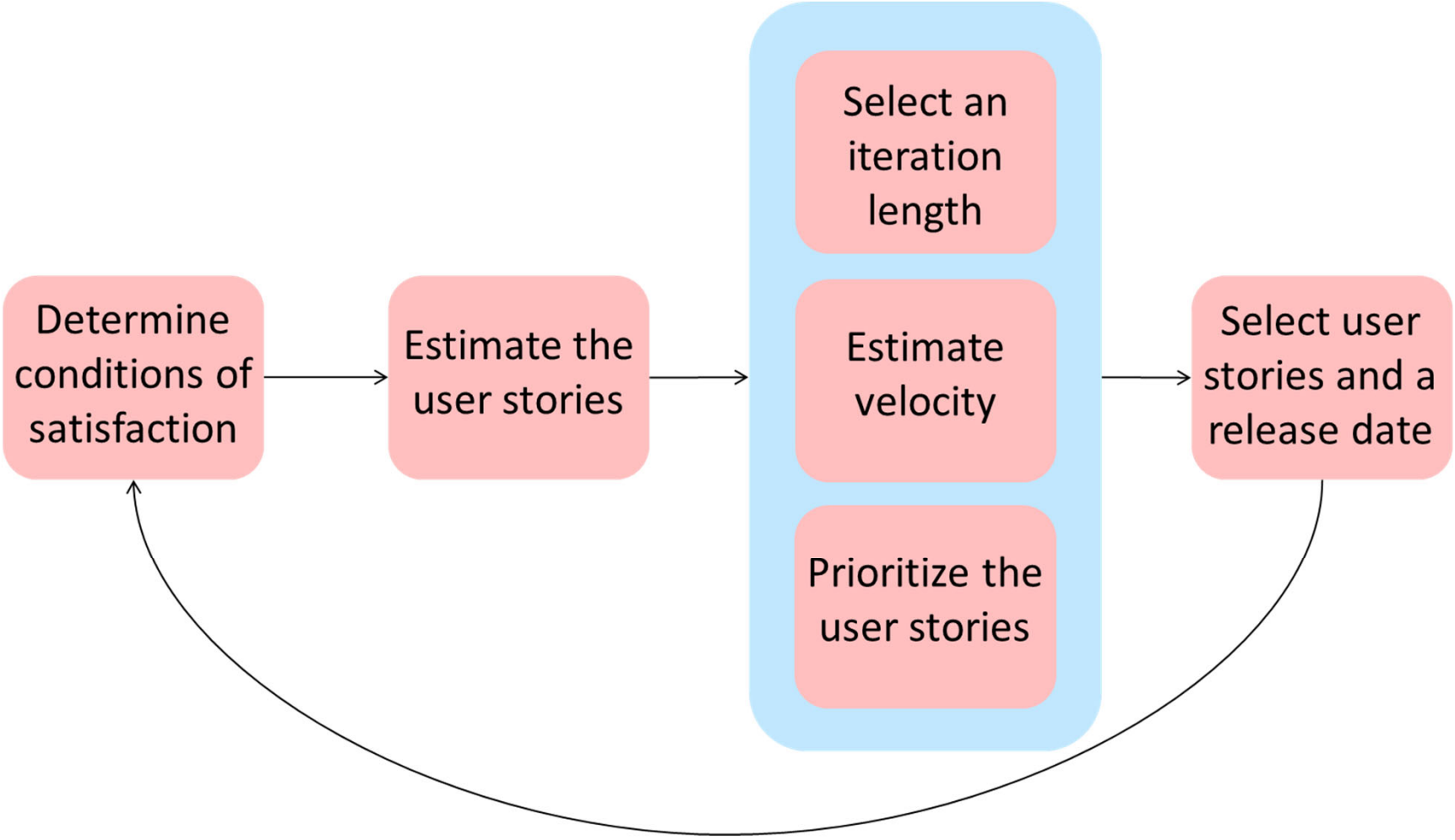
Agile planning: Scope vs. time



Agile planning: Scope vs. time



Release planning



From M. Cohn. *Agile Estimating and Planning.* Prentice-Hall, 2014

Release planning

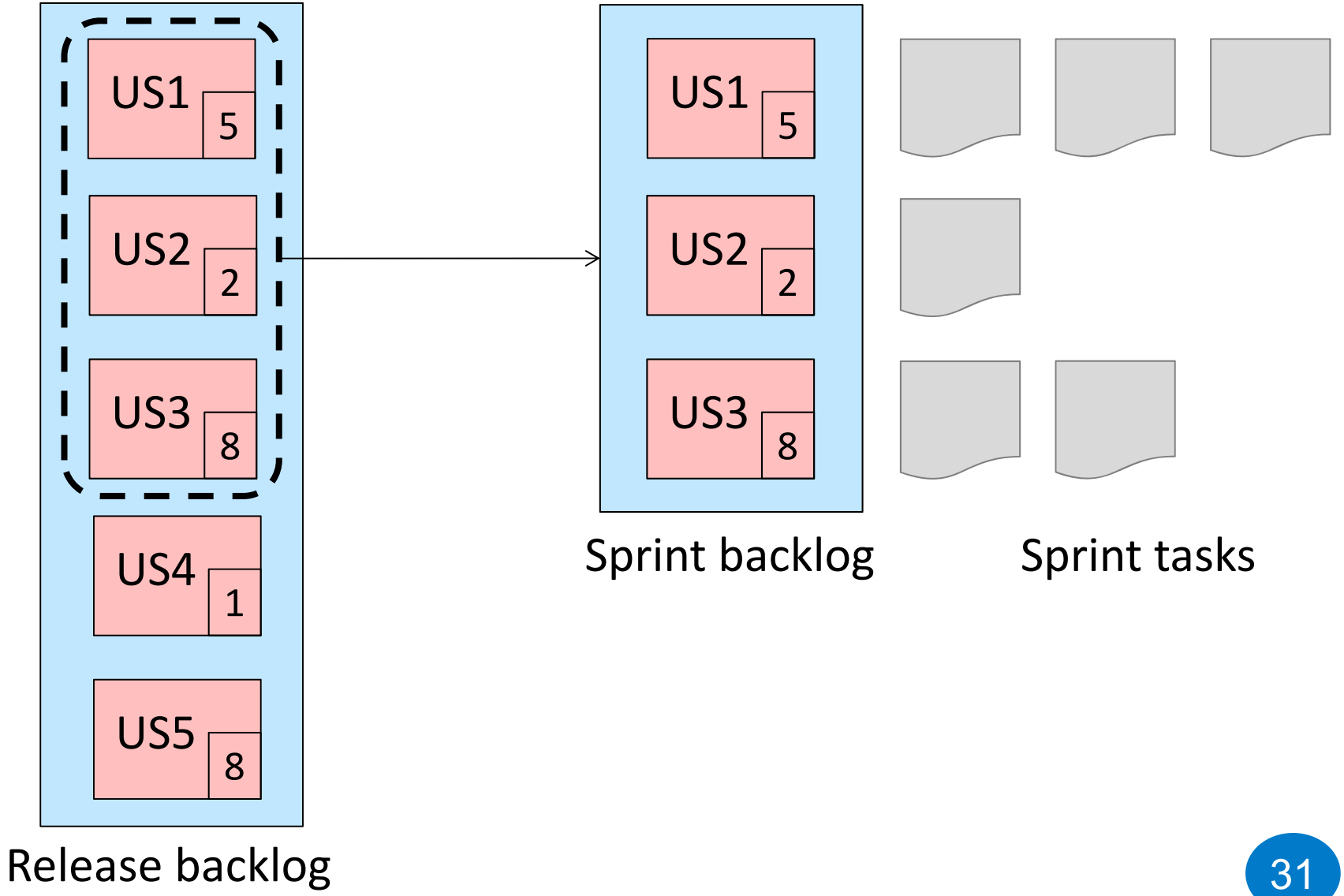
Avoid:

- assignment of concrete individuals to user stories
- decomposition of user stories into activities
  - especially, engineering tasks (code, test, …)
- sequence of work

This information is likely to appear in iteration plans

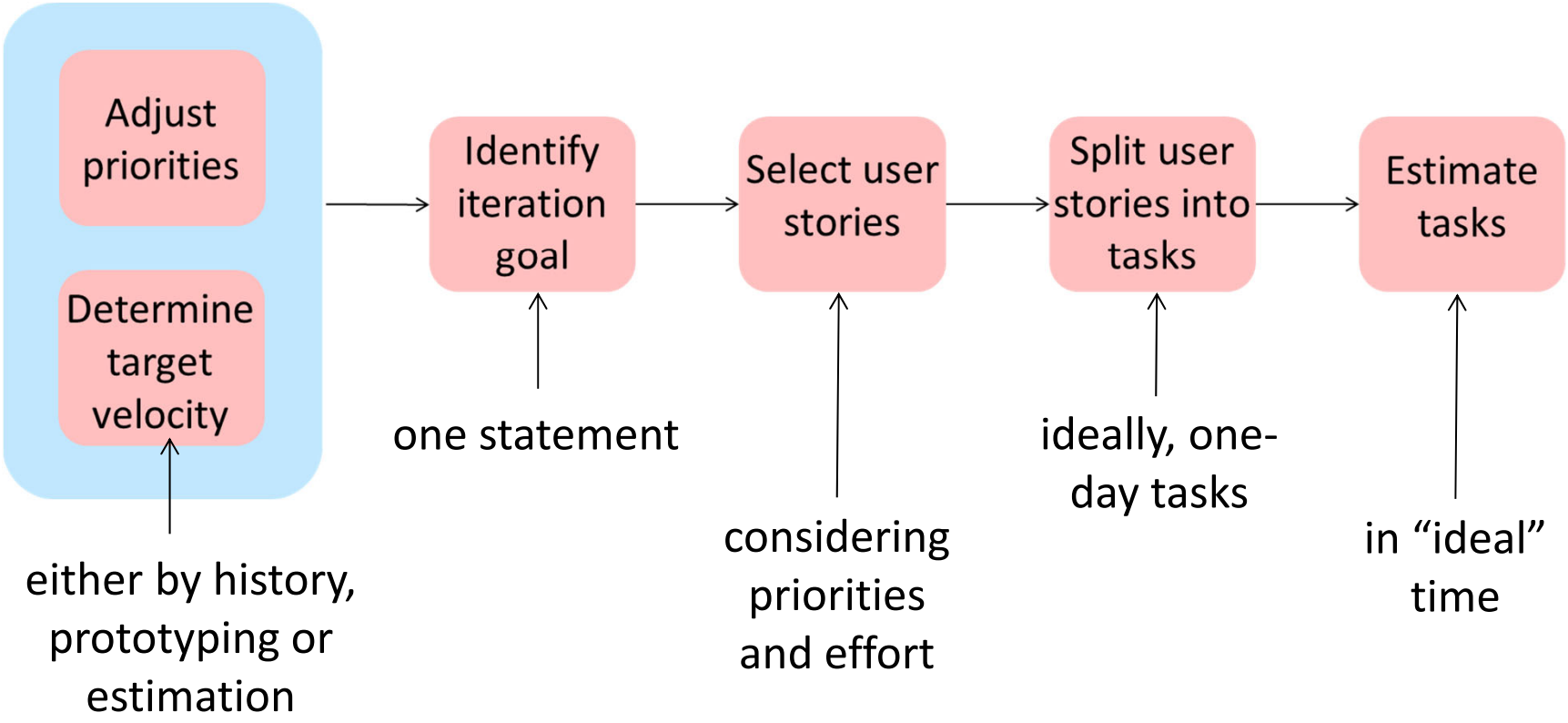
The release planning may (and will) be updated as the project proceeds

Iteration (sprint) planning



Iteration (sprint) planning

Velocity-driven approach:



From M. Cohn. *Agile Estimating and Planning.* Prentice-Hall, 2014

Release and iteration planning

Two options:

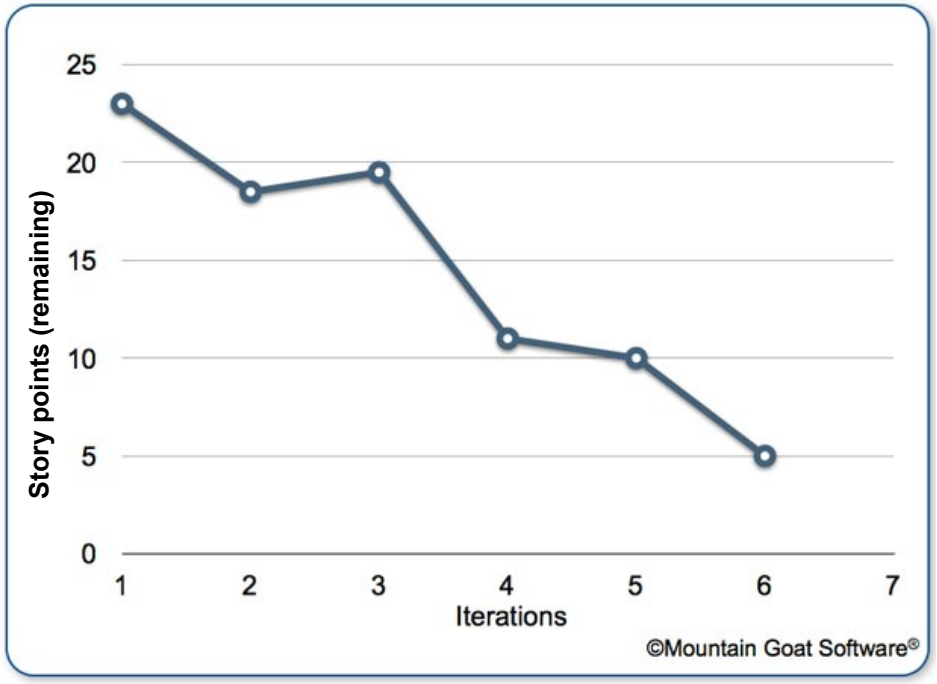
- make them independent
  - less time in the release planning
  - more adaptative
- include some iteration planning when developing the release planning
  - typically, the first two or three iterations

This information is likely to appear in iteration plans

Monitoring: release level

Release burndown chart – simplified

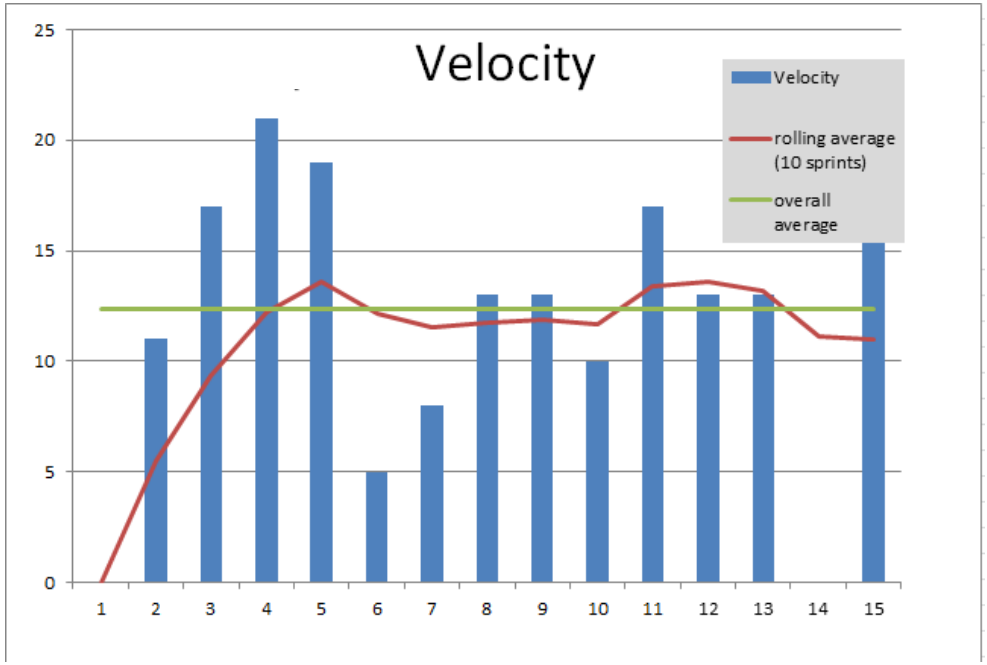
- Visualizes the progress of the release in effort
  - focus on total remaining story points
  - helps to estimate velocity and adjust estimation



Monitoring: release level

Velocity chart

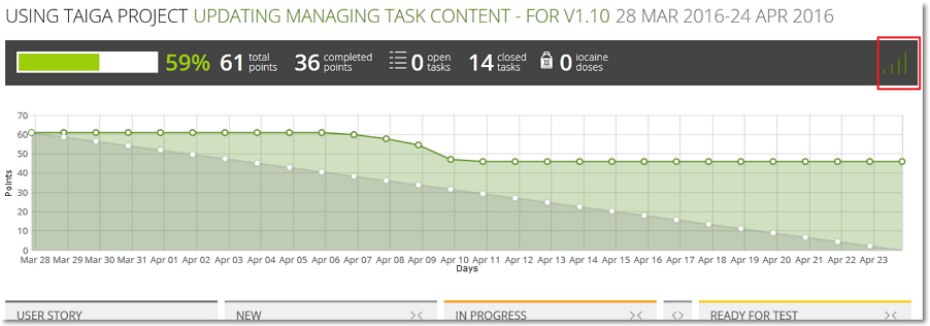
- Focuses on velocity only
  - some special values may be highlighted
    - e.g., mean of the last X iterations, mean of the worst Y iterations



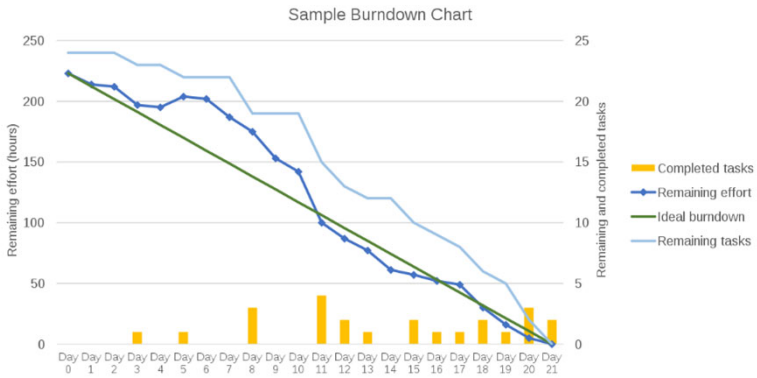
Monitoring: iteration level

Iteration burndown chart

- Visualizes the progress of the iteration in effort
  - focus on **total completed story points**
  - re-estimated daily
  - no "punishing" purposes, only helps to self-organization



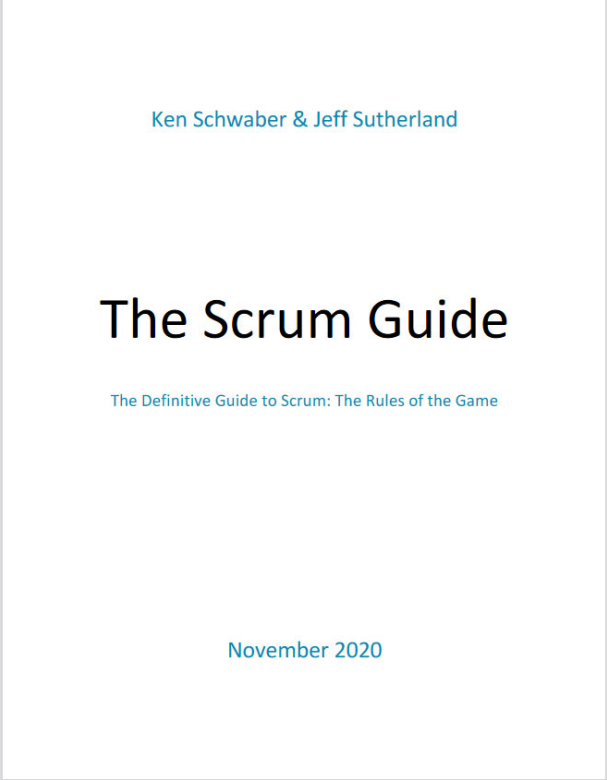
From http://scrumreferencecard.com/scrum-reference-card



https://upload.wikimedia.org/wikipedia/commons/thumb/8/8e/SampleBurndownChart.svg/1200px-SampleBurndownChart.svg.png

Scrum guides

- More information? See these two Scrum guides with up to 20 pages

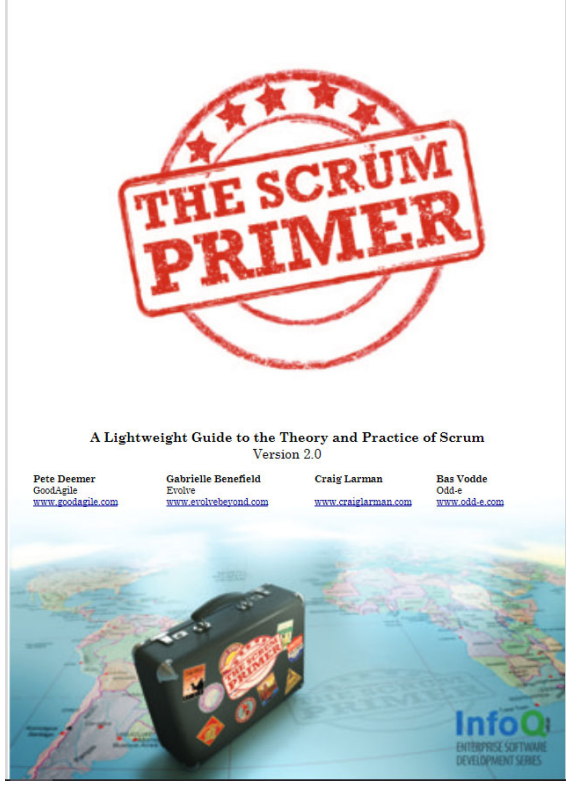

Ken Schwaber & Jeff Sutherland

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

November 2020

https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf



https://scrumprimer.org/scrumprimer20_small.pdf

Projecte d'Enginyeria del Software: Metodologies àgils

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona