IT Project Management

Lecture 4 – Agile Approach

Uwe Gühl

Contents

Contents

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

Source: https://agilemanifesto.org/

Agile Approaches

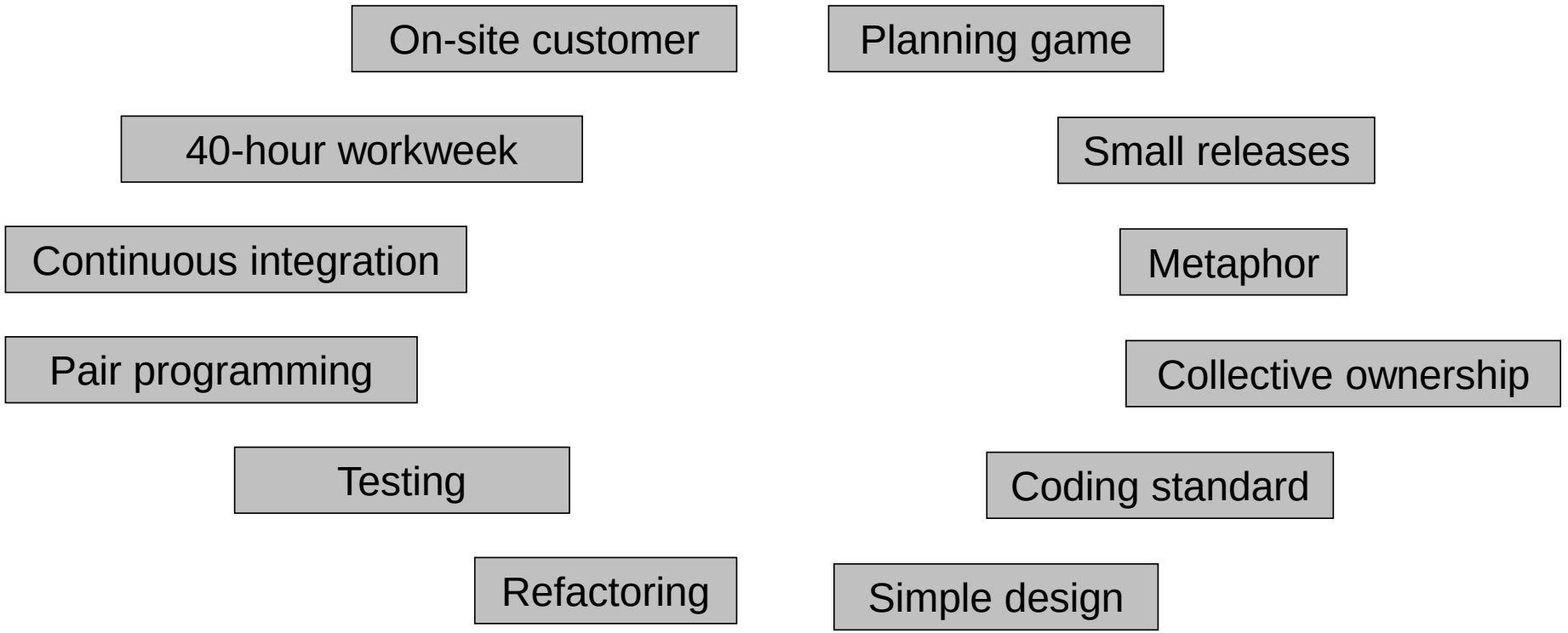
- Agile software development comprises various approaches to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s) [Col11]

Agile Approaches

- Kanban (literally, “signboard”) [Cle16]
  - Part of the Toyota Production System
  - Goal: Create an environment of “just-in-time” delivery – originally within the manufacturing context, but later adapted to software development and general project management
- Extreme programming (XP) [Bec99]
  - software development methodology which is intended to improve software quality and responsiveness to changing customer requirements.

Agile Approaches

- Extreme programming (XP) [Bec99]
  - 12 core practices



Agile Approaches

Planning game

Small releases

Metaphor

- Extreme programming (XP) [Bec99]
  - Planning game
    - Common release planning based on user stories
    - Prioritization by customer – Effort guess by developers
  - Small releases
    - About every +/- 4 weeks to get early customer feedback
  - Metaphor
    Simple story how the system should work instead of a complex architecture description

Agile Approaches

Collective ownership

- Extreme programming (XP) [Bec99]

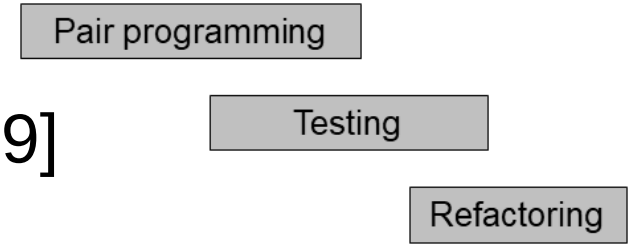
Coding standard

  - Collective ownership
    Everybody could change code everywhere

Simple design

  - Coding standards
  - Simple design
    - Design and code as simple as possible
    - Not needed code gets deleted immediately
    - Implement only what is needed to fulfill an user story

Agile Approaches



- Extreme programming (XP) [Bec99]
  - Refactoring
    - Every time when it is detected that the design could be improved, it has to be done
    - Unit tests assure, that the functionality still works
  - Testing – test driven development
    - Developers write (to be automated) unit tests before coding ("Test-First"-approach)
    - Customer defines parallel functionality tests
  - Pair programming
    - 2 developers work together
    - Always at least 2 know the code
    - Change of roles as necessary (other user stories)

Agile Approaches

On-site customer

- Extreme programming (XP) [Bec99]

40-hour workweek

  - Continuous integration

Continuous integration

    - If a user story is done, it gets integrated in the whole system
    - Testing before and after integration to ensure functionality
  - 40 hour workweek
    - Sustainable pace: The big needs in XP lead to intensive work, so that overtime should not be done
  - On-site customer
    - A representative of the customer is always available to discuss / answer questions and to get decisions concerning user stories and test

Agile Approaches

- Others:
  - Crystal Clear,
  - Feature Driven Development,
  - Dynamic Systems Development Method,
  - ...

- In a survey with different organizations world wide it was reported that about 97 % reported that agile development methods are in use [VO19]

Agile Approaches

Reasons for using agile methodologies [VO19]

Image source: https://www.stateofagile.com

Agile Approaches

Which agile methodologies are used? [VO19]

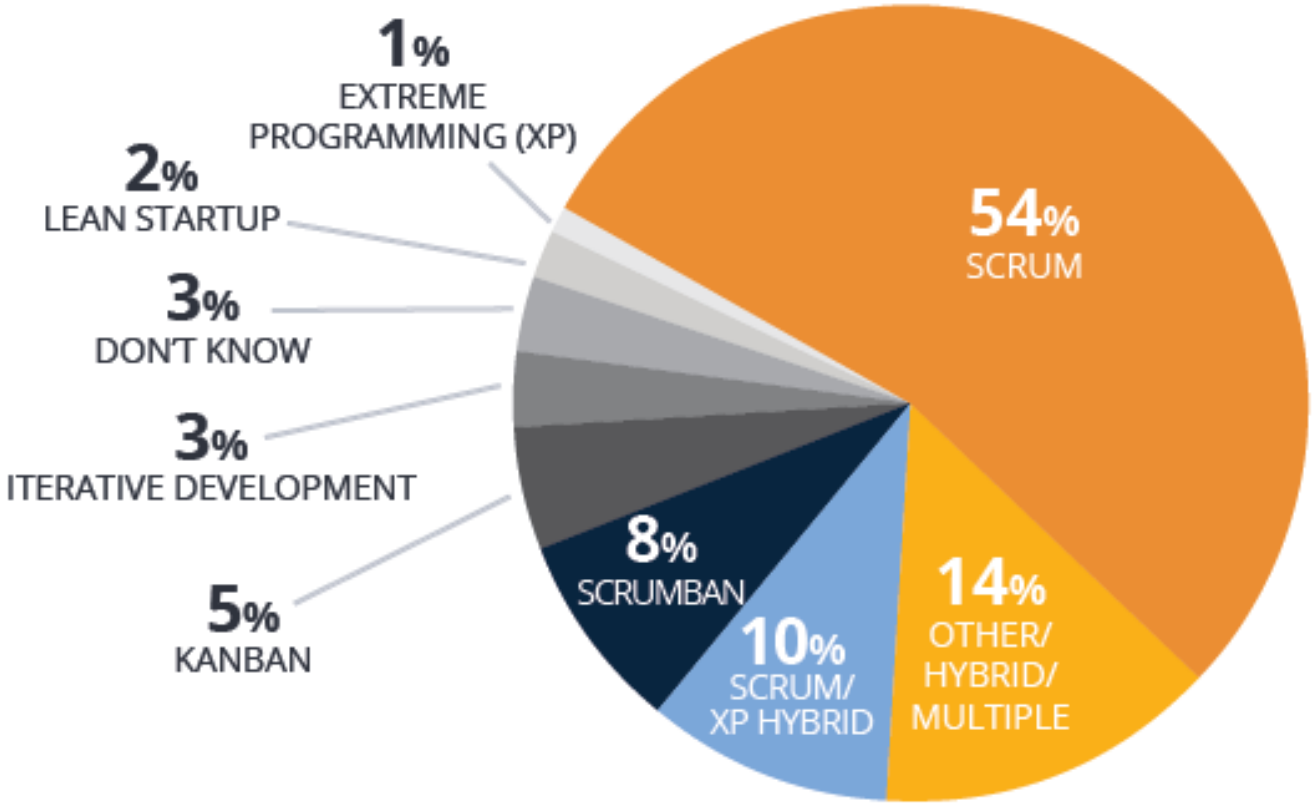
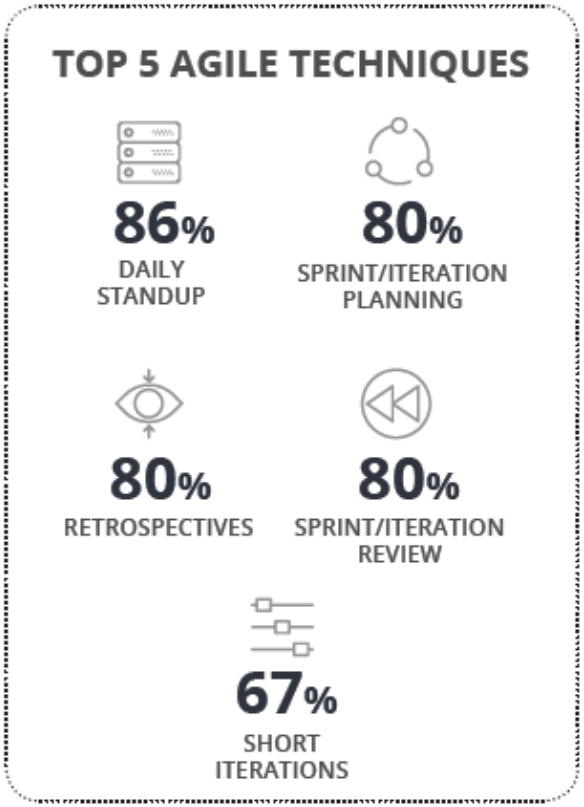
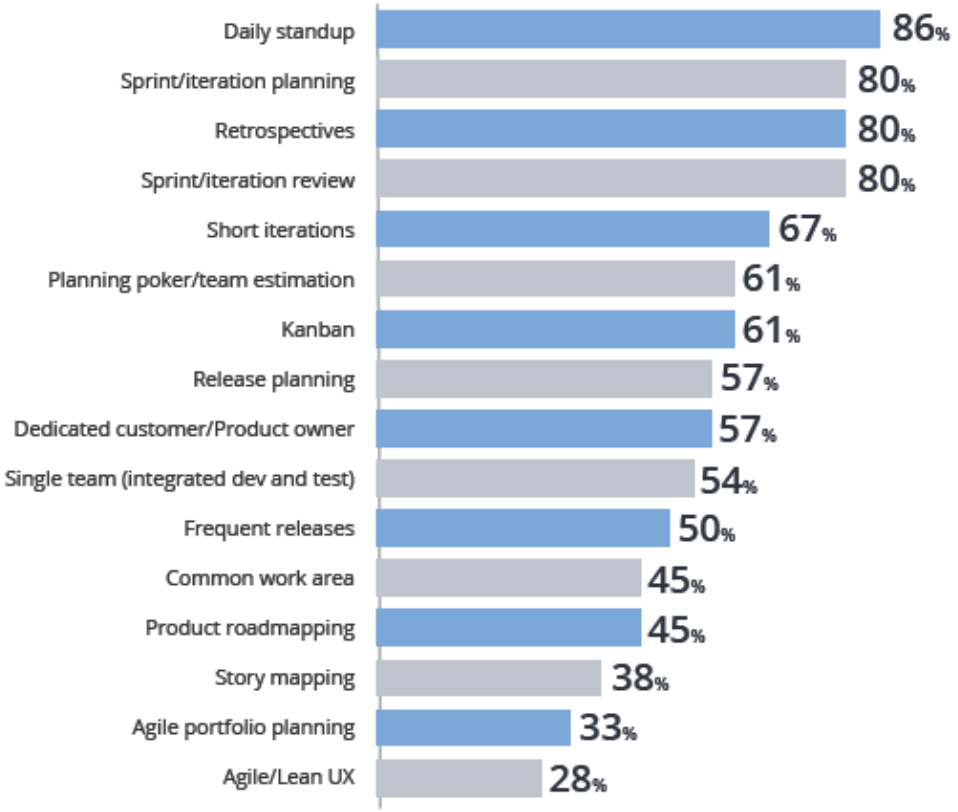


Image source: https://www.stateofagile.com

Agile Approaches

Which agile techniques are used? [VO19]





*Respondents were able to make multiple selections

Image source: https://www.stateofagile.com

Agile Approaches

Benefits of adopting agile [VO19]

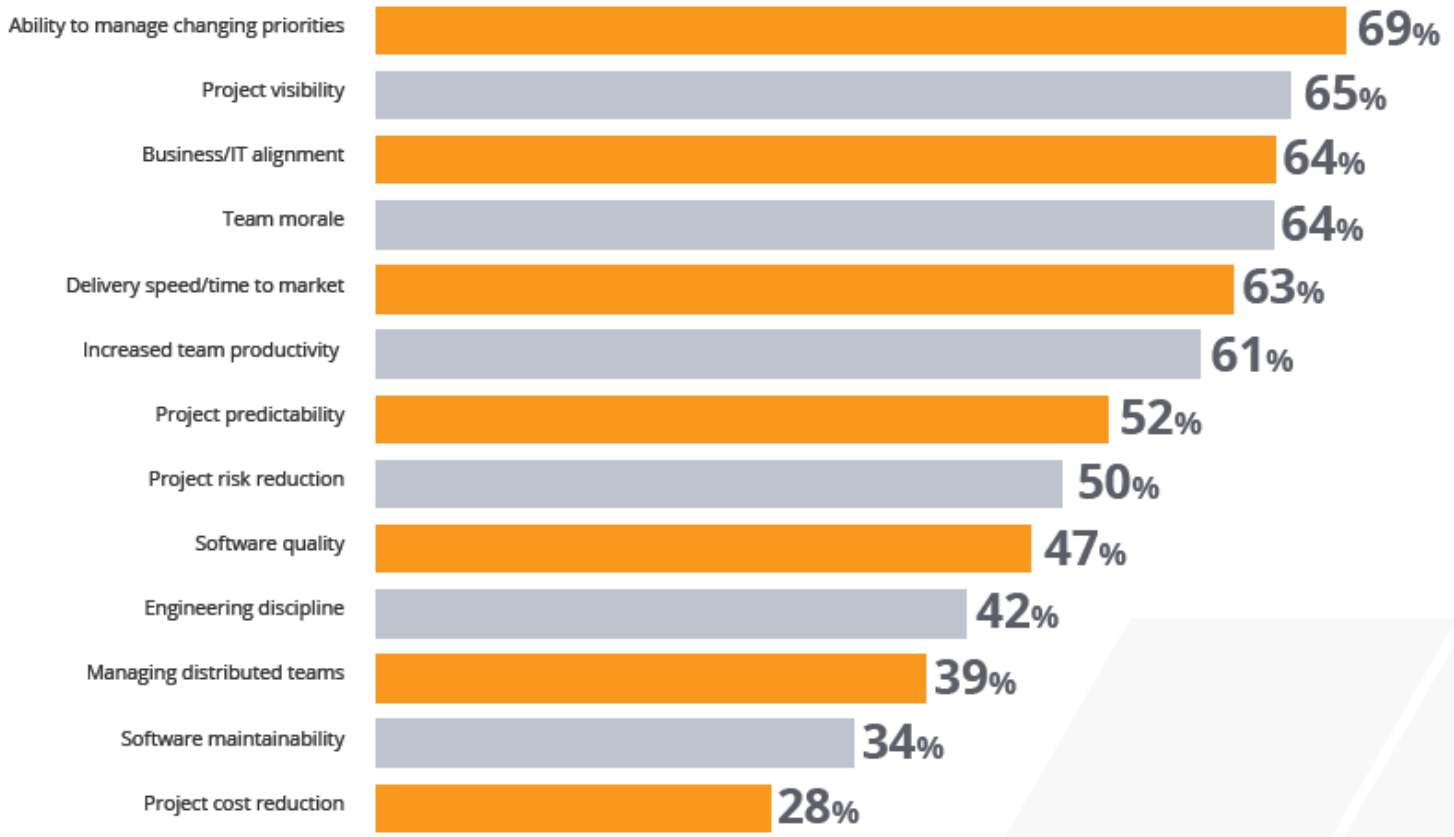


Image source: https://www.stateofagile.com

Contents

Scrum

- Scrum is an iterative, incremental framework for project management [SchS95]
- Presentation is based on the Scrum Guide™ [SchS17]

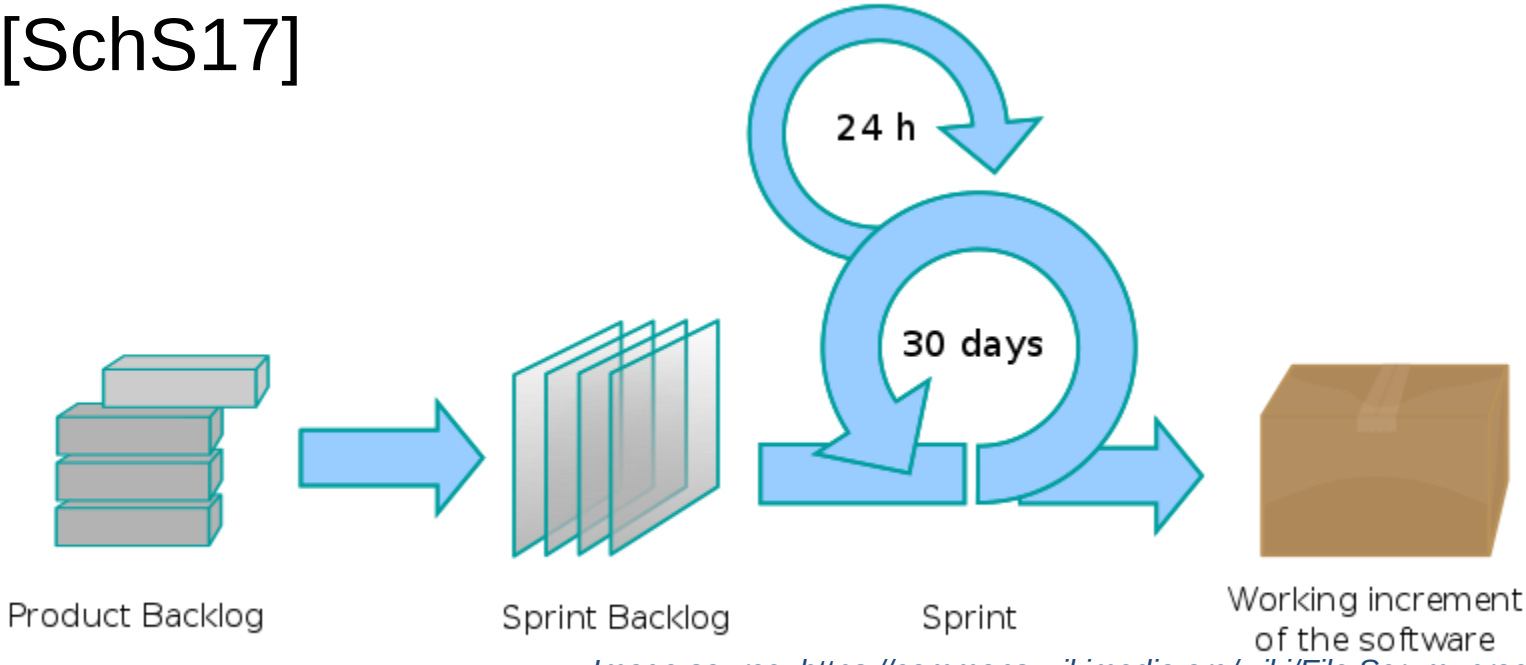


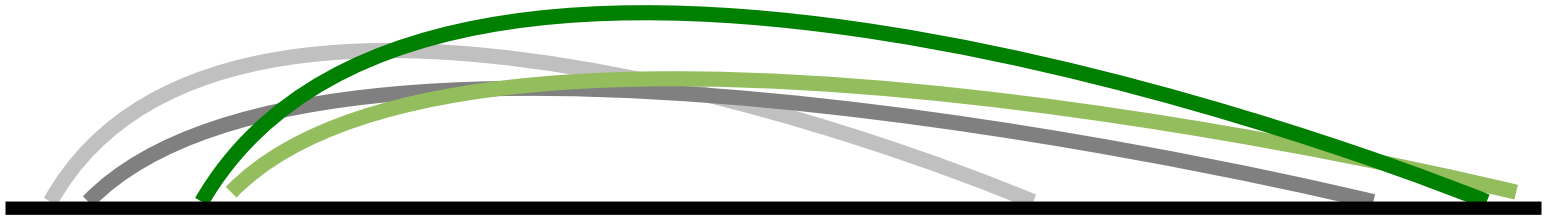
Image source: https://commons.wikimedia.org/wiki/File:Scrum_process.svg

Process Model – Scrum

Requirements | Design | Code | Test

Rather than doing all of one thing at a time...

...Scrum teams do a little of everything all the time



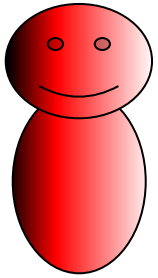
Contents

Scrum Team

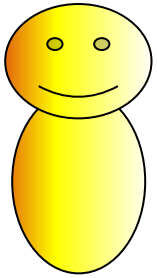
- consists of
  - Product Owner,
  - Scrum Master,
  - Development Team.
- self-organizing and cross-functional

Scrum Team

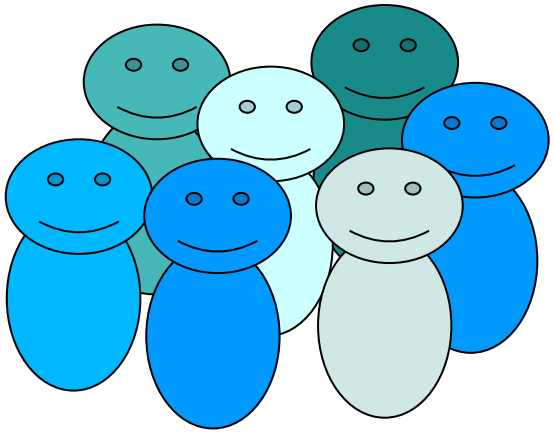
- responsible for managing the Product Backlog
- represents the interests of the stakeholders



Product Owner



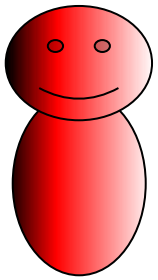
Scrum Master



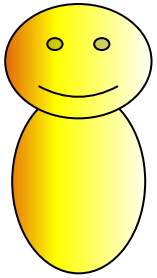
Development Team

Scrum Team

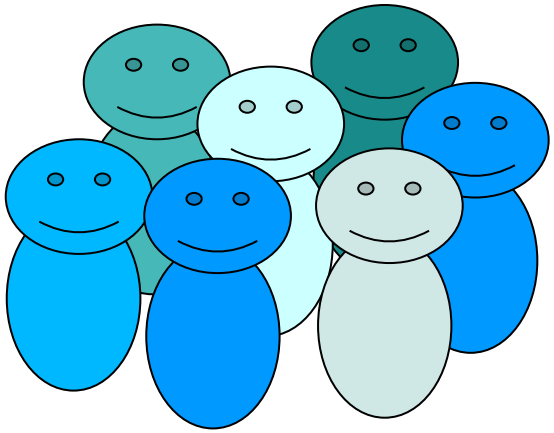
- Servant-leader for the Scrum Team
- responsible for the Scrum process, maximizing its benefits



Product Owner



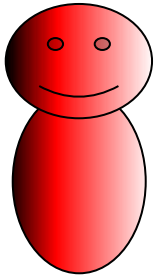
Scrum Master



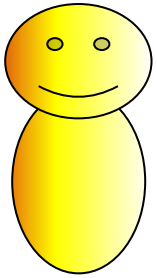
Development Team

Scrum Team

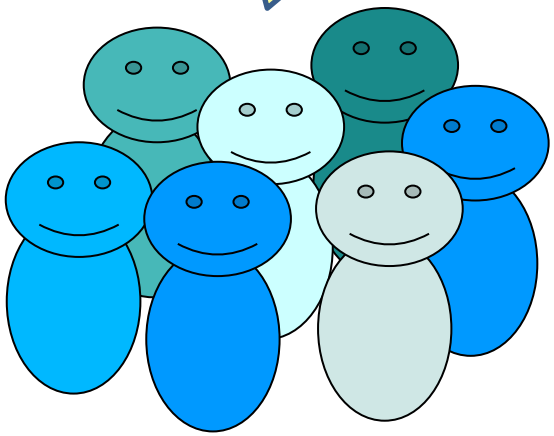
- responsible to develop the product
- self-organizing



Product Owner



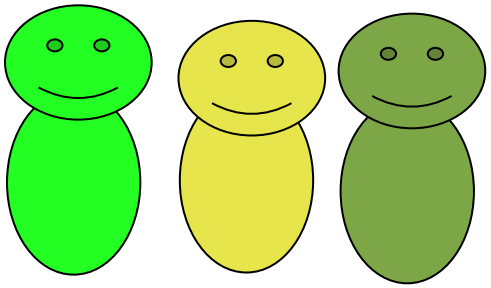
Scrum Master



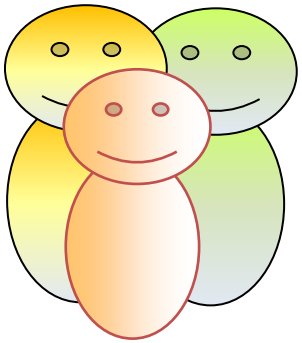
Development Team

Stakeholder

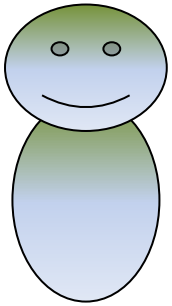
- roles outside of Scrum
- should be invited to Review meetings
- could support in definition and revision of user stories, e.g. in the Product Backlog Refinement



Customer



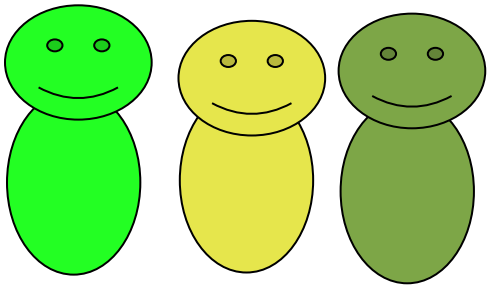
Users



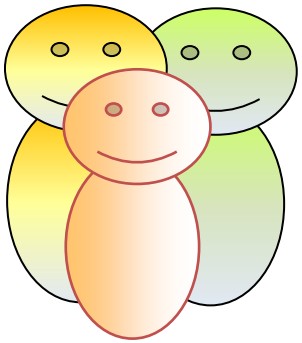
Management

Stakeholder

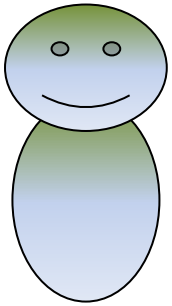
- could be business departments, external people
- should be in close relationship with the Product Owner



Customer



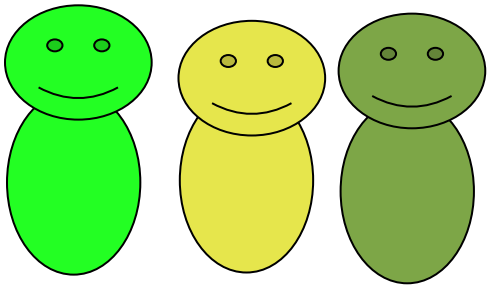
Users



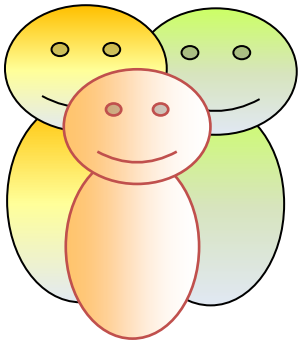
Management

Stakeholder

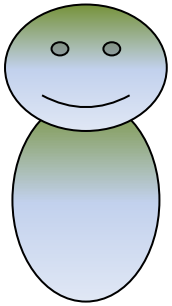
- People using the product
- could be customer, but also service and operation



Customer



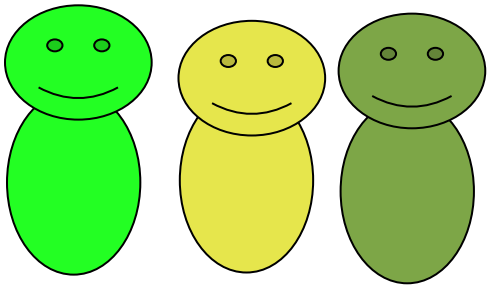
Users



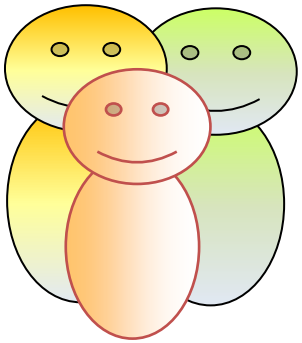
Management

Stakeholder

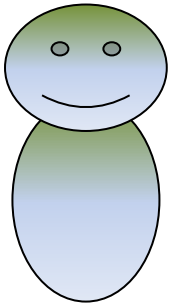
- Main task: supporting the Scrum team, e.g., with tools and rooms



Customer



Users



Management

Contents

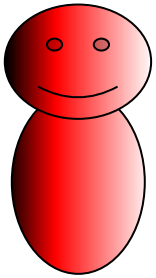
User Stories

- User stories to handle requirements
- During the project the Product Owner works with user stories
  - writes new ones,
  - deletes old,
  - changes,
  - prioritizes.
- Could be small piece of paper
- Template:
  “As a <role>, I want <desire> so that <benefit>“
- Writing good user stories using the INVEST model [Wak03]

User Story Template
As a <type of user>
I want <some goal>
so that <some reason>

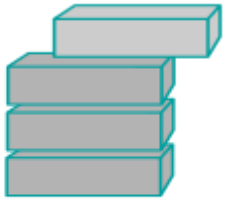
2.1

Product Backlog



Product Owner

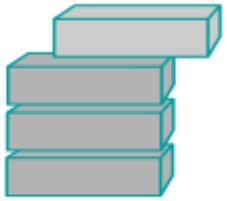
- Product Owner is responsible for the Product Backlog, including its content, availability, and ordering
- Contains backlog items:
  Collection of user stories as basic wish list what makes the product great.
- Lists all features, functions, requirements, enhancements, and fixes for the product in future releases
- Evolves through the project



Product Backlog

Product Backlog

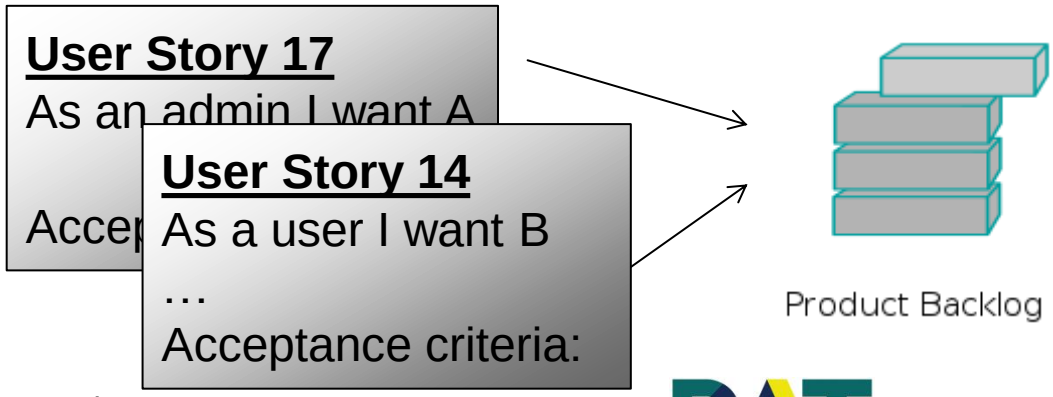
- Basis for Sprint Planning
- Ordered: Higher ordered Product Backlog items are usually clearer and more detailed than lower ordered ones.
  => High ordered items are candidates for the Sprint Backlog



Product Backlog

Product Backlog

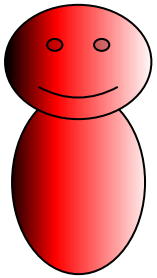
- When are user stories ready to be taken into the Product Backlog?
- ***Definition of Ready:***
  set of agreements to start with a user story, e.g.,
  - Subject and description are available
  - Preconditions and required input available
  - Acceptance criteria are defined



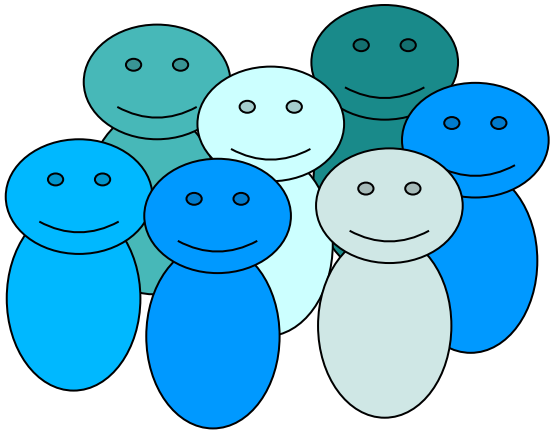
Product Backlog Refinement

Synonym: Backlog Grooming

- Goal: Review and revise Product Backlog items
- Ongoing process between Product Owner and Development Team
- Stakeholder could provide valuable input



Product Owner



Development Team

Product Backlog Refinement

- Scope
  - Sorting the entries
  - Delete entries that are no longer important
  - Adding new entries
  - Detailing Entries
  - Merging entries
  - Estimating entries
  - Release planning

Contents

Sprint Planning

- The purpose of each Sprint is to deliver Increments of potentially releasable functionality
- Goal: Agreement, how many and which user stories to be implemented in next sprint – following prioritization by customer

Sprint Planning

- Rule of thumb:
  Duration of the two parts of Sprint Planning altogether should be two hours for one Sprint week
  - 4 weeks Sprint: 8 hours Sprint Planning
  - 2 weeks Sprint: 4 hours Sprint Planning
- Typically divided in
  - **Part One: Determining the What**
  - **Part Two: Determining the How**
- Following a proposal for proceeding is presented

Sprint Planning – Part One

1a

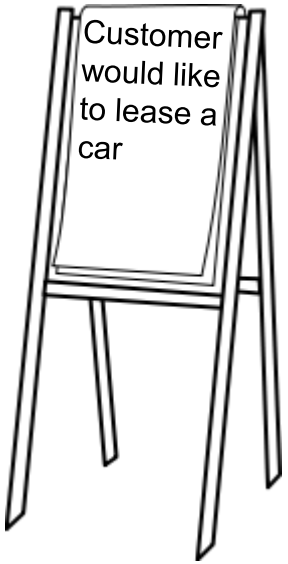
- Determination of how much capacity of the Development Team is available.
  To be subtracted:
  - holidays,
  - sick days,
  - other activities,
  - meetings and
  - possibly buffers.
- Result is the net capacity of the Development Team that could be used to implement backlog items.

Example:
140 person days
./. 10 % holidays
./. 10 % meetings
---
112 person days net
==========

Sprint Planning – Part One

1 b

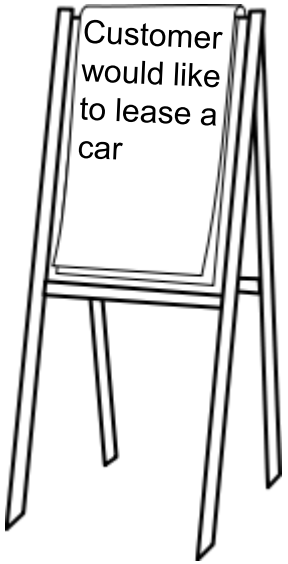
- Presentation of Backlog Items
  - Starting with the first item from the Product Backlog
  - The Product Owner
    - explains the expectations
    - answers all related questions
  - When done the Development Team should be able to estimate the effort to implement it

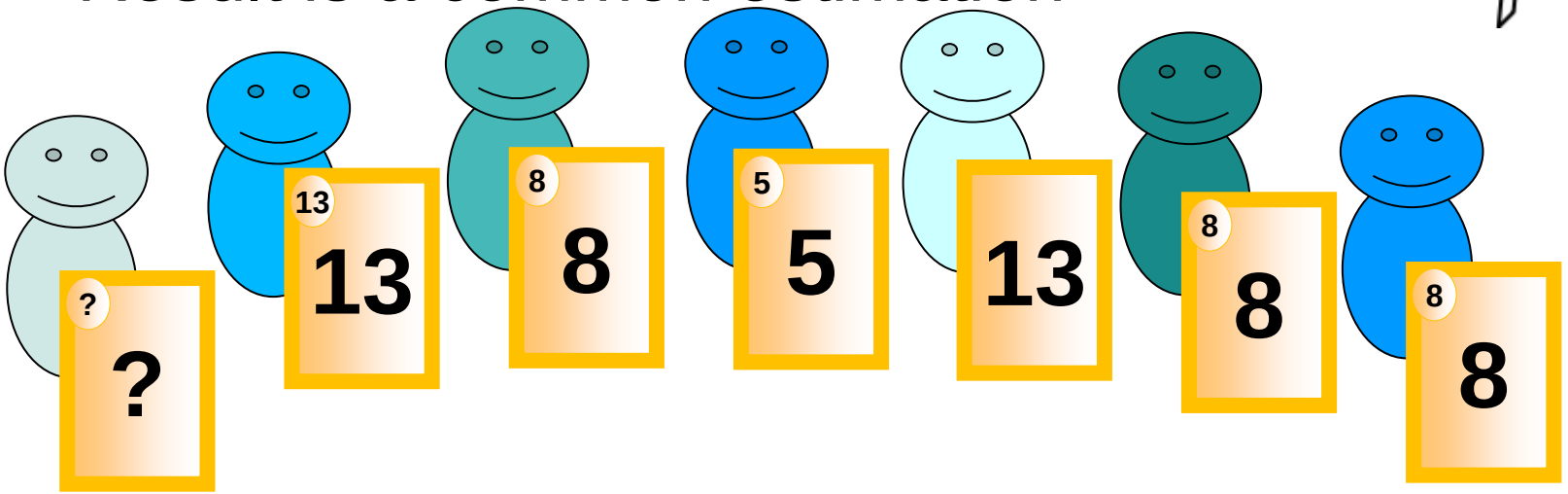


Sprint Planning – Part One

1c

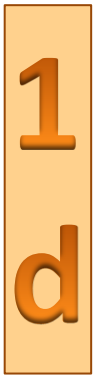
- Planning Poker:
  All team members guess the effort for a specific story
- Scrum master moderates
- Result is a common estimation





Sprint Planning – Part One

- Alternate continuation:
  - Product Owner presents a story
  - Development Team estimates effort
- Finally a number of user stories to be implemented has been defined.



Example:
112 person days (pd)
./. 12 pd Story 17
./. 14 pd Story 14
…..
---
0 person days
==========

User Story 17
As an admin I want A

12 pd

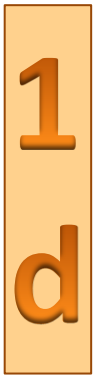
User Story 14
As a user I want B

14 pd

▪ ▪ ▪ ▪ ▪ ▪

Sprint Planning – Part One

- Hint: It might be possible that a story is identified that might be implemented, but could not be planned in a Sprint because, e.g., not enough capacity is left or a critical resource is not available
- Such a backlog item is marked as „Stretch objective (SO)“
- Scheduled but not committed



User Story 19 ***SO***
As a user I want help

4 hours

Sprint Planning – Part One

1e

- Defining a Sprint Goal:
  - Summary of all Backlog Items to be implemented
  - Objective that will be met within the Sprint through the implementation of the Product Backlog
  - Provides guidance to the Development Team on why it is building the Increment.

Sprint Planning – Part One

1
f

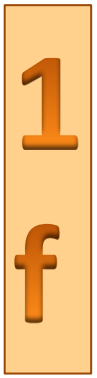
- Based on the Sprint Goal the Development Team gives a commitment concerning the delivery
- E.g., feedback with „Fist of five“ with showing 1 to 5 fingers.
  Will the team meet the Sprint Goal? .

not confident at all

completely confident

Sprint Planning – Part One

- If not all members of the Development Team show a commitment with 4 or 5 fingers:
  - Ask anyone with lower than a four to share their concerns about meeting the commitment.
  - Might result in a quick change to either the plan or the commitment itself to address the concerns of those that were not at a four or five.
  - With the modified plan, we'll re-vote until everyone is at a four or five.



Sprint Planning – Part Two

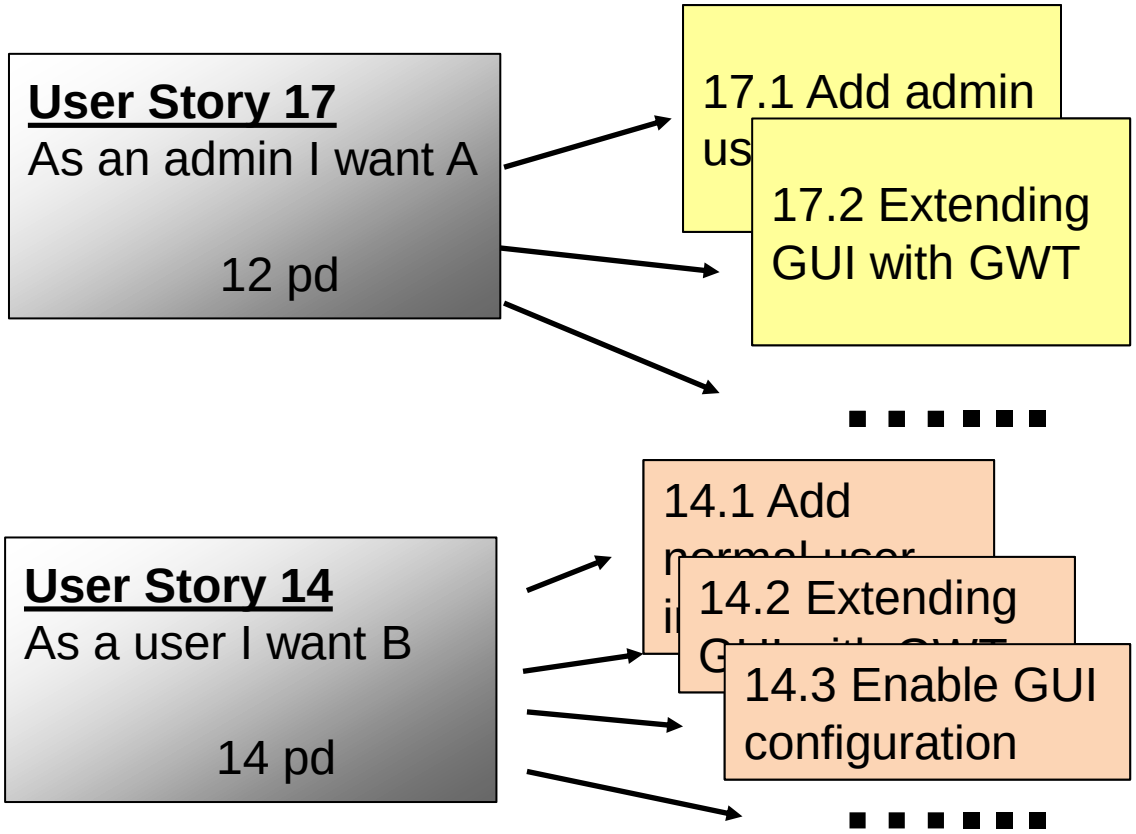
2a

- The Development Team plans in detail the tasks required to deliver the forecasted Product Backlog items.
- The Product owner should support, e.g., with answering questions.
- Clarification of various aspects, like architecture, data structures, interfaces, and so on, are often discussed in small groups.

Sprint Planning – Part Two

2b

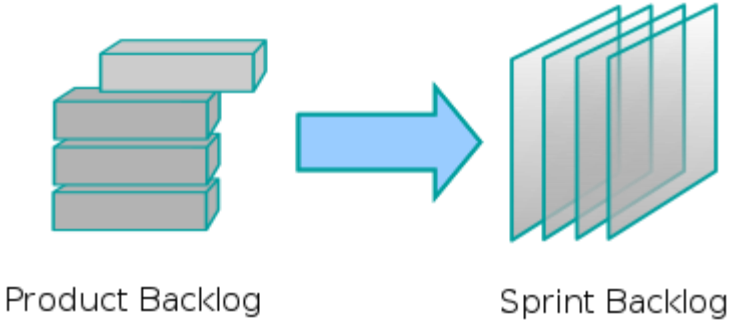
- Finally for all User Stories the corresponding tasks are defined



Sprint Planning – Part Two

2c

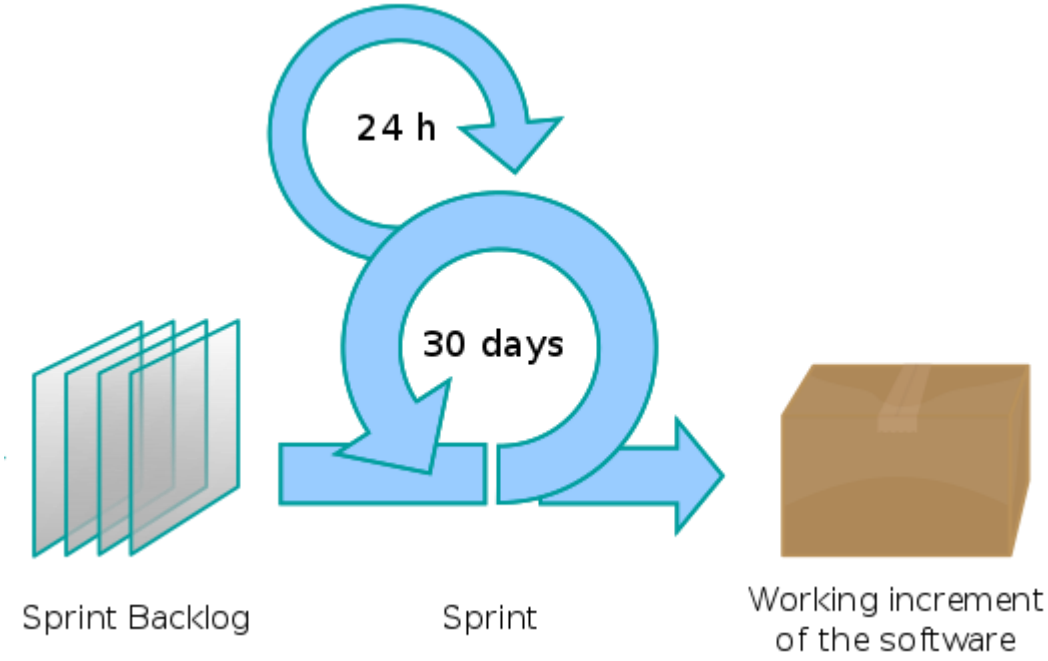
- ***Sprint Backlog***: Agreement what to realize in the next Sprint, contains
  - Product Backlog items to be implemented, following prioritization by customer
  - Tasks related to the Product Backlog items



Contents

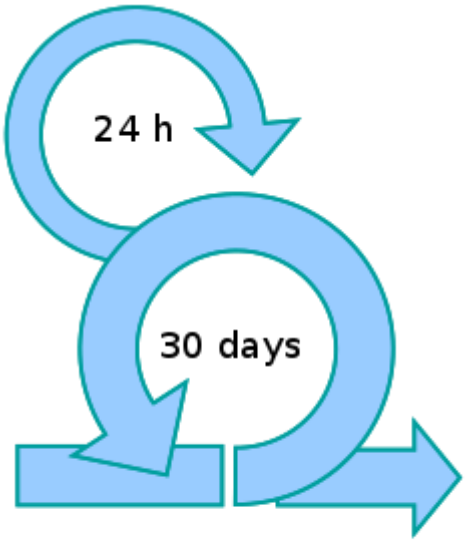
Sprint

- Input: Sprint Backlog
- Output at the end of a Sprint:
  Potentially deliverable product Increment



Sprint

- Goal: All backlog items in the Sprint Backlog should be developed
- Duration of a sprint depends on duration of project and release cycle
  - Typical 2 to 4 weeks



Sprint

Sprint

- When is a user story finished? → Definition of done
  - Everyone must understand what “Done” means to ensure transparency.
  - Evolves as the experience of the Scrum Team increases

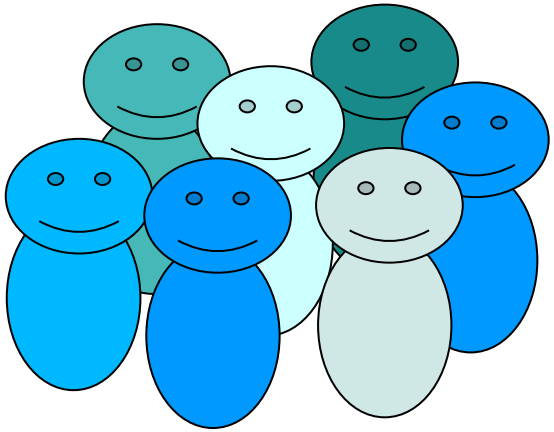
Sprint

- ***Definition of Done***
  - Common understanding of the Scrum Team when a work can be considered done.
  - Scope
    - Quality criteria
    - Constraints
    - General non-functional requirements
  - Example
    - writing comments,
    - writing and executing unit tests,
    - design of documents like system documentation.

Task Board

Visualization of the Sprint Backlog in a Task Board

Example



Development Team

User Story 17
As an admin I want A

12 pd

17.1 Add admin us

17.2 Extending GUI with GWT

User Story 14
As a user I want B

14 pd

14.1 Add

14.2 Extending

14.3 Enable GUI configuration

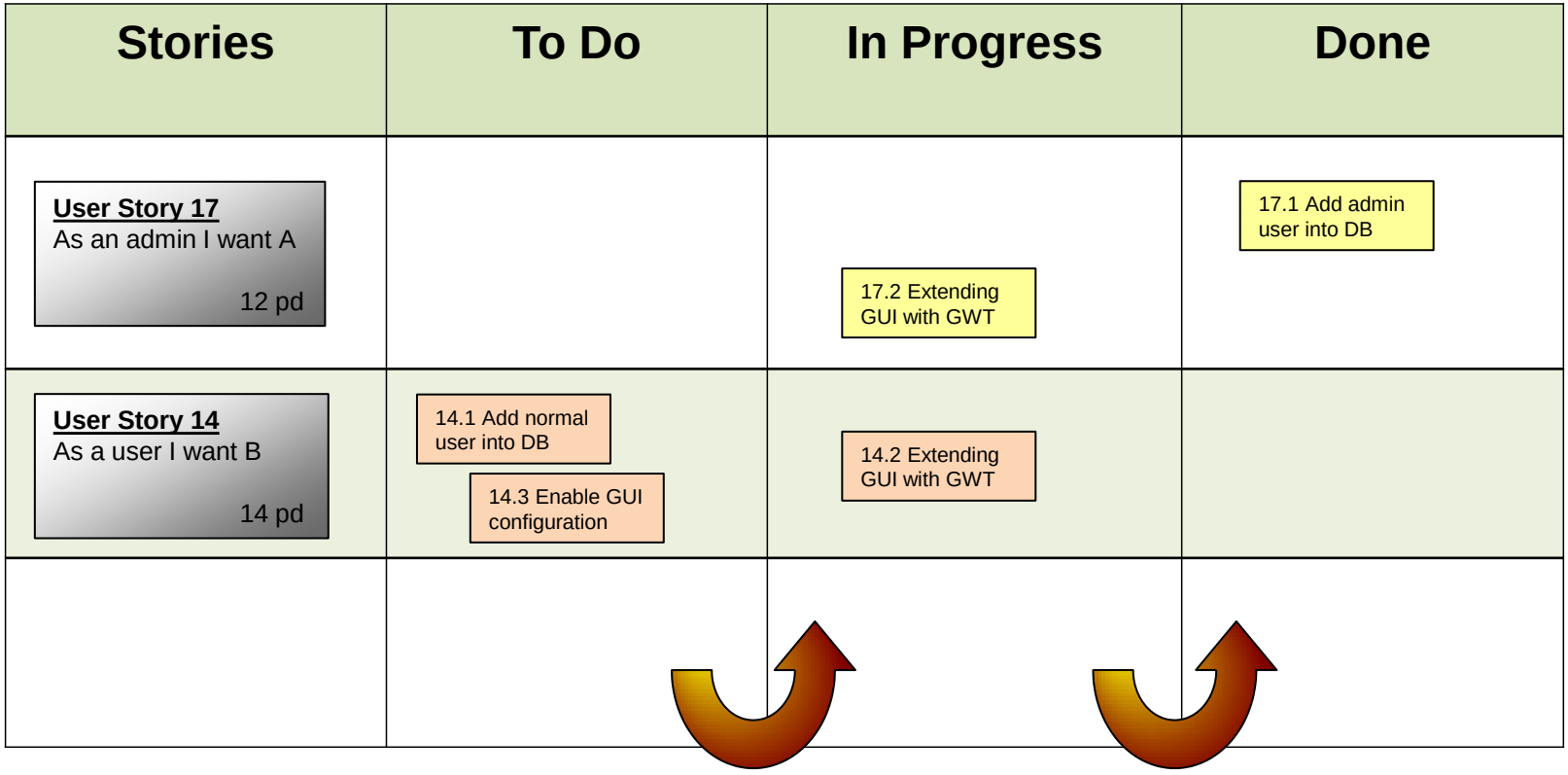
Task Board

Example

Stories	To Do	In Progress	Done
User Story 17 As an admin I want A 12 pd	17.1 Add admin user into DB 17.2 Extending GUI with GWT		
User Story 14 As a user I want B 14 pd	14.1 Add normal user into DB 14.2 Extending GWT 14.3 Enable GUI configuration		

Task Board

Example



Task Board

- Example

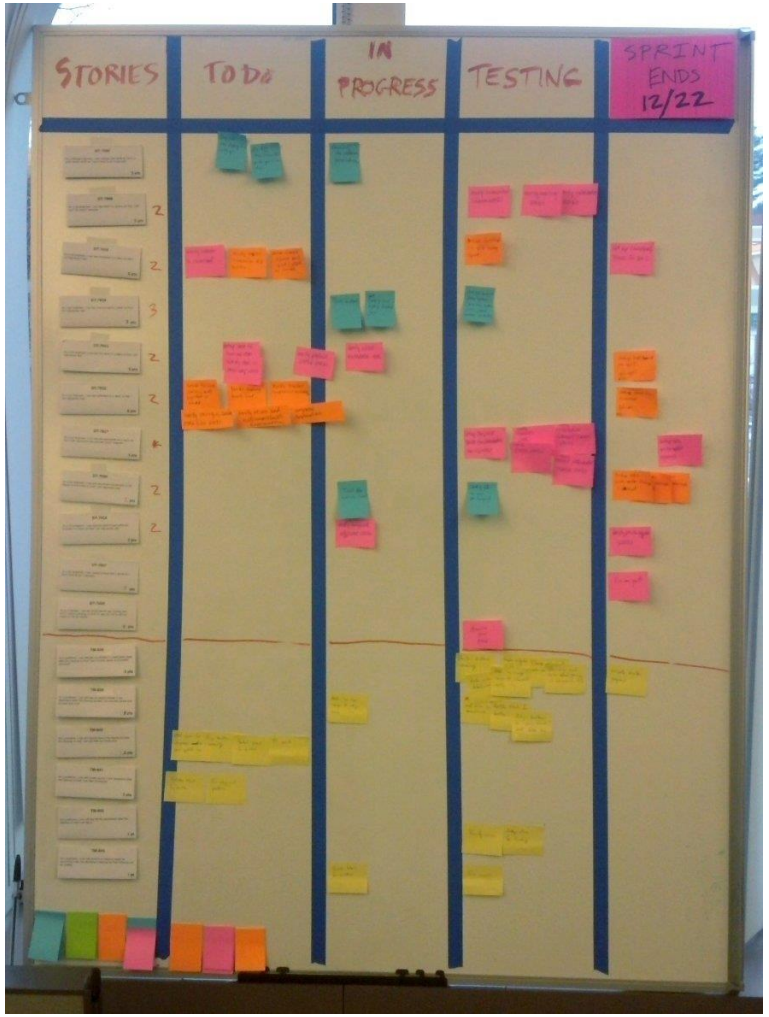
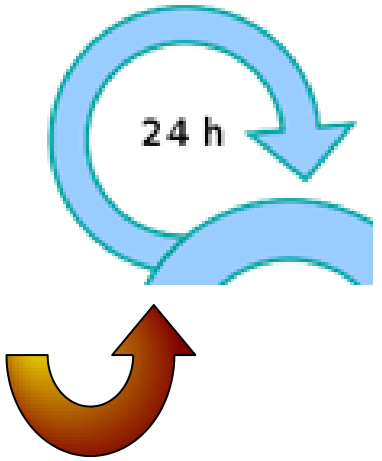


Image source: https://upload.wikimedia.org/wikipedia/commons/1/1b/Scrum_task_board.jpg

Daily Scrum

- Daily stand-up meeting about 15 minutes
- Focus is information exchange, not solving of problems
- Every team member answers related to the tasks on the task board
  - What did I complete yesterday?
  - What do I plan to complete today?
  - Is there any impediment?
- Moving tasks on the task board



Burndown Chart

- Monitoring progress of the project with burndown charts – to show the current status in a sprint
  - Sprint Burndown
    to track the progress of the sprint
  - Release Burndown
    to track product progress across multiple sprints
- Sprint Burndown
  - Horizontal axis: time in days
  - Vertical axis: Remaining effort
    (in days, hours, or number of tasks)

Burndown Chart

- Example

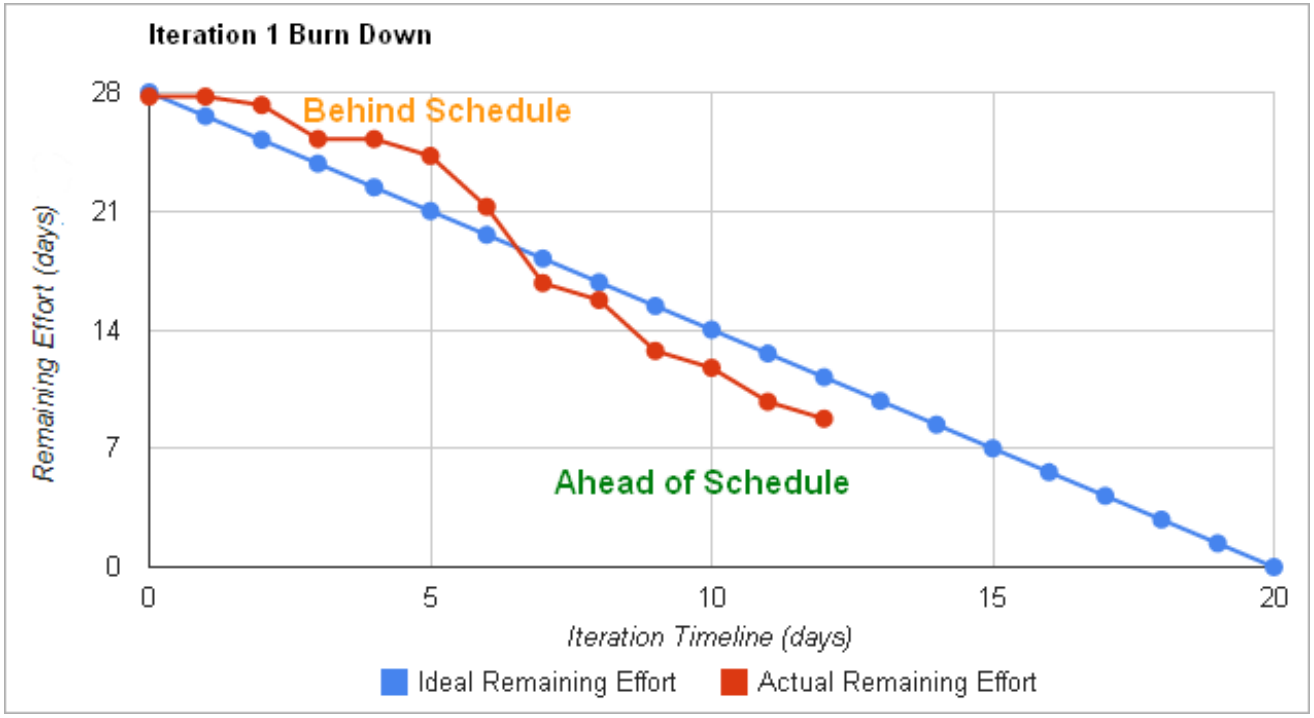
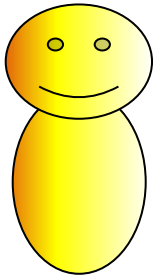


Image source: https://dzone.com/articles/the-ideal-burn-down-chart

Impediment Backlog

- The Scrum Master collects all work impediments
- Goal is to make and keep the Scrum Team working
- Proceeding – two proposals
  1. List of impediments and tasks to solve including status
  2. Task board gets extended to collect impediments as well



Scrum Master

Contents

Sprint Review

- Presentation of the developed Increment
  - Sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints.
- Preconditions
  - the new Increment must be in useable condition
  - All implemented user stories fulfill the “Definition of Done“
- Rule of thumb:
  Duration of Sprint Review should be in hours like the duration of Sprint in weeks
  - 4 weeks Sprint: 4 hours Sprint Review
  - 2 weeks Sprint: 2 hours Sprint Review

Sprint Review

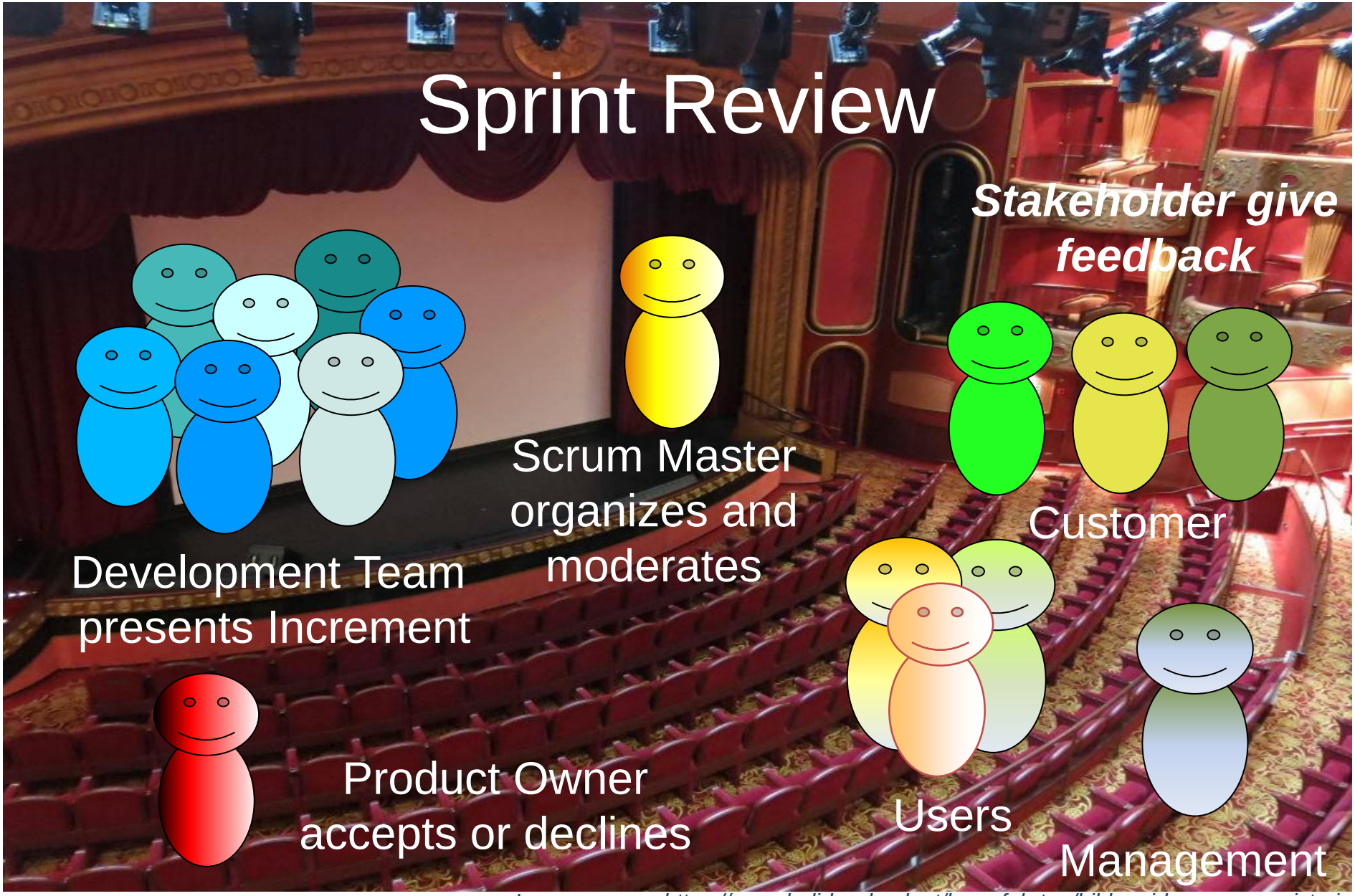


Image source: https://www.holidaycheck.at/kreuzfahrten/bilder-videos-queen-victoria

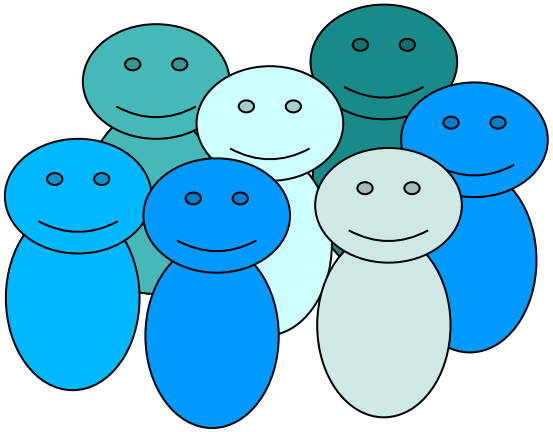
Sprint Review

- Team presents the delivered solution to the Product Owner
- Team members show every implemented user story
- Product owner approves a user story if all related acceptance criteria are fulfilled

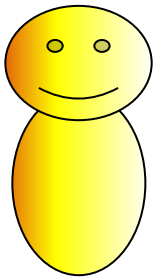
Contents

Sprint Retrospective

- All team members reflect on the past sprint
- Stakeholder participate only if invited
- Make continuous process improvements
- Duration about 45 minutes per Sprint week



Development Team



Scrum Master organizes and moderates

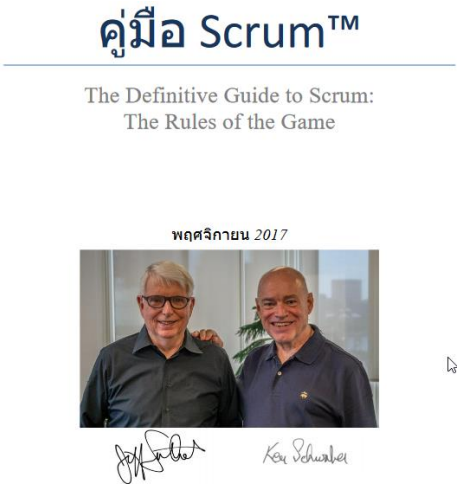
Sprint Retrospective

- Key questions:
  - What went well during the sprint?
    To be continued
  - What did not went will during the sprint?
    To be stopped
  - What should be started?
    To be implemented:
    Practices helping to work better

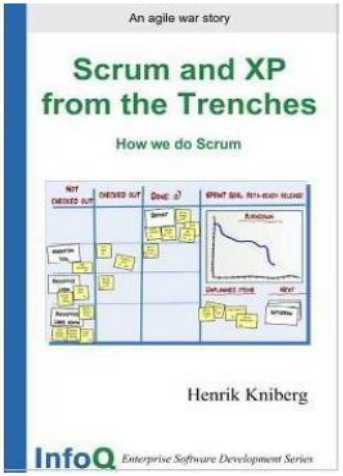
Contents

Want to Learn More?

- [SchS17] Ken Schwaber and Jeff Sutherland: The Scrum Guide, 2017; https://www.scrumguides.org/, available in Thai language

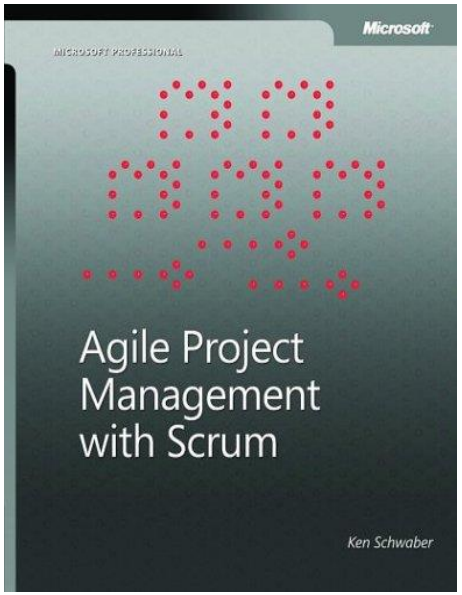


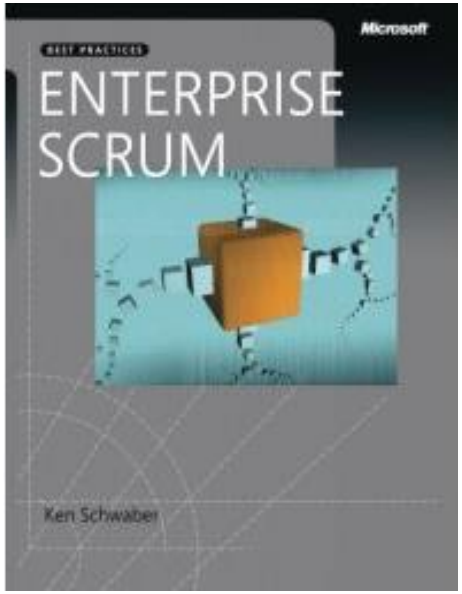
- [Kni15] Henrik Kniberg: “Scrum and XP from the Trenches - How we do Scrum”, 2nd edition, C4media, 2015, free version available (registration is required): https://www.infoq.com/minibooks/scrum-xp-from-the-trenches-2/

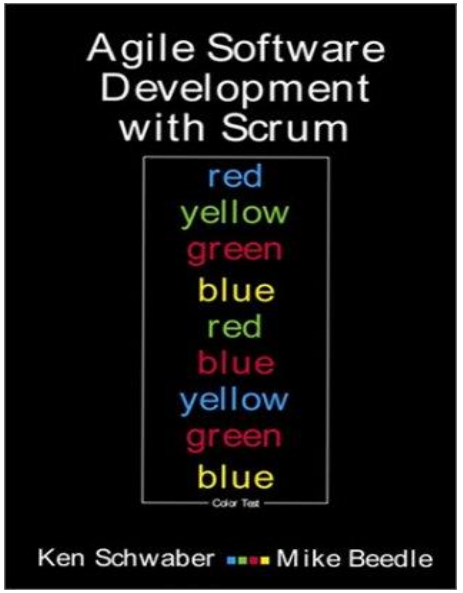


Want to Learn More?

- Ken Schwaber is author of three books about Scrum







Scrum Organizations

- There are several Scrum organizations around that offer certifications
- Two are presented as examples
  - Scrum Alliance
  - Scrum.org

Scrum Organizations

- Scrum Alliance
  https://www.scrumalliance.org/
  offers certifications like
  - Certified ScrumMaster®
  - Advanced Certified ScrumMaster
  - Certified Scrum Professional®-ScrumMaster
  - Certified Scrum Product Owner®
  - Advanced Certified Scrum Product Owner
  - Certified Scrum Professional®-Product Owner
  - Certified Scrum Developer®
  - Certified Scrum Professional®

Scrum Organizations

- Scrum.org
  https://www.scrum.org
  offers certifications like
  - Professional Scrum Master™
    in 3 levels PSM I, PSM II, and PSM III
  - Professional Scrum Product Owner™
    in 3 levels PSPO I, PSPO II, and PSPO III
  - Professional Scrum Developer™
  - … and more

Contents

Summary

- Agile methods and techniques are used by a large number of organizations
- Main reasons
  - Accelerate software delivery
  - Ability to change priorities
  - Increase productivity
- Scrum is the most used agile method

Summary

- XP practices could be used in other software development processes as well
  - "Test-First"-approach
  - Small releases and continuous / frequent integrations
  - Pair programming
  - Refactoring to keep “projects well”

Summary

- Scrum
  - Framework for lean software development
  - Main advantage: After every Sprint a usable product is released: The Increment
  - 3 main roles are defined:
    - Product Owner
    - Scrum Master
    - Development Team (7 +/- 2)
  - Requirements are managed in the Product Backlog

Summary

- Scrum
  - Time box approach with Sprints
    - Planning – prioritization
    - Run – on team's own responsibility
    - Review – for product
    - Retrospective – for process
  - Daily Scrum
  - Scrum Master is responsible to remove impediments, addressed in daily scrum meetings and the retrospective