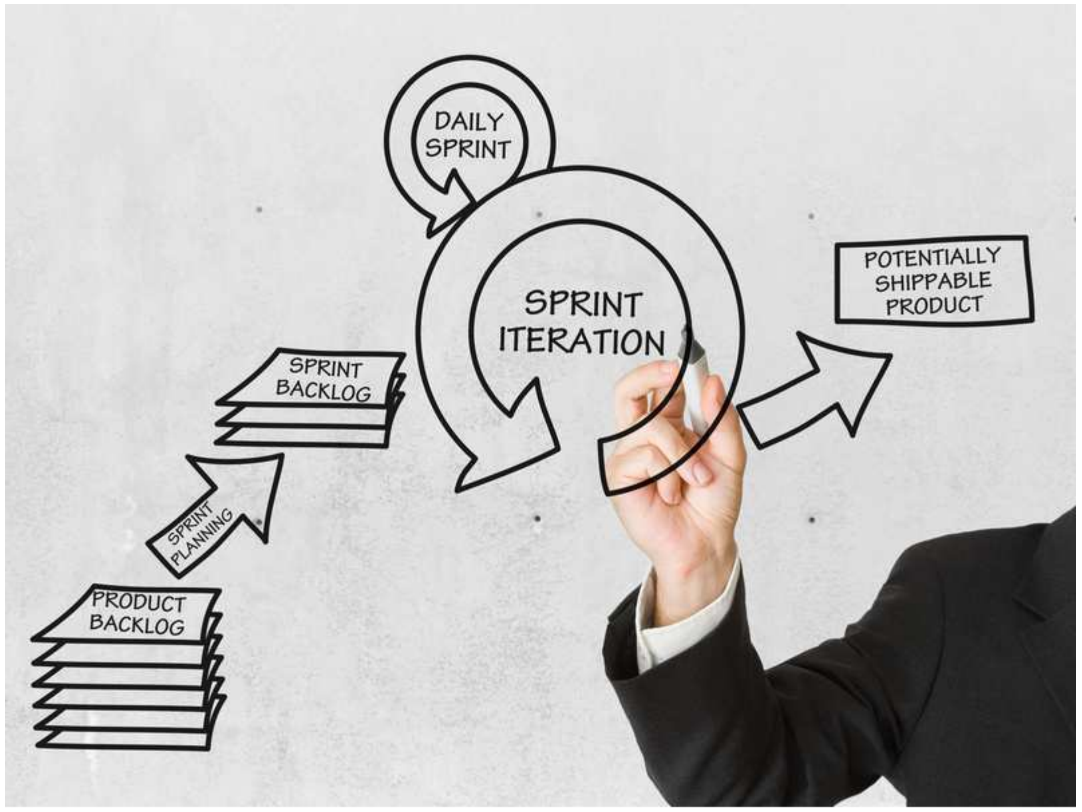
Agile development with

SCRUM



Shawn Hempel - Fotolia.com #67705572

Pictures and information from
Wikipedia : http://de.wikipedia.org/wiki/Scrum
Drach, Mathis: Scrum Compact http://improuv.com/sites/improuv/files/publikation/scrum-kompaktdownload_0.pdf
Jeff Sutherland, scrum.inc training slides

Prof. Dr. M. von Schwerin

Project Management – Why?

- High failure rate of software projects
- Software is a non-material product, only the developers can assess the status and quality of the product during the creation process

➢ Regular consultations within the team and with customer and management

➢ Reliable sub-goals (milestones)

➢ Time management

Common project management methods:

- Waterfall model
- V-Modell / Prince 2
- Rational Unified Process

…

- Agile Methods: Scrum and Kanban

Scrum - agile software project management

- Scrum has been known as an agile software process in software development since the mid-1990s
- After the document-centered and rigid processes of the 1980s (waterfall model, V-model), Scrum is based on the **idea of lean production** and value creation from the implicit knowledge of Japanese corporations (e.g. Toyota)
- Scrum is an **alternative to the command-and-control organization**. Scrum is based on highly qualified, interdisciplinary development teams that are given a clear objective but are self responsible for its implementation. This allows a team to unleash its creativity and performance and achieve better products.
- Scrum is now used in most national and international companies, but it is not the ideal solution for every project (certifications require different processes, hardware dependencies are difficult to map, inexperienced teams, etc.).

Agile manifesto

- ***Individuals and interactions are more important than processes and tools.***
  Although well-defined development processes and development tools are important, the qualifications of employees and efficient communication between them are more important.
- ***Functioning software is more important than comprehensive documentation***
  Well-written and detailed documentation can be helpful, but the real goal of development is the finished software.
- ***Cooperation with the customer is about more than just negotiating contracts***.
  Instead of sticking to originally formulated and now outdated service descriptions in contracts, the focus is instead on ongoing constructive and trusting coordination with the customer.
- ***Reacting to change is more than just following a plan.***
  In the course of a development project, many requirements and boundary conditions change, as does the understanding of the problem area. The team must be able to react quickly.

Guiding principle: The more you work according to plan, the more you get what you planned, but not what you need.

- Video https://www.youtube.com/watch?v=6qpcB82aUz4

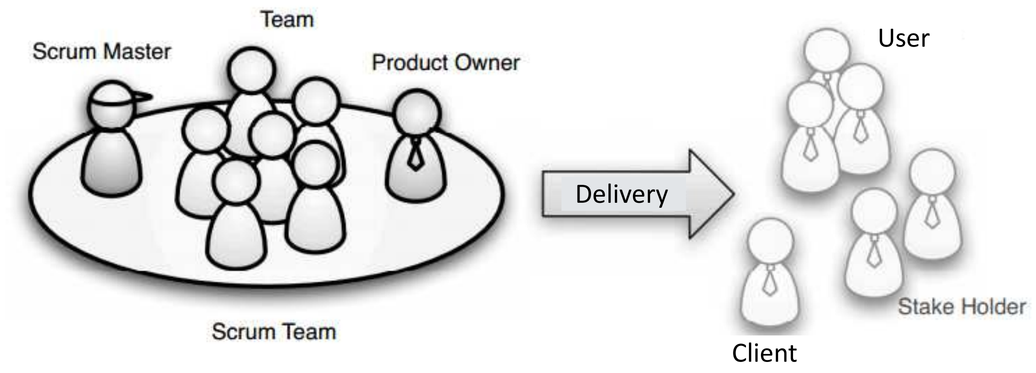
Scrum - Origin

Scrum is a term from rugby:
A team pursues a common goal.

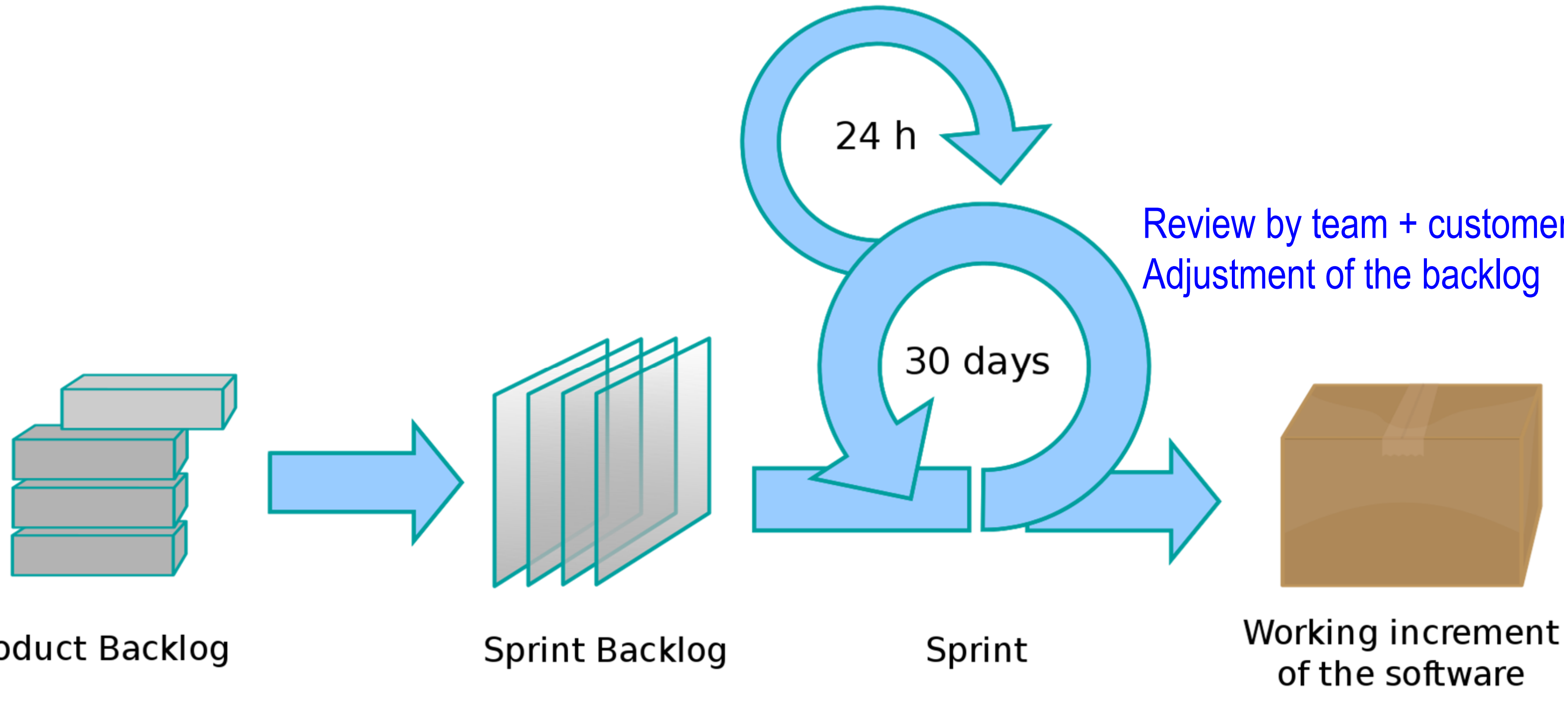
- Scrum is an Agile process to manage and control development work
- Scrum is a wrapper for existing practices
- Scrum is a self-organizing team-based approach to developing systems when requirements are changing rapidly

Scrum Principles

- **Self-organization**
  Few specialists work in a self-organized manner
- **Roles**
  Product owner with "product vision"
  Scrum Master defines Scrum rules,
  moderates meetings and
  ensures the team's working framework
- **Pull principle**
  Team itself controls tasks
- **Timeboxing**
  Goals and time frame must be given to the team, results are required within the time frames
- **Usable business functionality**
  At the end of each timebox, the team must deliver a deliverable that meets project targets



Structure of Scrum



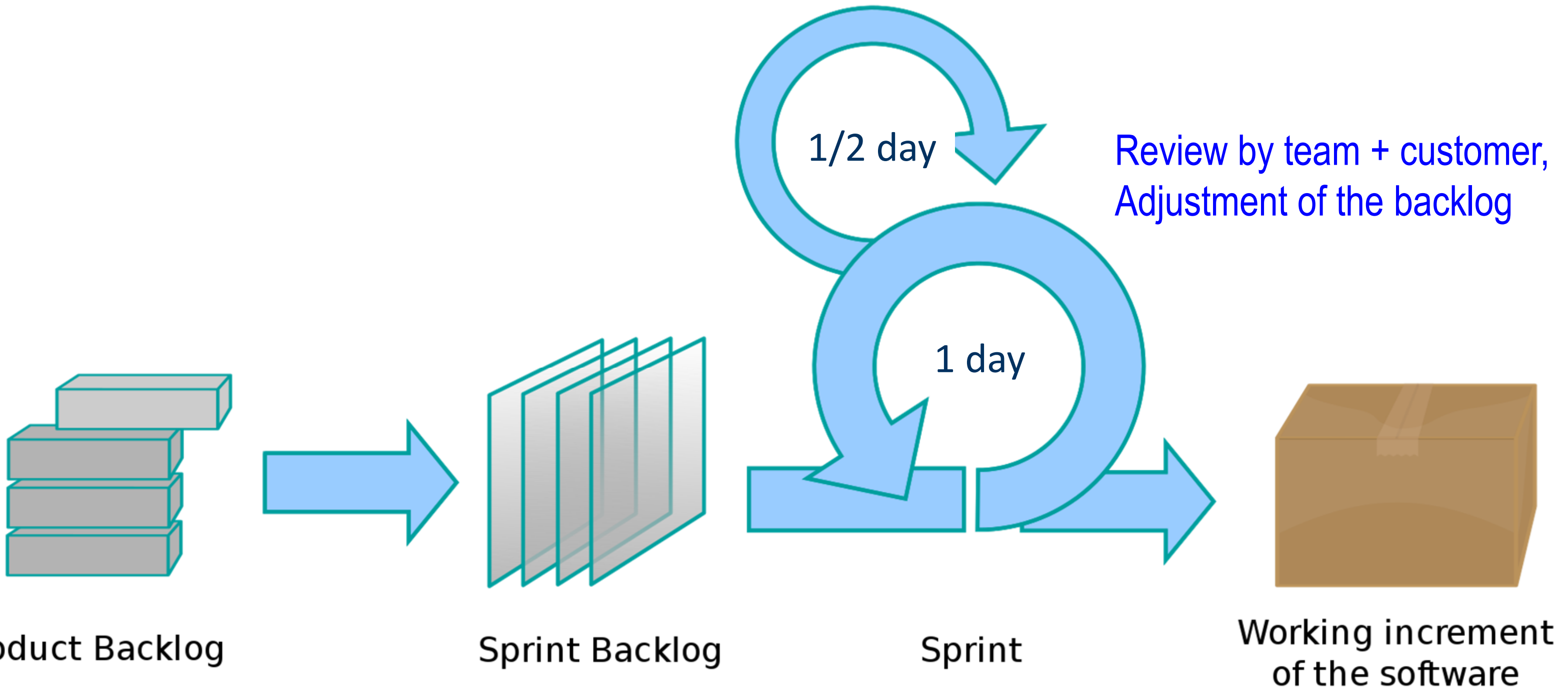
Review by team + customer,
Adjustment of the backlog

Backlog: User Stories
defined together with
the customers

Sprint: Implementation
of an iteration

[Wikipedia]

Customized Scrum for Capstone Project



Review by team + customer,
Adjustment of the backlog

Backlog: User Stories
defined together with
the customers

Sprint: Implementation
of an iteration

[Wikipedia]

Roles in Scrum

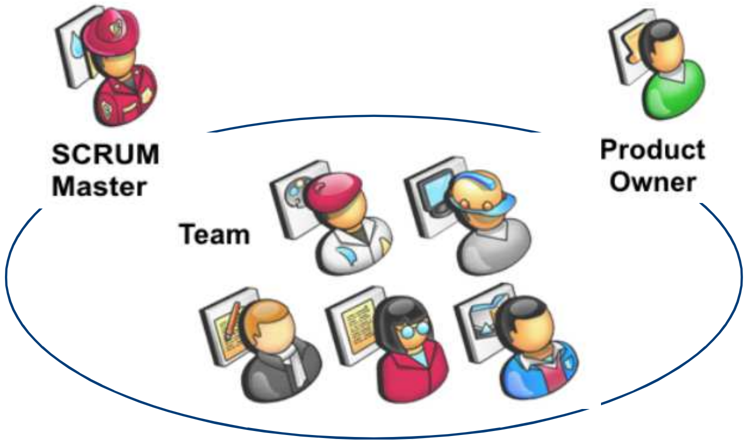
Scrum has a total of six **roles**

- **Internal roles in Scrum**: - Scrum Master
  - Product Owner
  - Development Team
- **External roles:** - Management
  - Customer / Client
  - User

Planning necessary for the

- Size of the **timeboxes**
- Size of the **backlogs** to be delivered and the
- **Speed of implementation** of the team
- Team estimates effort for units to be delivered itself (how many units per iteration can be processed?)

-> Experience required (Scrum Master can coordinate here)



Scrum Master

- ScrumMaster is **responsible for ensuring that Scrum succeeds**,
  - works together with the development team, but is not part of it
  - introduces the Scrum **rules** and ensures that they are adhered to
  - moderates the meetings
  - deals with any **disruption** to the Scrum process
- ScrumMaster is the leader of the development team
  - should not give work instructions to individual team members
  - has no disciplinary rights
  - Role is that of an "expedition leader"
- **Staffing**: Ideal case
  - ScrumMaster **is chosen by the development team**
  - Once Scrum is established, the ScrumMaster can also take on other roles

Product Owner

- The Product Owner is responsible for **strategic product development**
  - Conception and communication of a clear product vision
  - **Definition and prioritization** of the respective **product** features to be developed
  - Decision on delivery date, functionality and costs.
- Definition of product features in the product backlog with
  User stories (functionalities from the user's perspective)
  and prioritization of the user stories
- **Consultation with the customer**
  - The primary goal is the cost-effectiveness of the product
  - Bundling the interests and requirements of the interest groups

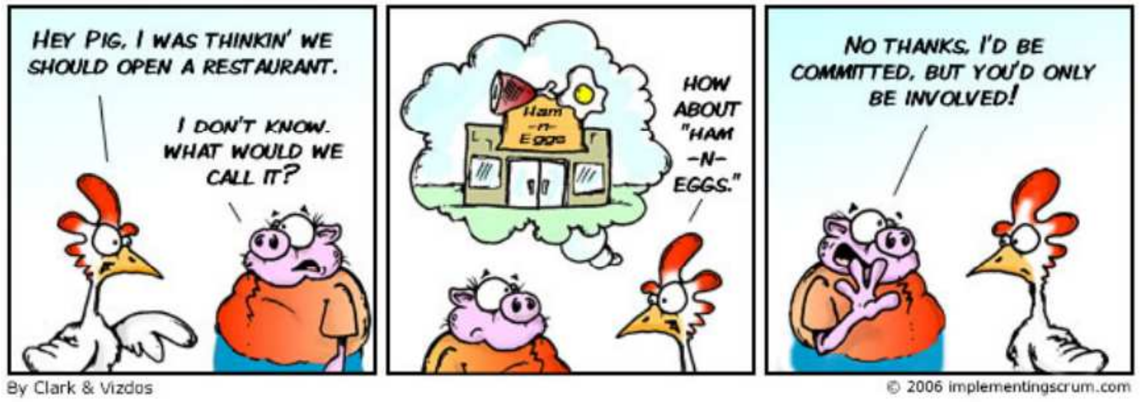
Development Team

- Responsible for the **delivery of product functionalities**
  Responsibility for **compliance with** the agreed **quality standards**
  Autonomy in the decision on sprint deliveries
- **Interdisciplinary** composition of the team
  Ideal team member is both a specialist and a generalist
  Team always acts as a unit
- A development team ideally consists of 7±2 people
- Further tasks of a development team are
  - Estimation of the scope of each user story,
  - Breaking down the user stories selected for a sprint into **tasks**
- In the implementation phase, team members are **developers, testers, planners ...**

Video
https://www.youtube.com/watch?v=u6XAPnuFjJc

Customer

- **Customer refers** to the **client** to whom the product is made available after completion.
- Depending on the situation, the customer can be either the internal specialist department or an external customer.
- Product owners and customers should be in close contact for the duration of the project.
  The customer should already have the opportunity to provide feedback after the first sprints.


HEY PIG, I WAS THINKIN' WE SHOULD OPEN A RESTAURANT.
I DON'T KNOW. WHAT WOULD WE CALL IT?
Ham -n- Eggs
HOW ABOUT "HAM -N- EGGS."
NO THANKS. I'D BE COMMITTED, BUT YOU'D ONLY BE INVOLVED!
By Clark & Vizdos
© 2006 implementingscrum.com

Customer, Users and Management

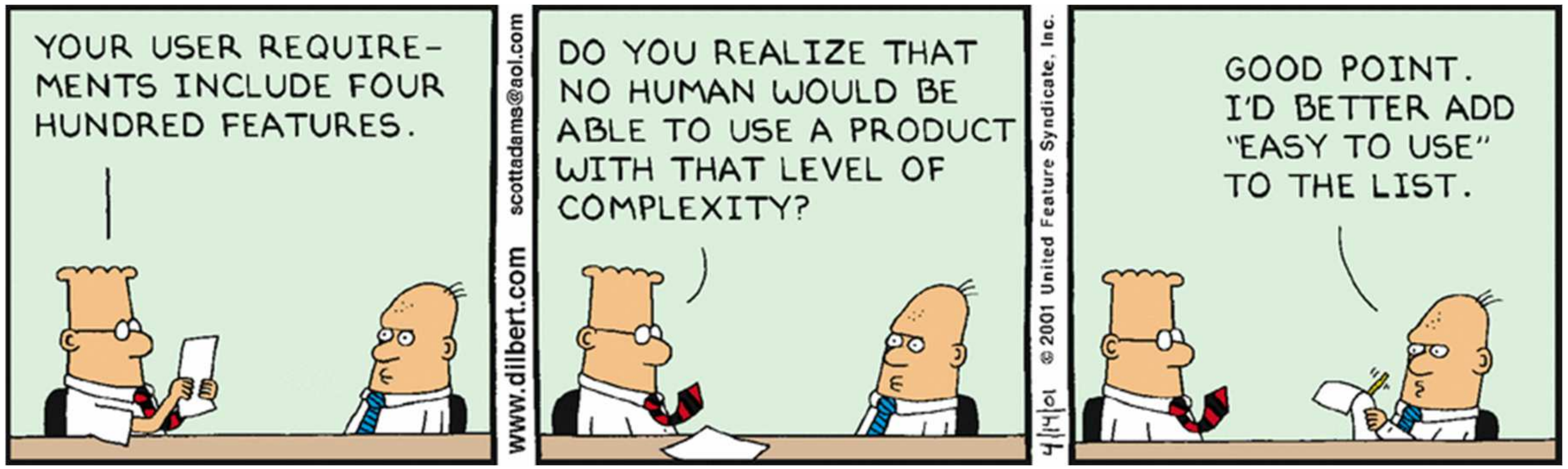
- **Customer:** Client, who ordered the product and whom it is made available after completion.
- **Users:**
  - Persons who use the product
  - User can, but does not have to be the customer at the same time
  - User/customer should be consulted during Sprint Planning 1 and Sprint Review for be consulted for feedback
- **Management:**
  - Is responsible for ensuring that there are working ***framework conditions*** for a successful Scrum
  - Provision of ***material resources*** (premises, work equipment)
  - ***General support*** for the chosen course (protection of the Scrum team from external work requirements, adequate staffing, support for Scrum masters).

Requirements definition

The starting point for development is an idea of what the product should achieve, usually formulated unspecifically as an **epic** or **feature**:

e.g. "The application shows how fit you are"
"Fitness data is recorded during operation"

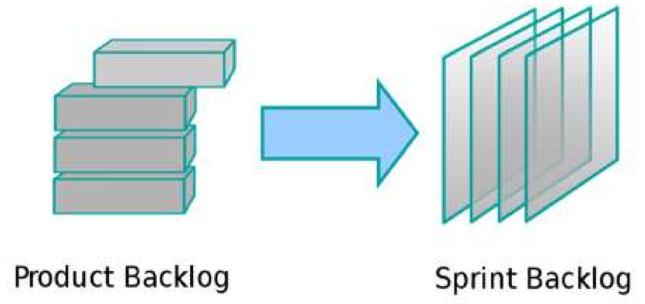
Epics are recorded informally and then formalized in the product backlog.


YOUR USER REQUIRE-MENTS INCLUDE FOUR HUNDRED FEATURES.
DO YOU REALIZE THAT NO HUMAN WOULD BE ABLE TO USE A PRODUCT WITH THAT LEVEL OF COMPLEXITY?
GOOD POINT. I'D BETTER ADD "EASY TO USE" TO THE LIST.
www.dilbert.com scottadams@aol.com
4/14/01 © 2001 United Feature Syndicate, Inc.

Product Backlog

Product backlog with product backlog items:

- List containing everything that should be included in the final product
- At the beginning the backlog contains the known and best understood requirements
- The requirements are not technical but user-oriented, so-called user stories
- Entries are ***prioritized*** according to aspects such as economic benefit, risk and importance
- Product backlog may change during the development process
- Items the highest prioritization are ***implemented*** first



Backlog: User Stories defined together with the customers

User Stories

- **Form**:
  *As a user (**Who?**), I want this functionality (**What?)** so that I have the following benefit (**What for?**).*
- Example: As a user, I can view the status of my mail item so that I can be at home for delivery.
- Product owner and team clarify ***the exact meaning*** of a user story in ***Sprint Planning 1*** and whether it should be broken down into smaller units (tasks)
- Each story is verified by at least one **user acceptance test (UAT)** (at least one test case per user story)
- Questions about the ***"how"***, i.e. the technical implementation of this user story, belong in ***Sprint Planning 2***; information about this does not go into the product backlog, but is recorded on the task board with the help of the individual tasks.

User stories

Epic

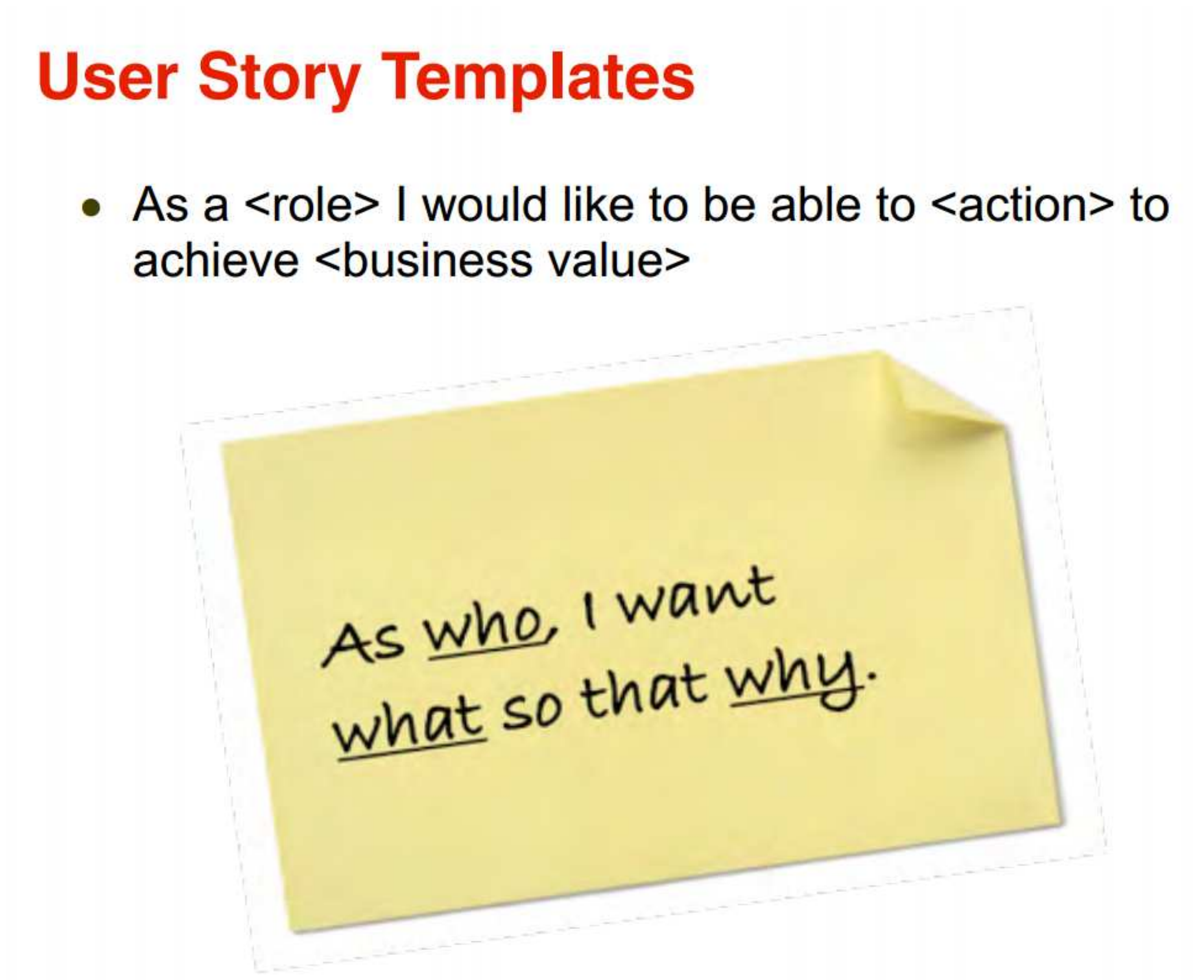
I am a frequent flyer and appreciate the service of a personalized booking process.

As a frequent flyer, I would like to book a trip and use the miles I have earned to save money.

As a frequent flyer, I simply want to book a flight that I take more often so that I save time.

As a premium frequent flyer, I would like to use available upgrades to fly more comfortably.

Product Backlog



User Story Templates

- As a <role> I would like to be able to <action> to achieve <business value>

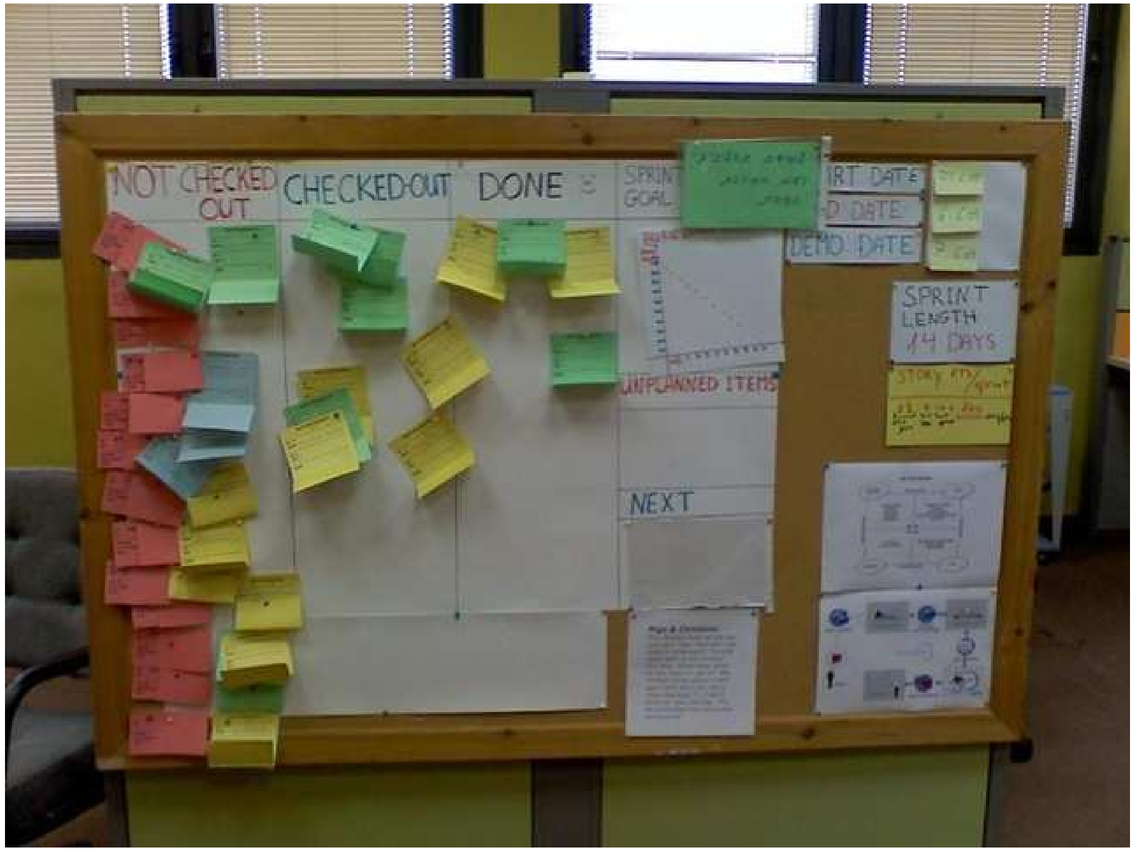
As who, I want what so that why.

[9a]

Management tools: Make Work Visible

Product backlog with **story cards**
Scrum board

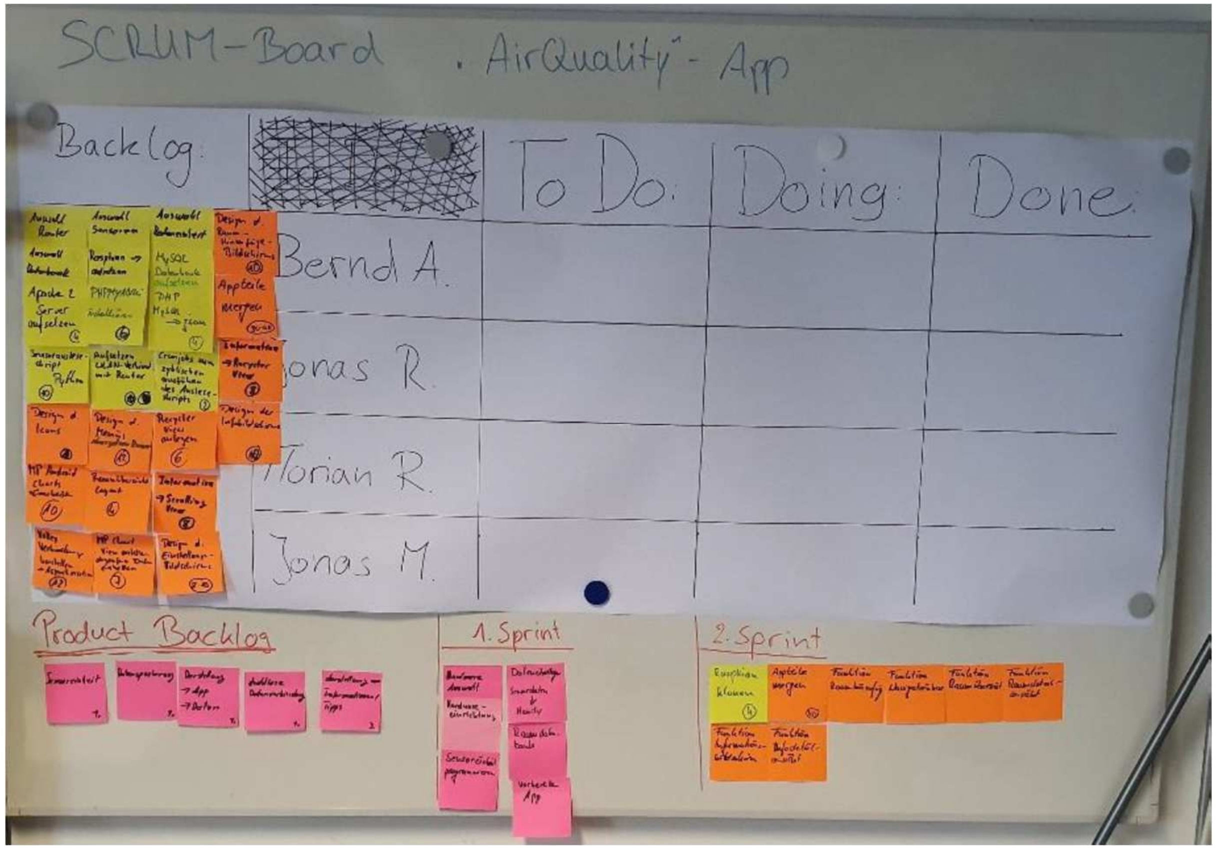
- User stories on index cards
- Product Owner explains individual backlog item and discusses it with the team
- Confirmation on the back
(User Acceptance Test)



Management tools: Make Work Visible

Product backlog with **story cards**
Scrum board

- User stories on index cards
- Product Owner explains individual backlog item and discusses it with the team
- Confirmation on the back
  (User Acceptance Test)



THU Technische Hochschule Ulm

Merge of two airlines

IN THE BEGINNING: BRIDGING

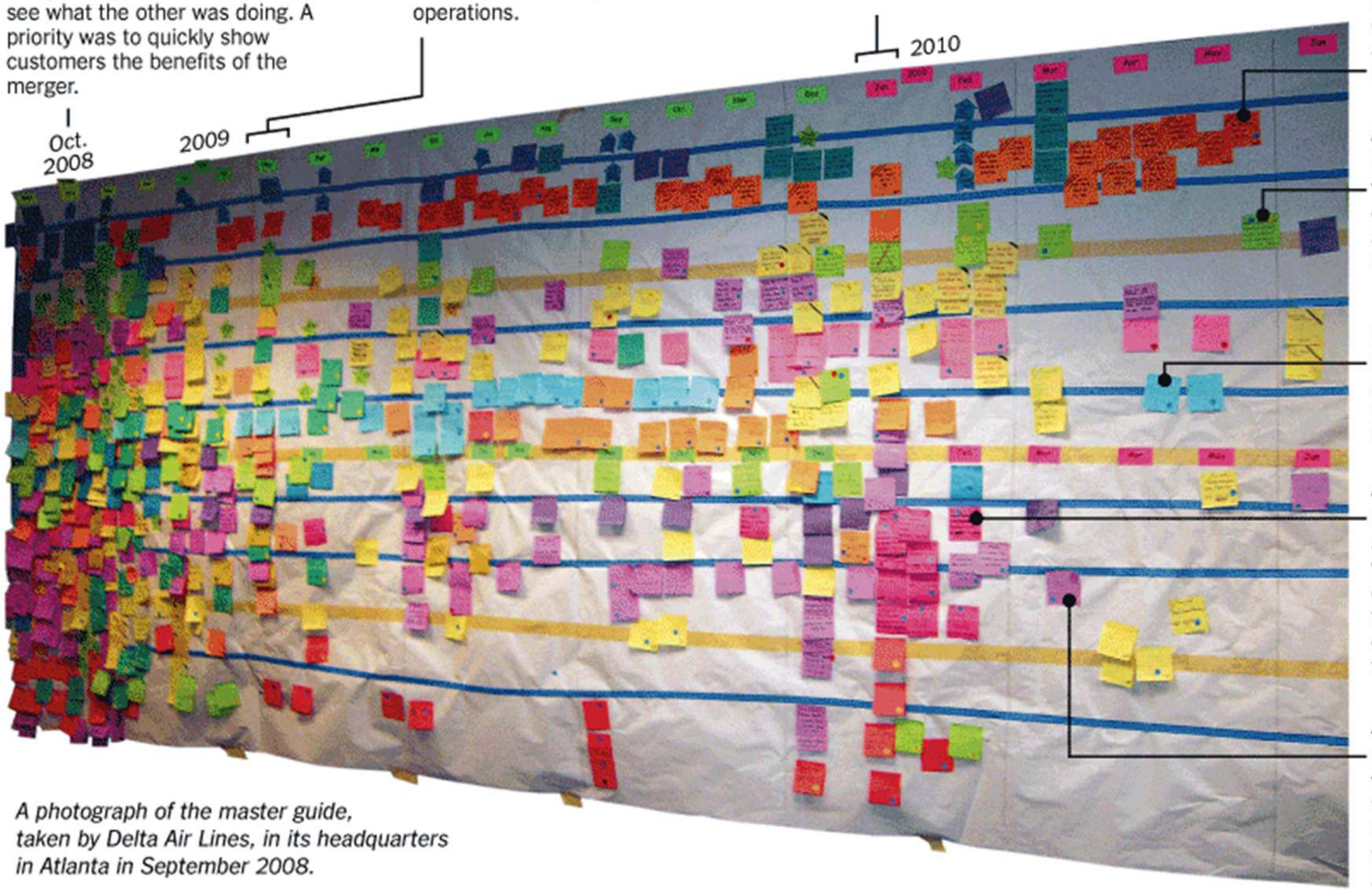
In the months around October 2008, when the merger was closed, there was a flurry of "bridging" projects: opening up access between the airlines' computer systems so each could see what the other was doing. A priority was to quickly show customers the benefits of the merger.

THE NEXT STEP

About five months after the merger, the two airlines began "cross-fleeting," when critical systems like reservations had to start talking to each other. But they remain separate operations.

ONE AIRLINE, ONE BRAND

Delta received final government approval to operate as a single airline in January 2010. At that point, all the computer systems could be switched to unified platforms. Many, like reservations and seat availability and pricing, had to be switched over at the same time.



AIRPORTS AND GATES

Orange notes indicate changes in customer service at airport counters and kiosks.

ALLIANCE PARTNERS

Bright green notes were for updates in coordinating with the airlines' partners, like Air France-KLM.

LOYALTY PROGRAMS

Light blue notes show steps in integrating customer loyalty programs.

AIRPORT OPERATIONS

Pink notes represent airlines' interaction with the airports — coordinating gates, flights and communications with the control tower.

AIRCRAFT CONTROL

Light purple notes were for changes in the systems that keep track of where flights are, rerouting and cancellations.

A photograph of the master guide, taken by Delta Air Lines, in its headquarters in Atlanta in September 2008.

https://archive.nytimes.com/www.nytimes.com/interactive/2011/05/18/business/delta-northwest-merger-graphic.html

Product backlog: Estimations

- Estimates are made for sprint planning
- Usually there are two estimations for every user story: priority and effort (we only consider the effort estimation here)
- The **estimation criterion** usually is complexity (nevertheless, **time** is often used)
- Estimates should be relative to a known quantity: storypoint / **hours**
- Estimates are made by the people who do the work (**team**), not by those who wait for the work result
- Estimates should be made again and again and improved if necessary
- Estimated variables: **complexity / effort**

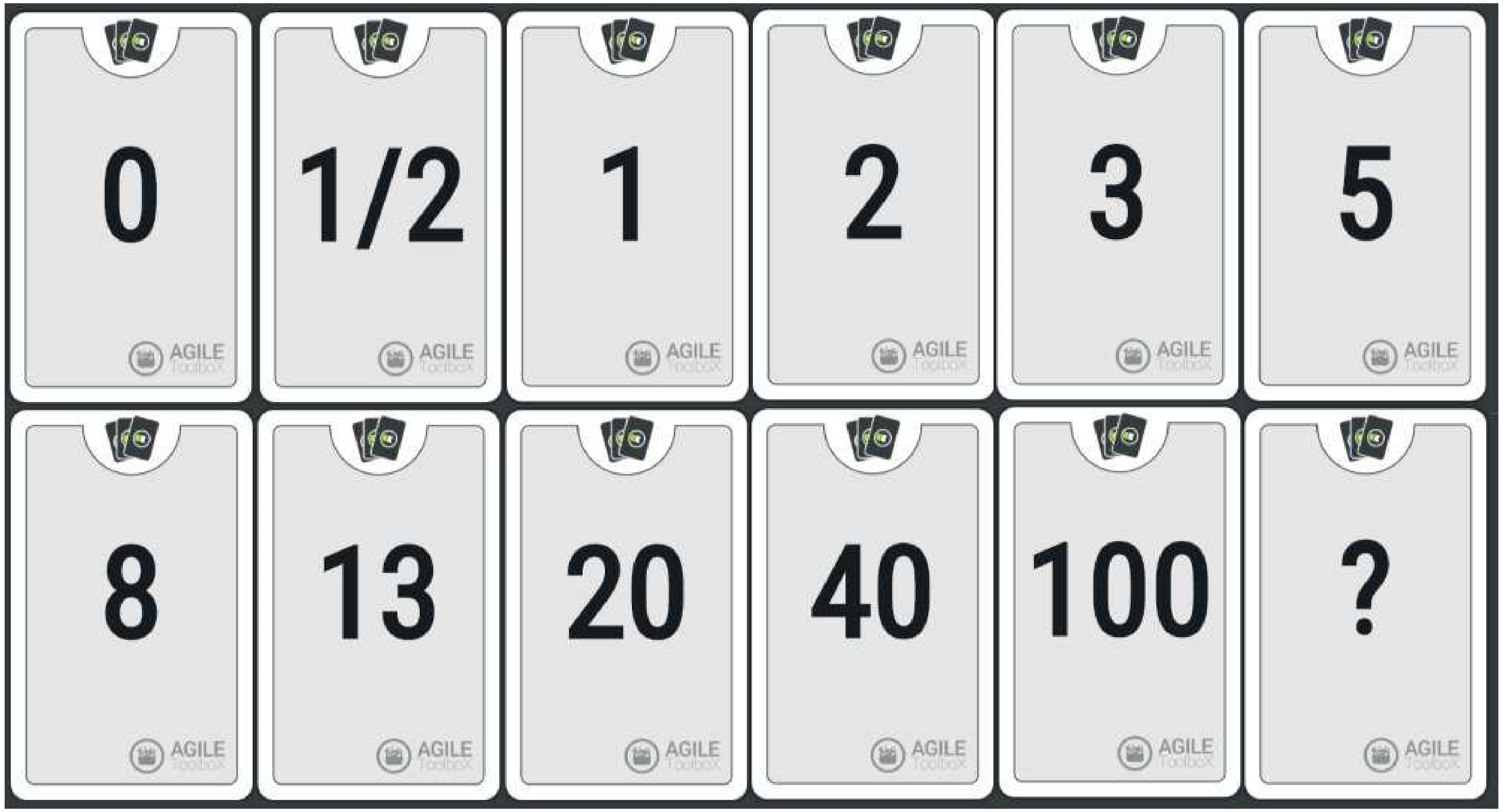
Methods for effort estimation

Estimation procedure for complexity / effort

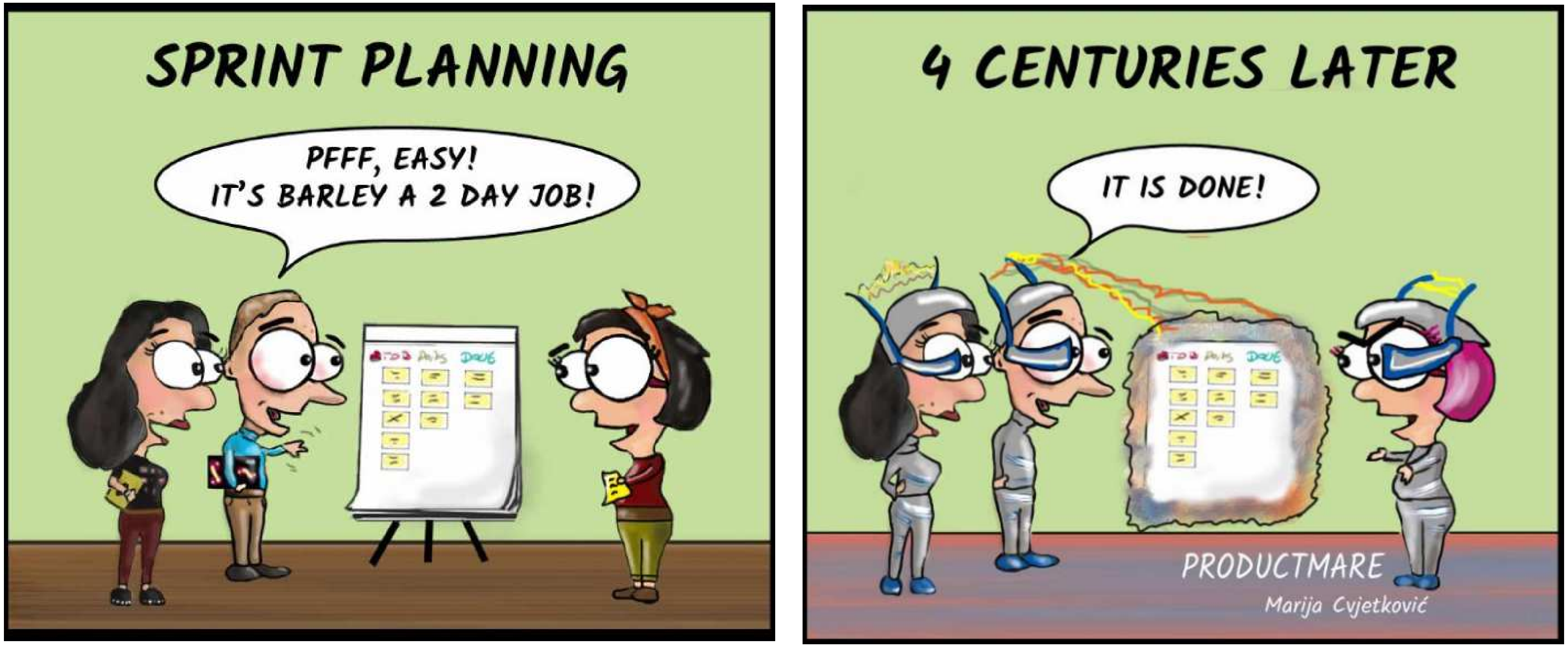
- **1000 ping-pong balls** are distributed according to importance
- **Kano** categorizes into basic functionalities and special features: Excitors, Linear, Indifferent, Questionable, Reverse Attributes
- **Relative weight** estimates relative benefits + relative penalty = BV in story points, Consideration of risk assessment results in priority number
- **T-shirt sizes:** S, M, L, XL
- **Planning Poker** with a set of Fibonacci playing cards (0,1,2,3,5,8,13,...) for all team members, play per backlog item, explain extreme values, play again, averaging

You will not have the time to estimate with Planning Poker, but it is important that the estimates are discussed and determined together as a team. This way, various eventualities are taken into account.

Planning Poker Cards



Estimation, why?



Ready and Done

- The objective of delivering finished software after each sprint led to the expansion of the word **"finished"** in Scrum, namely **"ready" and "done".**
- **Ready refers** to the **requirement**: a feature must be fully clarified (i.e. prioritized, estimated and understood) so that the team can commit to its implementation. Only when it is "ready" in this sense can it be included in a sprint.
- **Done refers** to the **sub-product**. An implementation is "done" when it meets the agreed quality criteria (definition of done) and when it has been presented and accepted in the **sprint review.**
- Software must function sustainably and verifiably, be tested and have an automated test suite so that it can be marked "Done"

Sprint

- Product development
- Responsibility solely with the team
- **Sprint Goal:**
  S pecific: defined goal
  M easurable: criteria for the measurability of goal achievement
  A ttractive: demanding, challenging goal
  R ealistic: achievable goal
  T erminated: Fixed end date for the goal
- **Duration**: 2-4 weeks / 1 day in capstone project

Sprint

Sprint Planning Meeting (Customized Scrum)

- Presentation of the **product backlog (user stories)**
  Joint clarification of the requirements
  Criteria for acceptance of every feature (validated on Review Meeting)
  User often participates
- **Estimation** of the effort for every user story
- Adoption of user stories by the team – **selection of the stories to be implemented** in the sprint (To Do) "What"
- Clarification of the technical implementation of the user stories selected for the sprint "How"

Sprint (Customized Scrum)

Initial Meeting – once at the beginning of project

- Elaboration of the **product backlog (user stories)**
- Joint clarification of the requirements behind the user stories
- **Criteria for acceptance** of every feature (validated on Review Meeting)
- **Rough estimation** of the effort for every user story

Sprint Planning Meeting

- Selection of user stories for the next sprint “What”
- Discussion of the details behind the User Stories “How“
- Re-estimation of the effort
- Check whether the quantity of user stories can be achieved by the team (velocity)

Sprint

Sprint Planning Meeting 2

- Sprint Planning 1: "**What**"
  Sprint Planning 2: "**How**"
- Clarification of the technical implementation of the user stories selected for the sprint (team organizes this independently)
- The results of the meeting are tasks, in this case **"tasks"**
  max. one day per task
- Tasks are posted on the pinboard -> Taskboard
  Recognizable here:
  - what is to be processed
  - what is in progress
  - what is finished
- Duration: 60 minutes per sprint week.
- Sprint Planning 1 and Sprint Planning 2 should be carried out on the same day.

Daily Scrum

Daily Scrum – team internal meeting

- Usually at the beginning of each working day
    - Purpose: Exchange information
    - Capstone Project: Half Daily
    - Short documentation of meeting (scrum master)
- Duration 15 minutes (not longer)
- Purpose: information exchange, overview
- Each team member reports
- If there is a need for discussion, a discussion is arranged outside the meeting

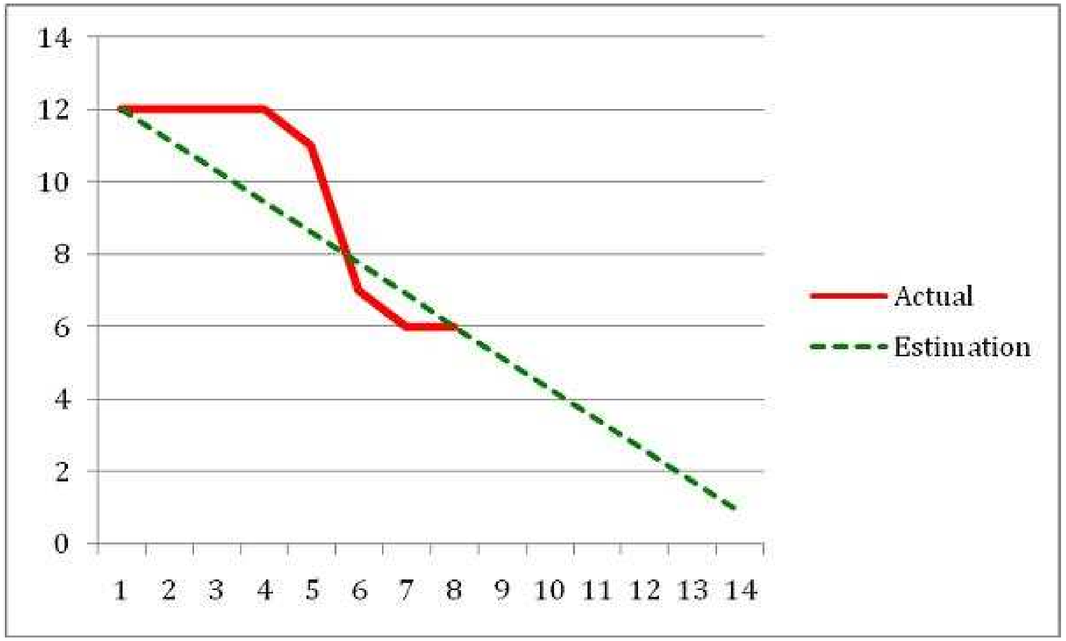
Sprint Review (with Retrospective)

Sprint Review with client/user

- At the end of the sprint
- Presentation and review of results (Product Owner with Team)
- Review (Product Owner)
- No compromises
- Unfinished features move to the next sprint
- Participation of the client/user
- Results may lead to new functions being added to the product backlog

Burndown Charts

- **Sprint burndown chart:**
  Visualization of work already completed and remaining work.
- x-axis: Time course (in days)
  y-axis: number of tasks not completed
- **Story burndown chart:**
  Visualization of the status of the selected user stories
- Contains not yet completed user stories of a sprint, possibly with specification of the story points
- x-axis: Duration of a sprint
  y-axis: Sum of the remaining remaining story points,
  Staircase-shaped curve



Customized Scrum for AICOSS course in a nutshell

Initial meeting

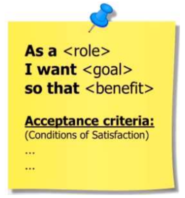
- Team building
  Assignment of roles   Scrum Master
                             Product Owner

- Discussion of the desired project result and formulation of the user stories   Product Backlog

- Effort estimation for user stories (in hours)

Sprint planning meeting (resp. PO)

- Selection of User Stories for Sprint (Scrum Board: To Do)
- Check effort (new estimate if necessary)


As a <role>
I want <goal>
so that <benefit>
Acceptance criteria:
(Conditions of Satisfaction)
...
...

Sprint

- Implementing of user stories (Scrum Board: In Progress)
- Implemented user stories (Scrum Board: Review)
- Half Daylies as Meetings to find out about the status

Sprint review meeting with customer representatives (resp. PO)

- Acceptance of increments (user stories) (Scrum Board: Done)