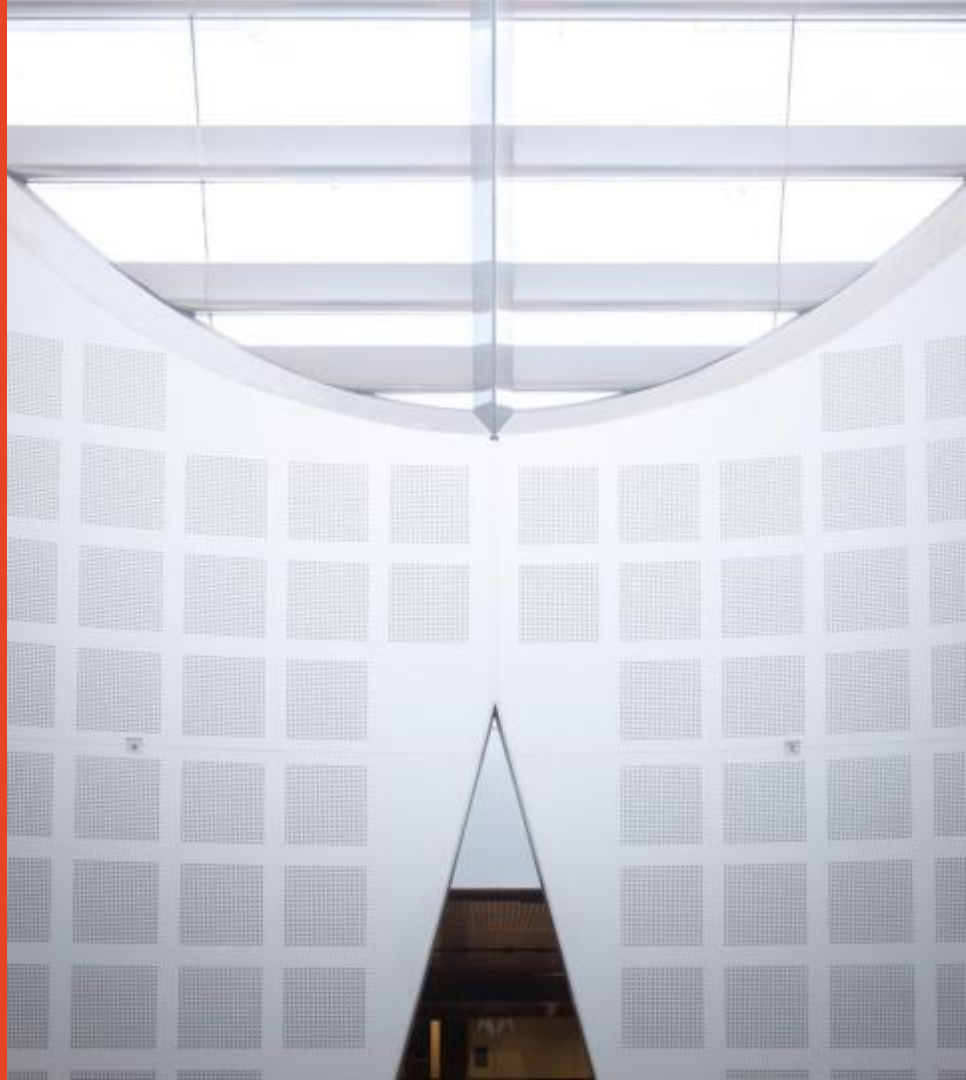
Agile Software Development Practices SOFT2412/COMP9412

Agile Methods - Scrum

Xinyi Sheng
School of Computer Science

Slides based on Dr. Basem Suleiman and Dr. Farshid Hajati



Agenda

- Agile Manifesto
  - Values and principles
- Agile Methods

- Scrum
  - Definition and Values
  - Teams and Roles
  - Events
  - Artifacts

Recall – Agile Process …

- Agile advocates believe:
  - Traditional software development processes are too heavy-weight
    - Too many things are done that are not directly related to software product being produced
  - Traditional software development is too rigid
    - Difficulty with incomplete or changing requirements
    - Short development cycles (Internet applications)
  - More active customer involvement needed

Recall – Agile Process …

- Agile process are considered
  - **Light-weight**
  - **People-based** rather than Plan-based
- Several agile methods
  - No single agile method
  - Extreme Programming (XP), Scrum
- No single definition
- Agile Manifesto closest to a definition
  - Set of principles
  - Developed by Agile Alliance

Agile Manifesto - Revisit

- "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

- The items on the left are more valued than those at the right

Agile Manifesto: http://agilemanifesto.org/

Agile Principles

1. Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable software

2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage

3. **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the **shorter timescale**

4. **Businesspeople** and **developers** must **work together daily** throughout the project

Agile Alliance: http://www.agilealliance.org

Agile Principles

5. Build projects around **motivated individuals**. Give them the **environment and support** they need, and **trust** them to get the job done

6. The most efficient and effective method of **conveying information** to and within a development team is **face-to-face conversation**

7. **Working software** is the primary **measure of progress**

8. Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a **constant pace** indefinitely

Agile Alliance: http://www.agilealliance.org

Agile Principles

9. Continuous attention to technical excellence and good design enhances agility

10. Simplicity - the art of maximizing the amount of work not done is essential

11. The best architectures, requirements, and designs emerge from **self- organizing teams**

12. At regular intervals, the **team reflects** on how to become more **effective**, then **tunes and adjusts its behavior** accordingly

Agile Alliance: http://www.agilealliance.org

Agile Methods

- All have shared goal; delivering valuable software iteratively
- Might have some different practices to achieve this goal
  - How coding should be done, how work is arranged daily, team structure and communication
- Examples of Agile methods:
  - Scrum ✓
  - Extreme Programming (XP) ✓
  - Lean software development (Toyota)
  - Kanban (lean development)

eXtreme Programming (XP)

- Development and delivery of **very small increments** of functionality
- **Relies on constant code improvement**, user involvement in the development team and pairwise programming
- Emphasizes **Test Driven Development (TDD)** as part of the small development iterations
- And many more

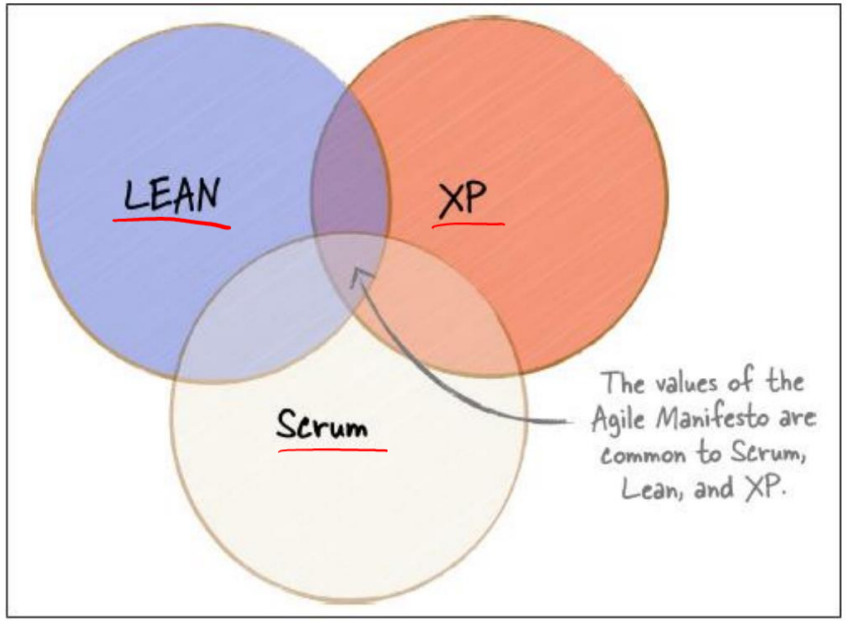
XP Practices

- Customer team member
- User stories
- Acceptance tests
- Pair programming
- Test-driven development (TDD)
- Short cycles
- Continuous integration
- Simple design
- Refactoring
- Collective ownership

Agile Methods

Agile methods; Scrum, XP and Lean

All have the agile values at their core and share some values, ideas and practices with each other



Scrum

- Originating in a broader “product development” lifecycle from H. Takeuchi and I. Nonaka (1986)
  - A new approach to increase speed and flexibility
- Used for software development by K. Schwaber, J. Sutherland and others
- Extensively used in various organizations (e.g., manufacturing, product development, software, hardware)

https://en.wikipedia.org/wiki/Scrum_%28software_development%29

Scrum (Software Development)

- An agile method for managing software development focused on delivering products of the highest value
- Focus on continuous improvement of the product, the team and the working environment
- Scrum is lightweight, simple to understand but **difficult to master**

https://en.wikipedia.org/wiki/Scrum_%28software_development%29

Scrum Theory

- Based on '*empirical process control*' theory
  - Knowledge comes from experience and making decisions based on knowns
  - Iterative, incremental approach to optimize predictability and control risks

- Pillars of empirical process control
  - 1 *Transparency*: significant aspects of the process must be visible to those responsible for the outcome
  - 2 *Inspection:* Scrum users must frequently inspect Scrum artifacts and progress toward an iteration goal to detect undesirable variances
  - 3 *Adaptation:* adjust the aspects of the process that lead to deviation outside the acceptable limits

Scrum Values

- The Scrum team members should learn and explore the following values as they work with the Scrum roles, events, and artifacts:
  - **Commitment**: personally commit to achieving the goals of the Scrum Team
  - **Courage**: members can do the right thing and work on tough problems
  - **Focus**: everyone focuses on the work of the iteration and the team's goals
  - **Openness**: the team and the stakeholders agree to be open about the work and the challenges with performing the work
  - **Respect**: Scrum team members respect each other to be capable, independent people

Scrum Practices

- Teams and their roles
  - Product Owner, Scrum Master, Dev Team
- Events
  - Sprint, Sprint Planning, Daily Scrum, Sprint Review, and Retrospective
- Artifacts
  - Product Backlog, Sprint Backlog, Increment

- Project estimation and Sprint estimation
- Rules govern the relationships between roles, events and artifacts
- All of the above components are essential for Scrum success

The Scrum Team

Scrum Team

- Team roles
  - Development Team
  - Product Owner (one person!)
  - Scrum Master (one person!)
- Scrum Team
  - Small enough to be agile
  - Cross-functional
  - Self-organizing
  - Deliver products iteratively and incrementally, maximizing opportunities for feedback

Scrum Team - The Product Owner

- **Product Owner** maximize the value of the product and the work
  - Talks to customer, understand requirements and its priorities
  - Managing the Product Backlog (only person)
    - Can assign it to the development team, but still accountable

- **Managing the product backlog:**
  - Record product backlog items and order it
  - Optimize the value of the work the development team performs
  - Ensure transparency and clarity of the product backlog
  - Ensure the development team understands product backlog

Scrum Team – The Development Team

- Professionals who do the work of delivering a potentially releasable product at the end of each iteration
- Creates the increment (only by dev team)
- **What's the optimal team size? Discuss**

Scrum Team - The Development Team

- Professionals who do the work of delivering a potentially releasable product at the end of each iteration
- Creates the increment (only by dev team)
- **What's the optimal team size? Discuss**
  - Small enough and large enough!
  - Two-Pizza team (4-9 members)
  - Less than 3; issues with productivity and interactions
  - More than 9; issues with coordination

Scrum Team - The Development Team

- Self-organizing: turns Product Backlog into increments of potentially releasable functionality
- Cross-functional: skills mix necessary to create a product
- No sub-teams: regardless of domains that need to be addressed
- Whole team is accountable
- No titles for Dev team members

Scrum Team - The Scrum Master

- Keeps the team focused on using Scrum properly ("*servant-leader*")
  - Everyone understands Scrum rules and values (coaching)
  - Remove impediments
  - Helps those outside the Scrum team which of their interactions with the Scrum team are/aren't helpful
  - Maximize the value created by the Scrum team through changing team interactions

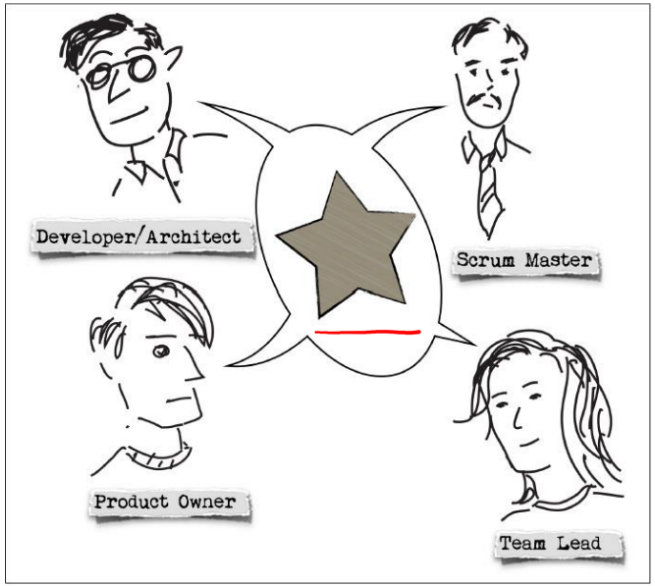
Scrum Team - The Scrum Master

- Serves the Product Owner;
  - Creating mutual understanding of goals, scope and product domain
  - Finding effective ways for managing the product backlog
  - Helping the Scrum team understands the need for clear and concise product backlog items;
  - Ensuring the product owner knows how to arrange the Product Backlog to maximize value;
  - Understanding and practicing agility
  - Facilitating Scrum events as requested or needed

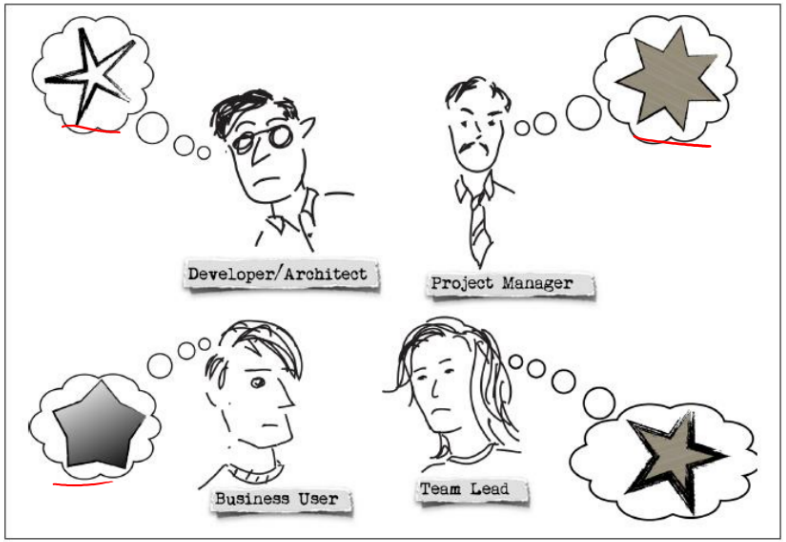
Scrum Team - The Scrum Master

- Serves the Dev team
  - To be self-organization and cross-functionality;
  - To create high-value products;
  - Removing impediments to the Dev team's progress;
  - Facilitating Scrum events as requested or needed; and,
  - Coaching the Dev team (in organizational environments in which Scrum is not yet fully adopted and understood)

Scrum Team



Team A



Team B

The Scrum Events

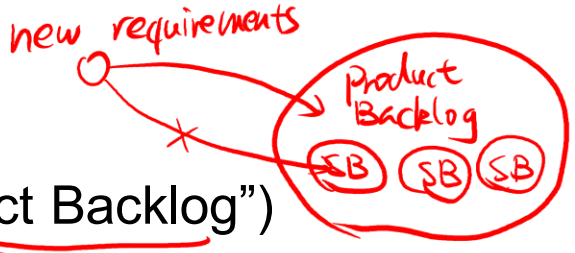
Scrum Events

- Used to create regularity and minimize the need for meetings
- All events are time-boxed (max. duration)
- Cannot be changed once an iteration (sprint) has started
- Designed to enable transparency and to provide a formal opportunity to inspect and adapt work

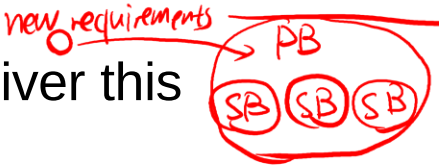
Scrum Events - The Sprint

- A development iteration (one cycle)
  - Useable and potentially releasable product increment is created
- Time-boxed (typically 2-4 weeks) no more than 1 month
  - Too long sprints may lead to changes in the definition
- Sprints have consistent durations during the product development
- Consists of the (1) Sprint Planning, (2) Daily Scrum, (3) the Development Work, the (4) Sprint Review and the (5) Sprint Retrospective

Scrum Events - Sprint Planning



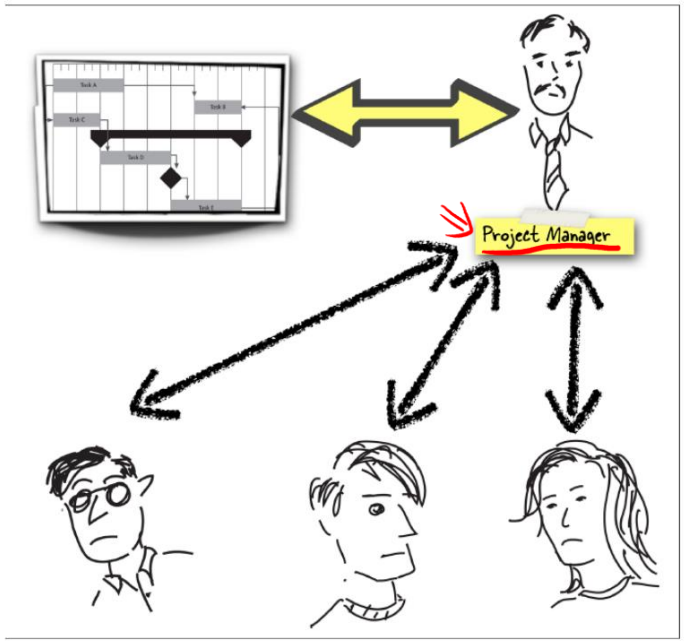
- Identify the Sprint Goal (items from the "Product Backlog")
- Identify work to be done to deliver this



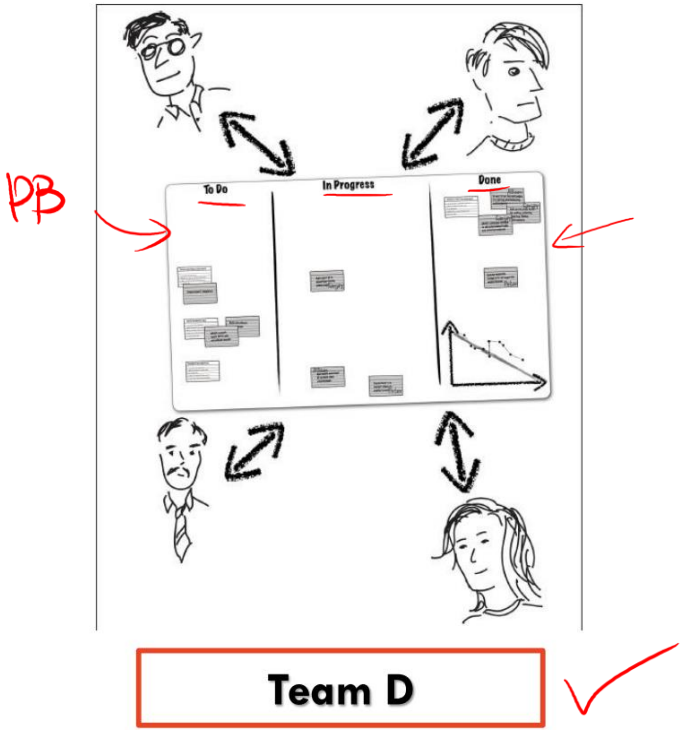
- Two-parts meeting (Scrum Master, Product Owner and Dev team)
  - **Before meeting**: Product Owner prepares prioritized list of most valuable items
  - **Meeting part 1** *(max. 4 hours)*: Product Owner & Dev team select items to be delivered at the end of the sprint based on their value and on the team's estimate of how much work needed
  - **Meeting part 2** *(max. 4 hours)*: Dev team (with the Product Owner's help) figure out the individual tasks they'll use to implement those items
- *Output*: Sprint Backlog (the items selected by the team for development)

Scrum - Team Structure

Which team organization better describes the Scrum Sprint/iteration planning?



Team C



Team D

Scrum Iteration Process

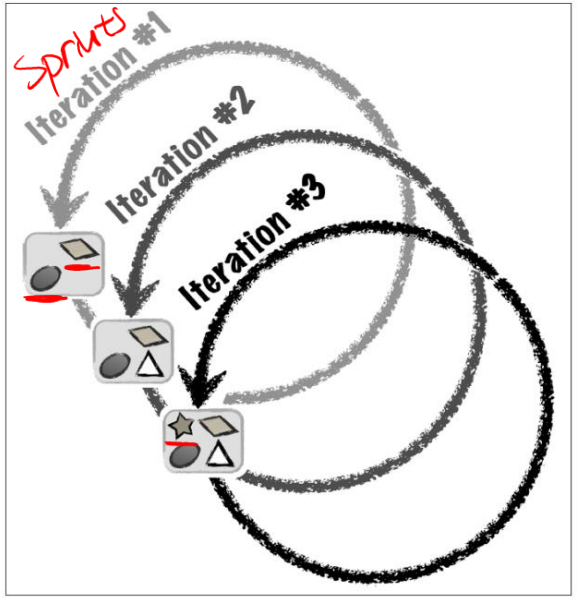
- Sprint (development iteration)
  - Timeboxed (typically 2–4 weeks – no more than one month)
  - Create a "*Done*" usable, potentially releasable product

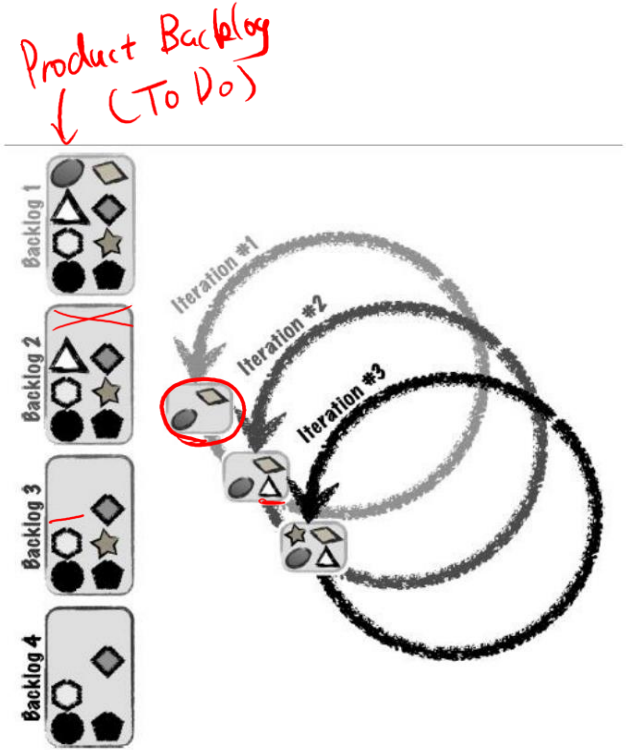
- A *Sprint (Scrum iteration)* contains a list of tasks and work product outputs that will be done in defined duration

  - At the beginning of the Sprint duration, each team member has a pretty good idea of what they will be working on

  - Management should not add new work product outputs to the Sprint – should be add to the Product Backlog instead

Scrum Iteration Process





Scrum Events - During Sprint

- **Daily Scrum meeting**
  - To ensure problems and obstacles are visible to the team
  - Timeboxed *15 minutes* (same time and place each day)
  - All team members including Scrum Master and Product Owner must attend
  - Interested *stakeholders* may attend as *observers*
  - Each briefly answers three questions:
    1. What did I do yesterday that helped the development team meet the Sprint Goal?
    2. What will I do today to help the development team meet the Sprint Goal?
    3. do I see any obstacles that prevent me or the Dev team to meeting the Sprint Goal?
  - No problem-solving during the meeting
    - Follow-up meetings if further discussion is required

Scrum Events - During The Sprint

- **Development Work; the Dev team**
  - Builds the items in the *Sprint Backlog* into working software
  - Should inform the Product Owner if they are overcommitted or can add extra items if time allows
  - Must update the Sprint backlog and keep it visible to everyone

Scrum Events - During The Sprint

The Sprint Review

- Informal meeting at end of the Sprint (max. 4-hours)
  - Dev team demonstrates working software to customers/stakeholders
    - Items actually done – completed, tested & accepted by the product owner
    - Only functional working software – not architecture, database design etc.
  - Stakeholders share their feedback, ideas, feelings, thoughts about the demo
  - The Product Owner explains “Done” and not “Done” items
- Output: revised Product Backlog and probable items for next iteration

Scrum Events - Retrospectives

– Opportunity for the Scrum team to inspect itself and create plan for improvements

– Purpose
  – Inspect how the last Sprint went with regards to people, relationships, process, and tools;
  – Identify and order the major items that went well and potential improvements;
  – Create a plan for implementing improvements to the way the Scrum Team does its work

Scrum Events - Retrospectives

Retrospective meetings (max. 1-2 hours)

- The Scrum Master and the Dev. team (maybe product owner)
- Each person answer two questions:
  - What went well during the Sprint?
  - What can be improved in the future?
- The Scrum Master notes improvements that should be added to the Product Backlog (non-functional items)
  - E.g., set-up a better build server, adopting design principles, changing office layout
- Output: identified improvements to be implemented in the next Sprint (adaptation)

Scrum Events - Retrospectives

The Retrospectives Prime Directive:

> ***Regardless of what we discover, we understand and truly believe that everyone did the best job they could, given what they knew at the time, their skills and abilities, the resources available, and the situation at hand.***

(From Norm Kerth 's book on Project Retrospectives
See also http://www.retrospectives.com)

- Why this rule?

Scrum Artifacts

Scrum Artifacts - Product Backlog

- Set of all features and sub-features (items) needed to build the product (the "Plan" for multiple iterations)
  - Features, functions, requirements, enhancements and fixes identified from previous Sprints
- Maintained by the Product Owner in collaboration with customers and team
- The source of the product requirements
  - Evolves over the time and never complete (dynamic)
- The items ordered by priority – value to the customer
  - To deliver some value to the customer in each iteration, put the most important things early

Scrum Artifacts - Product Backlog

- What does a Product Backlog look like?
  - Simple spreadsheet
  - Some items are "**customer features**"
    - Could be a user screen, an interaction scenario or use case, a new report/algorithm
  - Some items are **internal tasks** that contribute to the value of the product
    - *Can a design document be an item?*
    - *Can an early GUI prototype be an item?*
- Effort estimates – assigned to each item by the Dev team
  - *Should managers do the estimation of Product Backlog Items?* Yes / No

and the Dev Team

Scrum Artifacts - Sprint Backlog

– Set of Product Backlog items selected for the Sprint, and a plan for delivering the product increment and realize the Sprint Goal

– The Dev. team to forecasts next items to be implemented

– Includes at least one high-priority improvement identified from previous Sprint

– The Dev. team adds new work to the Sprint Backlog

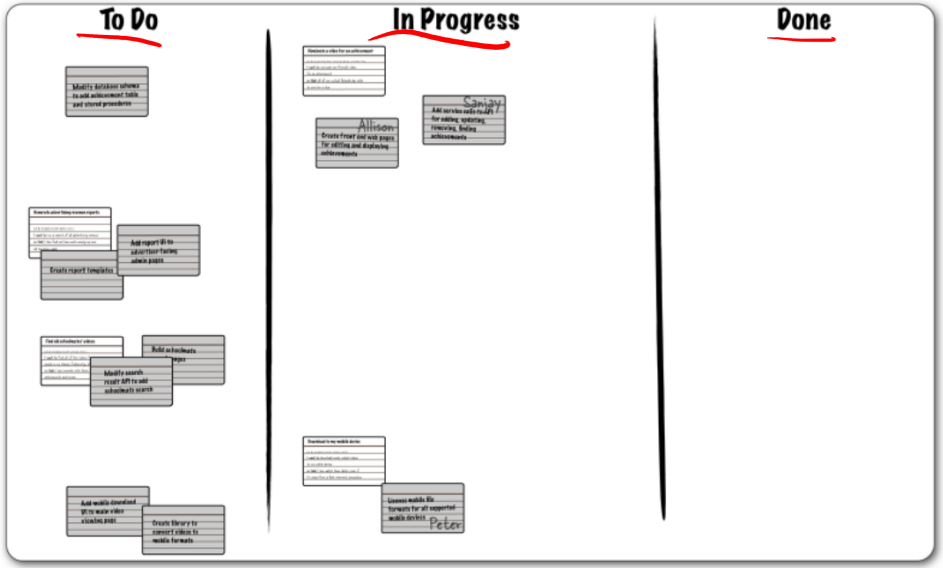
– The estimated remaining work is updated once an item is completed

– Visible to anyone and to be modified by the Dev. team

Sprint Backlog - Example

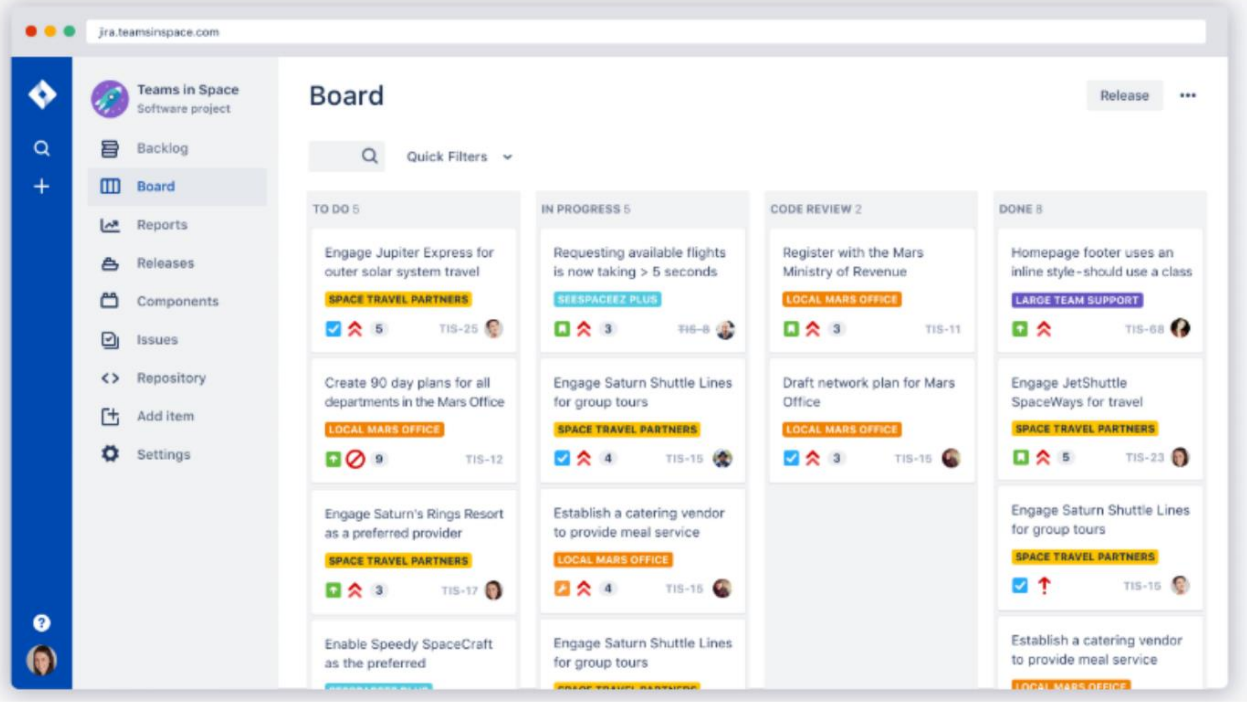
– Typically divided into 3 sections; To Do, In Progress, Done

– Tools to support Sprint planning and monitoring; e.g., Trello



Sprint Backlog - Example

–JIRA Agile - Scrum Board



Scrum Artifacts - The Increment

- Collection of the Product Backlog items that meet definition of “Done” by the end of the Sprint

- Definition of “Done”

  - Team’s shared agreement on the criteria that a Product Backlog item (user story) must meet before it is considered done

  - Work will not be counted toward the end of the Sprint if it does not meet the criteria

Scrum Estimation

Scrum - Progress Monitoring

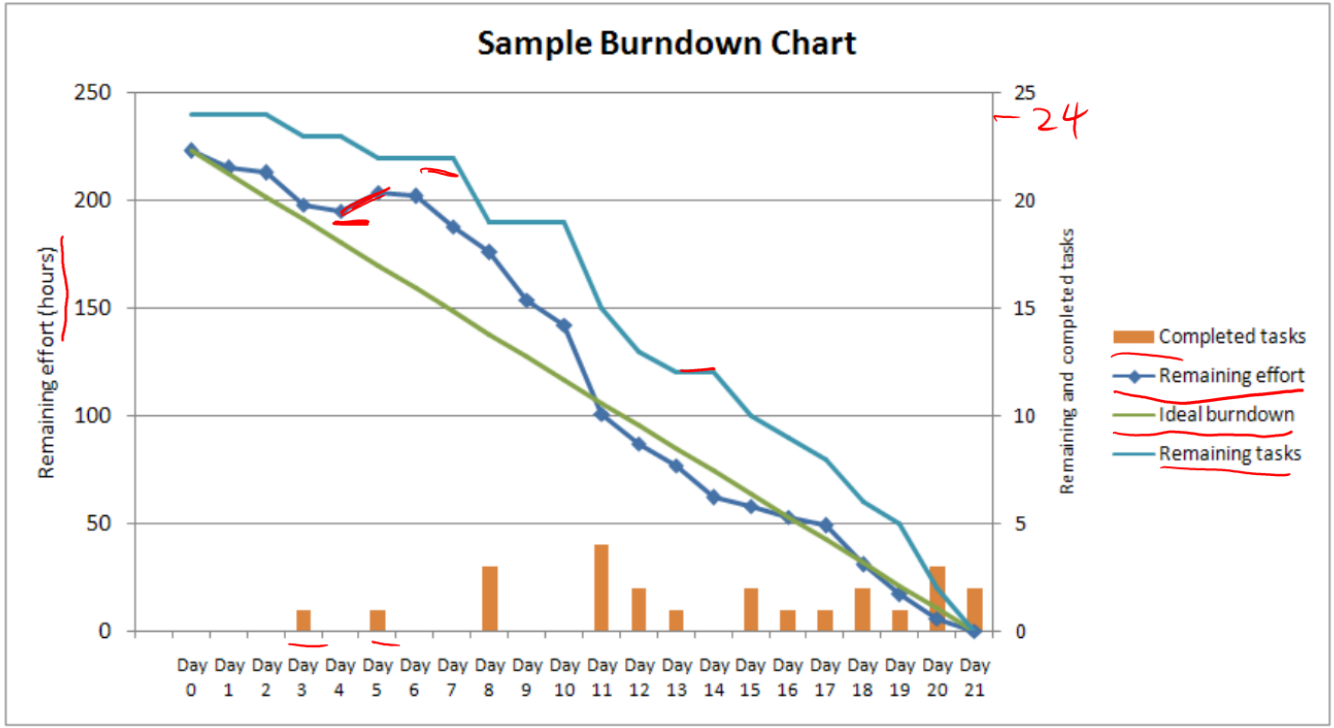
- Total work remaining to reach the goal – the product owner tracks this at least every Sprint Review
  - Compare with previous Sprints to assess progress toward projected work (transparent to all stakeholders)
  - Forecasting progress through burn-downs, burn-ups or cumulative flows
    - It's an estimate – there're some risks of unknowns

Scrum - Progress Monitoring

– "*Burndown" chart* tracks the amount of estimated effort remaining in a sprint
  – Maintained on a daily basis by the Scrum Master

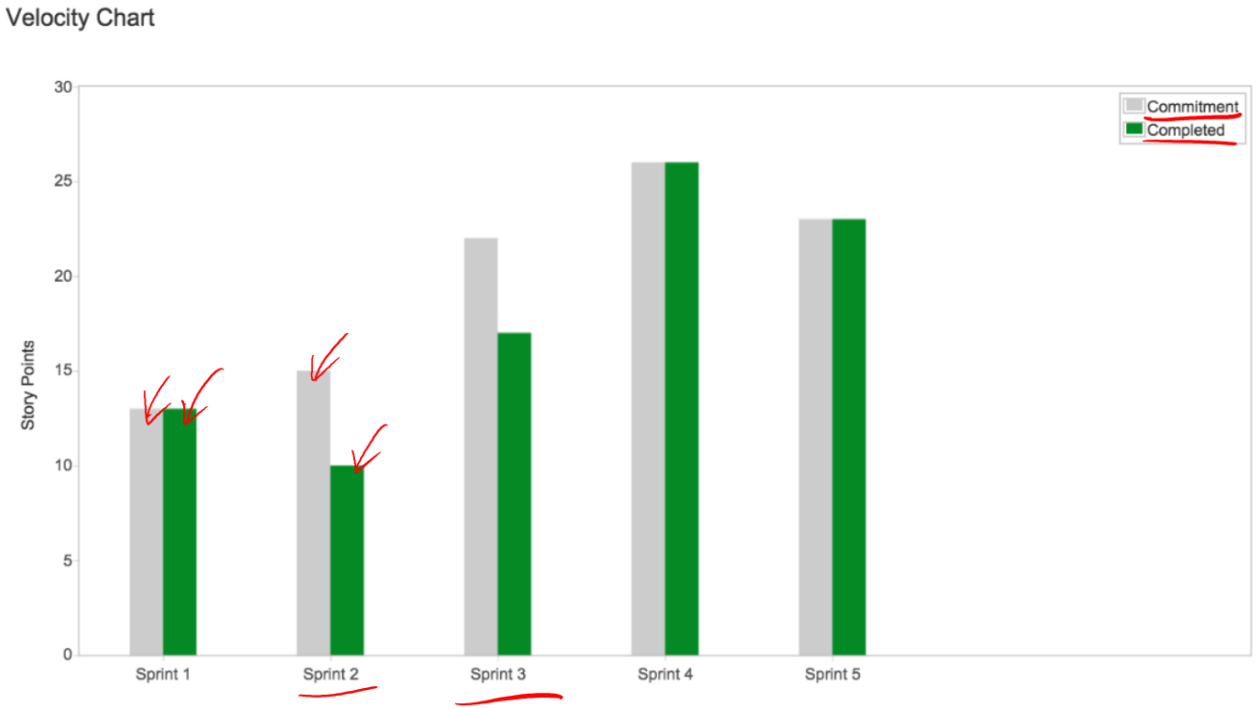
– *Velocity* tracks the amount of work from Sprint to Sprint
  – Velocity chart graphically shows project/team's velocity

Scrum - Burn-down Chart (Example)



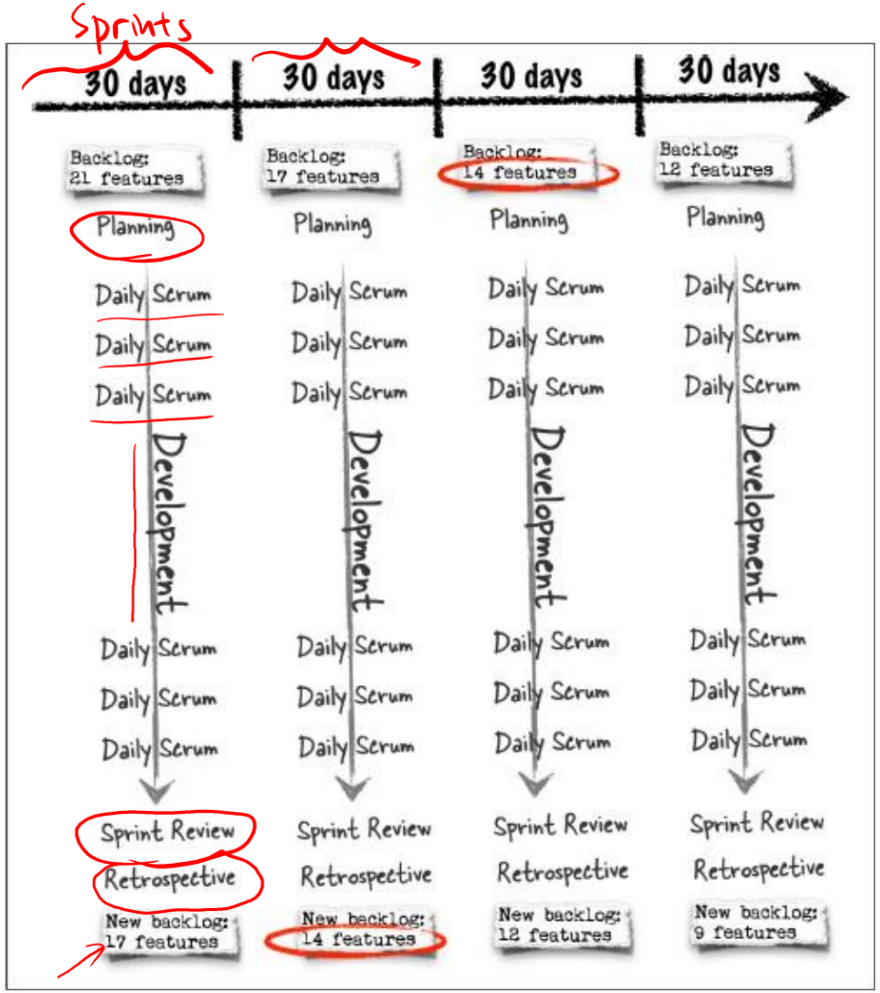
https://commons.wikimedia.org/wiki/File:SampleBurndownChart.png

Scrum - Velocity Chart (Example)

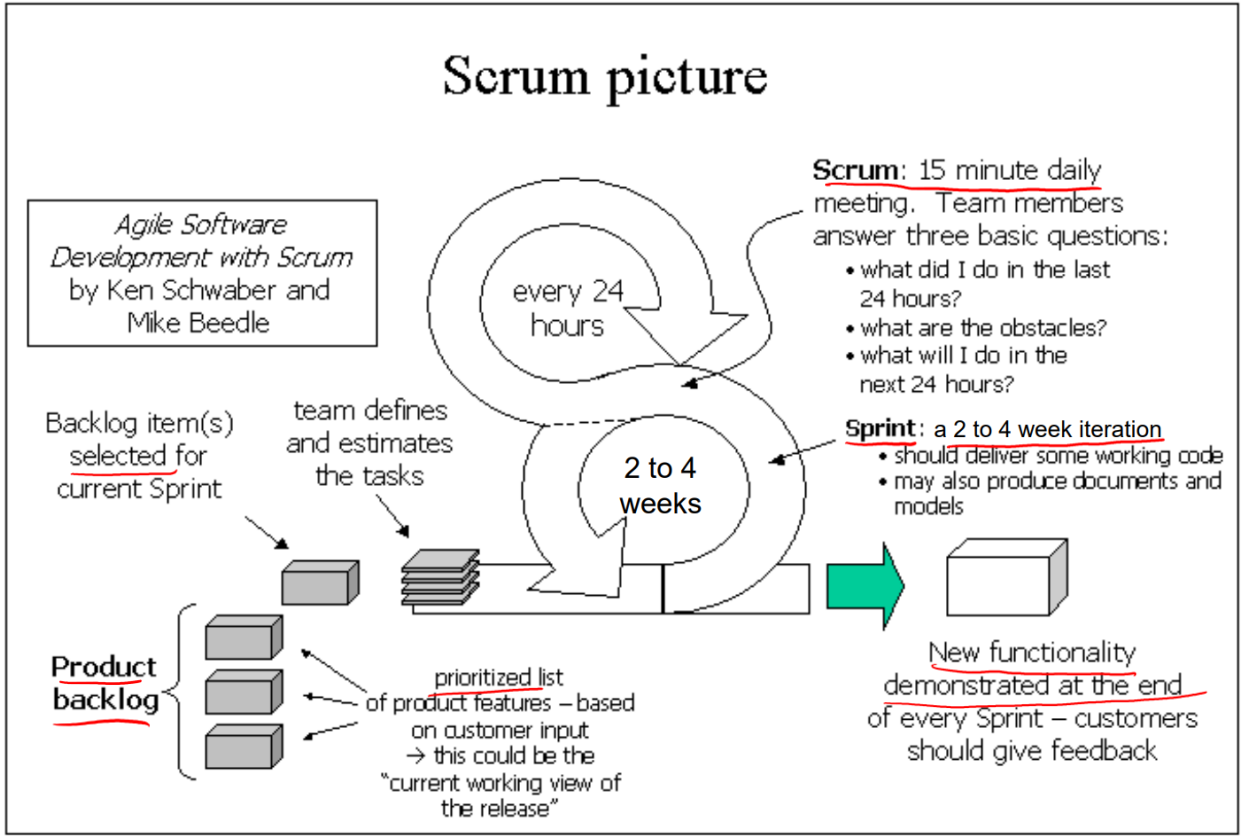


https://confluence.atlassian.com/jirasoftwareserver/velocity-chart-938845700.html

Summary of Scrum



Summary of Scrum



References

- Andrew Stellman, Margaret C. L. Greene 2014. Learning Agile: Understanding Scrum, XP, Lean and Kanban (1st Edition). O'Reilly, CA, USA.

- Ken Schwaber and Jeff Sutherland, The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game, https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100
- Agile Alliance. [agilealliance.info]
- Scrum Software Development. [https://en.wikipedia.org/wiki/Scrum_%28software_development%29]

Tutorial: Project Demo + Scrum Teams (Everyone must attend this week's Lab)

Next Lecture: Scrum - Expressing Requirements and Some Planning and Estimation

- Behaviour Driven Development (BDD)
  - User Stories
- User Interfaces, Scenarios and Storyboard
- Tools for Requirements in Agile Development