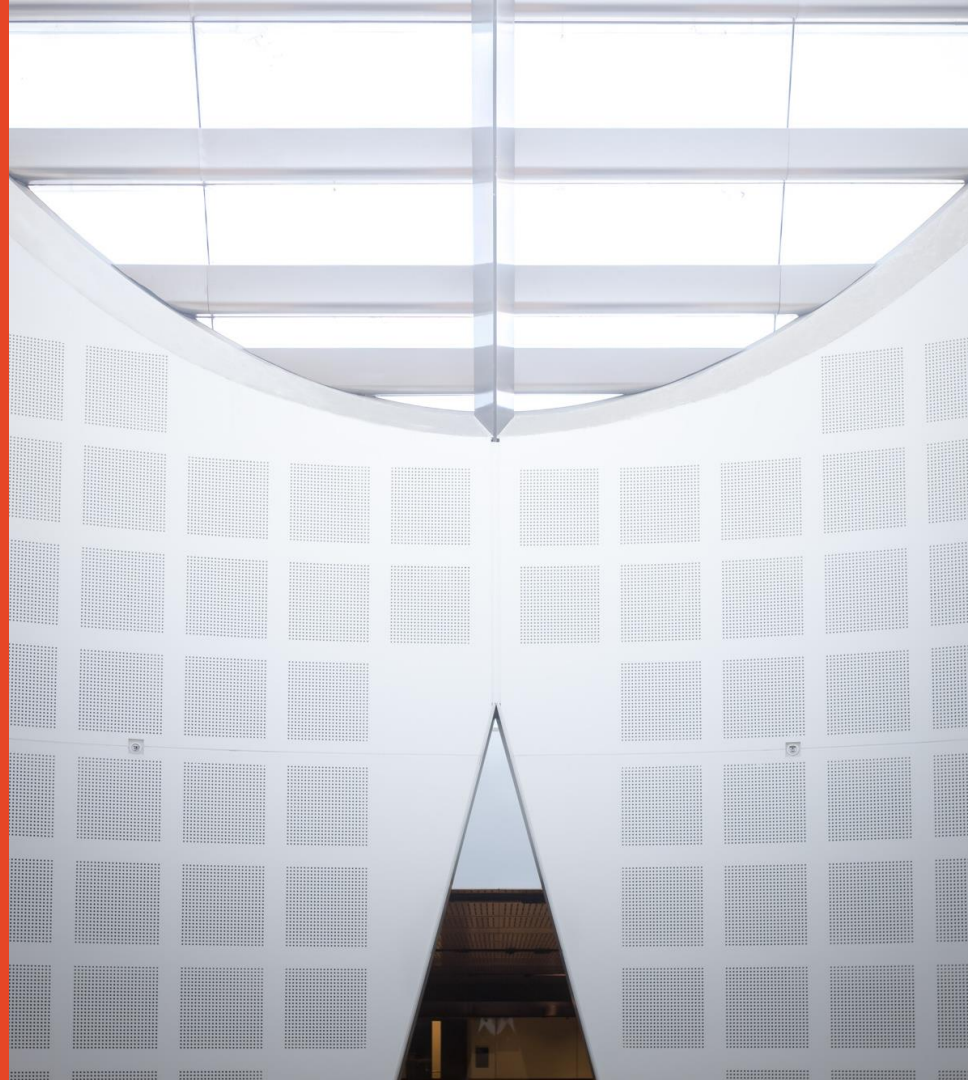


# Agile Software Development Practices (SOFT2412)

## Agile Methods – Scrum

Dr. Basem Suleiman

School of Information Technologies



# Agenda

- Agile Manifesto
  - Values and principles
- Agile Methods
  - Scrum, XP
- Scrum
  - Definition and Values
  - Teams and Roles
  - Events
  - Artifacts
  - Measuring progress

# Agile Process

- Agile advocates believe:
  - Current software development processes are too heavy-weight or cumbersome
  - Current software development is too rigid
  - More active customer involvement needed

# Agile Process

- Agile methods are considered
  - **Light-weight**
  - **People-based**
- Several agile methods
  - Extreme Programming (XP), **SCRUM**
- 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

Agile Manifesto: <http://agilemanifesto.org/>

© 2001, the above authors. This declaration may be freely copied in any form, but only in its entirety through this notice.

# 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 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 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 Methods

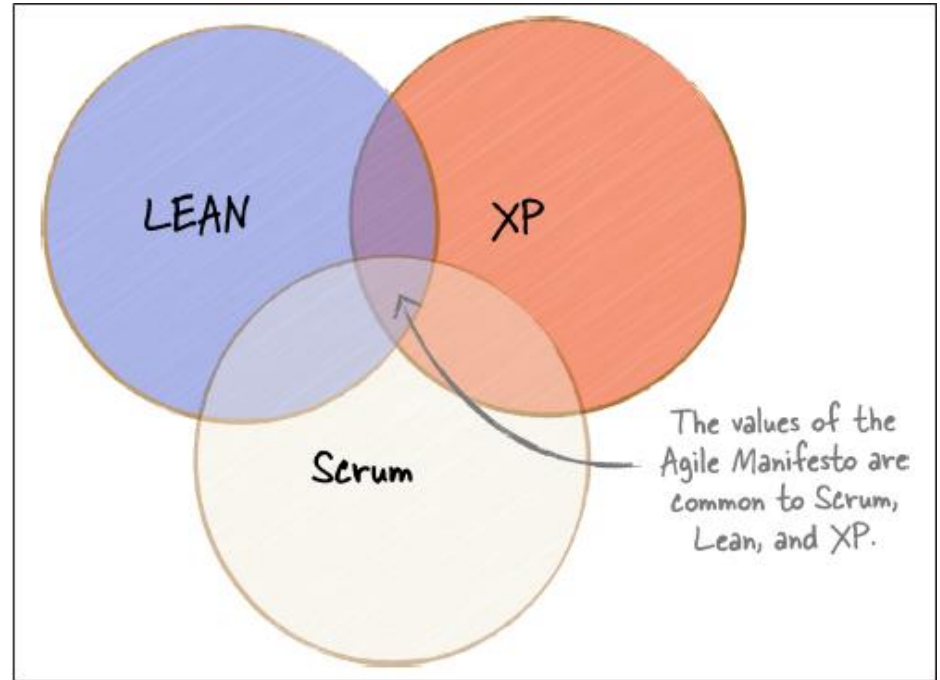
- Shared goal; delivering valuable software iteratively
- Might have some different practices to achieve this goal
- Examples of Agile methods:
  - Scrum
  - Extreme Programming (XP)
  - Lean software development
  - Kanban (lean development)

# eXtreme Programming (XP)

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

# Agile Methods

The common aspects among different Agile methods



# Scrum

- Method for “product development” lifecycle from H. Takeuchi and I. Nonaka (1986)
  - 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](https://en.wikipedia.org/wiki/Scrum_%28software_development%29)

# Scrum (Software Development)

- Managing software development to deliver products with highest value
- 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](https://en.wikipedia.org/wiki/Scrum_%28software_development%29)

# Scrum Theory

- Based on ‘*empirical process control*’ theory
  - Knowledge from experience and decisions based on knowns
  - Optimize predictability and control risks
- Pillars of empirical process control
  - **Transparency**: the process visible to those responsible for the outcome
  - **Inspection**: frequently inspect Scrum artefacts and progress to detect variances
  - **Adaptation**: adjust the process to the acceptable limits

# Scrum Values

- **Commitment:** personally commit to achieving the goals of the 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:** team members respect each other to be capable, independent

# 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

3

5

3



# The Scrum Team



# Scrum Team

- Scrum Team
  - Small enough to be agile
  - Cross-functional
  - Self-organizing
  - Deliver products iteratively and incrementally, maximizing opportunities for feedback
- Team roles
  - Development Team  $\leq 9$  (small team)
  - Product Owner (one person!) |
  - Scrum Master (one person!) |

# Scrum Team – The Product Owner

- **Product Owner:** maximize value of the product and the work
  - Understand requirements and its priorities
  - Manage 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 use
  - 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
  - Small enough and large enough!
  - Less than 3 members?
  - More than 9 members

$3 \leq \text{size} \leq 9$  ideally

# 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 or obstacles
  - 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

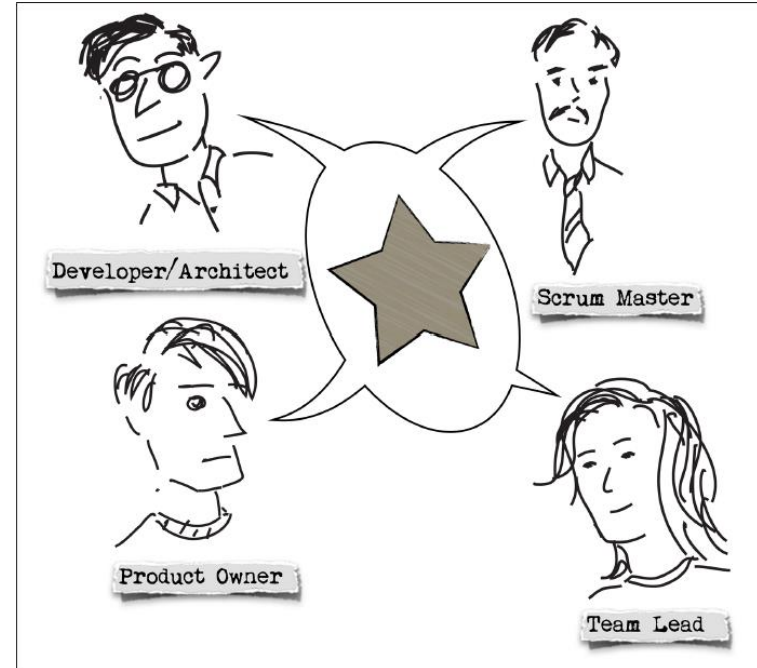
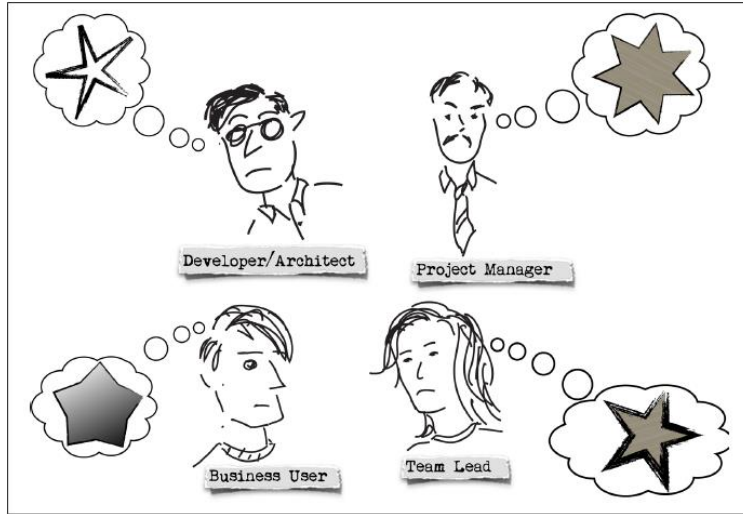
- Serves the product owner to ensure;
  - Mutual understanding of goals, scope and product domain
  - Effective ways for managing the product backlog
  - The Scrum team understands the need for clear and concise product backlog items;
  - Arranging the Product Backlog to maximize value;
  - Understanding and practicing agility
  - Facilitating Scrum events as requested or needed

# Scrum Team – The Scrum Master

- Serves the Dev team to ensure:
  - Be self-organizing and cross-functional;
  - Create high-value products;
  - Remove impediments to the Dev team's progress;
  - Facilitating Scrum events as requested or needed; and,
  - Coaching the Dev team



# Scrum Team – Interactions



# The Scrum Events



# Scrum Events

- To create regularity and minimize the need for meetings
- Time-boxed (max. duration)
- Cannot be changed once an iteration (sprint) has started
- Enable transparency and a formal approach 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)
  - Too long sprints may lead to changes in the definition
- Sprints have consistent durations during the product development
- Consists of the Sprint Planning, Daily Scrum, the Development Work, the Sprint Review and the Sprint Retrospective

# Scrum Events – Sprint Planning

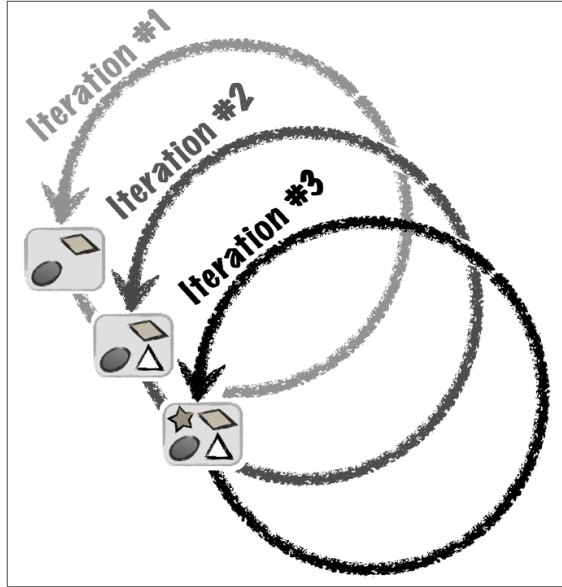
- Identify the Sprint Goal (items from the “Product Backlog”)
- Identify work to be done to deliver this
- Two-parts meeting (Scrum Master (SM), Product Owner (PO) and Dev team)
  - Before meeting: PO prepares prioritized list of most valuable items
  - Part 1 (max. 4 hours): PO & Dev team select items to be delivered at the end of the Sprint based on value and team’s estimate of how much work needed
  - Part 2 (max. 4 hours): Dev team (with the PO’s help) identify the individual tasks they’ll use to implement those items
- Output: Sprint Backlog (the items selected by the team for development)

meetings can take up to max 4 hours

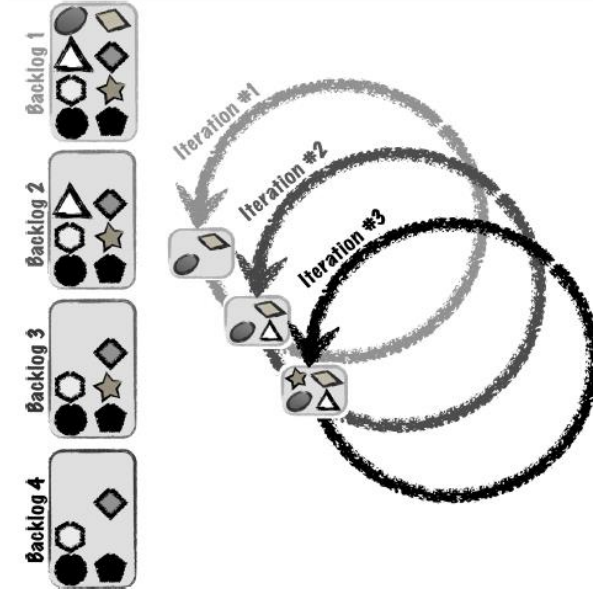
# Scrum Iteration Process

- 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
- once the sprint has started

# Scrum Iteration Process



mainly  
S1 log  
S2 log  
S3 log



# 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:
    - What did I do yesterday that helped the development team meet the Sprint Goal?
    - What I will do today to help the development team meet the Sprint Goal?
    - do I see any obstacles that presents 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

- End of Sprint meeting (max. 4-hours)
  - Dev team demonstrates working software to customers/stakeholders
    - Completed, tested & accepted features (by PO)
    - Functional working software
  - Stakeholders share their feedback/thoughts about the demo
  - The PO updates the Product Backlog with any changes for next Sprint planning
  - The SM ensures participants understand its purpose, and maintain within the time-box
- Output: revised Product Backlog and probable items for next iteration

# Scrum Events – Retrospectives

- Opportunity for the team to inspect itself and create plan for improvements
  - Inspect how the last Sprint went; people, relationships, process, & 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 SM and the Dev. team (PO) *all*
- Each to answer 2 questions:
  - What went well during the Sprint?
  - What can be improved in the future?
- The SM notes improvements that should be added to the Product Backlog
  - E.g., set-up a better build server, better design principles, changing office layout
- **Output:** identified improvements to be implemented in the next Sprint (adaptation)

# Scrum Artifacts



# Scrum Artifacts – Product Backlog

- Set of all features and sub-features to build the product (the “Plan” for multiple iterations)

<sup>1</sup> – Functions, <sup>2</sup> requirements, <sup>3</sup> enhancements and <sup>4</sup> fixes identified from previous Sprints

- Maintained by the PO in collaboration with customers and team
- The source of the product requirements
- The items ordered by priority – value to the customer

# ① Scrum Artifacts – Product Backlog

– What does a Product Backlog look like?

– Simple spreadsheet

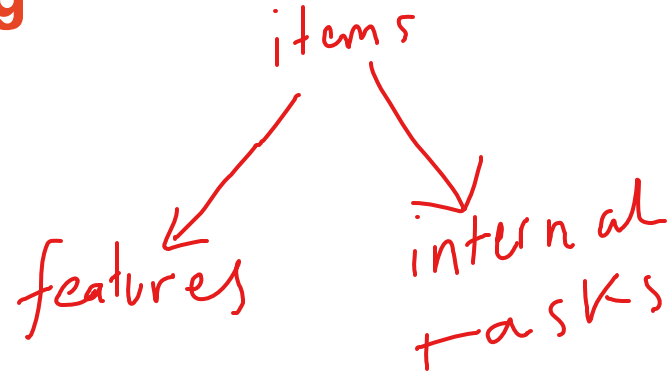
– Some items are “**customer features**”

- E.g., user screen, 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?*

– Effort estimates



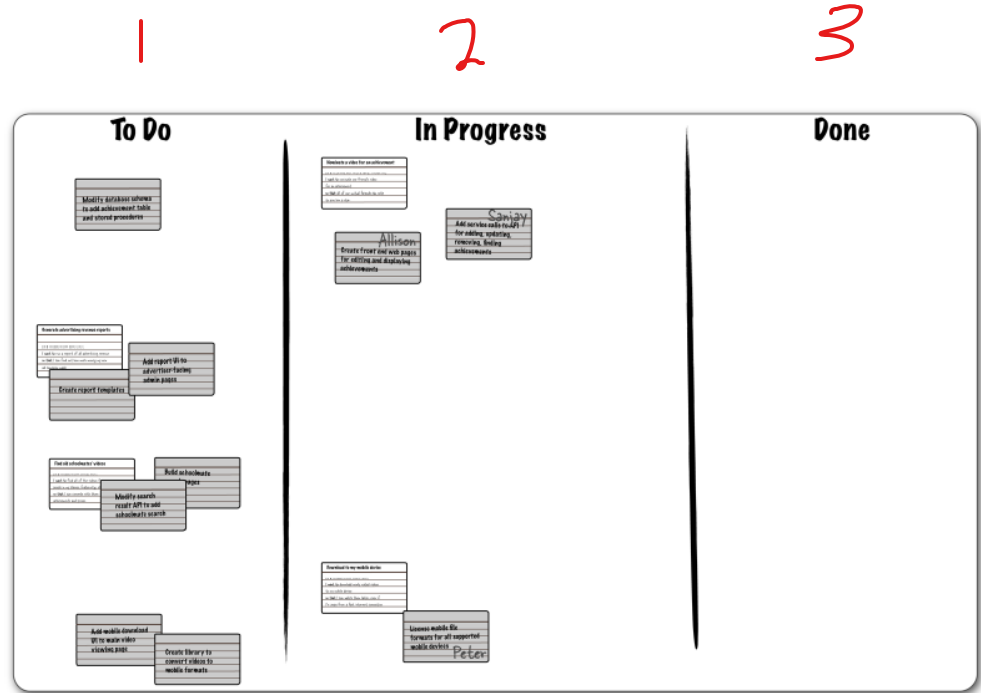
## Scrum Artifacts – Sprint Backlog

- Set of items selected for the Sprint, a plan for delivering the product increment and realize the Sprint Goal
- Identified by the Dev. team
- Includes at least one high-priority improvement 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



\_\_\_\_\_

- msteams  
Planner  
Github projects  
etc...



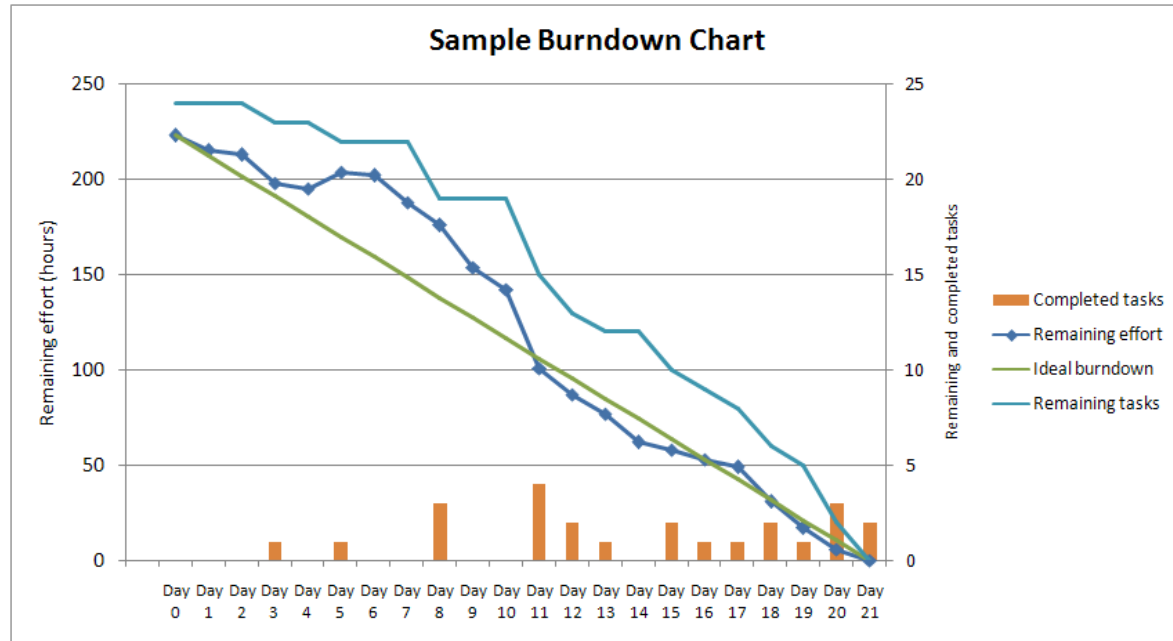
# Scrum Estimation



# Scrum Artifacts – Progress Monitoring

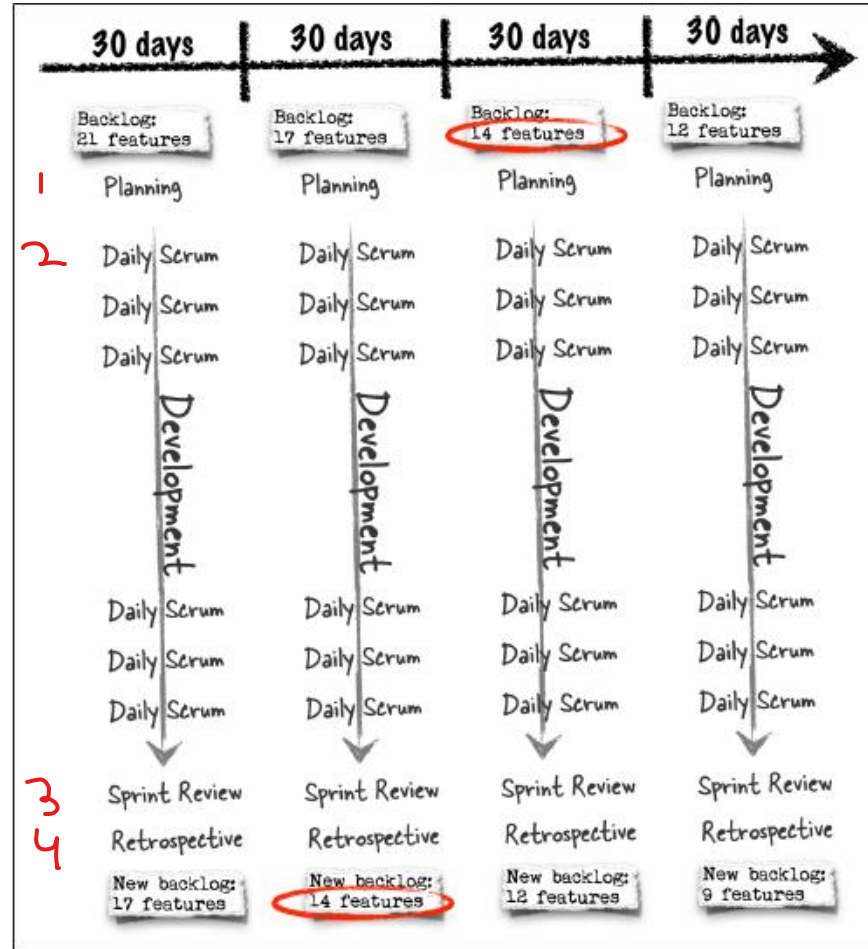
- Total work remaining to reach the goal – the product owner tracks this at least every Sprint Review
  - Compare with previous Sprints to assess progress toward projected work (transparent to all stakeholders)
  - Forecasting progress through burn-downs, burn-ups or cumulative flows
    - It's an estimate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

# Scrum Artifacts – Burn-down Chart (Example)



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

# Summary of Scrum



# References

- Andrew Stellman, Margaret C. L. Greene 2014. Learning Agile: Understanding Scrum, XP, Lean and Kanban (1<sup>st</sup> 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.  
[\[https://en.wikipedia.org/wiki/Scrum\\_%28software\\_development%29\]](https://en.wikipedia.org/wiki/Scrum_%28software_development%29)