Agile Methodologies

An introduction

Agenda

- Basics of Agile
  - Manifesto
  - Principles
- Scrum: an Agile Methodology

Agenda

- Basics of Agile
  - Manifesto
  - Principles
- Scrum: an Agile Methodology

What is Agile?

- A set of methods for software development
  - Iterative
  - Incremental
  - Assume changeability of requirements
- First appearance: 1957, IBM's Service Bureau Computation
- Consolidation and awareness: 2001, Agile Manifesto

Agile Manifesto

- Agile Values
  1. Individuals and interactions over processes and tools
  2. Working software over comprehensive documentation
  3. Customer collaboration over contract negotiation
  4. Responding to change over following a plan

Agile Manifesto

- Agile Values
  1. Individuals and interactions over processes and tools
     - People are the most important ingredient for success, even though a bad process can make great people ineffective
  2. Working software over comprehensive documentation
  3. Customer collaboration over contract negotiation
  4. Responding to change over following a plan

Agile Manifesto

- Agile Values
  1. Individuals and interactions over processes and tools
  2. Working software over comprehensive documentation
     - Software without documentation is a disaster, but that documentation needs to be human-readable, short, salient and high-level. Training people to lower level details is performed via close collaboration with trainees.
  3. Customer collaboration over contract negotiation
  4. Responding to change over following a plan

Agile Manifesto

- Agile Values
  1. Individuals and interactions over processes and tools
  2. Working software over comprehensive documentation
  3. Customer collaboration over contract negotiation
     - Successful projects involve customer feedback on a regular and frequent basis
  4. Responding to change over following a plan

Agile Manifesto

- Agile Values
  1. Individuals and interactions over processes and tools
  2. Working software over comprehensive documentation
  3. Customer collaboration over contract negotiation
  4. Responding to change over following a plan
     - Once customers see the system starts to function, they are likely to alter the requirements: make detailed plans for the next week (individual tasks), rough plans for the next 3 months (general requirements), and extremely crude plans beyond that (just an idea)

Agile Principles

- 12 principles, identified by 17 software developers* who met up at Snowbird, Utah
  1. Customer satisfaction by rapid delivery of useful software
  2. Welcome changing requirements, even late in development
  3. Working software is delivered frequently (weeks rather than months)
  4. Working software is the principal measure of progress
  5. Sustainable development, able to maintain a constant pace
  6. Close, daily cooperation between business people and developers
  7. Face-to-face conversation is the best form of communication (co-location)
  8. Projects are built around motivated individuals, who should be trusted
  9. Continuous attention to technical excellence and good design
  10. Simplicity - the art of maximizing the amount of work not done - is essential
  11. Self-organizing teams
  12. Regular adaptation to changing circumstances

* Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Stephen J. Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Agile Principles

- 12 principles, identified by 17 software developers who met up at Snowbird, Utah
  1. **Customer satisfaction by rapid delivery of useful software**
  2. Welcome changing requirements, even late in development
  3. **Working software is delivered frequently**
  4. **Working software is the principal measure of progress**
  5. Sustainable development, able to maintain a constant pace
  6. Close, daily cooperation between business people and developers
  7. Face-to-face conversation is the best form of communication (co-location)
  8. Projects are built around motivated individuals, who should be trusted
  9. Continuous attention to technical excellence and good design
  10. Simplicity - the art of maximizing the amount of work not done - is essential
  11. Self-organizing teams
  12. Regular adaptation to changing circumstances

  - **Strong connection between quality, customer satisfaction and frequency of delivery**
  - **Deliver every 2-4 weeks**
  - **Deliver working software, not documentation**

Agile Principles

- 12 principles, identified by 17 software developers who met up at Snowbird, Utah
  1. Customer satisfaction by rapid delivery of useful software
  2. **Welcome changing requirements, even late in development**
  3. Working software is delivered frequently (weeks rather than months)
  4. Working software is the principal measure of progress
  5. Sustainable development, able to maintain a constant pace
  6. Close, daily cooperation between business people and developers
  7. Face-to-face conversation is the best form of communication (co-location)
  8. Projects are built around motivated individuals, who should be trusted
  9. Continuous attention to technical excellence and good design
  10. Simplicity - the art of maximizing the amount of work not done - is essential
  11. Self-organizing teams
  12. **Regular adaptation to changing circumstances**

Agile Principles

- 12 principles, identified by 17 software developers who met up at Snowbird, Utah

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Working software is the principal measure of progress
5. **Sustainable development, able to maintain a constant pace**
6. **Close, daily cooperation between business people and developers**
7. **Face-to-face conversation is the best form of communication**
8. Projects are built around motivated individuals, who should be trusted
9. Continuous attention to technical excellence and good design
10. Simplicity - the art of maximizing the amount of work not done - is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstances

- **Co-located/interacting teams**
- **No overtime or final rush**

Agile Principles

- 12 principles, identified by 17 software developers who met up at Snowbird, Utah
  - **Focus on quality of both team and work**
  - **Distribution of tasks across team members, according to their skills without degrading to silos of responsibility: everyone is responsible for everything**

8. **Projects are built around motivated individuals, who should be trusted**
9. **Continuous attention to technical excellence and good design**
11. **Self-organizing teams**

Agile Principles

- 12 principles, identified by 17 software developers who met up at Snowbird, Utah
  1. Customer satisfaction by rapid delivery of useful software
  2. Welcome changing requirements, even late in development
  3. Working software is delivered frequently (weeks rather than months)
  4. Working software is the principal measure of progress
  5. Sustainable development, able to maintain a constant pace
  6. Close, daily cooperation between business people and developers
  7. Face-to-face conversation is the best form of communication (co-location)
  8. Projects are built around motivated individuals, who should be trusted
  9. Continuous attention to technical excellence and good design
  10. **Simplicity - the art of maximizing the amount of work not done - is essential**
  11. Self-organizing teams
  12. Regular adaptation to changing circumstances

Agenda

- Basics of Agile
  - Manifesto
  - Principles
- Scrum: an Agile Methodology

Some Agile Methods

- Agile Unified Process
- Disciplined Agile Development
- Dynamic System Development Method
- Crystal Clear
- Extreme Programming
- Feature-Driven Development
- Kanban
- Lean Software Development
- **Scrum**
- Test-Driven Development

Scrum

- Not a software development process, but a set of practices
- A framework to instantiate
- A software development process can leverage Scrum practices
- Based on an iterative, incremental and empirical approach
- Transparent

Scrum

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

3 Roles

- Product Owner
- Scrum Master
- Development Team

4 Meetings

- Sprint Planning Meeting
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Scrum

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

3 Roles

- Product Owner
- Scrum Master
- Development Team

4 Meetings

- Sprint Planning Meeting
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Scrum Sprint

- Single iteration of fixed length
- Length < 1 month
- Development is performed during a Sprint
- During each Sprint, the same amount of work shall be done
  - The amount of work done is called velocity and represents the development pace
- A new Sprint starts immediately after the end of the last Sprint
- Each Sprint ends with an increment on the product

Scrum

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

3 Roles

- Product Owner
- Scrum Master
- Development Team

4 Meetings

- Sprint Planning Meeting
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Artifact: Product Backlog

- Ordered list of everything that might be needed in the product
  - Features, functions, requirements, enhancements, and fixes
- Single source of requirements for any changes to be made to the product
- Never complete
- Product Backlog items have attributes: description, order and estimate
- Product Backlog refinement is the act of updating details, estimates and order to items in the Product Backlog
  - Only items on top of the product backlog are detailed at fine-grained level
- Backlog items can be grouped by custom criteria

Artifact: Sprint Backlog

- Selection of Product Backlog items to develop during a sprint
- Items are small and detailed enough so that they can be understood and developed within the sprint by the Development Team
- Items are detailed and modified as the Sprint proceeds
- No new items can be added from the Product Backlog during the Sprint

Artifact: Increment

- Sum of “Done” Product Backlog Items
  - The meaning of “Done” must be shared among stakeholders
  - No prescriptive definition
    - The team shall define one
      - E.g. define a checklist of actions to perform before declaring an item as “done”
- Shipped at the end of the Sprint
- Not necessarily a release

Scrum

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

3 Roles

- Product Owner
- Scrum Master
- Development Team

4 Meetings

- Sprint Planning Meeting
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Scrum Team and Roles

- Typically 5-11 members
- Three roles
  - Product Owner
  - Development Team
  - Scrum Master

Scrum Team and Roles

- Typically 5-11 members
- Three roles
  - **Product Owner**
    - Exclusive management and control of the product backlog
      - Create backlog items with the development team and assigns priorities
    - Represents the voice of the customers
    - Responsible for the value of the work done
    - Responsible for transparency and clarity
    - A single person, not a committee
  - Development Team
  - Scrum Master

Scrum Team and Roles

- Typically 5-11 members
- Three roles
  - Product Owner
  - **Development Team**
    - Developers who turn the backlog items into shippable software
    - Solely responsible on how to develop the system
    - Responsible for estimating the backlog items
    - Cross-functional
    - Self-organizing
    - All “developers”, no one is over the others
  - Scrum Master

Scrum Team and Roles

- Typically 5-11 members
- Three roles
  - Product Owner
  - Development Team
  - **Scrum Master**
    - Supports the Product Owner to effectively manage the product backlog
    - Supports the Development Team by coaching on agile and removing impediments
    - Supports the organization in planning and adopting Scrum

Scrum

3 Artifacts

- Product Backlog
- Sprint Backlog
- Increment

3 Roles

- Product Owner
- Scrum Master
- Development Team

4 Meetings

- Sprint Planning Meeting
- Daily Scrum
- Sprint Review
- Sprint Retrospective

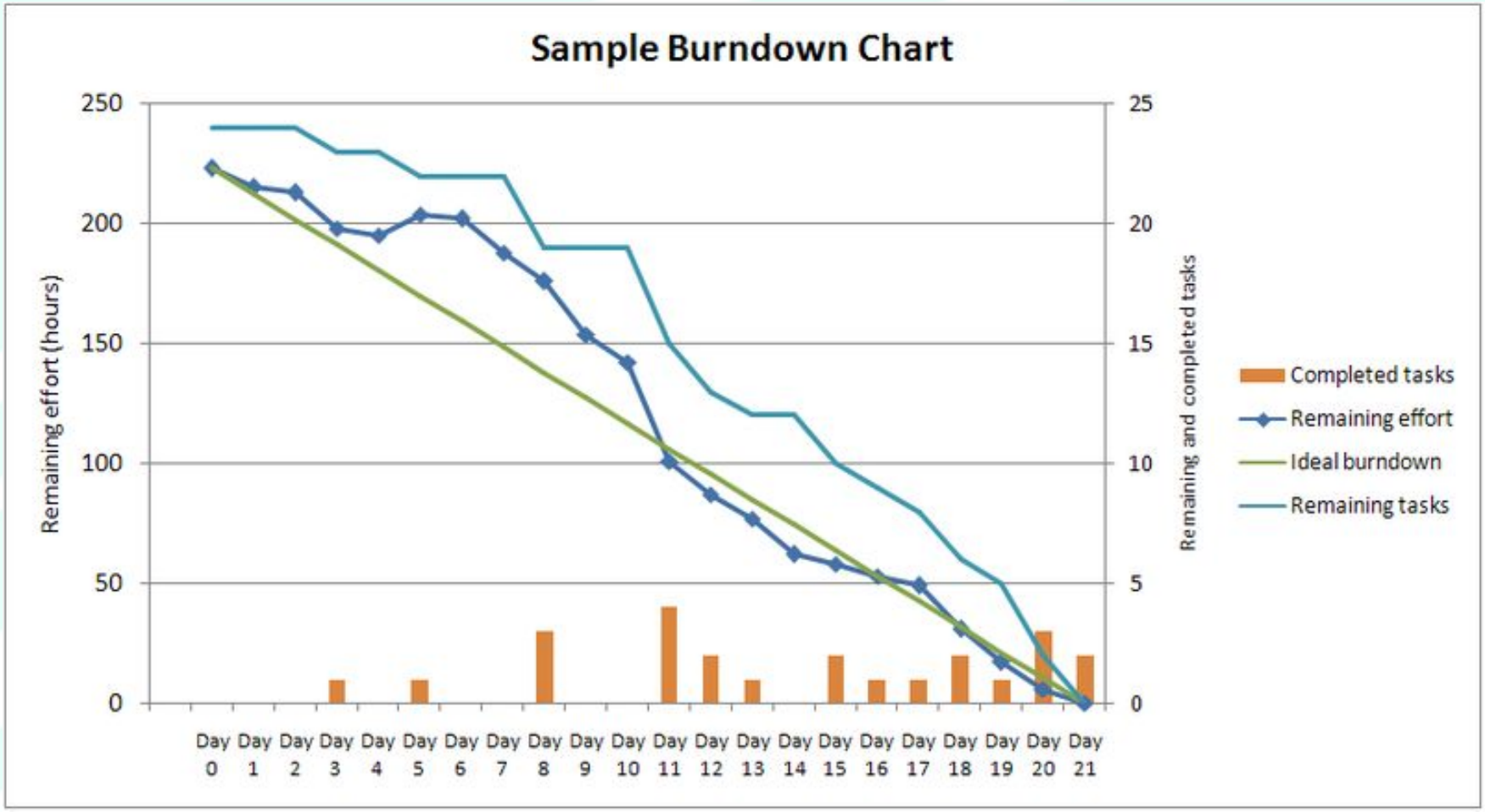
Sprint Planning Meeting

- Length ≤ 8 hours at the beginning of a Sprint
- The entire Scrum Team participates
- Product Owner presents the top priorities of the next Sprint
- Development Team
  - Picks what can be developed within the next Sprint
  - Defines a Sprint Goal, i.e. the objective to meet by the end of the Sprint
  - Finally, defines a plan to achieve the goal

Daily Scrum Meeting

- ~15 minutes at the beginning of the day
- Development Team and Scrum Master participate
- Synchronization of the work to be done during the day
- Each team member shares
  - Yesterday progress
  - Today plans
  - Encountered problems

Example: Burndown Chart



Sprint Review Meeting

- Length ≤ 4 hours at the end of a Sprint
- Scrum teams and all stakeholders participate
- Informal meeting, not a status meeting
- Discussion on the last Sprint
- The Product Owner reports on the achievement of the Sprint goal
- The Team reports on encountered issues and shows the new delivered increment
  - Incomplete work is not shown
- Product Backlog is discussed

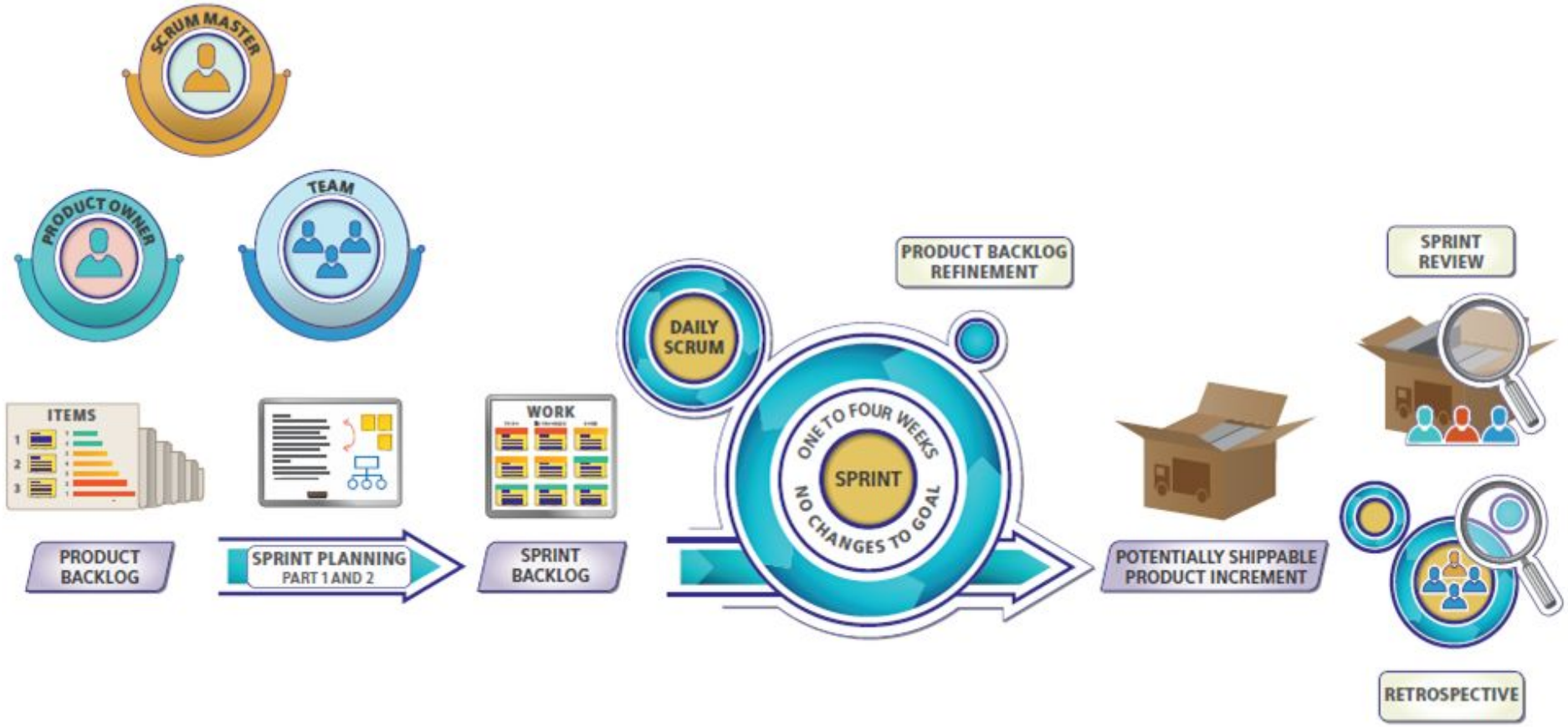
Retrospective Meeting

- Length ≤ 3 hours at the end of the Sprint
- The entire Scrum team participates
- Assessment of team dynamics and tools
- Adjustment of process, team and interactions to make the next Sprint more efficient and productive
- Plan on how to implement the improvement

Cancelling a Sprint

- It happens when the Sprint has no value anymore
  - I.e. the Sprint Goal becomes obsolete
- Only the Product Owner can make the decision
- Should be very rare: Sprints have very short-term deadlines

Sprint Snapshot



Source: Scrum Primer