SCRUM
An Agile Management Process

Characteristics

- An Agile process ‘to manage and control development work’
- No specific engineering practices are prescribed; ‘a wrapper for existing software engineering practices’
- Team-based approach - teams are self-organising
- Scalable for large projects
- Iterative - short iterations (30 days) called **sprints**
- Assumes rapidly changing requirements

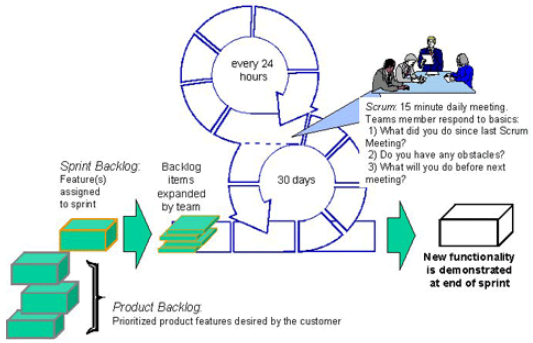
Some Other Claims

- Scrum, like many other (Agile) processes has a list of general claims, here are some examples:
  - Scrum is a process that controls the chaos of conflicting interests and needs
  - Scrum is a way to improve communications and maximise co-operation
  - Scrum is a way to detect and cause the removal of anything that gets in the way of developing and delivering products
  - Scrum is a way to maximise productivity
  - Scrum is a way for everyone to feel good about their job, their contributions, and that they have done the very best they possibly could.

Scrum Roles

- Scrum Team
  - 5-10 individuals, self-organising
  - Do not change team during sprint
- Product Owner
  - Customer/user representative who makes prioritisation decisions (Speaks for the customers and users)
  - Knowing what to build and in what sequence
- Scrum Master
  - Removes impediments (troubleshooter, facilitator)
  - Provides link between management and the team
  - Enforces scrum values (cf. the ‘coach in XP?)

Scrum Overview



Principles

- Each sprint delivers a product that is potentially releasable
- There should be no changes during a sprint; new requirements will be added to the Product Backlog
- Plan duration of sprint around Product Owner's commitment not to make changes during the sprint
  - original Guideline is 30 days
  - others suggest as low as two weeks
  - 'a constant duration leads to a better rhythm' (Mike Cohn)

Product and Sprint Backlogs

- **Product Backlog** defines the **requirements** for the product
  - List of all desired work on the project, combination of user functions ('stories') and system tasks
  - Prioritised by the Product Owner
  - Top items are selected for the next Sprint
  - Re-prioritised after each Sprint
- **Sprint Backlog** defines the tasks for the next sprint
  - Task-based, agreed at the Sprint Planning Meeting

Starting a Sprint

- Each Sprint starts with a **Planning Meeting**, in two parts, with the Product Owner initially and then just the team for detailed planning
- Decide on the **Sprint Goal**
  - Input from the Product Owner (and management)
- Team self-organises around how they will meet this Sprint Goal
  - Manager does not assign tasks to individuals
- Sprint Backlog is created
  - List of tasks necessary to achieve the work selected for this sprint

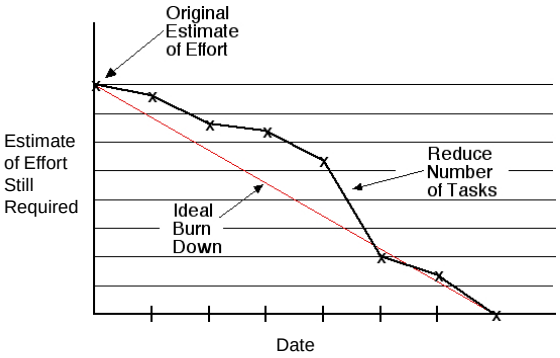
During the Sprint

- During a sprint there are **daily** scrum meetings
  - About 15 minutes long
  - Stand-up (face to face, not reports or e-mails)
  - Entire team sees the whole picture
  - Create peer pressure to deliver on promises
  - Not for problem solving
- Three questions
  - What did you do yesterday?
  - What will you do today?
  - What obstacles are in your way?
- Non-team members, 'chickens', may attend for information but do not contribute, only 'pigs' contribute

Sprint Burndown Chart

- Progress is measured by a Sprint Burndown Chart showing progress and estimating remaining effort
- Changes may be made to the Sprint Backlog:
  - add new tasks required to meet the Sprint Goal
  - remove unnecessary tasks
  - if a proposed change affects the functionality (the Sprint Goal) then this must be agreed with the Product Owner; this will also impact on the Product Backlog
- Update estimates to take account of new information about progress

Burndown Chart Example



Finishing a Sprint

- Each Sprint finishes with a **Review Meeting**
- Team presents what it has accomplished
- Normally demonstration of new functionality
- Guidelines
  - Informal, no Powerpoint!
  - Not more than two hours preparation time
- Open to product owner, customers, managers, other developers, … (Anybody can attend)

Scrum and the Agile Manifesto

- How does Scrum fit with the Agile Manifesto principles?
  - *Individuals and interactions over processes and tools*
    Emphasis very much on individuals and interactions; processes are lightweight and no tools are prescribed
    Management framework, not a complete development process
  - *Working software over comprehensive documentation*
    Progress is measured daily by delivered product items (daily meetings, Sprint Burndown); there is necessary documentation such as Product and Sprint Backlogs

Scrum and the Agile Manifesto (2)

  - *Customer collaboration over contract negotiation*
    Scrum is unusual because it tends to be more contractual than other Agile processes - the Product Owner agrees to the Sprint Goal and is only consulted if significant changes are required
  - *Responding to change over following a plan*
    Change is managed at two levels
    - Within a Sprint by monitoring the Burndown and modifying the Sprint Backlog (micro-management)
    - Using the Product Backlog to manage change on a larger scale (changes can be made to Product backlog at any time but will not affect the current Sprint)

Scrum and XP

- Scrum and XP are often seen as complementary
- Scrum provides the management framework while XP provides the implementation technology
  - Product Backlog is defined using stories with ideal time estimates
  - Sprint Backlog is task oriented
  - Sprint Burndown reflects the velocity of the team
  - Sprints correspond to iterations of XP using: pair programming, test first, refactoring, …
  - Use of stand-up meetings is common to both approaches

References

- The best reference source I have found for Scrum is:
  http://www.mountaingoatsoftware.com
- This site includes an overview of Scrum and several good presentations, including:
  - TelelogicScrum.pdf - an introduction to Scrum by one of the experts in the field, copy with the SEP4 Lecture Notes on Moodle.