SWINBURNE UNIVERSITY OF TECHNOLOGY

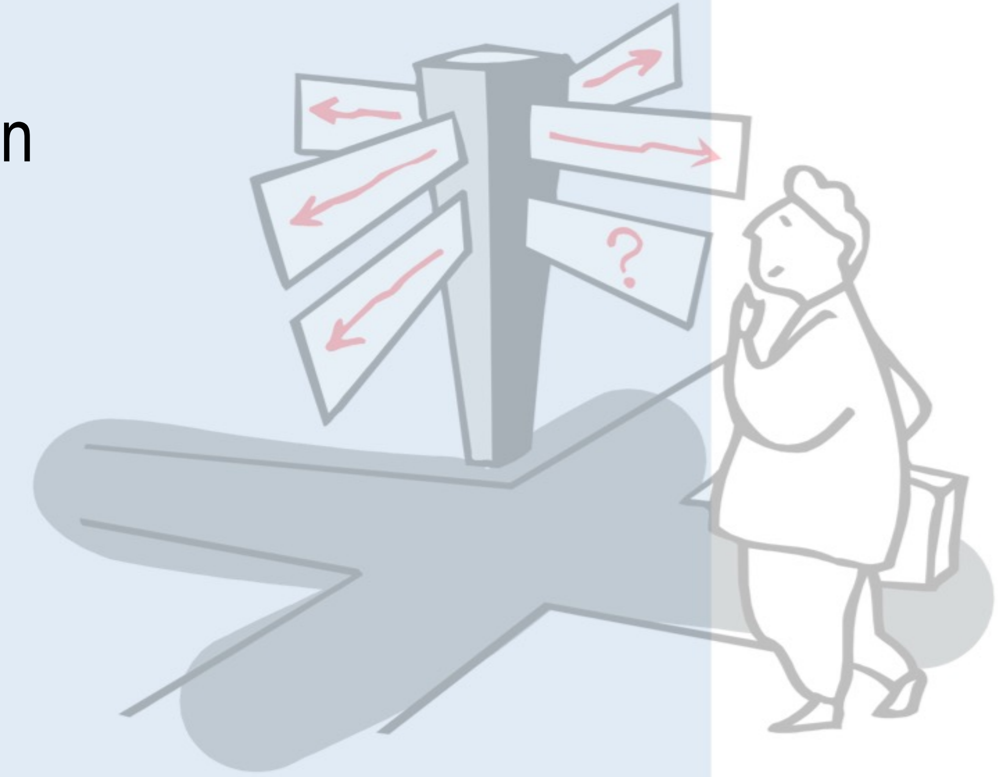
SWE20001 Managing Software Projects

Lecture 1b

Scrum

Roadmap

- Scrum - a brief Introduction
- Major Practices of Scrum
- Video by Ken Schwaber



Principal References

- Ken Schwaber and Mike Beedle, *Agile Software Development with SCRUM*, Prentice Hall, 2001.
- Ken Schwaber, *Agile Project Management with Scrum*, Microsoft Press, 2004.

Manifesto for Agile Software Development

Prefer	Still important
1. Individuals and interactions	Processes and tools
2. Working software	Comprehensive documentation
3. Customer Collaboration	Contract negotiation
4. Responding to change	Following a plan

Source: http://agilemanifesto.org/

What is Scrum?

- Not an acronym – derived from observing a rugby scrum
- Iterative software development process
  - □ Empirically controlled (rather than rigid, plan-based)
  - □ Focuses on maximizing business value
  - □ Constant improvement and priority/risk driven

What is Scrum? (cont’d)

- Philosophical underpinnings
  - “Methodologies that are repeatable are specific to only a very narrow subset of situations whereas more general methodologies are not repeatable. The cause of the conundrum is the noise.” http://codebetter.com/blogs/darrell.norton/pages/50339.aspx
  - Essentially abstraction causes a system to become more chaotic – this can be corrected using an empirical control system.
- Scrum assumes that the assumptions that underpin plan-driven methodologies cannot be fulfilled!

Scrum is an empirical process

- Empirical processes do not specify software development activities in advance…
  - Treat the development process as a black box
  - Development starts with a best estimate at what will work.
    Then the process is regularly inspected at defined intervals (i.e., every day), assessed, and adjusted, if necessary
  - Process is not well-defined, but based around techniques that focus on **learning from past experience**
  - Knowledge is used for **guidance going forward**, not for repeatability of past actions
  - Requires a high level of trust from the client

Basis for Empirical Processes

- Ziv's Uncertainty Principle in Software Engineering:
  - □ *Uncertainty is inherent and inevitable in software development processes and products.*
- Humphrey's Requirements Uncertainty Principle:
  - □ *For a new software system, the requirements will not be completely known until after the users have used it.*
- Wegner's Lemma:
  - □ *It is not possible to completely specify an interactive system.*

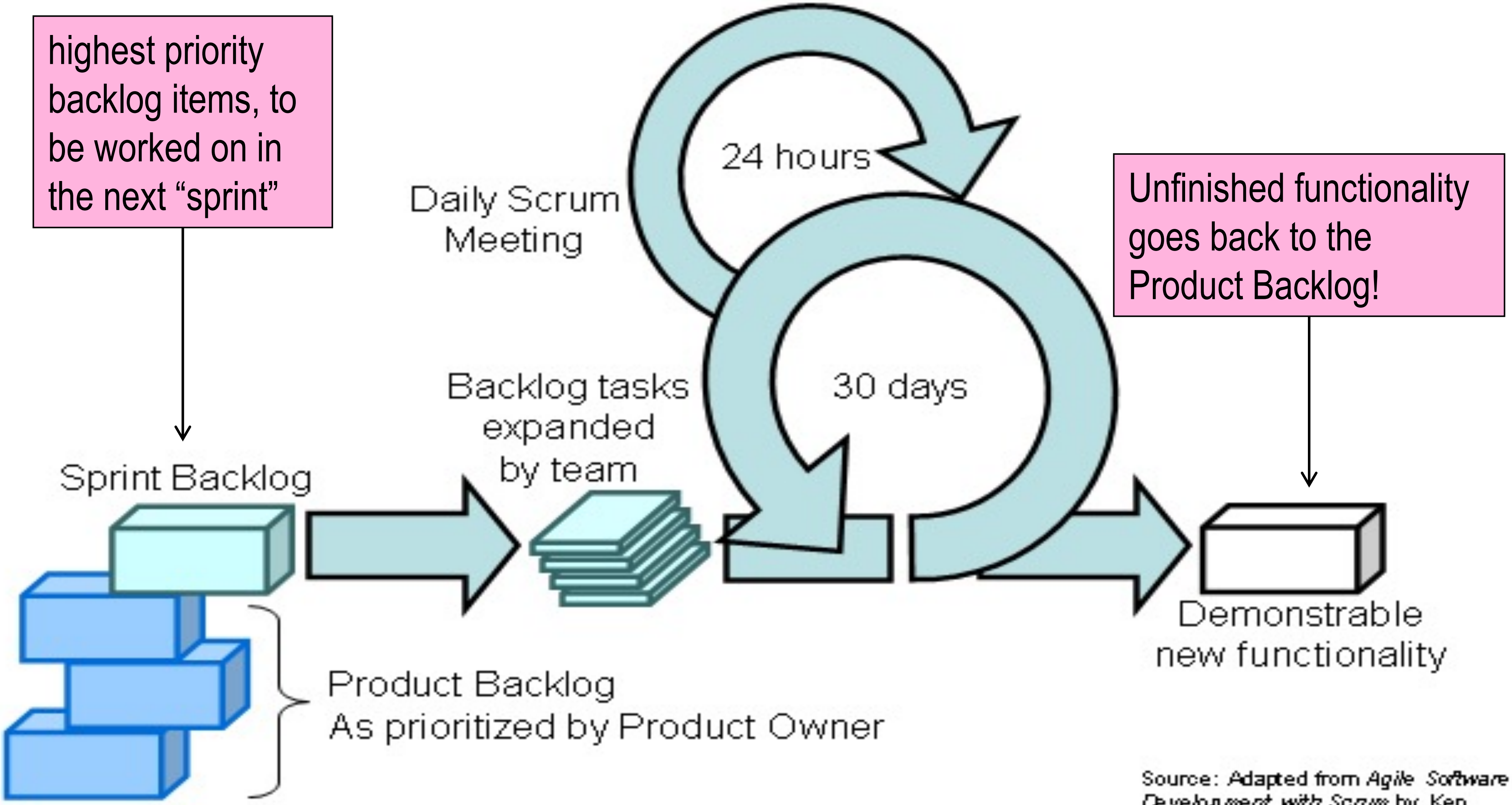
Foundations of Scrum

- Built-in instability
- Self-organizing project teams
- Overlapping development phases
- “Multi-learning”
- Subtle control
- Organizational transfer of learning

Scrum – The Process

highest priority backlog items, to be worked on in the next “sprint”

Unfinished functionality goes back to the Product Backlog!


24 hours
Daily Scrum Meeting
Backlog tasks expanded by team
30 days
Sprint Backlog
Product Backlog
As prioritized by Product Owner
Demonstrable new functionality

Source: Adapted from *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle.

Scrum Roles (broadly speaking…)

- Product Owner
- Scrum Master
- Scrum Team Member

Product Owner

- Manages the vision and ROI
- Decides on when to release the product into a production environment
- Understands business level risks as well as consequences of actions/decisions from a business perspective
- Develops and maintains the product backlog – team assists

Scrum Master

- Owns the process
- Motivates / coaches / guides team
- Ensures that the team does not go off target
- Makes all project level decisions
- Ensures that the team has the necessary environment and information to deliver on the vision within the constraints set

Scrum Team Member

- Plan tasks
- Manage their commitments
- Work as a team to achieve the vision set for them
- Is responsible for delivery of functionality

Product Backlog

- Everything currently assessed to be necessary to bring project to completion
  - could be features not yet implemented, bugs not yet resolved, documents not yet written, etc.
  - features, architecture, defects, quality attributes, constraints, etc.
- Owned by the ‘product manager or product owner’
  - single role ownership
- Non-binding (effort) estimates for each item in the backlog are put forward by the product owner/manager – the team assists in this effort

Scrum Team and Sprinting

- Once the product backlog is ready and estimates put forward – the team starts the sprint (i.e. the iteration), during which they will address chosen backlog items (the sprint backlog)
  - Sports metaphor – once the game starts, the rules cannot be changed, i.e. the sprint (product) backlog cannot be revised (until end of the sprint)
  - Each sprint is *30 calendar working days* (approx. 1½ months in business working days)
  - The team is *self-organising* (as is a footy team) – under the guidance of the *scrum master* (this is a role – does not mean a separate person is needed 100% for this)
  - Team expands backlog into tasks and gets to work on them
  - Team should be 7 members (+/- 2) [less than 5 discouraged]
    - Hypothesis – *odd member teams work better*

Sprint – Before each sprint

- Sprint Planning Meeting [first few hours of the sprint]
- The team takes time to plan for the sprint
  - Product owner, Scrum master and team meet to prioritise the functional requirements and organise the work
  - The team identifies the set of functionality they can deliver – the choice is made by assessing business value of the functionality; this is the **sprint backlog**
  - Ends in a short (but formal) presentation outlining the commitment from the team – any stakeholder may attend this session

Sprint Planning Meeting

- The following activities are undertaken in a sprint planning session
  - □ Sprint planning meeting time/date set (Scrum master)
  - □ Product backlog is presented, with any information helpful to the team from the business (Product owner)
  - □ Select product backlog items for the sprint (Entire Team) → sprint bklg
  - □ Define the goal (vision) for the sprint (Product Owner)
  - □ Construct sprint backlog (Scrum team + Scrum Master)
  - □ Formal presentation to stakeholders (Scrum master)

Note: There may be an additional round of planning after the formal presentation if there is any significant feedback, else that is used as input into detailed planning or for next iteration.

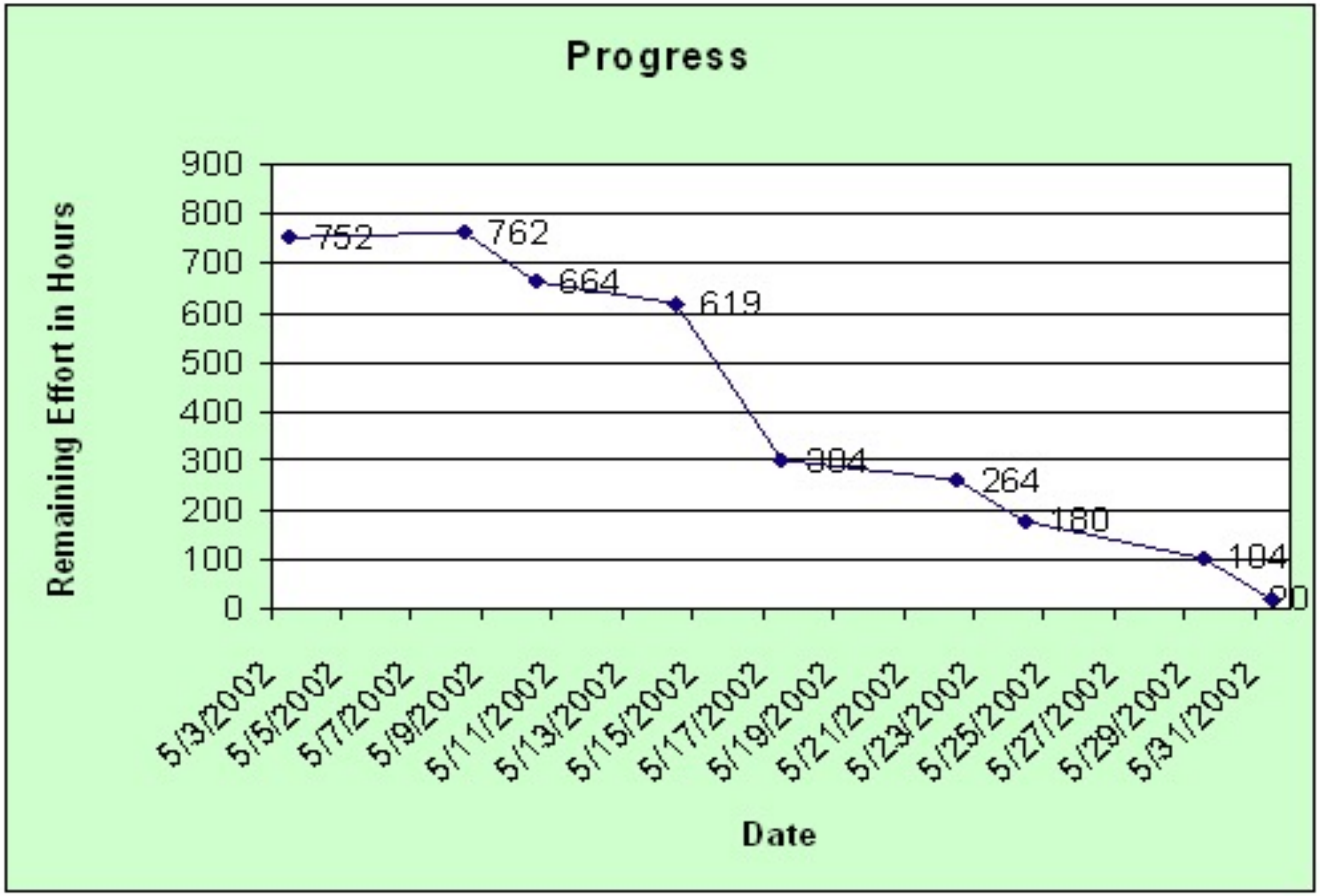
Sprint – In the sprint

- Team members do their jobs / tasks
- Perform daily standup (aka daily scrum)
- Track and report progress via burn-down chart

Daily Stand-up / Scrum

- In short it is nothing more than a *stand-up meeting*
- Usually at start of the day, some does it at the end
- Every team member answers the following questions:
    1. What did you do yesterday?
    2. What are you doing today?
    3. What prevented you from doing work?
- No detailed discussions in this forum – the aim here is to *increase transparency*, not debate!
- Must be held at the ‘same time, same location’ every day
- Scrum master responsible for correcting information gained from Question 3 above
- Some quick decisions may be taken, left to the discretion of the scrum master (works best just for information gathering)

Burn Down Chart Example



http://codebetter.com/blogs/darrell.norton/pages/50339.aspx

Managing Sprints

- Burn-down charts are used to get an idea of progress (or lack of) – quantitative data is essential for this
  - Burn-down charts use estimated hours remaining on the project (in terms of work/effort)
  - The slope of the line is used to derive *team velocity*
- Information gained in daily meeting used to cross-verify data being shown on burn-down chart
- A number of sprints is generally undertaken before a product is released

Sprint – After the sprint

- Sprint Review (= product review [traditional])
- Sprint Retrospective (= process review [traditional])
- Last few hours in the sprint
- The team reviews the previous iteration (sprint) and identifies what they can learn from it all
  - Similar to an iteration retrospective but includes a presentation of the work done in the previous iteration (i.e. a product functionality demonstration)

Challenges when using Scrum

- Need a good, effective manager
  - Must be willing to make positive changes (quickly)
  - Must constantly monitor both quantitatively and qualitatively
- Team makes most decisions normally made by managers in traditional approaches – challenging to implement in practice
  - Managers are not generally encouraged to lead this way
- Not every person is happy to accept responsibility for their own actions, the amount of transparency makes it hard to be average
  - Some people work better in a structured environment where work is delegated – not everyone is comfortable making decisions

Benefits of Scrum

- A *transparent development process*
- Productivity increases – once transition is made
- Improvements are slow, but continuous
  - □ The transparency forces the team to take corrective actions – a sports team would not intentionally take actions to lose
- Accepts software development is not always a predictable, stable and repeatable process
  - □ Progress is made even when requirements are evolving
  - □ Constant feedback, empirical evidence to guide team
  - □ Team communicates better

Will Scrum always work?

- If scope of project is very well-defined, not expected to change, and solution domain is well understood:
    - *Better use a "traditional" approach (e.g., Waterfall) than Scrum.*
- **Scrum does not directly apply (i.e. unmodified):**
    - High risk projects
    - Safety critical systems
    - Very large teams (especially multi-disciplinary teams)
    - Fixed-price (high-risk for team)
    - Fixed-deadline (say to meet some Govt. regulation)
- Aspects and the good practices of Scrum can however be applied to above types of projects (choose wisely!)

Scrum Video – Ken Schwaber

- www.youtube.com/watch?v=_47VWIvOKH8
- This is a 1 hour video by Ken Schwaber given to Google Employees on Scrum
  - Excellent talk!
  - Watch it and take notes
  - Note some high-level concepts
- Ken Schwaber and Jeff Sutherland developed the Scrum process

Resources/References

- Scrum Development on a Page
  - http://xp123.com/xplor/xp0401/
- Scrum - Wiki
  - https://en.wikipedia.org/wiki/Scrum_(software_development)
- Scrum Tutorial, by Ken Schwaber
  - http://www.controlchaos.com
  - http://www.agilealliance.com