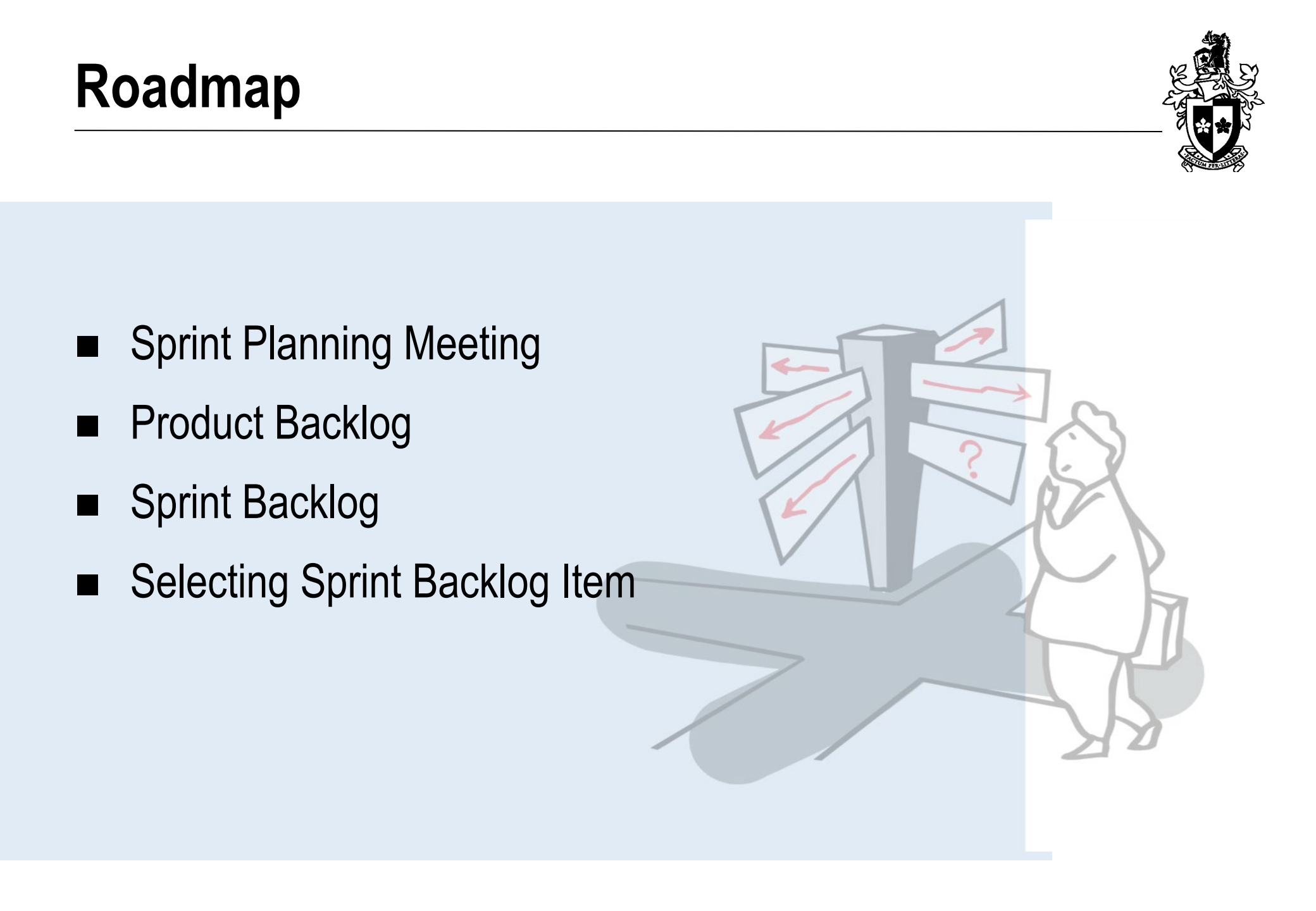
COS30049 Computing Technology Innovation Project

(Week 2)

SWINBURNE UNIVERSITY OF TECHNOLOGY

Sprint Planning Meeting

Roadmap

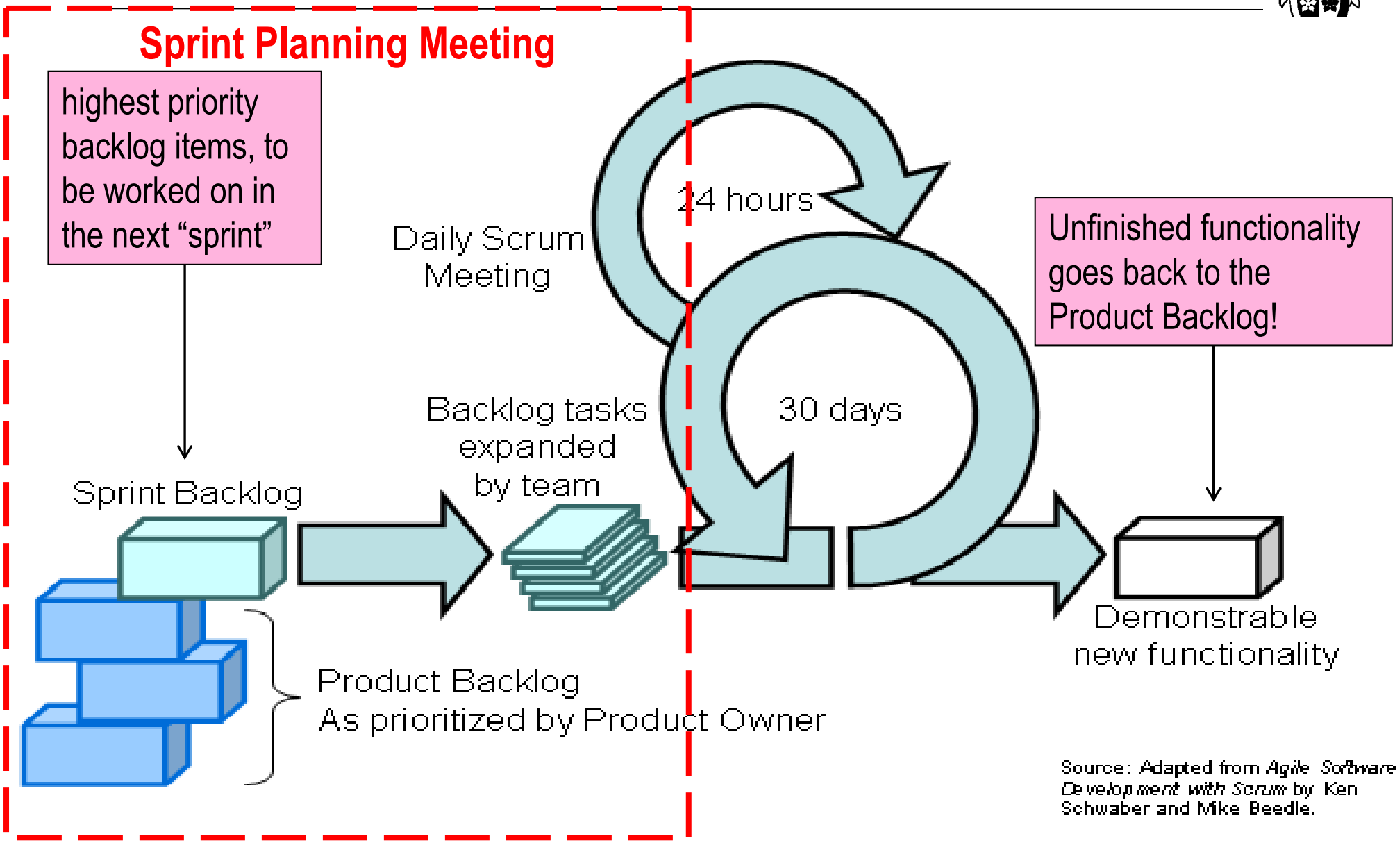


- Sprint Planning Meeting
- Product Backlog
- Sprint Backlog
- Selecting Sprint Backlog Item

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.

Scrum – The Process – Recap



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

Product Backlog – Recap

- 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.

Product Backlog – Another view

- A collection of items **to be done** by the team to complete the project
  - Each item must satisfy “definition of done” in order to be useful to the client
- Backlog items – things put into a “bag” for future consideration

Product Backlog – Related Issues

- Who writes the product backlog items?
  - Product owner? Developers? Scrum Master?
- Under what conditions a software project is good and can be terminated?
  - All product backlog items are done?

Sprint Backlog

- A collection of items **to be done** by the team in each sprint
  - Each item must satisfy “definition of done” in order to be useful to the client
  - When the client is OK, the item will be “integrated” into the main software for client to use

Sprint Backlog – Related Issues

- Who selects the sprint backlog items? How to select?
  - Product owner? Developers? Scrum Master?
- What if the sprint backlog item is done badly or it does not satisfy the definition of done at the end of a sprint?
  - Put back as a sprint backlog item for next sprint?

Sprint Planning Meeting

- Done in Day 1 of a sprint
  - Usually 4 hours for a 2-week sprint
- The development team discusses what should be developed in the “current” sprint
  - The jargon here usually is the “next” sprint, but I prefer it to be the “current” sprint
- Focus is on selecting the items in the product backlog to be developed in the “current” sprint
  - Product backlog item → Sprint backlog item

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 in 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.

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

SWINBURNE UNIVERSITY OF TECHNOLOGY

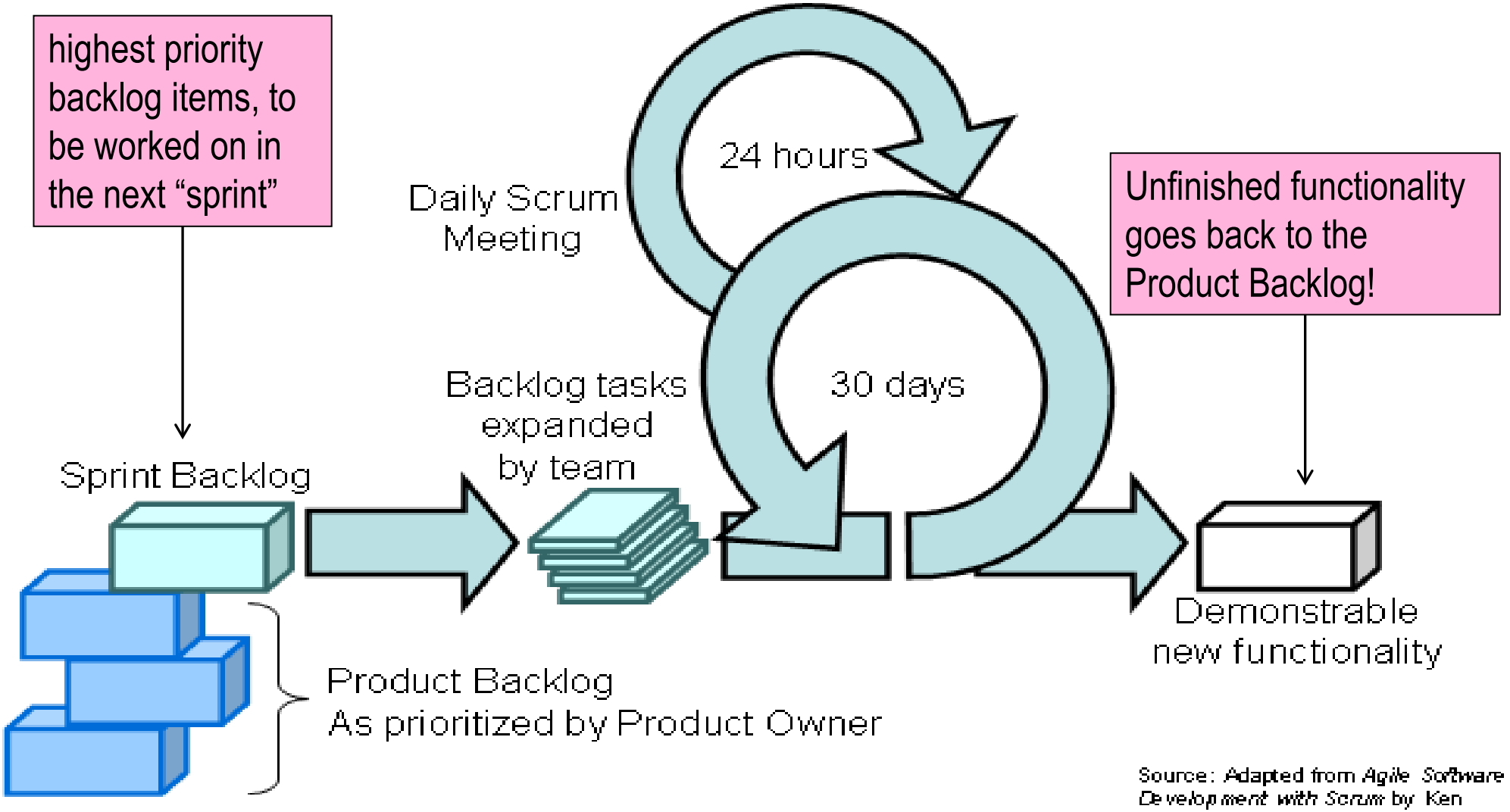
Getting Sprint Backlog Item

- WBS (Estimating)

Sprint Backlog – What is it?

- A list of **items*** that are required to be done during the sprint
- **Item** = Item + its tasks (breakdown tasks) + estimated time for each task
- Determined by the Scrum team during the Sprint Planning Meeting

Scrum – The Process (Recap)



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

Items in Sprint Backlog – Where from?

- From those items in the Product Backlog
- Developers to “discuss” with Product Owner to decide whether a particular product backlog is good for the “next” sprint

Items in Sprint Backlog – How to?

- Team members (Developers and Product Owner) to
  - Think about what to do with the item
  - Ask questions about the item so as to collect enough information to develop the item
  - Determine whether the item can be completed in the "*next*" sprint
  - Break down the item into smaller tasks
  - (for each task) Estimate the time required (efforts) to complete the task
  - If "total efforts required > a sprint", break the item down to smaller pieces so that it can be completed in one sprint
  - If "total efforts required < a sprint", fit "several items" into one sprint

Paint Your Bedroom Example (Recap – Lec1)

- Scope
  - ☐ An empty room
  - ☐ No holes to patch
  - ☐ 4 Walls – same colour [what colour?]
  - ☐ No doors
  - ☐ No trims
  - ☐ Ceiling – different colour from walls [what colour?]
  - ☐ Primer (Undercoat) + 3 coats of paint

Paint Your Bedroom Ex. (Recap – Lec1 cont'd)

- Product Backlog 1
  - ☐ Get Tools
  - ☐ Determine the colour of Walls and Ceiling
  - ☐ Get Paint
  - ☐ Paint the Walls
  - ☐ Paint the Ceiling
- Product Backlog 2
  - ☐ Get Tools
  - ☐ Get Undercoat for Walls
  - ☐ Get Undercoat for Ceiling
  - ☐ Get Paint for Walls
  - ☐ Get Paint for Ceiling
  - ☐ Get Masking Tapes
  - ☐ Paint Undercoat
  - ☐ Paint First Coat
  - ☐ Paint Second Coat
  - ☐ Paint Third Coat

Example: Peer Review System – Sprint Backlog

- Item 1: Allow a student to submit their peer review assessments about their team members
  - ☐ Peer Review Form (?)
  - ☐ Online submission (via Web site?) / Submission via mobile apps (?)
  - ☐ One member per submission (?)
  - ☐ All team members in one submission (?)
  - ☐ Any other questions (?)

Example: Peer Review System – Sprint Backlog

- Item 1: Allow a student to submit their peer review assessments about their team members [Task breakdown via WBS]
  - ☐ T1: Design the form
  - ☐ T2: Program the form (Web ? / GUI ?)
  - ☐ T3: Design database table / schema for the peer review submission
  - ☐ T4: Program the module for submission (extract info and save to database)
  - ☐ T5: Design test cases for submission
  - ☐ T6: Test the correctness of the submission module
  - ☐ Any other tasks (?)
  - ☐ Any dependencies (?)

Example: Peer Review System – Sprint Backlog Item 1 + Tasks

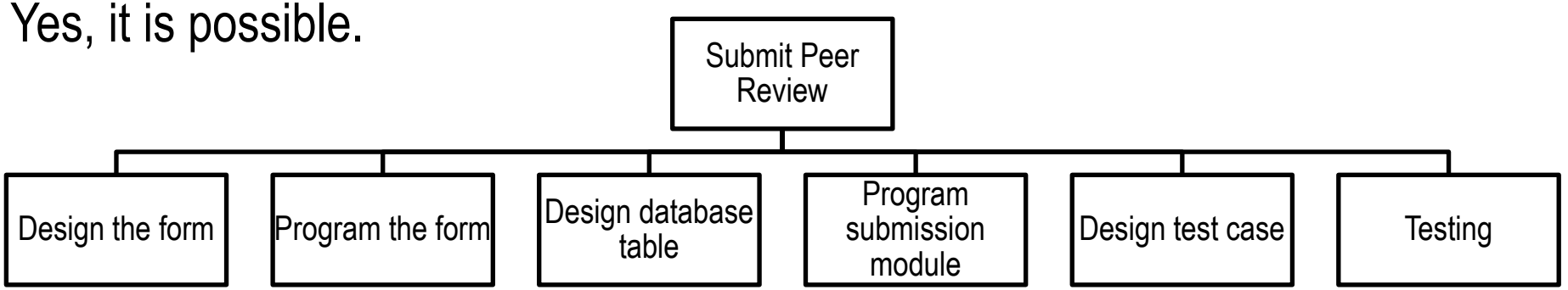
Task Id	Desc	Depends on	Duration (hrs)
T1	Design the form		1
T2	Program the form	T1	1
T3	Design database table / schema for the peer review submission	T1	1
T4	Program the module for submission (extract info and save to database)	T2, T3	3
T5	Design test case for submission		1
T6	Test the correctness of the submission module	T4, T5	1

Ex.: Peer Review System – Sprint Backlog Item 1 – WBS

Qn 1: Is this good enough?
Ans: Depends.

Qn 2: Can it be broken down further to even smaller tasks?
Ans: Yes, it is possible.

Qn 3: But how?
Ans: Let's try.



SWINBURNE UNIVERSITY OF TECHNOLOGY

Work Breakdown Structure

Planning for Software Development

- Split items into *tasks* or *activities* using a suitable “SDLC” as an anchor
- Create a *Work-Breakdown-Structure* (WBS)
  - breaks the project down into a set of well-defined, discrete tasks
- For each task or subtask, estimate the time for completion and assess resources required

Work Breakdown Structure, WBS

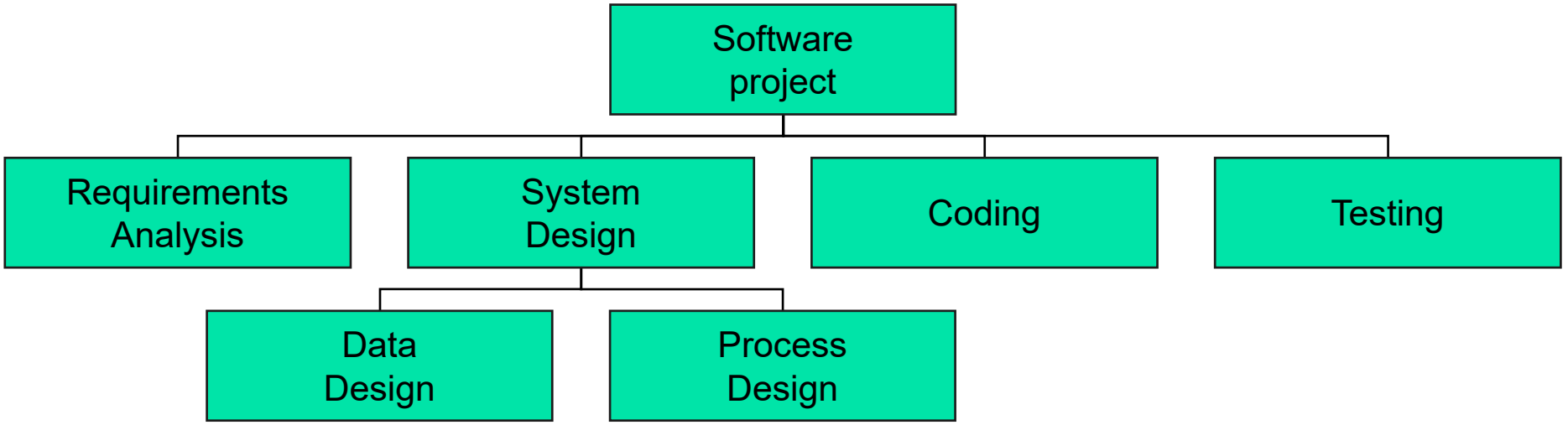
- An *outcome-oriented analysis* of the work involved
- Aim: To break the work required into smaller and more manageable pieces
- Different approaches to generate a WBS
  1. Activity-based approach (focus on the different things to be done)
  2. Product-based approach (focus on the different things to be produced)
  3. *A Hybrid approach* (focus first on the different things to be produced, then for each of these, focus on the things to be done)

Activity-based Approach

- The decomposition is based on activities to be undertaken
- This involves the following steps:
  - Identify the ***main activities*** of the project
  - Break each main activity into sub-activities
  - Continue to divide each sub-activity into lower level activities until the activities *can be finished with acceptable levels of effort*

☞ The chosen software development lifecycle model should give a good sense of the top level breakdown*: analysis, detailed design, implementation, testing at some appropriate level of granularity*

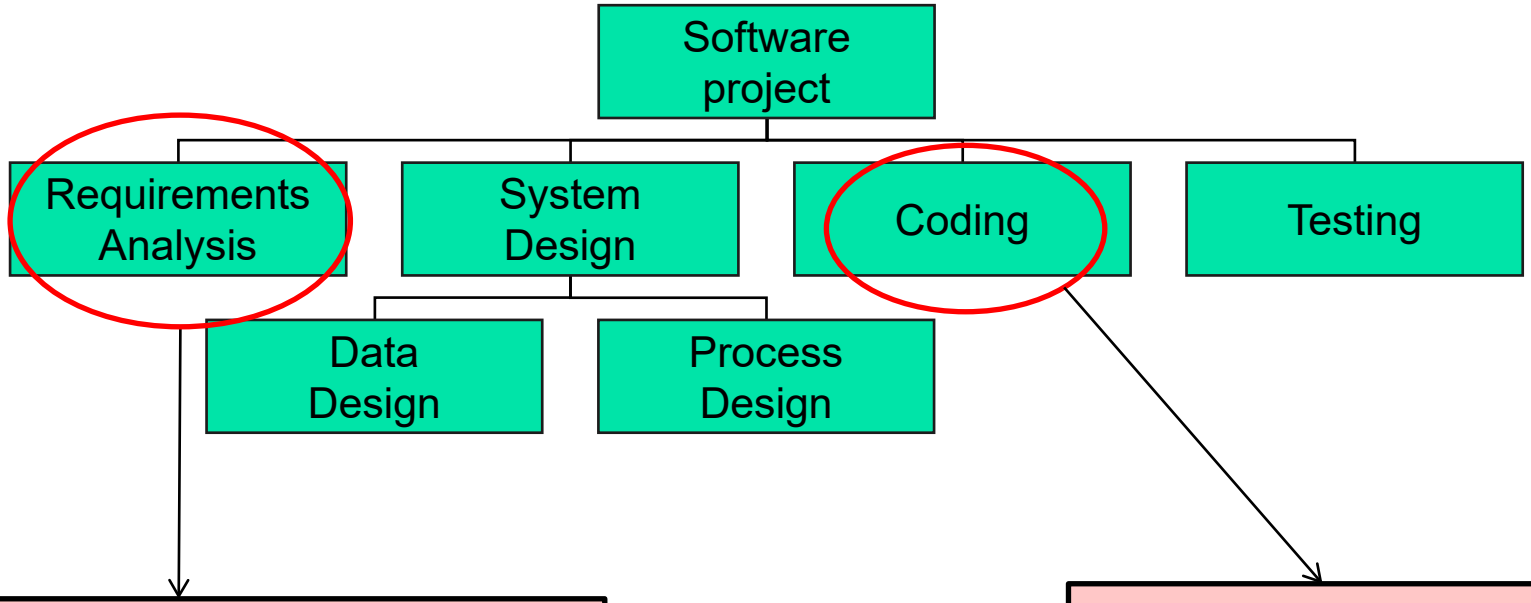
Activity-based Approach – Example



This is a very generic decomposition at a high level, applicable to many projects using a waterfall approach.

See Lecture 1a SDLC for those major steps

Activity-based Approach – Example



Plausibly some coding could start before all rqts have been determined. So we may have task dependencies between subtasks of different "main tasks" – pure waterfall would not have this!

Decompose this (and other top level "tasks") into subtasks that relate to coding of various modules, where there may be some dependencies

Activity-based Approach (cont.)

- Advantages:
  - It is more likely to obtain a structure that is complete and is composed of non-overlapping activities
  - The structure can be refined as the project proceeds
  - The structure already suggests the dependencies among the activities/tasks
  - The structure can be readily used as a basis for project scheduling
  - The structure is easy to understand and can be used to communicate with project stakeholders
- Disadvantage:
  - *It is likely to miss some of the products/deliverables to be produced!*

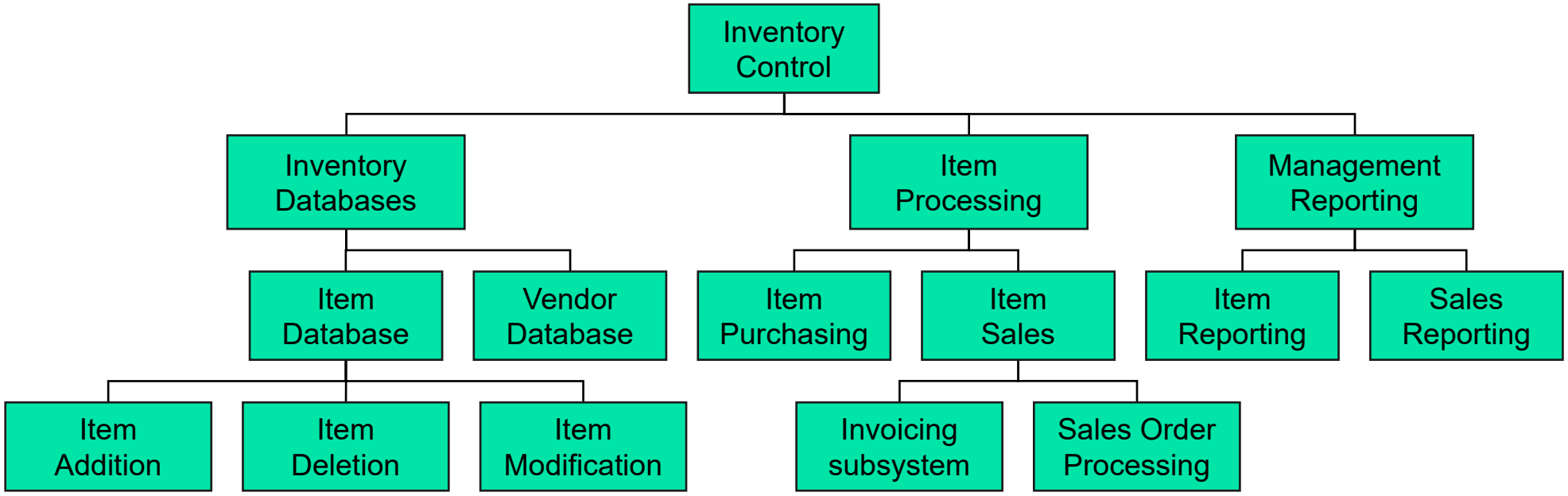
Common Issues in WBS

- If there are too many levels in WBS,
  there will be a large number of small tasks
- If there are too few levels (the WBS is too shallow),
  the details for project control will be insufficient
- Ideally, each leaf (the lowest level work) of a WBS can be finished by an individual team member within several hours of work
- The actual durations appropriate for individual tasks depend from project to project
- Getting it right is a challenge!!

Product-based Approach

- The decomposition is based on the products or deliverables to be produced
  - Examples: SRS, SDD, Source, STP, STD, User Manual, …
- Also called *Product Breakdown Structure* (PBS)
- Product Flow Diagram (PFD)
  - To indicate, for each product, which other products are required as ‘inputs’

Product-based Approach – Example



Danger is that dependencies between products is missed

Product-based Approach (cont.)

- Advantage
  - It is less likely to miss a product which is expected from the structure.
  - Good for agile projects – aim at delivering subsystems at the end of iterations
- Disadvantage
  - The activities or tasks used to create a product are not specified and may be missed, and some may be distributed amongst several products.

A Hybrid Approach

- More commonly used approach
  - A mix of activity-based approach and product-based approach
- The WBS consists of
  - a list of the products of the project; and
  - a list of activities for each product

  NB : There may be some cross-product activity dependencies

A Hybrid Approach – Example

- MITP methodology by IBM
  (Managing the Implementation of the Total Project)
  {which partly inspired PRINCE2}
  - Level 1: Project
  - Level 2: Deliverables (software, manuals etc)
  - Level 3: Components of each deliverable
  - Level 4: Work-packages
  - Level 5: Tasks (individual responsibility)

SWINBURNE UNIVERSITY OF TECHNOLOGY

Estimating Part

Planning

- Split project into *tasks* or *activities* using the chosen SDLC as an anchor
- Create a *Work-Breakdown-Structure* (WBS)
  - breaks the project down into a set of well-defined, discrete tasks
- For each task or subtask, **estimate** the time for completion and assess resources required

What to estimate?

Ultimate goal

Assign a duration (usually, time expressed in working days or hours) to each problem / task / outcome identified in a work breakdown structure (WBS)

What to estimate?

Normal approach:

1. First estimate size and *complexity* of a given problem, task, or outcome
2. Then use this to estimate the effort / time required for the task

- Note that duration depends also on number of people available to do the work
- Also note that estimation is hard, and generally, for software development, is not done very effectively!

The meaning of Time and Effort

- People cannot work 100% productively, 100% of their available time:
  - Need to consider interruptions, socializing, email etc.

  - Rule of thumb: productivity of IT people ~70%
    - eg, for a 40-hour work week, assume 28 hours of productive work
    - Varies from person to person
    - FACT (borne out by serious research): productivity decreases as total hours worked per week increases above about 40

The meaning of Time and Effort – Definitions

- Ideal Time
  - fully productive time on given task without interruptions
- Ideal Effort
  - amount of ideal time it takes to complete a task
- Real Effort
  - Amount of real time taken to complete a task

The meaning of Time and Effort : Example

- If it would take 20 hours of **ideal time** to write a user manual*, then assume it will take **???** of real time

- *How do we obtain the ideal time estimate?

The meaning of Time and Effort : Example

- If it would take 20 hours of **ideal time** to write a user manual*, then assume it will take **???** of real time

- *How do we obtain the ideal time estimate?
  - Past experience
  - Measurement of actual time on a small task, multiplied to give estimate for full task
  - Measurement of task according to a reasonable size estimate
  - Magic??!!

Example

- I can mark 1 exam paper in 10 minutes
- I have 100 papers to mark
- How long will it take?

Example

- I can mark 1 exam paper in 10 minutes, on average
- I have 100 papers to mark
- How long will it take?

- Simple answer is 1000 mins = 16 hrs and 40 mins
- But I can't keep up the rate of 1 paper every 10 mins
- Although I will probably only ever take 10 mins to mark a paper, over a day's work I will have time spent away from the direct task – probably 30% of my time {estimated from past experience}
- So, in reality, it will take me approx. 24 hrs (= 1000 / 0.7 / 60) to complete the task

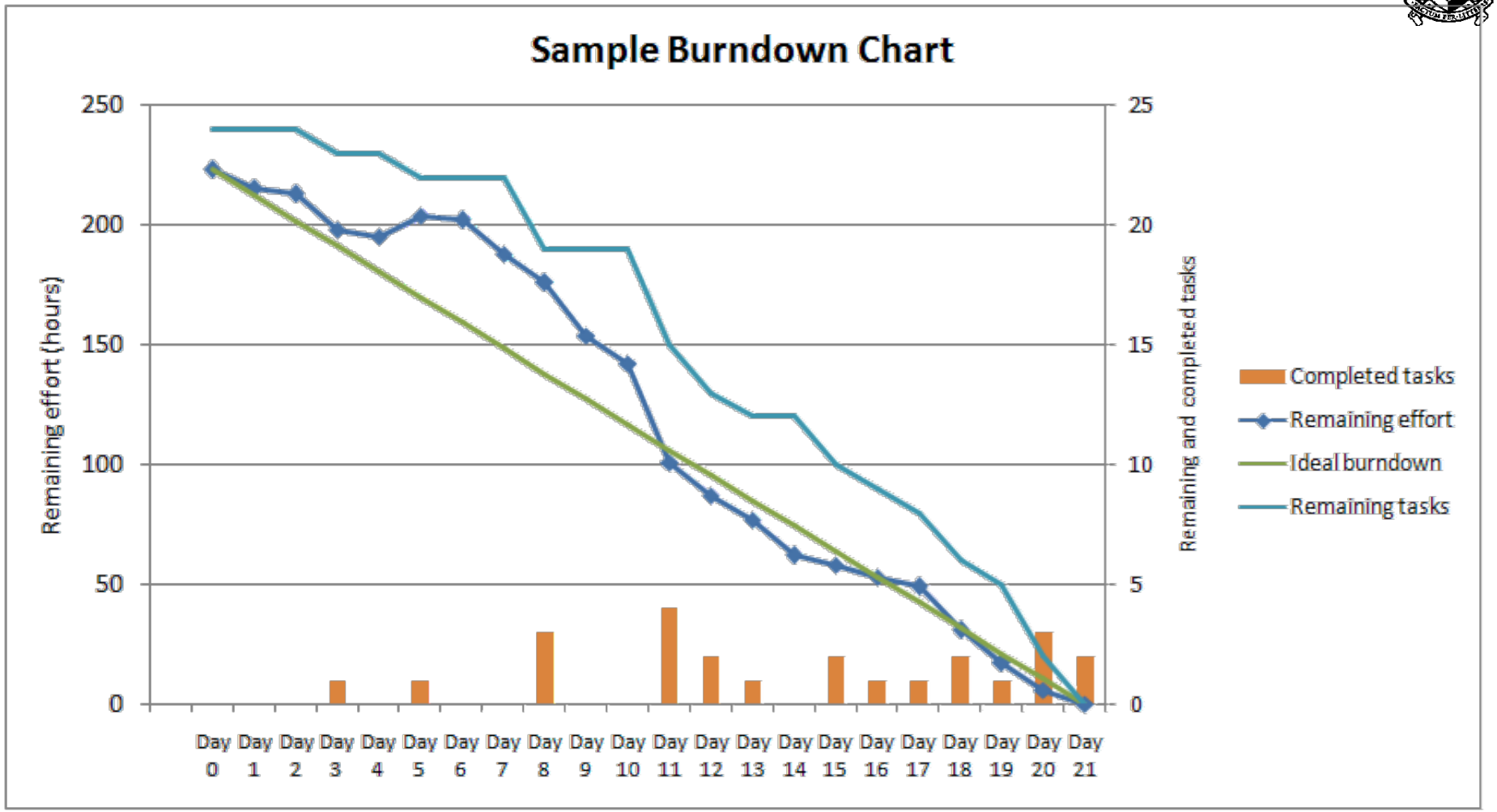
SWINBURNE UNIVERSITY OF TECHNOLOGY

Burndown chart

Burndown Chart – What to do?

- On a daily basis, each team member needs to report the remaining efforts (number of hours) to complete the task that they have signed off for “doing”
- Once a task is completed, the remaining efforts is 0 hours
- If a task is incomplete, the **remaining** efforts is the number of hours still needed to do the task at hand
  - This again is a guess from the team member
  - The number may be bigger than the original estimated value (meaning that the team under-estimated the level of difficulty / complexity of that particular task)

Burndown Chart – An Example



From https://upload.wikimedia.org/wikipedia/commons/0/05/SampleBurndownChart.png

http://en.wikipedia.org/wiki/Fair_use