CSC3045 & CSC3052

The Agile Modules

Scrum Project Management

School of Electronics, Electrical Engineering & Computer Science

Queen's University, Belfast

Dr Darryl Stewart

What does **SCRUM** stand for?

Scrum in a nutshell

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.
- It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).
- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.
- Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

Scrum has been used by:

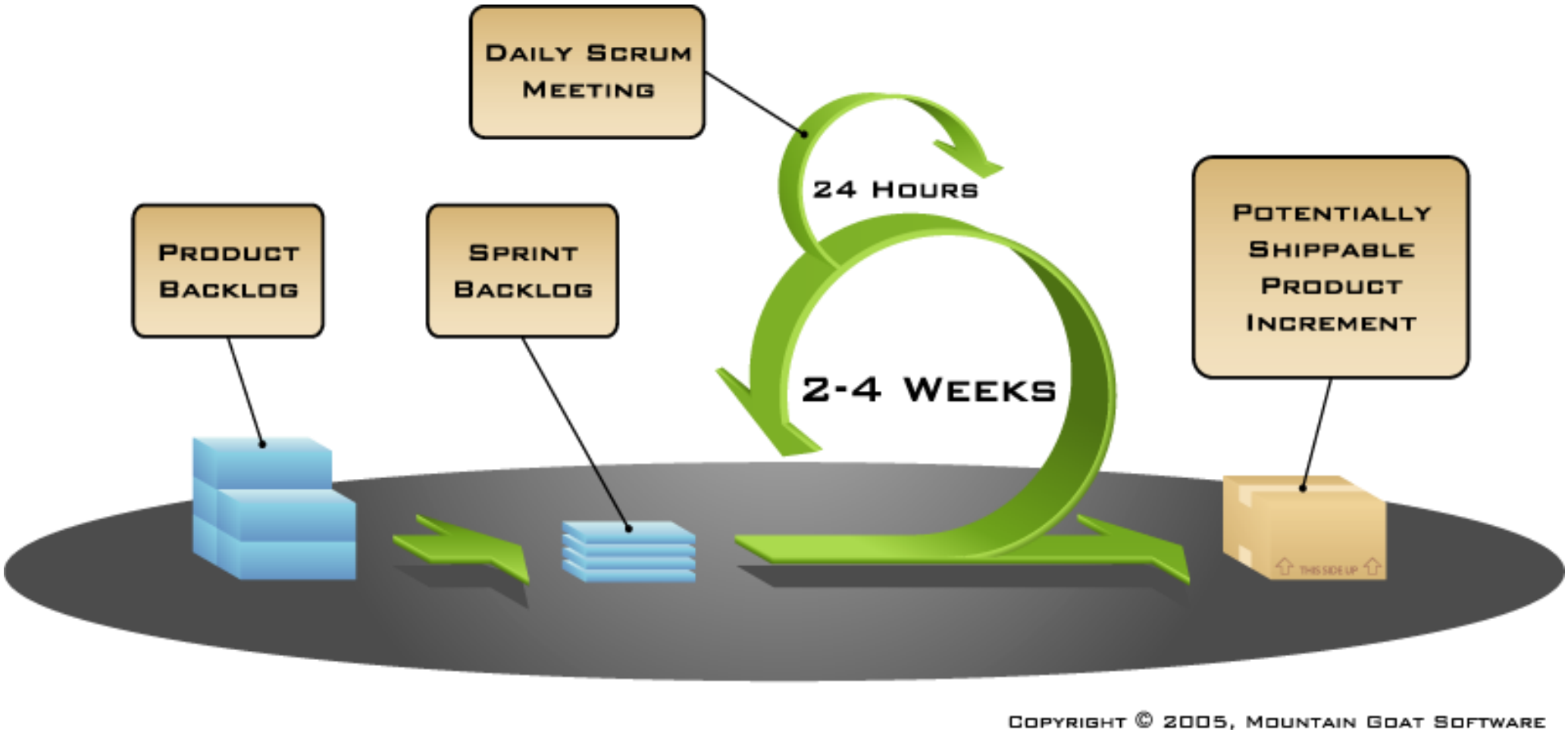
- Microsoft
- Yahoo
- Google
- Electronic Arts
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- First American Real Estate
- Time Warner
- Oce
- BT

Scrum has been used for:

- Commercial software
- In-house development
- Contract development
- Fixed-price projects
- Financial applications
- ISO 9001-certified applications
- Embedded systems
- 24x7 systems with 99.999% uptime requirements
- The Joint Strike Fighter

- Game development
- Life-critical systems
- Satellite-control software
- Websites
- Handheld software
- Mobile phones
- Network switching applications
- Some of the largest applications in use

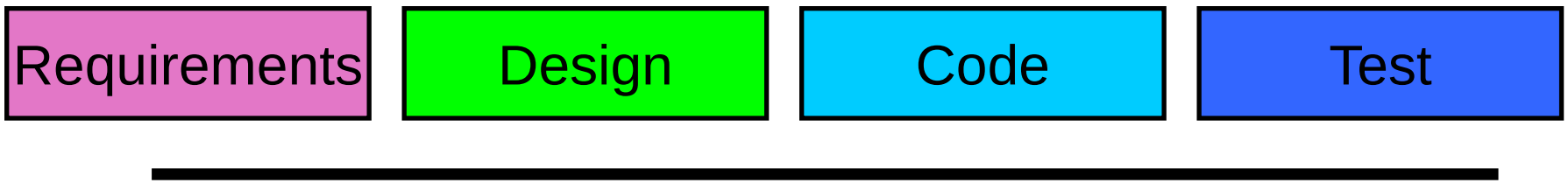
What Scrum looks like



Sprints

- Scrum projects make progress in a series of iterations called "sprints"
- Typical duration is 2–4 weeks or a calendar month at most
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint – "**done done**"

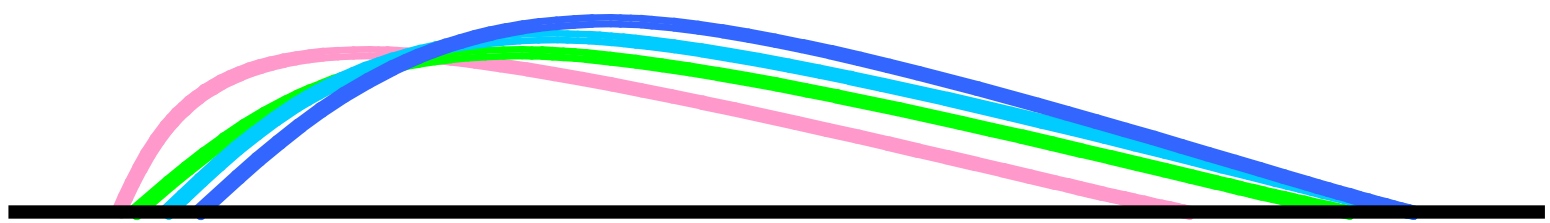
Sequential *vs* Overlapping development



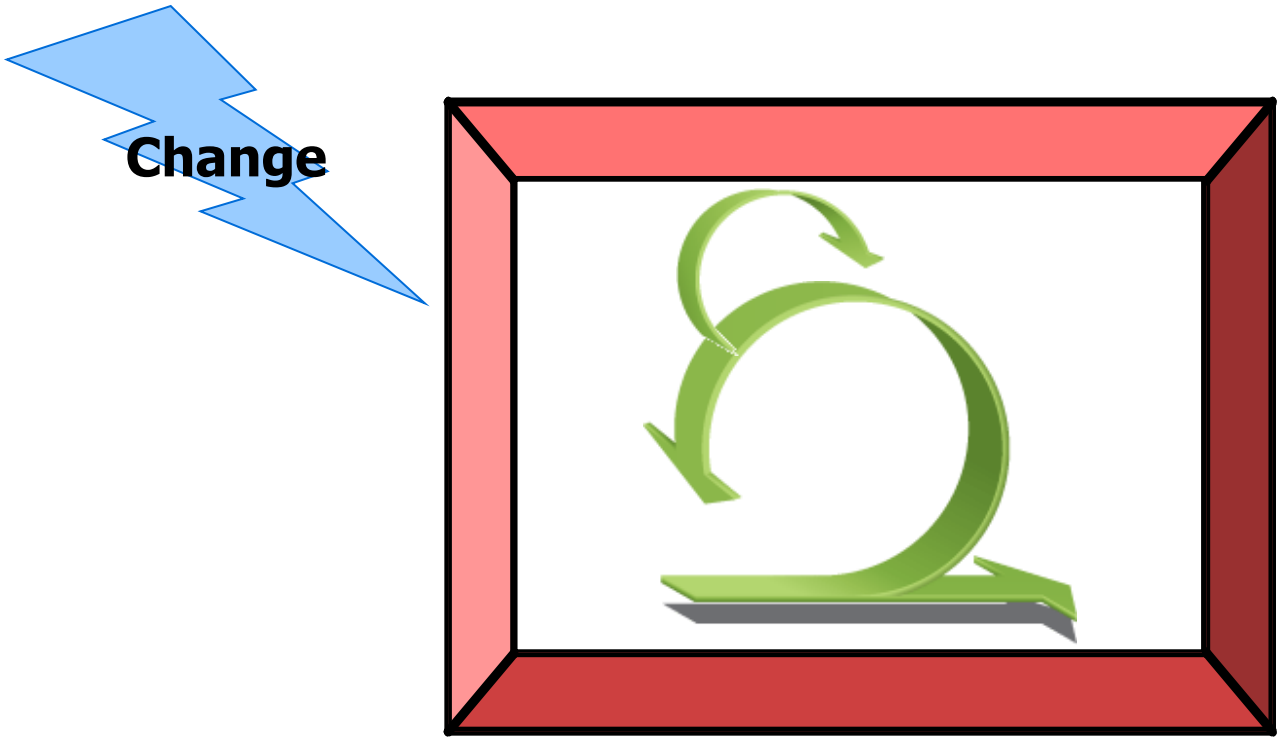
Rather than doing all of one thing at a time...

...Scrum teams do a little of everything all the time

Source: “The New New Product Development Game” by Takeuchi and Nonaka. *Harvard Business Review,* January 1986.

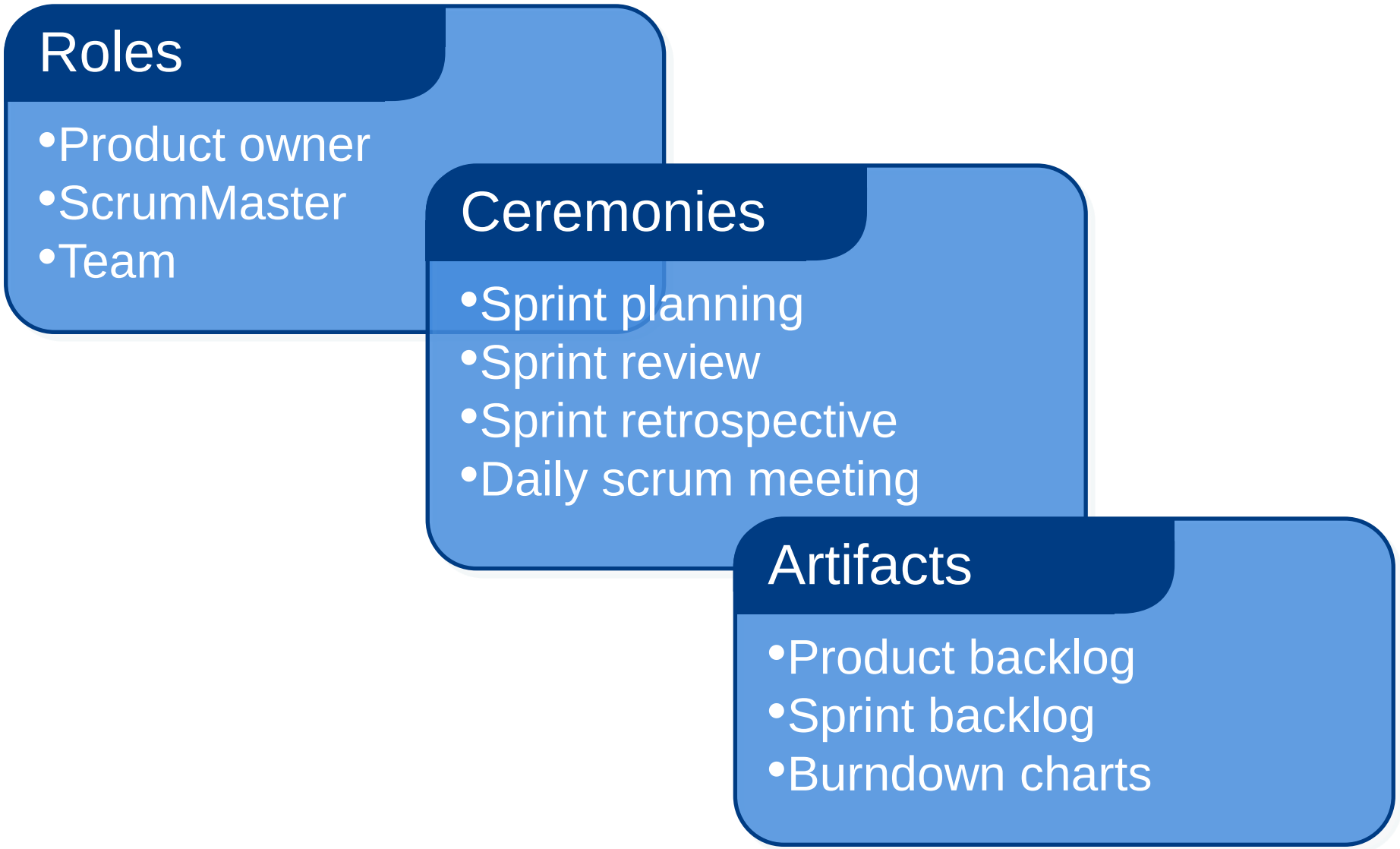


No changes during a sprint



- Plan sprint durations around how long you can commit to keeping change out of the sprint
- In exceptional circumstances a Sprint can be aborted
  - **When would this happen?**

Scrum framework

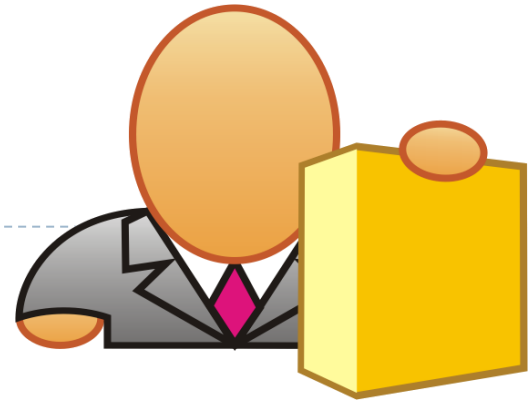


Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

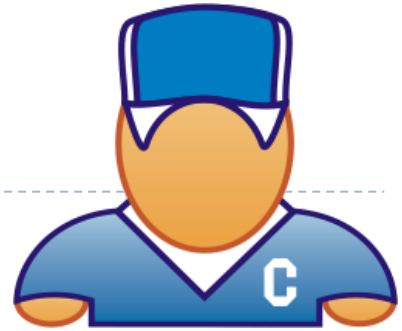
Product owner



- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every sprint, as needed
- Accept or reject work results
- Decide if increment will be released

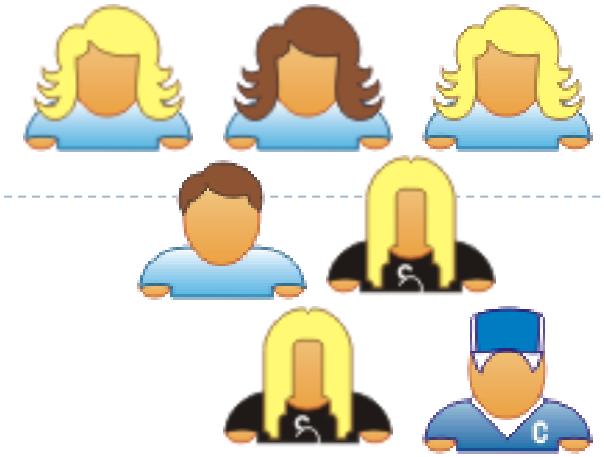
(In your projects Phil or I take this role)

The Scrum Master



- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences
- **In your project this role will be rotated every 2 days through the team until all have experienced it**
- **Then the team decides who will be SM for the remainder of the project**

The Scrum Team

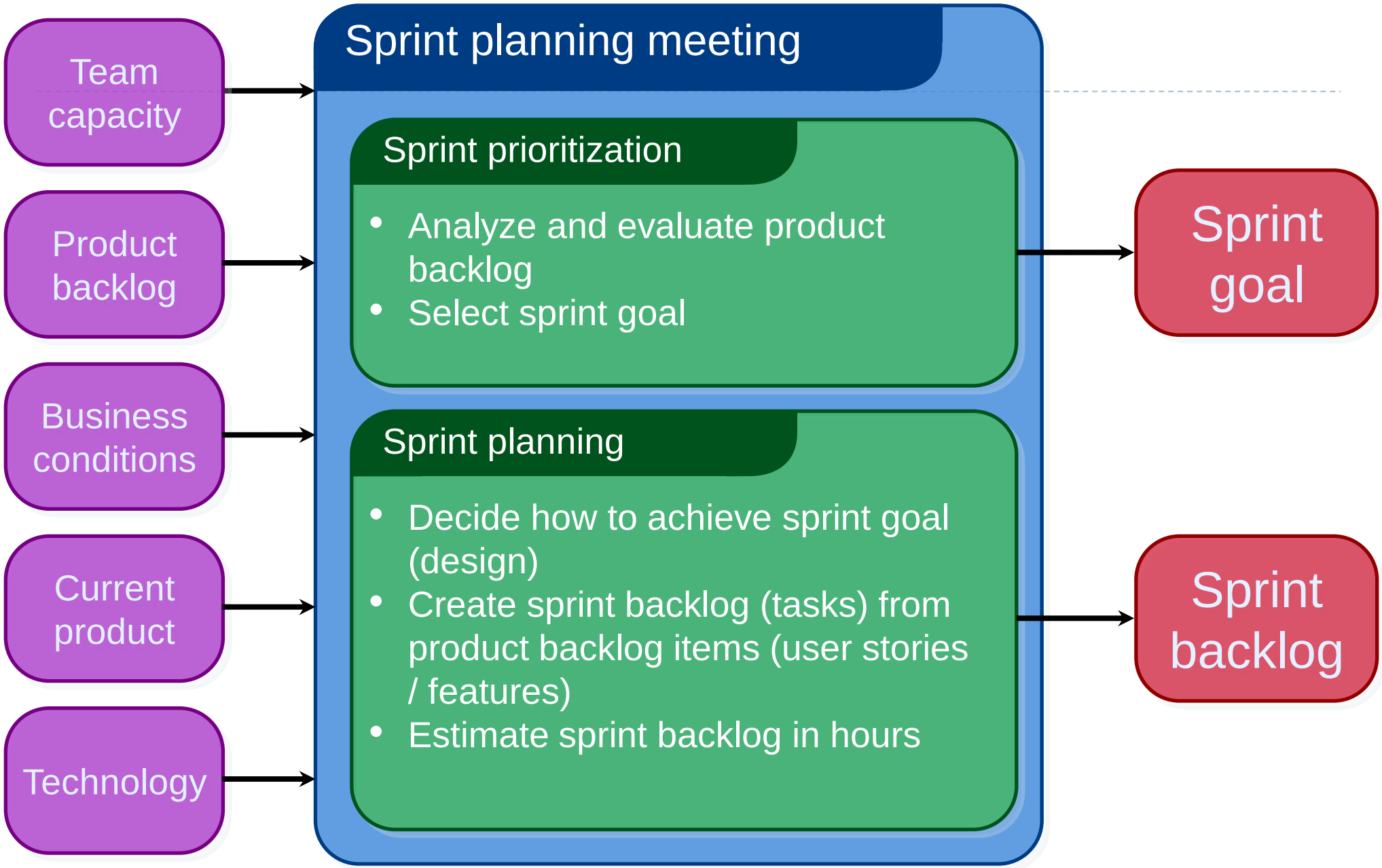


- Book says 7±2 people.
  - More recently the guidance is 3 to 9
  - SM and PO only included in this if they are actively developing
- Cross-functional: Programmers, testers, user interface designers, etc.
  – All the skills needed to deliver the next sprint goal
- Members should be full-time (according to the book)
  - Can be exceptions (e.g. database administrator might be shared across teams)
- Teams are self-organizing – decide how they will do the work
- Membership should change only between sprints according to next sprint goal – expect slight impact on productivity

Scrum framework

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting


Team capacity
Product backlog
Business conditions
Current product
Technology
Sprint planning meeting
Sprint prioritization
Analyze and evaluate product backlog
Select sprint goal
Sprint planning
Decide how to achieve sprint goal (design)
Create sprint backlog (tasks) from product backlog items (user stories / features)
Estimate sprint backlog in hours
Sprint goal
Sprint backlog

Sprint planning

- Time boxed to max 8 hours for one month sprint, less than this for shorter sprints
- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
  - Tasks are identified and each is estimated (1-7 hours)
  - Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

As a holiday planner, I want to see photos of the hotels.

Code the middle tier (7 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

The daily scrum

- Daily, 15-minutes (time boxed!), Stand-up
- Not for problem solving
- Whole world is invited but only the Scrum Team must attend
  - **BUT only Scrum Team, Scrum Master and Product Owner can talk**
  - These are called the **Pigs** others are **Chickens**

By Clark & Vizdos

- Helps avoid other unnecessary meetings

The Sprint Review

- Scrum Team presents what it accomplished during the sprint
- Typically a demo of new features or underlying architecture (yours will be pre-prepared as a set of videos)
  - Informal
    - 2-hour prep time rule (time boxed!)
    - 4-hour meeting (time boxed!) for a one-month Sprint. For shorter Sprints, the event is usually shorter.
    - Typically no slides – (we will be slightly unconventional with our videos)
  - Attendees: Scrum Team and any stakeholders invited by Product Owner

The Sprint Review

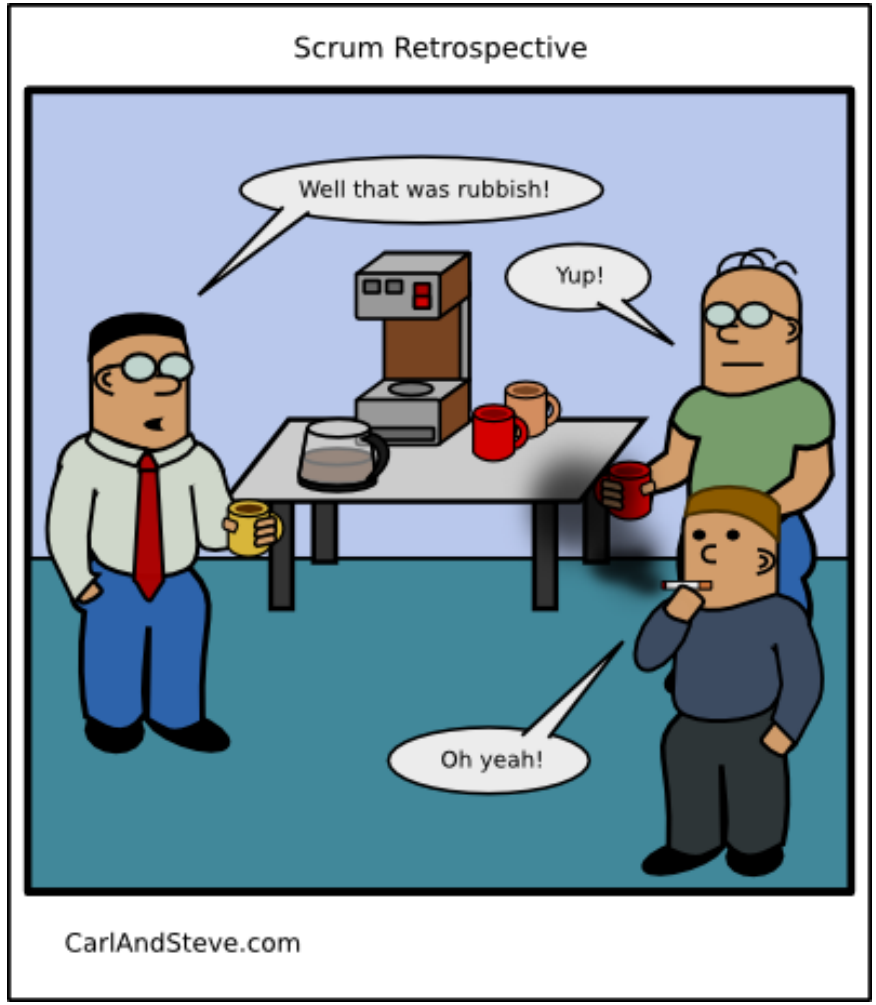
- Scrum Team discusses what went well, problems it ran into, and how problems were solved
- Scrum Team demonstrates "Done Done" work and answers questions
- Product Owner provides feedback – expect some constructive criticism and change requests – embrace change!
- Product Owner discusses current Product Backlog and may project likely **completion dates** based on progress to date
- Entire group collaborates on what to do next – giving valuable input to subsequent Sprint Planning

Sprint Retrospective

- Look at what *is* and *is not* working
- Occurs after every Sprint Review and before Sprint Planning
- 3-hour (time boxed!) meeting for one-month Sprints – less for shorter sprints
- Whole team participates
  - ScrumMaster
  - Product owner
  - Team
  - Possibly customers and others

Sprint Retrospective

- The purpose is to:
  1. Inspect how the Sprint went with regards to people, relationships, process, and tools
  2. Identify and order the major items that went well and potential improvements
  3. Create a plan for making improvements to the way the Scrum Team does its work
- Help the Scrum Team develop better processes and practices to make their work more effective and enjoyable for the next Sprint.
- Help the Scrum Team increase product quality by adapting the definition of "Done Done"



Scrum framework

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product Backlog

This is the product backlog

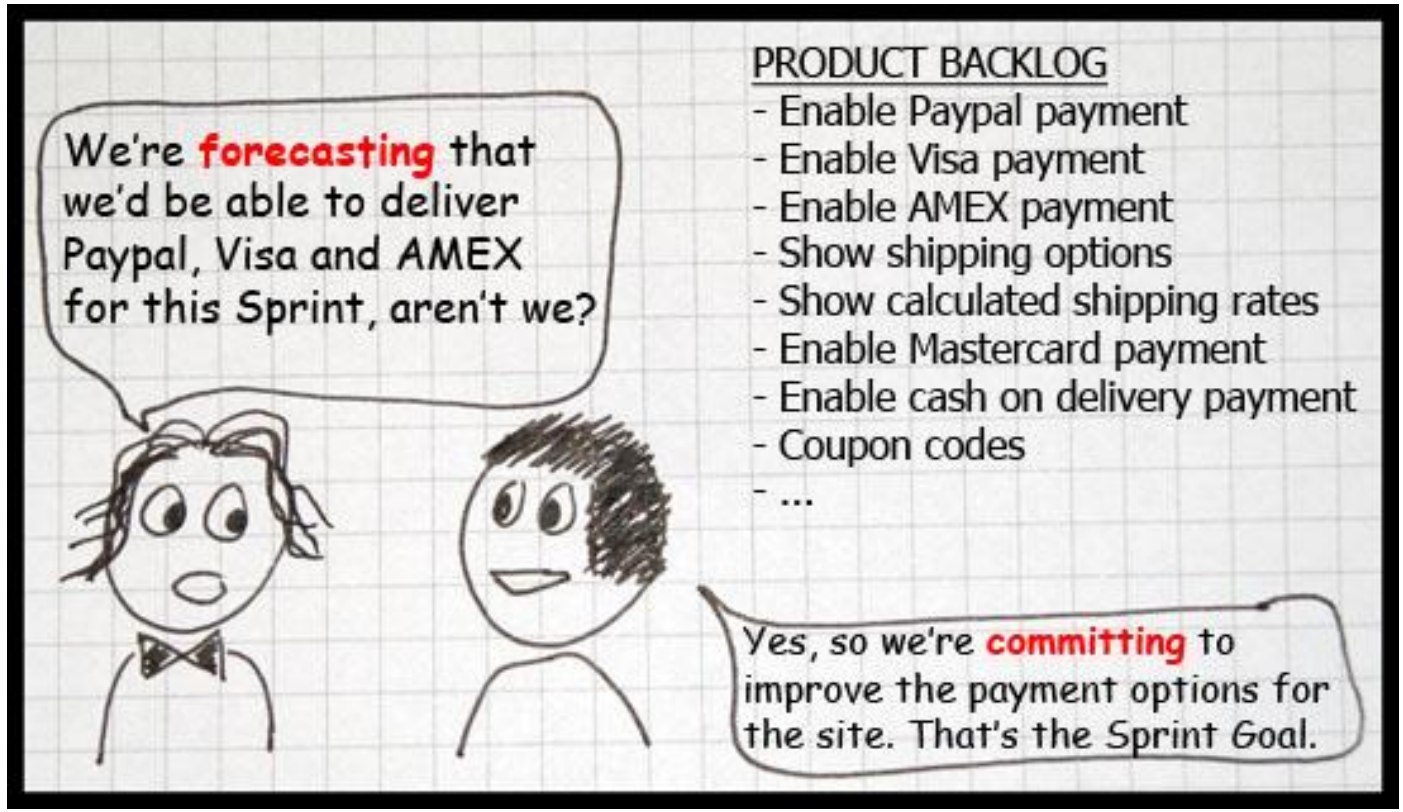
- The requirements
- A list of all desired work on the project that is currently known
- Ideally expressed such that each item has value to the users or customers of the product – this is often in the form of ***User Stories***
- Prioritized by the product owner
- **Reprioritized at the start of each sprint – HOW?**

A sample **product** backlog

Backlog item	Estimate
As a guest I can make a reservation so that…	3
As a guest I can cancel a reservation so that...	5
As a guest I can change the dates of a reservation so that…	3
As a hotel employee I can run RevPAR reports (revenue-per-available-room) so that we can…	8
…	8
...	30
...	50

The sprint **goal**

- A short statement of what the work will be focused on during the sprint – the unifying theme of the work to be done



- Used to assess if the sprint has been successful

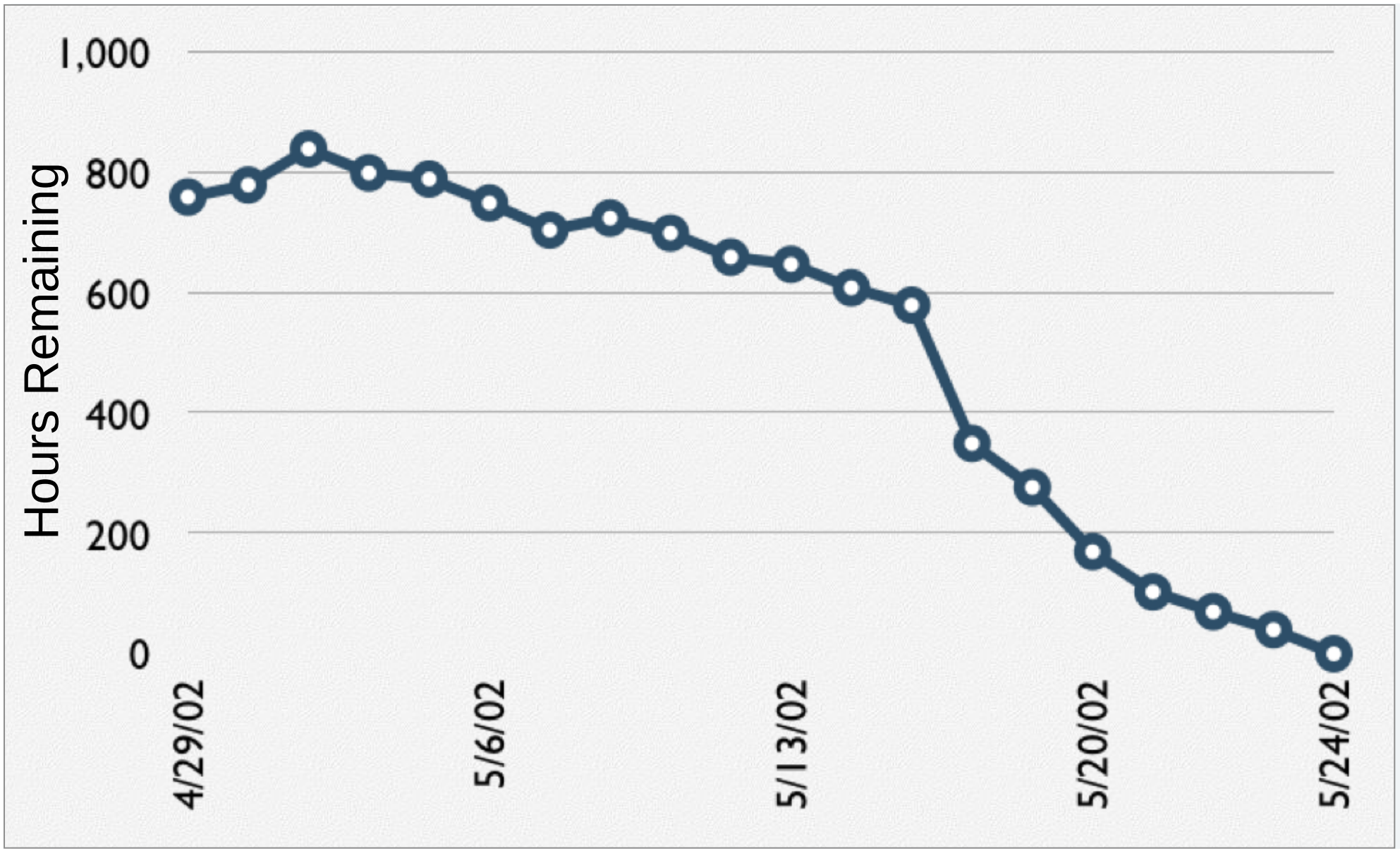
Managing the **sprint** backlog

- Individuals sign up for work of their own choosing
  - **Work is never assigned**
- Estimated work **remaining** is updated **daily** - Why not work **done**?
- Any team member can add, delete or change the **sprint** backlog
- Work for the sprint **emerges**
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work **remaining** as more becomes known
- **Keep only a small number of active tasks and stories**, i.e. finish what you start before starting something new

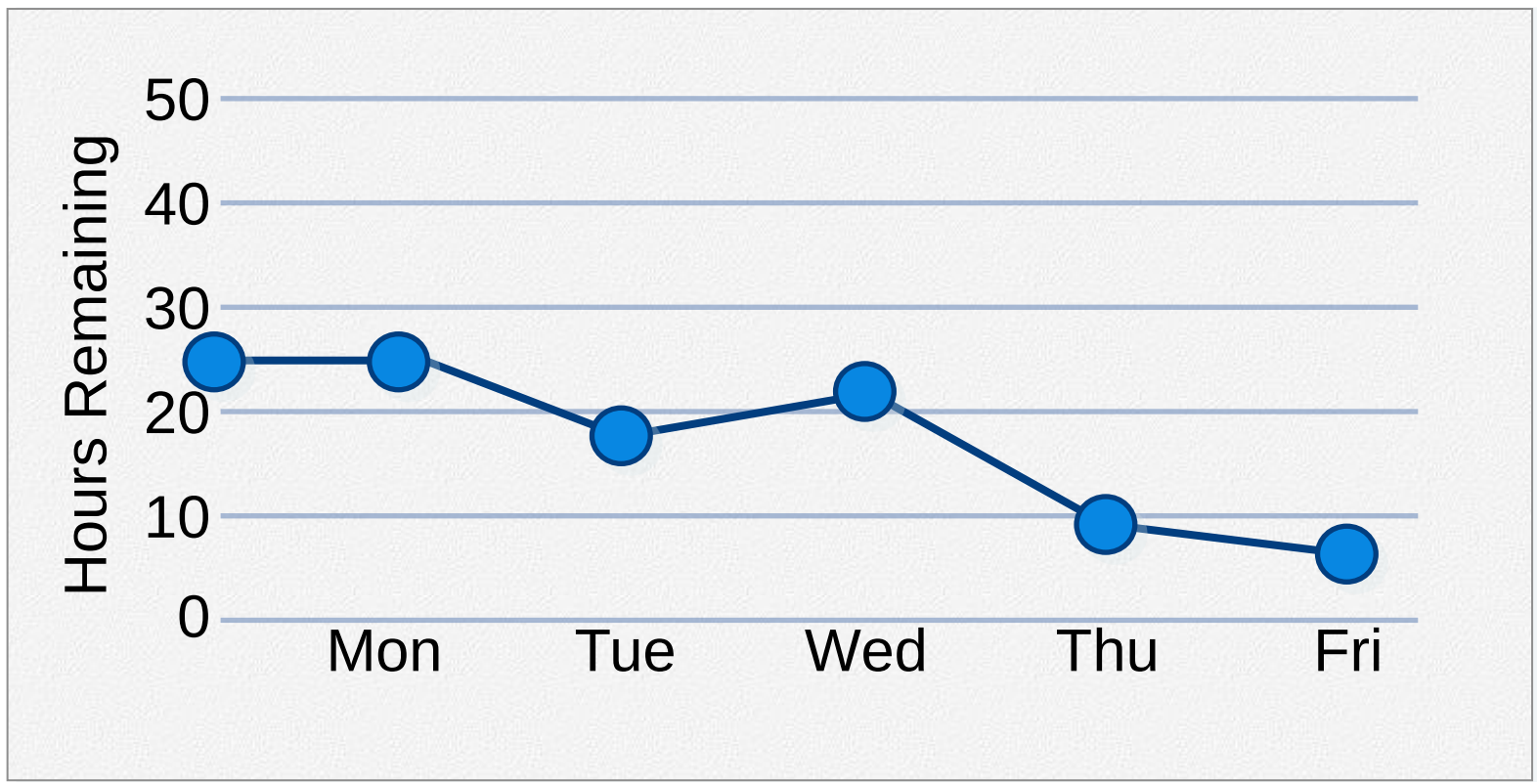
A **sprint** backlog

Tasks	Est.	Mon	Tues	Wed	Thur	Fri
Code the user interface	6	4	6	4		
Code the middle tier	5	7	5	6	2	
Test the middle tier	4	4	4	4	2	3
Write online help	6	6				
Write the foo class	4	4	4	4	4	4
Add error logging	0			4	2	

A sprint burndown chart

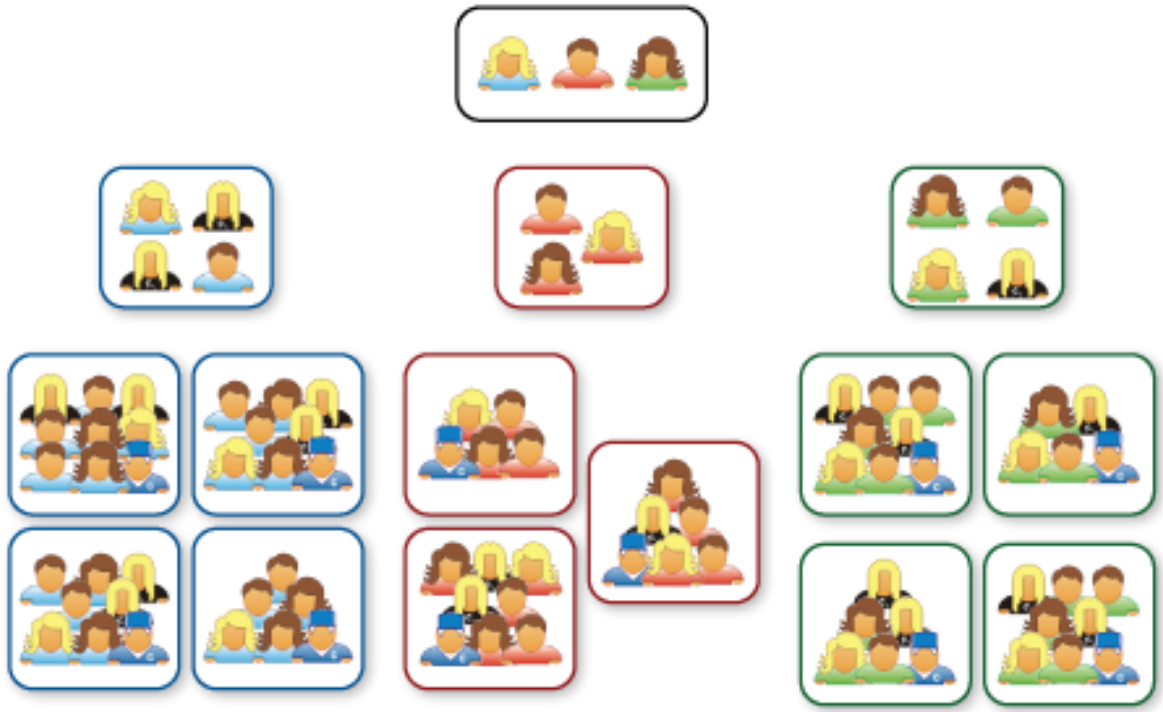


Tasks	Est.	Mon	Tues	Wed	Thur	Fri
Code the user interface	6	4	6	4		
Code the middle tier	5	7	5	6	2	
Test the middle tier	4	4	4	4	2	3
Write online help	6	6				
Write the foo class	4	4	4	4	4	4
Add error logging	0			4	2	



Scalability

- Typical individual team is 3 to 9 people
  - Scalability comes from teams of teams known as "scrum of scrums" https://www.scrumalliance.org/community/articles/2007/may/advice-on-conducting-the-scrum-of-scrums-meeting
- Scrum has been used on multiple 500+ person projects



Attribution

- This presentation makes use of parts of a presentation provided by **Mike Cohn** of Mountain Goat Software
- Similar presentations can be found here:
- http://www.mountaingoatsoftware.com/presentation/30-an-overview-of-scrum

Some recommended Scrum resources

- *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle
- Lots of weekly articles at www.scrumalliance.org

- Ken Schwaber Google Tech Talk
- "The Scrum Guide" at Scrum.org

Take home messages

- Scrum is probably the most widely used agile project management method
- It involves several Roles, Ceremonies (time boxed!) , and Artifacts
- It involves small self organising teams
- The measure of success is the 'done done' functionality added in a sprint – "**User Stories** Burned Down"

- You must apply this process in your project and describe everything in your videos