

An Introduction to Agile Programming and Scrum

21. 11. 2011

Martin Hincá (martin@hinca.net)

Example project scenario “Cowboy”

- You are rushing to start coding
 - Formal design is left out, it's too expensive
 - Team members do whatever they feel is right
- After a time it gets tricky
 - Things don't behave exactly as expected
- Whole application becomes inconsistent
 - And a complete mess to maintain and extend

Example project scenario “Cowboy”

- You are rushing to start coding
 - Formal design is left out, it's too expensive
 - Team members do whatever they feel is right
- After a time it gets tricky
 - Things don't behave exactly as expected
- Whole application becomes inconsistent
 - And a complete mess to maintain and extend



Example project scenario “Waterfall”

- We have a great design prepared up front
 - All *foreseeable* eventualities considered
 - Difficult and costly
- Change requests start to arrive
- Unforeseen circumstances, tweaks, bugs
- Application ends up different than envisioned
 - Initial design overgrown and modified on the run
- In the end you might find yourself in a mess like before

What went wrong?

- No reasonable design evolving in 1st case
- Design can't live up to the changes in the 2nd

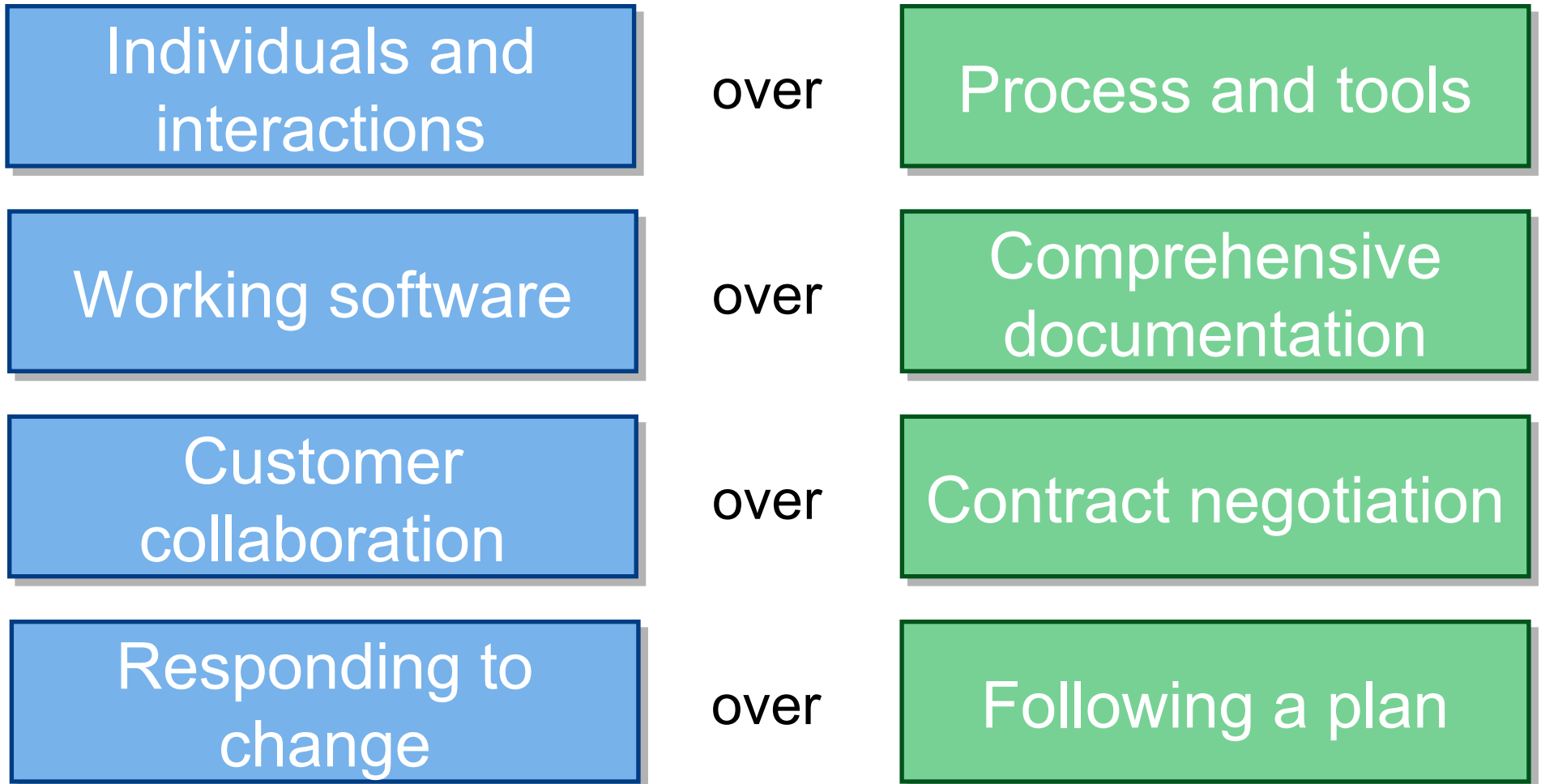
- Both cases are inflexible and may leave you stuck with
 - a huge pile of unmaintainable code,
 - difficulties to get back on track,
 - and a lot of headache... :-)

- Although of course № 2 is much more advisable

Agile programming

- Collection of principles to increase flexibility compared to strict design-based development
 - Adaptive rather than predictive
 - Flexible rather than binding
 - Uses relatively simple techniques
 - Repeated opportunities to assess project's direction
 - Doesn't require extreme excellence at design
 - Aimed at real-world situations and average teams
- It's possible to adapt it partially

The Agile Manifesto – a statement of values



Source: www.agilemanifesto.org

Core Agile Rules

- KISS!
 - Occam's razor of software development
 - Use the simplest design that meets your needs
- Design on the run
 - Solve design tasks as they emerge
 - Always keep the code base updated

Core Agile Rules (2)

- Incremental and Independent steps
 - When coding, prefer taking small steps and test the results as you go
 - Easier to find mistakes
 - Test-driven development
- Focus on one thing at a time
- Keep the code clean! (Refactoring)

Keep it Simple

- Planning overhead is expensive and inefficient
- Concentrate on the next step ahead and pick the most simple design you can
- Don't focus on things you may need later on
 - Your overview will become more complete over time, and chances are things will change
- Design must be evolving with the project
 - Evolution vs. Perfect creation :-)

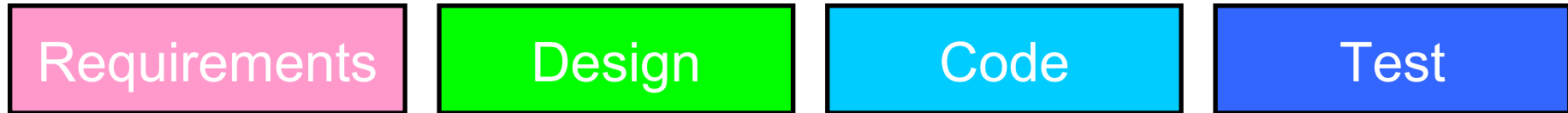
Agile Methodology Example:

SCRUM

Scrum in 100 words

- 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.

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.

Scrum has been used by:

- IBM
- Google
- Microsoft
- Yahoo
- Electronic Arts
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce

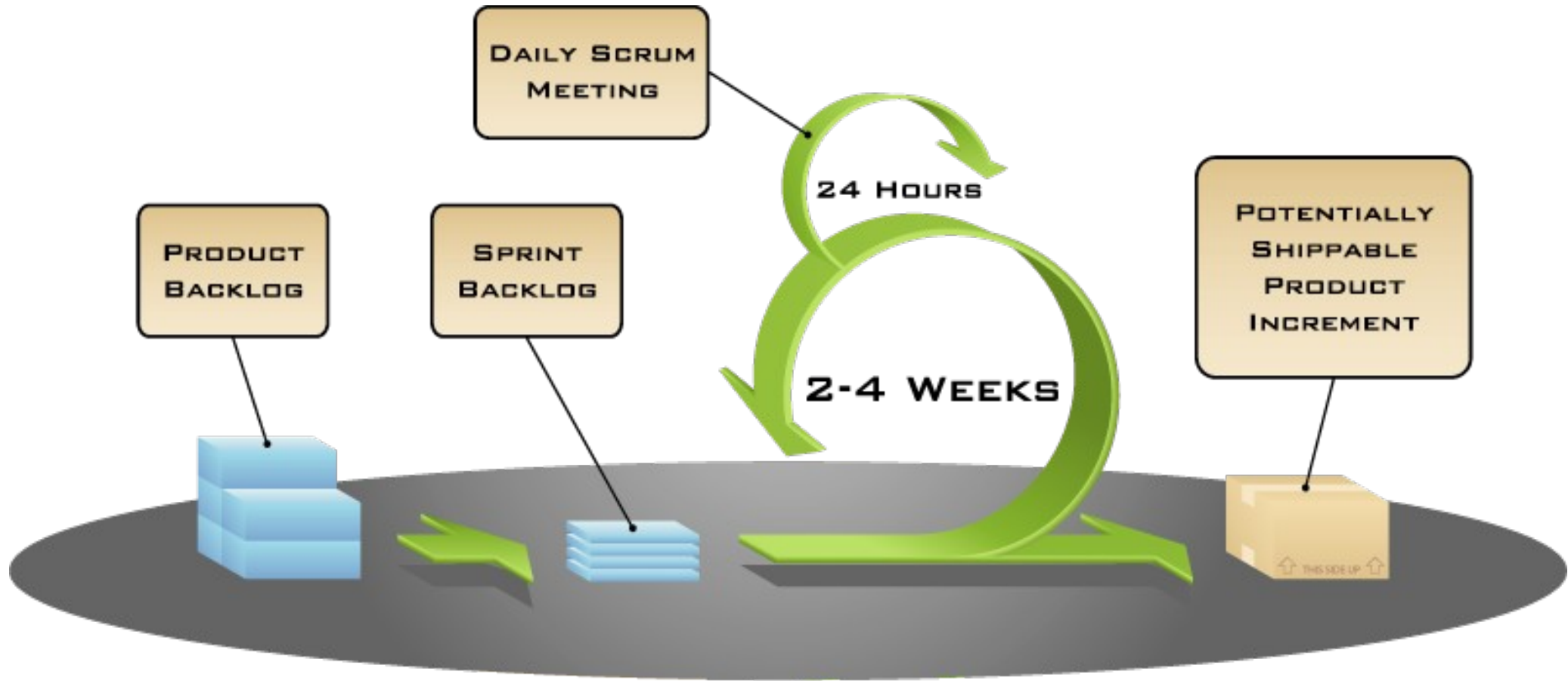
Scrum has been used for:

- 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
- Video game development
- FDA-approved, life-critical systems
- Satellite-control software
- Websites
- Hand-held software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use

Characteristics

- Self-organizing teams
- Product progresses in a series of month-long “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- One of the “agile processes”

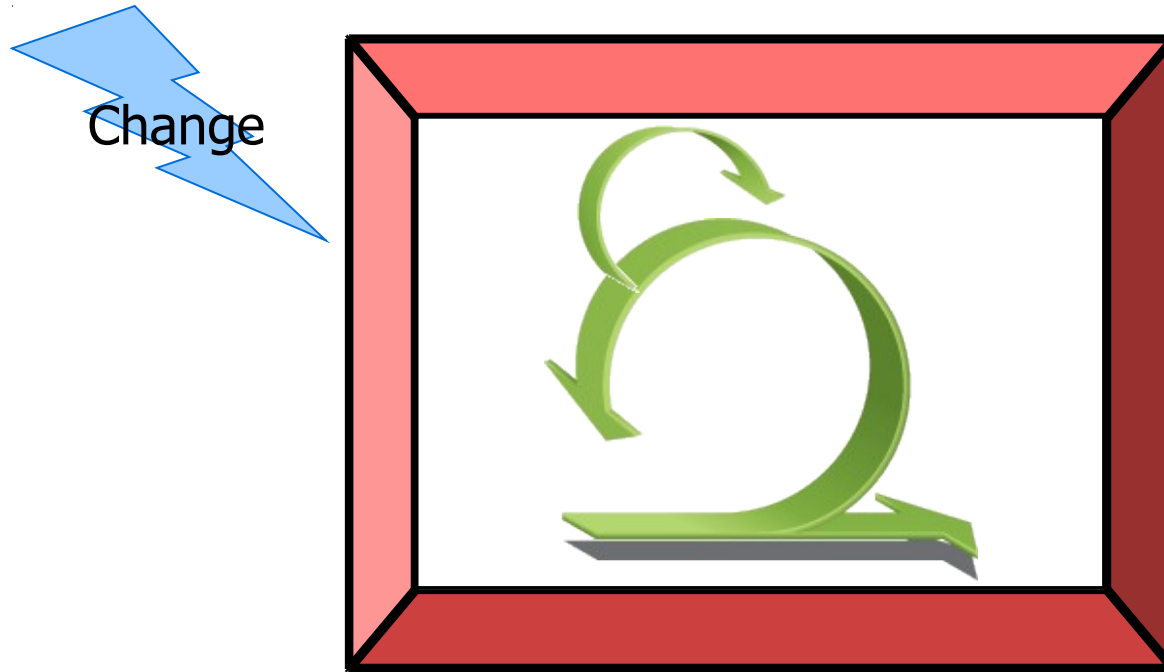
Putting it all together



Sprints

- Scrum projects make progress in a series of “sprints”
 - Analogous to Extreme Programming iterations
- Typical duration is 2 to 4 weeks
- A constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint

No changes during a sprint



- Plan sprint durations around how long you can commit to keeping change out of the sprint

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

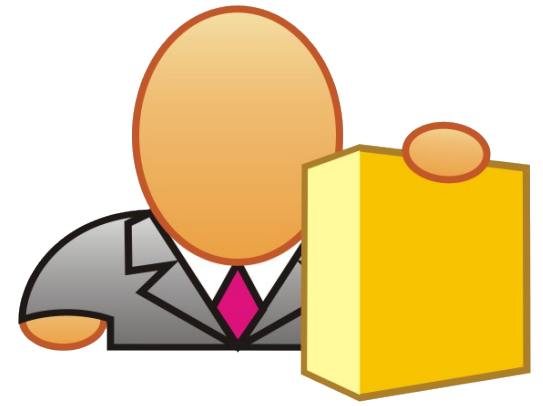
Events

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

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 iteration, as needed
- Accept or reject work results

- Balancing between “Pig” and “Chicken” role
 - (Just like in the story... :-))

The ScrumMaster



- 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

- “Pig” role

The team



- Typically 5-9 people
- Cross-functional:
 - Programmers, testers, user experience designers, etc.
- Members should be full-time
 - May be exceptions (e.g., database administrator)
- Teams are self-organizing
 - Ideally, no titles but rarely a possibility
- Membership should change only between sprints

- “Pig” role

Scrum framework

Roles

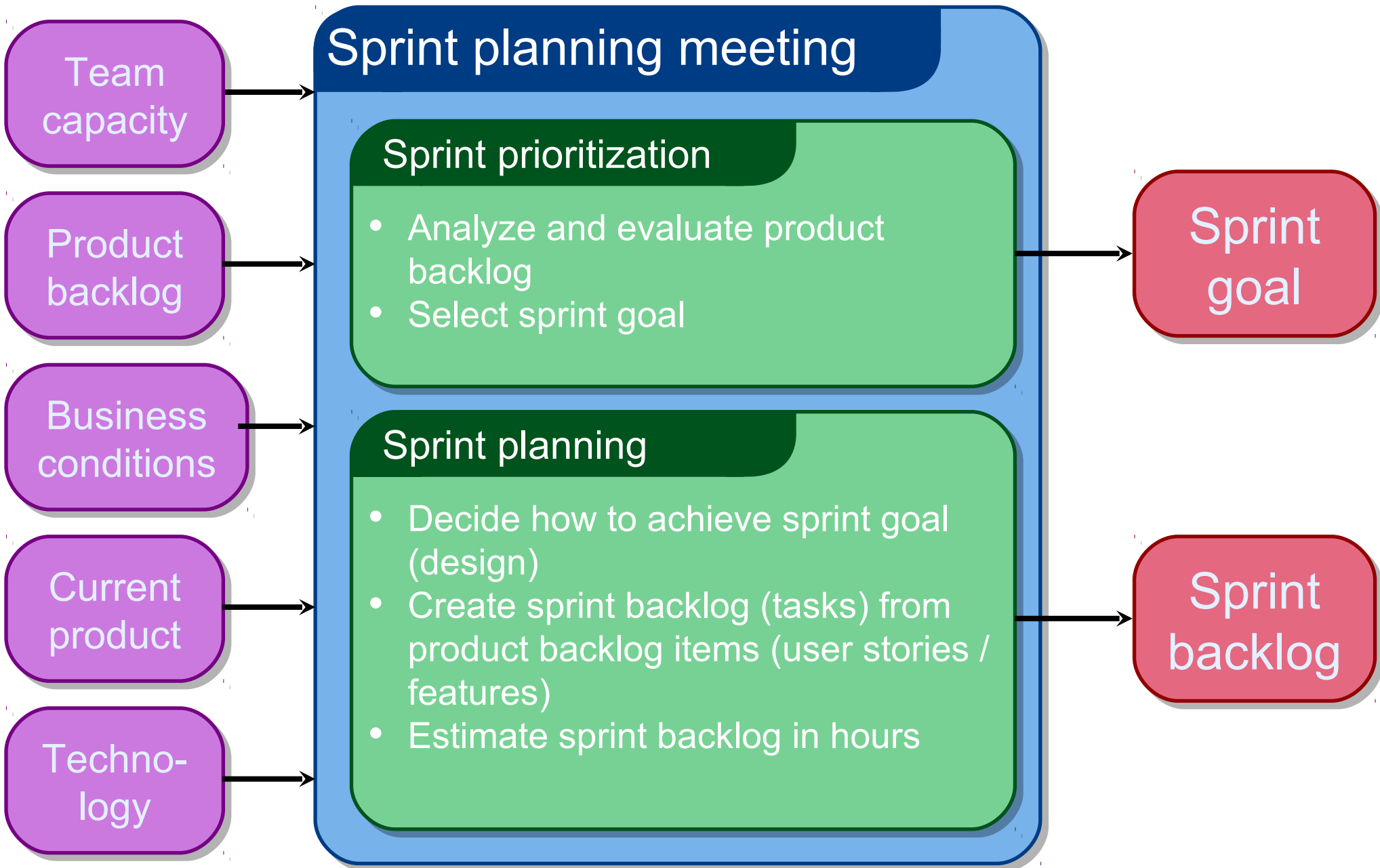
- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts



Sprint planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
 - Tasks are identified and each is estimated (1-16 hours)
 - Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

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

Code the middle tier (8 hours)
Code the user interface (4)
Write test fixtures (4)
Code the foo class (6)
Update performance tests (4)

The daily scrum

- Parameters
 - Daily
 - 15 minutes
 - Stand-up
- Not for problem solving
 - Whole world is invited
 - Only “pig” roles and the product owner can talk
- Allows team to stay focused on the task
- Helps avoid other unnecessary meetings



Everyone answers 3 questions

1
What did you do yesterday?

2
What will you do today?

3
Is anything in your way?

- These are *not* status for the ScrumMaster
 - They are commitments in front of peers
 - At least in the ideal world. :-)

The sprint review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
 - 2-hour prep time rule
 - No slides
- Whole team participates
- Invite the world



Sprint retrospective

- Periodically take a look at what is and is not working
- Typically 15 to 30 minutes
- Done after every sprint
- Everyone participates
 - ScrumMaster
 - Product owner
 - Team
 - Possibly customers and others

Start / Stop / Continue

- Whole team gathers and discusses what they'd like to:

Start doing

Stop doing

Continue doing

This is just one of many ways to do a sprint retrospective.

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Re-prioritized at the start of each sprint



This is the product backlog

A sample product backlog

Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
Improve exception handling	8
...	30
...	50

The sprint goal

- A short statement of what the work will be focused on during the sprint

DB Application

Make the application run on SQL Server in addition to Oracle.

Life Sciences

Support features necessary for population genetics studies.

Financial services

Support more technical indicators than company ABC with real-time, streaming data.

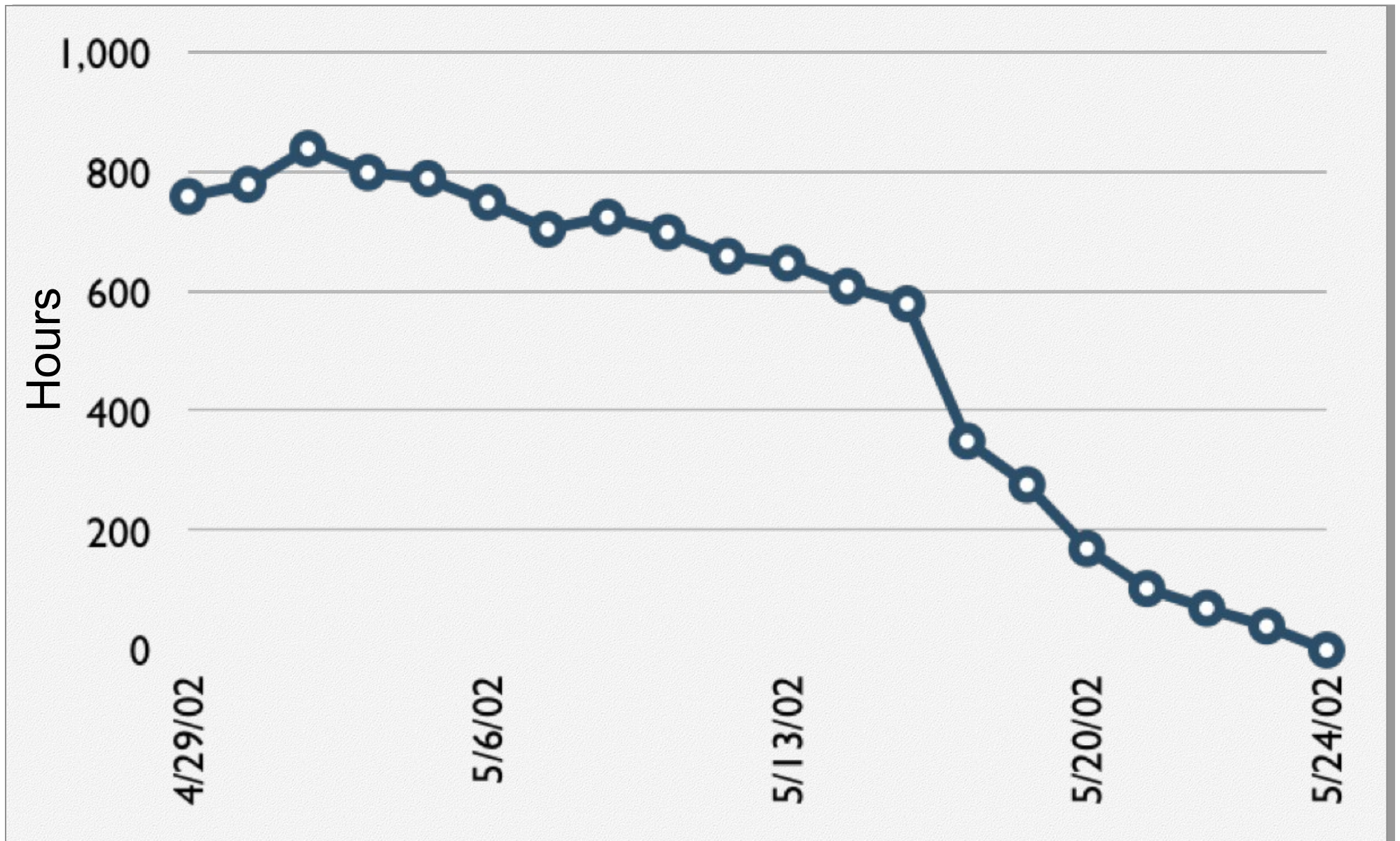
Managing the sprint backlog

- Individuals sign up for work of their own choosing
 - Work is never assigned. Ideally. :-)
- Estimated work remaining is updated daily
- Any team member can modify 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

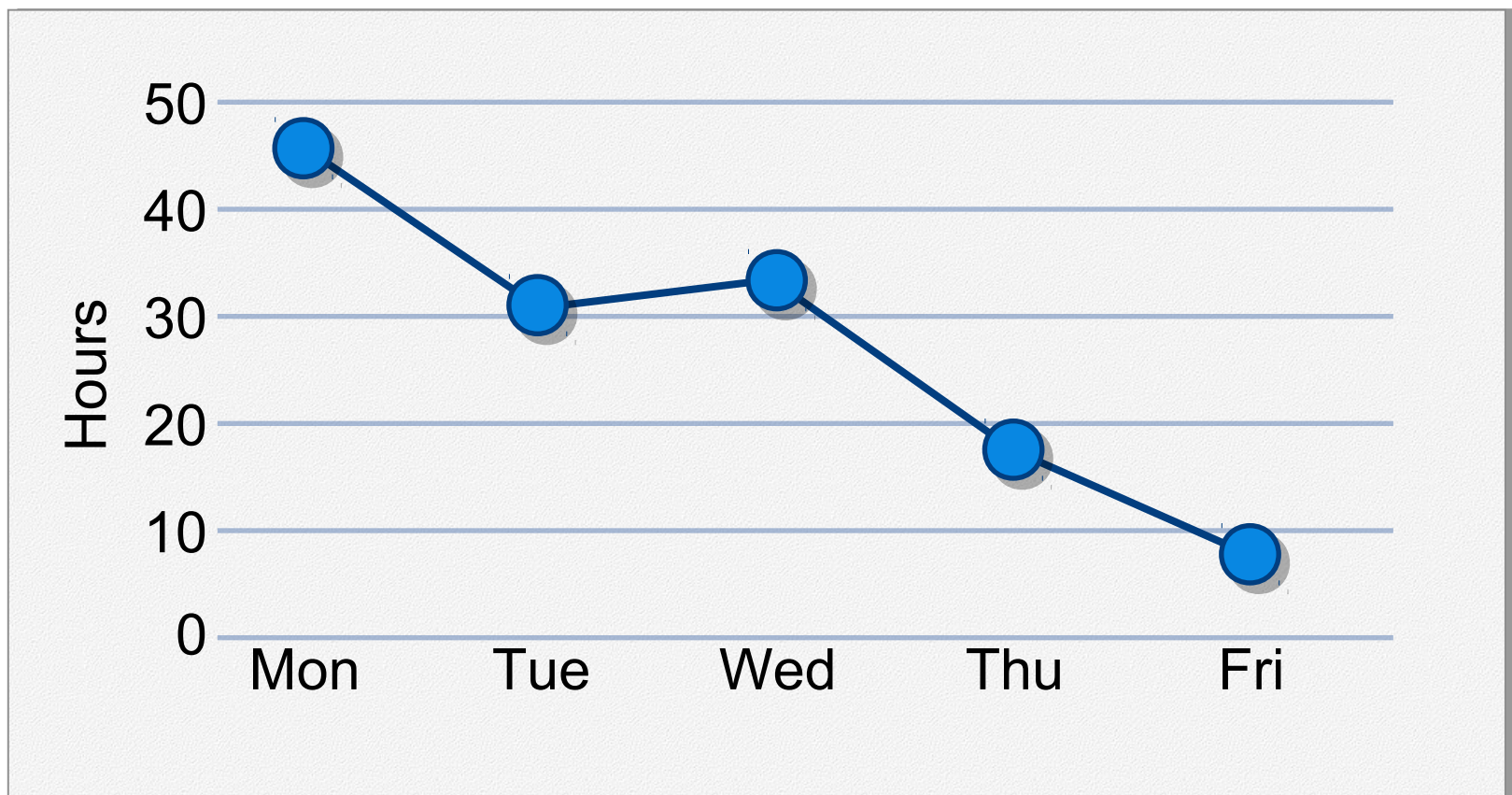
A sprint backlog

Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

A sprint burndown chart



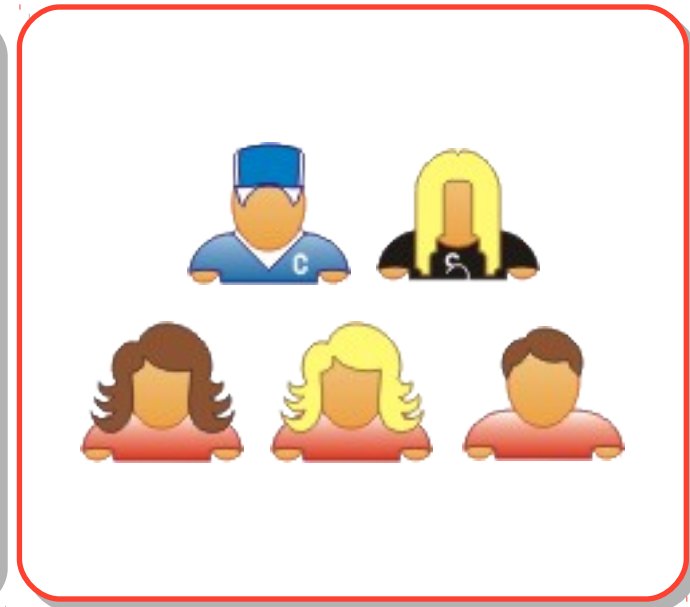
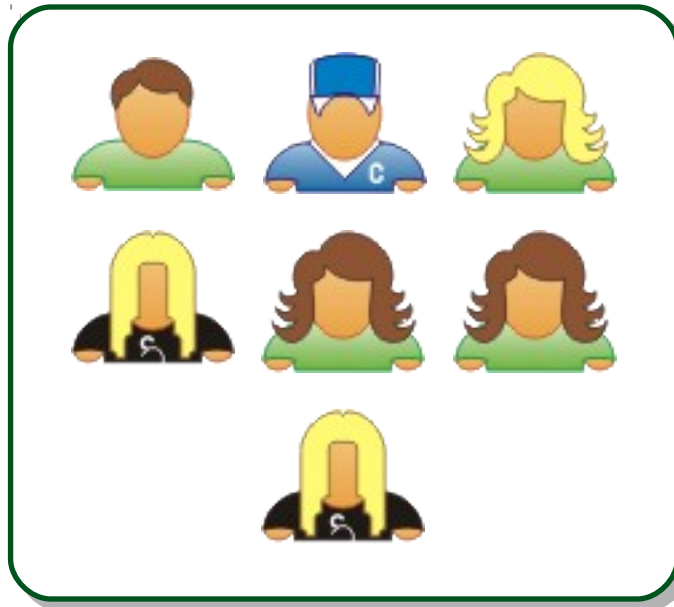
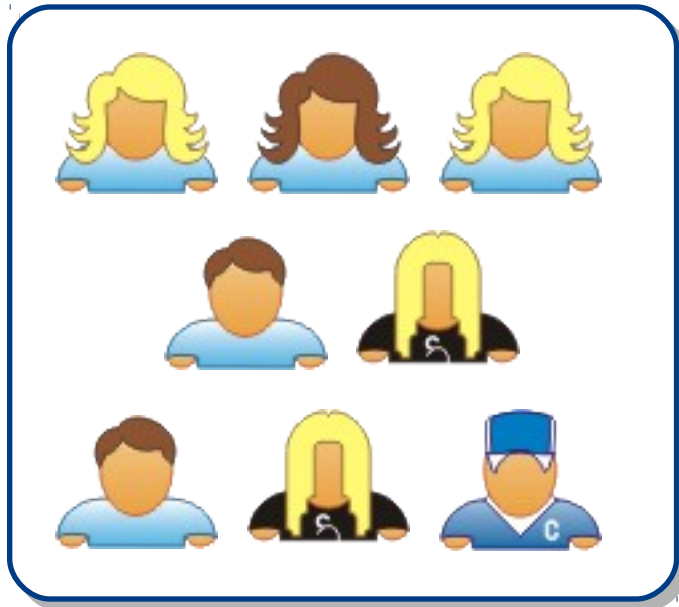
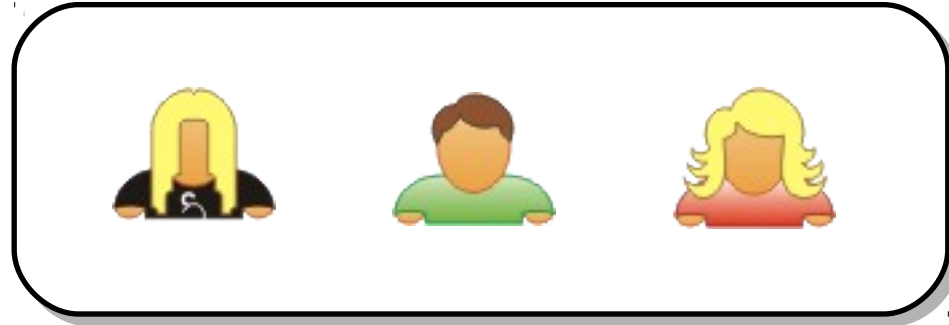
Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



Scalability

- Typical individual team is 7 ± 2 people
 - Scalability comes from teams of teams
- Factors in scaling
 - Type of application
 - Team size
 - Team dispersion
 - Project duration
- Scrum has been used on multiple 500+ person projects

Scaling through the Scrum of scrums



Scrum of scrums of scrums



Limits of Agile Programming

- No method is perfect...
- Requirements on the team
 - Open-minded developers, not complete newbies
 - Being communicative and disciplined
- Can obfuscate the big picture
- Occasional need to rewrite code

Where to go next

- www.scrum.org/scrumguides/
- www.scrumalliance.org/resources
- www.mountangoatsoftware.com/scrum
- www.controlchaos.com
- scrumdevelopment@yahogroups.com

A Scrum reading list

- *Agile and Iterative Development: A Manager's Guide* by Craig Larman
- *Agile Estimating and Planning* by Mike Cohn
- *Agile Project Management with Scrum* by Ken Schwaber
- *Agile Retrospectives* by Esther Derby and Diana Larsen
- *Agile Software Development Ecosystems* by Jim Highsmith
- *Agile Software Development with Scrum* by Ken Schwaber and Mike Beedle
- *Scrum and The Enterprise* by Ken Schwaber
- *User Stories Applied for Agile Software Development* by Mike Cohn
- Lots of weekly articles at www.scrumalliance.org

Copyright notice

- Original reusable Scrum presentation from Mountain Goat Software Adapted for Academic use by Martin Hince
- You are free:
 - to Share—to copy, distribute and transmit the work
 - to Remix—to adapt the work
- Under the following conditions
 - Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- Nothing in this license impairs or restricts the author's moral rights.
- For more information see <http://creativecommons.org/licenses/by/3.0/>



Happy Scrumming!

