An Introduction to Scrum

Mountain Goat Software, LLC

We're losing the relay race

"The… 'relay race' approach to product development…may conflict with the goals of maximum speed and flexibility. Instead a holistic or 'rugby' approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today's competitive requirements."

Hirotaka Takeuchi and Ikujiro Nonaka, "The New New Product Development Game", *Harvard Business Review*, January 1986.

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.

Scrum origins

- Jeff Sutherland
  - Initial scrums at Easel Corp in 1993
  - IDX and 500+ people doing Scrum
- Ken Schwaber
  - ADM
  - Scrum presented at OOPSLA 95 with Sutherland
  - Author of three books on Scrum
- Mike Beedle
  - Scrum patterns in PLOPD4
- Ken Schwaber and Mike Cohn
  - Co-founded Scrum Alliance in 2002, initially within the Agile Alliance

Scrum has been used by:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- 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
- Handheld software
- Mobile phones
- Network switching applications
- ISV applications
- Some of the largest applications in use

Characteristics

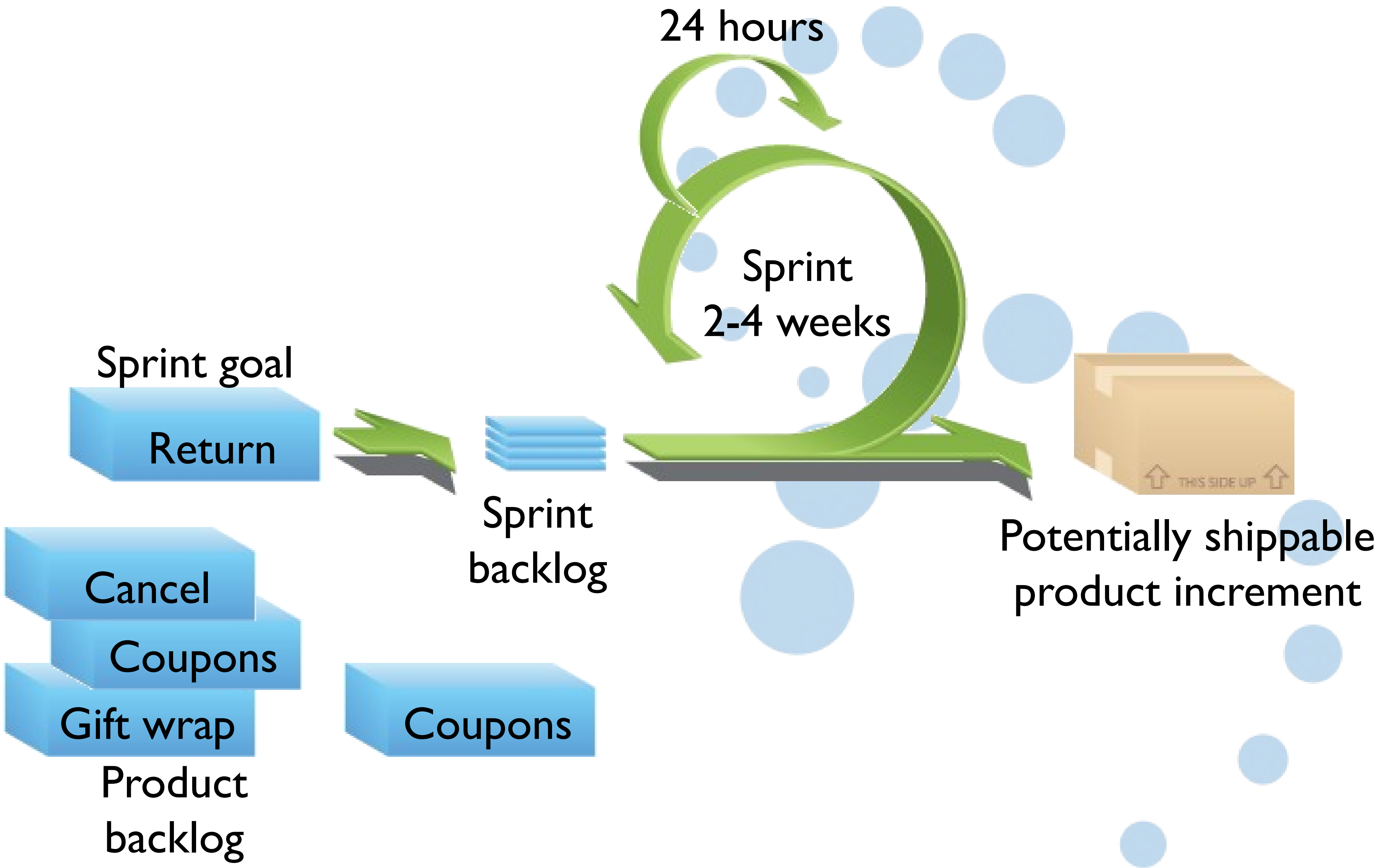
- Self-organizing teams
- Product progresses in a series of month-long "sprints"
- Requirements are captured as items in a list of "product backlog"
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the "agile processes"

The Agile Manifesto–a statement of values

Individuals and interactions	over	Process and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

Source: www.agilemanifesto.org

Scrum


24 hours
Sprint
2-4 weeks
Sprint goal
Return
Sprint
backlog
Potentially shippable
product increment
Cancel
Coupons
Gift wrap
Coupons
Product
backlog
THIS SIDE UP

Putting it all together

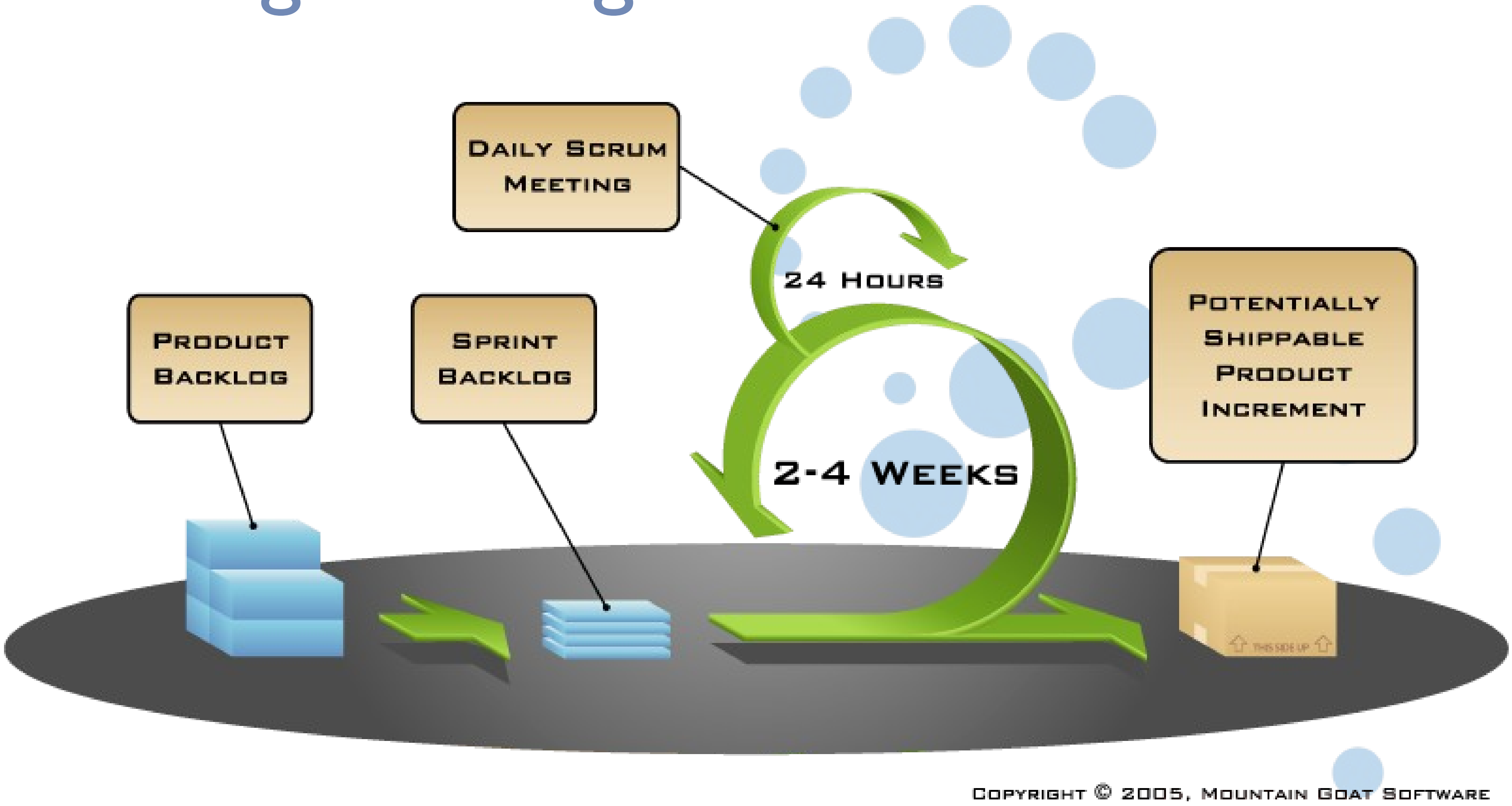
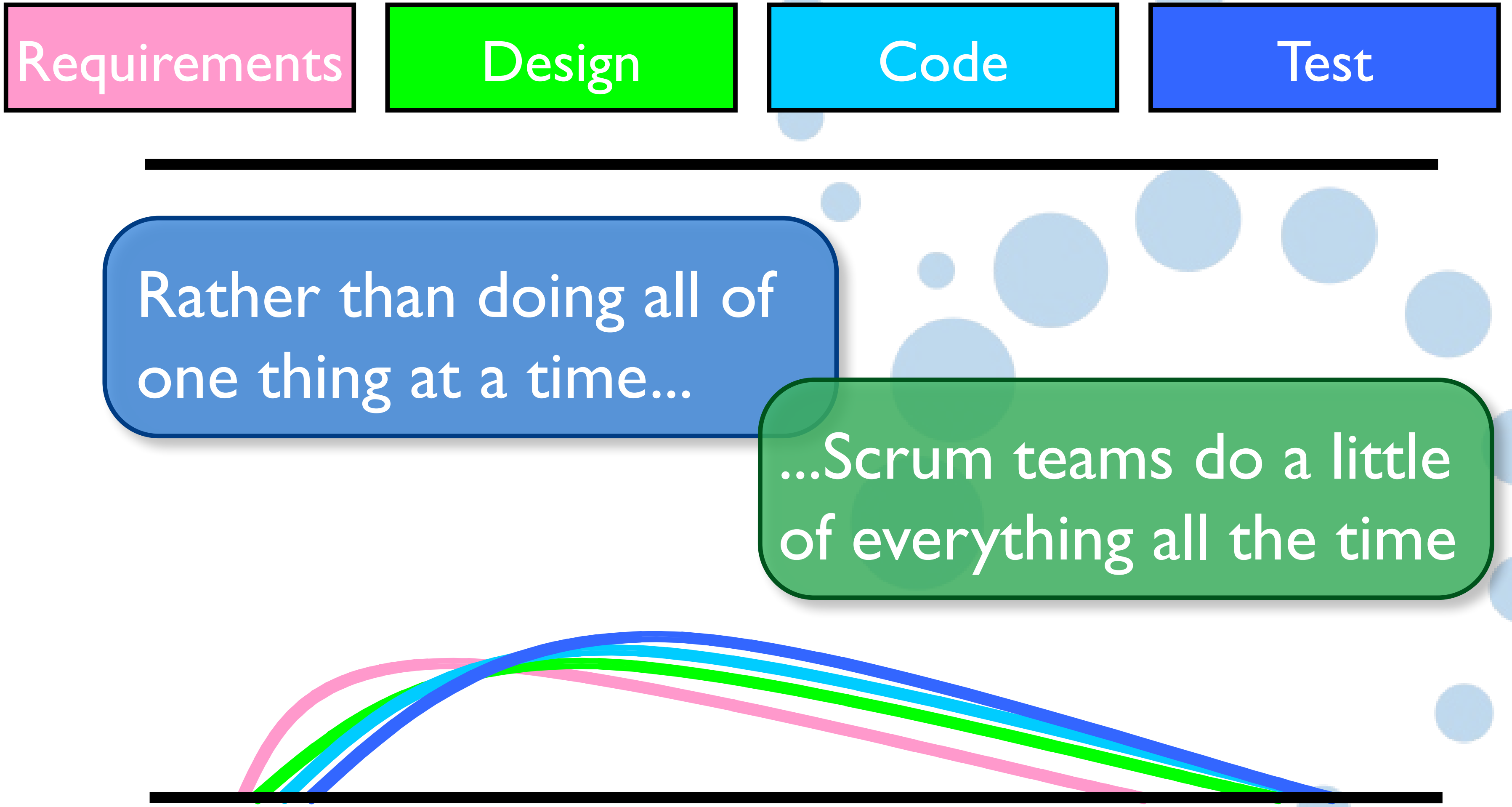


Image available at
www.mountaingoatsoftware.com/scrum

Sprints

- Scrum projects make progress in a series of “sprints”
  - Analogous to Extreme Programming iterations
- 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

Sequential vs. overlapping development

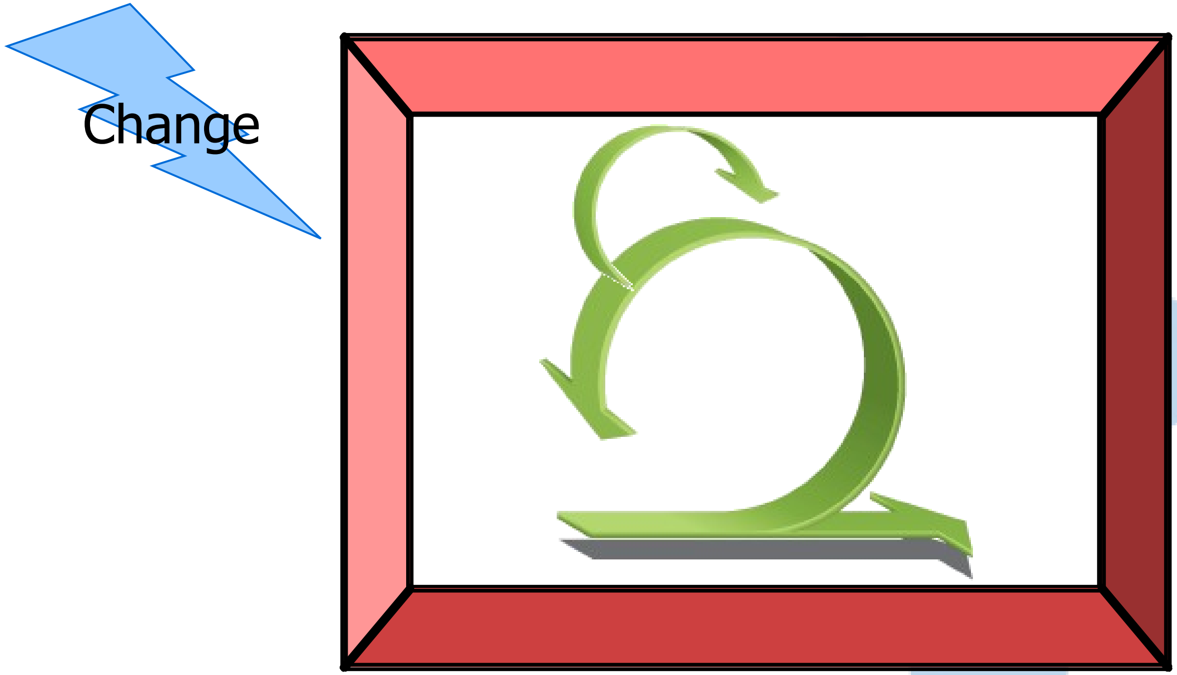


Rather than doing all of one thing at a time...

...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

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

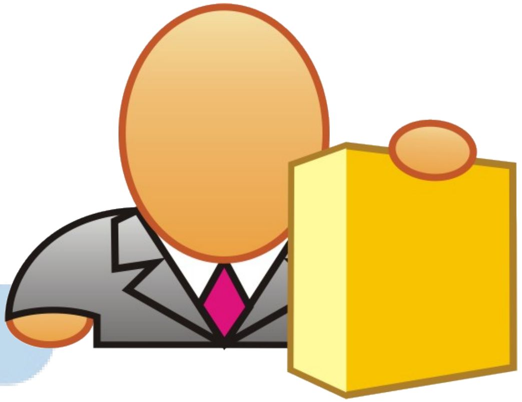
Ceremonies

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

Artifacts

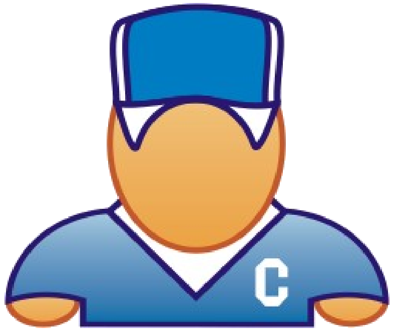
- Product backlog
- Sprint backlog
- Burndown charts

Product owner



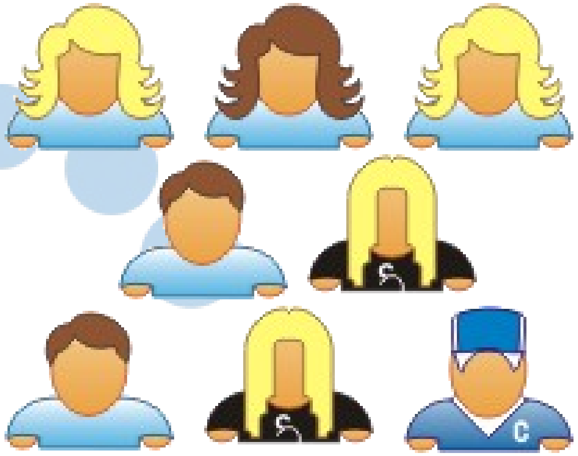
- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

The ScrumMaster



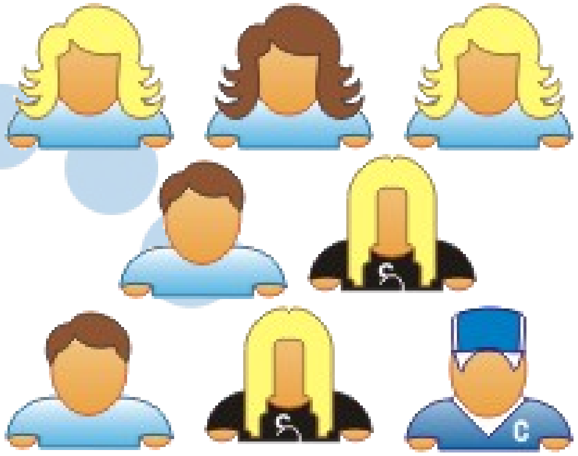
- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

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)

The team



- Teams are self-organizing
  - Ideally, no titles but rarely a possibility
- Membership should change only between sprints

Scrum framework

Roles

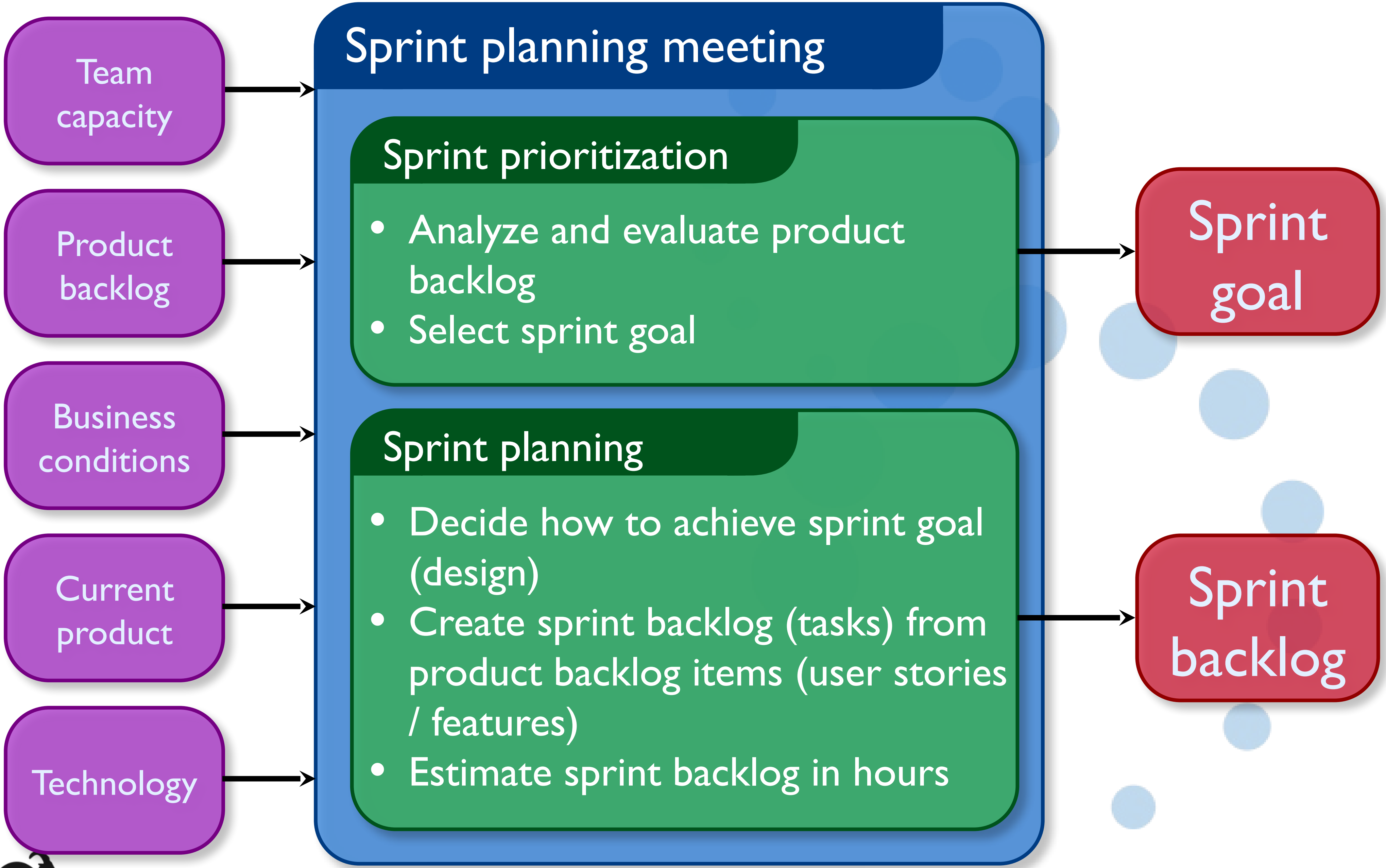
- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts



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

Team Capacity

- During sprint planning, teams face the challenge of sprint commitments.
  - How many stories can we commit in this sprint?
  - How to plan for the team capacity?
- Team capacity is calculated as per people availability in that sprint.
  - Ex. Team is of 5 people, then total capacity assuming 8 hour day, 2 weeks sprint(10 days) is = 5*8*10 = 400 hours. NOOOOO!
  - Planning for this total capacity will be disaster. It will lead to team working over time, rushing towards the end, quality cuts and low team morale.

Team Capacity & Focus Factor

Focus Factor (F.F)

- is teams ability to remain focussed on the sprint goals without any other distractions.
- lies in the range **0.6 – 0.8**
- After multiplying total capacity with focus factor you get real capacity against which you can make sprint commitments or forecasting. This is the effective hours you can expect from the team.
- applying focus factor say 0.6, then this team real capacity will be 400*0.6 = 240 hours
- Team will take on the stories till the time all the tasks sum to not more than 240 hours(in this example).

Team Capacity & Focus Factor

- Use lesser **focus factor** on the following situations:
  - When team is **starting new on a project**
  - Team is using **scrum for the first time**
  - Team is working on a **complex product or new to technology domain**
  - Team is **less matured, needs lot of handholding** …
  - **people being allocated to multiple projects, overhead of task switching comes to play**

Tips: start a team on successful note(improves morale). Using lesser focus factor when you start fresh and then if team meets sprint goals early, then they can take up more in the current sprint. Retrospect on this in coming sprints to see if you want to **increase focus factor marginally and fine tune**, iterate this factor as you go, to reach sustainable pace/Flow. Going beyond 0.8 can be risky and can derail teams too.

Team Capacity & Focus Factor

- organisation or product development is very chaotic then this factor will remain on extreme left like 0.6 or may be below.
  - **Chaotic organisations** have lot of **unplanned meetings,** pre-sales urgency, hiring team coming to the project team at a **last minute with a interview request**, not having defined core working hours, lesser clarity sprint backlog, wrong team structures(read too much inter-dependency) and list goes on… To summarise – **no rhythm**.
- **Tips:** start a team on successful note(improves morale). Using lesser focus factor when you start fresh and then if team meets sprint goals early, then they can take up more in the current sprint. Retrospect on this in coming sprints to see if you want to **increase focus factor marginally and fine tune**, iterate this factor as you go, to reach sustainable pace/Flow. Going beyond 0.8 can be risky and can derail teams too.

Team Capacity vs Velocity

- **What is Team's velocity**
  Number of story points delivered/demo in a Sprint is called velocity. For example, if team planned 30 story point(Business value) worth of user stories in a sprint and able to deliver as planned then team's velocity is 30.

User Stories

- very slim and high-level requirements artifacts (functional, non functional, constraints…)
  - Students can purchase monthly parking passes online.
  - Parking passes can be paid via credit cards.
  - Parking passes can be paid via PayPal.
  - Professors can input student marks.
  - Students can obtain their current seminar schedule.
  - Students can order official transcripts.
  - Students can only enroll in seminars for which they have prerequisites.
  - Transcripts will be available online via a standard browser.

User Stories

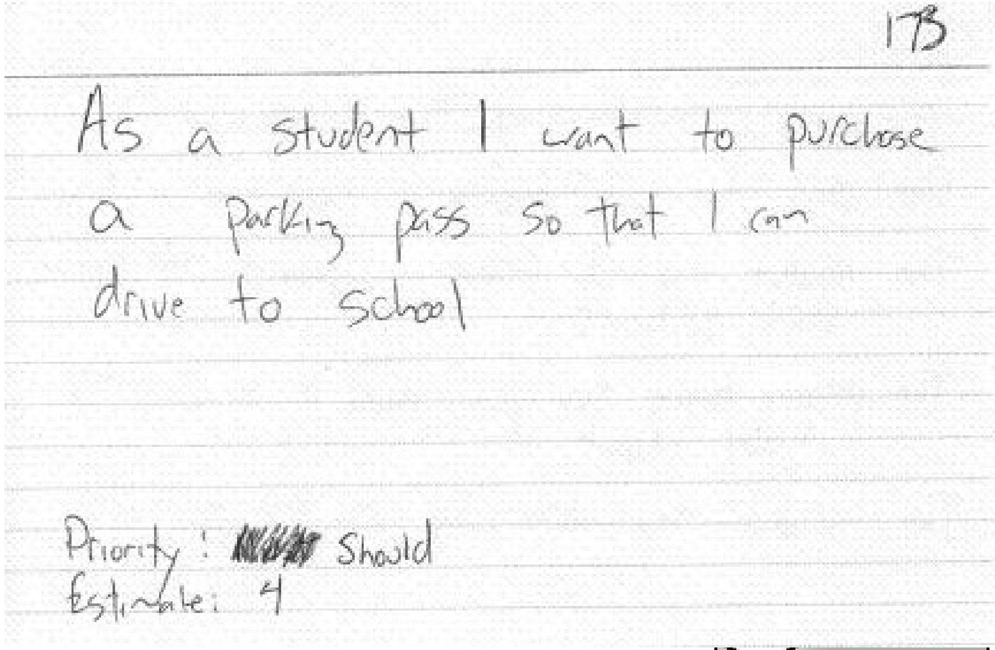
Format:

As a (role) I want (something) so that (benefit)

"As a Student I want to purchase a parking pass so that I can drive to school"

User story card
(stakeholder)

Epics & Themes

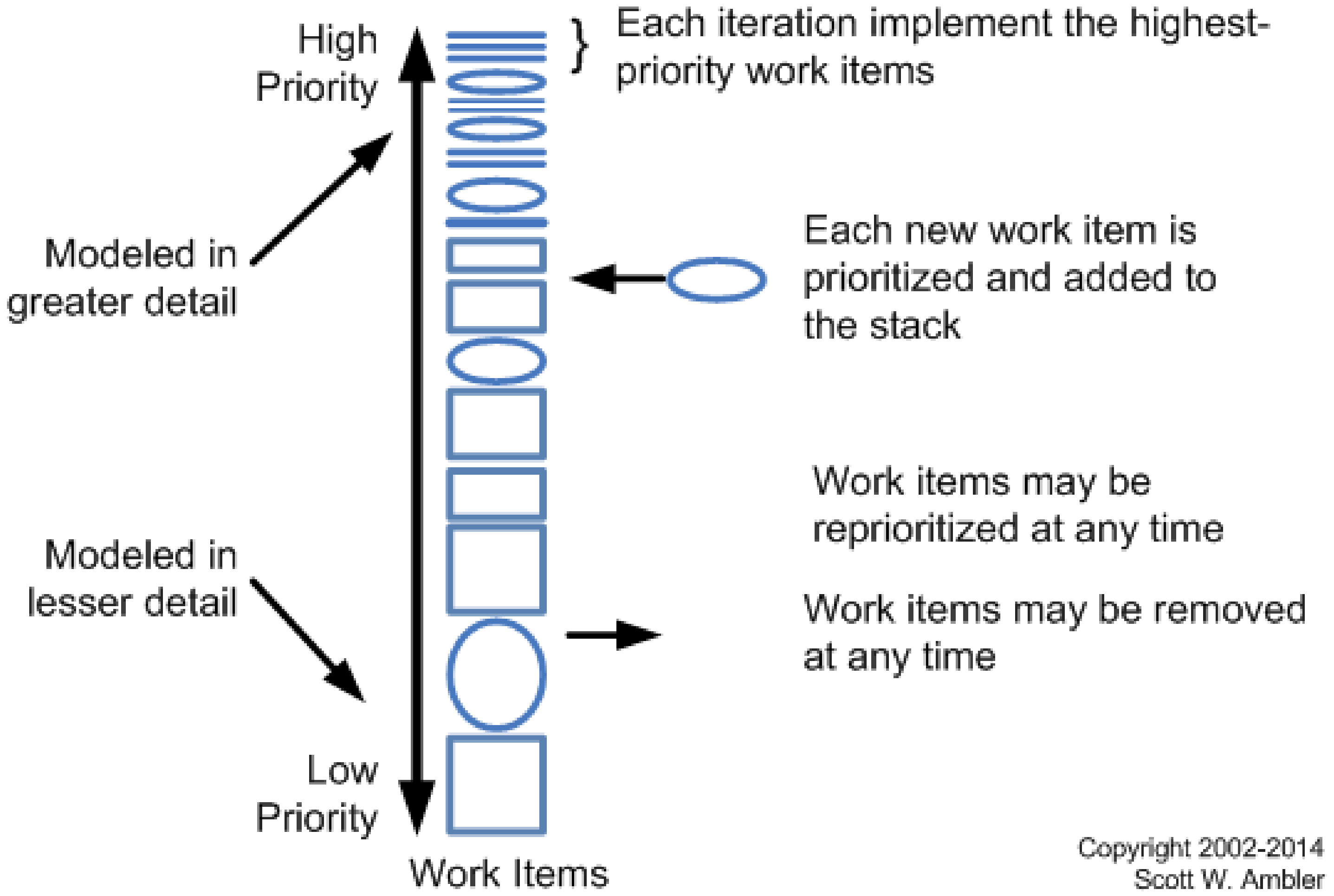


173

As a student I want to purchase a parking pass so that I can drive to school

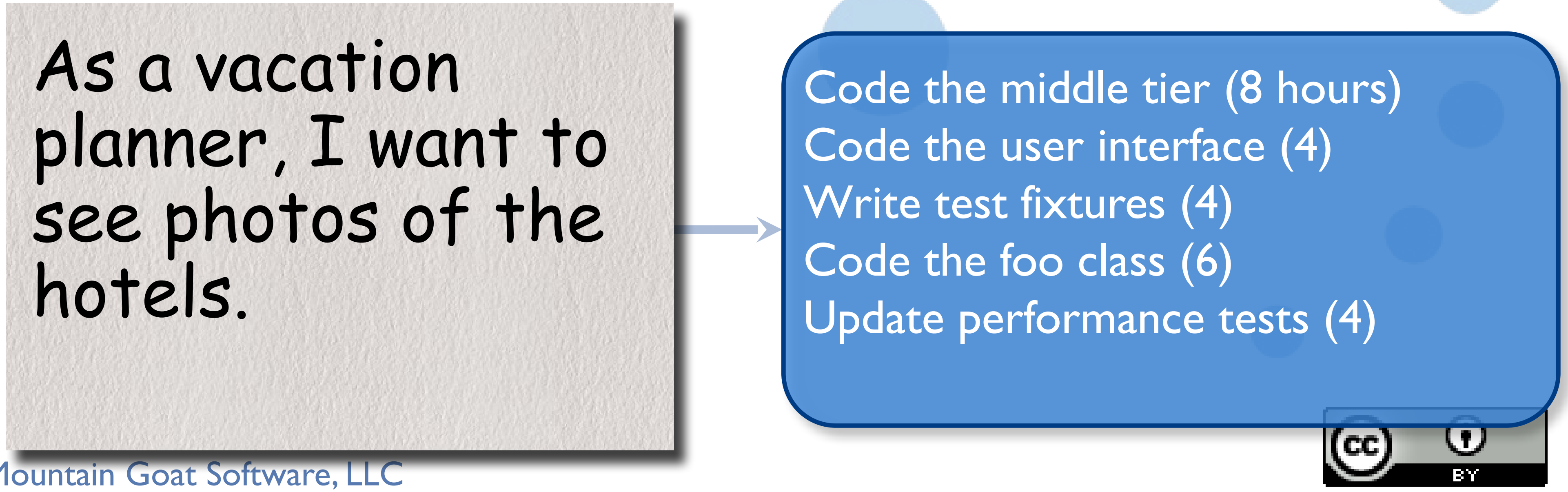
Priority: Should
Estimate: 4

Disciplined agile change management process


High
Priority
Each iteration implement the highest-priority work items
Modeled in greater detail
Each new work item is prioritized and added to the stack
Work items may be reprioritized at any time
Modeled in lesser detail
Work items may be removed at any time
Low
Priority
Work Items

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 team members, ScrumMaster, product owner, can talk
- 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

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 15–30 minutes
- Done after every sprint
- Whole team 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

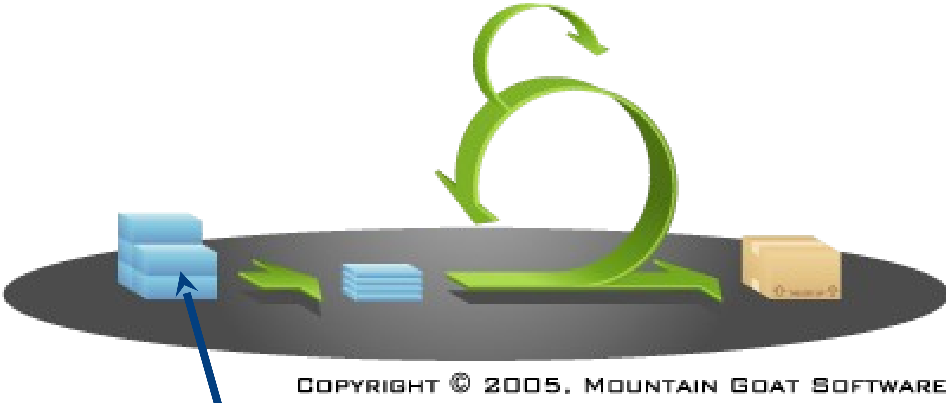
Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product backlog



This is the 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
- Reprioritized at the start of each sprint

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

Database 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
- Estimated work remaining is updated daily

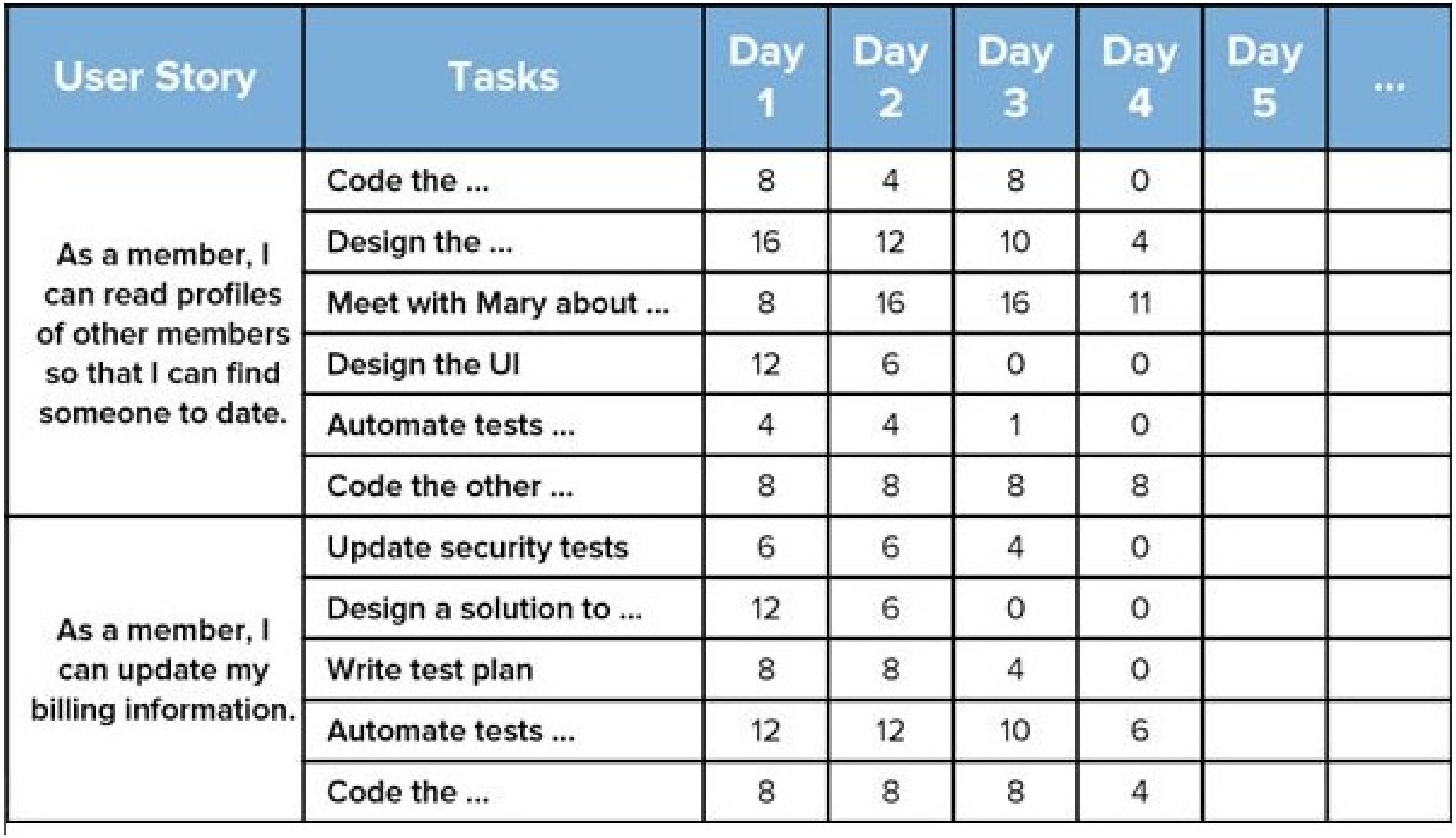
Managing the sprint backlog

- 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

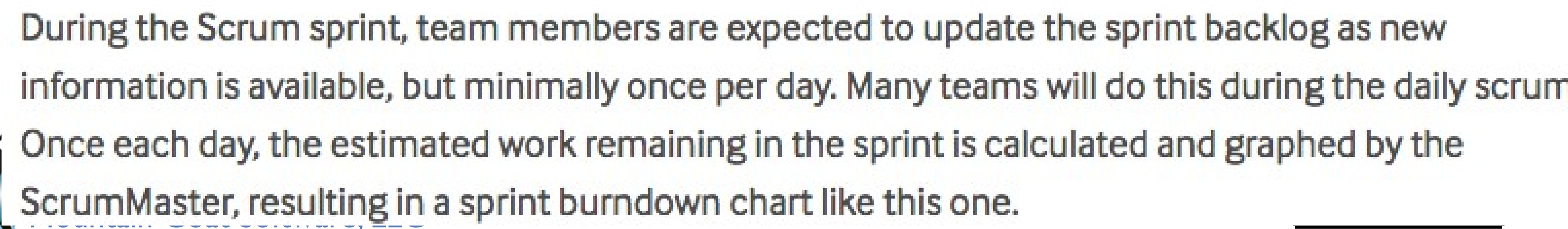
A sprint backlog

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 backlog with user stories



User Story	Tasks	Day 1	Day 2	Day 3	Day 4	Day 5	...
As a member, I can read profiles of other members so that I can find someone to date.	Code the ...	8	4	8	0		
	Design the ...	16	12	10	4		
	Meet with Mary about ...	8	16	16	11		
	Design the UI	12	6	0	0		
	Automate tests ...	4	4	1	0		
	Code the other ...	8	8	8	8		
As a member, I can update my billing information.	Update security tests	6	6	4	0		
	Design a solution to ...	12	6	0	0		
	Write test plan	8	8	4	0		
	Automate tests ...	12	12	10	6		
	Code the ...	8	8	8	4		

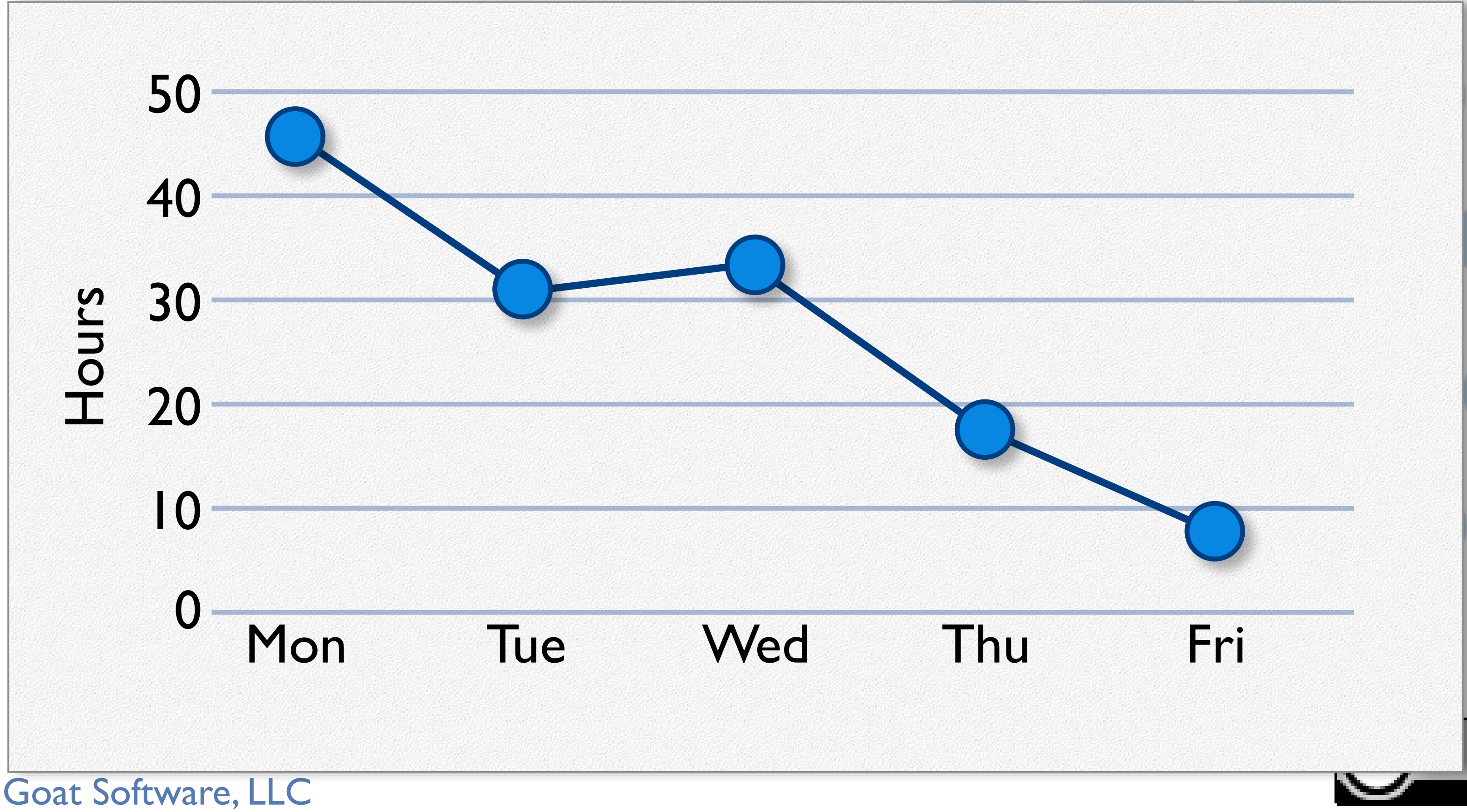


During the Scrum sprint, team members are expected to update the sprint backlog as new information is available, but minimally once per day. Many teams will do this during the daily scrum Once each day, the estimated work remaining in the sprint is calculated and graphed by the ScrumMaster, resulting in a sprint burndown chart like this one.

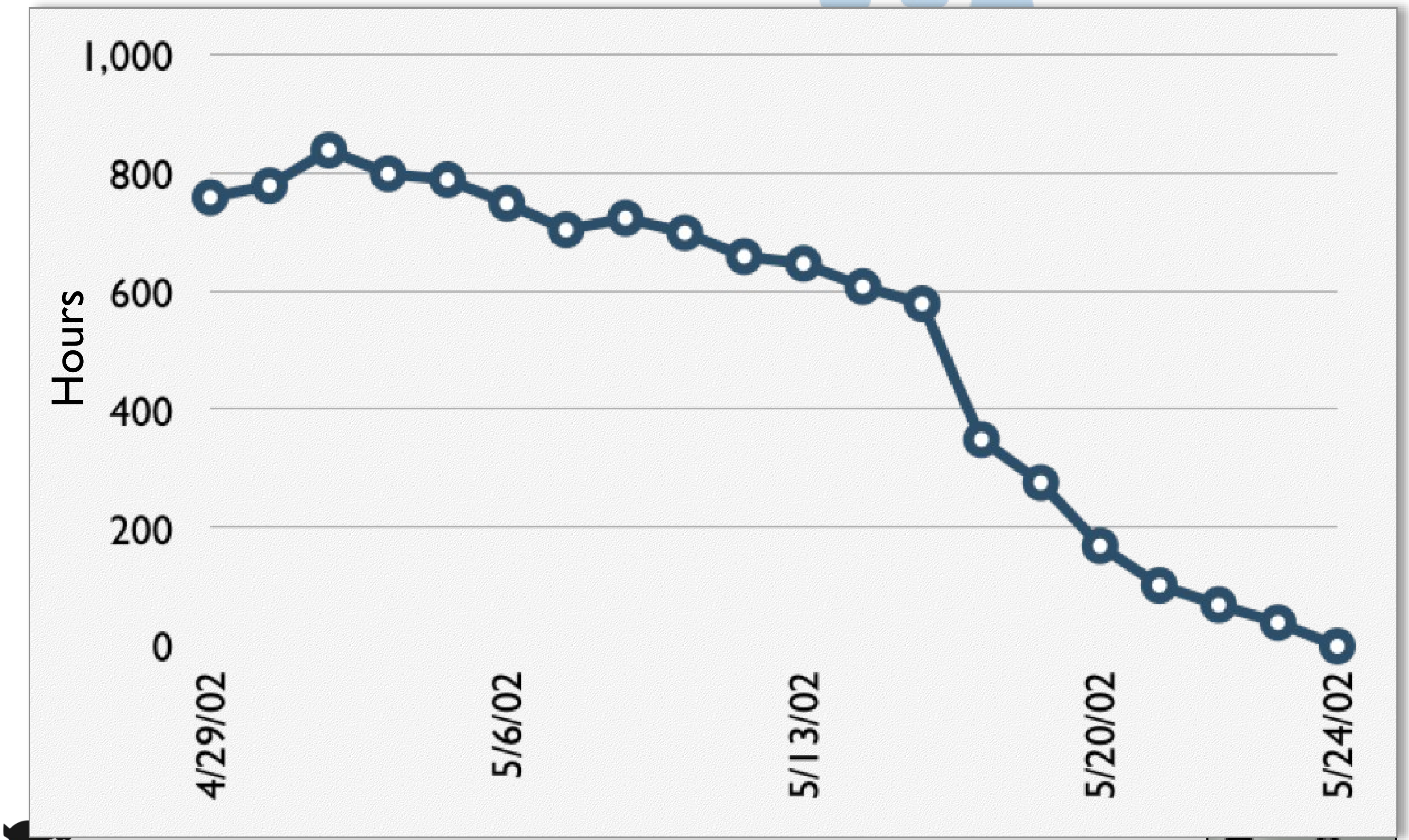
Sprint Burndown Chart

- Teams use the sprint Burndown chart to track the product development *effort remaining in a sprint.*
- It should consist of:
  - X axis to display working days
  - Y axis to display remaining effort
  - Ideal effort as a guideline
  - Real progress of effort

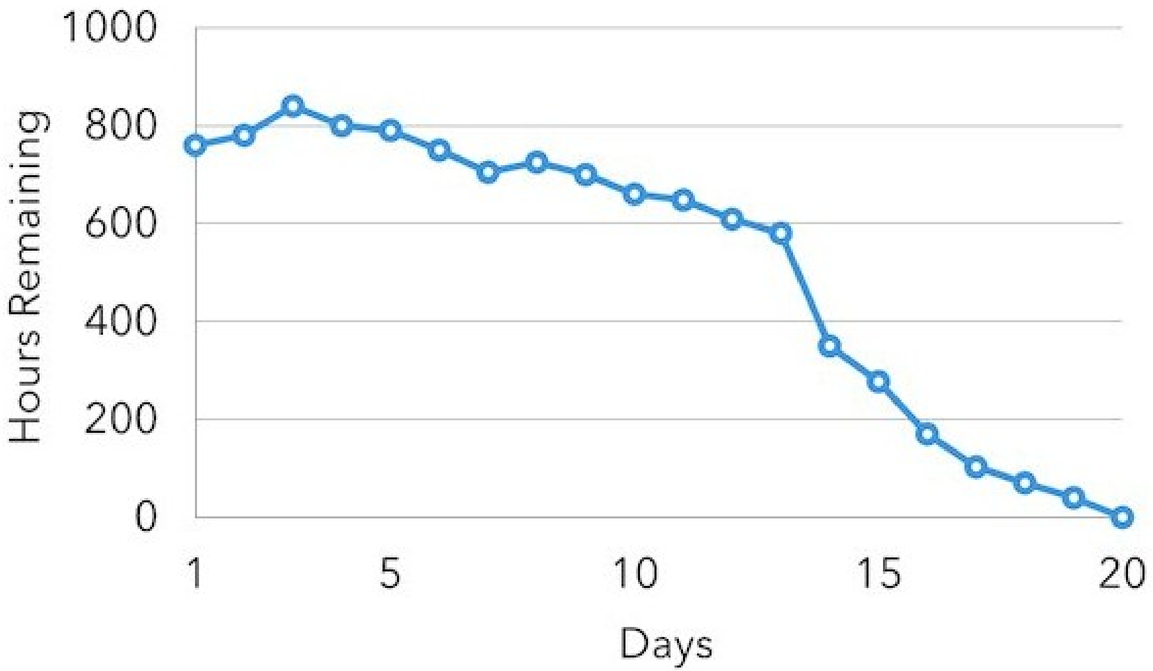
Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



A sprint burndown chart



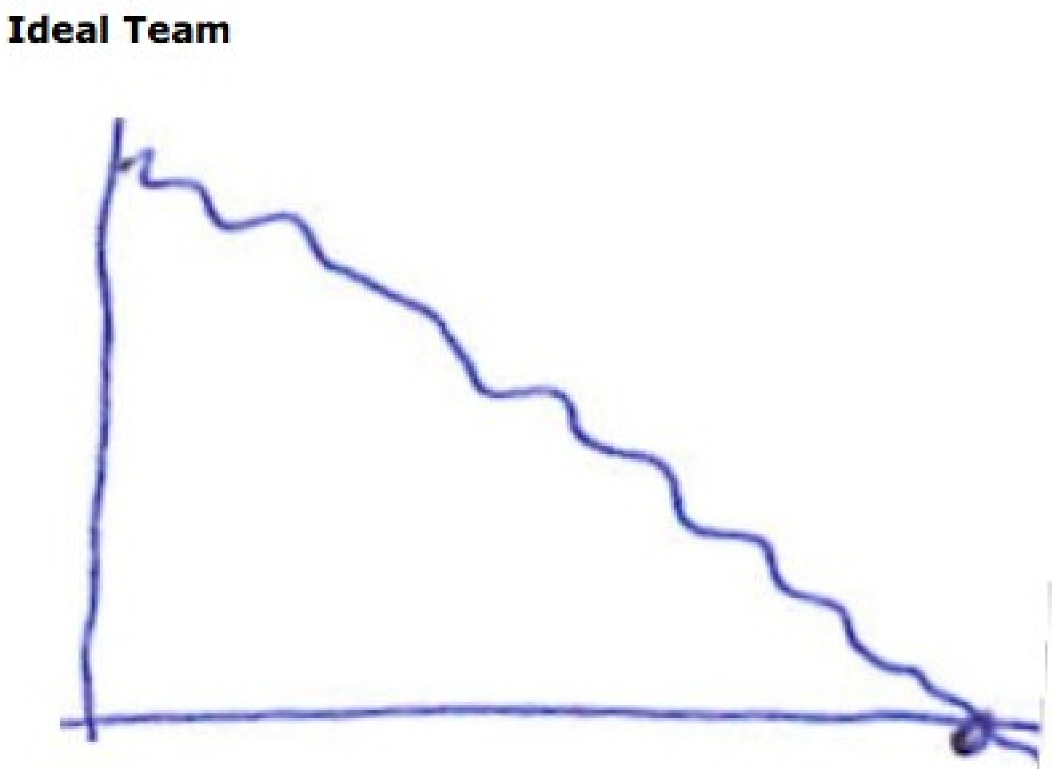
A sprint burndown chart



The team does its best to pull the right amount of work into the Scrum sprint, but sometimes too much or too little work is pulled in during planning. In this case, the team needs to add or remove tasks.

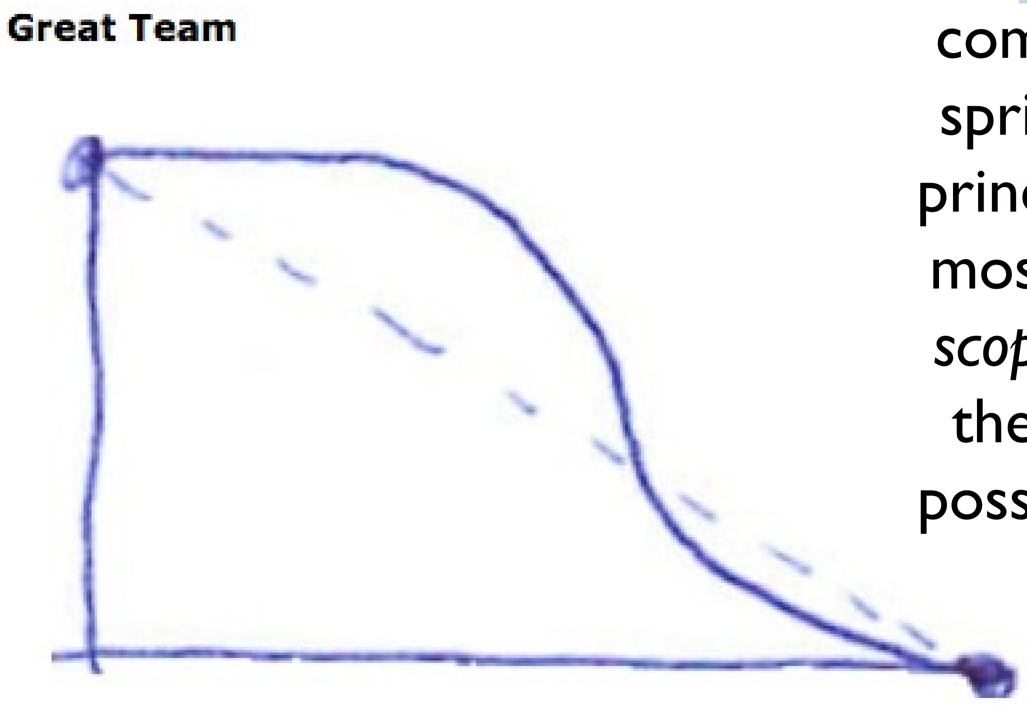
The team pulled in too much work initially into the sprint backlog, and still had nearly 600 hours to go on day 13 of a 20-day sprint. The product owner was consulted and agreed to remove some user stories from the sprint. This resulted in the big drop on the chart between days 13 and 14. From there, the team made consistent progress and finished the Scrum sprint successfully.

What sprint burndown chart says



Such diagram indicates *the great team* able to organize itself. It indicates a *great product owner* who understands the reason for a locked sprint backlog and a *great Scrum master* able to help the team. The team is not over-committing and finished the spring backlog on time. The team is also able to estimate capacity correctly. No corrective action is necessary in such case.

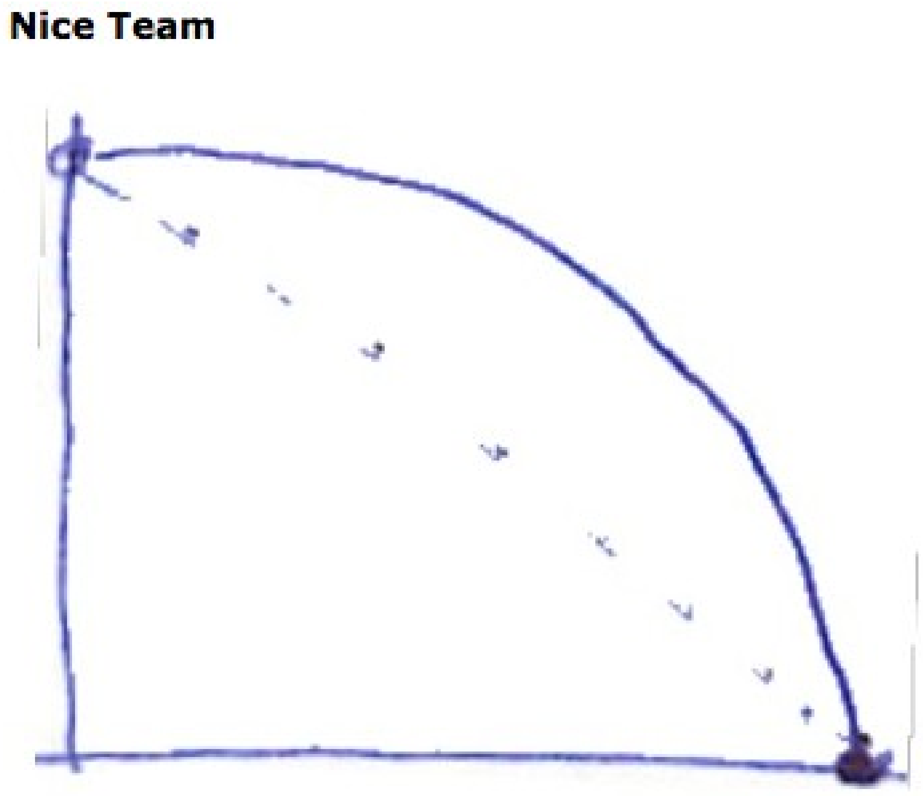
What sprint burndown chart says



Experienced teams. The team has completed work on time and met the sprint goal. They also have applied the principle of getting things done, but the most important is they have *adapted a scope of the sprint backlog* to complete the sprint. At the end the team has a possibility to complete some additional work.

In the retrospective, the team should discuss the reasons of late progress in the first half of the sprint and solve issues so they are better in the next sprint. The team should also consider the capacity that they are able to complete.

What sprint burndown chart says



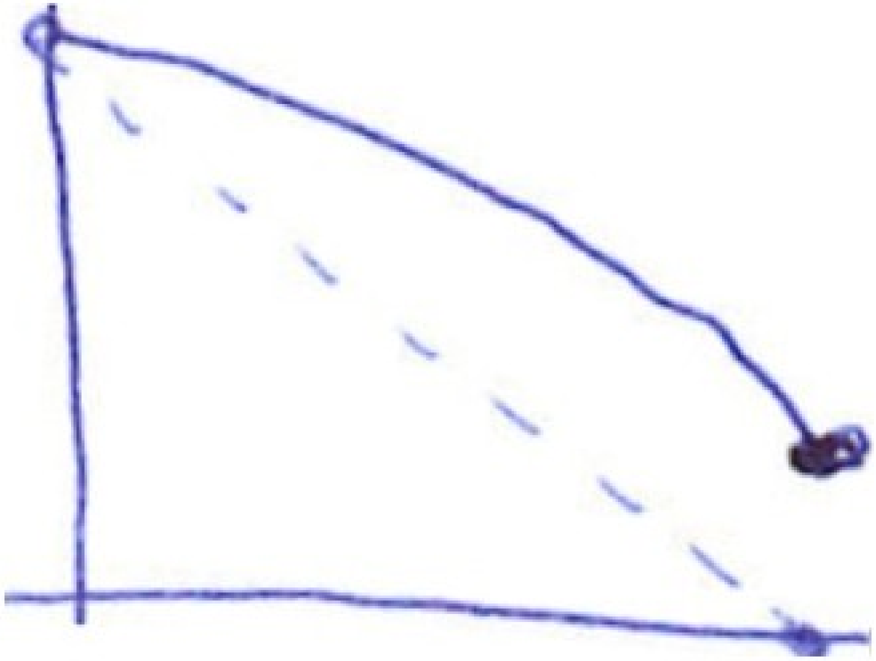
This is a *typical progress* that can be observed in many experienced agile teams.

The chart says again that the team was able to complete their commitment on time. They adapted the scope or worked harder to complete the sprint. The team is self-reflecting.

The team should discuss change of plan immediately as they see the progress has been slowing down from the beginning of the sprint. Typically it is suggested to move a low priority item from the sprint backlog to the next sprint or back to the product backlog.

What sprint burndown chart says

Boom. It Is Too Late.



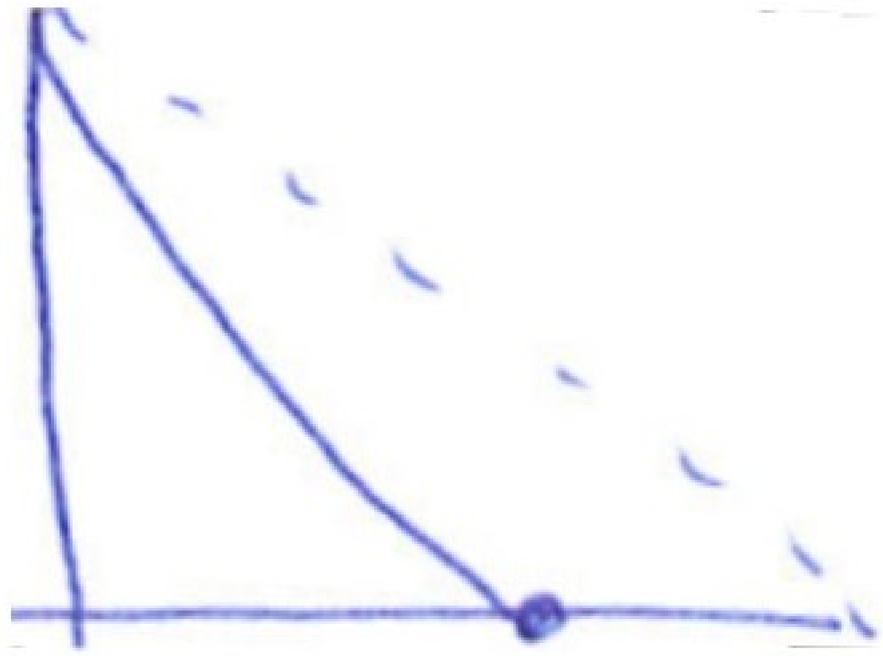
This burndown chart says: "You have not completed your commitment".

The team has been late for the entire sprint. The team did not adapt the sprint scope to appropriate level. It shows that the team has not completed stories that should have been split or moved to the next sprint.

In such situation the capacity of the next sprint should be lowered. If this happens again, corrective actions should be taken after a few days when slower progress is observed. Typically, lower priority story should be moved to the next sprint or back to the product backlog.

What sprint burndown chart says

Boom. Too Early.

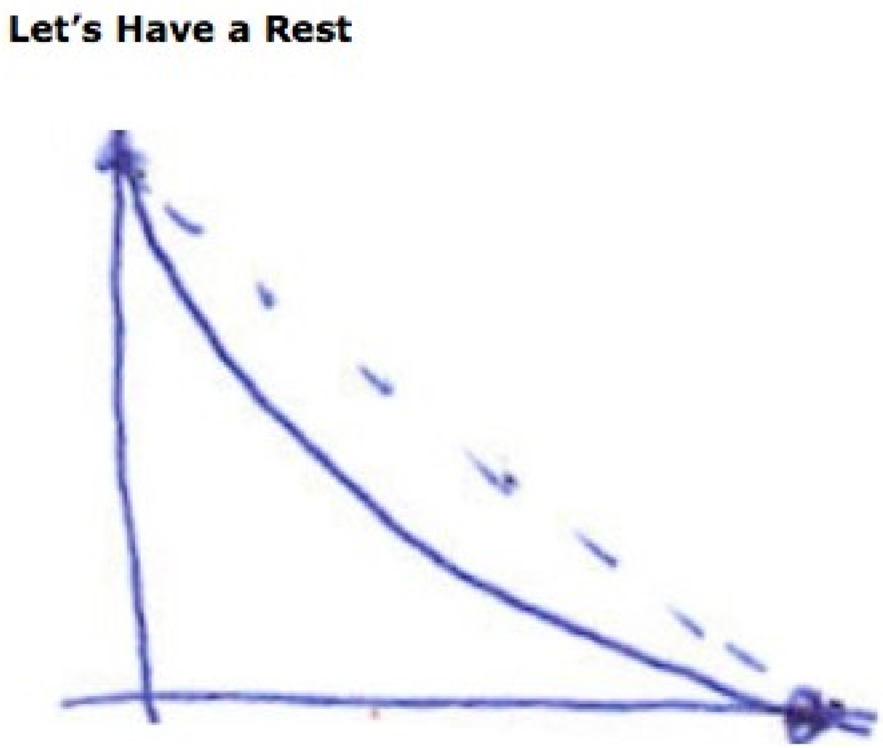


The team finishes its work sooner than expected. The stories were implemented, but the team didn't work on additional stories even it had the capacity to do it.

The stories were probably overestimated, therefore the team finished them earlier. Also the velocity of the team has not been probably estimated correctly.

The Scrum Master must be more proactive in either getting the team to fix estimation or ensure additional stories are ready to be added into the current sprint.

What sprint burndown chart says

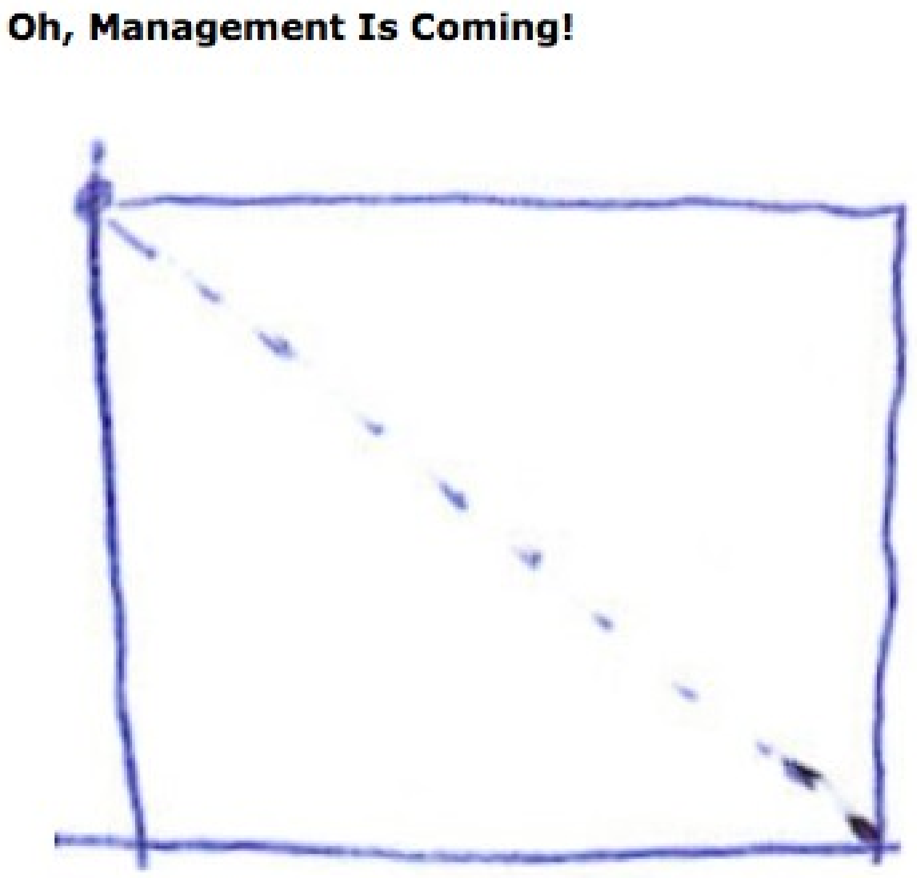


The team with such progress has a problem. The problem is either the team committed to less than they are able to complete or the product owner does not provide enough stories for the sprint.

The reason might be also an over-estimation of complexity, which ends up in completion earlier than expected at the beginning of the sprint.

The Scrum Master should identify this problem earlier and ask the product owner to provide the team with more work. Even if stories are over-estimated, the team should at least continue with stories from the next, already preplanned, sprint.

What sprint burndown chart says

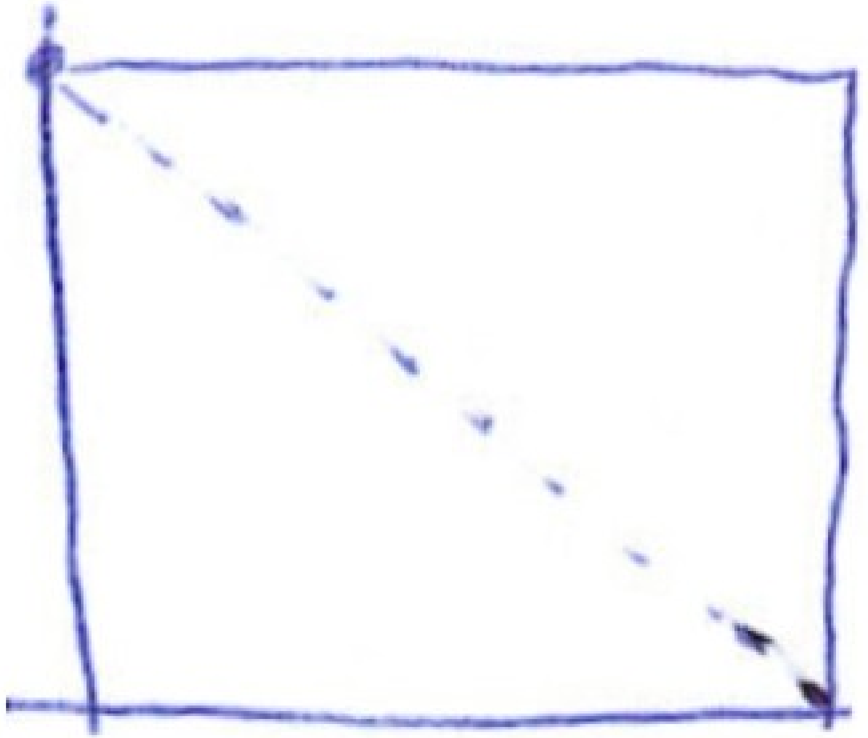


- The team is probably doing some work, but maybe it does not update its progress accordingly.
- Another reason might be that the product owner has added the same amount of work that was already completed, therefore the line is straight.
- The team is not able to predict the end of the sprint or even to provide the status of the current sprint.

The Scrum Master should improve it Scrum masterships and coach the team on why it is necessary to track the progress and how to track it. Such team should be stopped after two or three days that shows a flat the line of progress and should immediately apply corrective actions.

What sprint burndown chart says

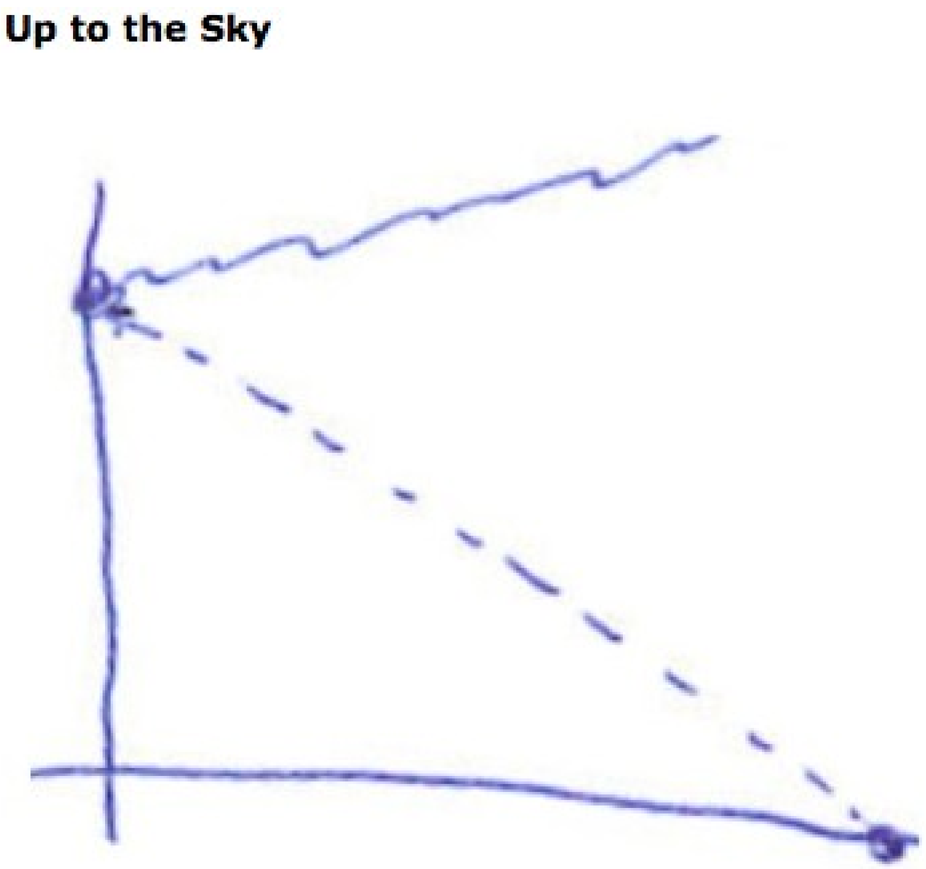
Oh, Management Is Coming!



- *The team is non-functional on many levels.*
- The Scrum Master of this team is not able to coach the team why it is necessary to track progress on daily basis.
- The product owner does not care about development progress either.

To fix this situation the team should restart. *Restart from scratch* by training and do a retrospective to figure out why this is happening.

What sprint burndown chart says



<u>*The first sprint typically looks like that.*</u>

Stories or tasks were added into the sprint backlog everyday without any progress recorded.

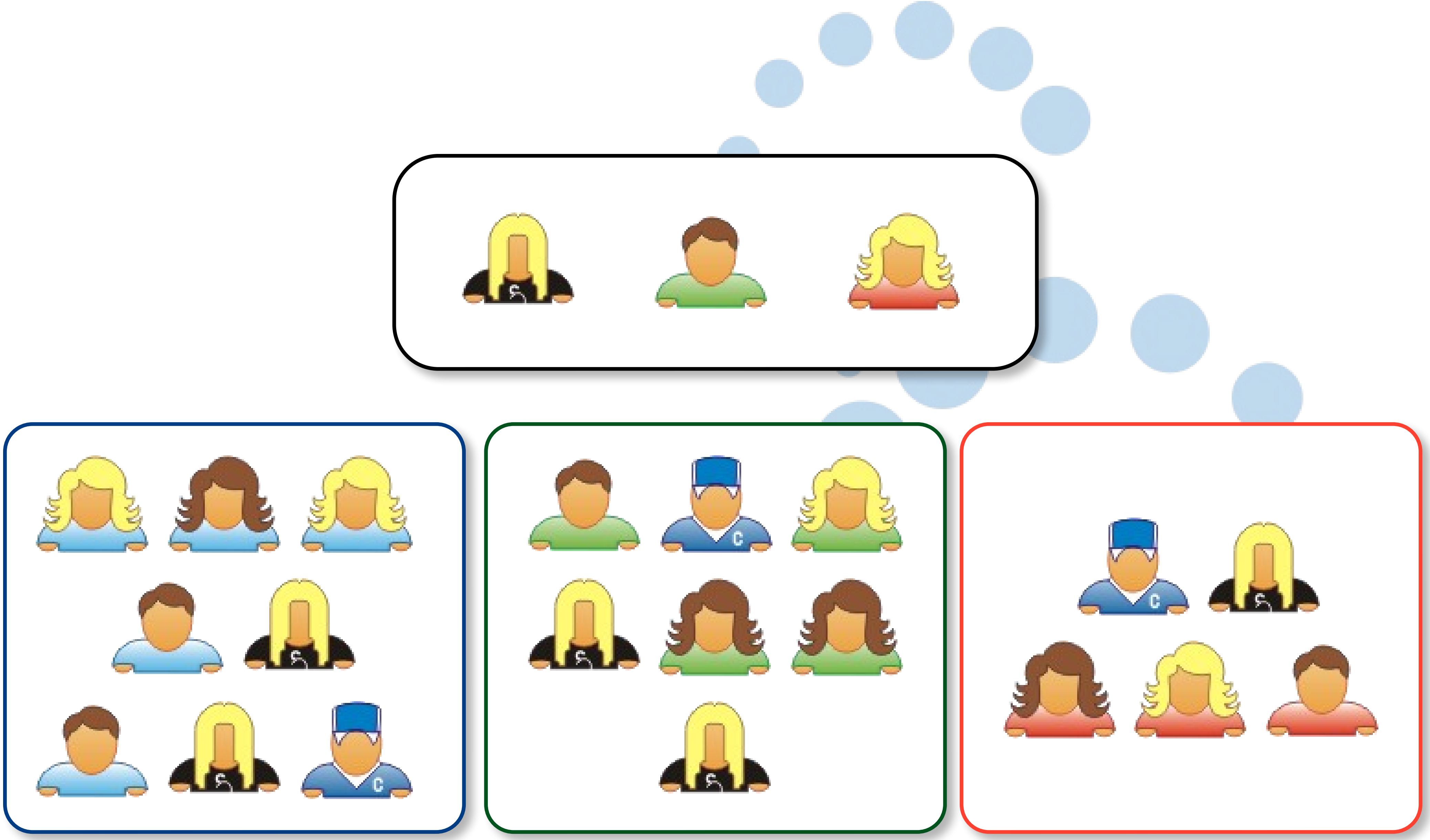
Another reason might be that tasks were re-estimated constantly during the sprint.

The mistake is that the team did not identify the problem: The sprint backlog should be reevaluated and rearranged immediately.The coach might be helpful, as an experienced Scrum master and product owner should often facilitate this situation.

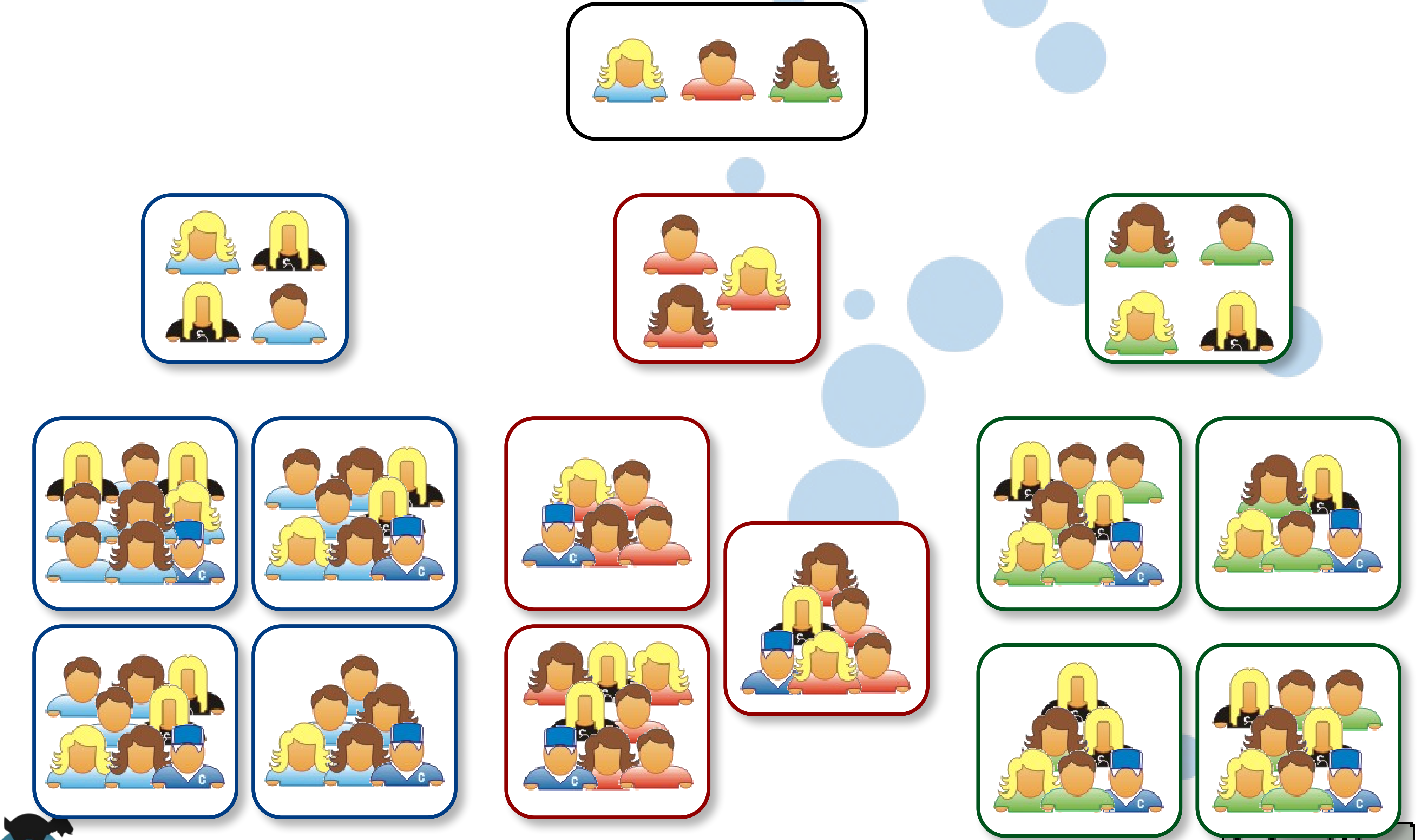
Scalability

- 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



Where to go next

- www.mountaingoatsoftware.com/scrum
- www.scrumalliance.org
- www.controlchaos.com
- scrumdevelopment@yahoogroups.com

A Scrum reading list

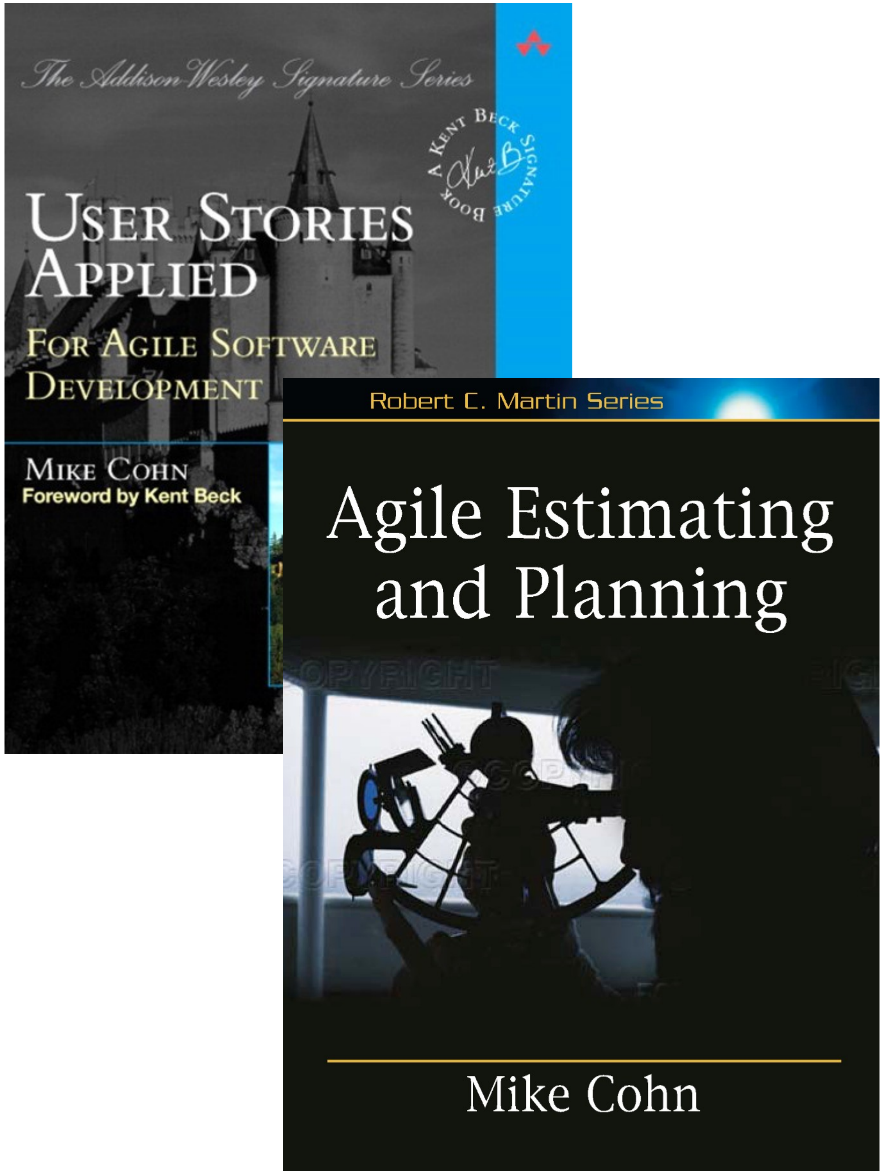
- *Agile and Iterative Development: A Manager's Guide* by Craig Larman
- *Agile Estimating and Planning* by Mike Cohn
- *Agile Project Management with Scrum* by Ken Schwaber
- *Agile Retrospectives* by Esther Derby and Diana Larsen

A Scrum reading list

- *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
- *Succeeding with Agile* by Mike Cohn
- *User Stories Applied for Agile Software Development* by Mike Cohn

Copyright notice

Contact information


The Addison-Wesley Signature Series
A Kent Beck Signature Book
User Stories Applied
For Agile Software Development
Mike Cohn
Foreword by Kent Beck
Robert C. Martin Series
Agile Estimating and Planning
Mike Cohn

Presentation by: Mike Cohn
mike@mountaingoatsoftware.com
www.mountaingoatsoftware.com
(720) 890-6110 (office)

You can remove this (or any slide) but you must credit the source somewhere in your presentation. Use the logo and company name (as at bottom left, for example) or include a slide somewhere saying that portions (or all) of your presentation are from this source. Thanks.