An Introduction to Agile Estimating and Planning

Mike Cohn
12 March 2008

Mike Cohn - background

Agile coach and trainer

- Founding member and director of Agile Alliance and Scrum Alliance
- Founder of Mountain Goat Software
- Ran my first Scrum project back in 1995
- Typical programmer to manager, etc. progression

© 2003–2007 Mountain Goat Software®

Agenda

- The right units for estimating
- How to estimate
- Release planning
- Planning with multiple teams

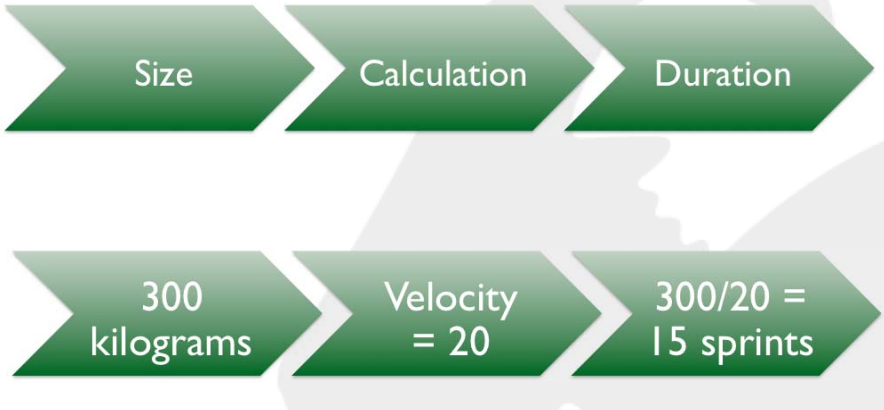
© 2003–2007 Mountain Goat Software®

How long will it take…

- …to read the latest Harry Potter book?
- …to drive to Paris?

© 2003–2007 Mountain Goat Software®

Estimate size; derive duration



© 2003–2007 Mountain Goat Software®

Measures of size

© 2003–2007 Mountain Goat Software®

Ideal days

- How long something would take if
  - it's all you worked on
  - you had no interruptions
  - and everything you need is available
- The ideal time of a basketball game is 40 minutes
  - Four 10-minute quarters
- The elapsed time is much longer (2+ hours)

© 2003–2007 Mountain Goat Software®

Story points

As a user, I want to be able to have some but not all items in my cart gift wrapped.

- The "bigness" of a task
  - Influenced by
  - How hard it is
  - How much of it there is
- Relative values are what is important:
  - A login screen is a 2.
  - A search feature is an 8.
- Points are unit-less

© 2003–2007 Mountain Goat Software®

Zoo points

What value in "zoo points" would you put on these zoo animals?

Lion
Kangaroo
Rhinocerus
Bear
Giraffe
Gorilla
Hippopotamus
Tiger

© 2003–2007 Mountain Goat Software®

Comparing the approaches

- Story points help drive cross-functional behavior
- Story point estimates do not decay
- Story points are a pure measure of size
- Estimating in story points is typically faster
- My ideal days cannot be added to your ideal days
- Ideal days are easier to explain outside the team
- Ideal days are easier to estimate at first
- Ideal days can force companies to confront time wasting activities

© 2003–2007 Mountain Goat Software®

The problem with mixing units

As a frequent flyer, I want…	30
As a frequent flyer, I want…	20
As a frequent flyer, I want…	60
As a frequent flyer, I want…	40
As a frequent flyer, I want…	20

Code the…	12
Design the…	10
Automate…	5
Test the…	8

© 2003–2007 Mountain Goat Software®

Agenda

- [x] The right units for estimating
- [ ] How to estimate
- [ ] Release planning
- [ ] Planning with multiple teams

© 2003–2007 Mountain Goat Software®

Estimate by analogy

- Comparing a user story to others
  - "This story is like that story, so its estimate is what that story's estimate was."
- Don't use a single gold standard
  - Triangulate instead
- Compare the story being estimated to multiple other stories

© 2003–2007 Mountain Goat Software®

Use the right units

- Can you distinguish a 1-point story from a 2?
  - How about a 17 from an 18?
- Use a set of numbers that make sense; I like:
  - 1, 2, 3, 5, 8, 13
- Stay mostly in a 1-10 range
- Nature agrees:
  - Musical tones and volume are distinguishable on a logarithmic scale

Use 0 and ½ if you like

© 2003–2007 Mountain Goat Software®

Planning poker

- An iterative approach to estimating
- Steps
  - Each estimator is given a deck of cards, each card has a valid estimate written on it
  - Customer/Product owner reads a story and it's discussed briefly
  - Each estimator selects a card that's his or her estimate
  - Cards are turned over so all can see them
  - Discuss differences (especially outliers)
  - Re-estimate until estimates converge

© 2003–2007 Mountain Goat Software®

Planning poker—an example

Estimator	Round 1	Round 2
Erik	3	5
Martine	8	5
Inga	2	5
Tor	5	8

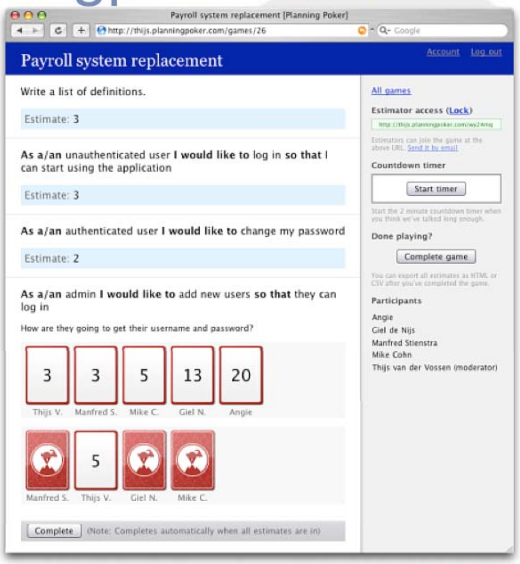
© 2003–2007 Mountain Goat Software®

Estimate these

Product backlog item	Estimate
Read a high-level, 10-page overview of agile software development in a celebrity magazine.	
Read a densely written 5-page research paper about agile software development in an academic journal.	
Write the product backlog for a simple eCommerce site that sells only clocks.	
Recruit, interview, and hire a new member for your team.	
Create a 60-minute presentation about agile estimating and planning for your coworkers.	
Wash and wax your boss' Porsche.	
Read a 150-page book on agile software development.	
Write an 8-page description of agile development for your boss.	

© 2003–2007 Mountain Goat Software®

www.planningpoker.com



Free, or I wouldn't mention it

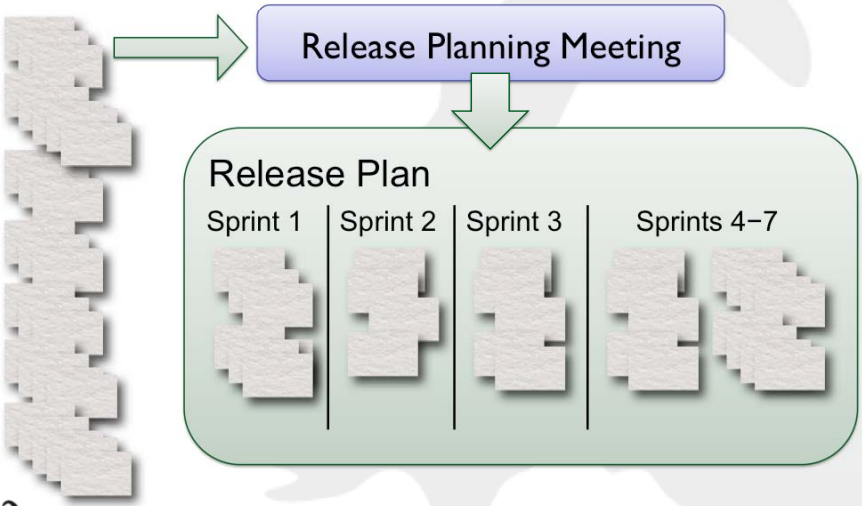
© 2003–2007 Mountain Goat Software®

Agenda

- ☑ The right units for estimating
- ☑ How to estimate
- ☐ Release planning
- ☐ Planning with multiple teams

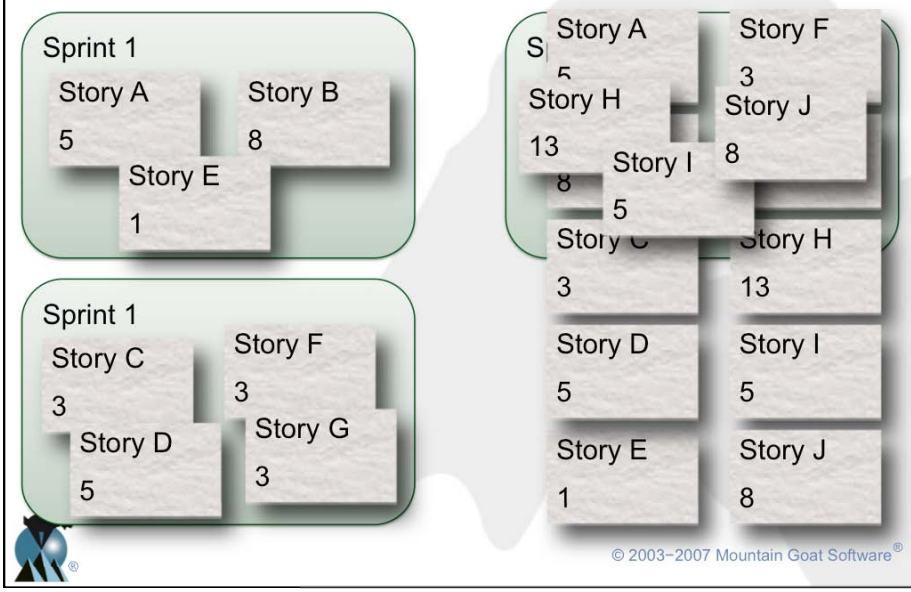
© 2003–2007 Mountain Goat Software®

Release planning



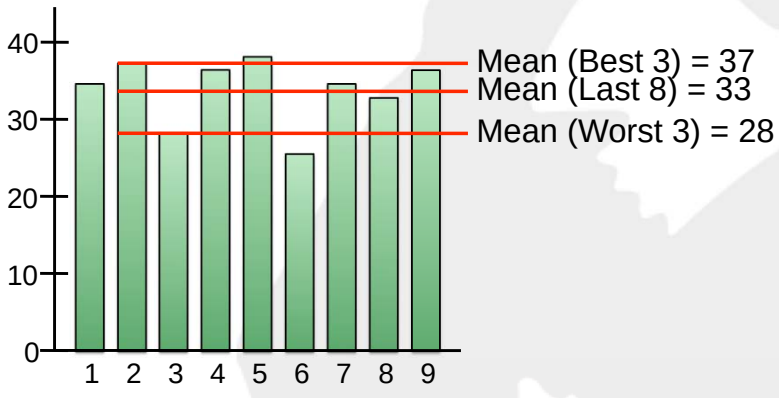
© 2003–2007 Mountain Goat Software®

An example with velocity = 14



© 2003–2007 Mountain Goat Software®

Projections based on velocity



© 2003–2007 Mountain Goat Software®

Extrapolate from velocity

Assume 5 sprints left

- At our slowest velocity, we'll end here (5 × 28)
- At our average velocity, we'll end here (5 × 33)
- At our average velocity, we'll end here (5 × 37)

© 2003–2007 Mountain Goat Software®

Fixed-date planning

How much can I get by <date>?

- Determine how many sprints you have
- Estimate velocity as a range
- Multiply low velocity × number of sprints
  - Count off that many points; These are "Will Have" items
- Multiply high velocity × number of sprints
  - Count off that many more points; these are "might haves"

© 2003–2007 Mountain Goat Software®

Fixed-date planning example

Desired release date	30 June
Today's date	1 January
Number of sprints	6 (monthly)
Low velocity	15
High velocity	20

Will have

6 × 15

Might have

6 × 20

Won't have

© 2003–2007 Mountain Goat Software®

Agenda

- ☑ The right units for estimating
- ☑ How to estimate
- ☑ Release planning
- ☐ Planning with multiple teams

© 2003–2007 Mountain Goat Software®

Three issues

1. Estimating in a common unit
2. Sprint planning
3. Dependencies

© 2003–2007 Mountain Goat Software®

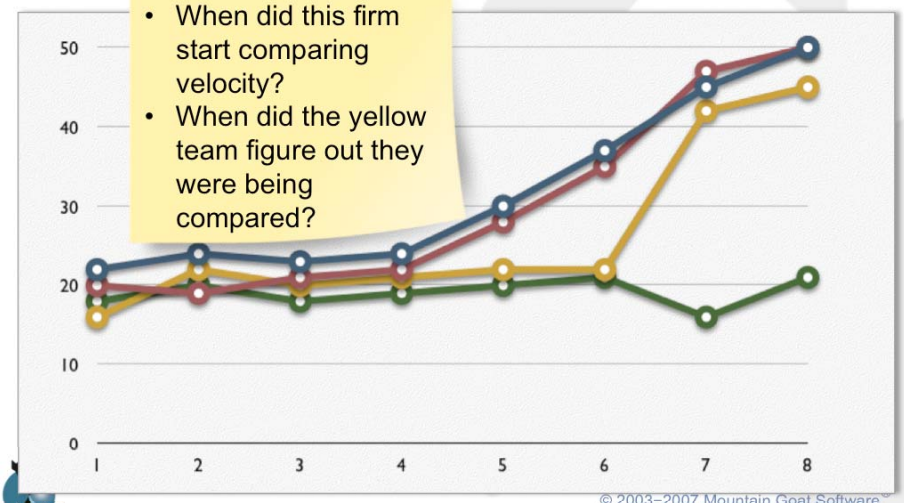
Establish a common baseline

- All teams should agree on story points or ideal days
- Establish a common baseline
  - Select a dozen or so user stories that were done recently or are on the product backlog
  - Estimate them en masse with Planning Poker

© 2003–2007 Mountain Goat Software®

Be careful with cross-team comparisons

- When did this firm start comparing velocity?
- When did the yellow team figure out they were being compared?



© 2003–2007 Mountain Goat Software®

Two approaches to sprint planning

1. Stagger by a day

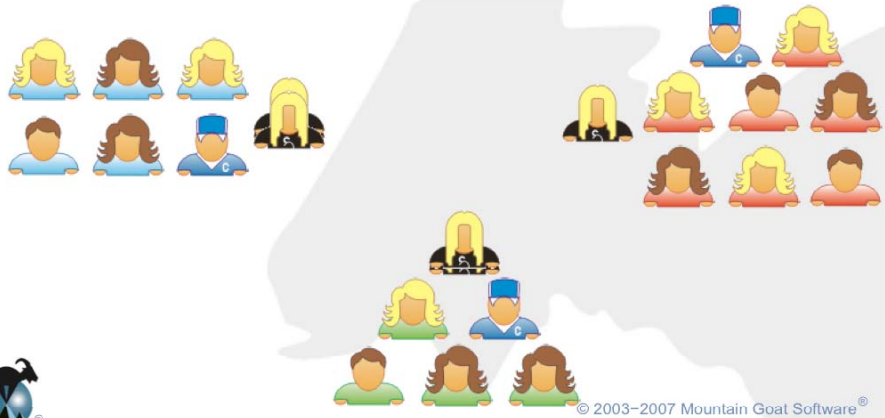
- Sprints end by ± a day
- Helps a key resource (e.g., a product owner or architect) fully participate in many planning meetings

© 2003–2007 Mountain Goat Software®

2

The Big Room

- All sprints end on same day
- All planning is on same day and in one room
- Key resources shift between teams on demand



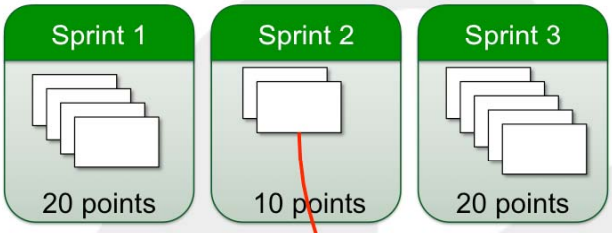
© 2003–2007 Mountain Goat Software®

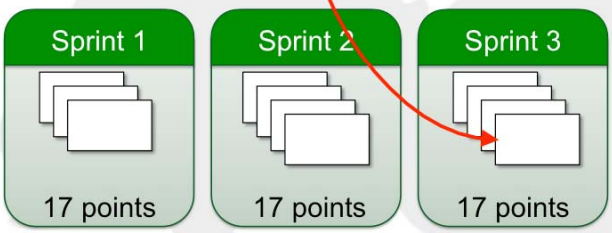
Dependencies

- Critical dependencies between teams
  - Must be done in this order and likely to influence overall ship date
  - Fewer of these than you may think
- Emergent dependencies
  - "OK, we're going to start on such-and-such soon. As you know we need this-and-that first."

© 2003–2007 Mountain Goat Software®

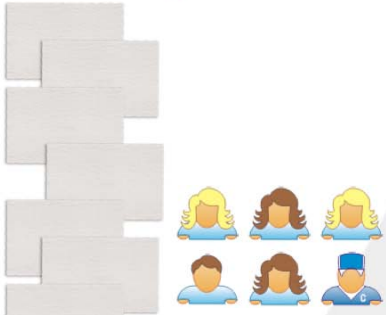
Buffer critical dependencies





© 2003–2007 Mountain Goat Software®

Rolling lookahead planning

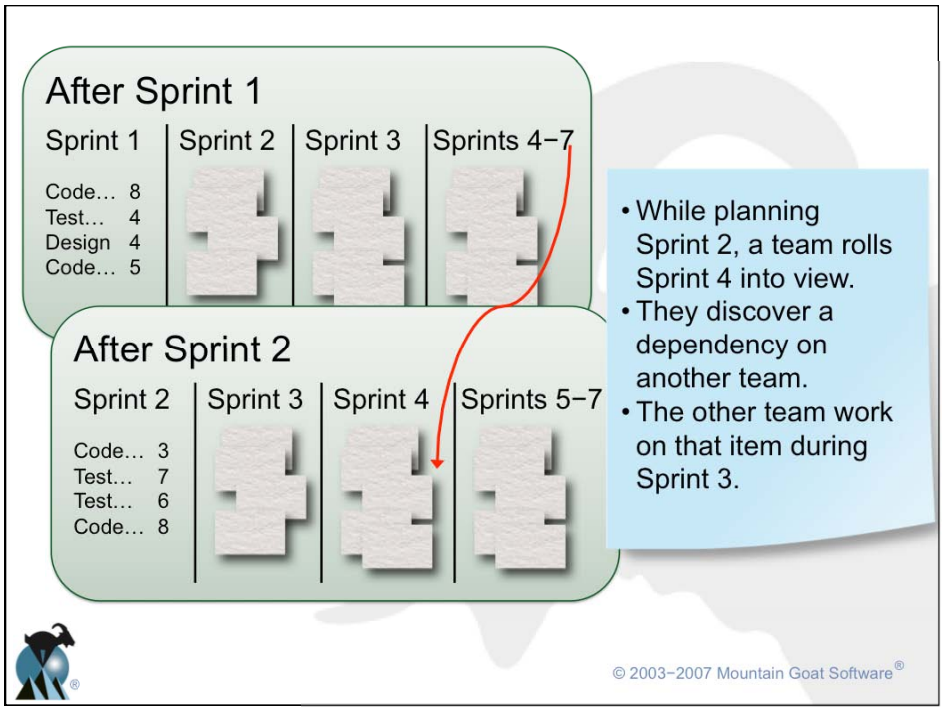


Sprint 1

Task	Hours
Code the …	8
Test the …	16
Integrate with …	5
Code the …	8
Design the …	4

Sprint 2 | Sprint 3

© 2003–2007 Mountain Goat Software®


After Sprint 1
Sprint 1
Sprint 2
Sprint 3
Sprints 4–7
Code… 8
Test… 4
Design 4
Code… 5
After Sprint 2
Sprint 2
Sprint 3
Sprint 4
Sprints 5–7
Code… 3
Test… 7
Test… 6
Code… 8
• While planning Sprint 2, a team rolls Sprint 4 into view.
• They discover a dependency on another team.
• The other team work on that item during Sprint 3.
© 2003–2007 Mountain Goat Software®

Mike Cohn contact info
Robert C. Martin Series
Agile Estimating and Planning
Mike Cohn
User Stories Applied
For Agile Software Development
Mike Cohn
mike@mountaingoatsoftware.com
www.mountaingoatsoftware.com
(720) 890-6110
MOUNTAIN GOAT SOFTWARE
© 2003–2007 Mountain Goat Software®