An Introduction to Agile Estimating and Planning

Mike Cohn
February 28, 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

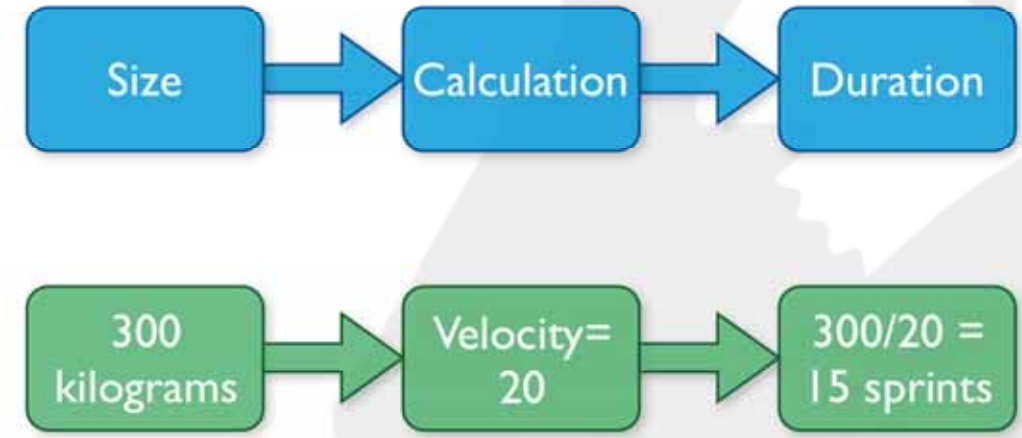
Agenda

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

How long will it take...

- ...to read the latest Harry Potter book?
- ...to drive to Austin?

Estimate size; derive duration



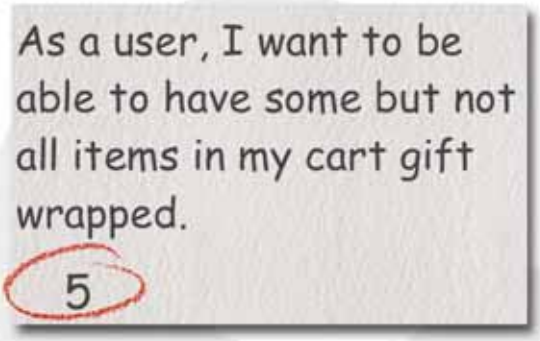
Measures of size

- Traditional and agile measure size differently

Ideal time

- 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 football game is 60 minutes
  - Four 15-minute quarters
- The elapsed time is much longer (3+ hours)

Story points



- 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

Dog points

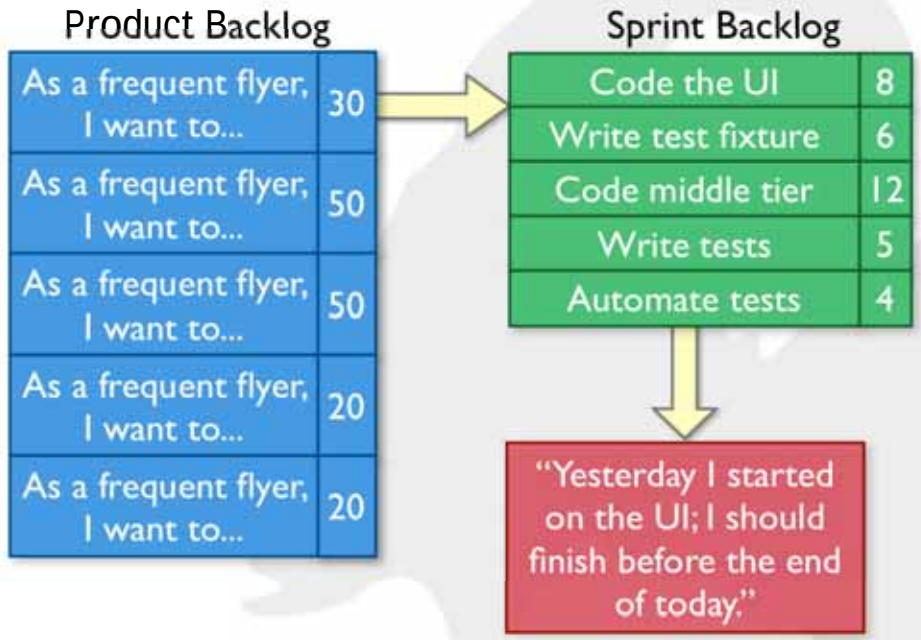
What value in "dog points" would you put on these breeds?

Labrador retriever
Dachshund
Great Dane
German Shepherd
Poodle
St. Bernard

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

The problem with mixing units



Agenda

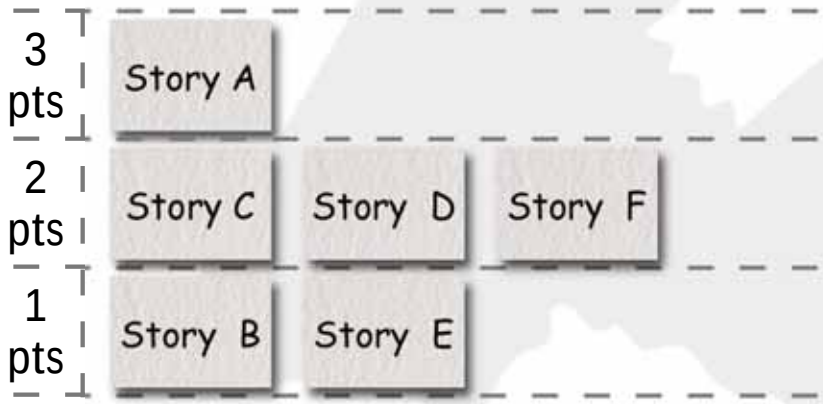
- [x] The right units for estimating
- [ ] How to estimate
- [ ] Release Planning
  - [ ] Fixed-date projects
  - [ ] Planning with multiple teams

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

Triangulation

- Confirm estimates by comparing the story to multiple other stories.
- Group like-sized stories on table or whiteboard



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

Include 0 and ½ if you want

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

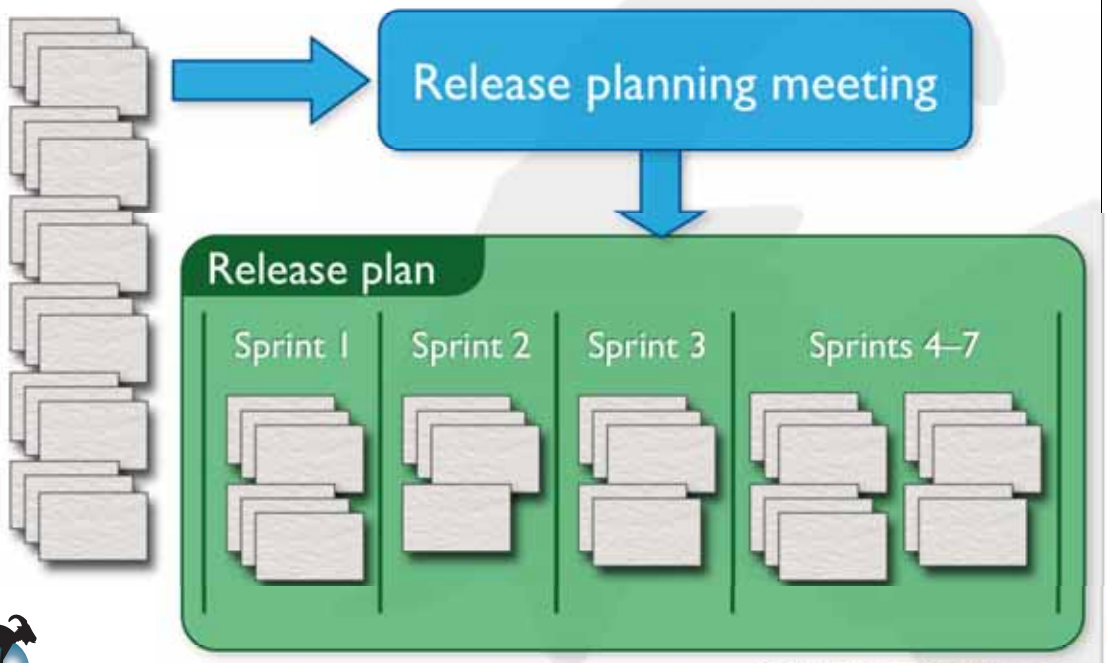
Planning poker - an example

Estimator	Round 1	Round 2
Susan	3	5
Vadim	8	5
Ann	2	5
Chris	5	8

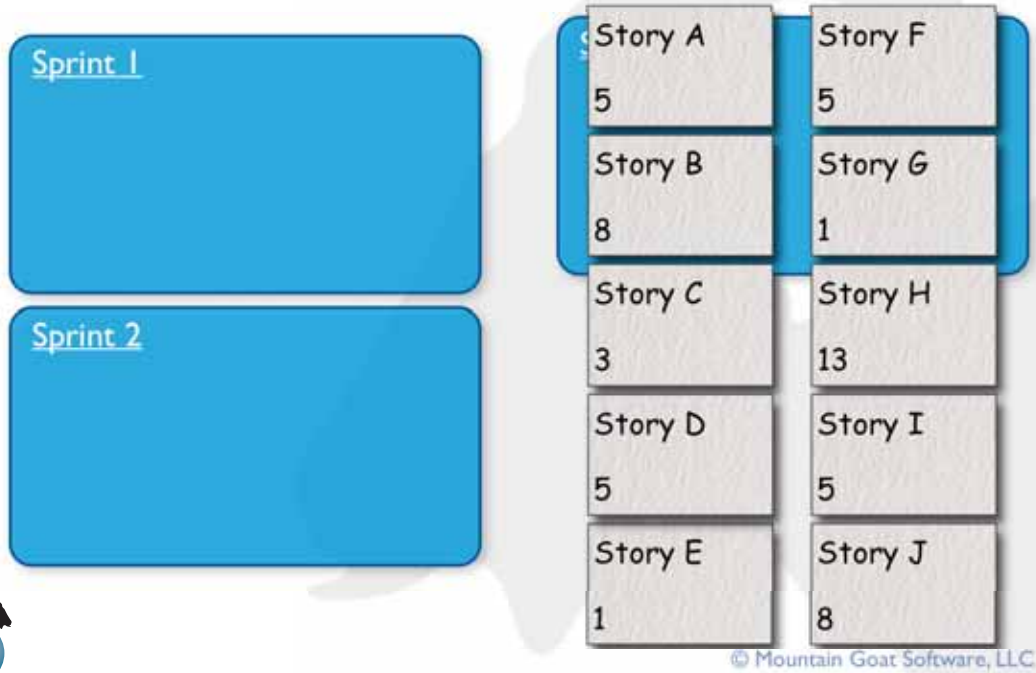
Agenda

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

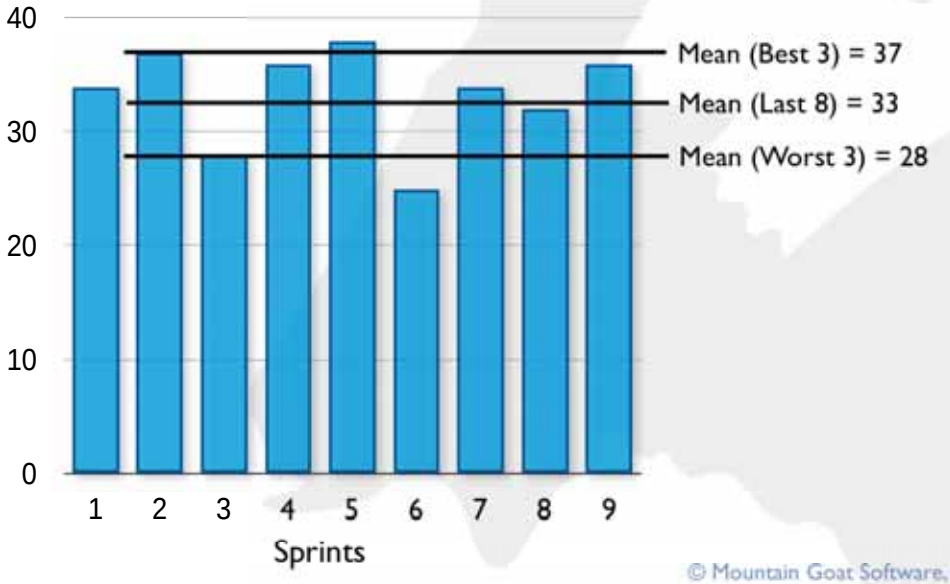
Release planning



An example with velocity=14



Projections from velocity



Extrapolate from velocity

Assume:
There are five sprints left.

At our slowest velocity we'll finish here (5×28)

At our long-term average we'll finish here (5×33)

At our best velocity we'll finish here (5×37)

Consider drawing an arrow at most recent velocity as well.

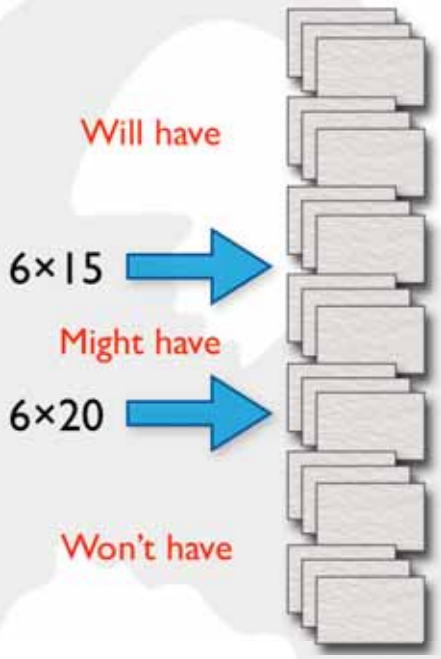
Fixed-date planning

How much can I get by <date>?

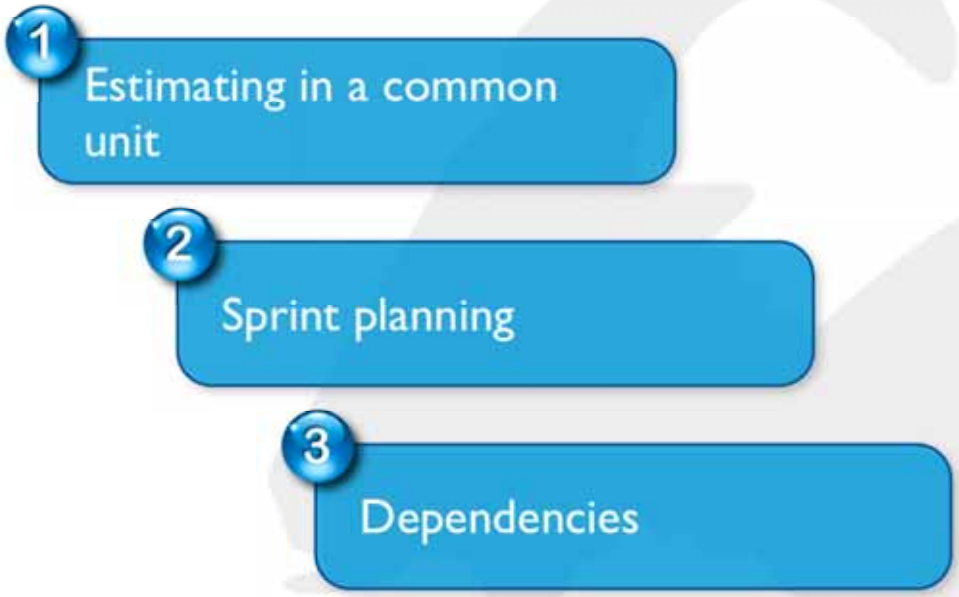
1. Determine how many sprints you have
2. Estimate velocity as a range
3. Multiply low velocity × number of sprints
   - Count off that many points
     - These are "Will Have" items
4. Multiply high velocity × number of sprints
   - Count off that many more points
     - These are "Might Have items"

Fixed-date planning: an example

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



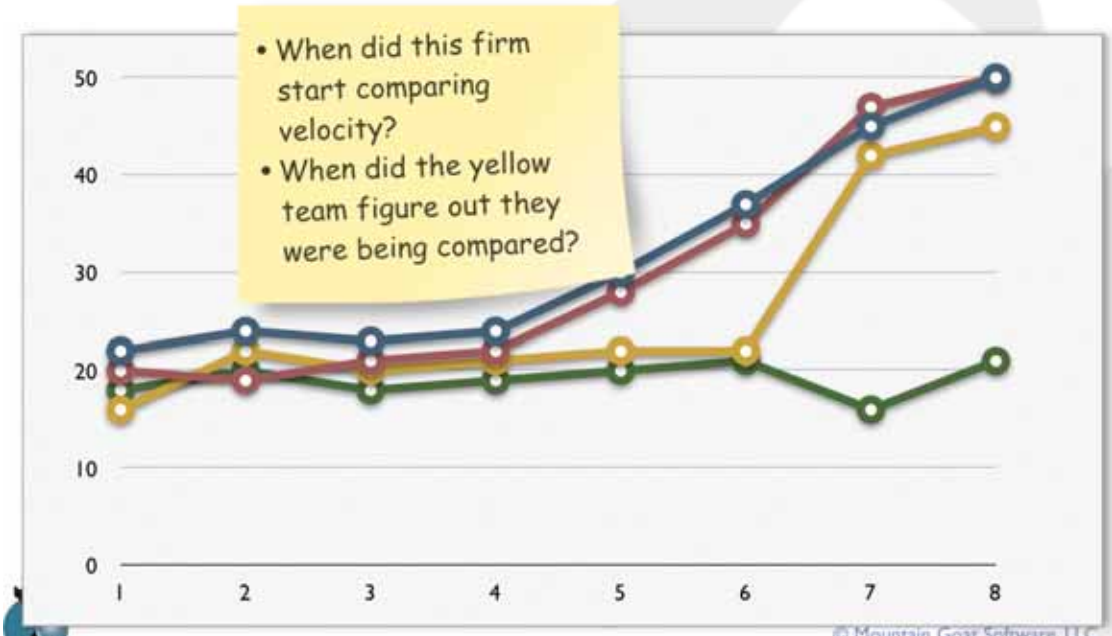
Planning with multiple teams



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

Be careful with cross-team comparisons



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

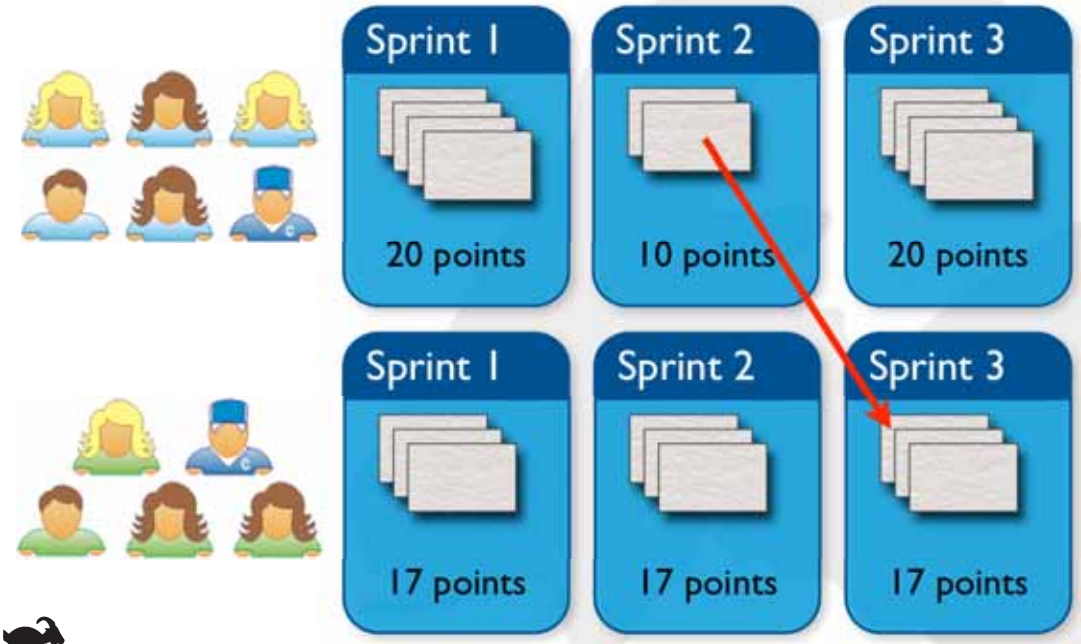
2 The Big Room

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

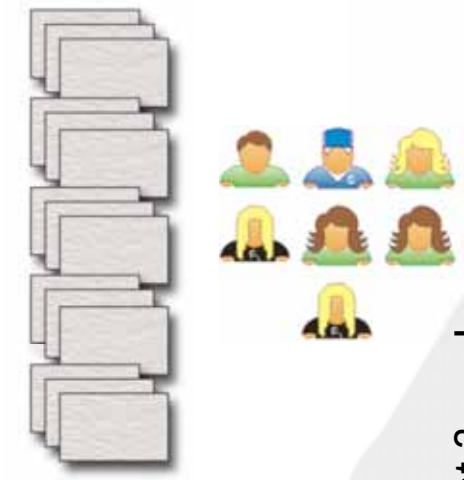
Dependencies

- 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."

Buffer critical dependencies



Rolling lookahead planning

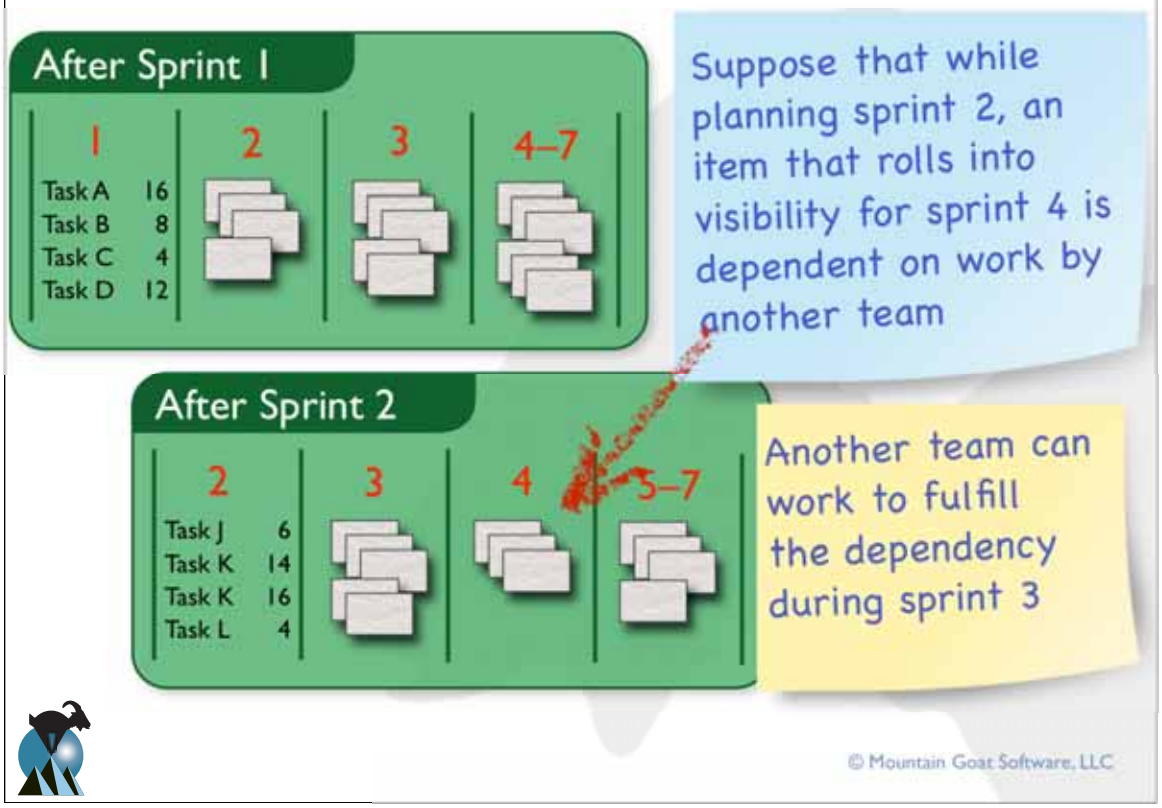


Sprint 1

Tasks	Est
Code the	8
Test the...	16
Integrate with the...	8
Code the ...	12
Design the	8

Sprint 2

Sprint 3



33

Mike Cohn contact info

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)

MOUNTAIN GOAT SOFTWARE

34