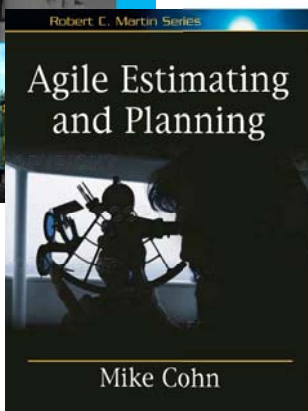


# Agile Estimating and Planning



1

## Mike Cohn - background



2

# Imagine...

- That you're fed up with software development as a career
- And you decide to go into the landscaping business
- Your first job is moving this pile of rock from the front of my house to the back



© Mountain Goat Software, LLC

3



# How might you estimate this?

- One way:
  - Look at the pile of rock and estimate how many wheelbarrow loads it represents
- After an hour, see how many wheelbarrow loads you've moved then extrapolate the total duration



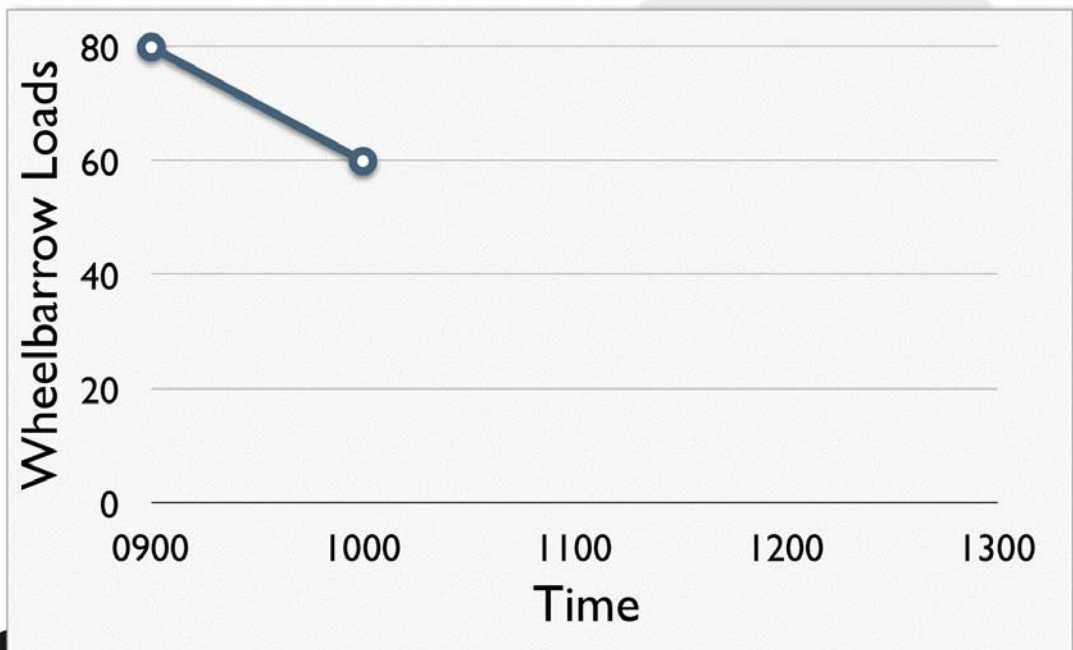
- I think that's 80 wheelbarrow loads
- After an hour I've moved 20 loads
- So, I'll be done in a total of 4 hours

© Mountain Goat Software, LLC

4

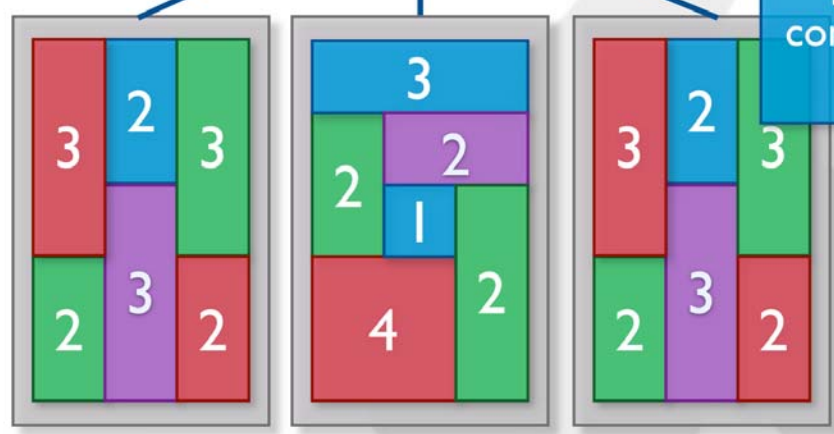


# My landscaping



• An iteration is a short, constrained period of time  
• Typically 1-4 weeks

Velocity is the amount of work planned or completed in an iteration.



A release typically comprises more than one iteration



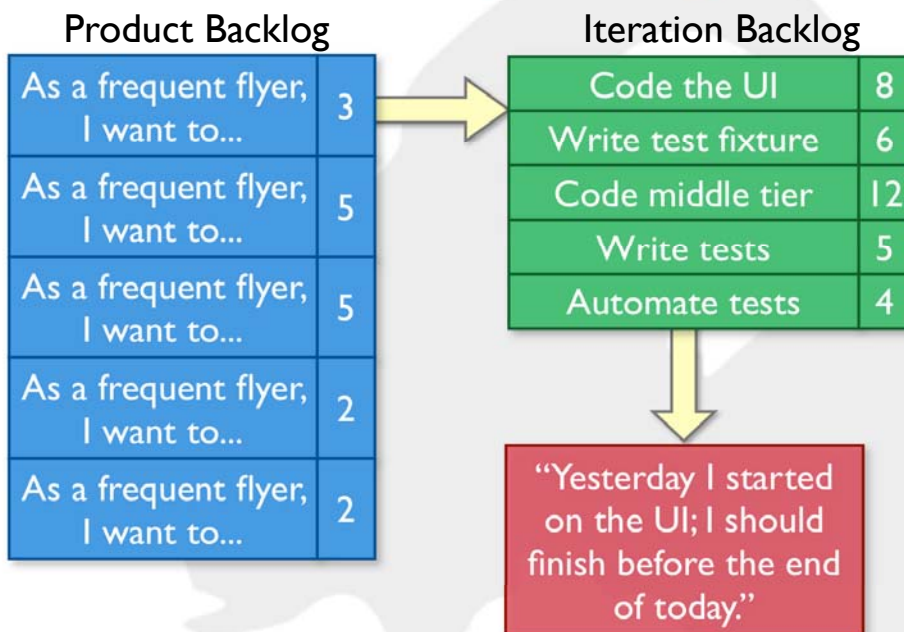
# The planning onion



- Agile teams plan at the innermost three levels.
- Others (on the team in the company) plan at the outer levels.

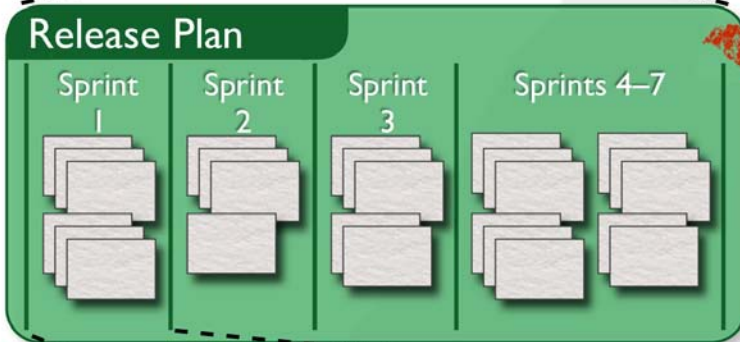


# Relating the different planning levels





# Product, release, sprint planning



We'll focus here today

Task A	8 hours
Task B	16 hours
Task C	5 hours
Task D	8 hours



## Agenda



- Estimating
- Release planning



# Story points

- Probably the most commonly used estimating unit among agile teams today
  - Name is derived from agile teams commonly expressing requirements as “user stories”
- Based on a combination of the size and complexity of the work
- Unitless but numerically relevant estimates
  - A 10-point user story is expected to take twice as long as a 5-point user story



# Dog points

Assign “dog points” to the following dogs

Labrador retriever  
Dachshund  
Great Dane  
Terrier  
German Shepherd  
Poodle  
St. Bernard  
Bulldog



## Consider these two piles of work



What story point values might we put on these?



© Mountain Goat Software, LLC

13

## Three key advantages

- Estimating in story points:
  1. Forces the use of relative estimating
    - Studies have shown we're better at this<sup>†</sup>
  2. Focuses us on estimating the size, not the duration
    - We derive duration empirically by seeing how much we complete per iteration
  3. Puts estimates in units that we can add together
    - Time based estimates are not additive

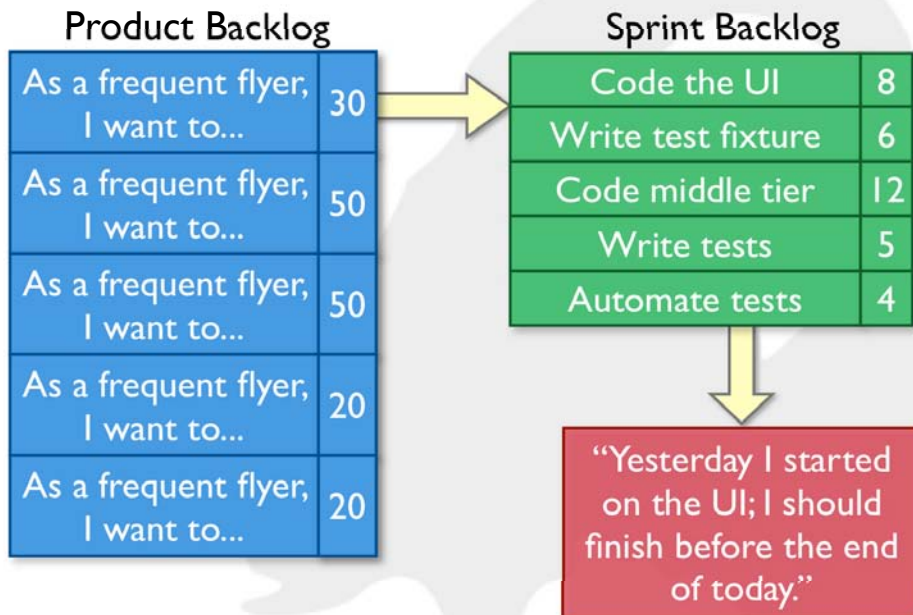
<sup>†</sup>Lederer and Prasad, 1998. *A Causal Model for Software Cost Estimating Error* and Vicinanza et al., 1991. *Software Effort Estimation: An Exploratory Study of Expert Performance*.



© Mountain Goat Software, LLC

14

# Comparing apples to apples



# 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

© Mountain Goat Software, LLC

17

# Why planning poker works

- Those who will do the work, estimate the work<sup>1</sup>
- Estimators are required to justify estimates<sup>2, 3</sup>
- Focuses most estimates within an approximate one order of magnitude<sup>4, 5</sup>

<sup>1</sup>Jørgensen, Magne. 2004. *A Review of Studies on Expert Estimation of Software Development Effort*.

<sup>2</sup>Hagafors, R., and B. Brehmer. 1983. *Does Having to Justify One's Decisions Change the Nature of the Decision Process?*

<sup>3</sup>Brenner, et al. 1996. *On the Evaluation of One-sided Evidence*.

<sup>4</sup>Miranda, Eduardo. 2001. *Improving Subjective Estimates Using Paired Comparisons*.

<sup>5</sup>Saaty, Thomas. 1996. *Multicriteria Decision Making: The Analytic Hierarchy Process*.

© Mountain Goat Software, LLC

18

# Why planning poker works

- Combining of individual estimates<sup>6</sup> through group discussion<sup>7</sup> leads to better estimates
- Emphasizes relative rather than absolute estimating
- Estimates are constrained to a set of values so we don't waste time in meaningless arguments
- Everyone's opinion is heard
- It's quick and fun

<sup>6</sup>Hoest, Martin, and Claes Wohlin. 1998. *An Experimental Study of Individual Subjective Effort Estimations and Combinations of the Estimates.*

<sup>7</sup>Jørgensen, Magne, and Kjetil Moløkken. 2002. *Combination of Software Development Effort Prediction Intervals: Why, When and How?*



© Mountain Goat Software, LLC

19

## Reduces likelihood of anchoring

### Control group

- Given a product spec

• 456 hours

### High anchor group

- Given the same product spec
- Told the customer thinks 500 hours is a reasonable estimate but that
  - The customer knows very little about the implications of his spec on the estimate
  - You shouldn't let his number influence you

• 555 hours

### Low anchor group

- Same as high but customer thinks 50 hours

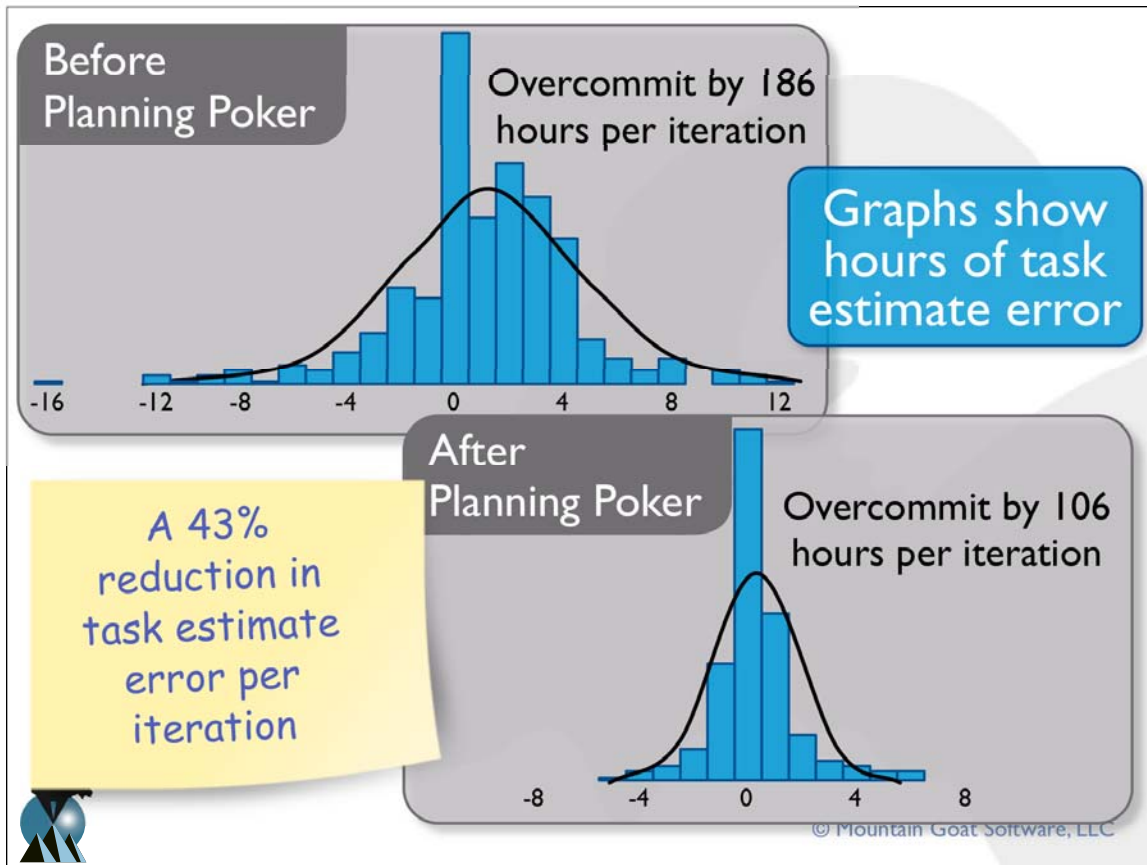
• 99 hours



Source: *How to avoid impact from irrelevant and misleading information on your cost estimates*, Magne Jørgensen and Stein Grimstad, Simula Research Laboratory, Simula Research Labs Estimation Seminar, Oslo, Norway 2006.

© Mountain Goat Software, LLC


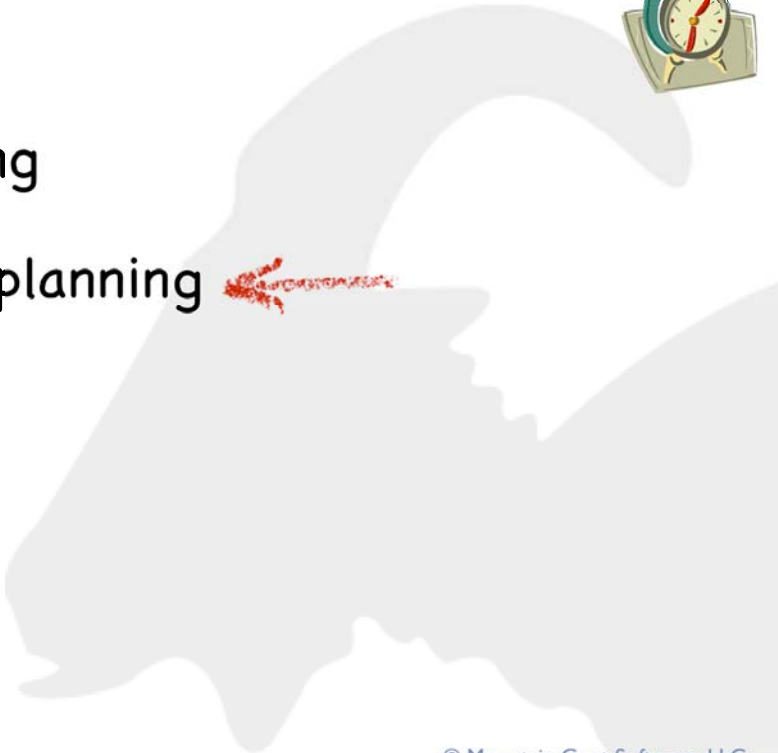
20



21

# Agenda

- Estimating
- Release planning

© Mountain Goat Software, LLC

22

# Release planning

## Purpose

To answer questions such as:

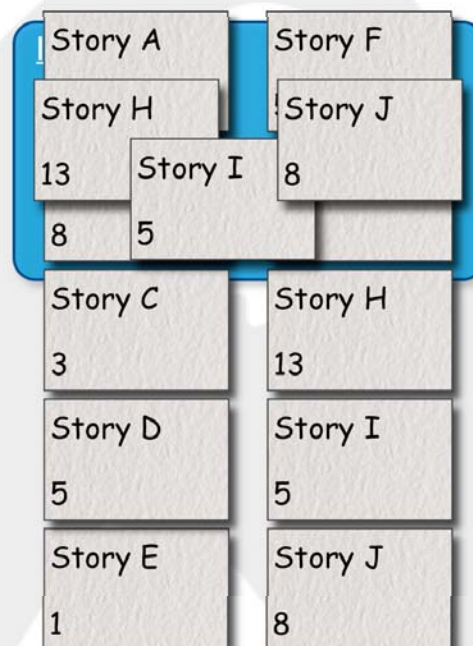
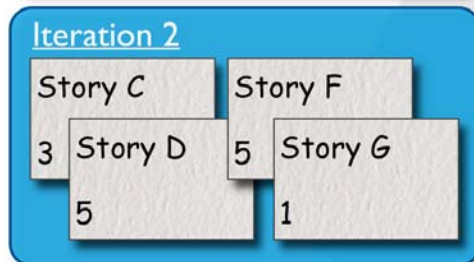
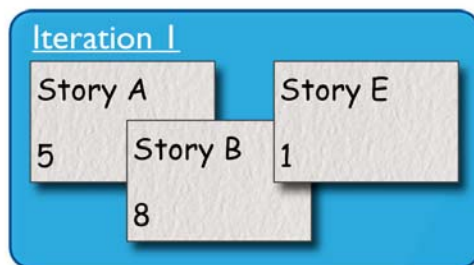
- How much will be done by 30 June?
- When can we ship with this set of features?
- How many people or teams should be on this project?

## Inputs

- Velocity
- The length of the project
- Prioritized product backlog



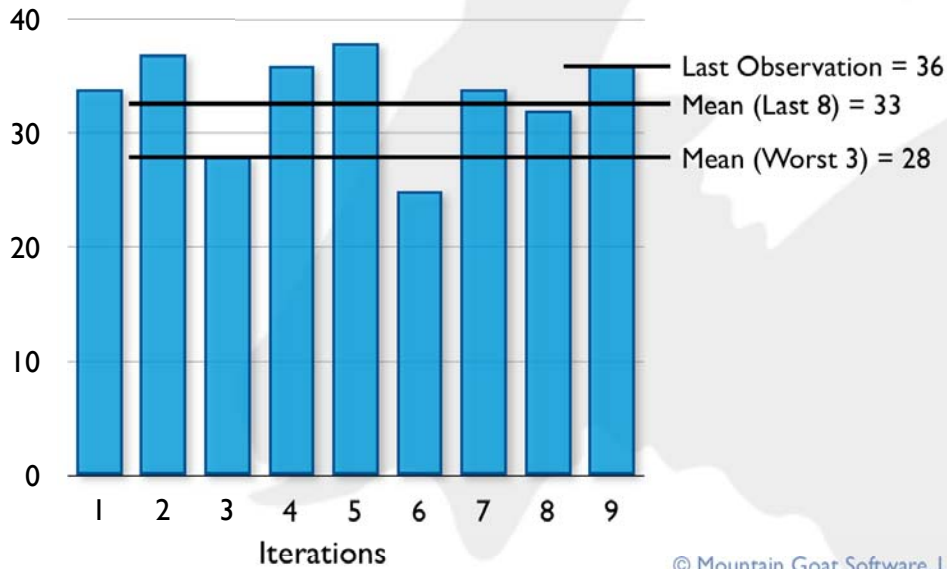
# An example with velocity=14





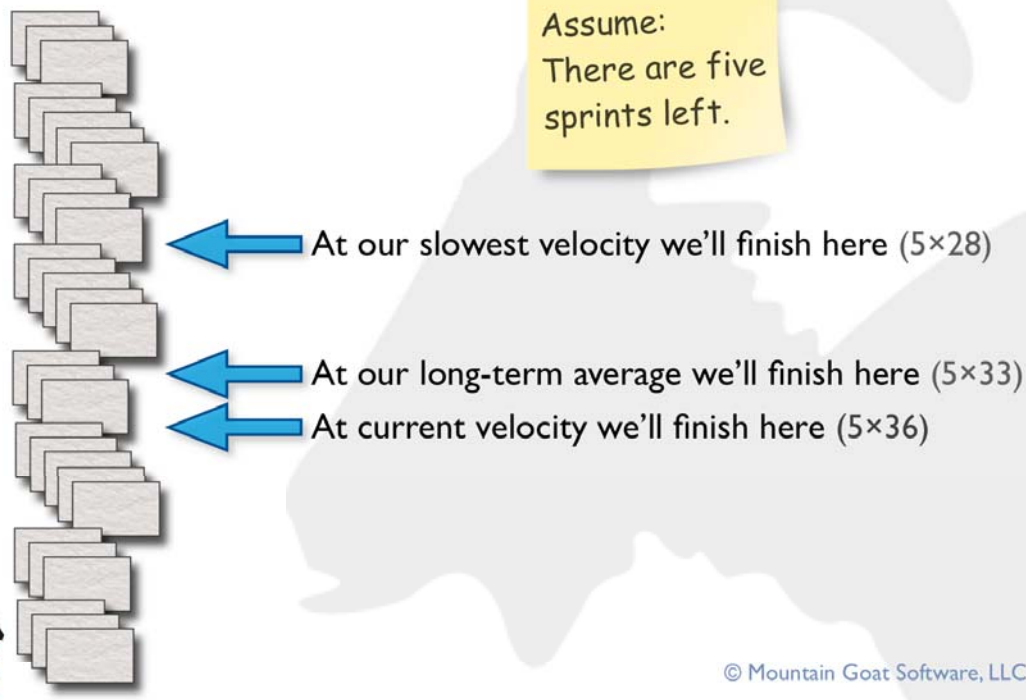
# Updating the release plan

- Use multiple views of observed velocity



25

# Extrapolate from velocity



26

# 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”

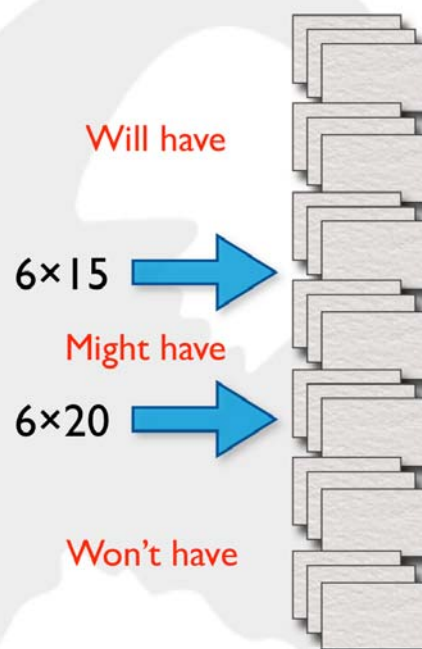


© Mountain Goat Software, LLC

27

## 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



© Mountain Goat Software, LLC

28

# Fixed-date contracting

Will have

6×15



Might have

6×20



Won't have



If you write a contract for just the will haves:

- You won't likely win the contract
- But you'll probably make money if you do

If you write a contract that includes the might haves:

- You will likely win the contract
- But probably not make money on it

It's a risk issue

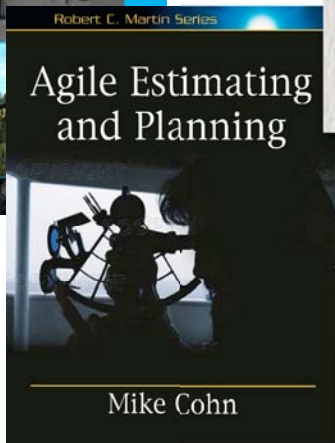
Where do you want to be?



© Mountain Goat Software, LLC

29

# Mike Cohn contact info



mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)



© Mountain Goat Software, LLC

30