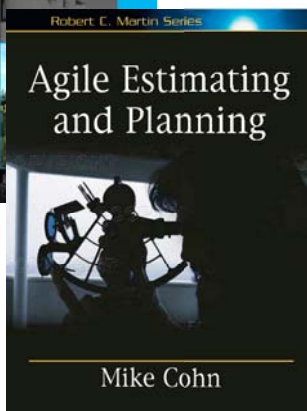


An Introduction to Agile Estimating and Planning



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



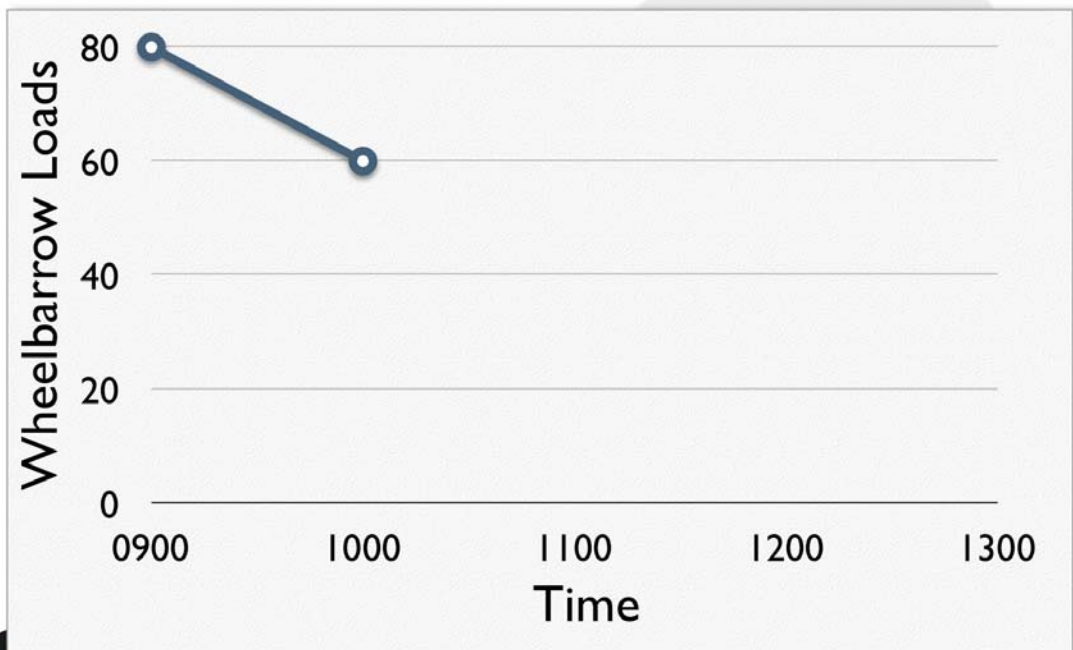
- I think that's 80 wheelbarrow loads
- After an hour I've moved 20 loads
- So, I'll be done in a total of 4 hours

© Mountain Goat Software, LLC

4

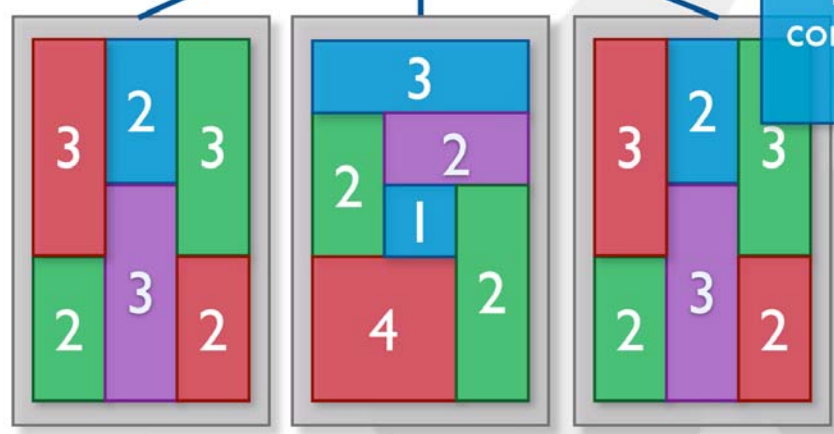


My landscaping



• 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



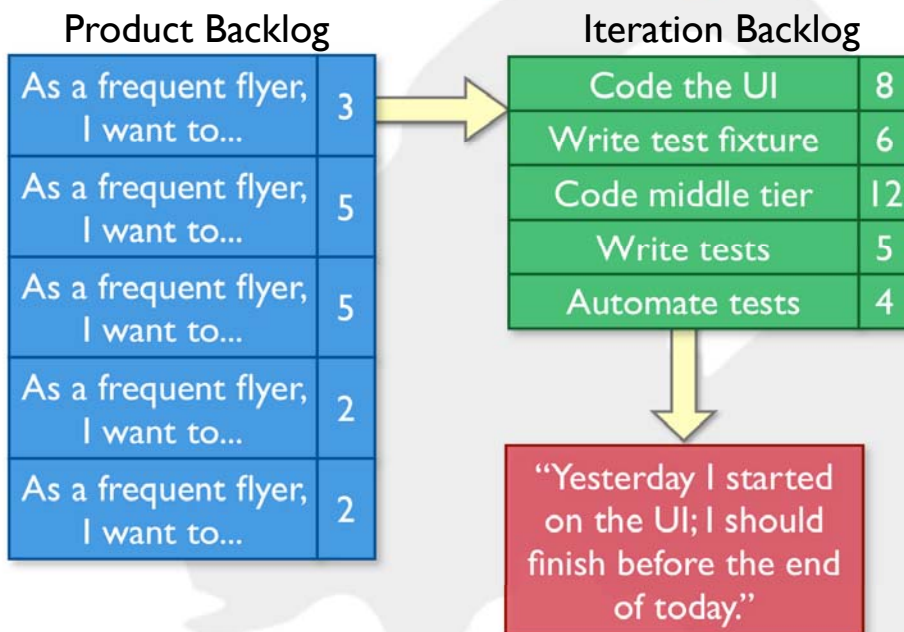
The planning onion



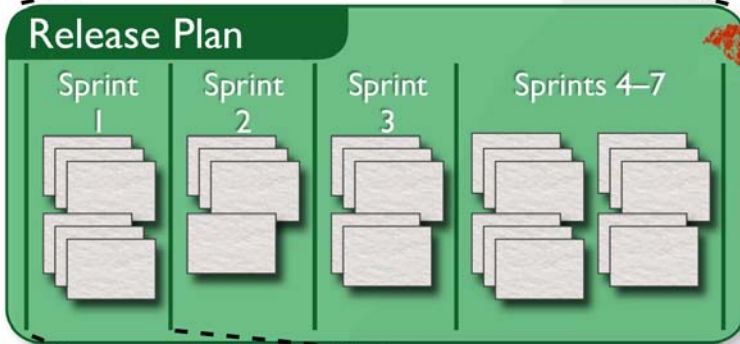
- Agile teams plan at the innermost three levels.
- Others (on the team in the company) plan at the outer levels.



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[†]
 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

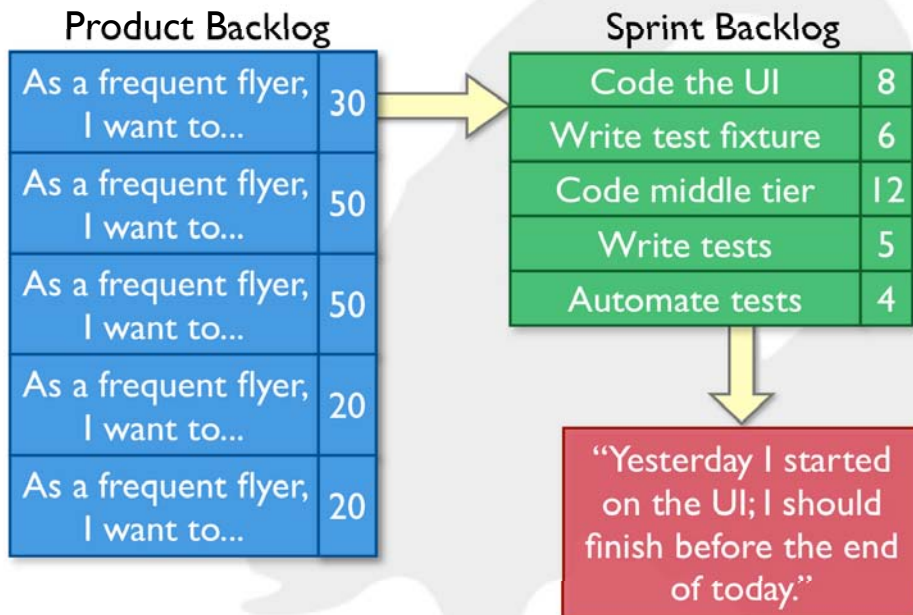
[†]Lederer and Prasad, 1998. *A Causal Model for Software Cost Estimating Error* and Vicinanza et al., 1991. *Software Effort Estimation: An Exploratory Study of Expert Performance*.



© 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

Estimate these



Product backlog item	Estimate
Read a high-level, 10-page overview of agile software development in <i>People</i> 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 to introduce agile software development to your non-agile team.	
Wash and wax your boss' Porsche.	
Read a 150-page book on agile software development.	
Write an 8-page summary of that book for your boss.	

© Mountain Goat Software, LLC

18

Why planning poker works

- Those who will do the work, estimate the work¹
- Estimators are required to justify estimates^{2, 3}
- Focuses most estimates within an approximate one order of magnitude^{4, 5}

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

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

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

⁴Miranda, Eduardo. 2001. *Improving Subjective Estimates Using Paired Comparisons*.

⁵Saaty, Thomas. 1996. *Multicriteria Decision Making: The Analytic Hierarchy Process*.



© Mountain Goat Software, LLC

19

Why planning poker works

- Combining of individual estimates⁶ through group discussion⁷ 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

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

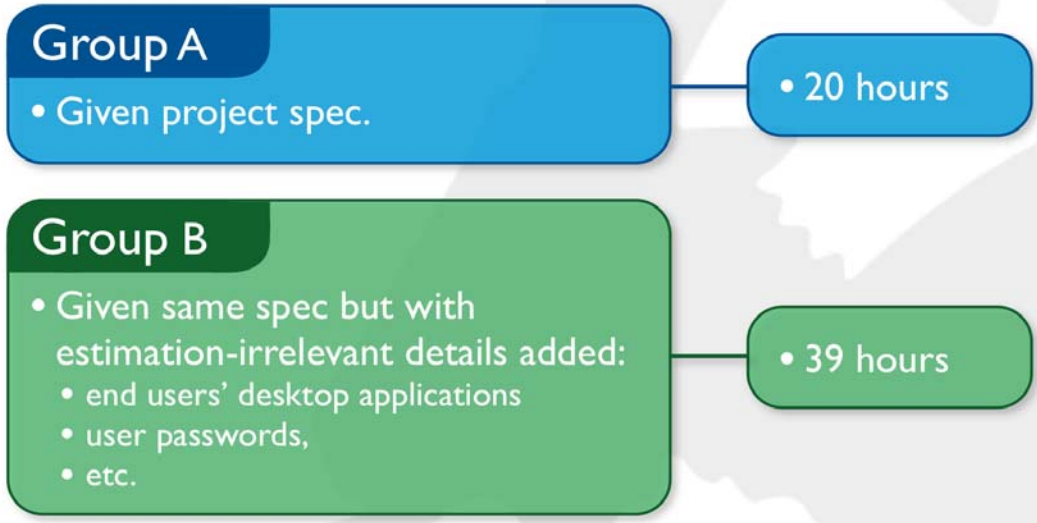
⁷Jørgensen, Magne, and Kjetil Moløkken. 2002. *Combination of Software Development Effort Prediction Intervals: Why, When and How?*



© Mountain Goat Software, LLC

20

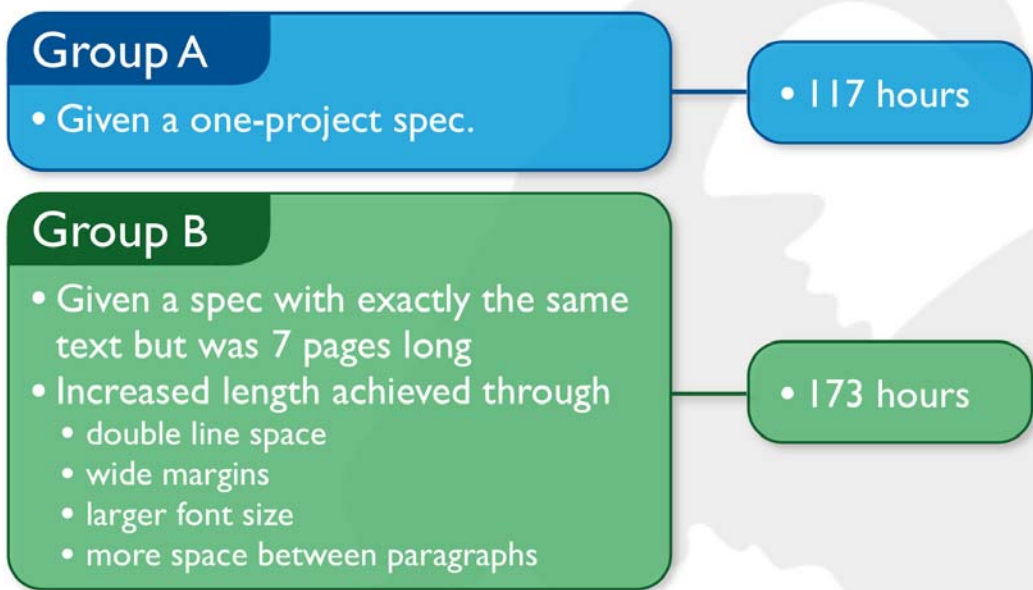
Reduces impact of irrelevant information



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

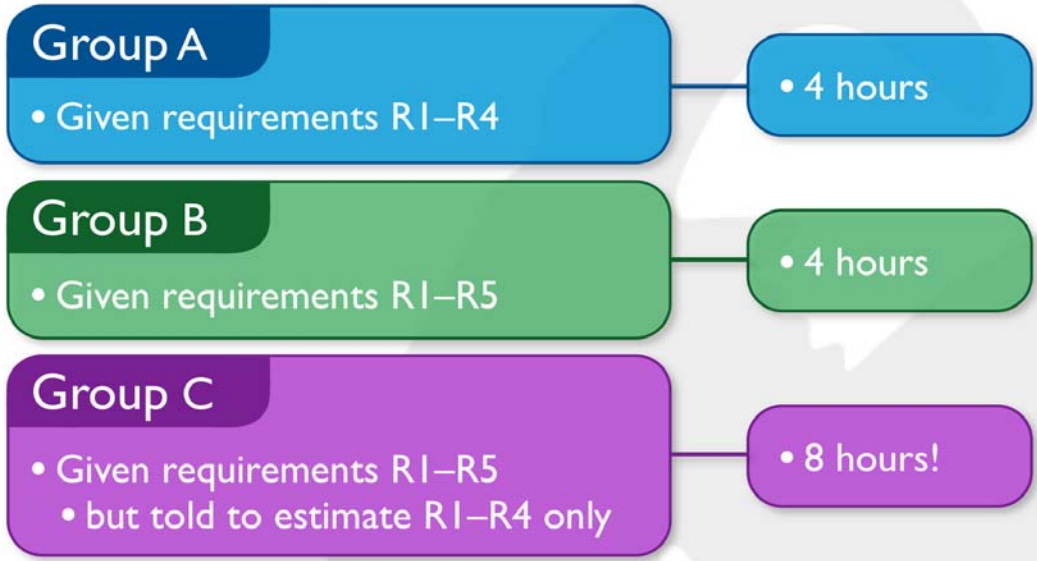
Specification length



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

Extra requirements



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

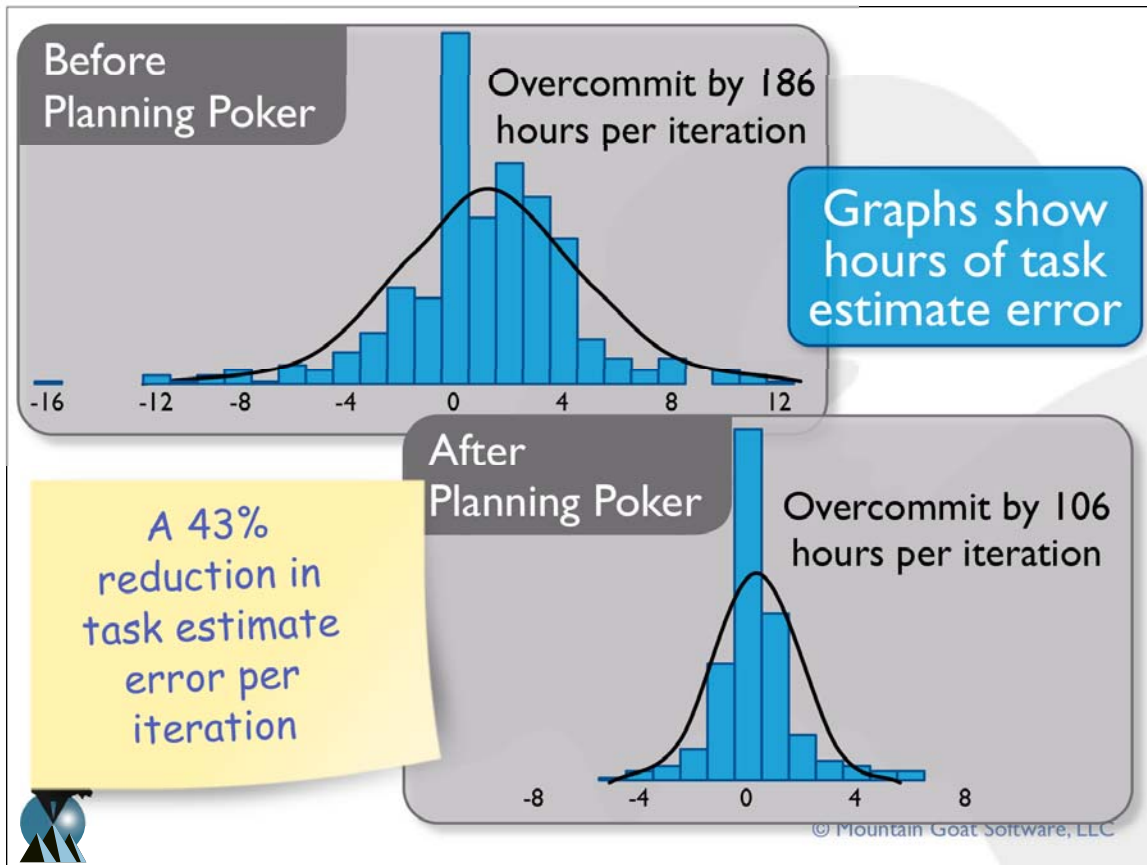
Reduces likelihood of anchoring



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.




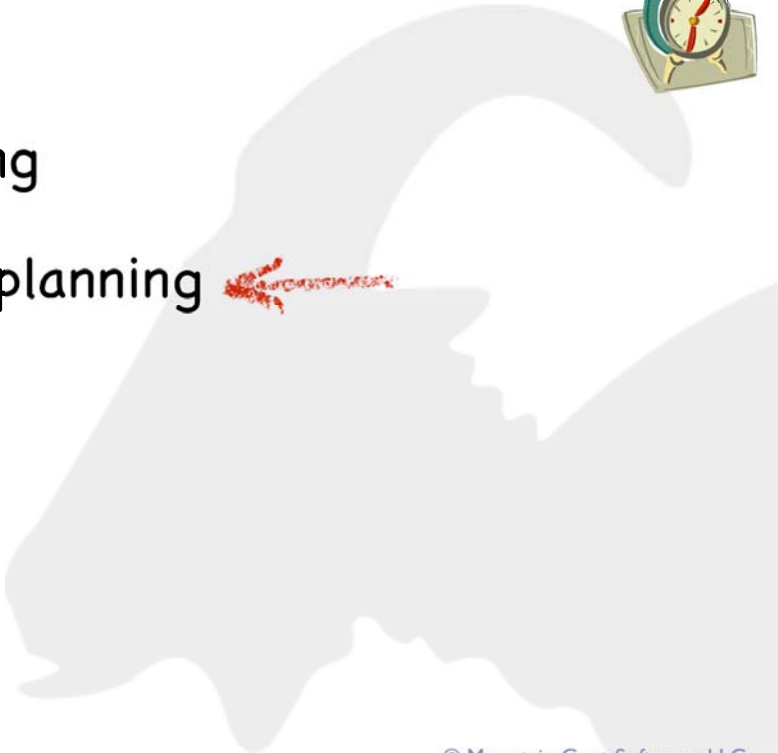
© Mountain Goat Software, LLC



25

Agenda

- Estimating
- Release planning

© Mountain Goat Software, LLC

26

Release planning

Purpose

To answer questions such as:

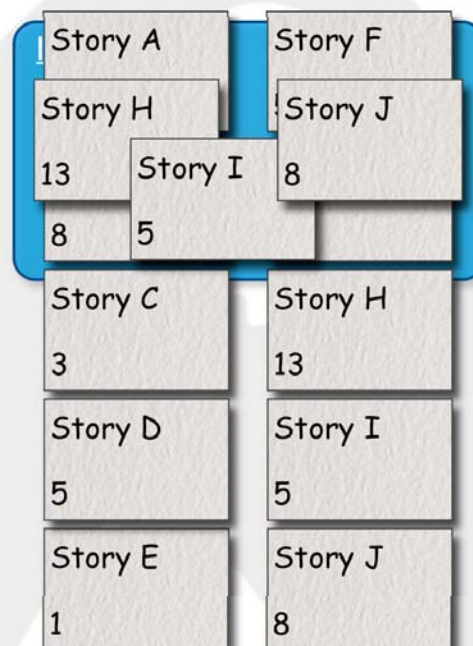
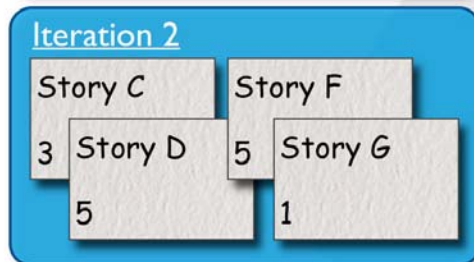
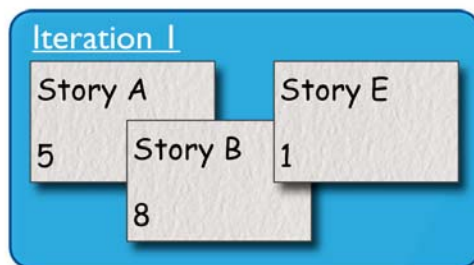
- How much will be done by 30 June?
- When can we ship with this set of features?
- How many people or teams should be on this project?

Inputs

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

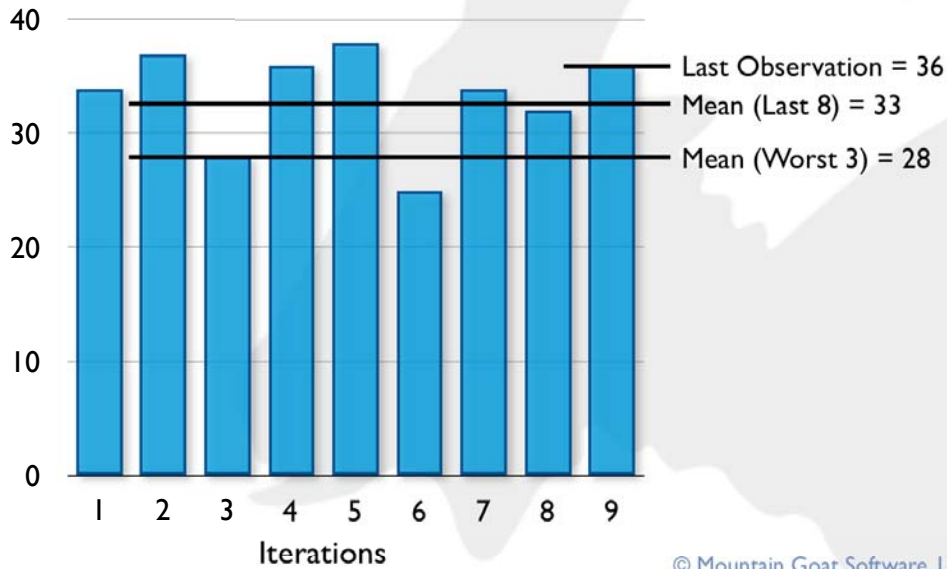


An example with velocity=14



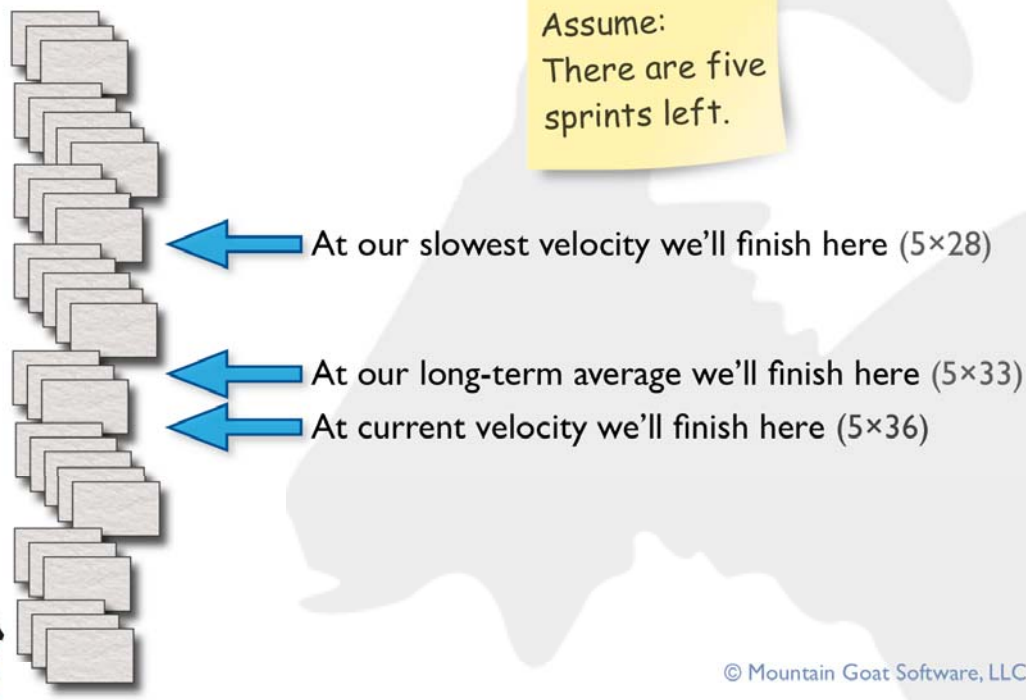
Updating the release plan

- Use multiple views of observed velocity



29

Extrapolate from velocity



30

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 \times number of sprints
 - Count off that many points
 - These are “Will Have” items
4. Multiply high velocity \times number of sprints
 - Count off that many more points
 - These are “Might Have items”

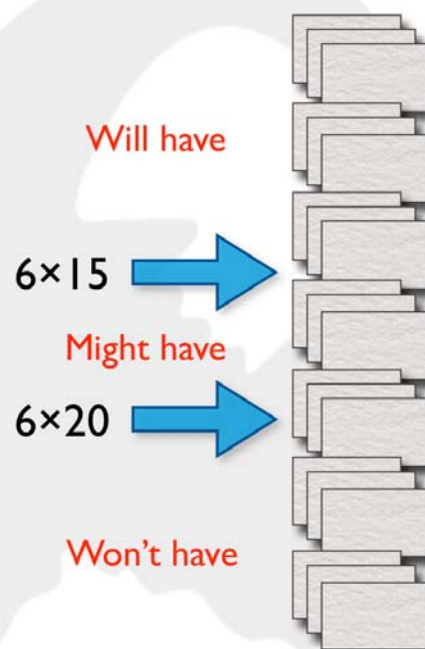


© Mountain Goat Software, LLC

31

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

32

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

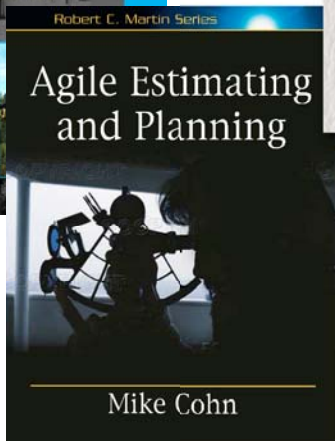
Where do you want to be?



© Mountain Goat Software, LLC

33

Mike Cohn contact info



mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)



© Mountain Goat Software, LLC



34