

# Agile Estimating & Planning

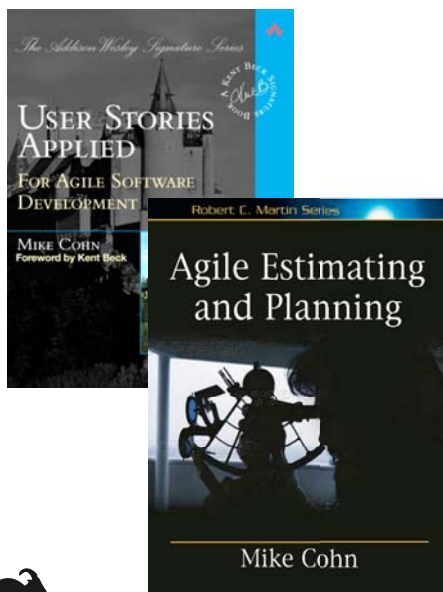
Mike Cohn  
President  
Mountain Goat Software  
Lafayette, Colorado  
mike@mouнтаingoatsoftware.com

**SD**  
WEST 2007

MARCH 19-23, 2007, SANTA CLARA, CA

1

## Mike Cohn - background



Consultant, author,  
and speaker

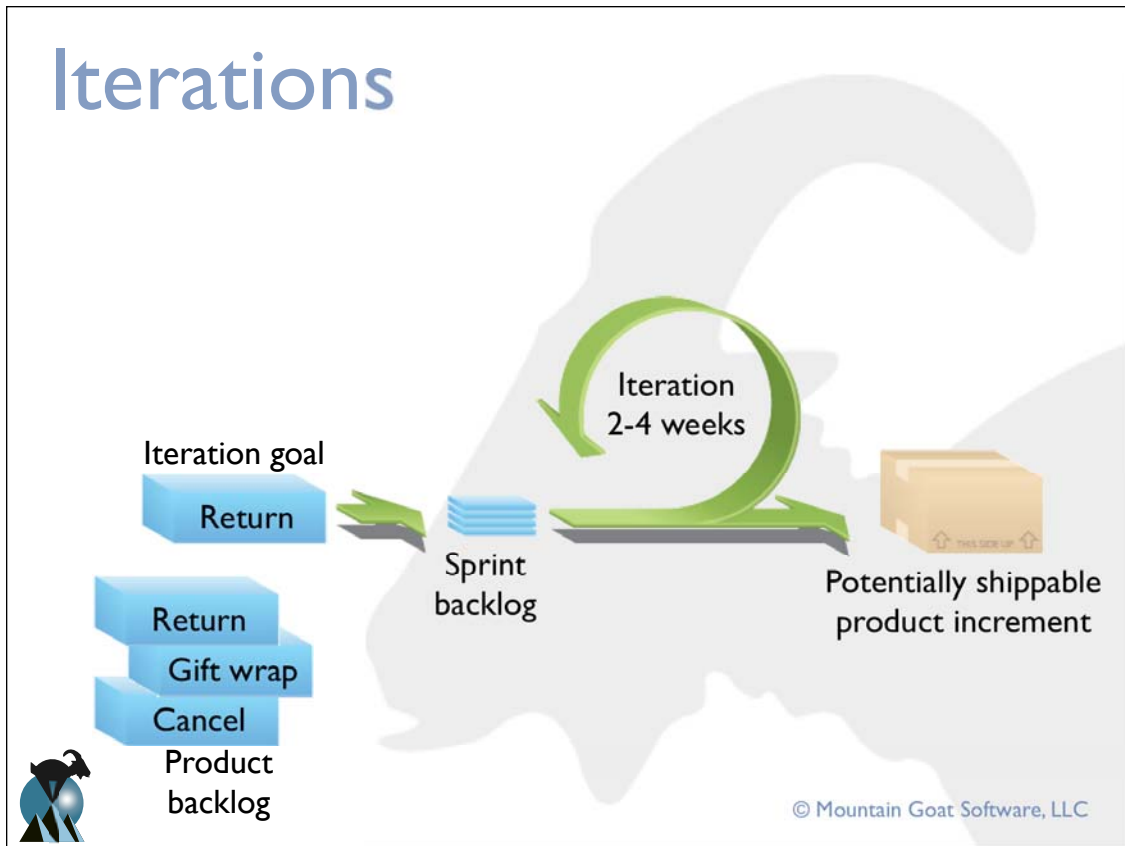
- Founding member and director of Agile Alliance, Scrum Alliance, and Agile Project Leadership Network
- Founder of Mountain Goat Software



© Mountain Goat Software, LLC

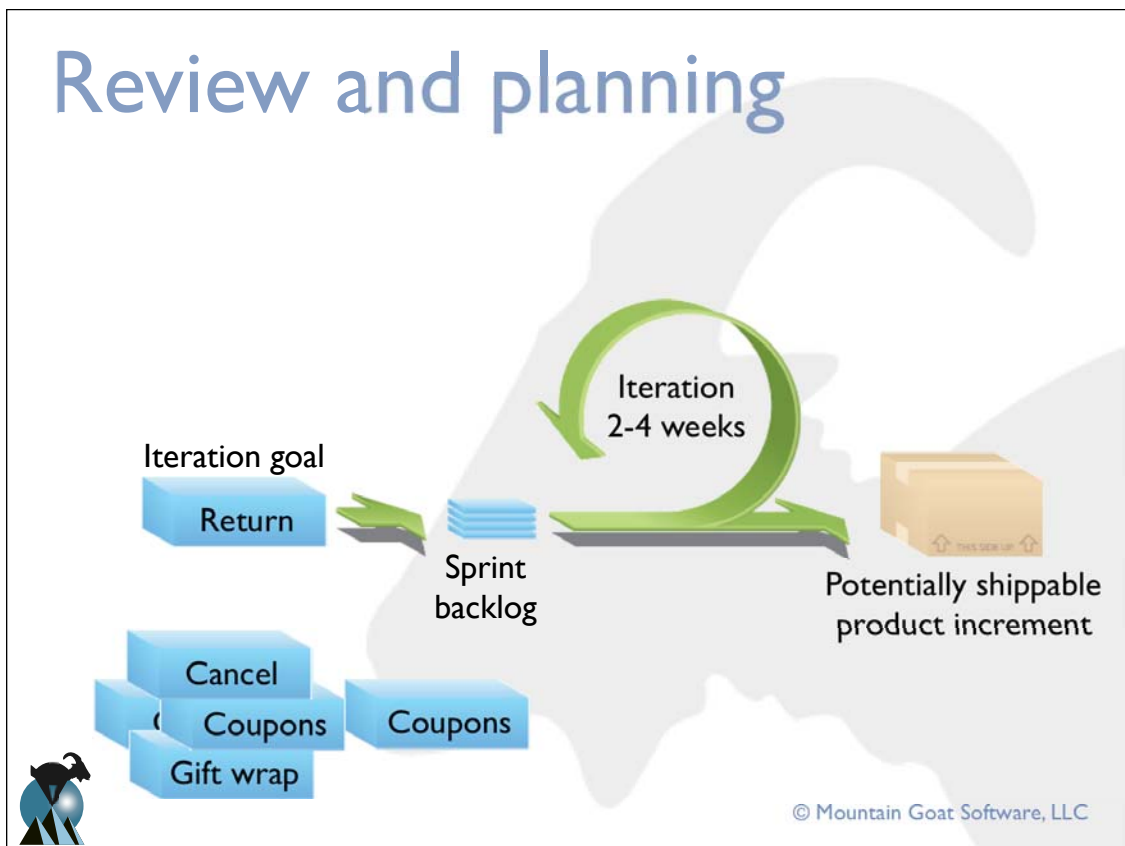
2

# Iterations



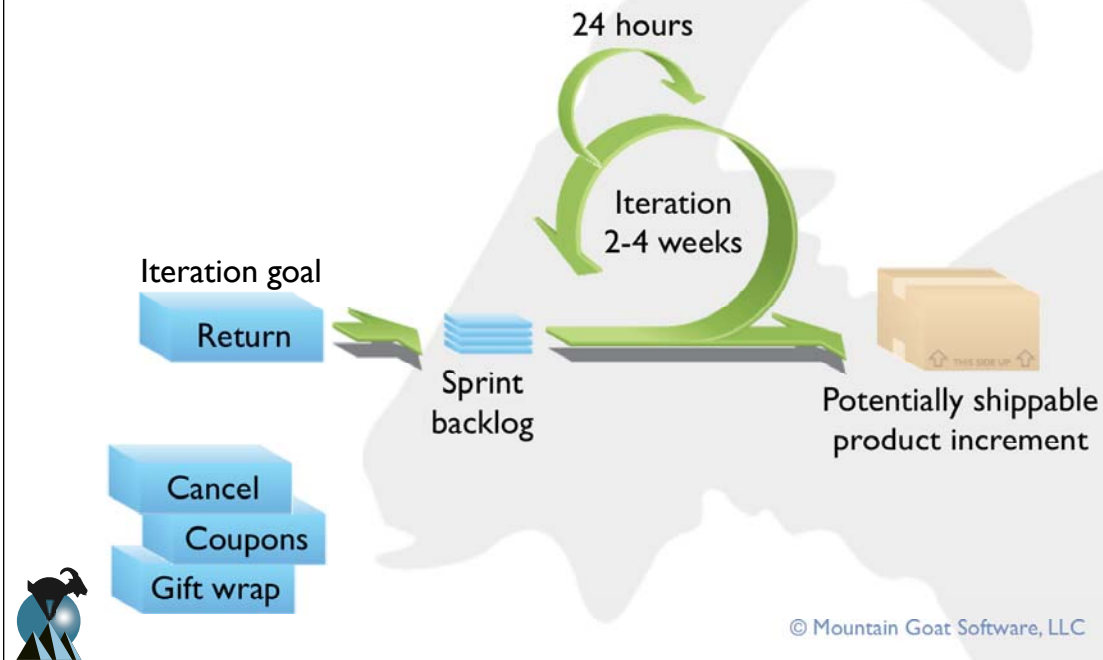
3

# Review and planning



4

# The daily standup



5

# What's a good plan?

- A good plan is one that supports reliable decision-making
- Will go from
  - We'll be done in the fourth quarter
  - We'll be done in November
  - We'll be done November 7th



© Mountain Goat Software, LLC

6

# What makes planning agile?

Is more focused on planning than the plan

Encourages change

Results in plans that are easily changed

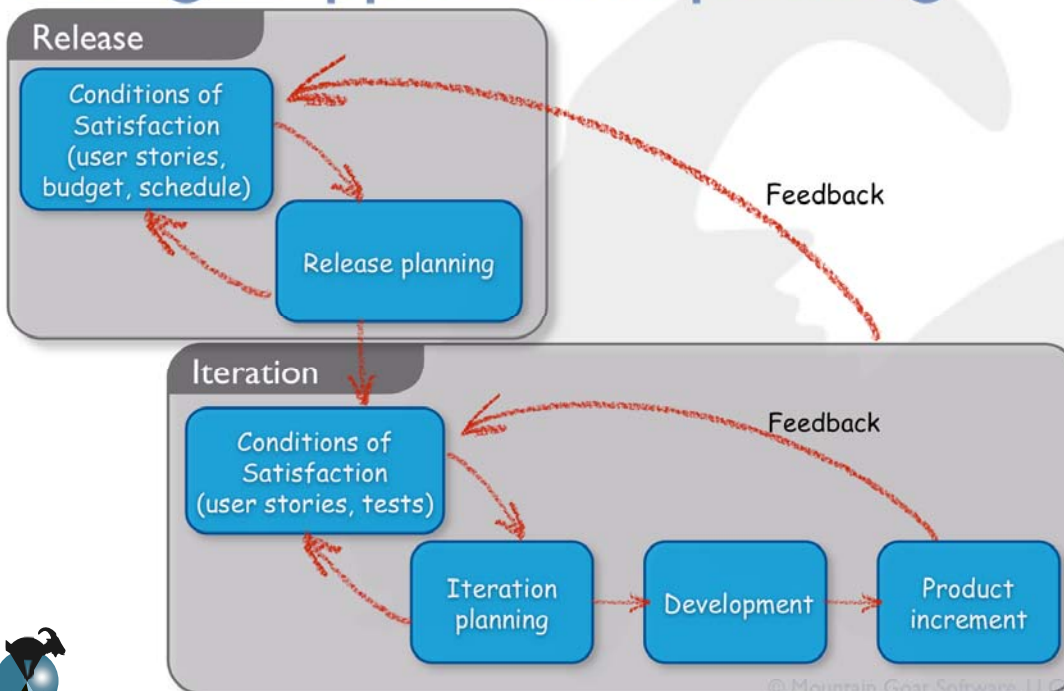
Is spread throughout the project



© Mountain Goat Software, LLC

7

# An agile approach to planning



8

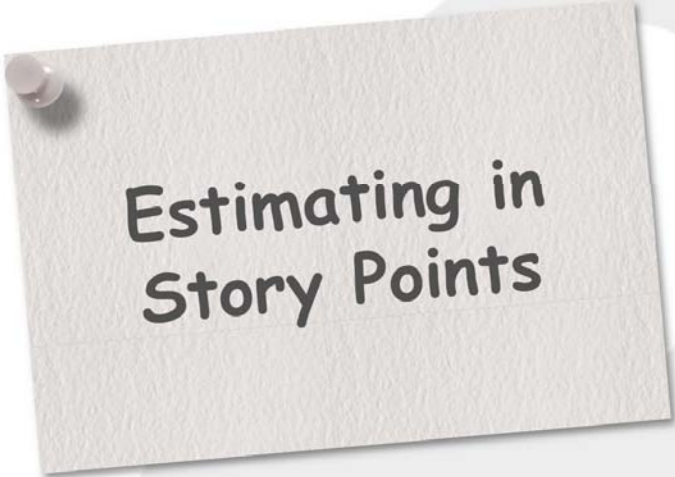
## Agenda

- Estimating in story points
- Estimating in ideal time
- Techniques for estimating
- Iteration planning
- Release planning
- Estimating velocity



© Mountain Goat Software, LLC

9



Estimating in  
Story Points

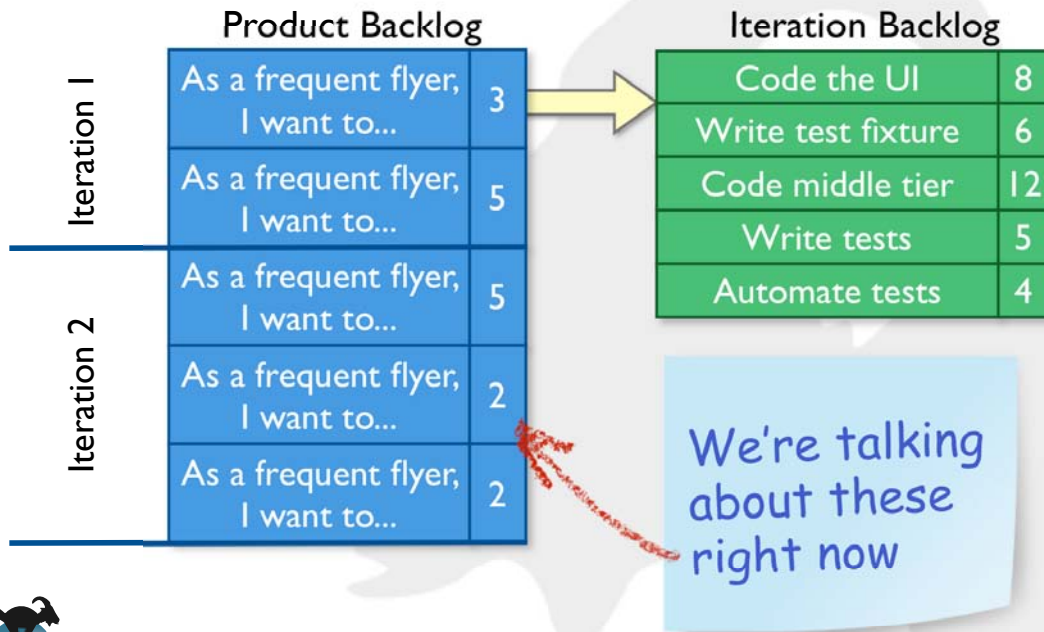


© Mountain Goat Software, LLC

10

Monday, March 19, 2007

# Which we're talking about



© Mountain Goat Software, LLC

11

# How long will it take...



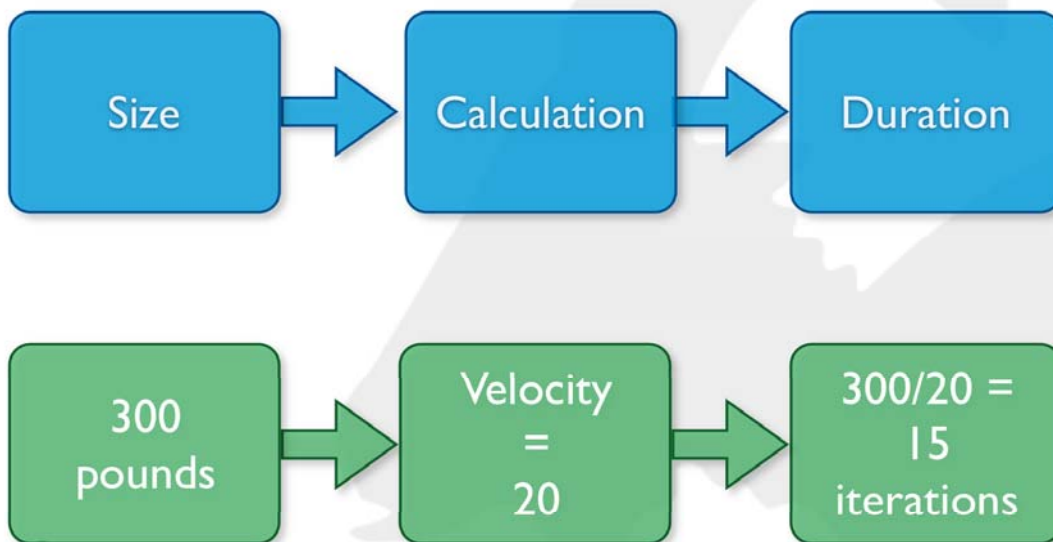
- ...to read the latest Harry Potter book?
- ...to drive to Seattle?



© Mountain Goat Software, LLC

12

## Estimate size; derive duration



© Mountain Goat Software, LLC

13

## Measures of size

- Traditional and agile measure size differently



© Mountain Goat Software, LLC

14

# 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

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

5



© Mountain Goat Software, LLC

15

# Zoo points

Assign “zoo points” to the following breeds

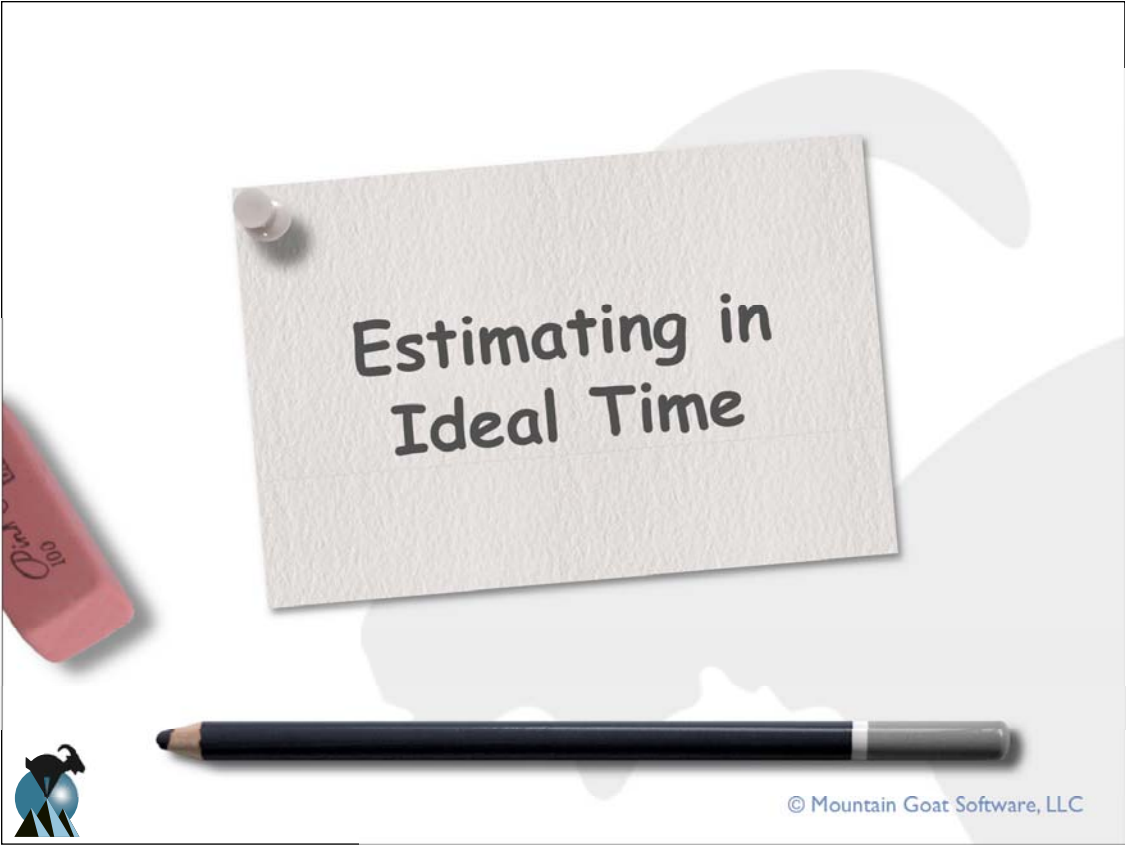
Lion  
Kangaroo  
Rhinocerus  
Bear  
Giraffe  
Gorilla  
Hippopotamus  
Tiger



© Mountain Goat Software, LLC

16





## Estimating in Ideal Time

© Mountain Goat Software, LLC

17

## 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?)

© Mountain Goat Software, LLC

18

# Elapsed time vs. ideal time

## Ideally

- Monday has 8 hours
- Each week has 40 hours

So, this developer will only make four hours of progress on Monday.

It will take two calendar days to complete one ideal day of work.

## But instead...

Monday has:

- 3 hours of meetings
- 1 hour of email
- 4 hours left for the project

"How long will this take?"



© Mountain Goat Software, LLC

19

# Ideal time vs. elapsed time

- It's easier to estimate in ideal time
- It's too hard to estimate directly in elapsed time
  - Need to consider all the factors that affect elapsed time at the same time you're estimating



© Mountain Goat Software, LLC

20

# 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



© Mountain Goat Software, LLC

21

# What I usually do

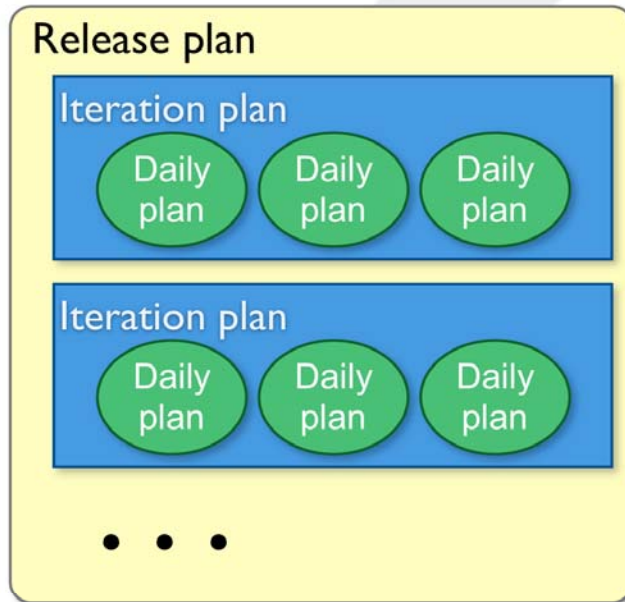
- I prefer story points
- ...but they make some teams uncomfortable, so I'll
  - Start with ideal time
    - Gives the team a nice foundation for the initial stories
    - Helps team get started
  - Define "1 story point = 1 ideal day"
  - Then
    - Gradually convert team to thinking in unit-less story points
    - "This story is like that story."
    - Stop talking about how long it will take



© Mountain Goat Software, LLC

22

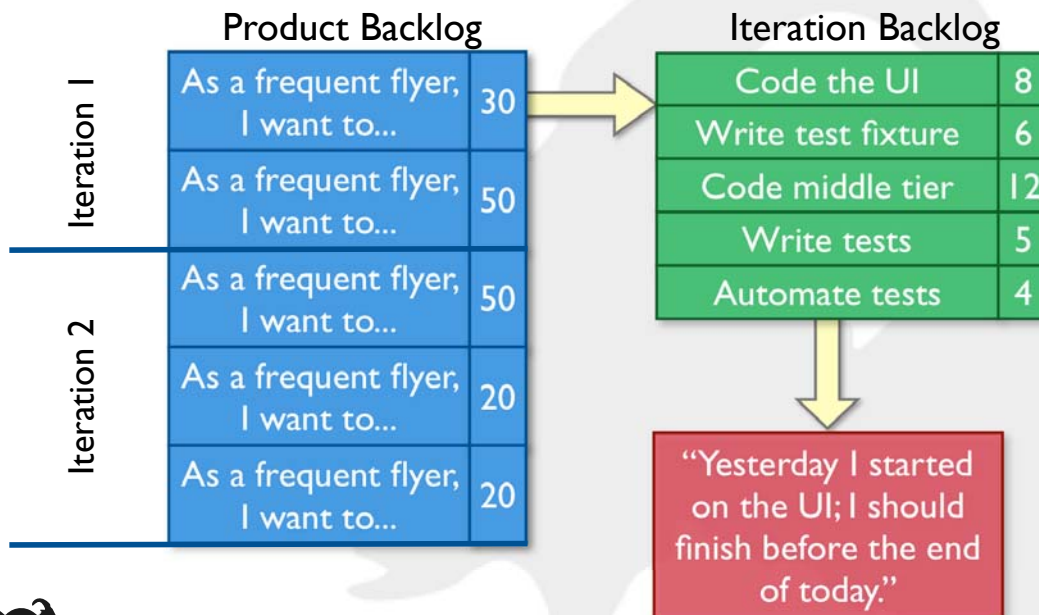
# Three levels of planning...



© Mountain Goat Software, LLC


23

# ...three levels of precision



© Mountain Goat Software, LLC

24



## Techniques for Estimating

© Mountain Goat Software, LLC

25

## Estimate by analogy

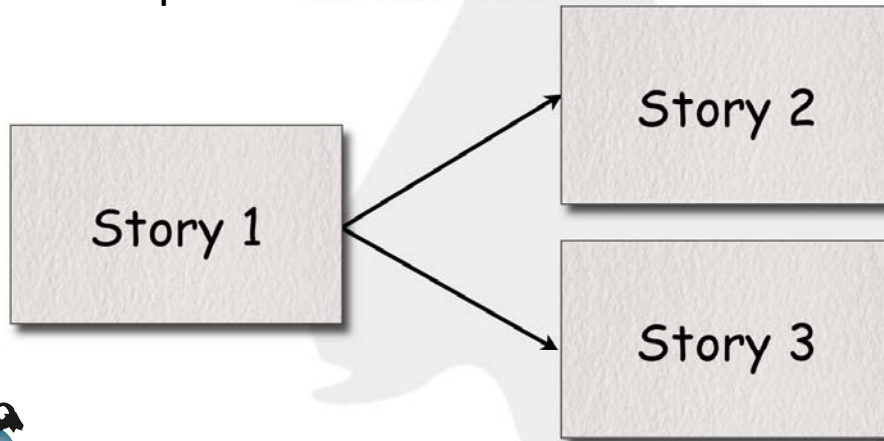
- Comparing a user story to others
  - “This story is like that story, so its estimate is what that story’s estimate was.”
- Don’t use a single gold standard
- Triangulate instead
  - Compare the story being estimated to multiple other stories

© Mountain Goat Software, LLC

26

# Triangulation

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



© Mountain Goat Software, LLC

27

# Disaggregation

- Breaking a big story into smaller stories or tasks
  - You know how long the smaller tasks take
  - So, disaggregating to something you know lets you estimate something bigger you don't know
- Sometimes very useful
- But disaggregating too far causes problems
  - Forgotten tasks
  - Summing lots of small errors can be big number

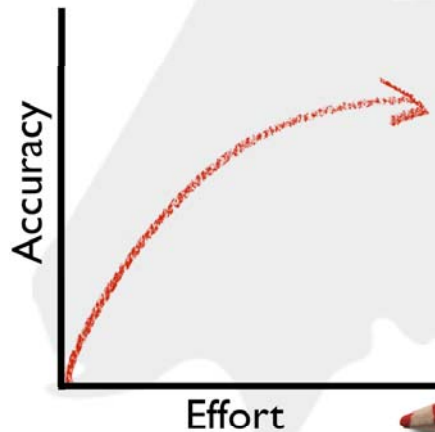


© Mountain Goat Software, LLC

28

# How much effort?

- A little effort helps a lot
- A lot of effort only helps a little more



© Mountain Goat Software, LLC

29

# Use the right units

- Can you distinguish a 1-point story from a 2?
- Can you distinguish a 17 from an 18?
- Use units that make sense, such as
  - 1, 2, 3, 5, 8, 13
  - 1, 2, 4, 8
- Stay mostly in a 1-10 range

Include 0 and  
 $\frac{1}{2}$  if you  
want

© Mountain Goat Software, LLC

30

# 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



© Mountain Goat Software, LLC

31

## 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

32



# 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 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 summary of this session for your boss.	



© Mountain Goat Software, LLC

33

# Why planning poker works

- Emphasizes relative estimating
- Focuses most estimates within an approximate one order of magnitude
- Everyone's opinion is heard
- Estimators are required to justify estimates
- It's quick
- It's fun

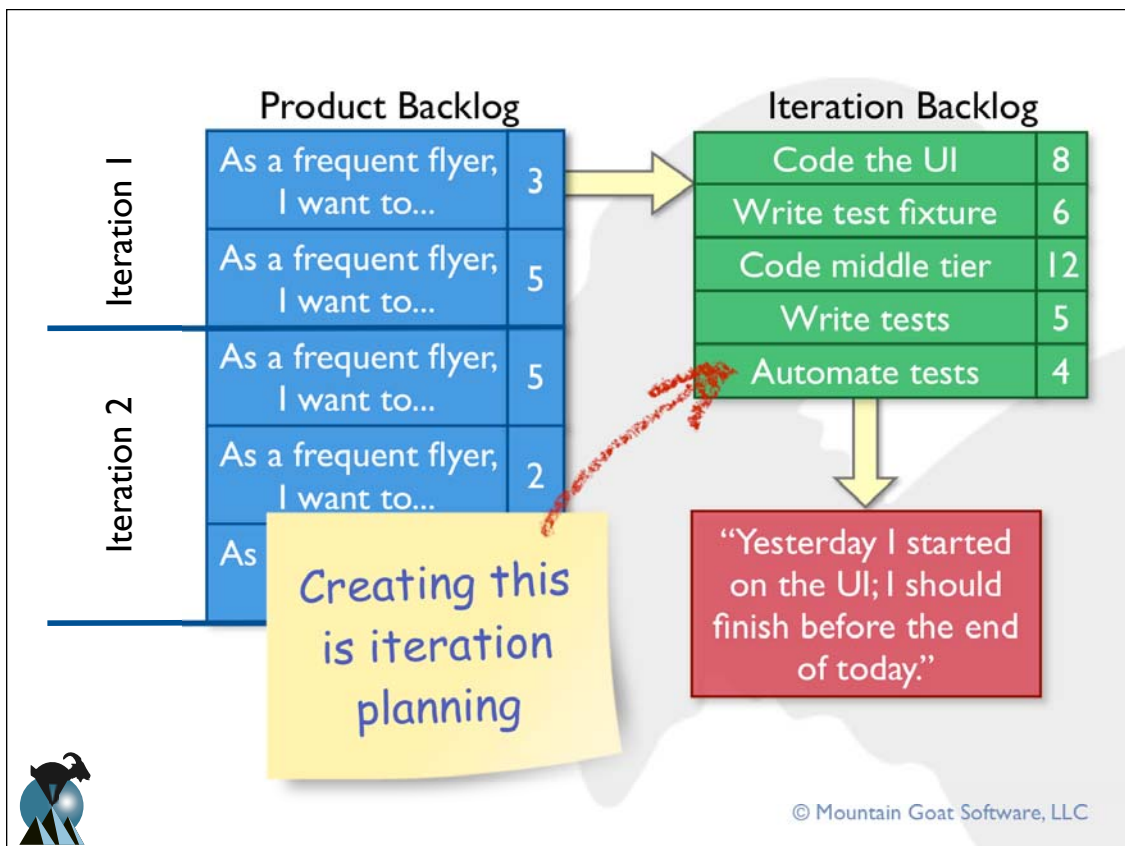


© Mountain Goat Software, LLC

34



35



36

# Two approaches

- Velocity-driven iteration planning
  - “We finished 15 story points last time, let’s plan on 15 story points this time.”
  - Very unreliable in what will be accomplished during an iteration
    - Velocity is mostly useful over the long term
- Commitment-driven iteration planning
  - More likely to lead to realistic iteration commitments



© Mountain Goat Software, LLC

37

## Commitment-driven iteration planning

- Discuss the highest priority item on the product backlog
- Decompose it into tasks
- Estimate each task
  - Whole team estimates each task
- Ask ourselves, “Can we commit to this?”
  - If yes, see if we can add another backlog item
  - If not, remove this item but see if we can add another smaller one



© Mountain Goat Software, LLC

38

# Estimate availability

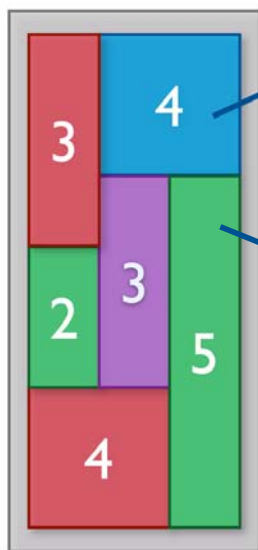
Person	Hours per Day	Hours per Iteration
Sergey	4-6	40-60
Yuri	5-7	50-70
Carina	2-3	20-30
Total		110-160



© Mountain Goat Software, LLC

39

# It looks something like this



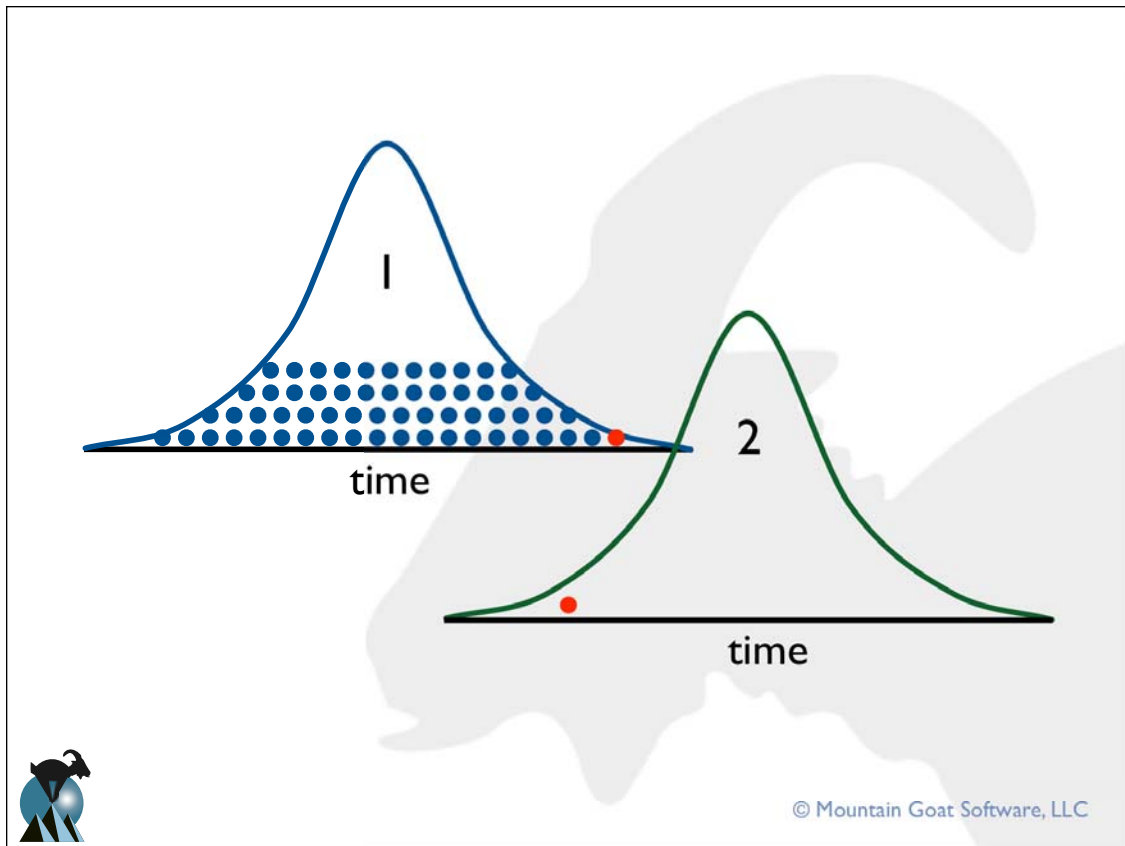
- Code the abc class (8 hours)
- Code the user interface (4)
- Write test fixtures (4)
- Code the xyz class (6)
- Update performance tests (4)

- Prototype the UI (8 hours)
- Demo UI to 3 outside users (3)
- Code new UI (12)
- Update documentation (3)



© Mountain Goat Software, LLC

40



© Mountain Goat Software, LLC

41

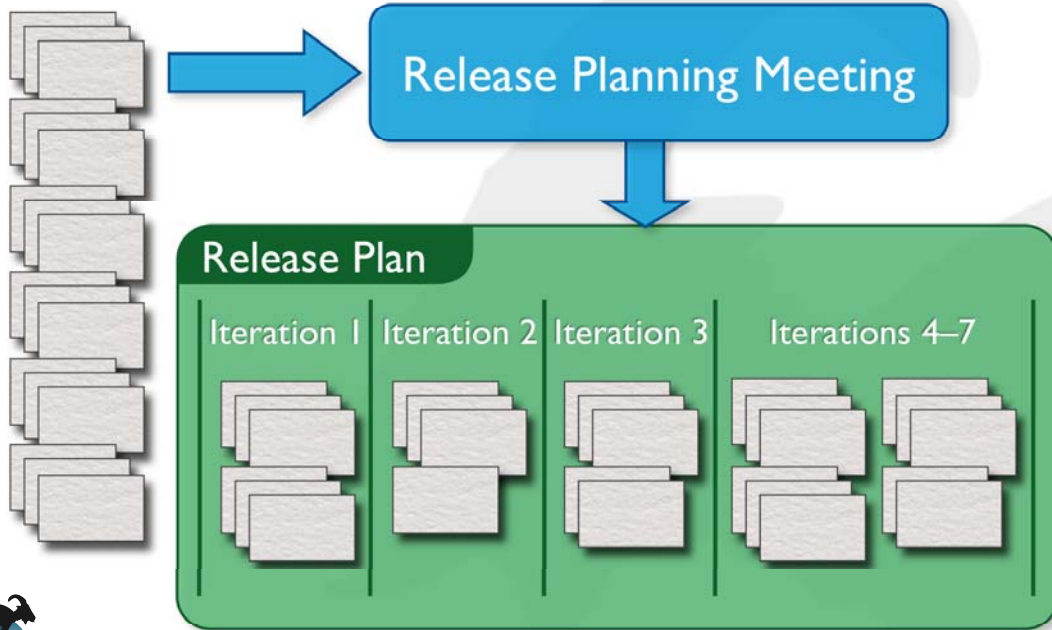


© Mountain Goat Software, LLC

42

Monday, March 19, 2007

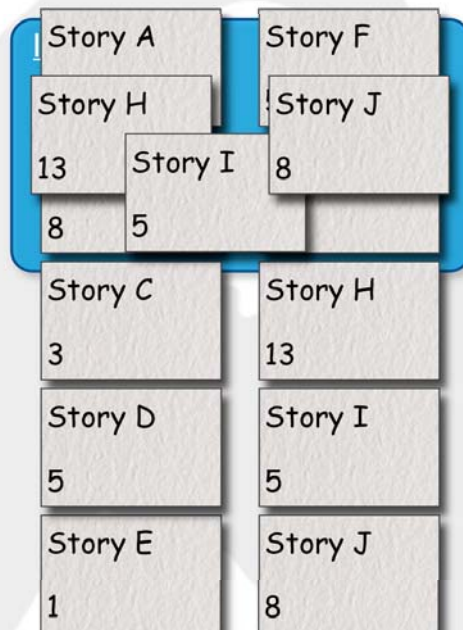
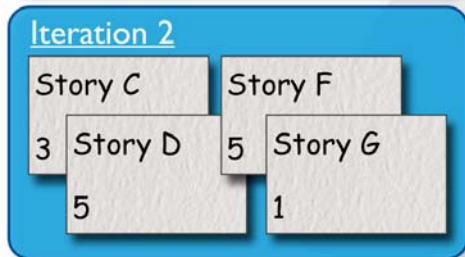
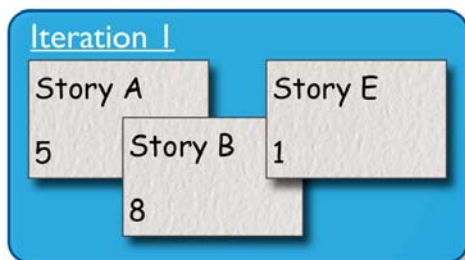
# Release planning



© Mountain Goat Software, LLC

43

# An example with velocity=14



© Mountain Goat Software, LLC

44

# Updating the release plan

- Revisit the release plan at the end of every iteration
- Update it based on:
  - Current understanding of velocity
  - Current prioritization of the product backlog
- This should be a very short and sweet process



© Mountain Goat Software, LLC

45

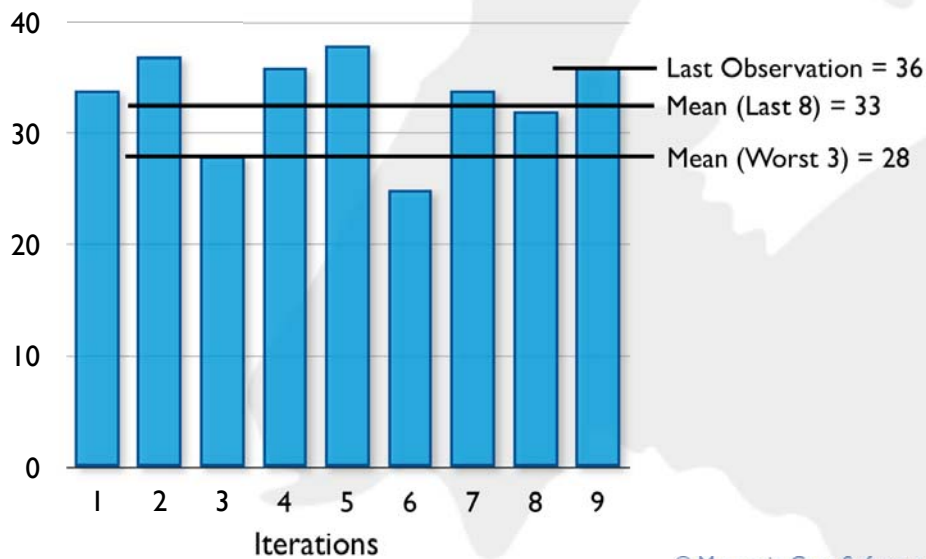
# Changing the release plan



© Mountain Goat Software, LLC

46

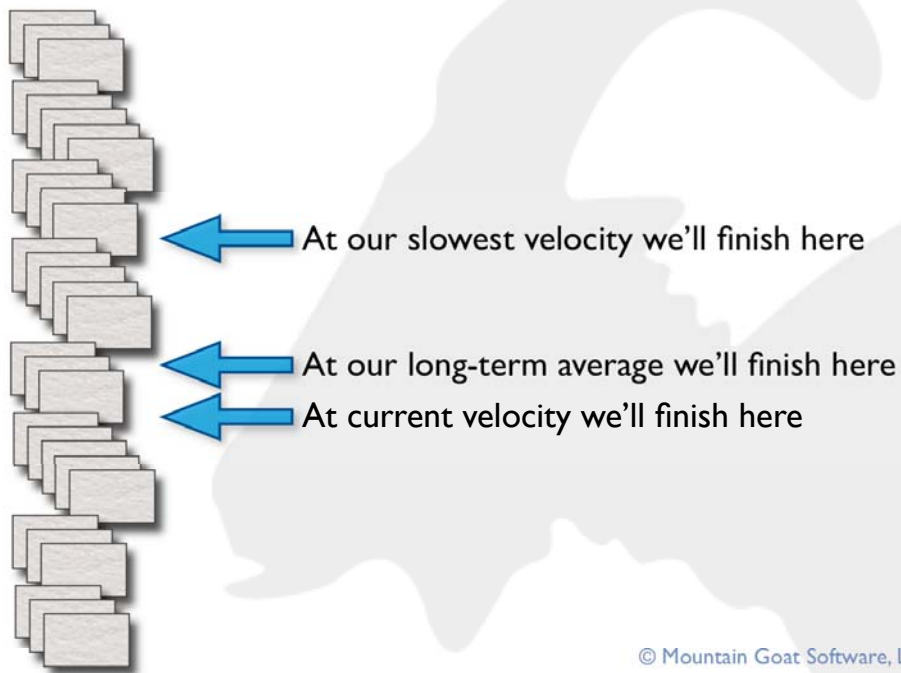
# Track velocity multiple ways



© Mountain Goat Software, LLC

47

# Extrapolate from velocity



© Mountain Goat Software, LLC

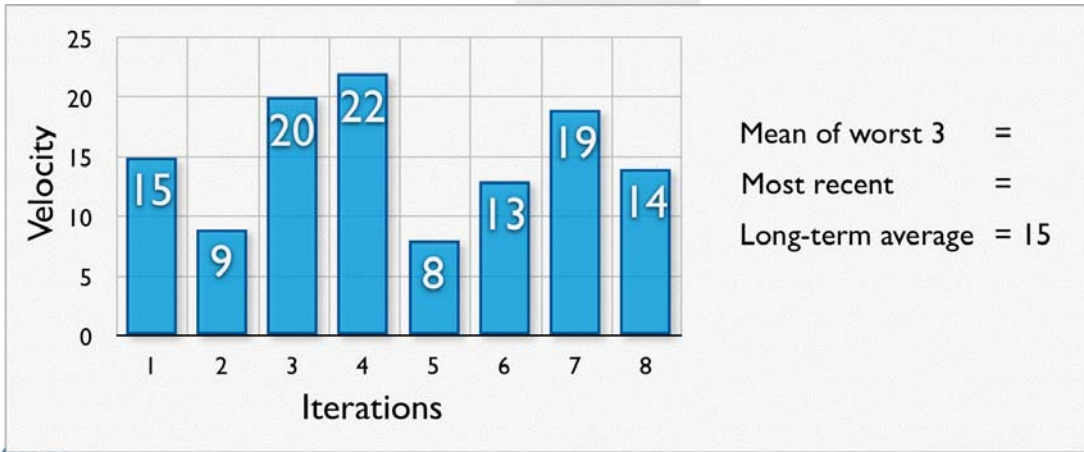
48



# Updating the release plan



Here are the results of the last 8 iterations. There are 6 iterations left. Using this data, update the release plan on the following slide by drawing three arrows into it.



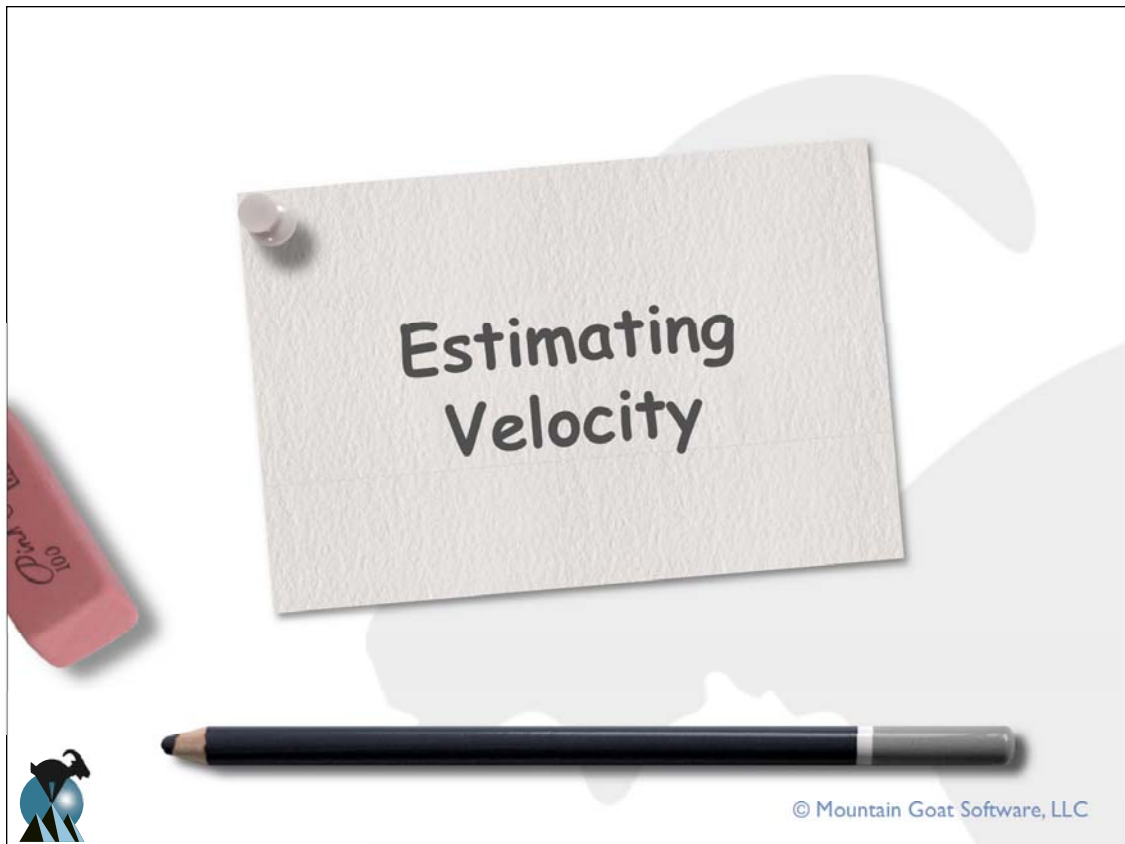
© Mountain Goat Software, LLC

# Update this release plan

6 × worst 3 = \_\_\_\_\_    6 × average of last 8 = \_\_\_\_\_    6 × most recent = \_\_\_\_\_

Running Total	Estimate	Story
5	5	As a user, I can...
10	5	As a user, I can...
23	13	As a user, I can...
31	8	As a user, I can...
51	20	As a user, I can...
59	8	As a user, I can...
64	5	As a user, I can...
72	8	As a user, I can...
77	5	As a user, I can...
85	8	As a user, I can...
90	5	As a user, I can...
93	3	As a user, I can...

C



51

## Initial velocity

- Three ways to come up initial velocity

- 1 Use historical averages
- 2 Wait until you run at least one iteration
- 3 Forecast it

- Express velocity as a range that matches your uncertainty in it



© Mountain Goat Software, LLC

52

# Forecasting velocity

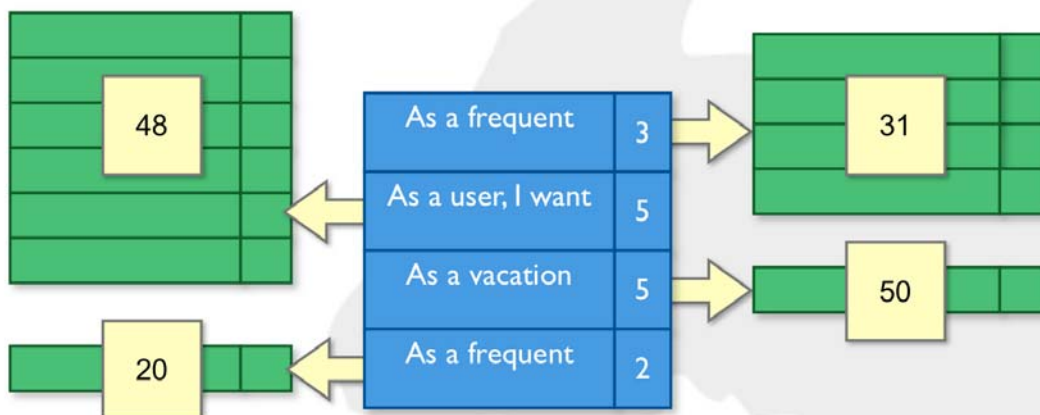
- Just like commitment-driven iteration planning
- Estimate available hours for the iteration
- Pick a story, break into tasks, estimate each task
  - Repeat until full
- Ideally, plan more than one iteration



© Mountain Goat Software, LLC

53

Sergey, Yuri, and Carina have 110-160 available hours. What is their likely velocity?



© Mountain Goat Software, LLC

54

# Upcoming public classes

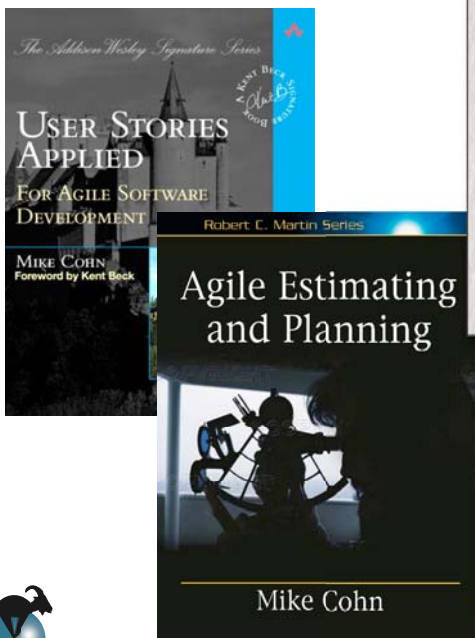
Date	What	Where
April 10-11 April 12	Certified ScrumMaster Agile Estimating and Planning	Santa Clara, CA
May 30-31	Certified Scrum Product Owner (with Ken Schwaber)	Boston, MA
June 11-12 June 13	Certified ScrumMaster Agile Estimating and Planning	Dallas
Jul 31-Aug 1 August 2	Certified ScrumMaster Agile Estimating and Planning	Denver, CO
Sept 11-12 Sept 13	Certified ScrumMaster Agile Estimating and Planning	Orlando, FL
Other classes in London, Stockholm, and Oslo if you're up for a longer trip.		

Register at  
[www.mountaingoatsoftware.com](http://www.mountaingoatsoftware.com)



© Mountain Goat Software, LLC

# Mike Cohn contact info



mike@mountaingoatsoftware.com  
 www.mountaingoatsoftware.com  
 (720) 890-6110 (office)  
 (303) 810-2190 (mobile)



MOUNTAIN GOAT  
SOFTWARE

© Mountain Goat Software, LLC

