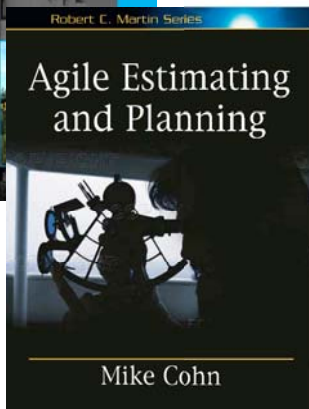


Agile Estimating and Planning



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



2

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



What makes planning agile?

Is more focused on
planning than the plan

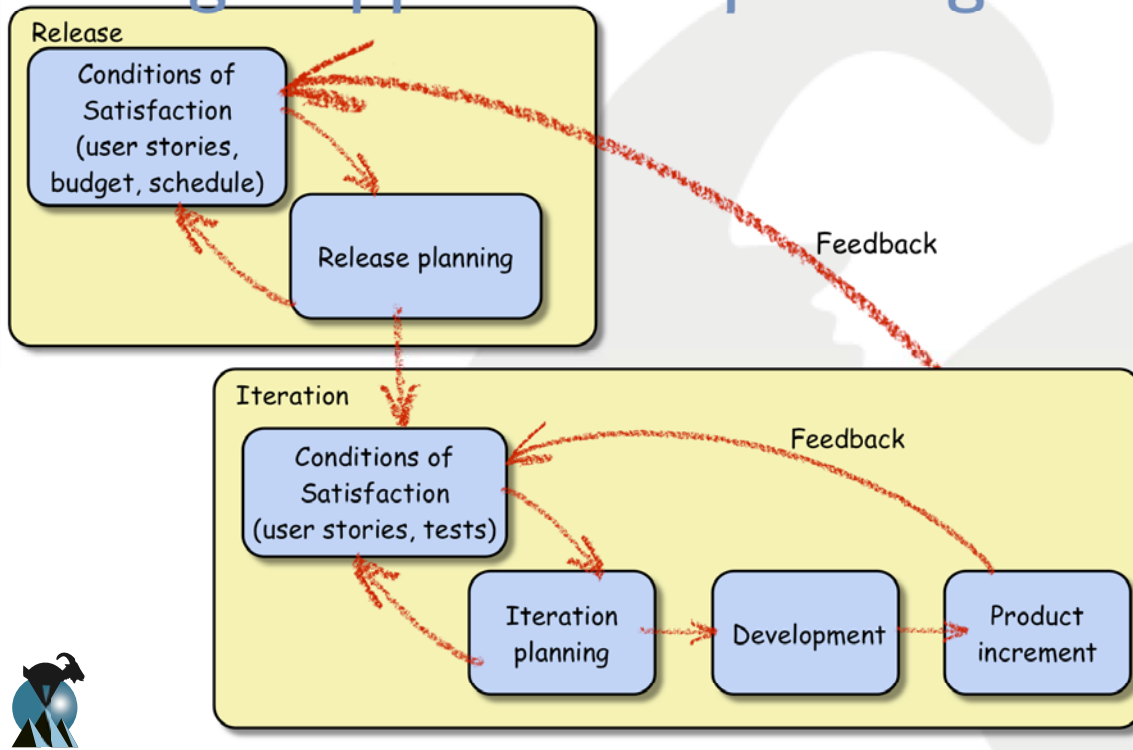
Encourages change

Results in plans that are
easily changed

Is spread throughout
the project



An agile approach to planning



5

Agenda

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

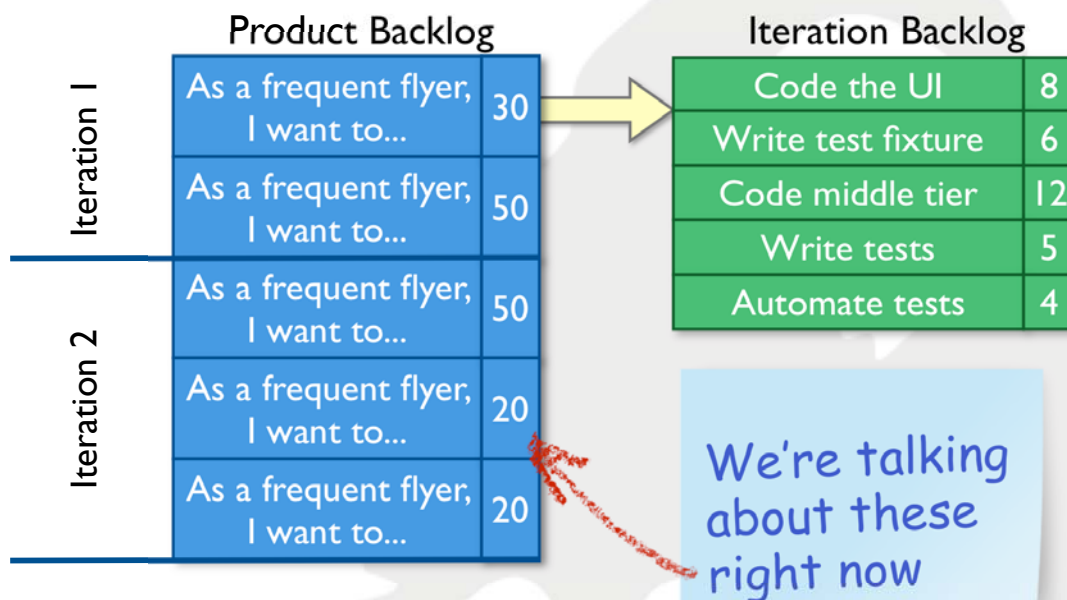
6

Estimating in Story Points

© Mountain Goat Software, LLC

7

Which we're talking about



© Mountain Goat Software, LLC

8

How long will it take...



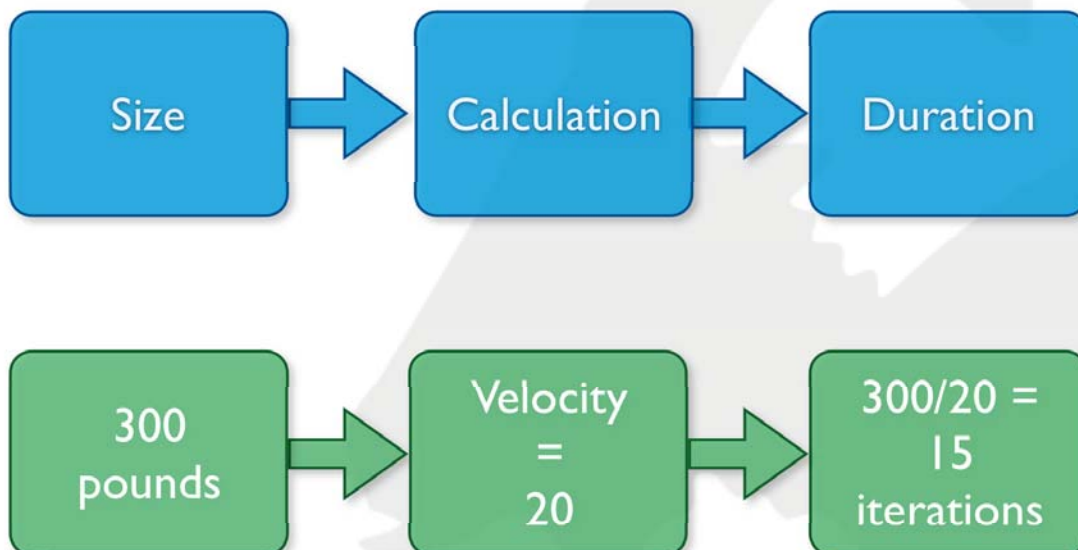
- ...to read the latest Harry Potter book?
- ...to drive to Seattle?



© Mountain Goat Software, LLC

9

Estimate size; derive duration



© Mountain Goat Software, LLC

10

Measures of size

- Traditional and agile measure size differently



© Mountain Goat Software, LLC

11

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

12

Dog points



Assign "dog points" to the following breeds

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



Estimating in
Ideal Time



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?)



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?"



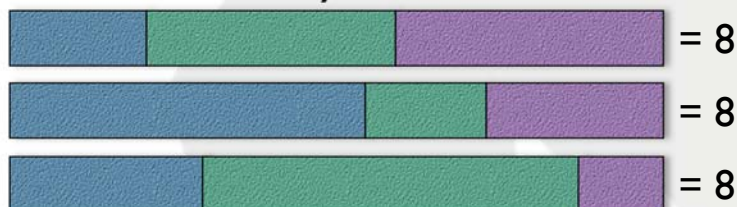
Ideal time vs. elapsed time

- 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



Specialization

- First, don't worry about it too much
 - We're usually better off with fairly rapid, imprecise estimates than spending more time
- Second
 - Just add up the components and report one total estimate of ideal days



The great debate



Which do you prefer:

Story points or ideal days?



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



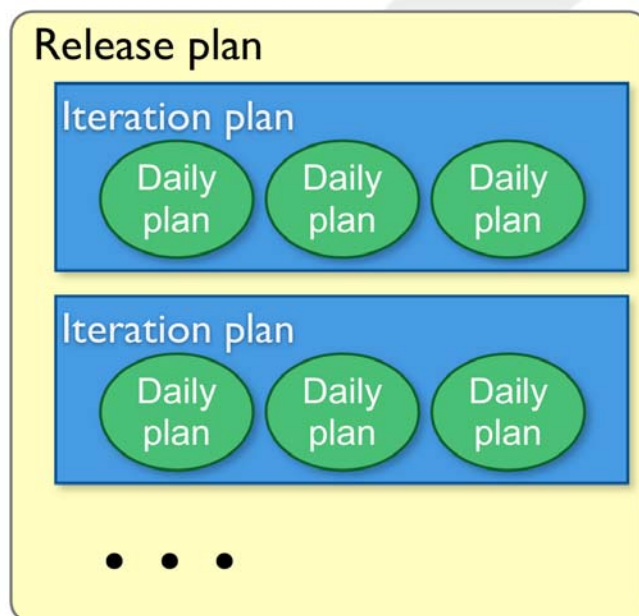
What I usually do



© Mountain Goat Software, LLC

21

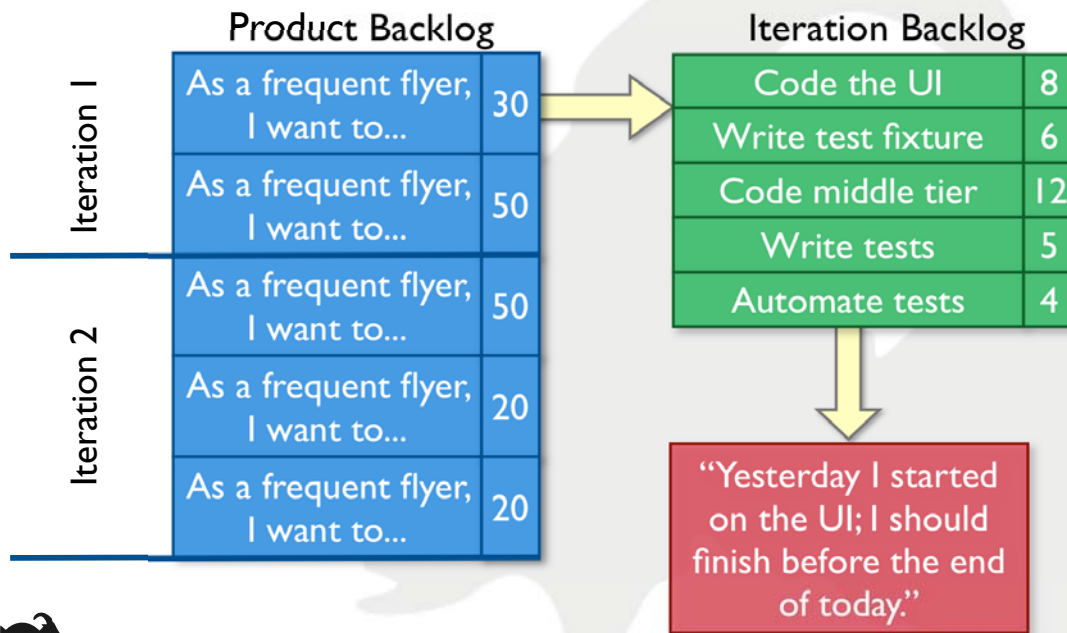
Three levels of planning...



© Mountain Goat Software, LLC

22

...three levels of precision



Techniques for Estimating

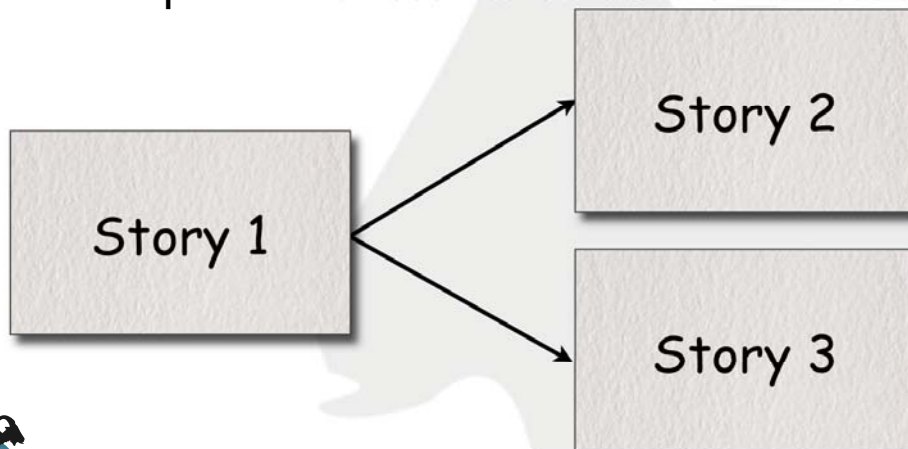
Estimate by analogy

- 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



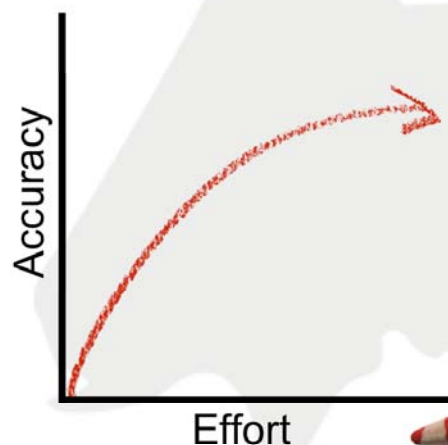
Disaggregation

- 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



How much effort?

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



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
 - 1, 2, 4, 8
- Stay mostly in a 1-10 range

Include 0 and
 $\frac{1}{2}$ 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



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.	
Read a 150-page book on agile software development.	
Write an 8-page summary of this session for your boss.	
Create a 60-minute presentation about agile estimating and planning for your coworkers.	
Wash and wax your boss' Porsche.	
Write the product backlog for a simple eCommerce site that sells only clocks.	
Recruit, interview, and hire a new member for your team.	



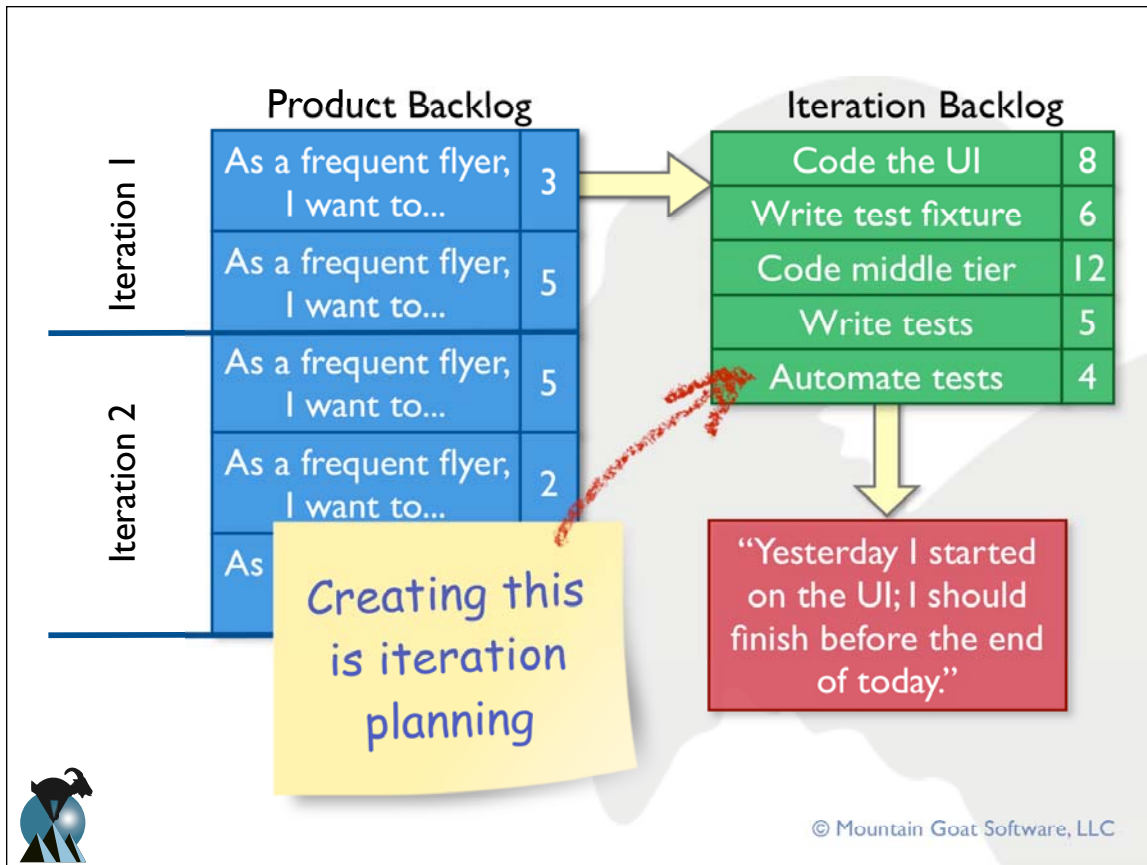
Why planning poker works

- 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



Iteration
Planning

A white sticky note with a white pushpin in the top-left corner, pinned to a light gray background. The text "Iteration Planning" is written in a bold, black, sans-serif font. To the left of the note is a red eraser, and below it is a black pencil.



35

Two approaches

- Velocity-driven iteration planning
 - “We finished 15 story points last time, let’s plan on 15 story points this time.”
- Commitment-driven iteration planning



36

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

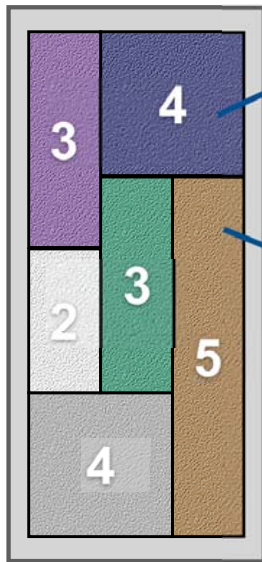


Estimate availability

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

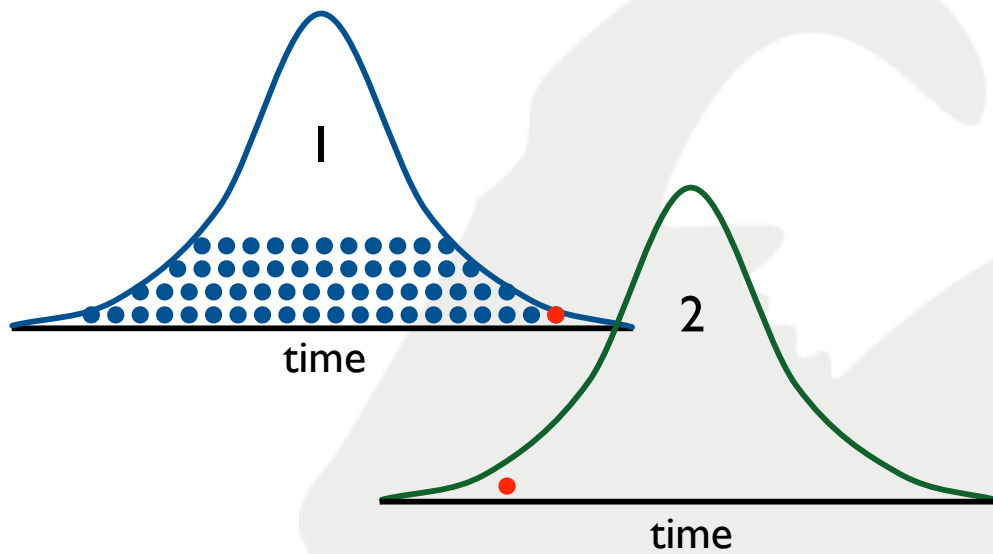


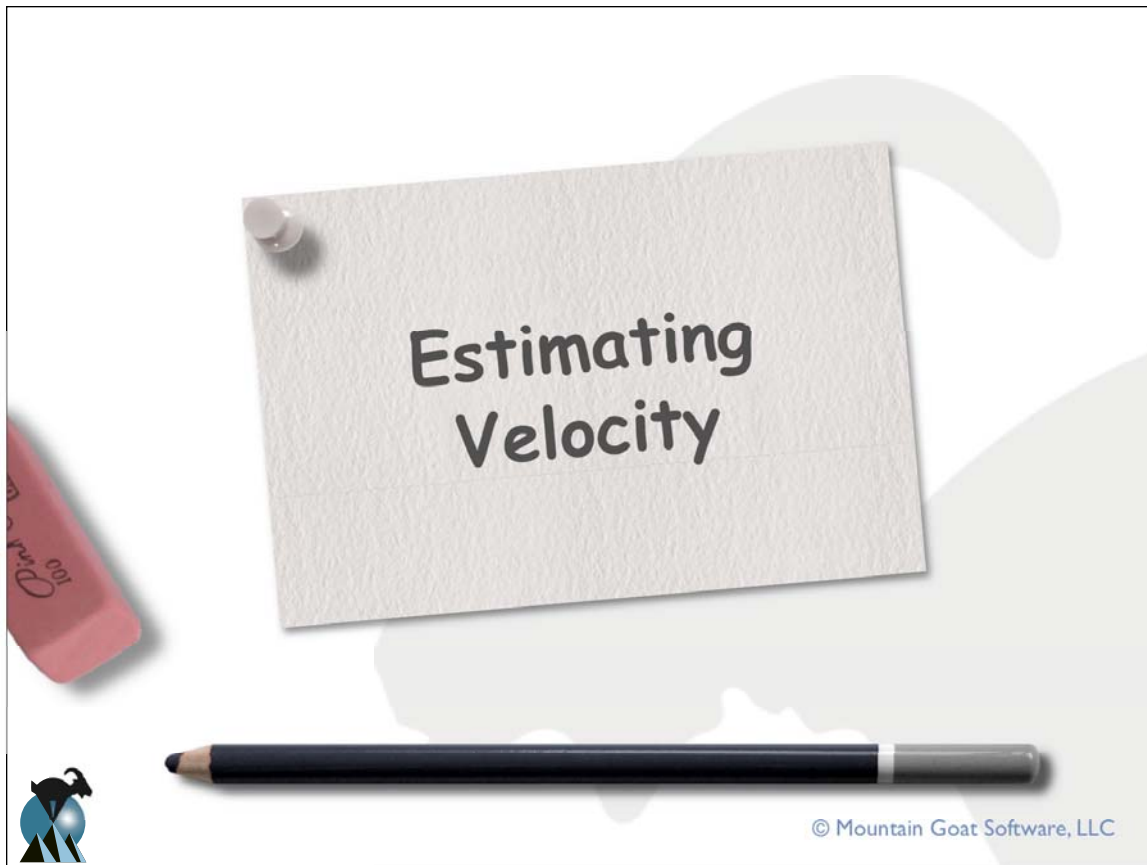
It looks something like this



- 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)





41

Initial velocity

- Three ways to come up initial velocity

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

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



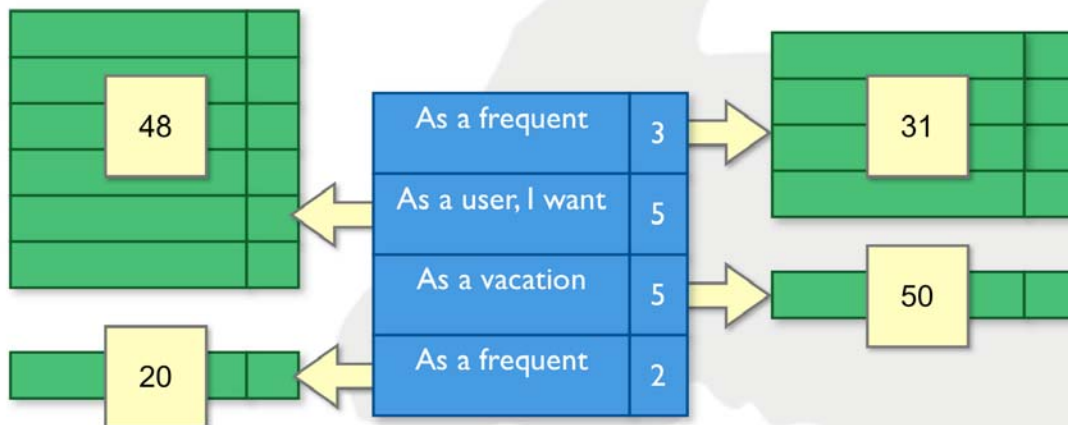
42

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



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

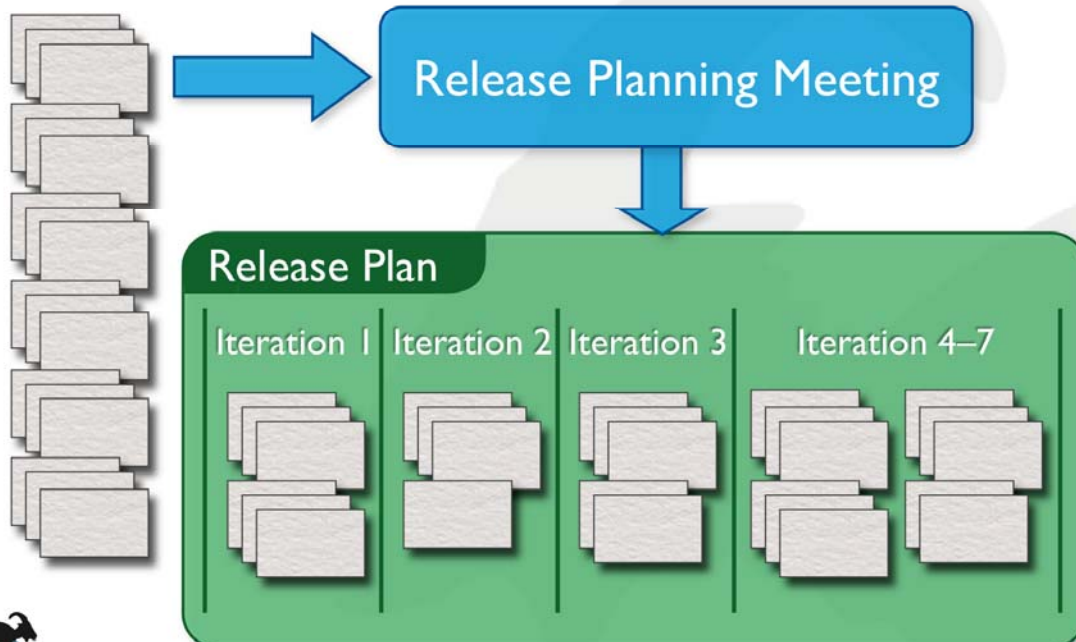


Release Planning

© Mountain Goat Software, LLC

45

Release planning

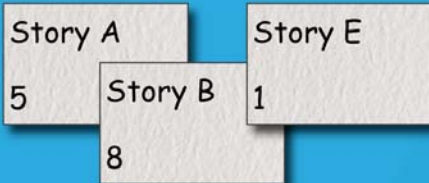


© Mountain Goat Software, LLC

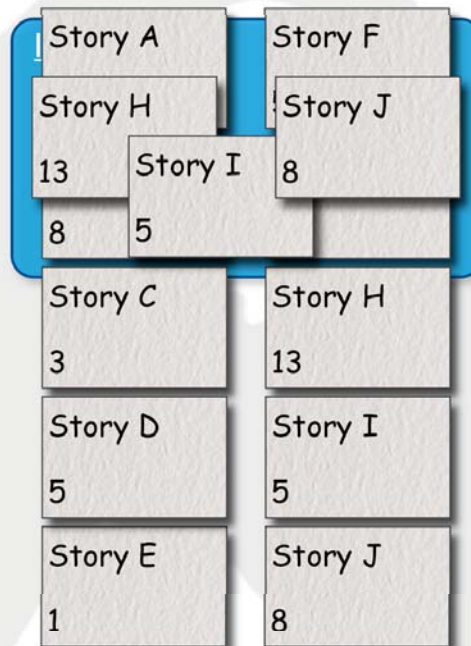
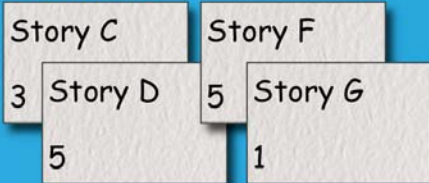
46

An example with velocity=14

Iteration 1



Iteration 2



© Mountain Goat Software, LLC

47

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

48

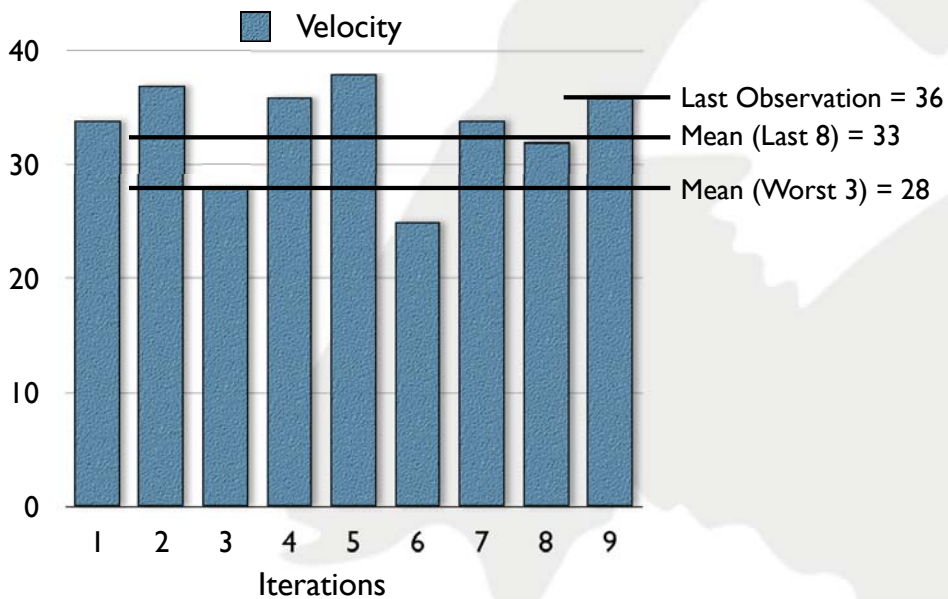
Changing the release plan



© Mountain Goat Software, LLC

49

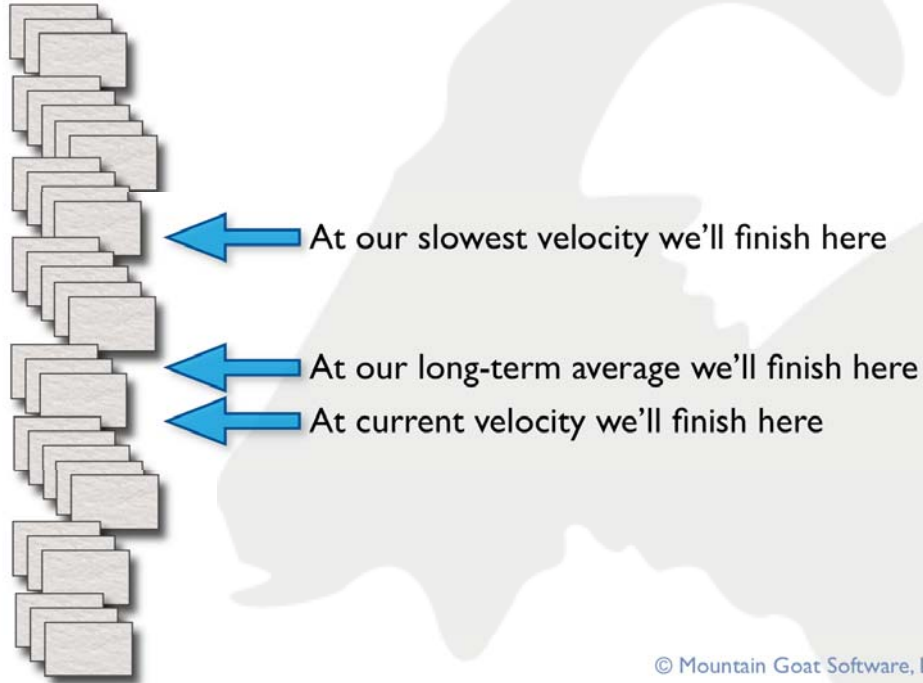
Look at velocity in a few ways



© Mountain Goat Software, LLC

50

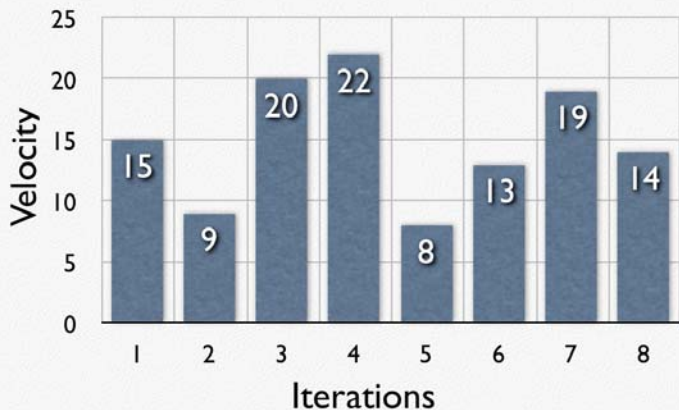
Extrapolate from velocity



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.



Mean of Worst 3 =
Most Recent =
Long-term Average = 15



Update this release plan

6 × worst 3 = _____ 6 × average of last 8 = _____ 6 × average of last 8 = _____

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...



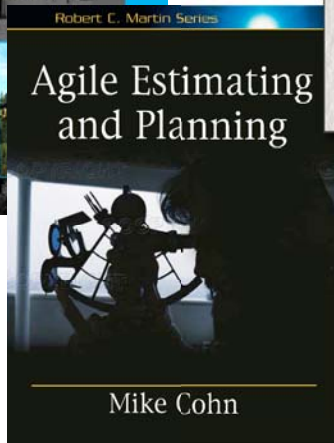
Upcoming public classes

Date	What	Where
November 7-8, 9	Certified ScrumMaster Agile Estimating & Planning	Santa Clara, CA
November 29-30	Certified Product Owner (with Ken Schwaber)	Boulder, CO
December 4-7	Certified ScrumMaster, User Stories for Agile Requirements Agile Estimating & Planning	London, UK
January 16-17	Certified ScrumMaster (with Ken Schwaber)	Orlando, FL

Register at
www.mountaingoatsoftware.com



Mike Cohn contact info



mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)



© Mountain Goat Software, LLC