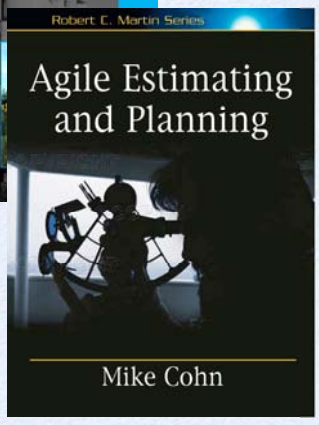
Agile Estimating and Planning

1

Mike Cohn - background



Consultant, author, and speaker

Founding member and director of Agile Alliance

Founder of Mountain Goat Software
-Process and project management training and consulting

2

Today's agenda

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

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?

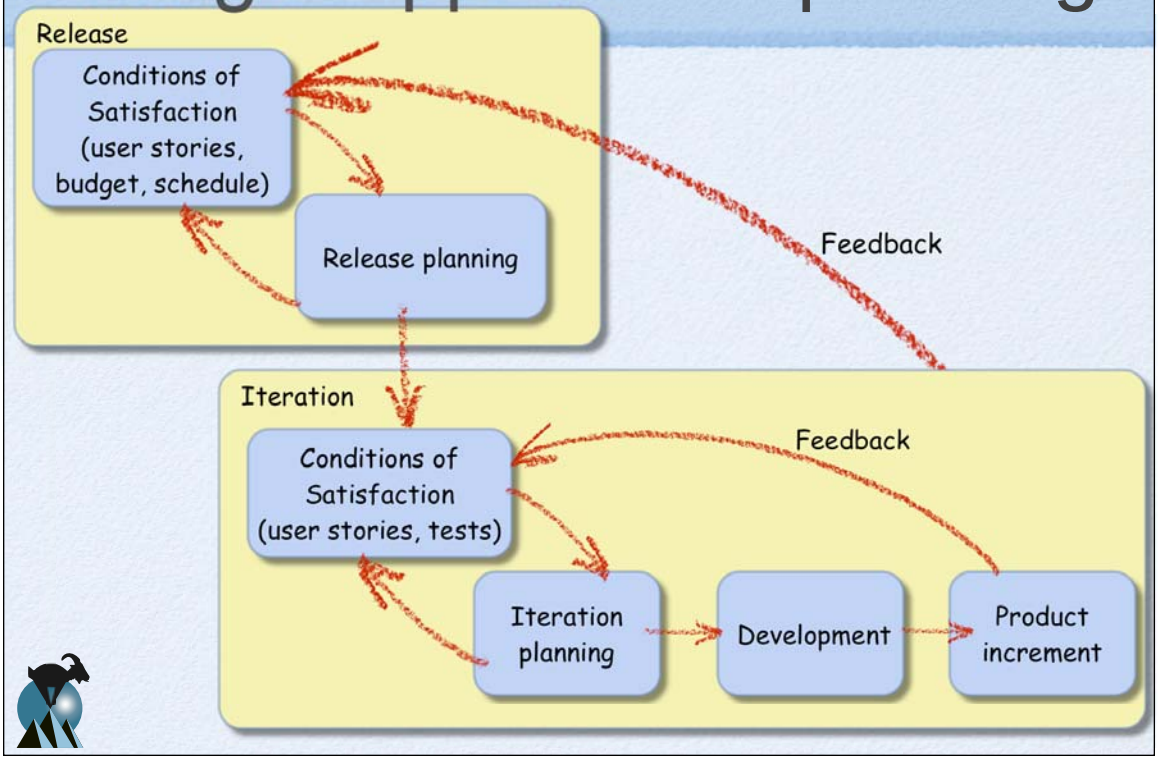
Is more focused on planning than the plan

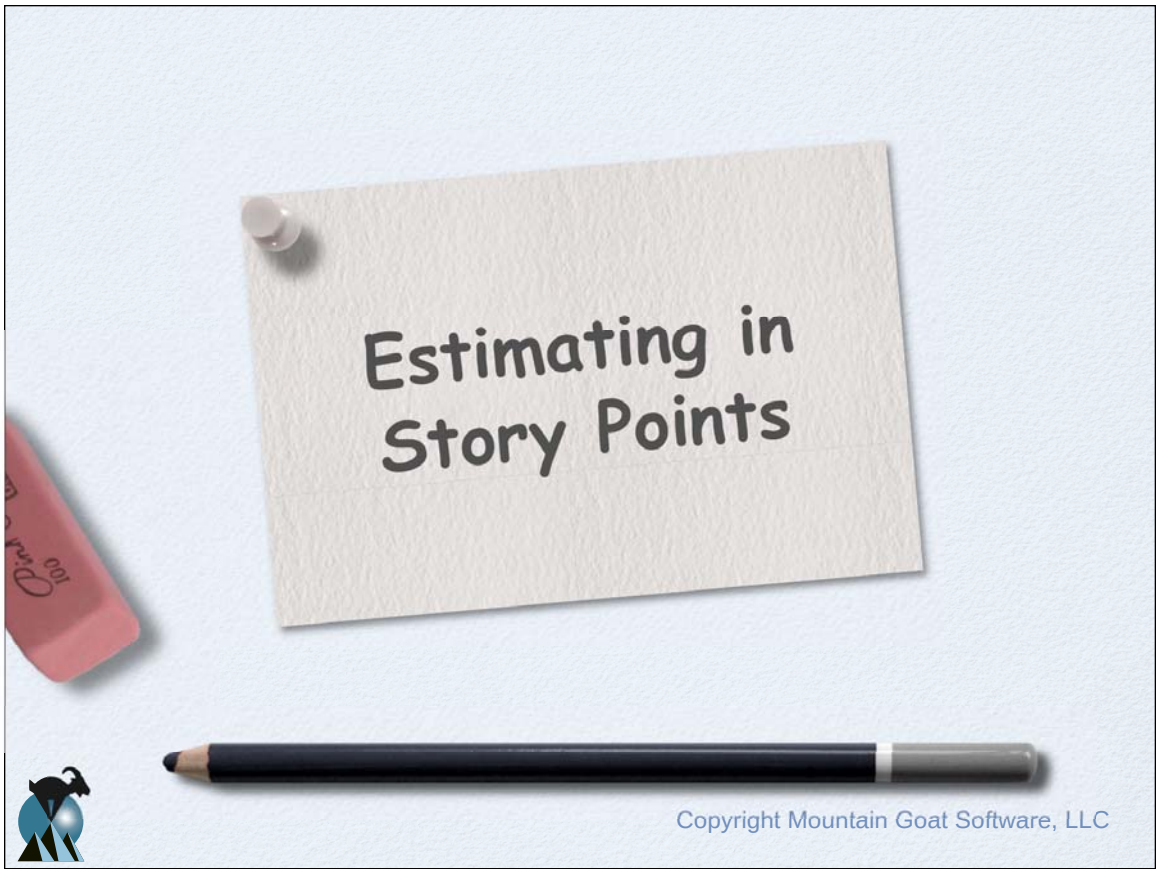
Encourages change

Results in plans that are easily changed

Is spread throughout the project

An agile approach to planning





Estimates on the product backlog

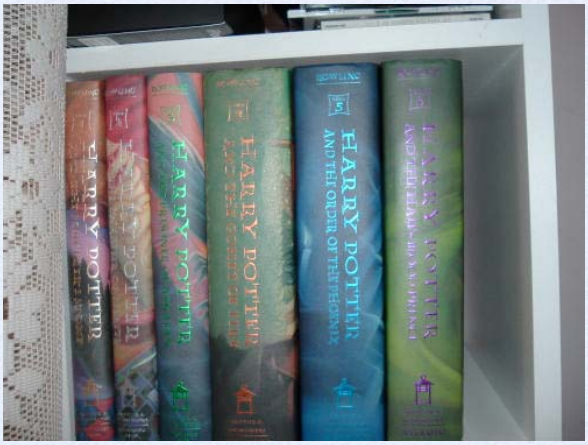
	Product Backlog	
Iteration 1	As a frequent flyer, I want to...	3
	As a frequent flyer, I want to...	5
Iteration 2	As a frequent flyer, I want to...	5
	As a frequent flyer, I want to...	2
	As a frequent flyer, I want to...	2

Iteration Backlog	
Code the UI	8
Write test fixture	6
Code middle tier	12
Write tests	5
Automate tests	4

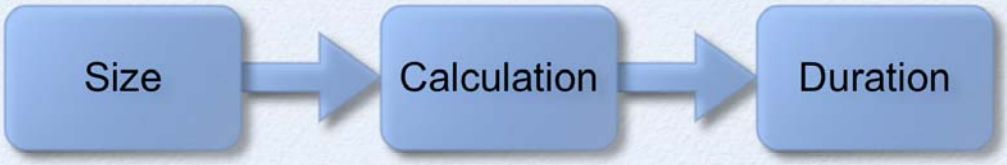
We're talking about these right now

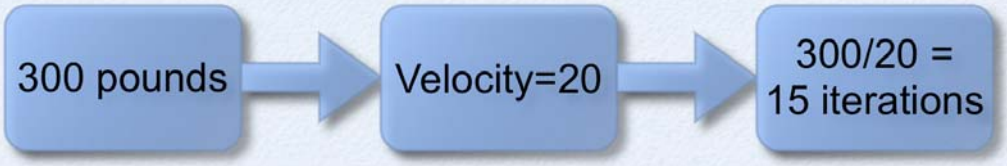
How long will it take...

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



Estimate size; derive duration



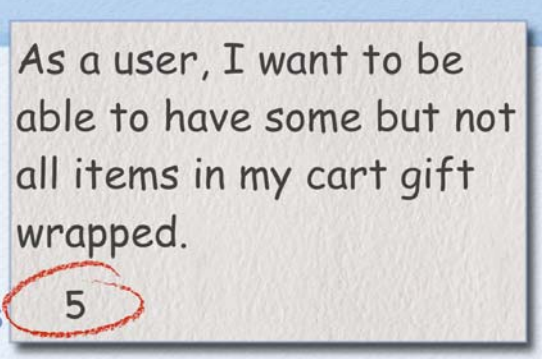


Measures of size

- Traditional and agile measure size differently

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

Assign "dog points" to the following breeds

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

13

Estimating in Ideal Time

14

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 basketball game is 40 minutes
  - Four 10-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

But instead

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

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.

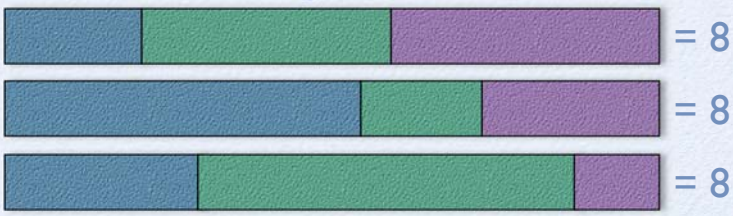
"How long will this take?"
☐ Are you answering what is being asked?

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

Story points or ideal days?

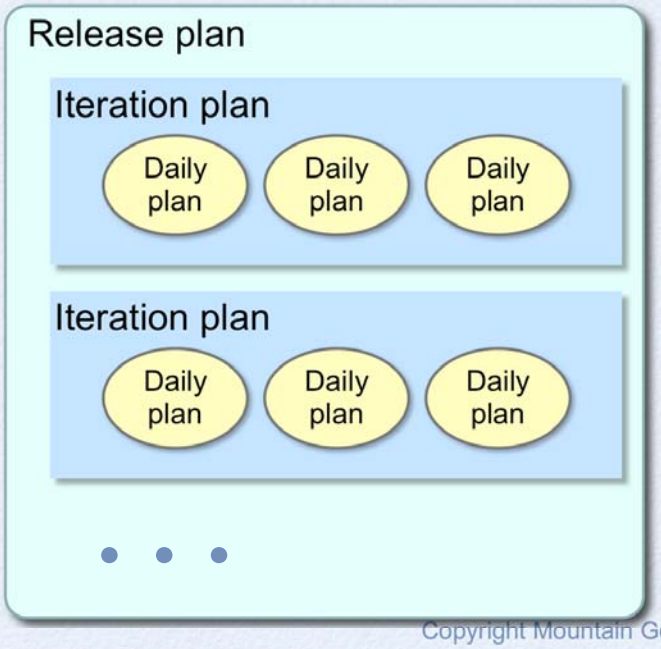
Comparing the approaches

- 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

- 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

Three levels of planning...



...three levels of precision

	Product Backlog		Iteration Backlog	
Iteration 1	As a frequent flyer, I want to...	30	Code the UI	8
	As a frequent flyer, I want to...	50	Write test fixture	6
Iteration 2	As a frequent flyer, I want to...	50	Code middle tier	12
	As a frequent flyer, I want to...	20	Write tests	5
	As a frequent flyer, I want to...	20	Automate tests	4

"Yesterday I started on the UI; I should finish before the end of today."

Approaches to estimating

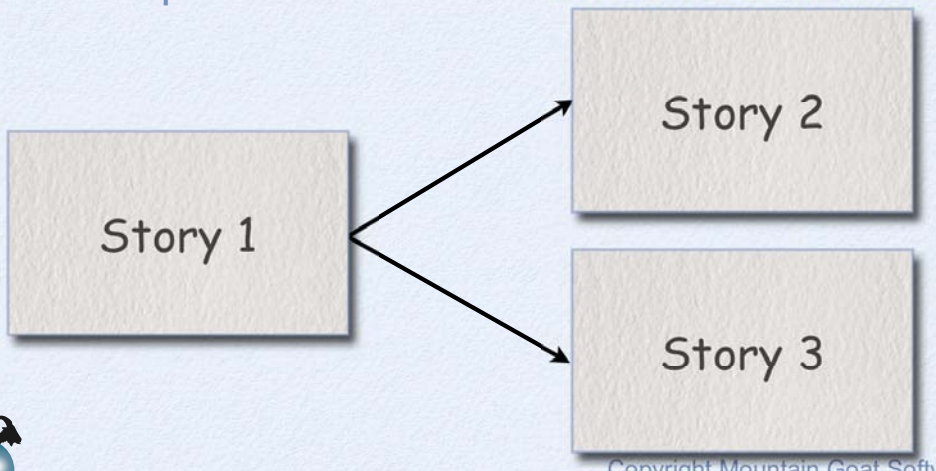
- Analogy
- Disaggregation
- Planning poker

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

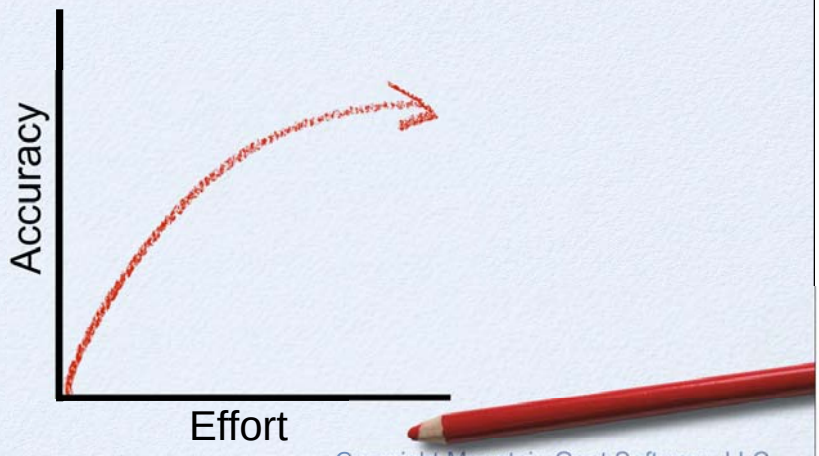


Disaggregation

- Breaking a big story into smaller stories or tasks
- 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

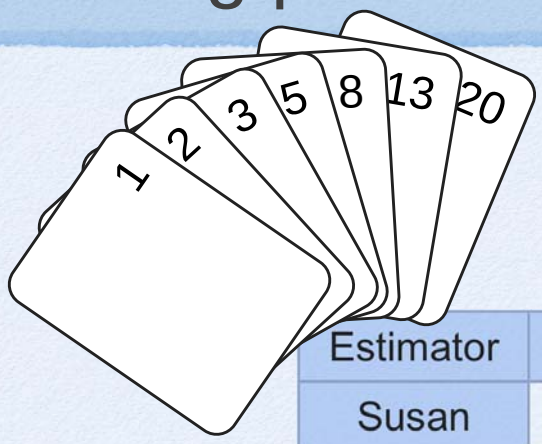
- Can you distinguish a 1-point story from a 2?
- Can you distinguish a 17 from an 18?
- Use units that make sense, such as
  - 1, 2, 3, 5, 8
  - 1, 2, 4, 8
- Stay mostly in a 1-10 rang

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

Remodeling my kitchen

1. Install new hardwood floor
2. Refinish (remove, sand, repaint) the cabinets
3. Replace my tile countertop with granite
4. Repaint entire kitchen
5. Lay shelf paper
6. Install recessed lighting
7. Replace electric stove with gas stove
8. Install built-in refrigerator
9. Install a new oven
10. Plumb the island and add sink

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

35

Iteration planning

Product Backlog

As a frequent flyer, I want to...	3
As a frequent flyer, I want to...	5
As a frequent flyer, I want to...	5
As a frequent flyer, I want to	2
As	

Iteration 1

Iteration 2

Iteration Backlog

Code the UI	8
Write test fixture	6
Code middle tier	12
Write tests	5
Automate tests	4

Creating this is iteration planning

"Yesterday I started on the UI; I should finish before the end of today."

36

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

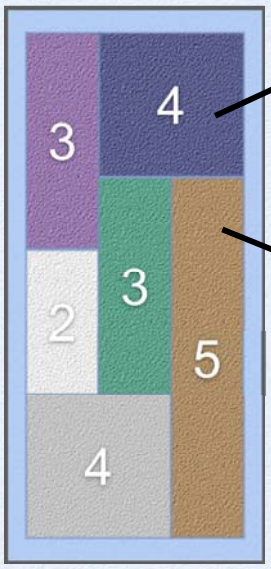
Commitment-driven iteration planning

- 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

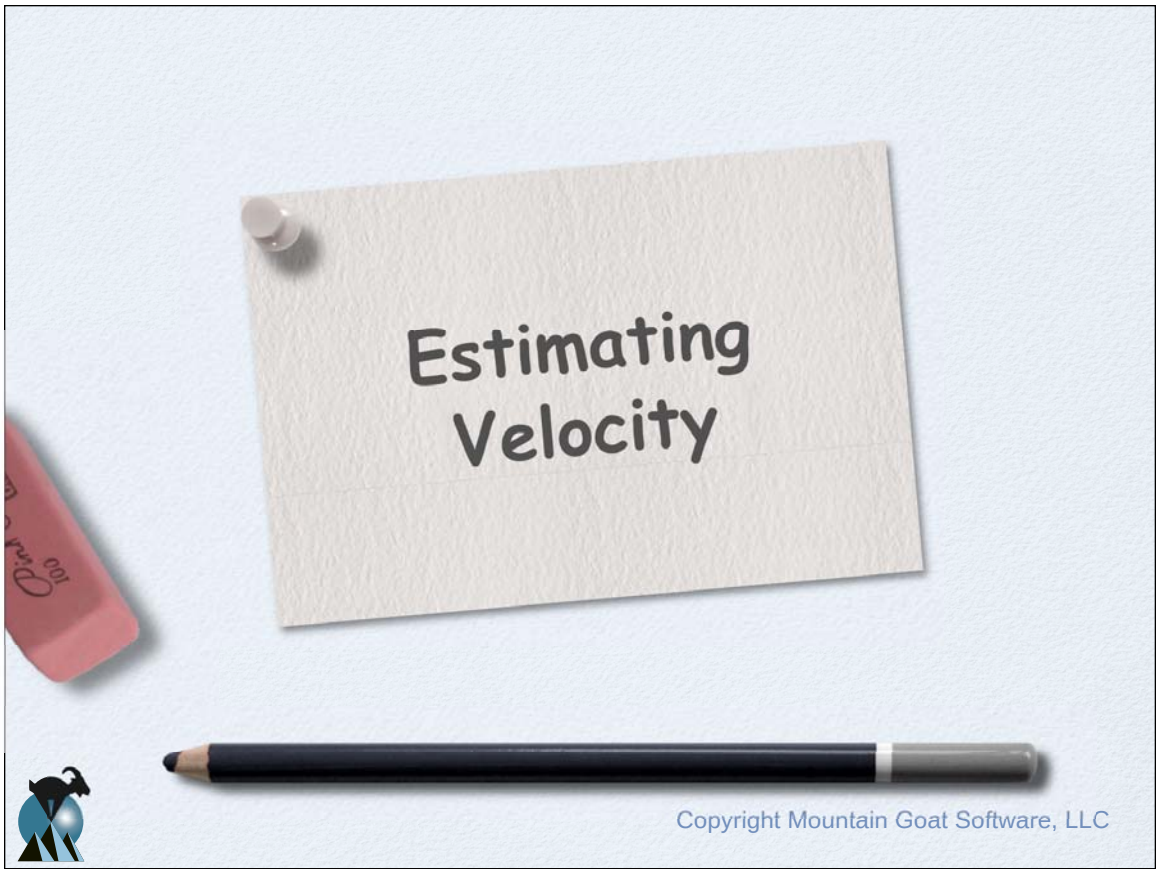
Person	Hours per Day	Hours per Iteration
Sergey	4-6	40-60
Yuri	5-7	50-70
Carina	2-3	20-30
Total		110-160

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)



Predicting velocity

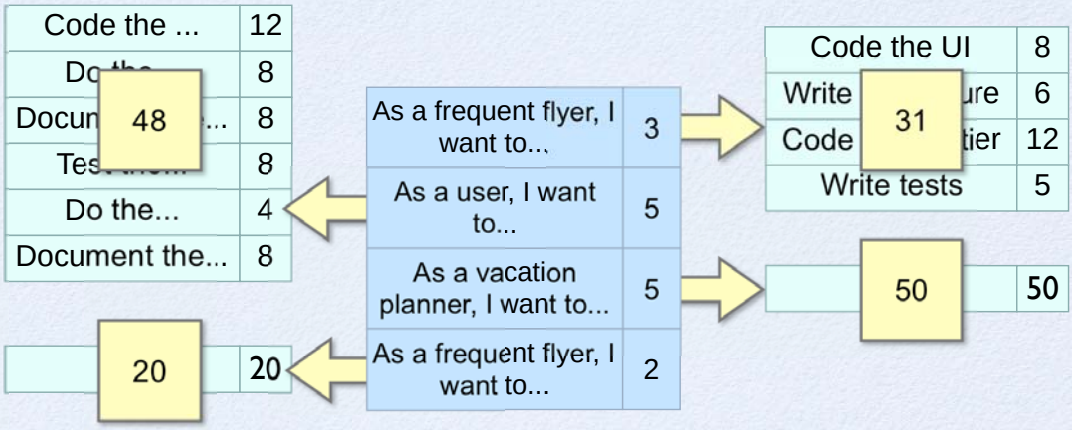
- Three ways to come up with a velocity:
  - Use historicals
  - Run a few iterations
  - Forecast it
- Express velocity as a range that matches your uncertainty in it

Forecasting velocity

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

Ideally, "plan" more than one iteration

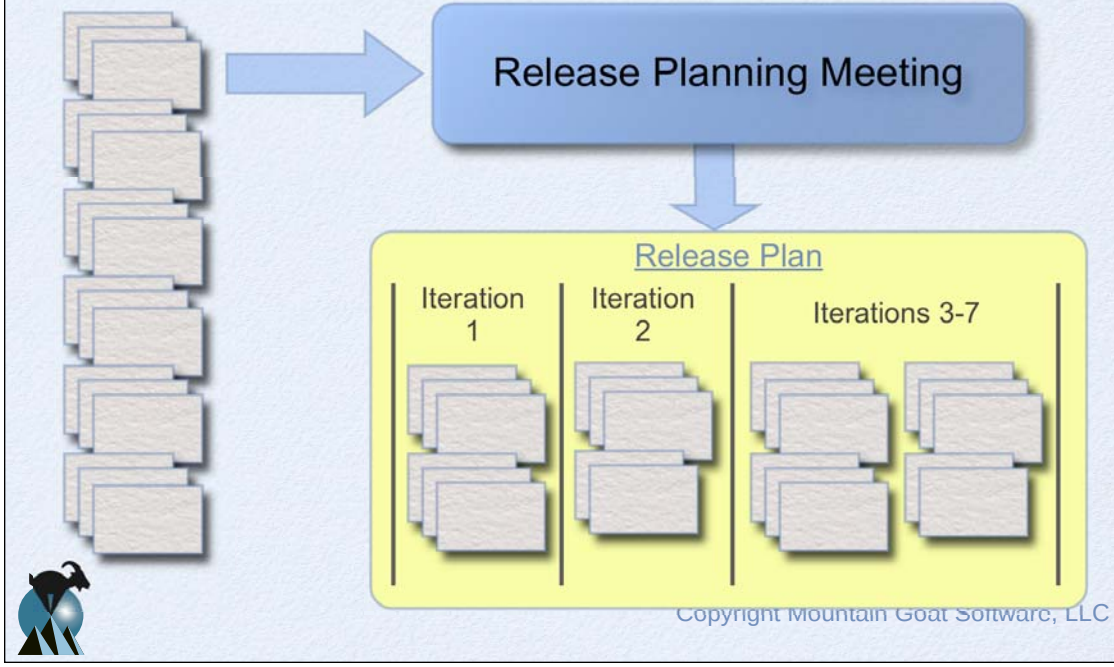
An example



At 110-160 available hours per iteration, what is the team's velocity?

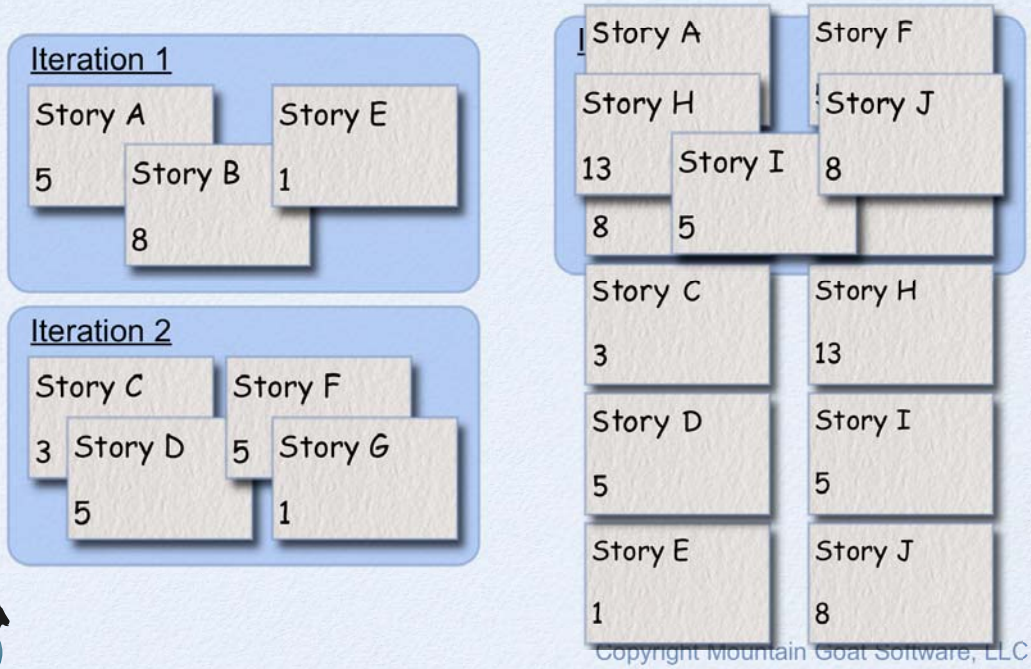
45

Release planning



46

An example with velocity=14



Updating the release plan

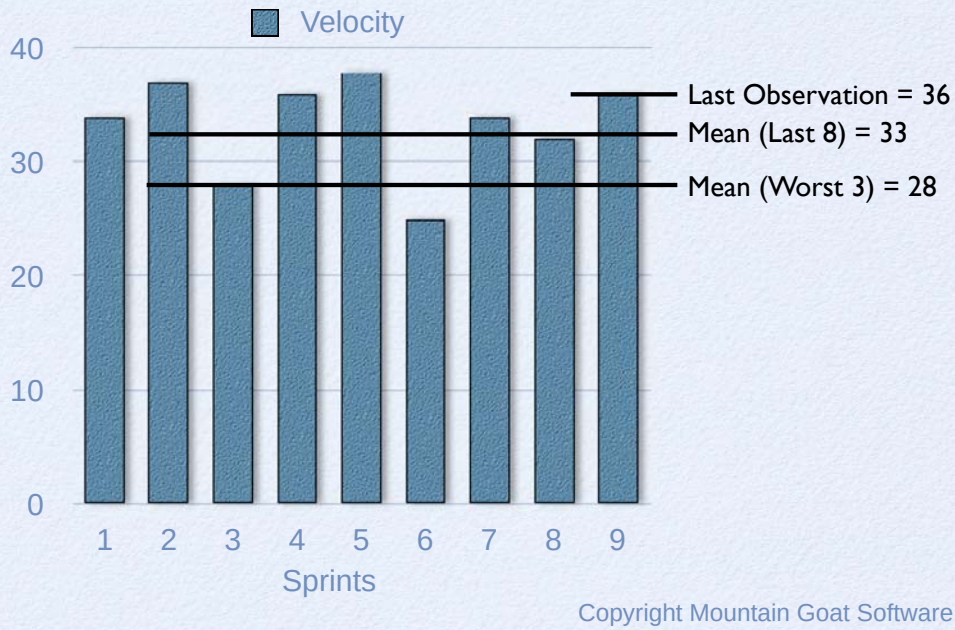
- Revisit the release plan at the end of every sprint
- Update it based on:
  - Current understanding of velocity
  - Current prioritization of the product backlog
- This should be a very short and sweet process

A simple updating example

	Story A	5
	Story B	3
	Story C	5
	Story F	3
	Story D	5
	Story E	5
	Story G	3
	Story I	3
	Story H	5
	Story J	2
	Story K	5
	Story L	3

✓	Story A	5
✓	Story B	3
✓	Story C	5
	Story F	3
	Story D	5
	Story E	5
	Story G	3
	Story I	3
	Story H	5
	Story J	2
	Story K	5
	Story L	3

Look at velocity in a few ways



Extrapolate from velocity

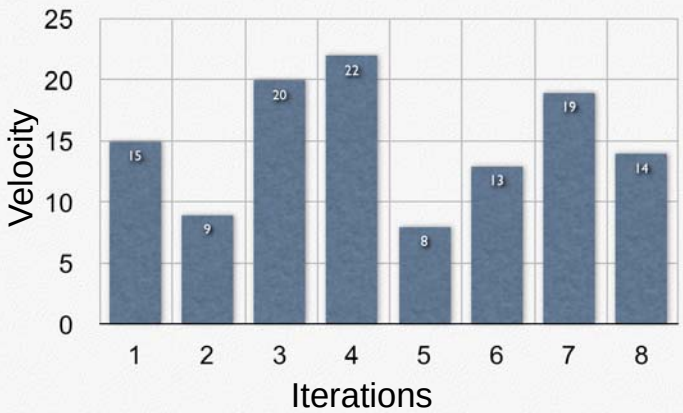
At our slowest velocity we'll finish here

At current velocity we'll finish here

At our long-term average we'll finish here

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 = 14

Update this release plan

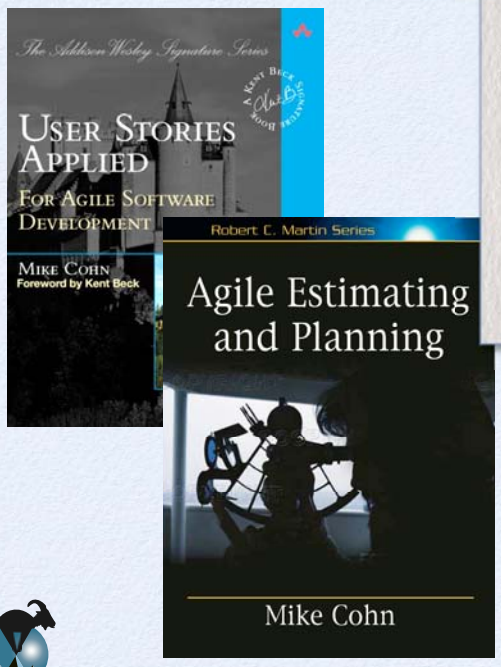
Running Total	Estimate	Story
5	5	As a user, I can...
10	5	As a user, I can...
23	13	As a user, I can...
31	8	As a user, I can...
51	20	As a user, I can...
59	8	As a user, I can...
64	5	As a user, I can...
72	8	As a user, I can...
77	5	As a user, I can...
85	8	As a user, I can...
90	5	As a user, I can...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

Upcoming public classes

Date	What	Where
August 1-2	Certified ScrumMaster	Denver
August 3	Agile Estimating & Planning	Denver
November 7-8	Certified ScrumMaster	Santa Clara
November 9	Agile Estimating & Planning	Santa Clara
November 29-30	Certified Product Owner (with Ken Schwaber)	Boulder

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