Agile Estimating and Planning

Mike Cohn

December 4, 2007

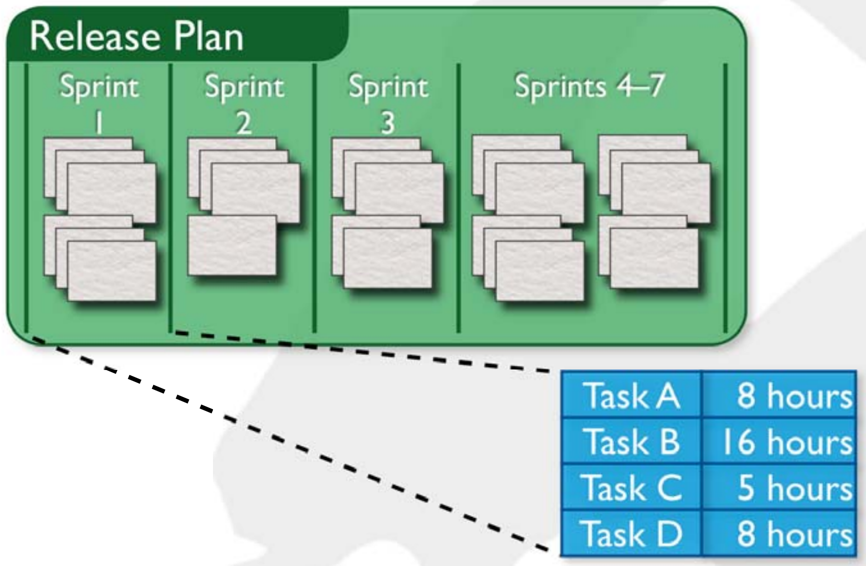
Mike Cohn - background

The Addison-Wesley Signature Series
A Kent Beck Signature Book
User Stories Applied
For Agile Software Development
Mike Cohn
Foreword by Kent Beck
Robert C. Martin Series
Agile Estimating and Planning
Mike Cohn

Agile coach and trainer

- Founding member and director of Agile Alliance and Scrum Alliance
- Founder of Mountain Goat Software
- Ran my first Scrum project back in 1995
- Typical programmer to manager etc. progression

Release and sprint planning



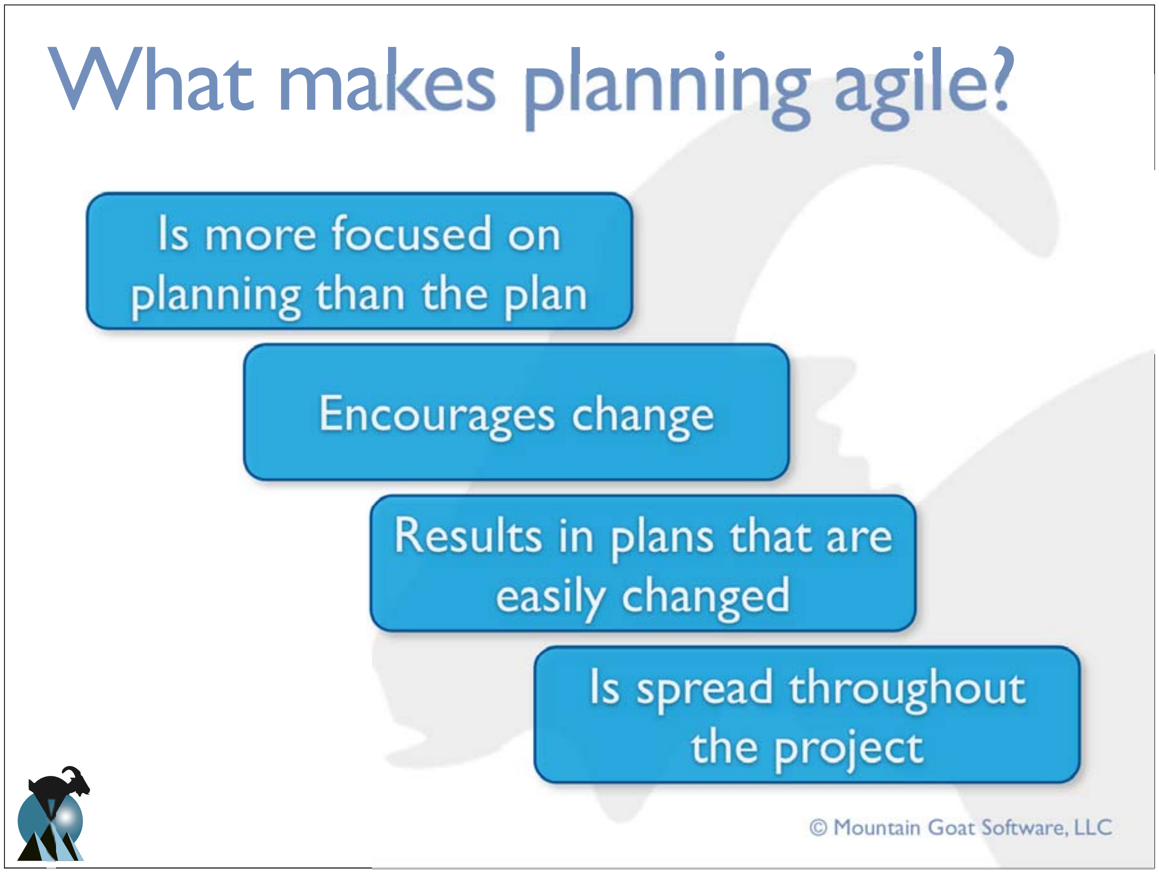
Task A	8 hours
Task B	16 hours
Task C	5 hours
Task D	8 hours

What's a good plan?

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

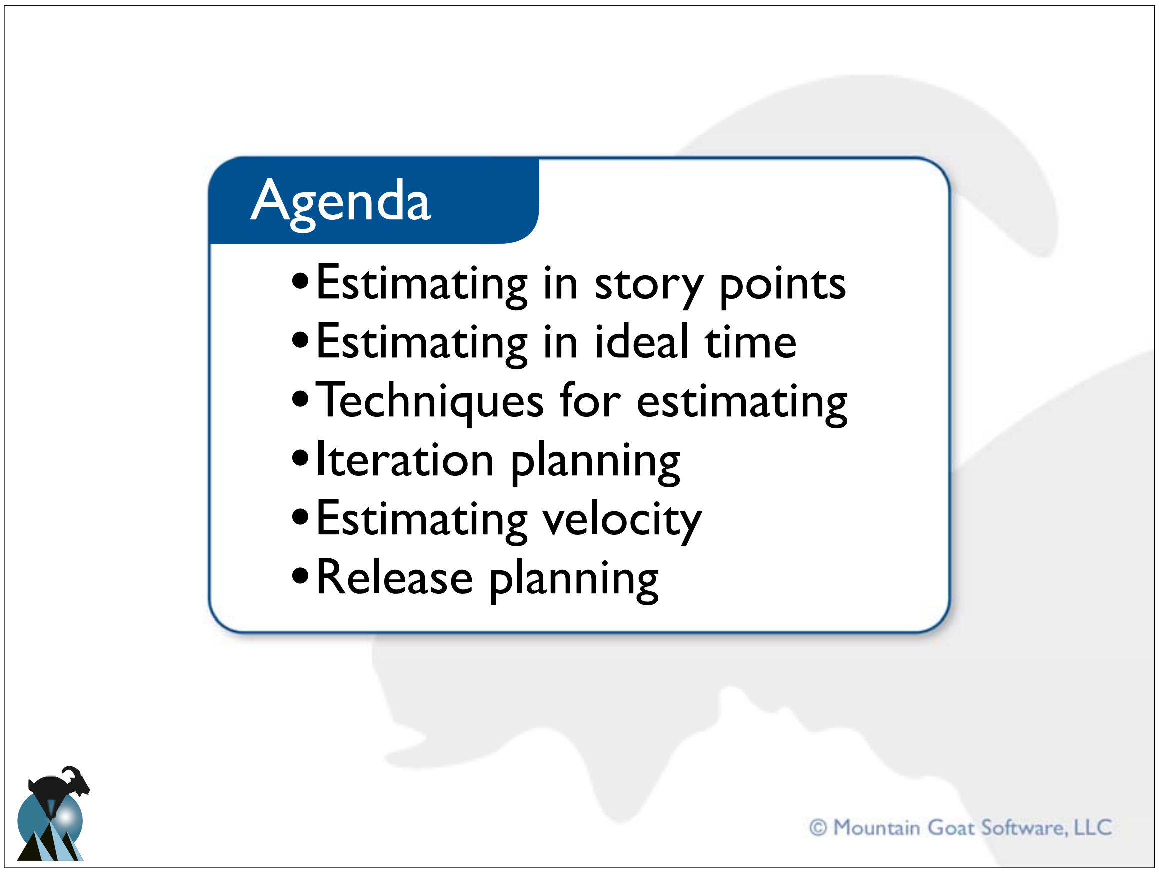
"It's better to be roughly right than precisely wrong."

~John Maynard Keynes



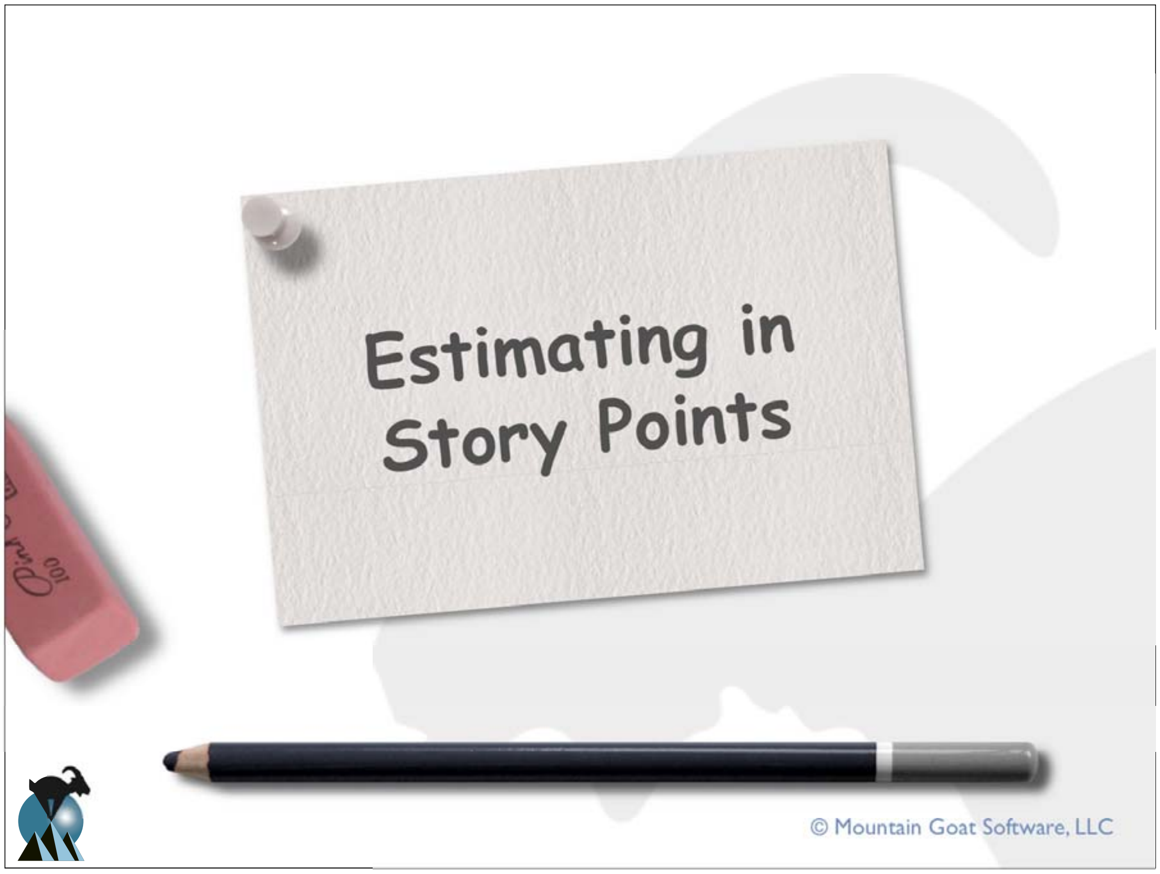
What makes planning agile?

- Is more focused on planning than the plan
- Encourages change
- Results in plans that are easily changed
- Is spread throughout the project



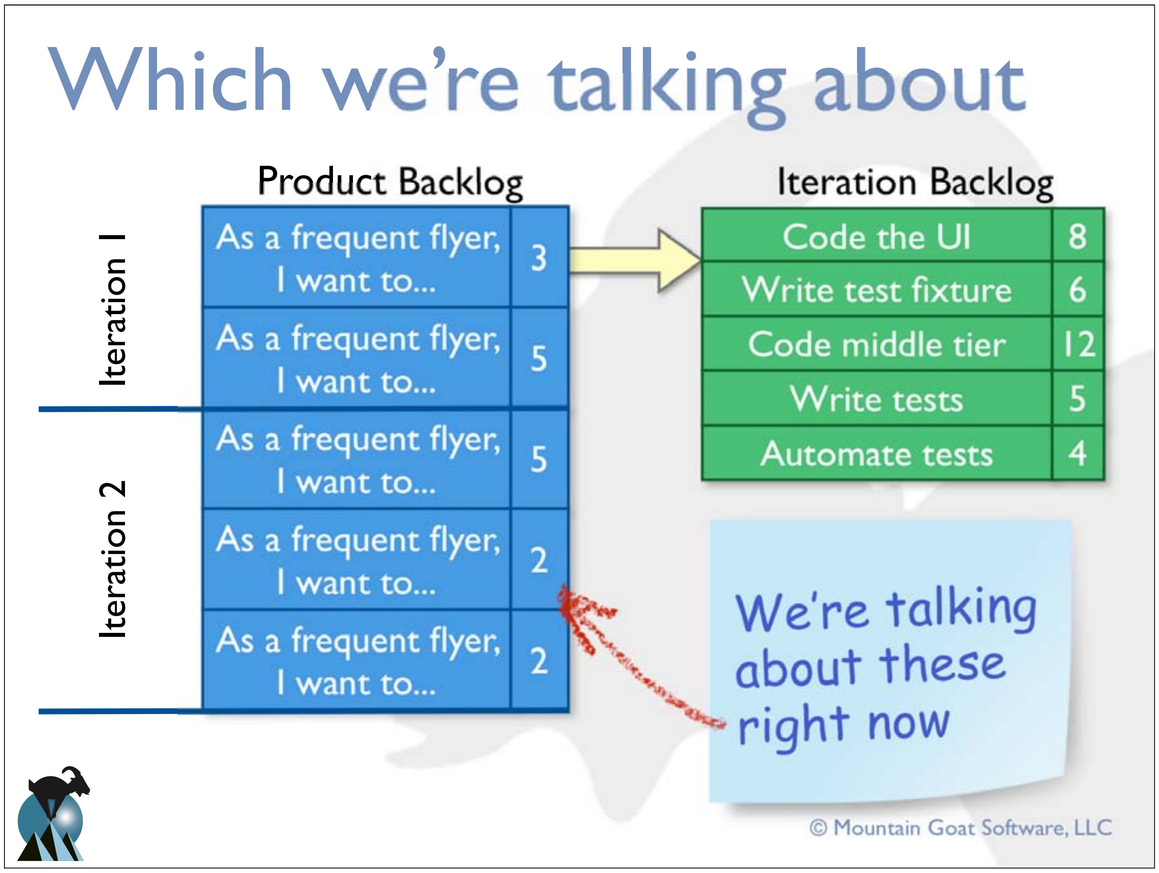
Agenda

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



Estimating in Story Points

© Mountain Goat Software, LLC



Which we're talking about

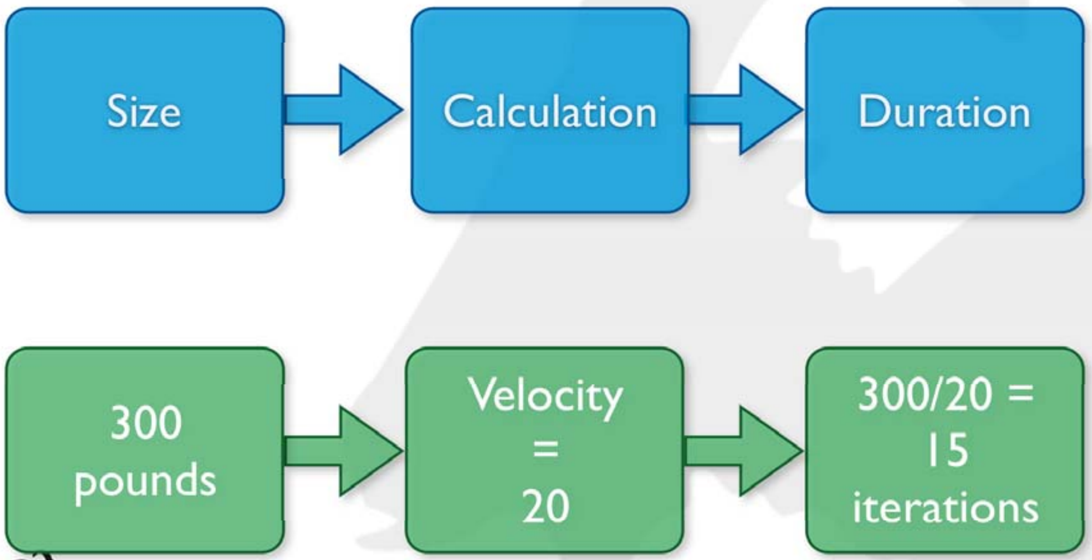
Product Backlog
Iteration 1
As a frequent flyer, I want to... 3
As a frequent flyer, I want to... 5
Iteration 2
As a frequent flyer, I want to... 5
As a frequent flyer, I want to... 2
As a frequent flyer, I want to... 2
Iteration Backlog
Code the UI 8
Write test fixture 6
Code middle tier 12
Write tests 5
Automate tests 4
We're talking about these right now

© Mountain Goat Software, LLC

How long will it take...

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

Estimate size; derive duration



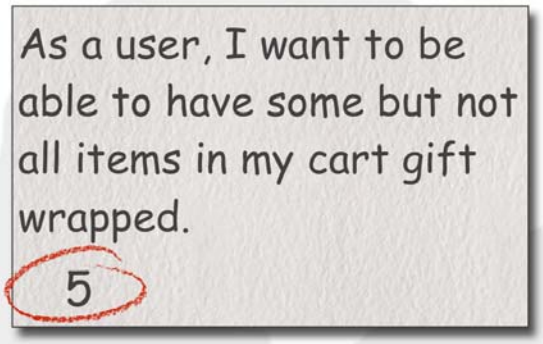
Measures of size

- Traditional and agile measure size differently

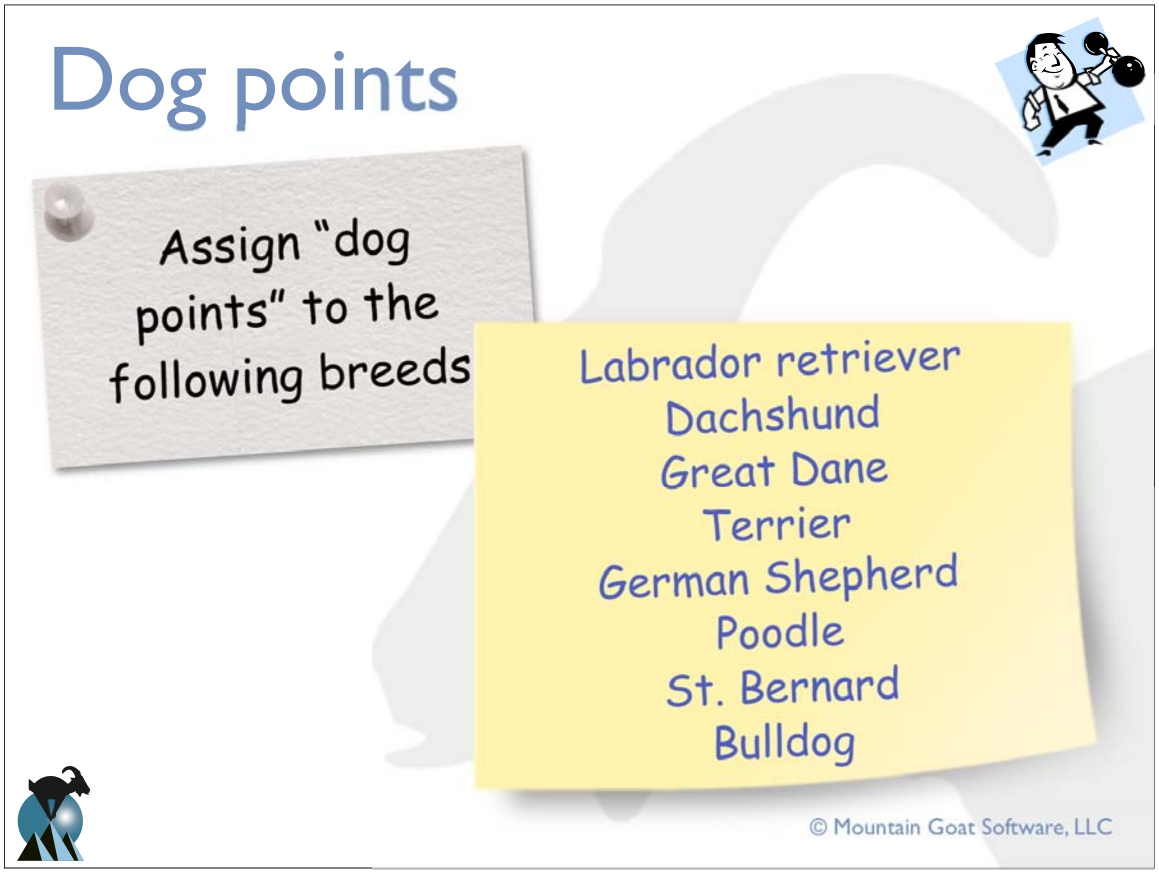
Traditional measures of size
Lines of Code
Function Points
Agile measures of size
Story points
Ideal days

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.

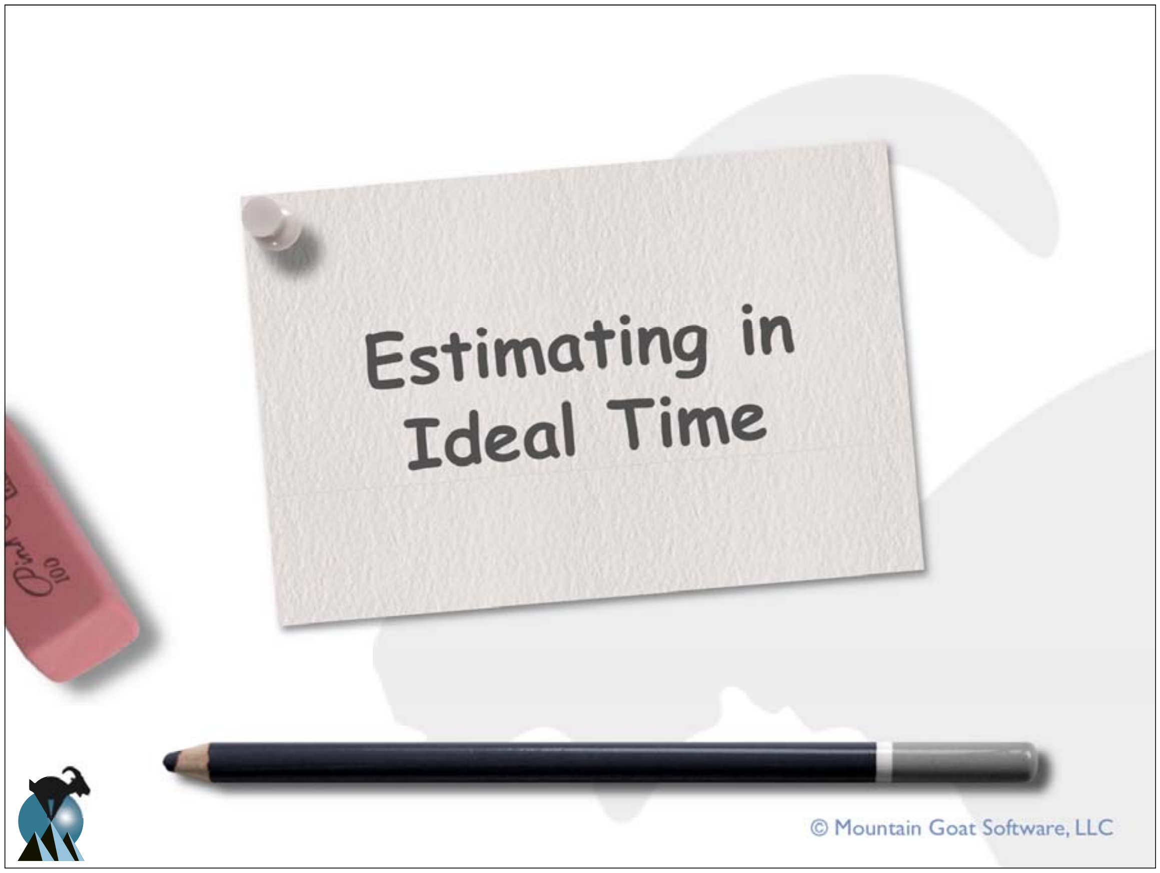
5



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

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?"

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

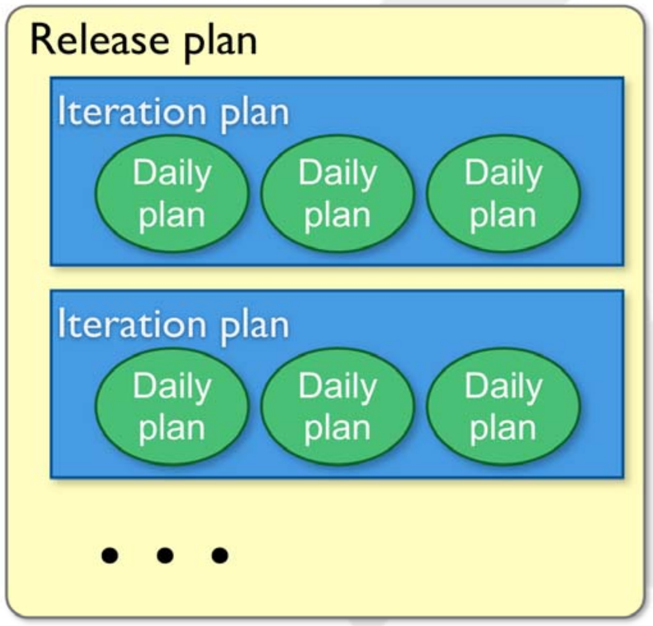
Comparing the approaches

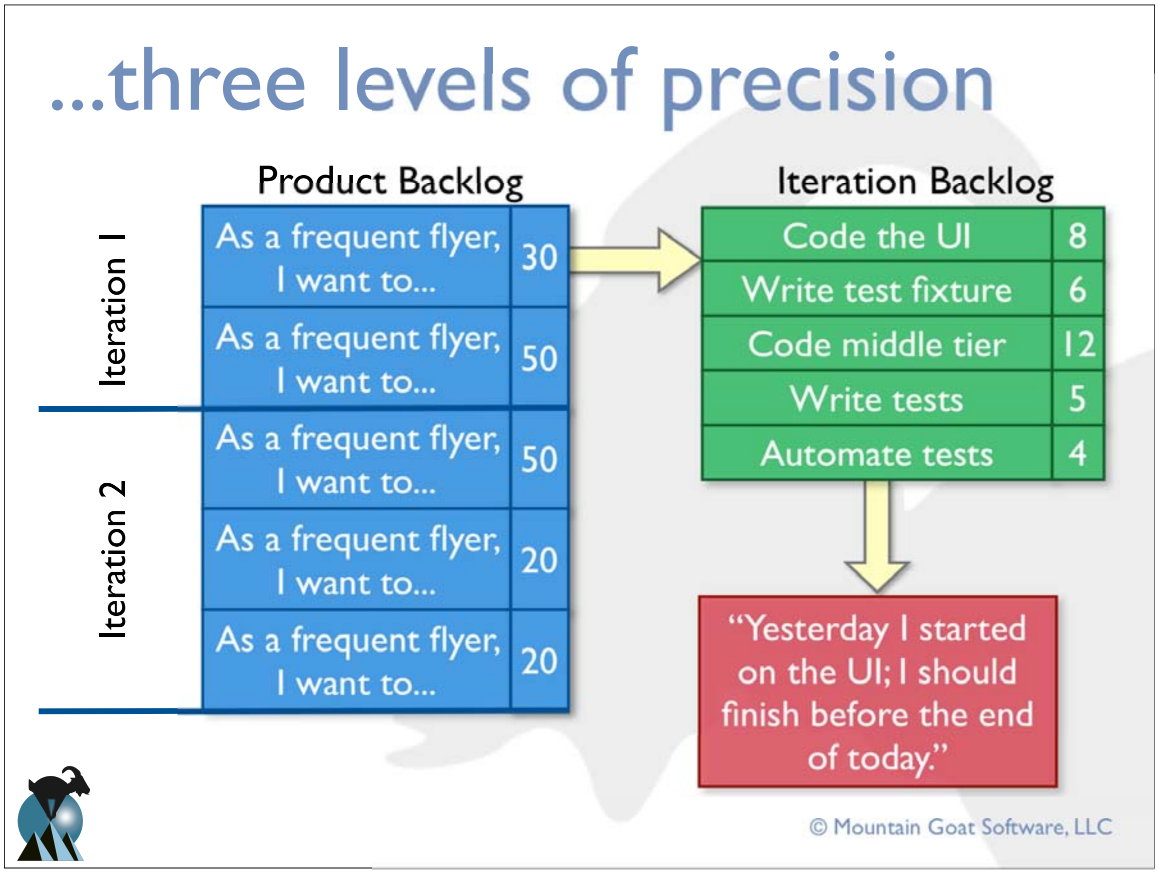
- Story points help drive cross-functional behavior
- Story point estimates do not decay
- Story points are a pure measure of size
- Estimating in story points is typically faster
- My ideal days cannot be added to your ideal days
- Ideal days are easier to explain outside the team
- Ideal days are easier 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 1

As a frequent flyer, I want to...	30
As a frequent flyer, I want to...	50

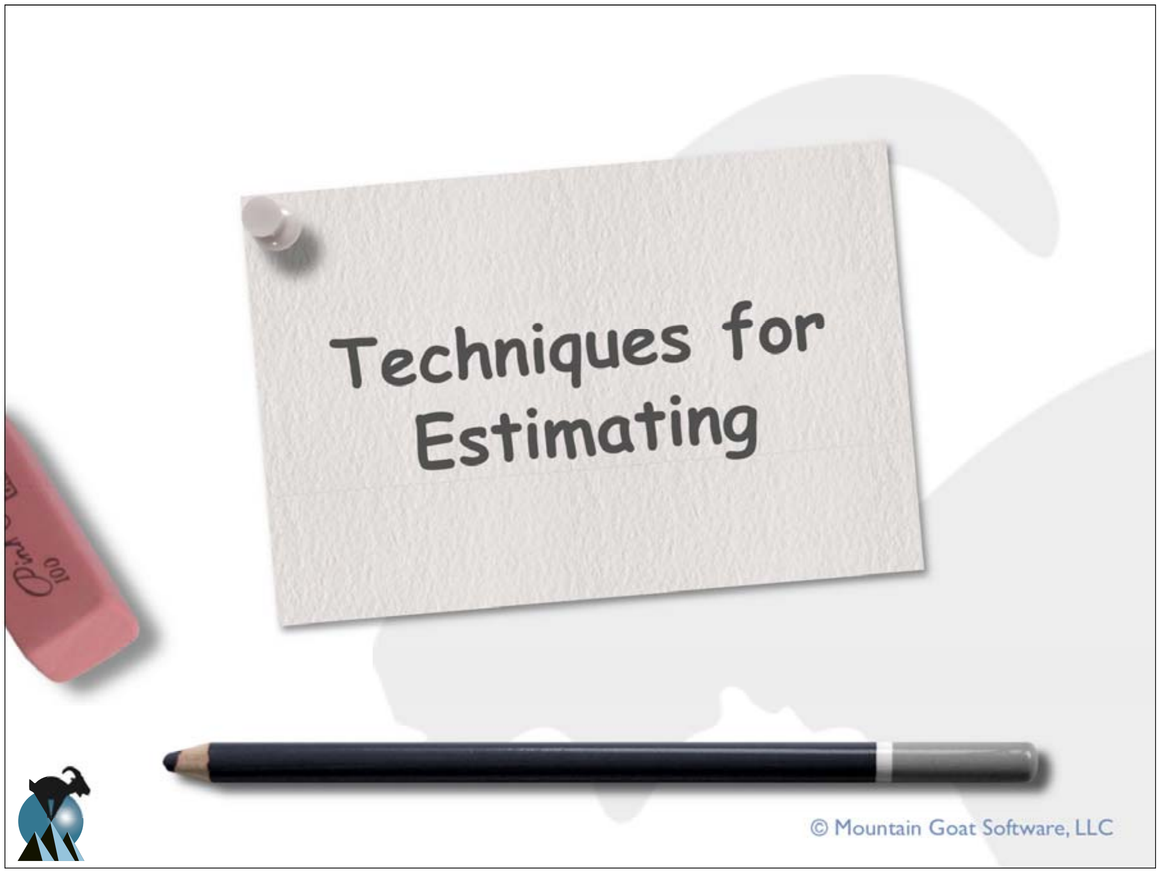
Iteration 2

As a frequent flyer, I want to...	50
As a frequent flyer, I want to...	20
As a frequent flyer, I want to...	20

Iteration Backlog

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

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



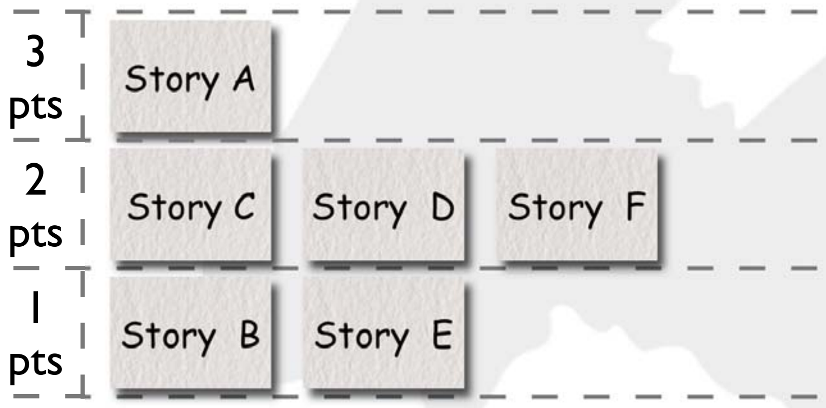
Techniques for Estimating

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

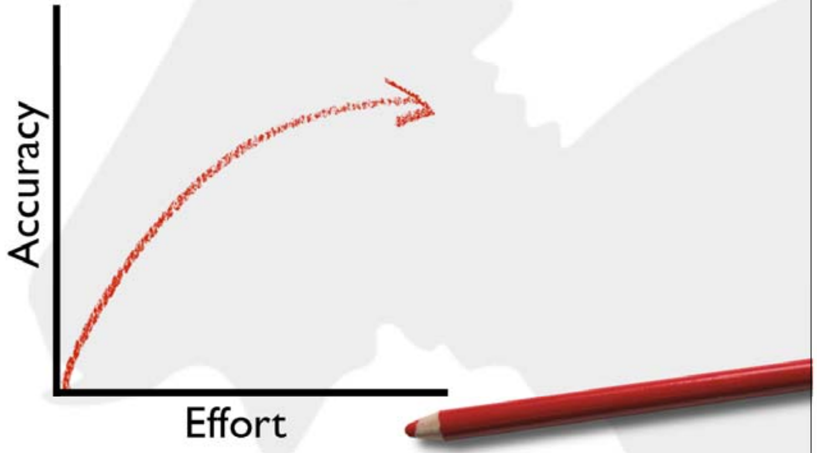


Disaggregation

- 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, 13
  - 1, 2, 4, 8
- Stay mostly in a 1-10 range

Include 0 and ½ if you want

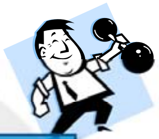
Planning poker

- 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

Estimate these

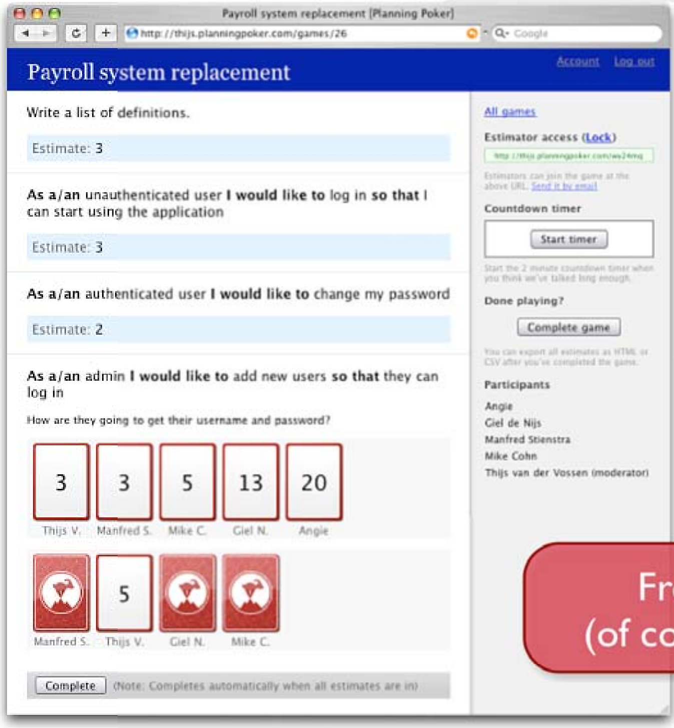


Product backlog item	Estimate
Read a high-level, 10-page overview of agile software development in *People* magazine.	
Read a densely written 5-page research paper about agile software development in an academic journal.	
Write the product backlog for a simple eCommerce site that sells only clocks.	
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.	

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

www.planningpoker.com


Payroll system replacement [Planning Poker]
http://thijs.planningpoker.com/games/26
Google
Account
Log out
All games
Start timer
Complete game
3
3
5
13
20
5
Complete

Payroll system replacement

Write a list of definitions.

Estimate: 3

As a/an unauthenticated user **I would like to** log in **so that** I can start using the application

Estimate: 3

As a/an authenticated user **I would like to** change my password

Estimate: 2

As a/an admin **I would like to** add new users **so that** they can log in

How are they going to get their username and password?

Thijs V. Manfred S. Mike C. Giel N. Angie

Manfred S. Thijs V. Giel N. Mike C.

(Note: Completes automatically when all estimates are in)

Estimator access (Lock)

http://thijs.planningpoker.com/ws24mq

Estimators can join the game at the above URL. Send it by email

Countdown timer

Start the 2 minute countdown timer when you think we've talked long enough.

Done playing?

You can export all estimates as HTML or CSV after you've completed the game.

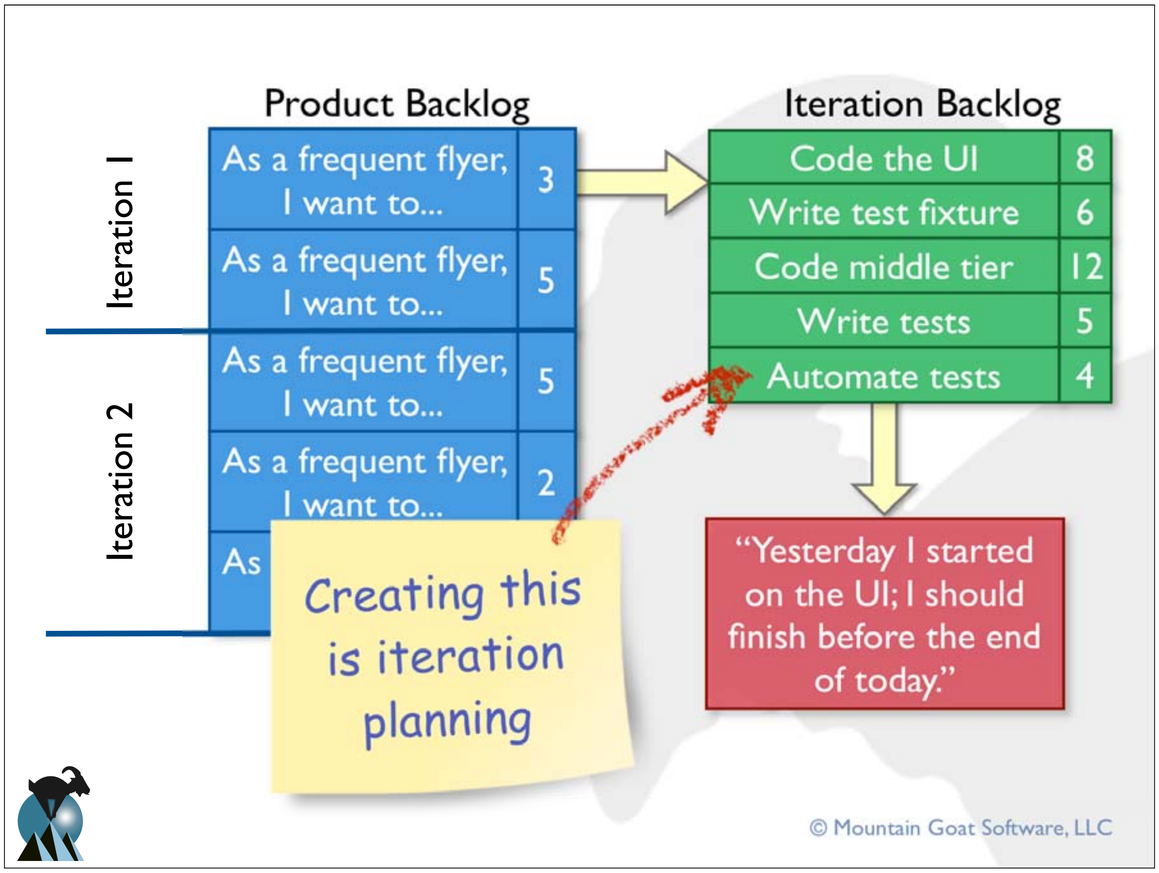
Participants

Angie
Giel de Nijs
Manfred Stienstra
Mike Cohn
Thijs van der Vossen (moderator)

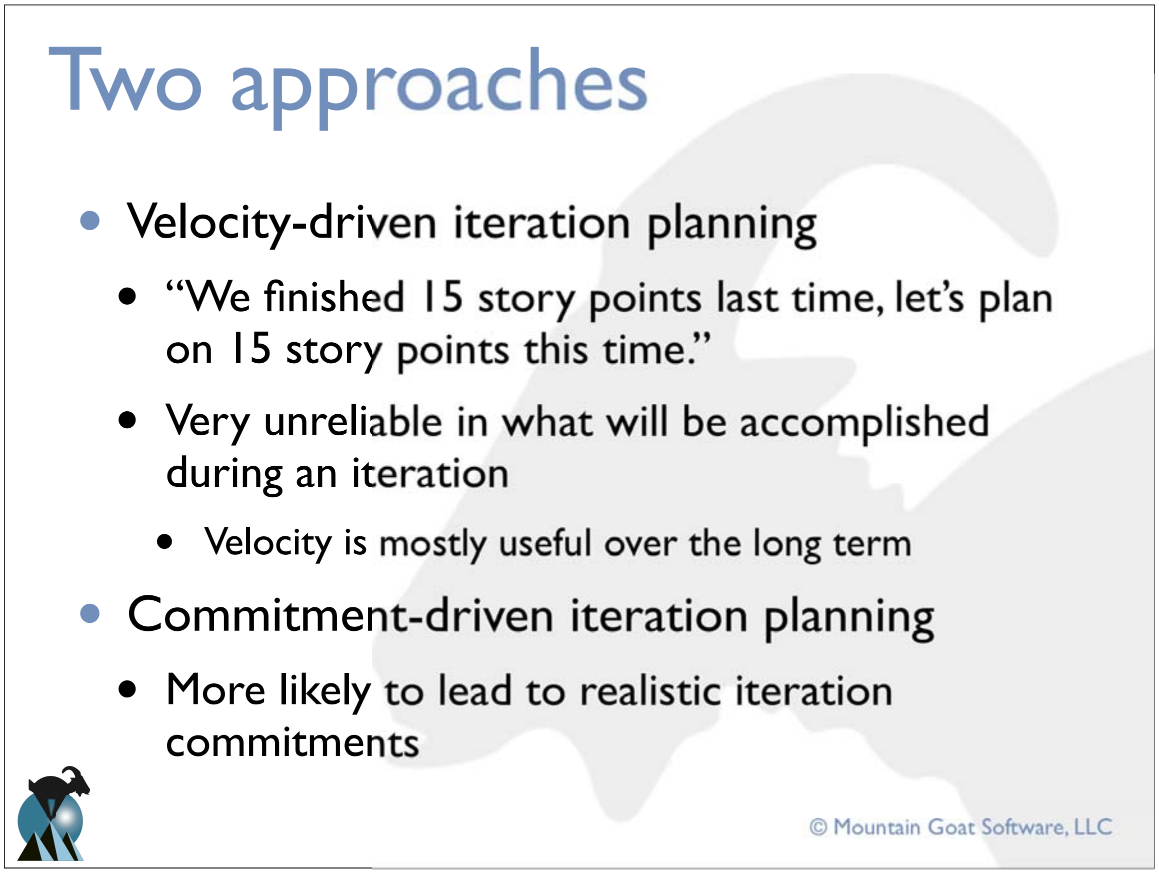
Free
(of course)

Iteration Planning

33

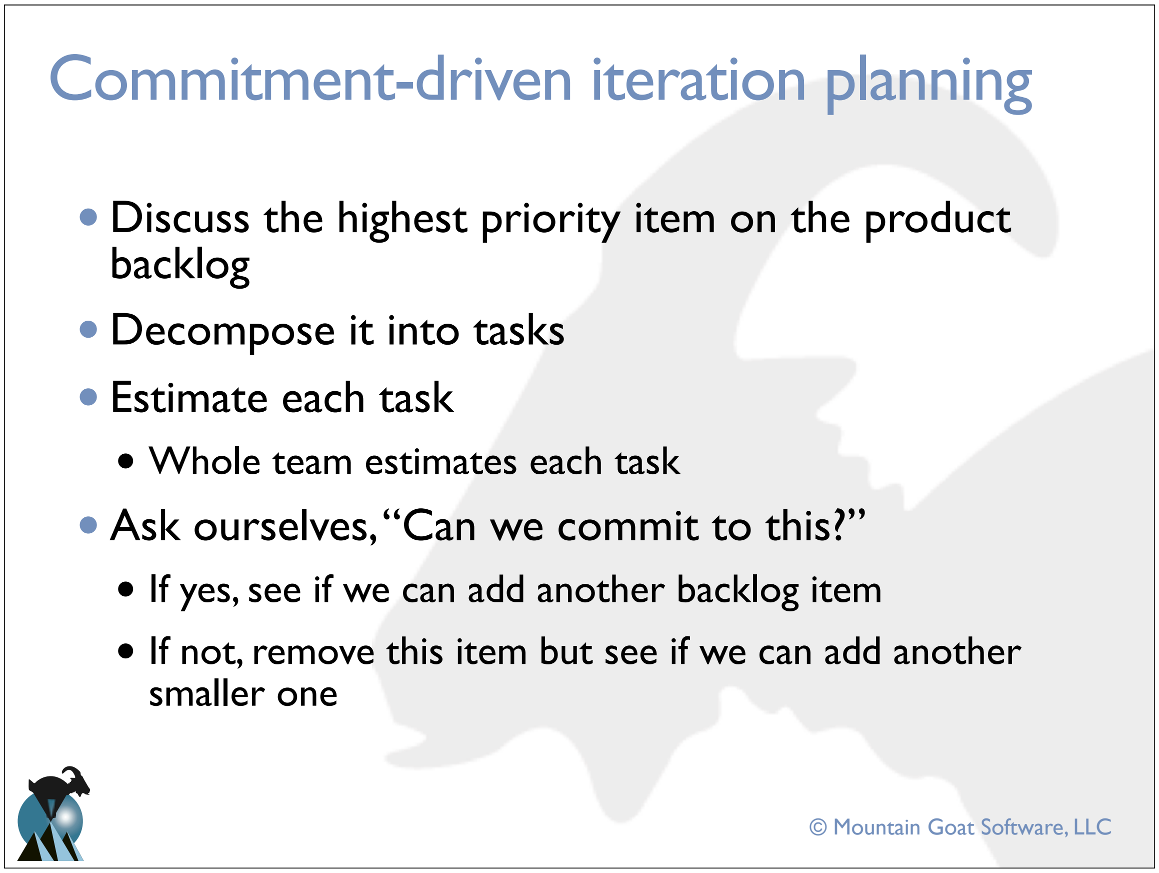

Product Backlog
Iteration Backlog
Iteration 1
Iteration 2
As a frequent flyer, I want to... 3
As a frequent flyer, I want to... 5
As a frequent flyer, I want to... 5
As a frequent flyer, I want to... 2
As
Code the UI 8
Write test fixture 6
Code middle tier 12
Write tests 5
Automate tests 4
Creating this is iteration planning
"Yesterday I started on the UI; I should finish before the end of today."

34



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



Commitment-driven iteration planning

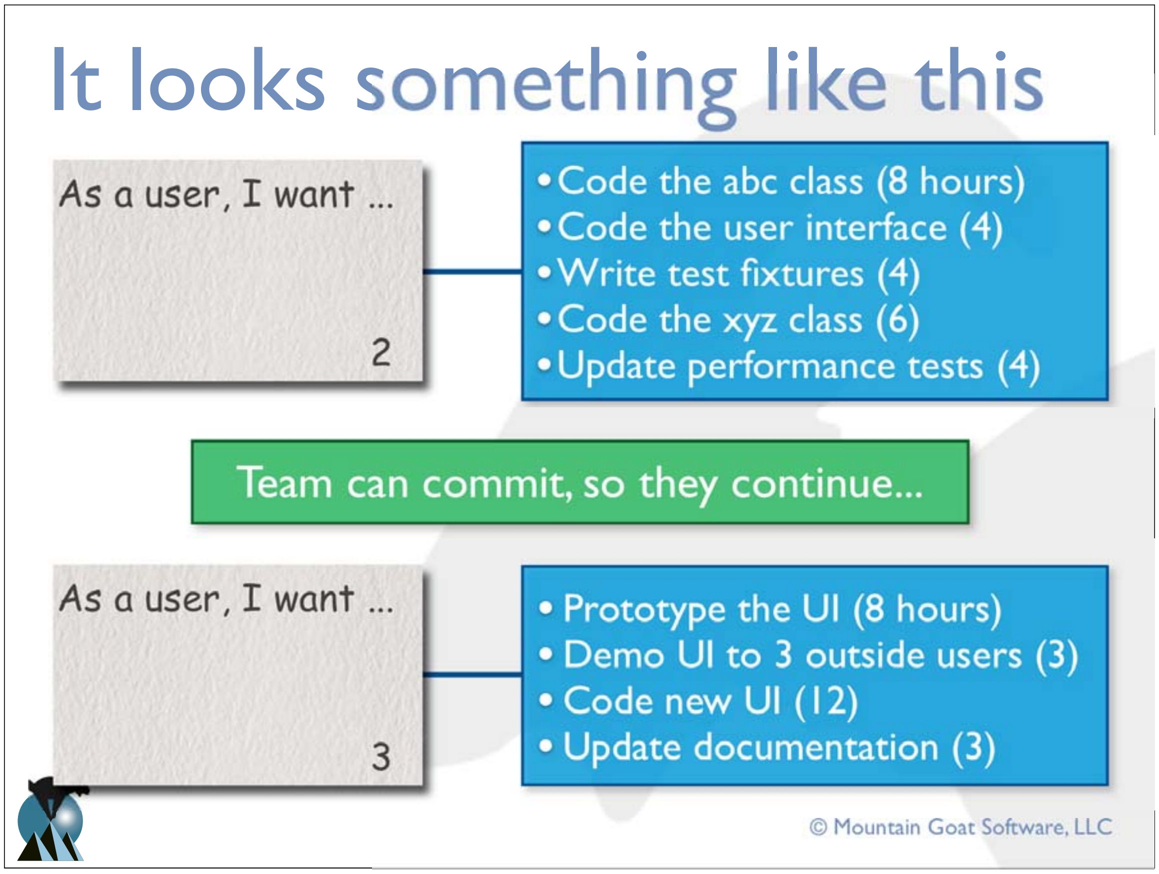
- Discuss the highest priority item on the product backlog
- Decompose it into tasks
- Estimate each task
  - Whole team estimates each task
- Ask ourselves, “Can we commit to this?”
  - If yes, see if we can add another backlog item
  - If not, remove this item but see if we can add another smaller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

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

As a user, I want ...

2

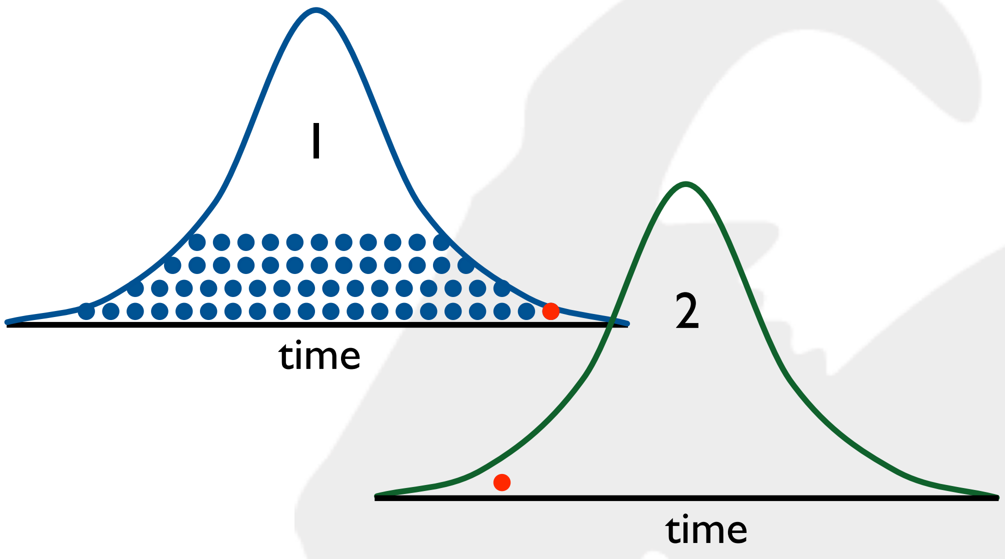
- Code the abc class (8 hours)
- Code the user interface (4)
- Write test fixtures (4)
- Code the xyz class (6)
- Update performance tests (4)

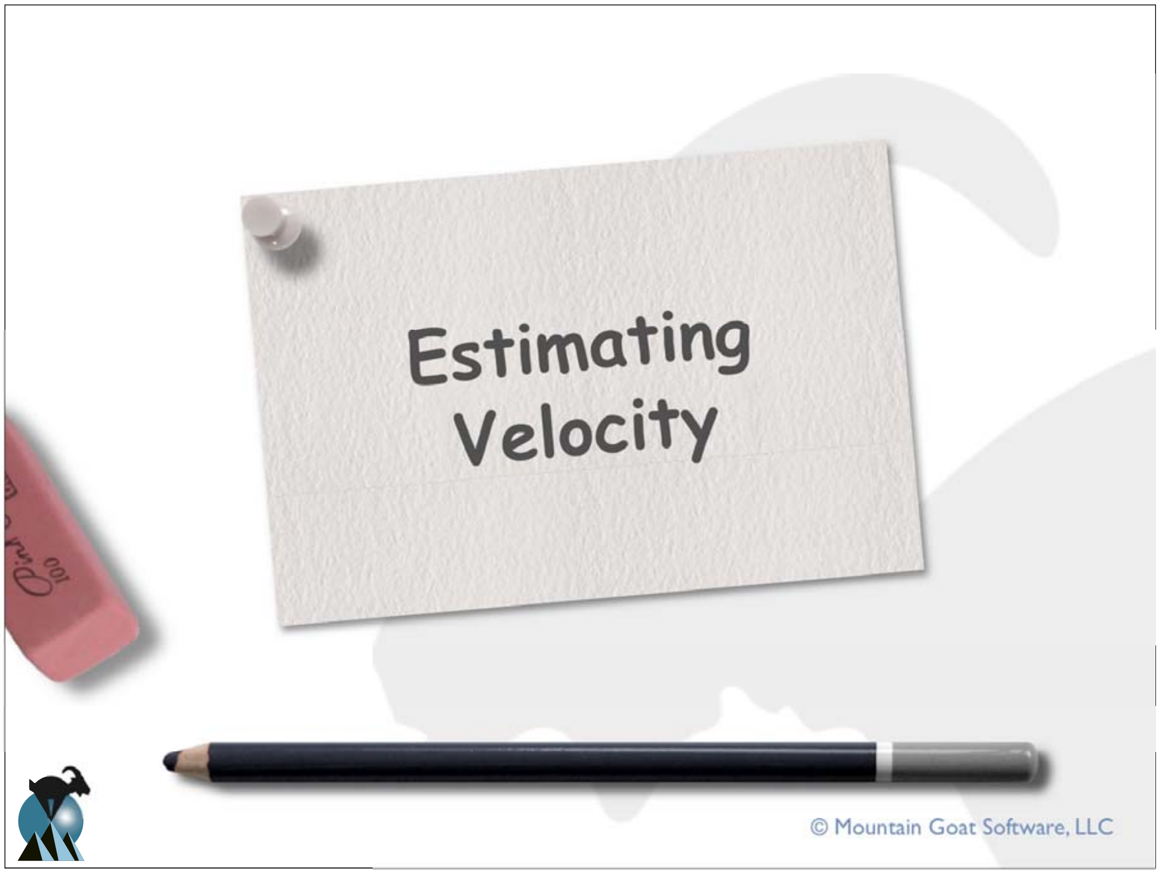
Team can commit, so they continue...

As a user, I want ...

3

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





How to estimate velocity

1. Use historical values
2. Don't, until you've run 1-3 sprints
3. Forecast it

Forecasting velocity

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

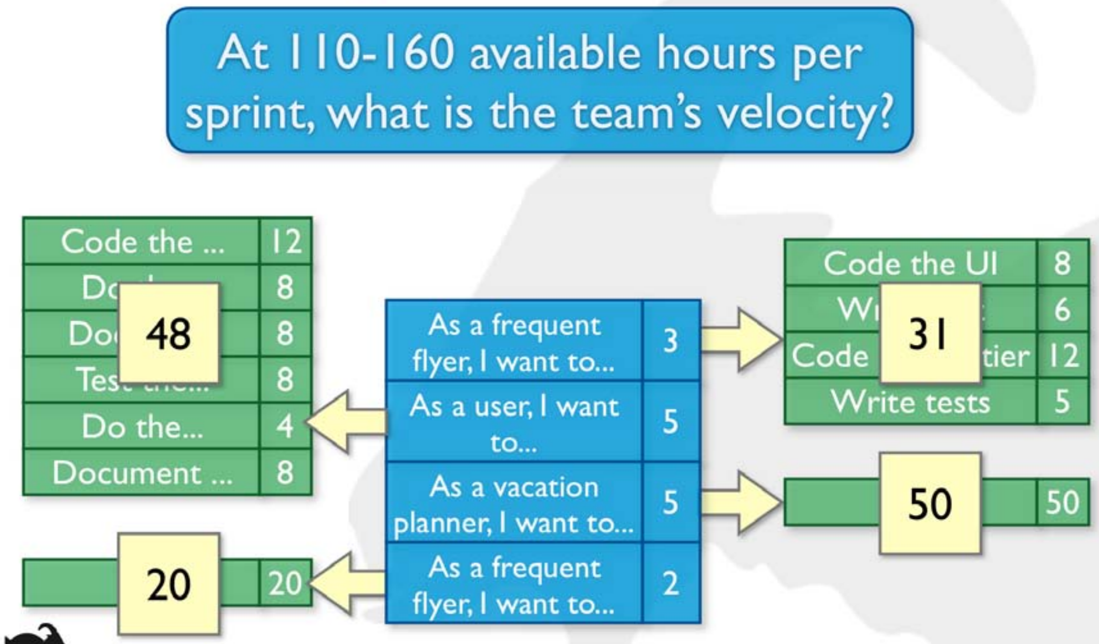
Ideally, "plan" more than one sprint

An example

- Estimating available hours

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

An example


At 110-160 available hours per sprint, what is the team's velocity?
Code the ... 12
Do... 8
Do... 8
Test the... 8
Do the... 4
Document ... 8
48
20
20
As a frequent flyer, I want to... 3
As a user, I want to... 5
As a vacation planner, I want to... 5
As a frequent flyer, I want to... 2
Code the UI 8
Write ... 6
Code ... tier 12
Write tests 5
31
50
50

© Mountain Goat Software, LLC

Put a range around it

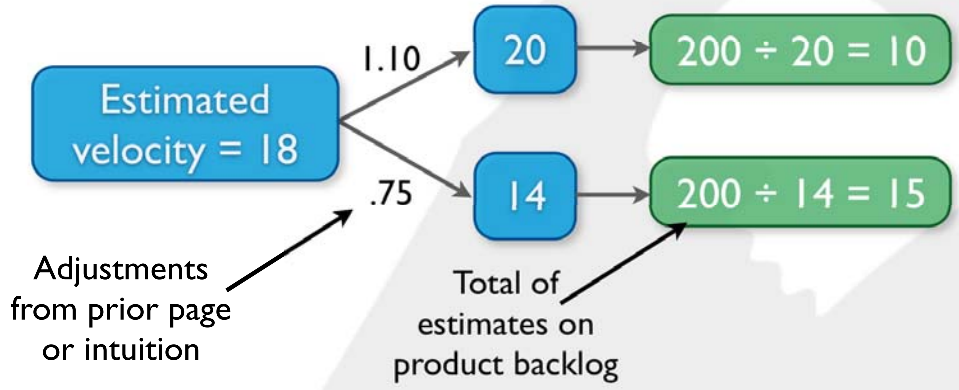
- You're unlikely to have precisely forecasted the exact velocity the team will average
- So, put a range around your velocity estimate:

Known team and known domain	+5% −10%
↕	+10% −25%
Unknown team or unknown domain	+25% −50%

†Numbers based on PMI advice on progressive accuracy of estimates.

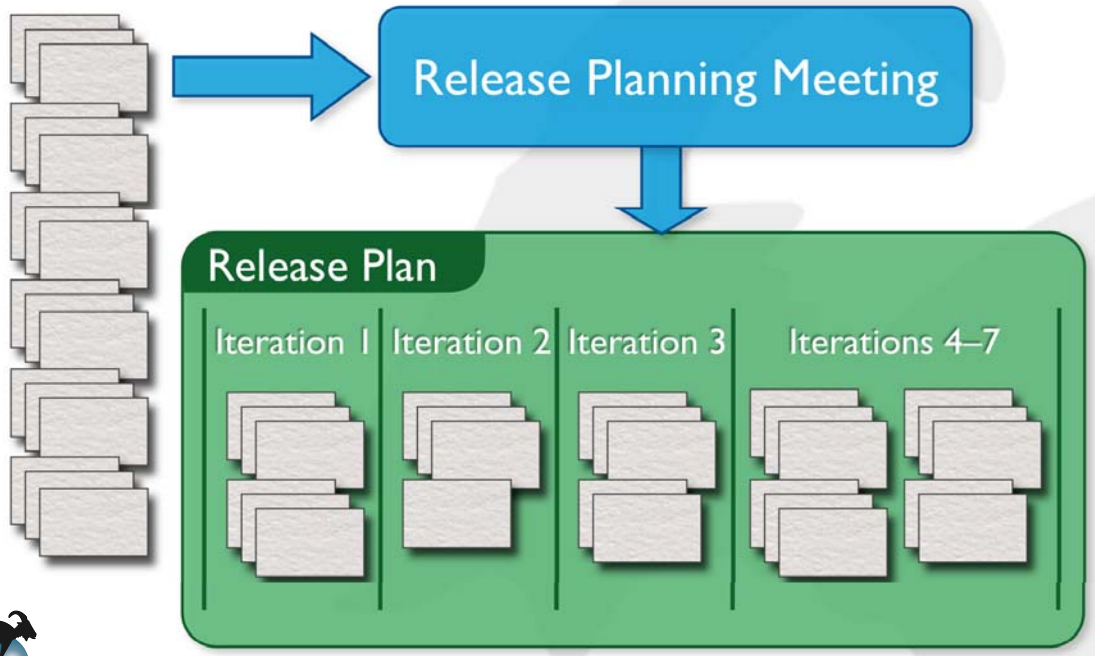
© Mountain Goat Software, LLC

Expressing velocity as a range


Estimated velocity = 18
1.10
.75
20
14
200 ÷ 20 = 10
200 ÷ 14 = 15
Adjustments from prior page or intuition
Total of estimates on product backlog

"Right now, before we start this project, our best estimate is that it will take between 10 and 15 sprints."

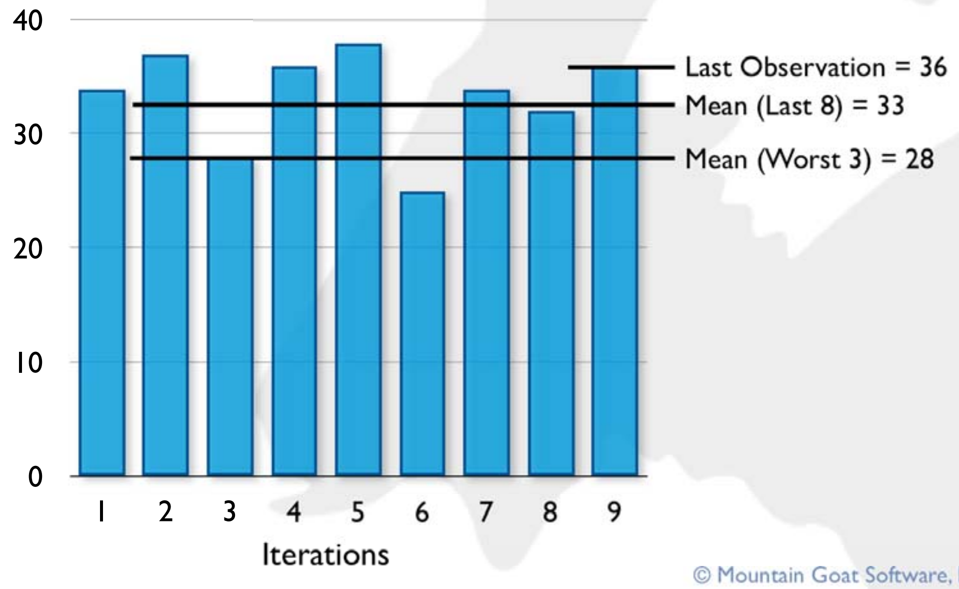
Release planning



Updating the release plan

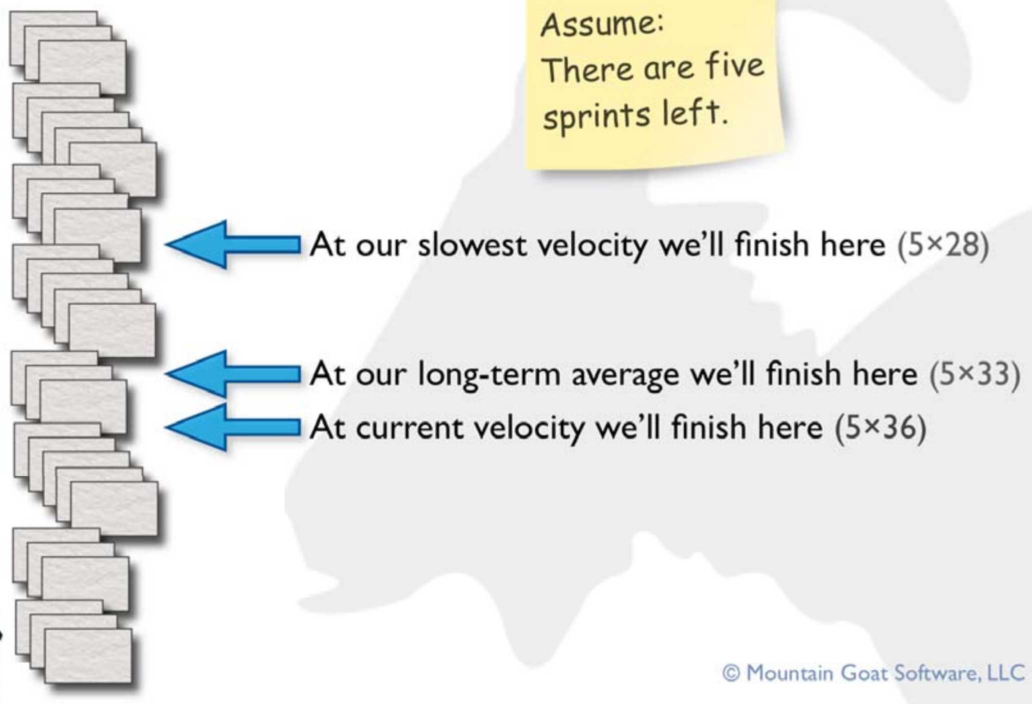
- Revisit the release plan at the end of every iteration
- Update it based on:
  - Current understanding of velocity
  - Current prioritization of the product backlog
- This should be a very short and sweet process

Track velocity multiple ways



Extrapolate from velocity

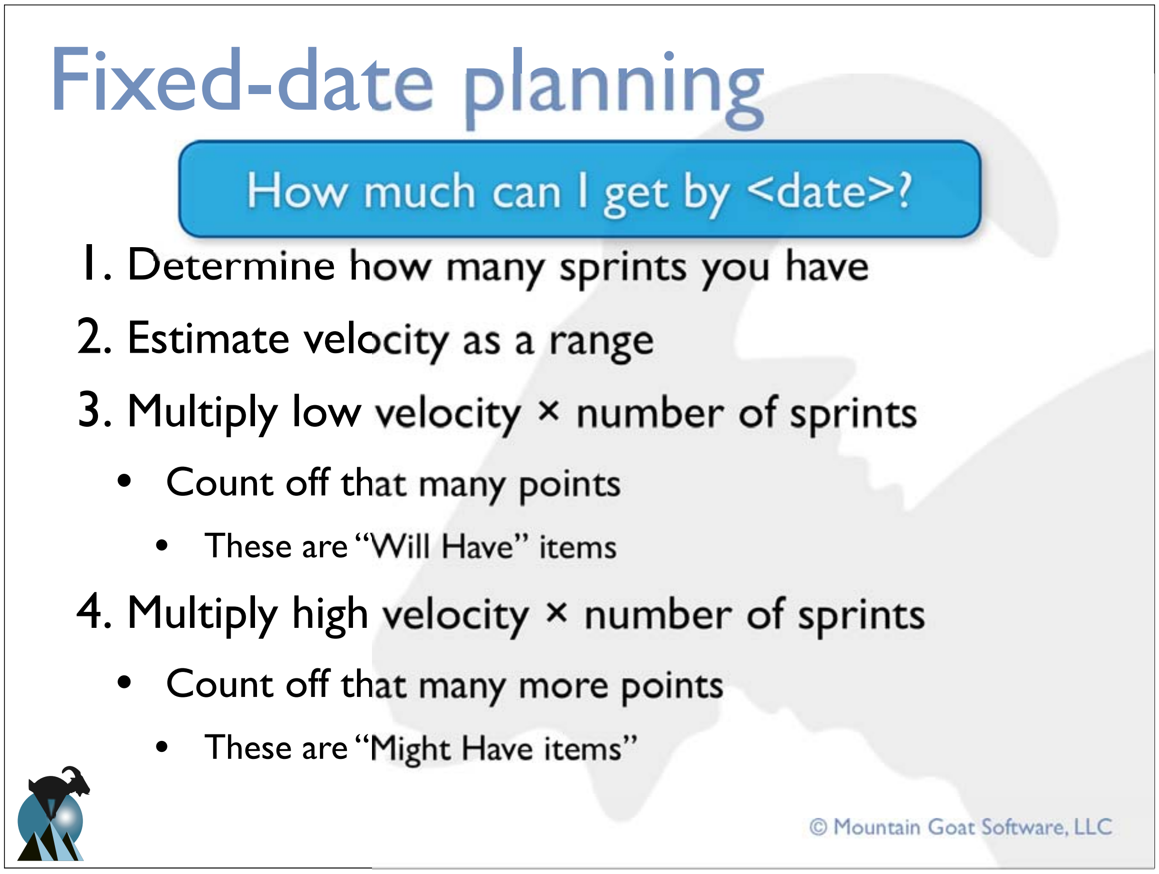
Assume:
There are five sprints left.



At our slowest velocity we'll finish here (5×28)

At our long-term average we'll finish here (5×33)

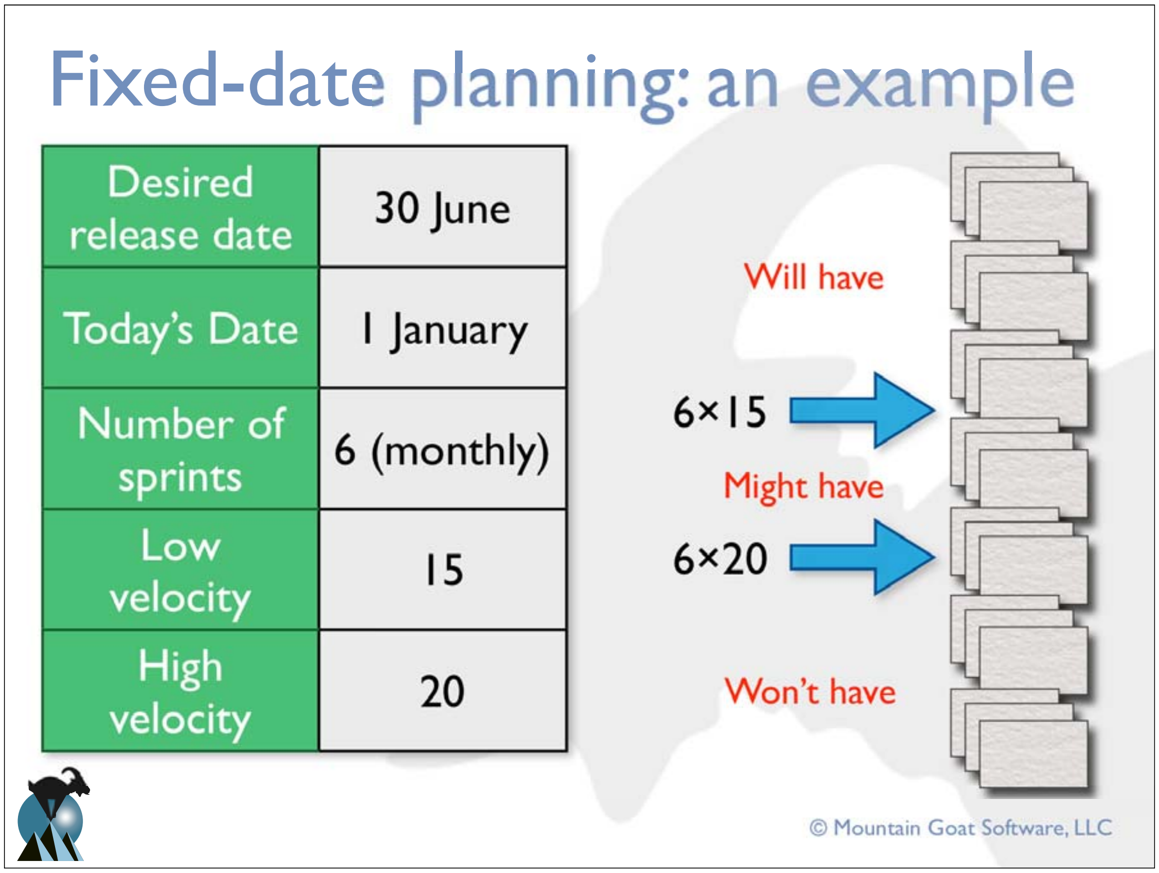
At current velocity we'll finish here (5×36)



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"



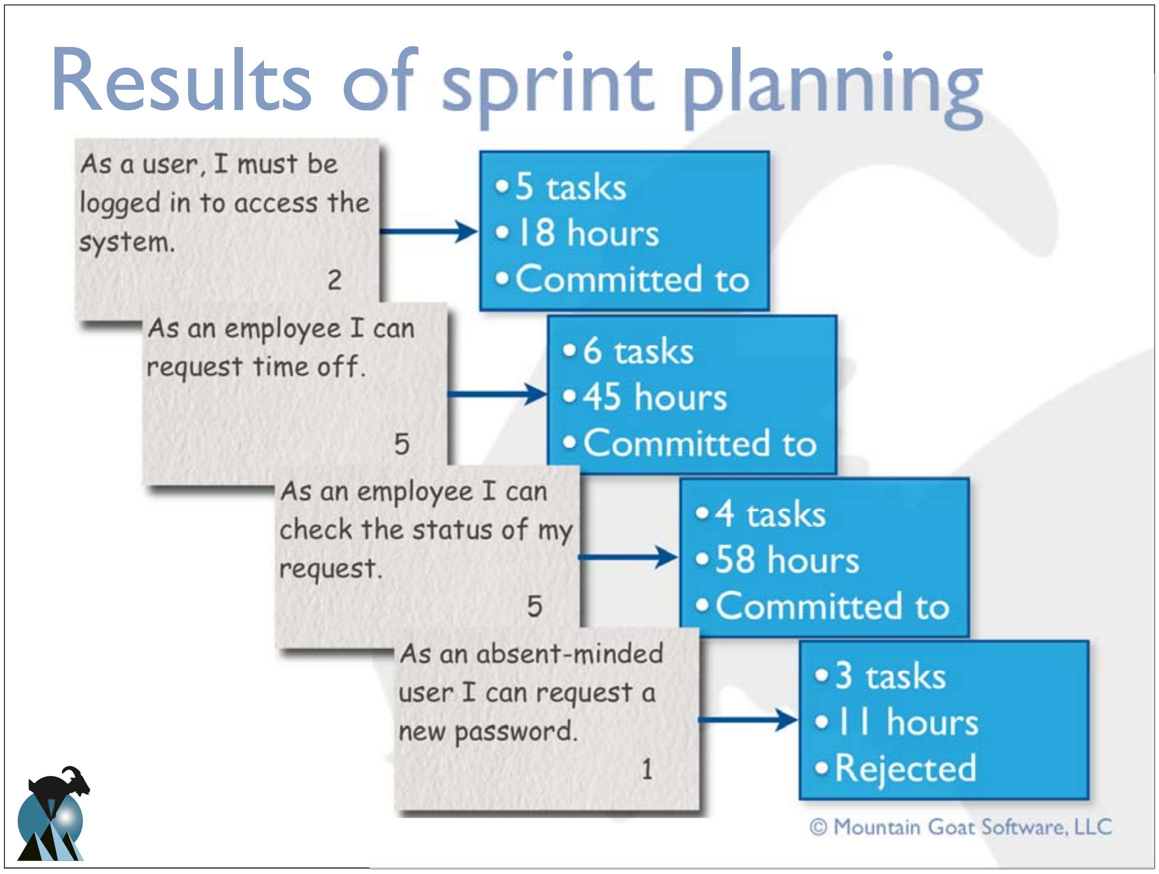
Fixed-date planning: an example

Desired release date	30 June
Today's Date	1 January
Number of sprints	6 (monthly)
Low velocity	15
High velocity	20

A fixed-date plan

1. Using the product backlog on the next page, determine which backlog items your team (Sergei, Yuri, and Carina) can commit to delivering in five iterations.
2. The product backlog has been prioritized by your product owner (me).
3. Members of this team have worked together in various combinations in the past but not recently so they don't have a historical velocity
4. They know the technologies well and the domain is familiar.
5. The team has already held their first iteration planning meeting. Results are on the next page.
6. Reminder: Make velocity a range!

Intentionally blank



Results of sprint planning

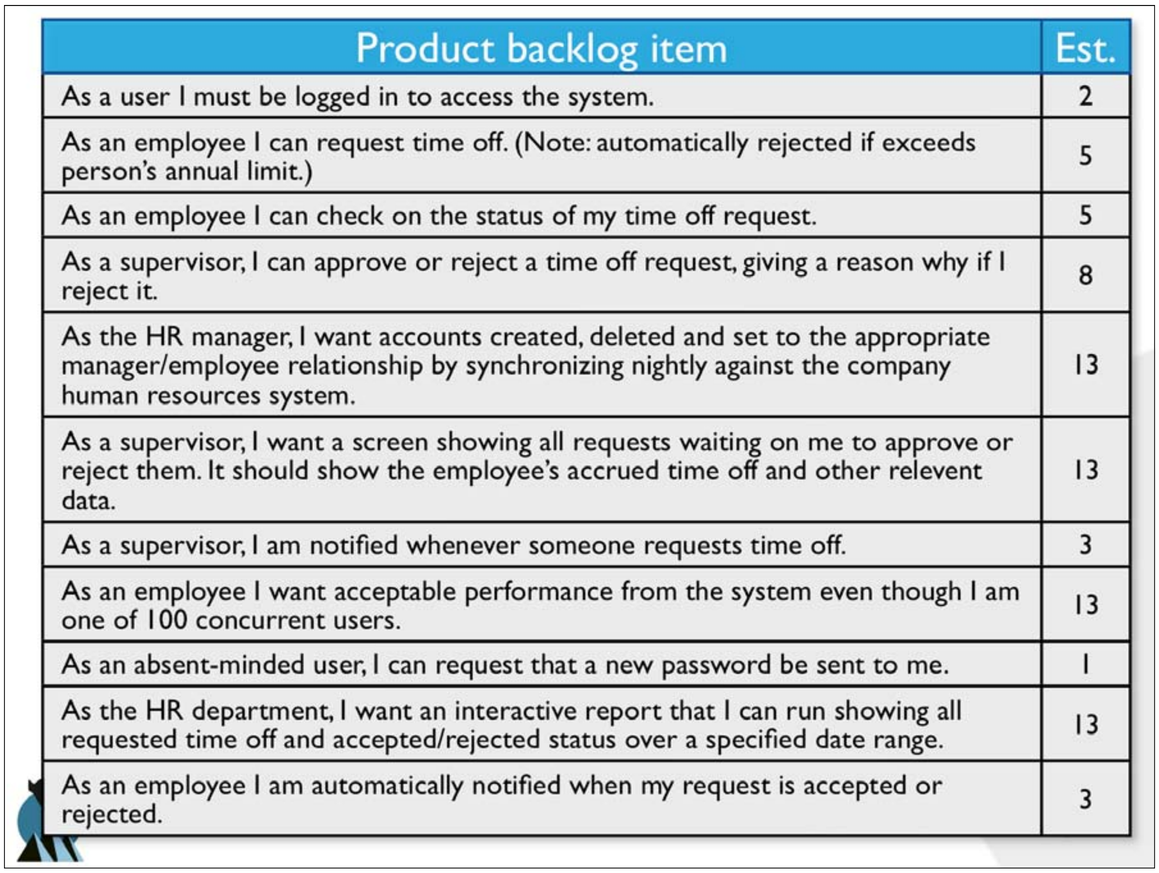
As a user, I must be logged in to access the system. 2
- 5 tasks
- 18 hours
- Committed to

As an employee I can request time off. 5
- 6 tasks
- 45 hours
- Committed to

As an employee I can check the status of my request. 5
- 4 tasks
- 58 hours
- Committed to

As an absent-minded user I can request a new password. 1
- 3 tasks
- 11 hours
- Rejected

© Mountain Goat Software, LLC



Product backlog item	Est.
As a user I must be logged in to access the system.	2
As an employee I can request time off. (Note: automatically rejected if exceeds person's annual limit.)	5
As an employee I can check on the status of my time off request.	5
As a supervisor, I can approve or reject a time off request, giving a reason why if I reject it.	8
As the HR manager, I want accounts created, deleted and set to the appropriate manager/employee relationship by synchronizing nightly against the company human resources system.	13
As a supervisor, I want a screen showing all requests waiting on me to approve or reject them. It should show the employee's accrued time off and other relevent data.	13
As a supervisor, I am notified whenever someone requests time off.	3
As an employee I want acceptable performance from the system even though I am one of 100 concurrent users.	13
As an absent-minded user, I can request that a new password be sent to me.	1
As the HR department, I want an interactive report that I can run showing all requested time off and accepted/rejected status over a specified date range.	13
As an employee I am automatically notified when my request is accepted or rejected.	3

Upcoming public classes

Date	What	Where
Jan 15-16 Jan 17	Certified ScrumMaster Agile Estimating and Planning	Atlanta
Feb 24-25 Feb 26	Certified ScrumMaster Agile Estimating and Planning	Seattle
April 8-9 April 10	Certified ScrumMaster Agile Estimating and Planning	Dallas
June 3-4 June 5	Certified ScrumMaster Agile Estimating and Planning	Washington, DC (Reston)
Other classes in London, Oslo and Stockholm if you're up for a longer trip.		

Information and registration at
www.mountaingoatsoftware.com

Mike Cohn contact info

The Addison-Wesley Signature Series
User Stories Applied
For Agile Software Development
Mike Cohn
Foreword by Kent Beck
Robert C. Martin Series
Agile Estimating and Planning
Mike Cohn

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)

MOUNTAIN GOAT
SOFTWARE