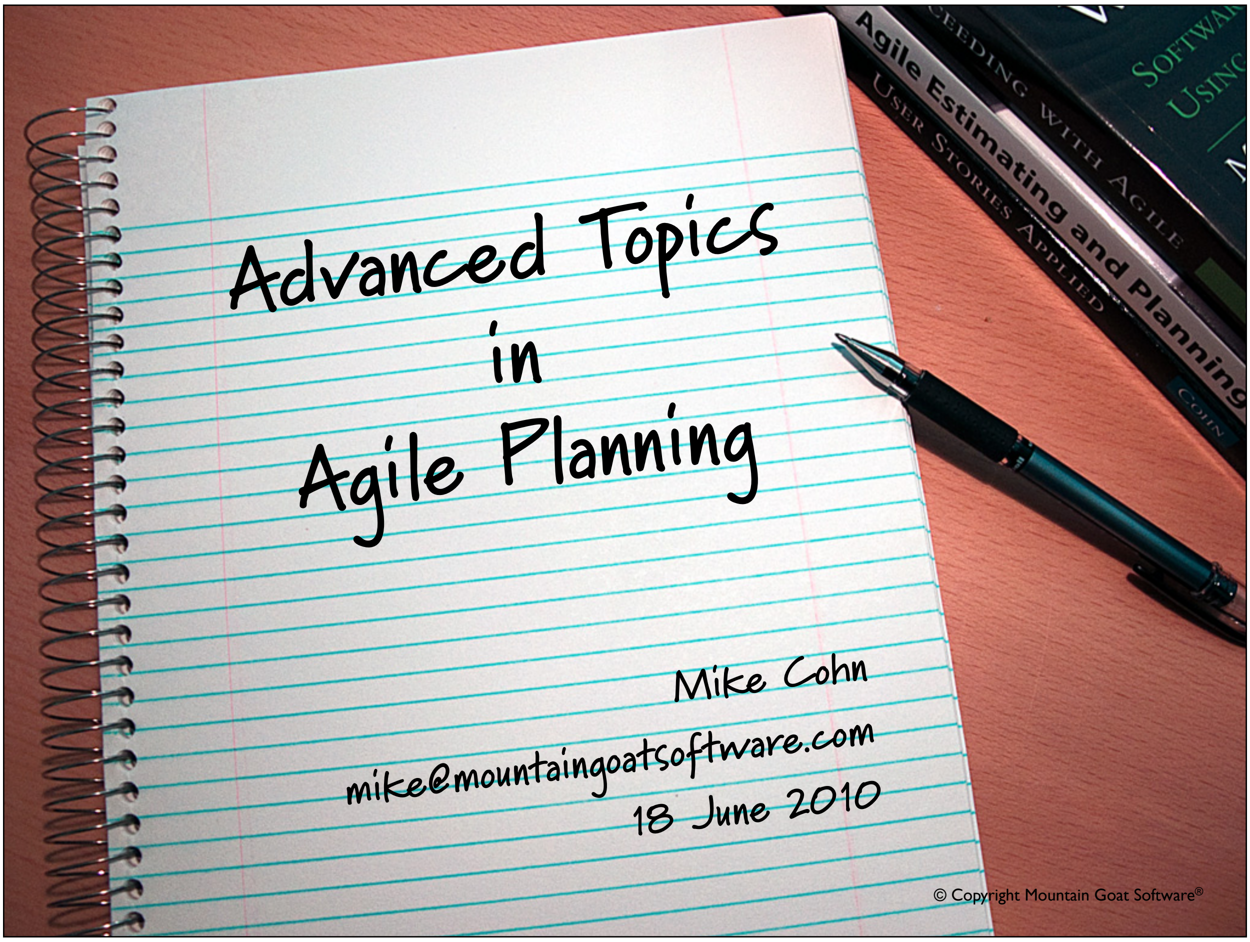

Agile Estimating and Planning
User Stories Applied
Succeeding with Agile
Software Development Using Scrum
Cohn

Advanced Topics in Agile Planning

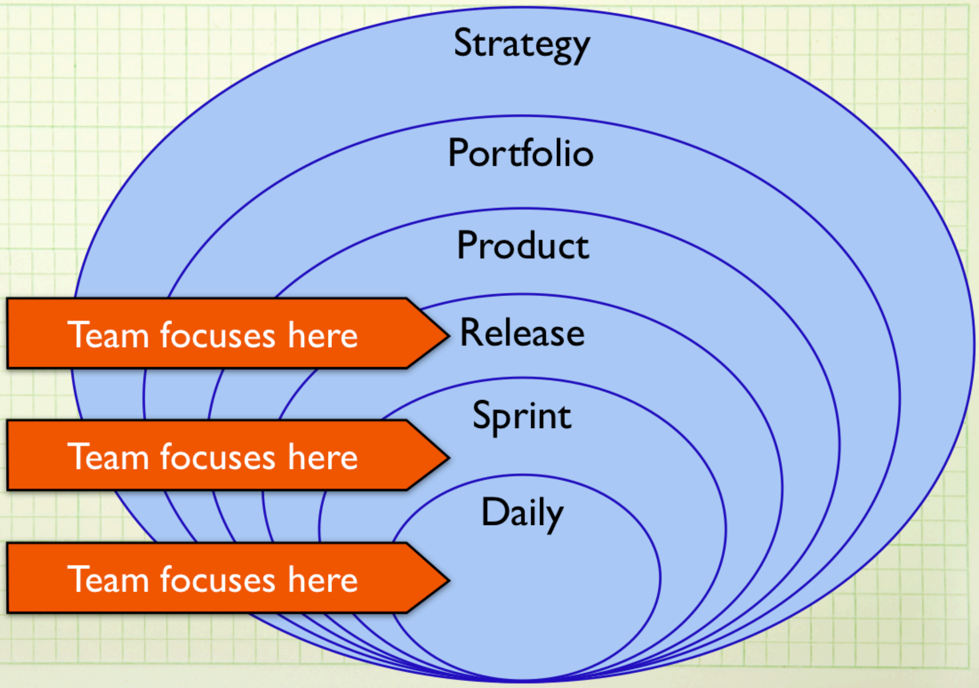
Mike Cohn
mike@mountaingoatsoftware.com
18 June 2010

Mike Cohn

- Founding member and director of Agile Alliance and Scrum Alliance
- Founder of Mountain Goat Software
- Doing Scrum since 1995
- Started my career as a programmer; worked as VP Engineering in 4 companies

Agile Estimating and Planning
User Stories Applied
For Agile Software Development
Mike Cohn
Succeeding with Agile
Software Development Using Scrum
Mike Cohn

The planning onion



Strategy

Portfolio

Product

Release

Sprint

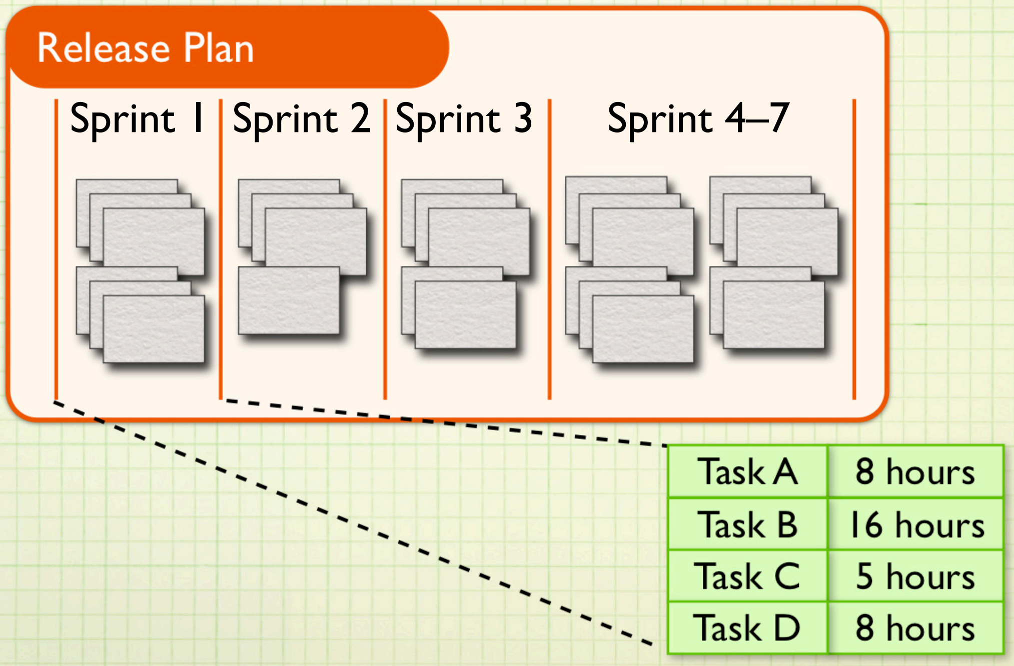
Daily

Team focuses here

Team focuses here

Team focuses here

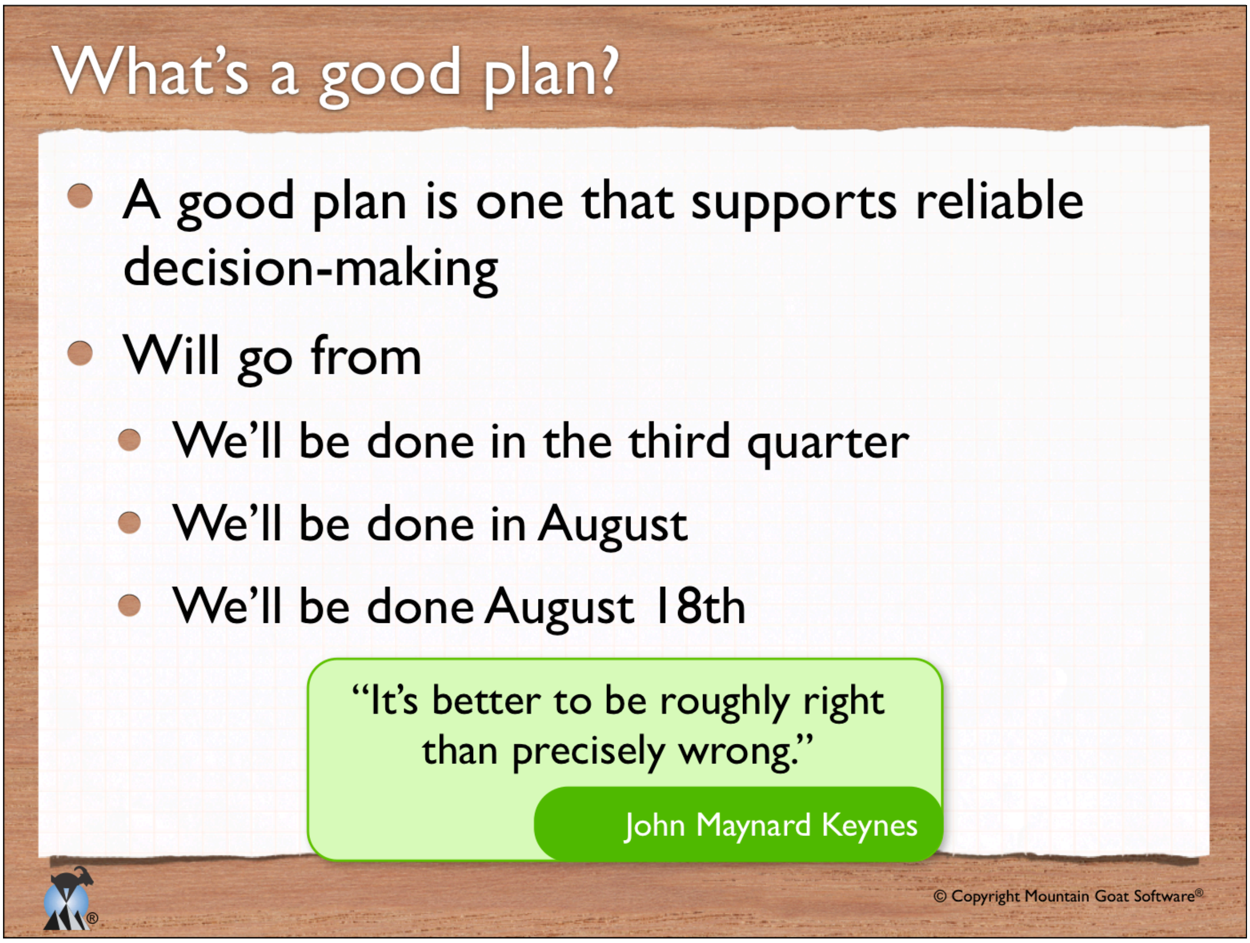
Release and sprint planning



Release Plan

Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4–7

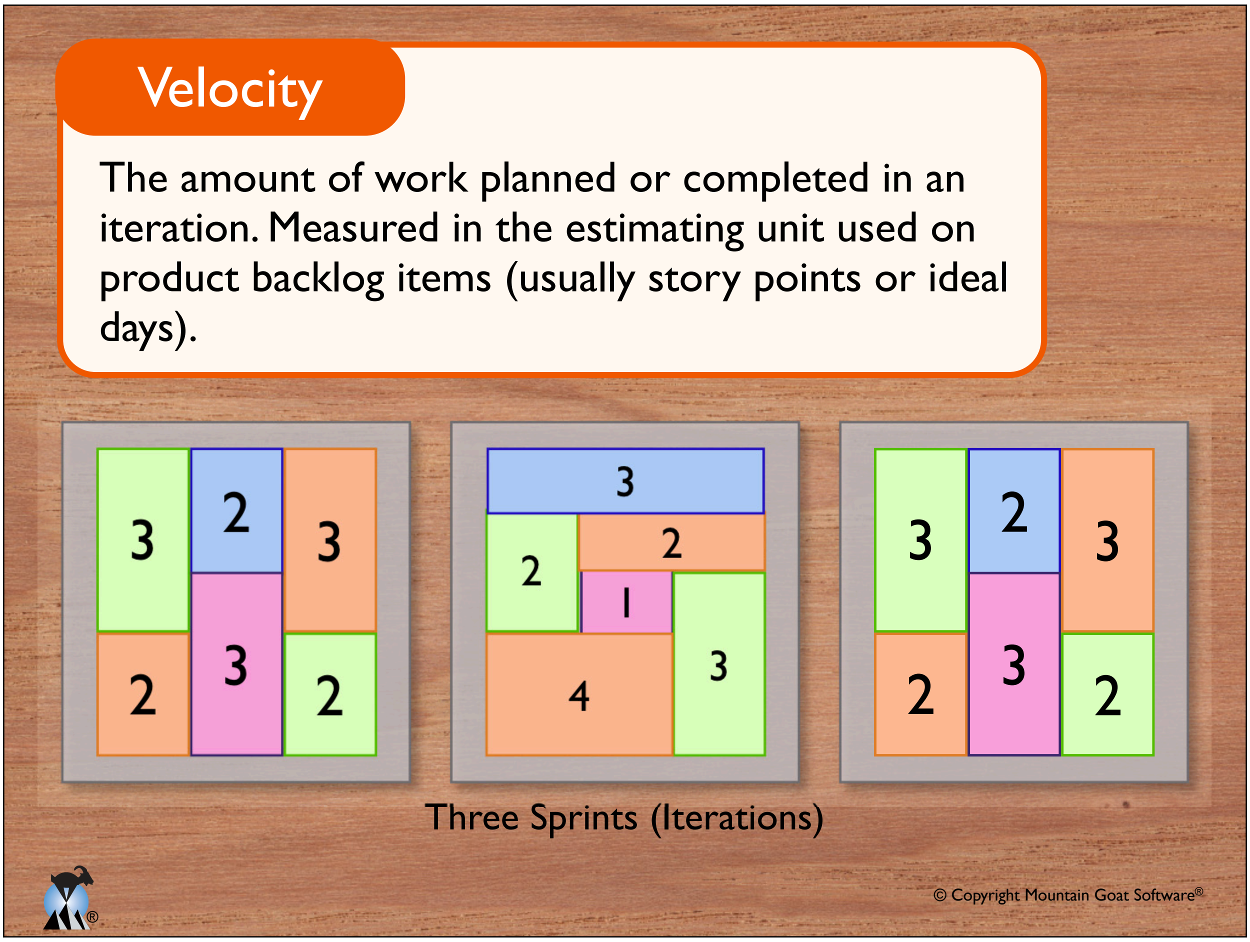
Task A	8 hours
Task B	16 hours
Task C	5 hours
Task D	8 hours



What's a good plan?

- A good plan is one that supports reliable decision-making
- Will go from
  - We'll be done in the third quarter
  - We'll be done in August
  - We'll be done August 18th

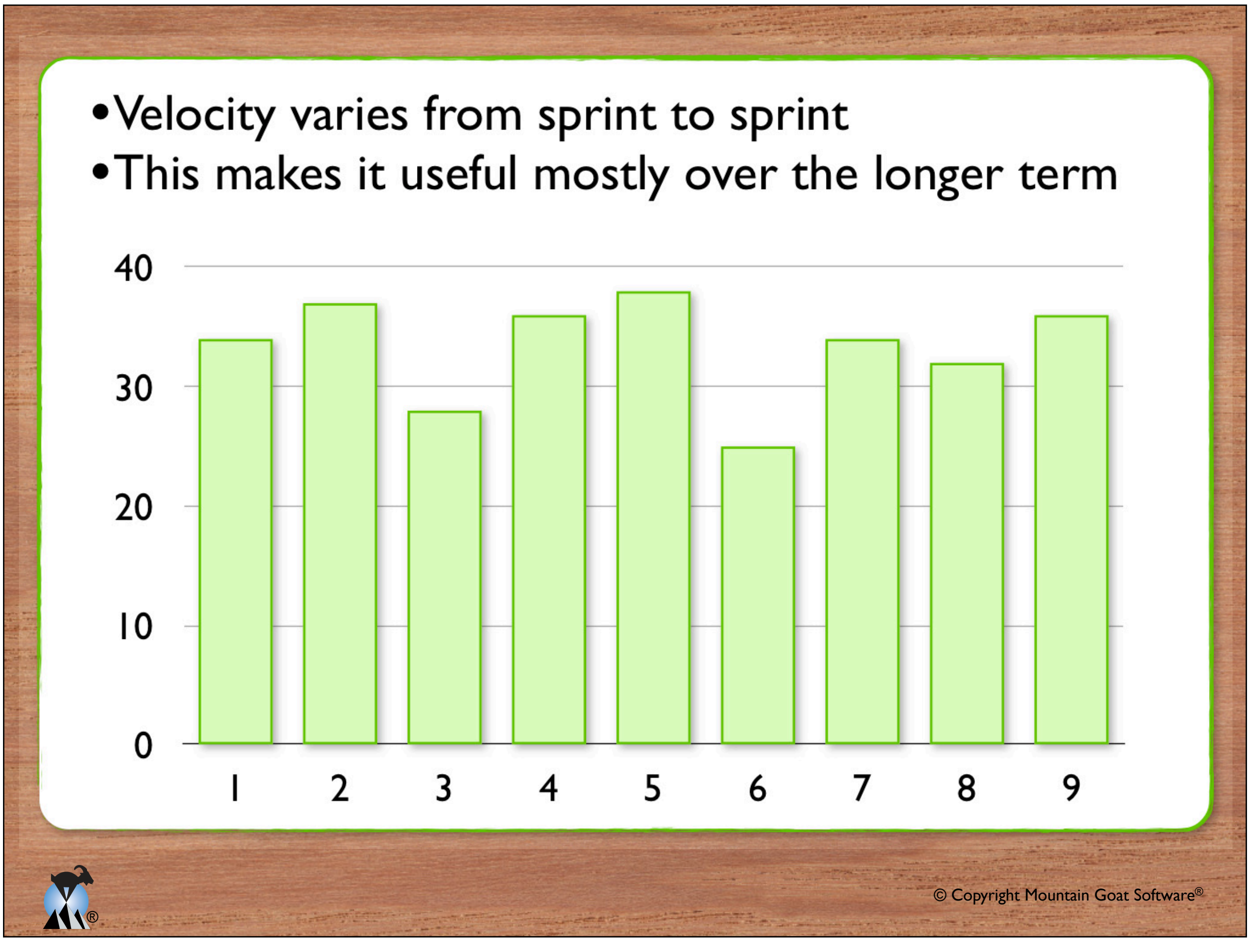
> "It's better to be roughly right than precisely wrong."
>
> John Maynard Keynes



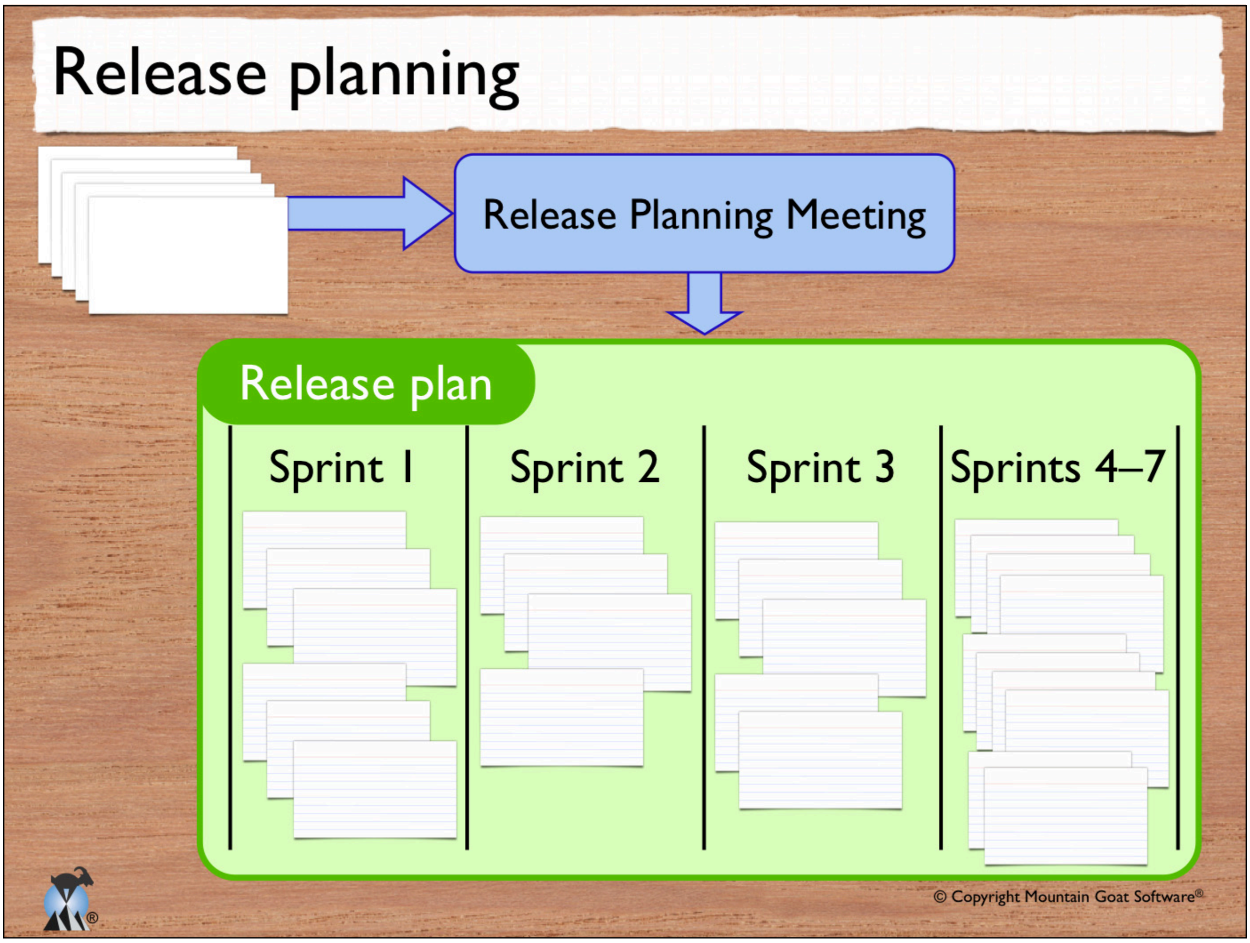
Velocity

The amount of work planned or completed in an iteration. Measured in the estimating unit used on product backlog items (usually story points or ideal days).

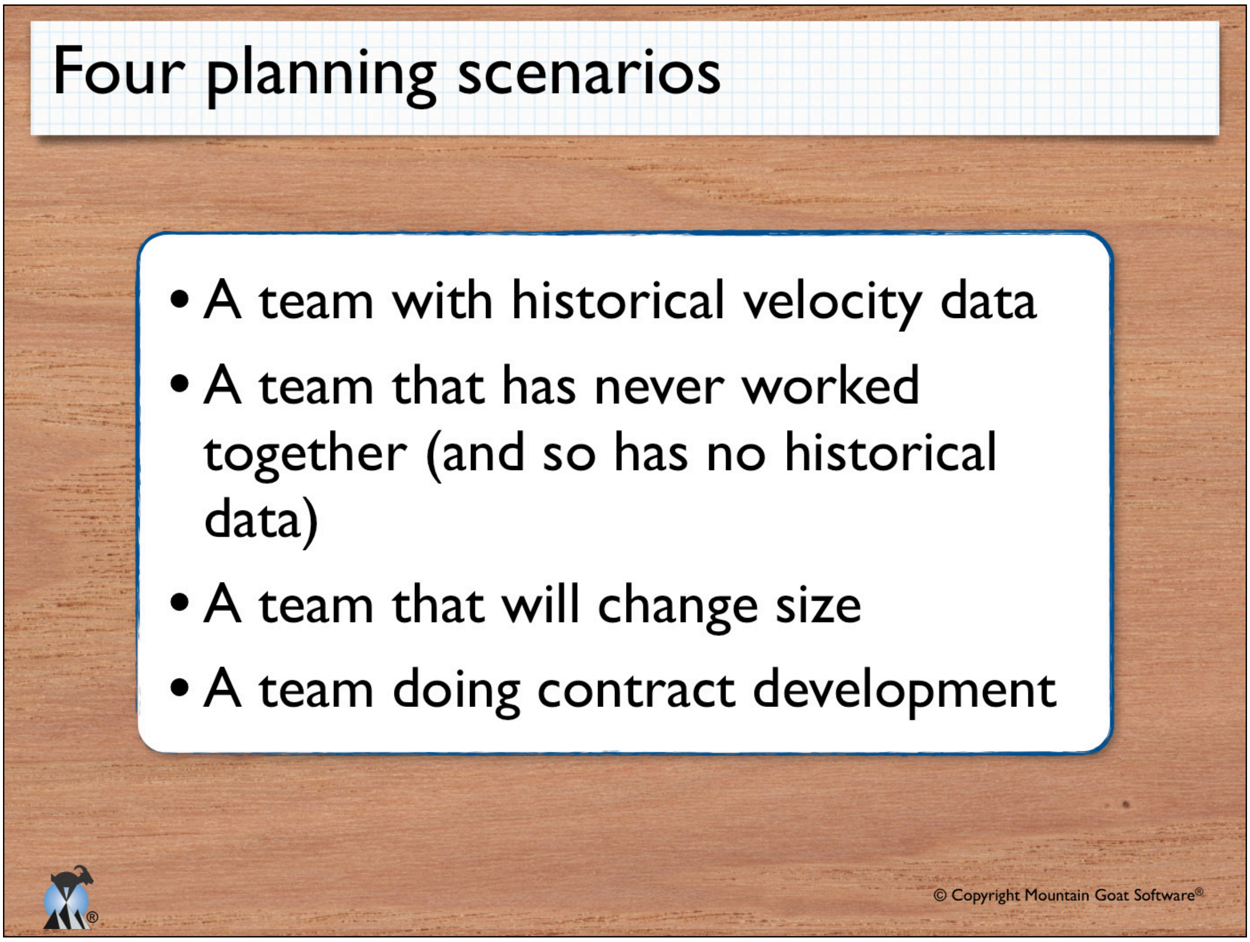
Three Sprints (Iterations)



- Velocity varies from sprint to sprint
- This makes it useful mostly over the longer term

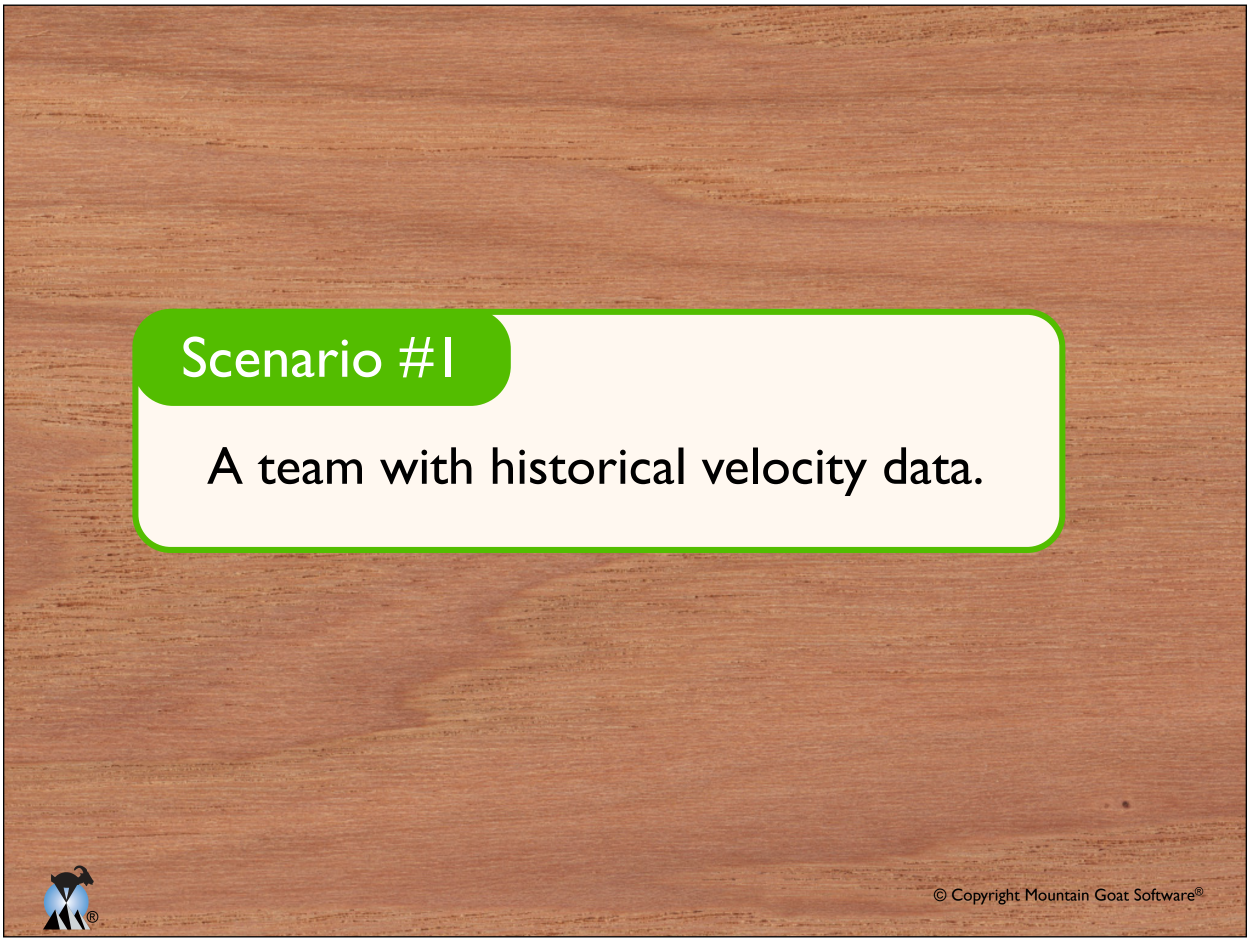


Release planning



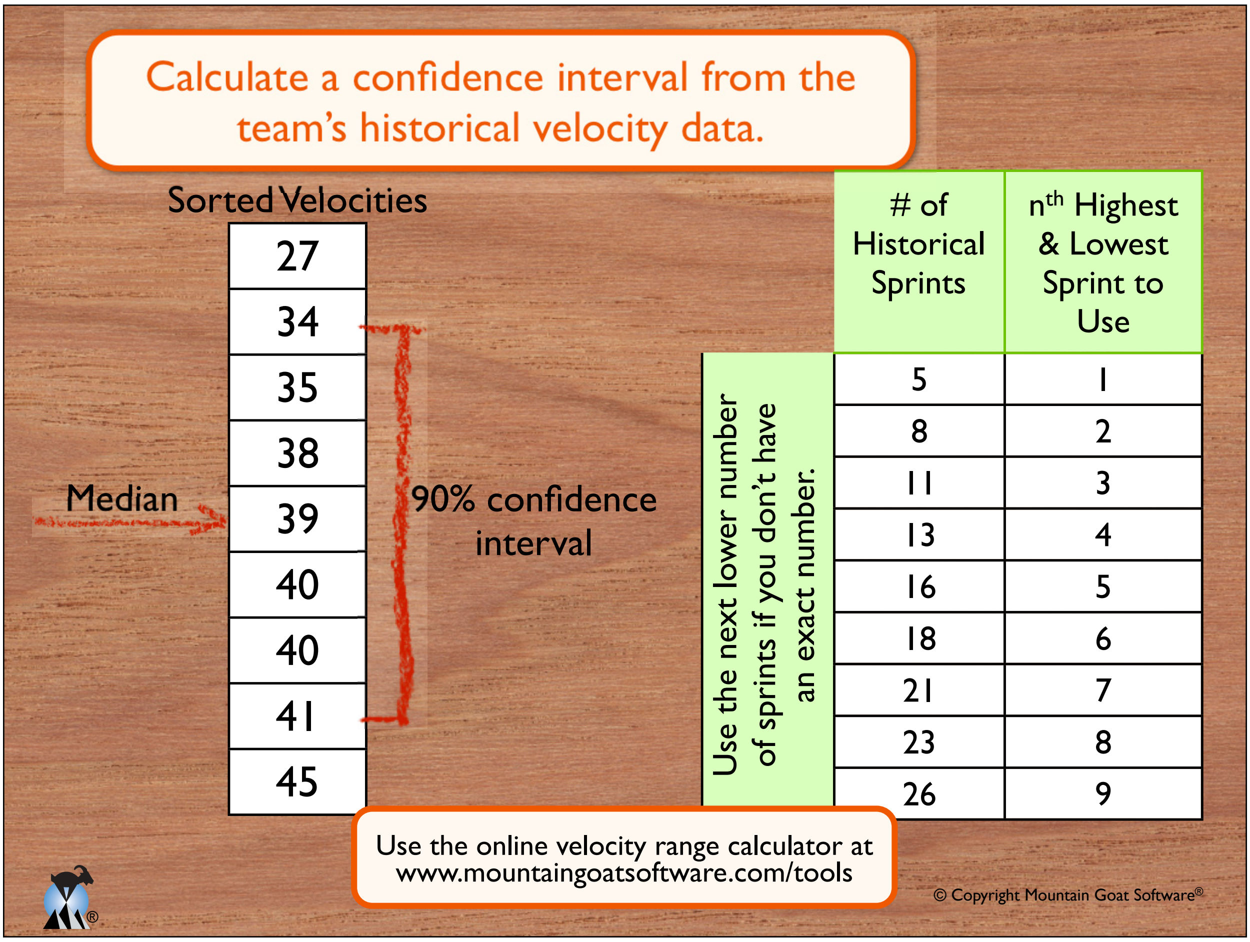
Four planning scenarios

- A team with historical velocity data
- A team that has never worked together (and so has no historical data)
- A team that will change size
- A team doing contract development



Scenario #1

A team with historical velocity data.



Calculate a confidence interval from the team's historical velocity data.

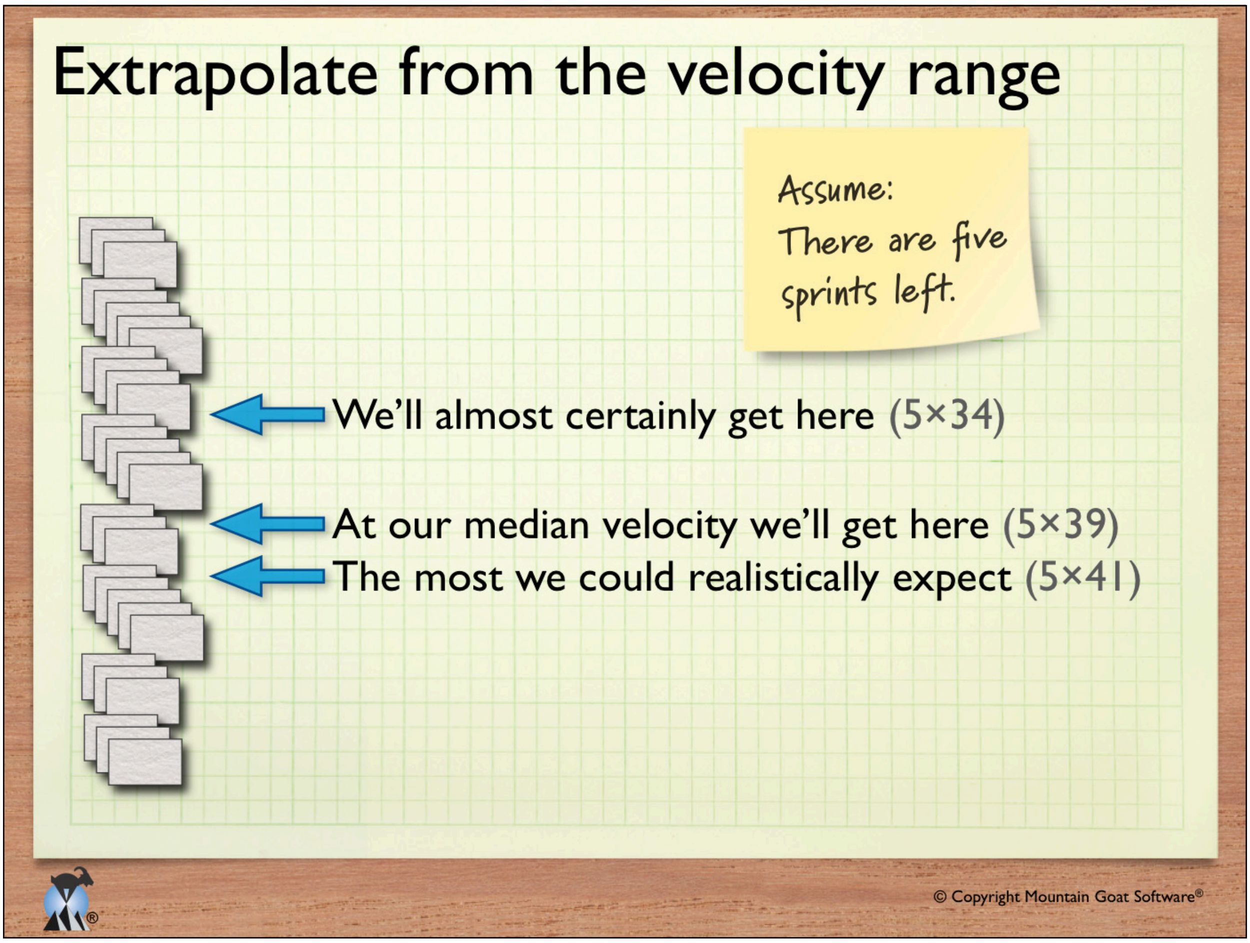
Sorted Velocities
27
34
35
38
39
40
40
41
45

Median
90% confidence interval

# of Historical Sprints	n^{th} Highest & Lowest Sprint to Use
5	1
8	2
11	3
13	4
16	5
18	6
21	7
23	8
26	9

Use the next lower number of sprints if you don't have an exact number.

Use the online velocity range calculator at www.mountaingoatsoftware.com/tools



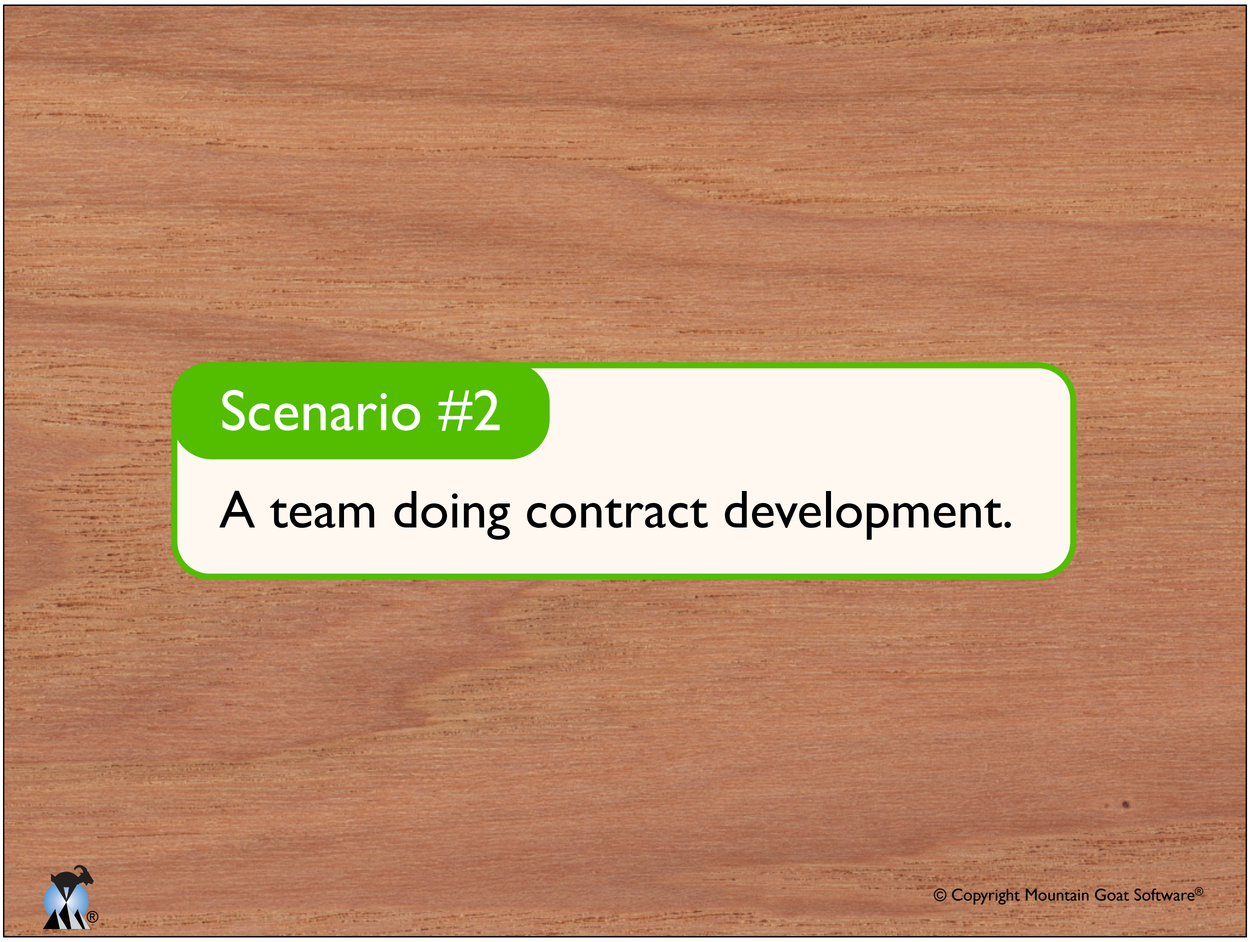
Extrapolate from the velocity range

Assume:
There are five sprints left.

We'll almost certainly get here (5×34)

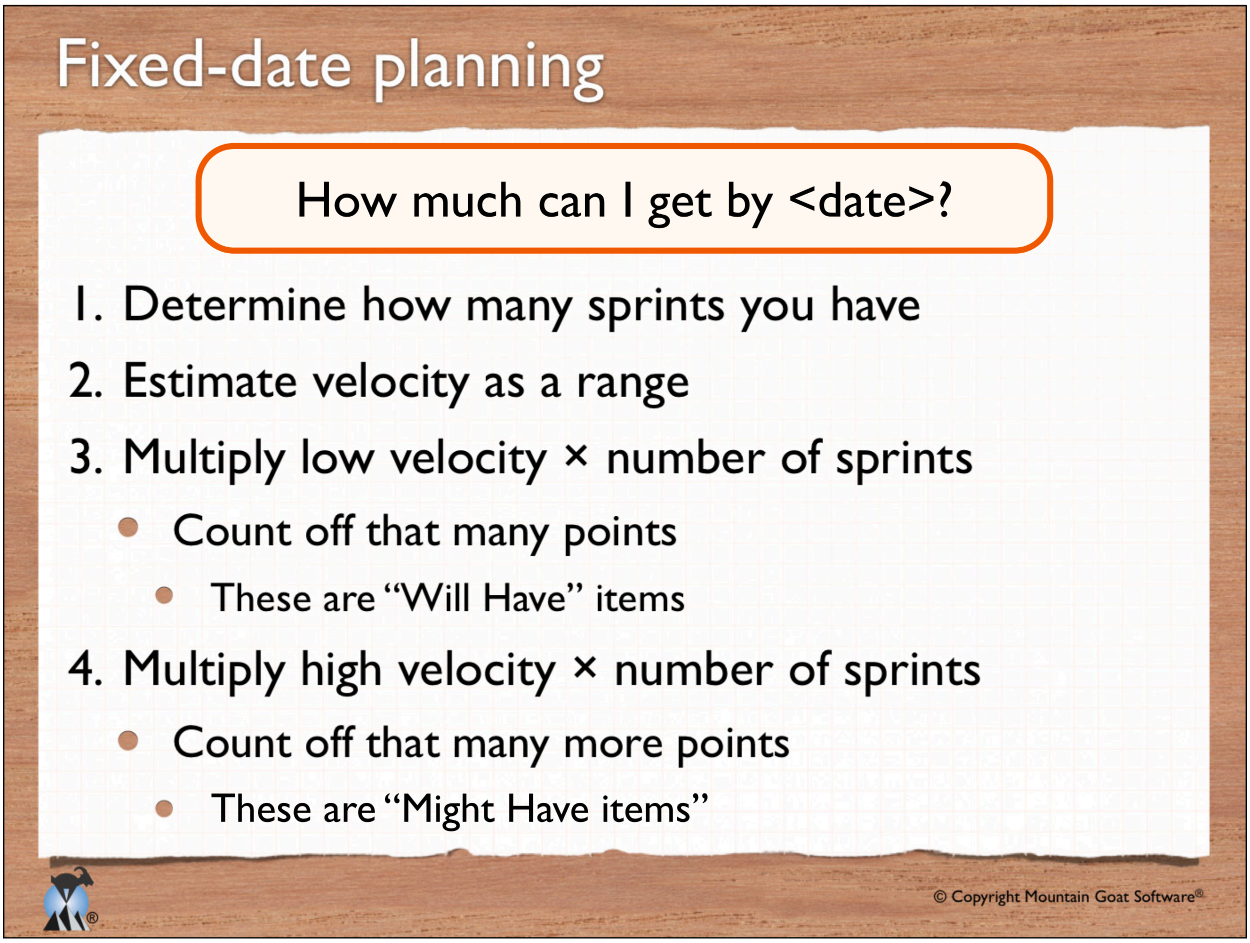
At our median velocity we'll get here (5×39)

The most we could realistically expect (5×41)



Scenario #2

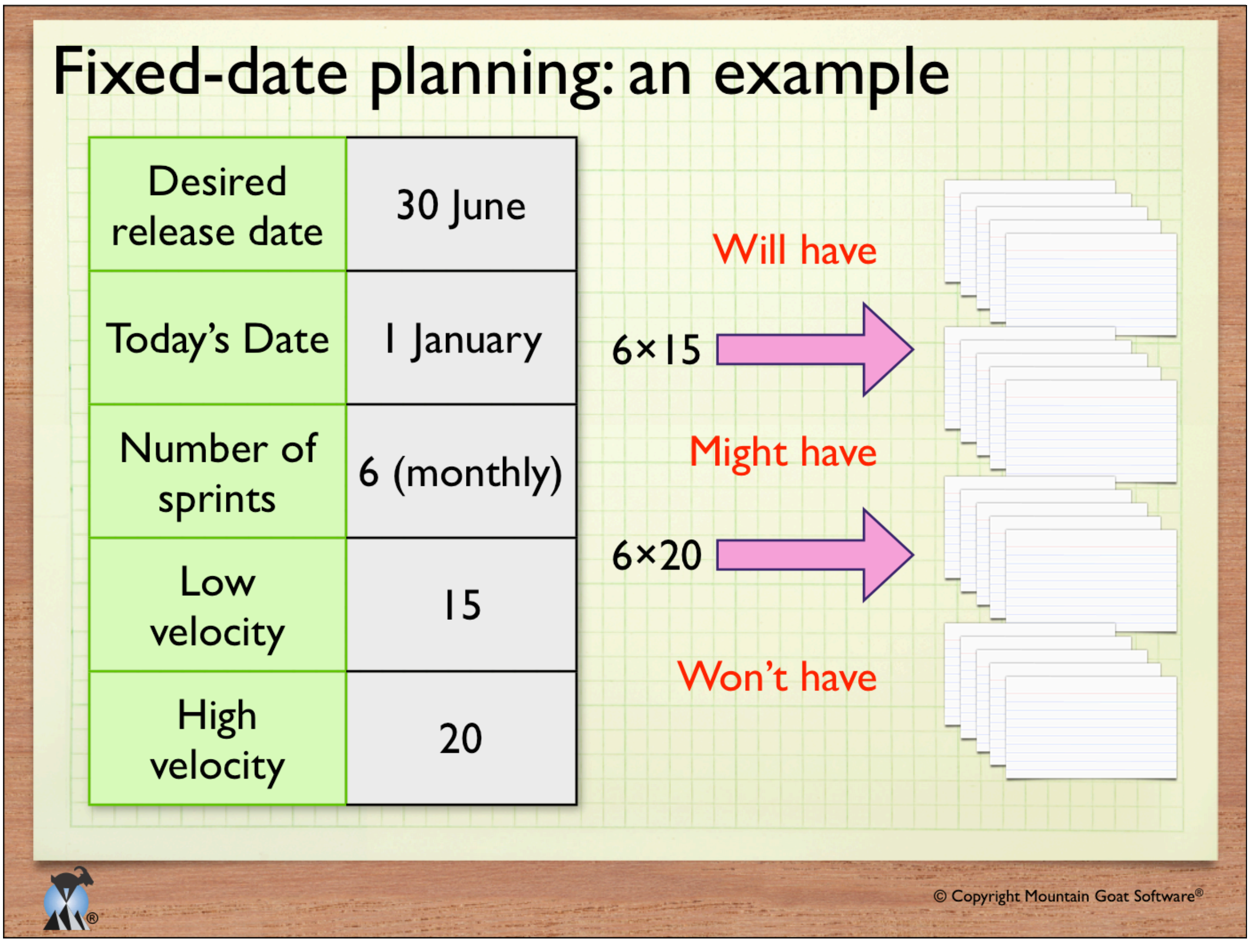
A team doing contract development.



Fixed-date planning

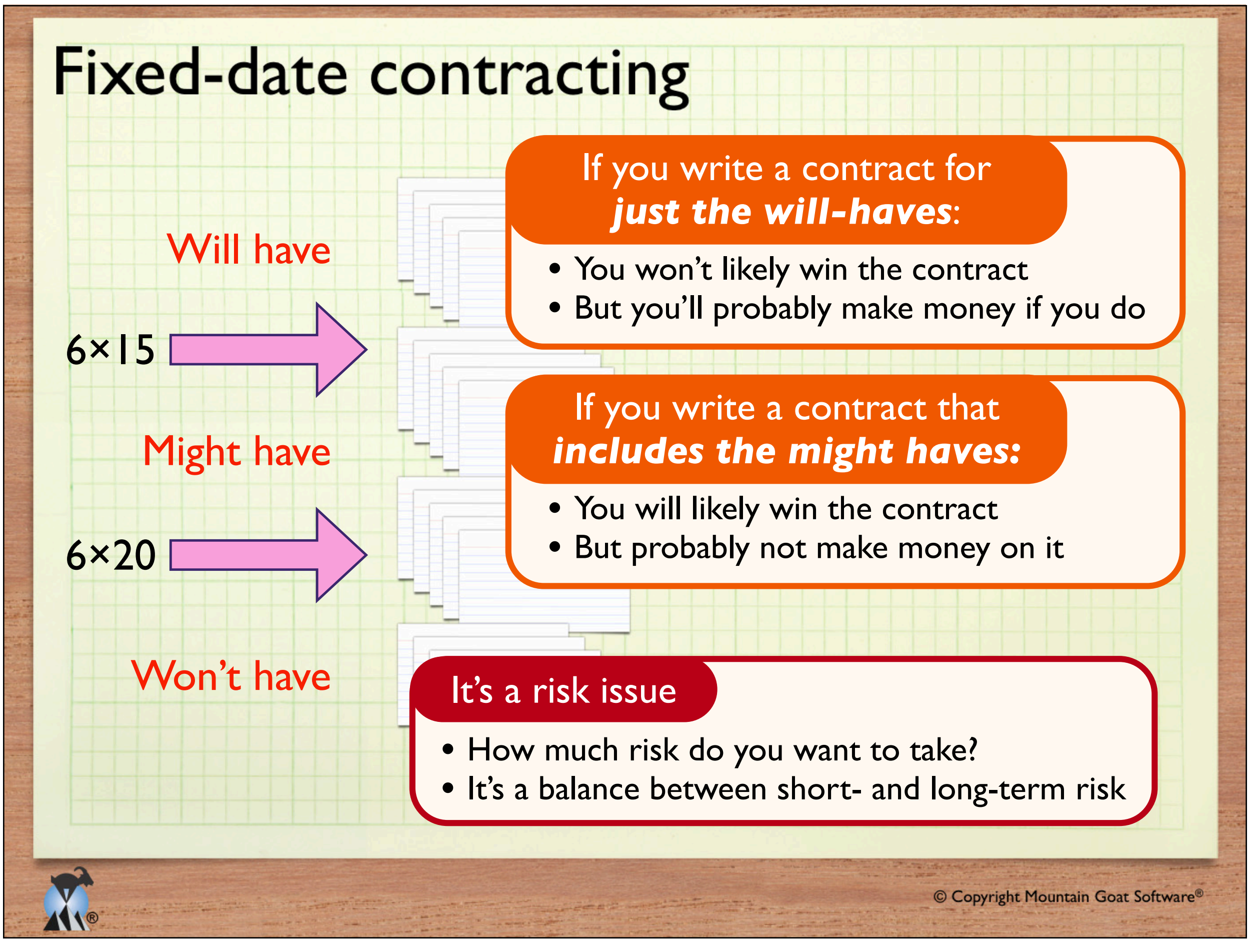
How much can I get by <date>?

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



Fixed-date contracting

If you write a contract for ***just the will-haves***:

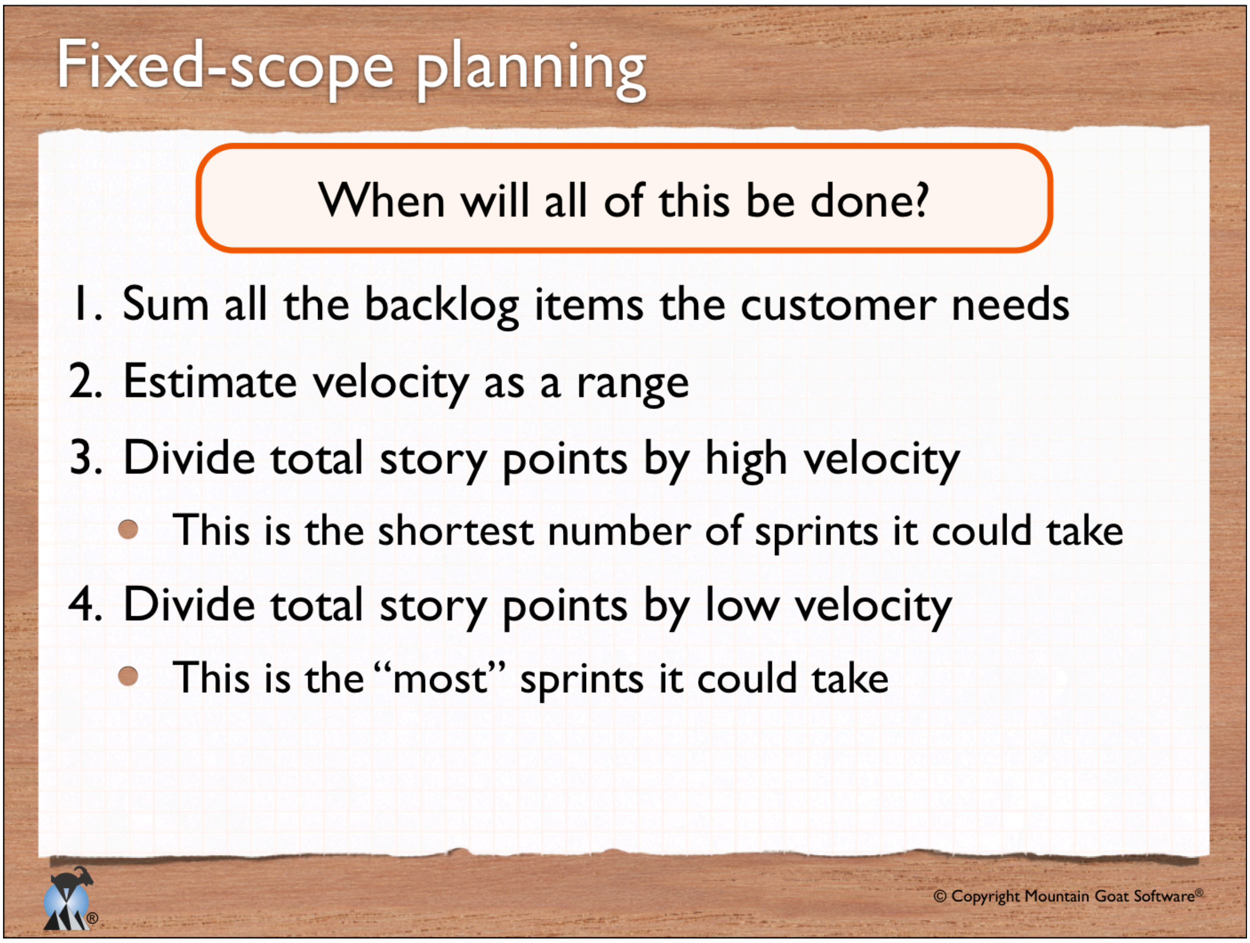
- You won't likely win the contract
- But you'll probably make money if you do

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

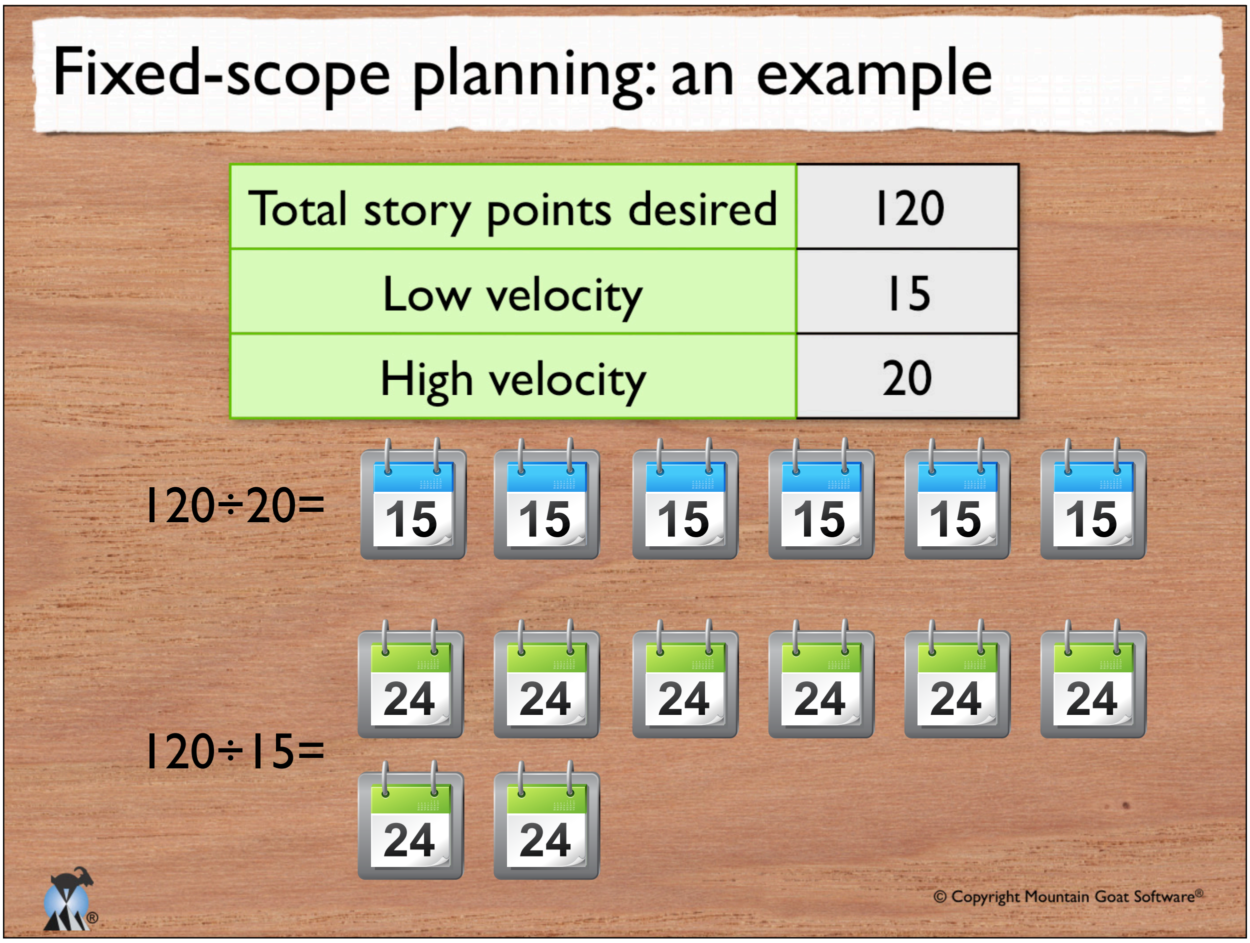
- How much risk do you want to take?
- It's a balance between short- and long-term risk



Fixed-scope planning

When will all of this be done?

1. Sum all the backlog items the customer needs
2. Estimate velocity as a range
3. Divide total story points by high velocity
   - This is the shortest number of sprints it could take
4. Divide total story points by low velocity
   - This is the "most" sprints it could take

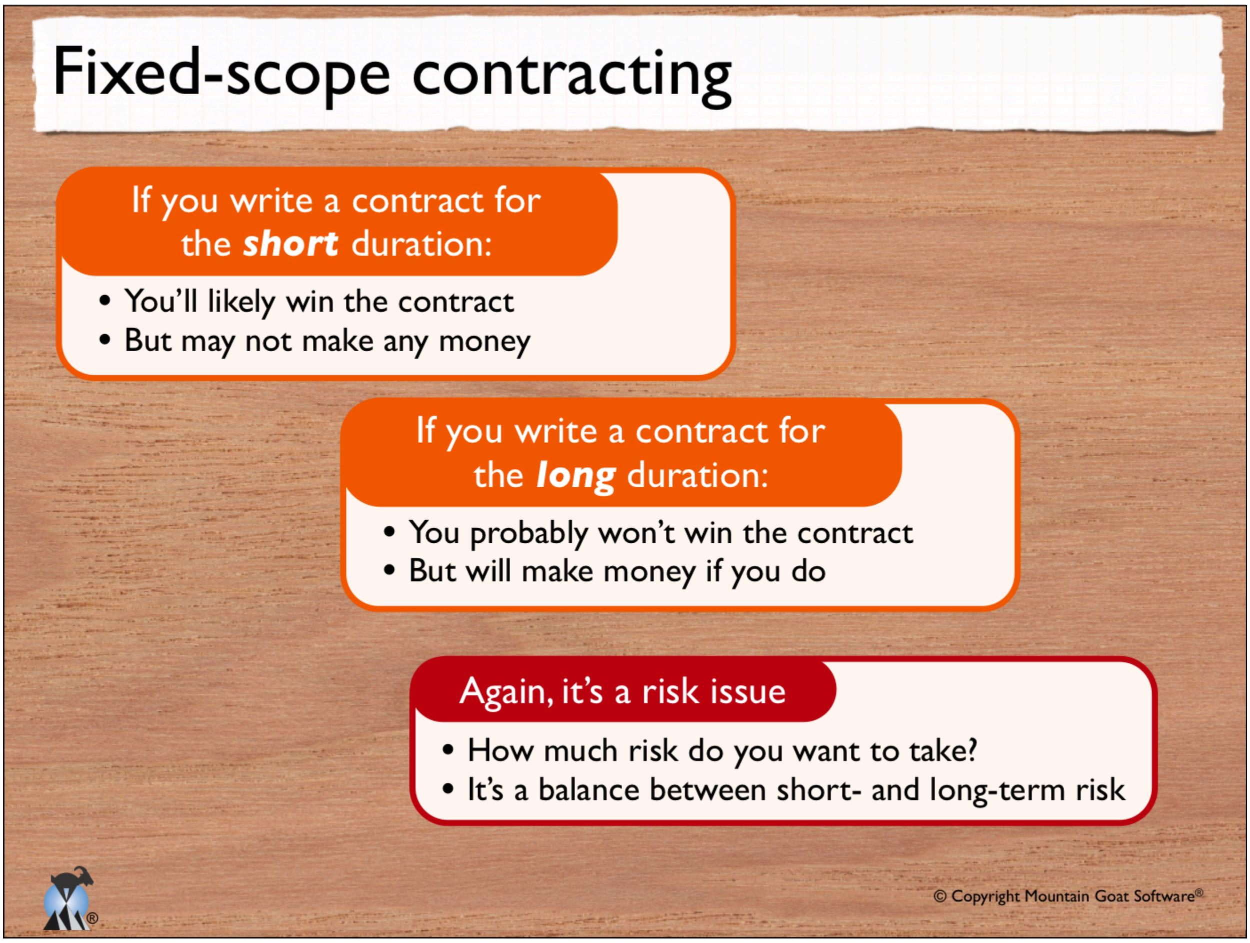


Fixed-scope planning: an example

Total story points desired	120
Low velocity	15
High velocity	20

120÷20= 15 15 15 15 15 15

120÷15= 24 24 24 24 24 24 24 24



Fixed-scope contracting

If you write a contract for the ***short*** duration:

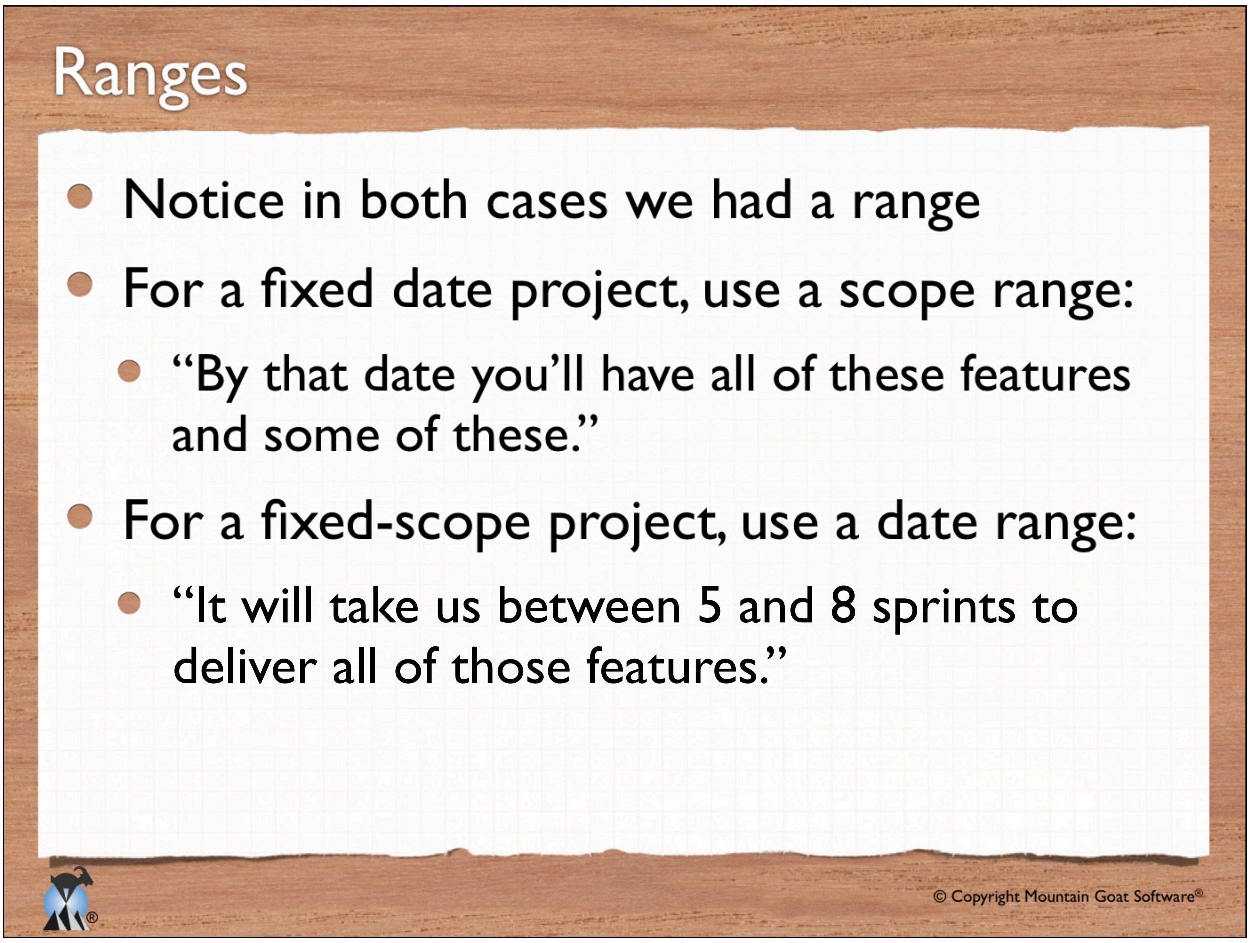
- You'll likely win the contract
- But may not make any money

If you write a contract for the ***long*** duration:

- You probably won't win the contract
- But will make money if you do

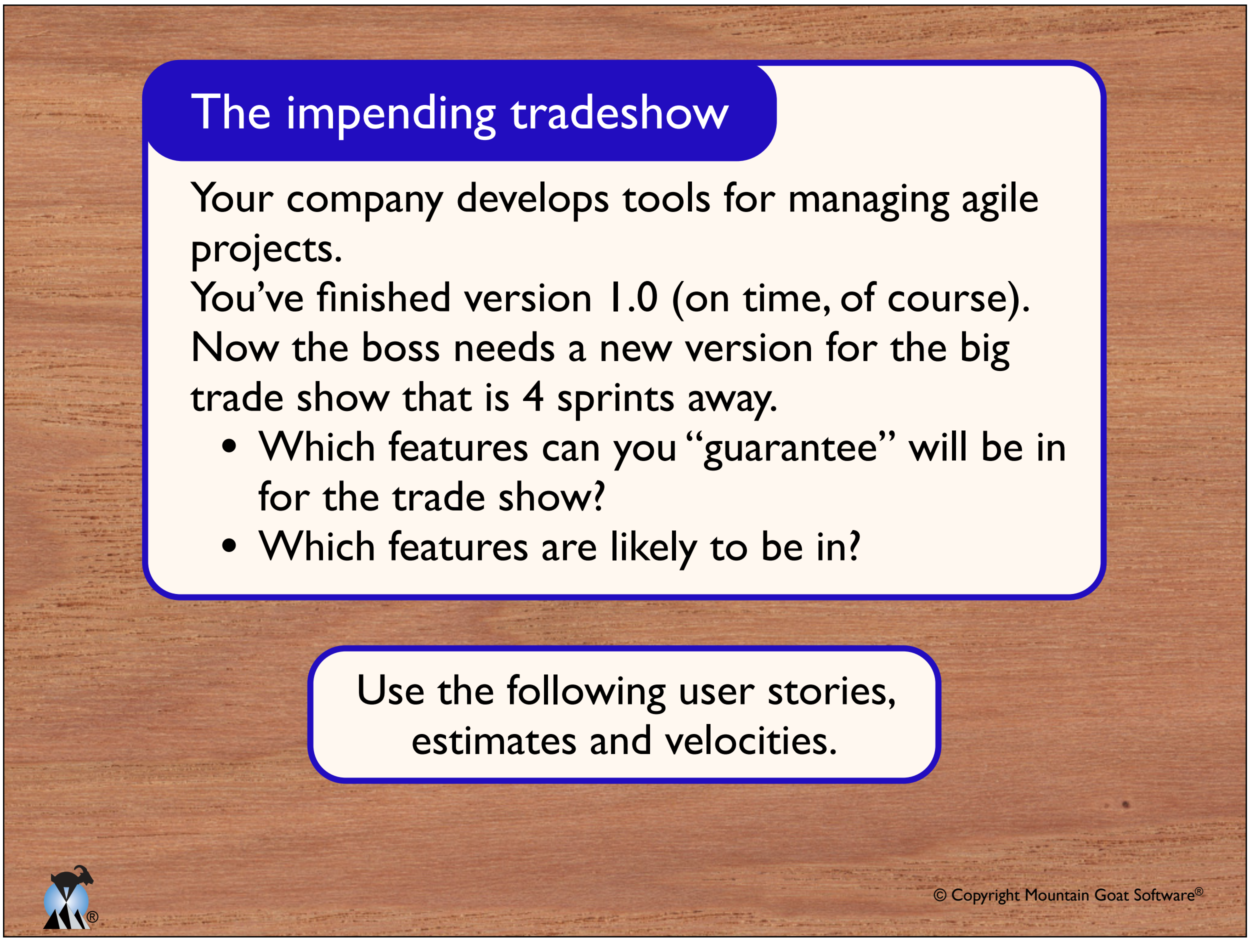
Again, it's a risk issue

- How much risk do you want to take?
- It's a balance between short- and long-term risk



Ranges

- Notice in both cases we had a range
- For a fixed date project, use a scope range:
  - "By that date you'll have all of these features and some of these."
- For a fixed-scope project, use a date range:
  - "It will take us between 5 and 8 sprints to deliver all of those features."



The impending tradeshow

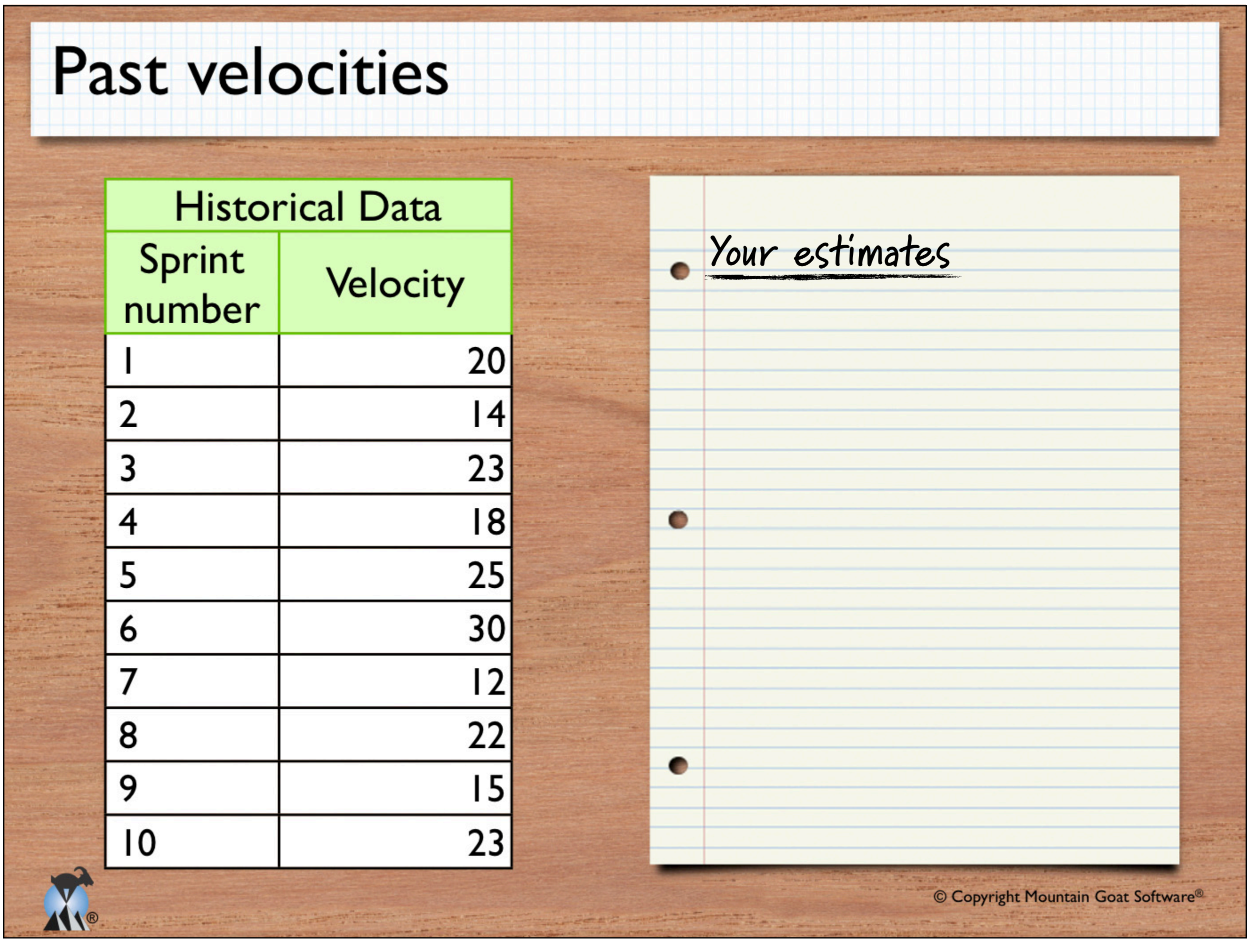
Your company develops tools for managing agile projects.

You've finished version 1.0 (on time, of course).

Now the boss needs a new version for the big trade show that is 4 sprints away.

- Which features can you "guarantee" will be in for the trade show?
- Which features are likely to be in?

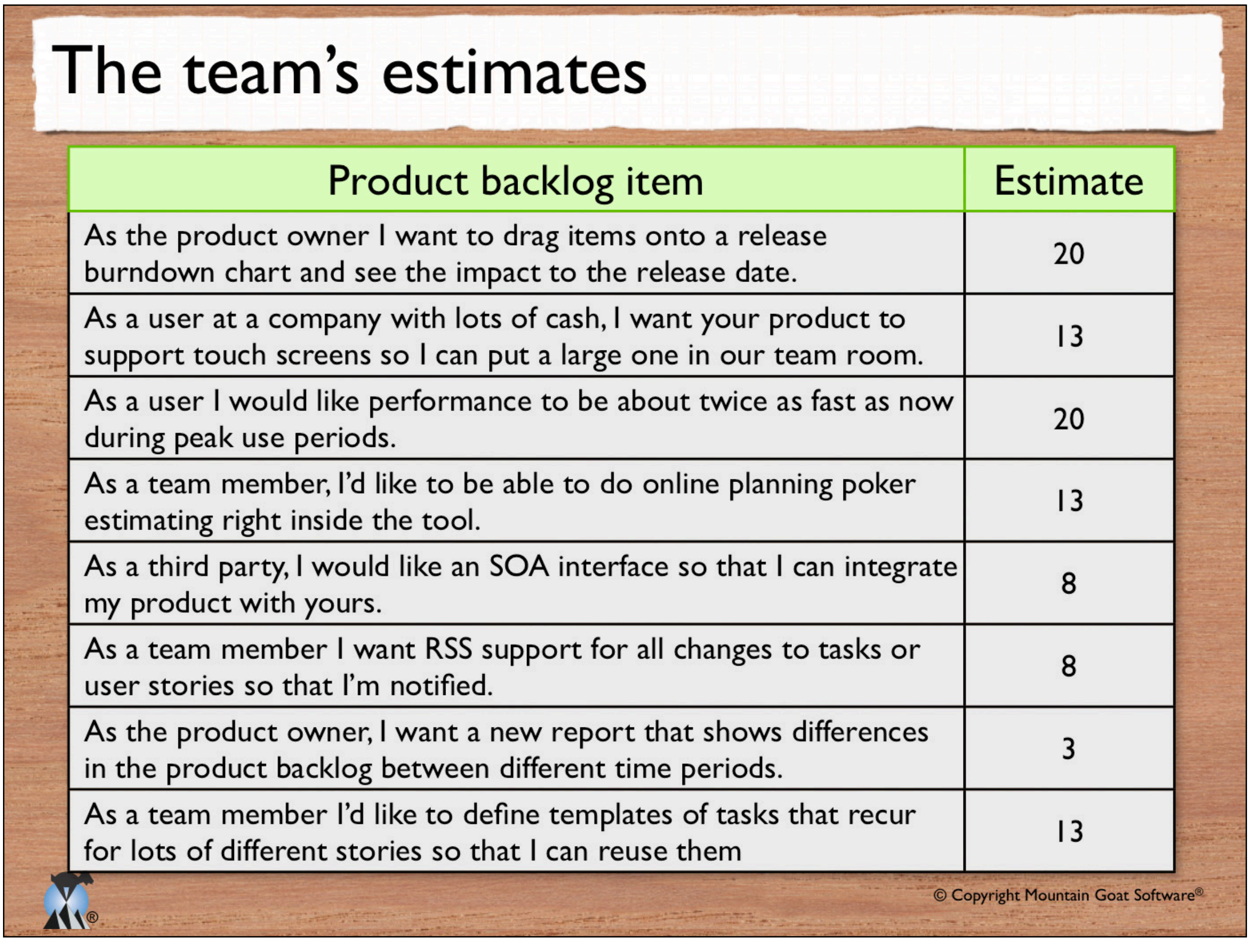
Use the following user stories, estimates and velocities.



Past velocities

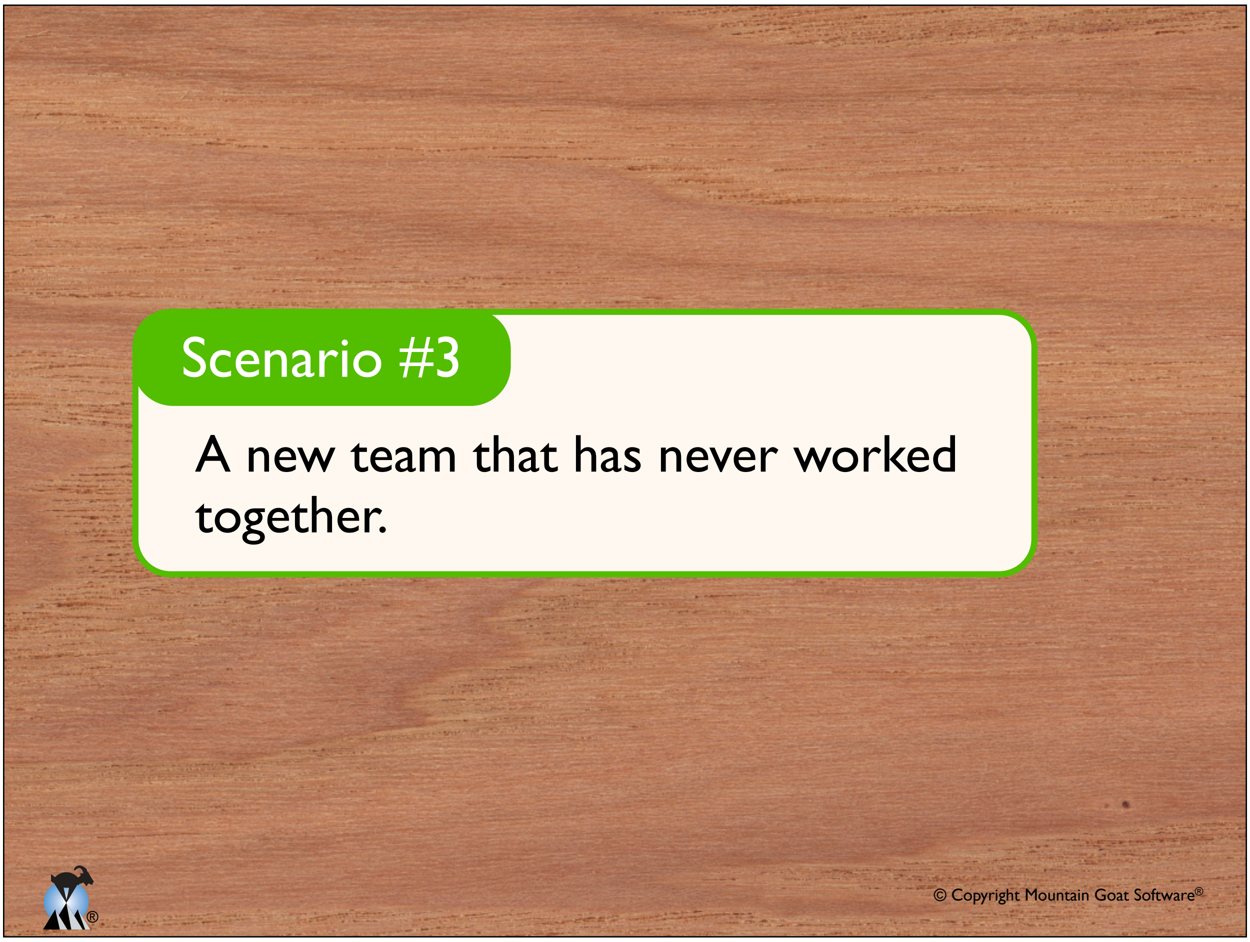
Historical Data	
Sprint number	Velocity
1	20
2	14
3	23
4	18
5	25
6	30
7	12
8	22
9	15
10	23

Your estimates



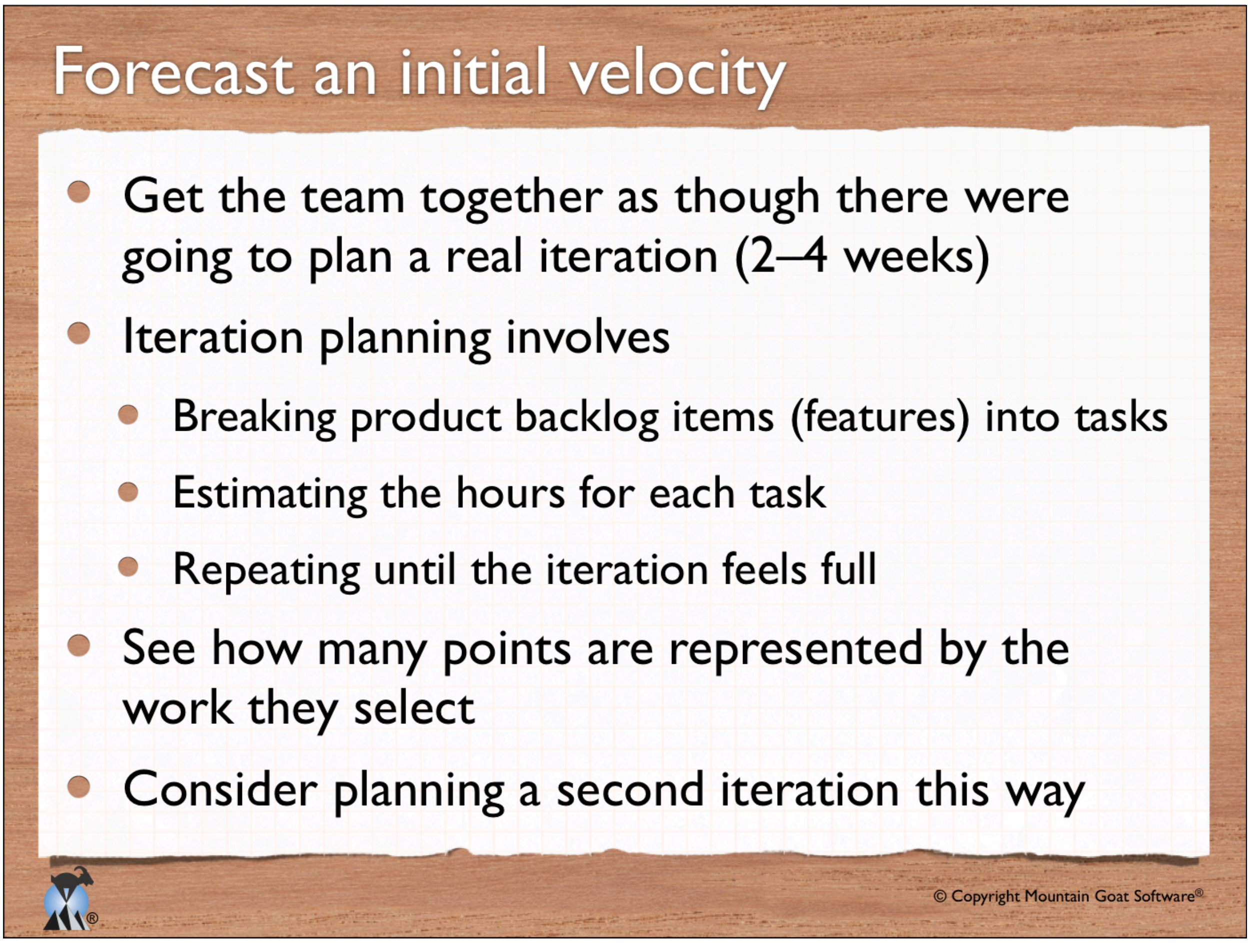
The team’s estimates

Product backlog item	Estimate
As the product owner I want to drag items onto a release burndown chart and see the impact to the release date.	20
As a user at a company with lots of cash, I want your product to support touch screens so I can put a large one in our team room.	13
As a user I would like performance to be about twice as fast as now during peak use periods.	20
As a team member, I’d like to be able to do online planning poker estimating right inside the tool.	13
As a third party, I would like an SOA interface so that I can integrate my product with yours.	8
As a team member I want RSS support for all changes to tasks or user stories so that I’m notified.	8
As the product owner, I want a new report that shows differences in the product backlog between different time periods.	3
As a team member I’d like to define templates of tasks that recur for lots of different stories so that I can reuse them	13



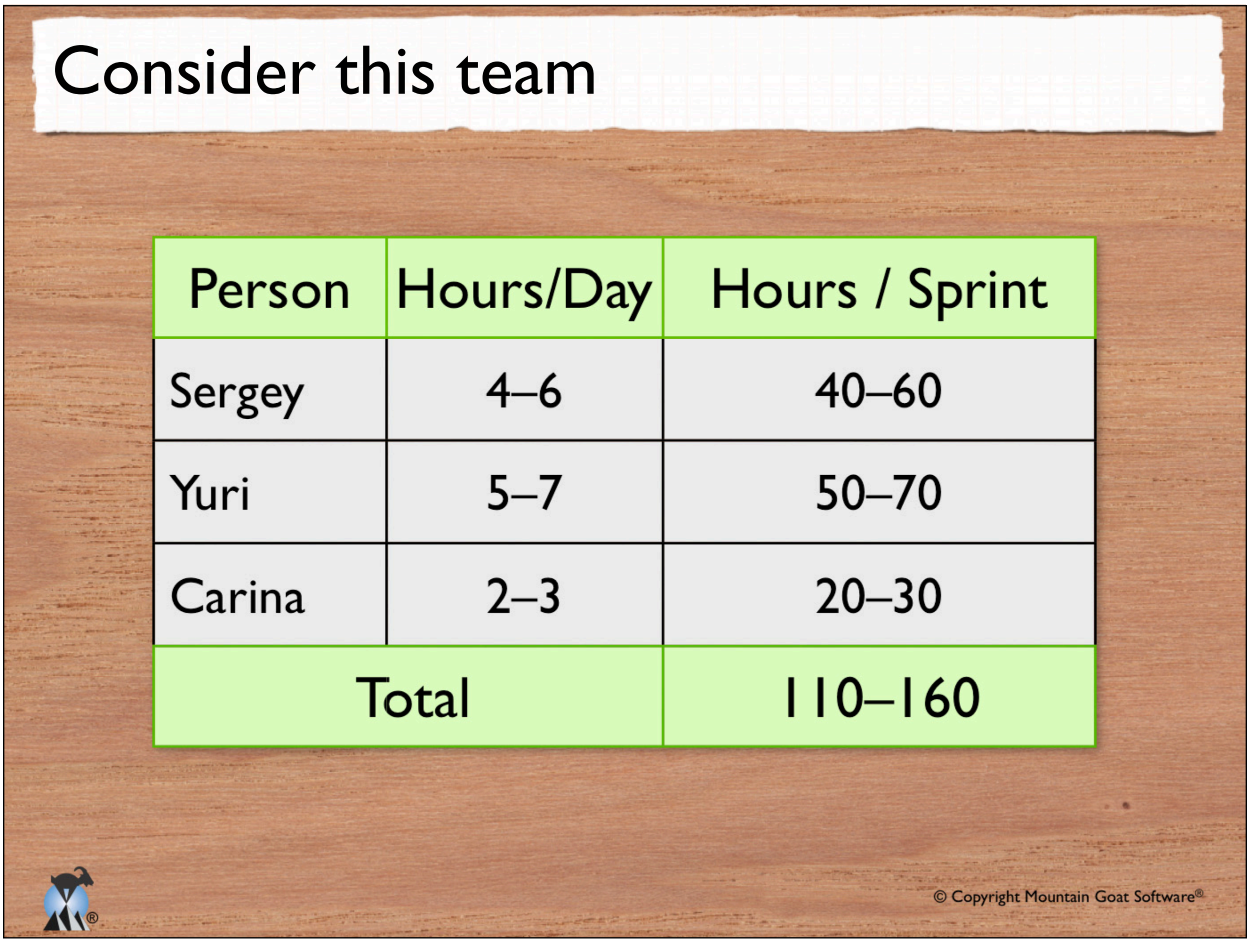
Scenario #3

A new team that has never worked together.



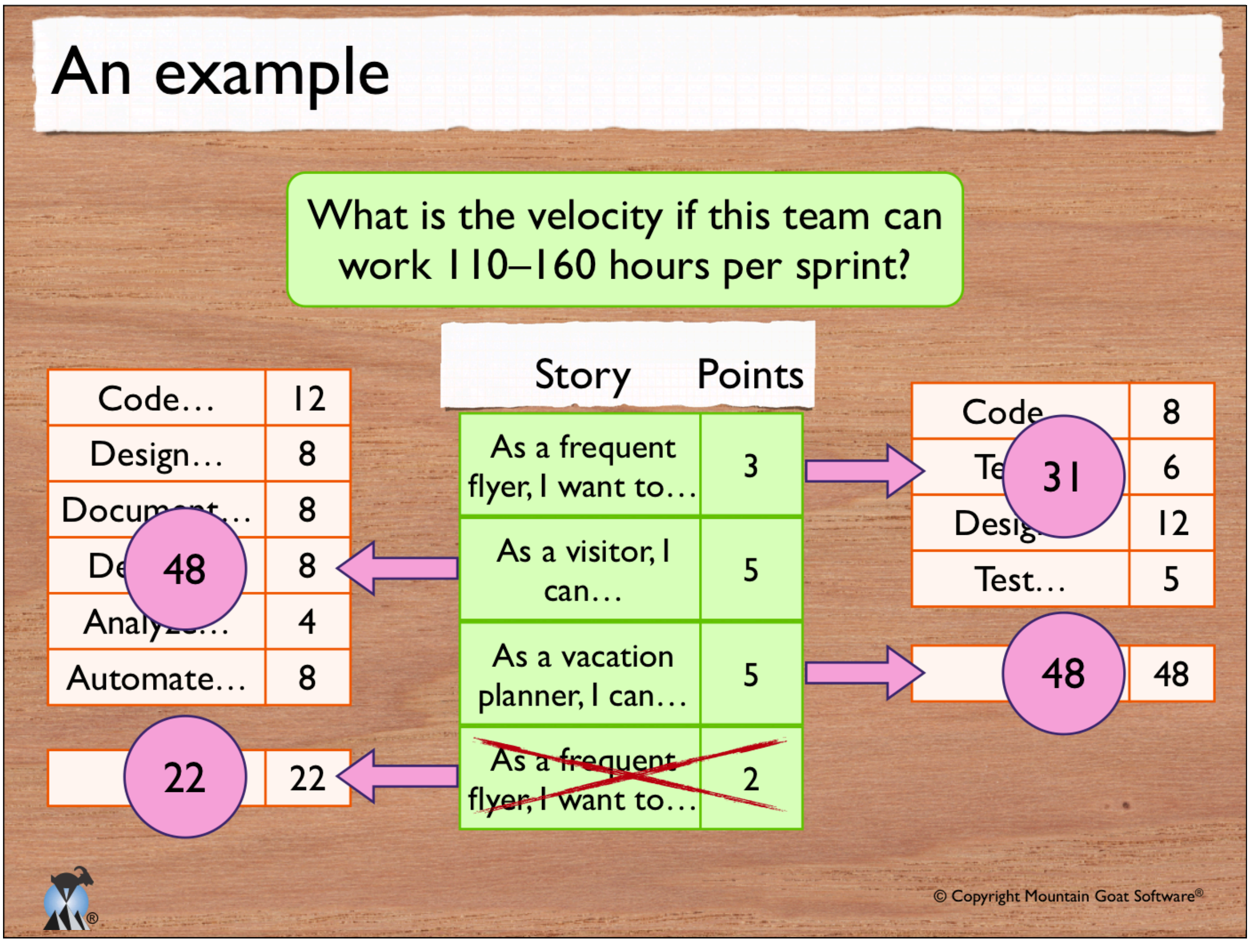
Forecast an initial velocity

- Get the team together as though there were going to plan a real iteration (2–4 weeks)
- Iteration planning involves
  - Breaking product backlog items (features) into tasks
  - Estimating the hours for each task
  - Repeating until the iteration feels full
- See how many points are represented by the work they select
- Consider planning a second iteration this way



Consider this team

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



An example

What is the velocity if this team can work 110–160 hours per sprint?

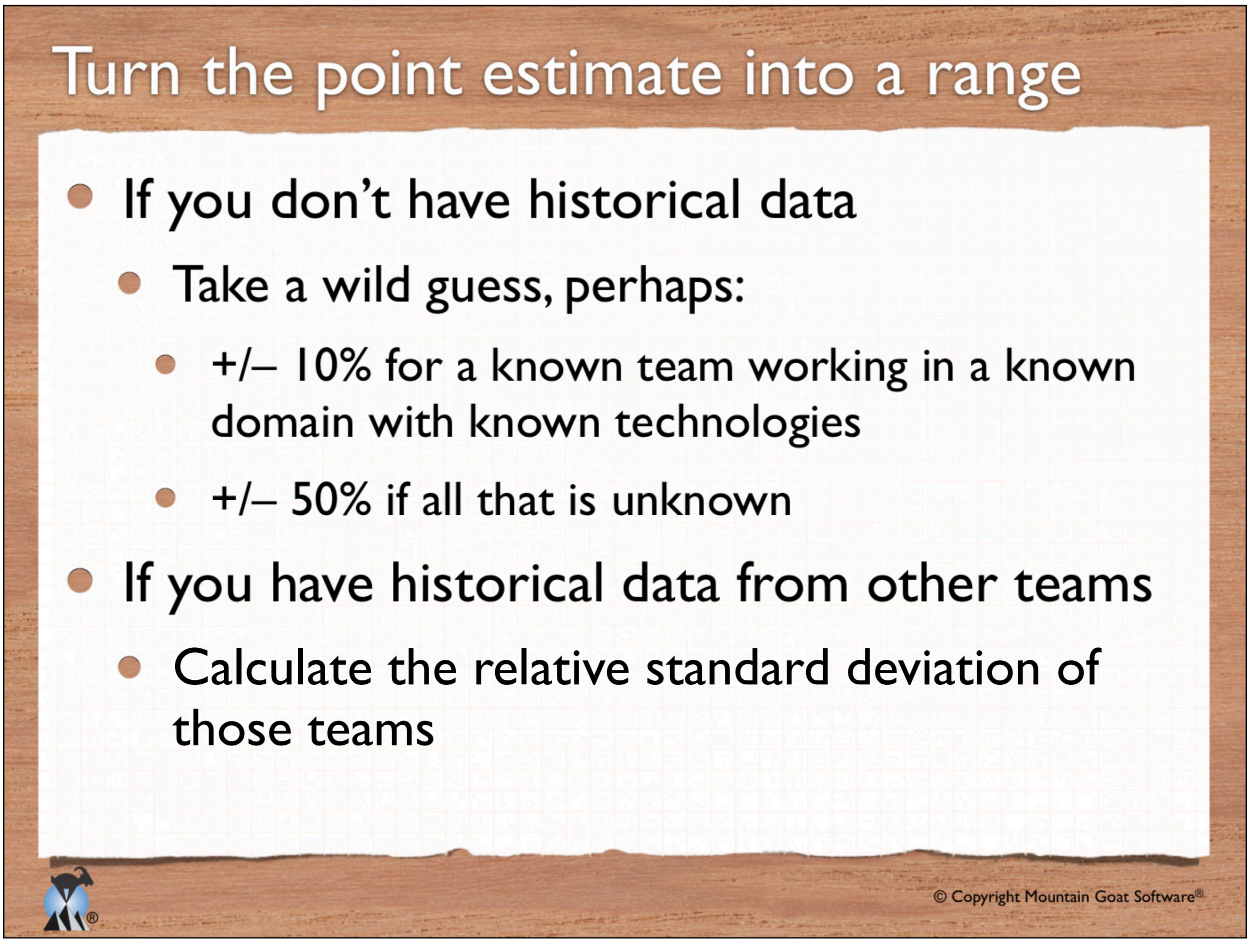
Story	Points
As a frequent flyer, I want to…	3
As a visitor, I can…	5
As a vacation planner, I can…	5
~~As a frequent flyer, I want to…~~	~~2~~

Code…	12
Design…	8
Docum…	8
De	8
Analy…	4
Automate…	8
22	22

48

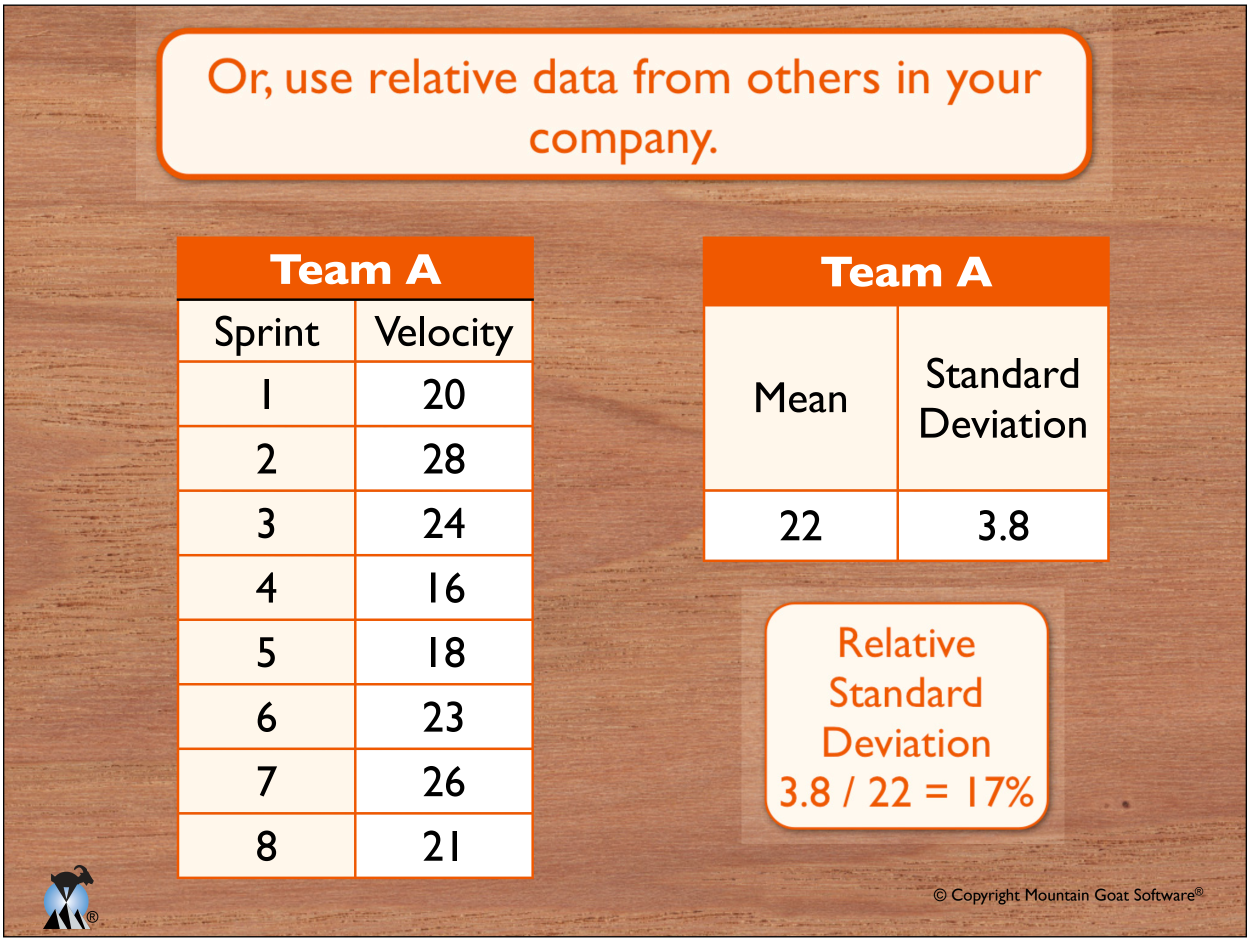
Code	8
Te	6
Desig	12
Test…	5
48	48

31



Turn the point estimate into a range

- If you don’t have historical data
  - Take a wild guess, perhaps:
    - +/– 10% for a known team working in a known domain with known technologies
    - +/– 50% if all that is unknown
- If you have historical data from other teams
  - Calculate the relative standard deviation of those teams

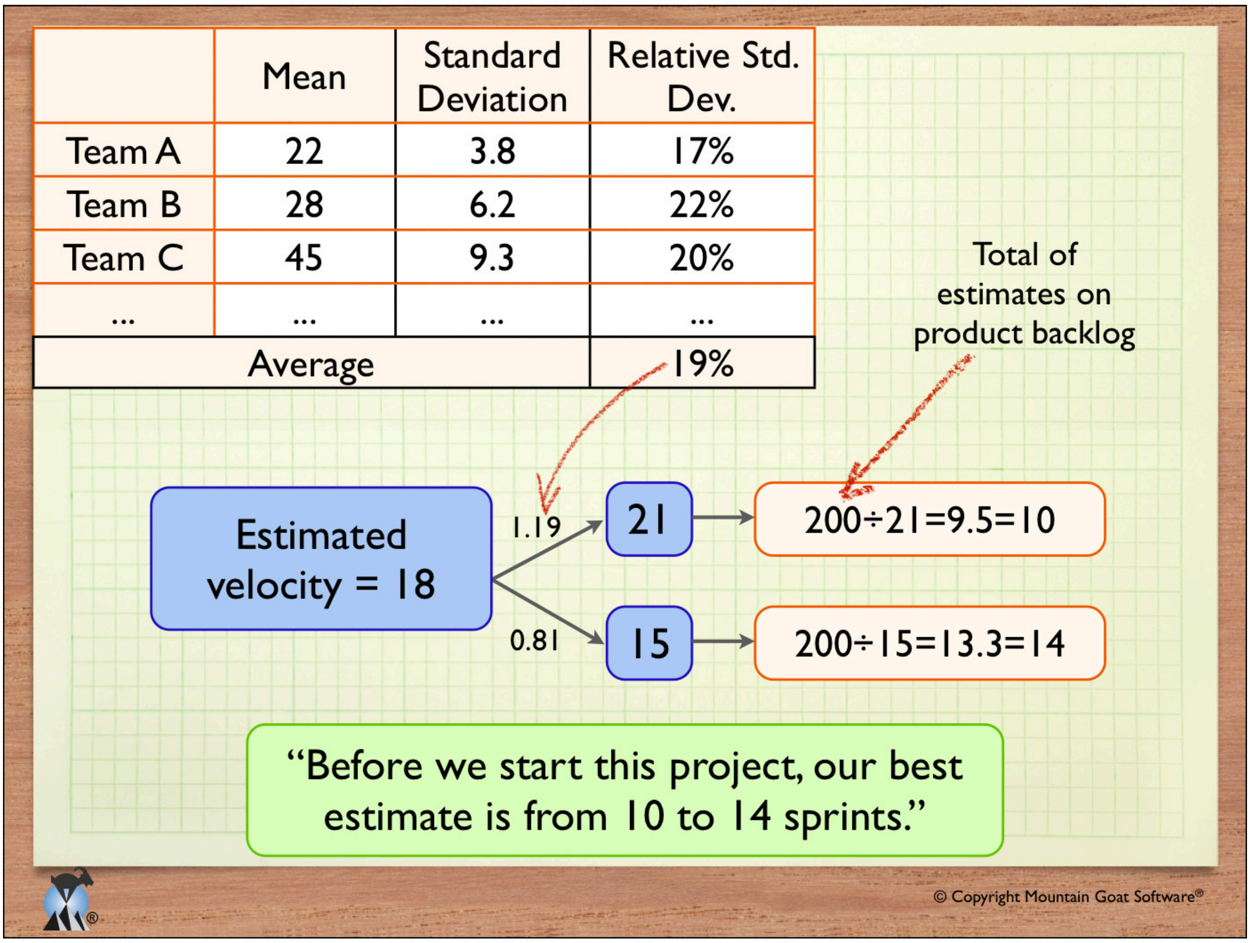


Or, use relative data from others in your company.

Team A	
Sprint	Velocity
1	20
2	28
3	24
4	16
5	18
6	23
7	26
8	21

Team A	
Mean	Standard Deviation
22	3.8

Relative Standard Deviation
3.8 / 22 = 17%



	Mean	Standard Deviation	Relative Std. Dev.
Team A	22	3.8	17%
Team B	28	6.2	22%
Team C	45	9.3	20%
...	...	...	...
Average			19%

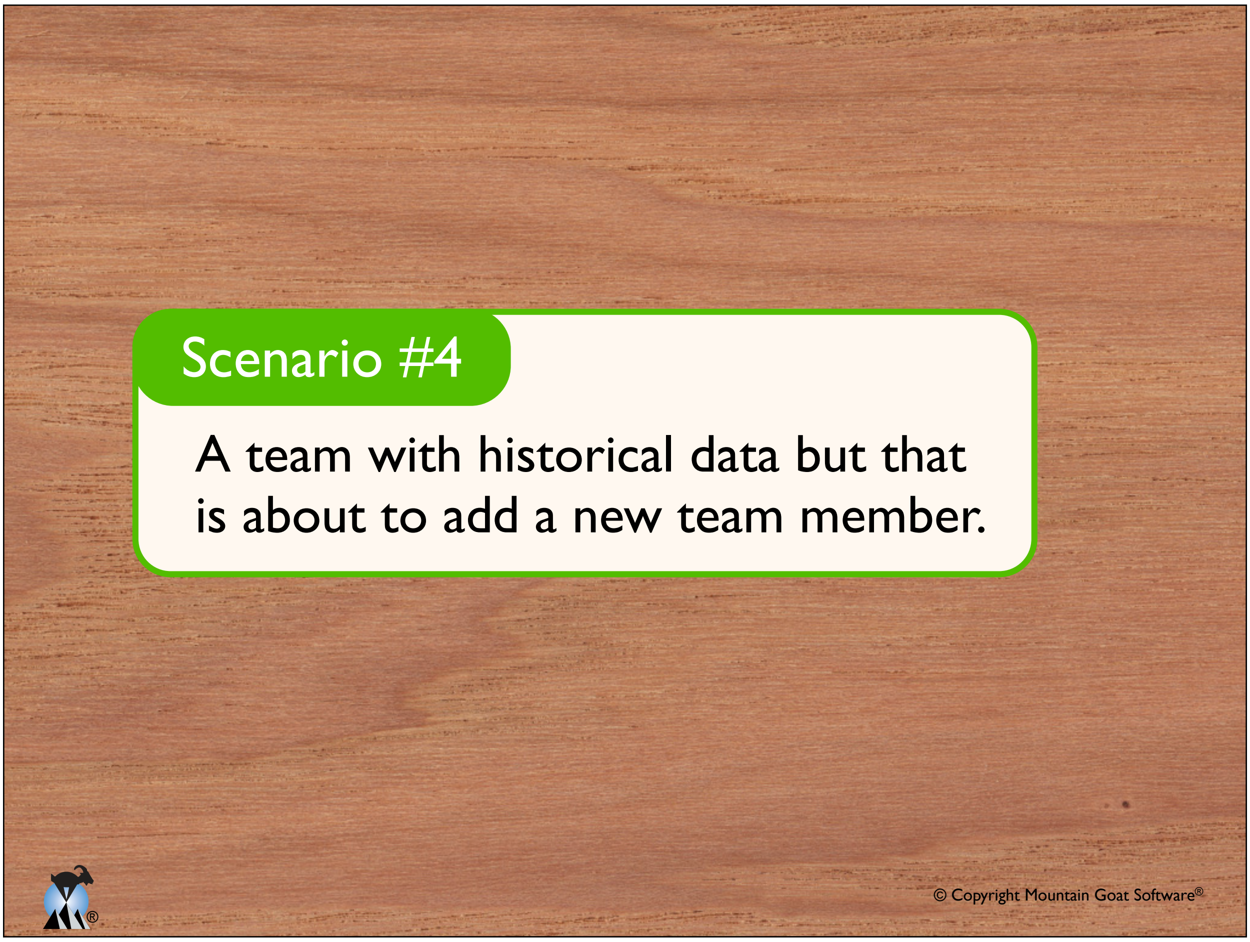
Total of estimates on product backlog

Estimated velocity = 18

200÷21=9.5=10

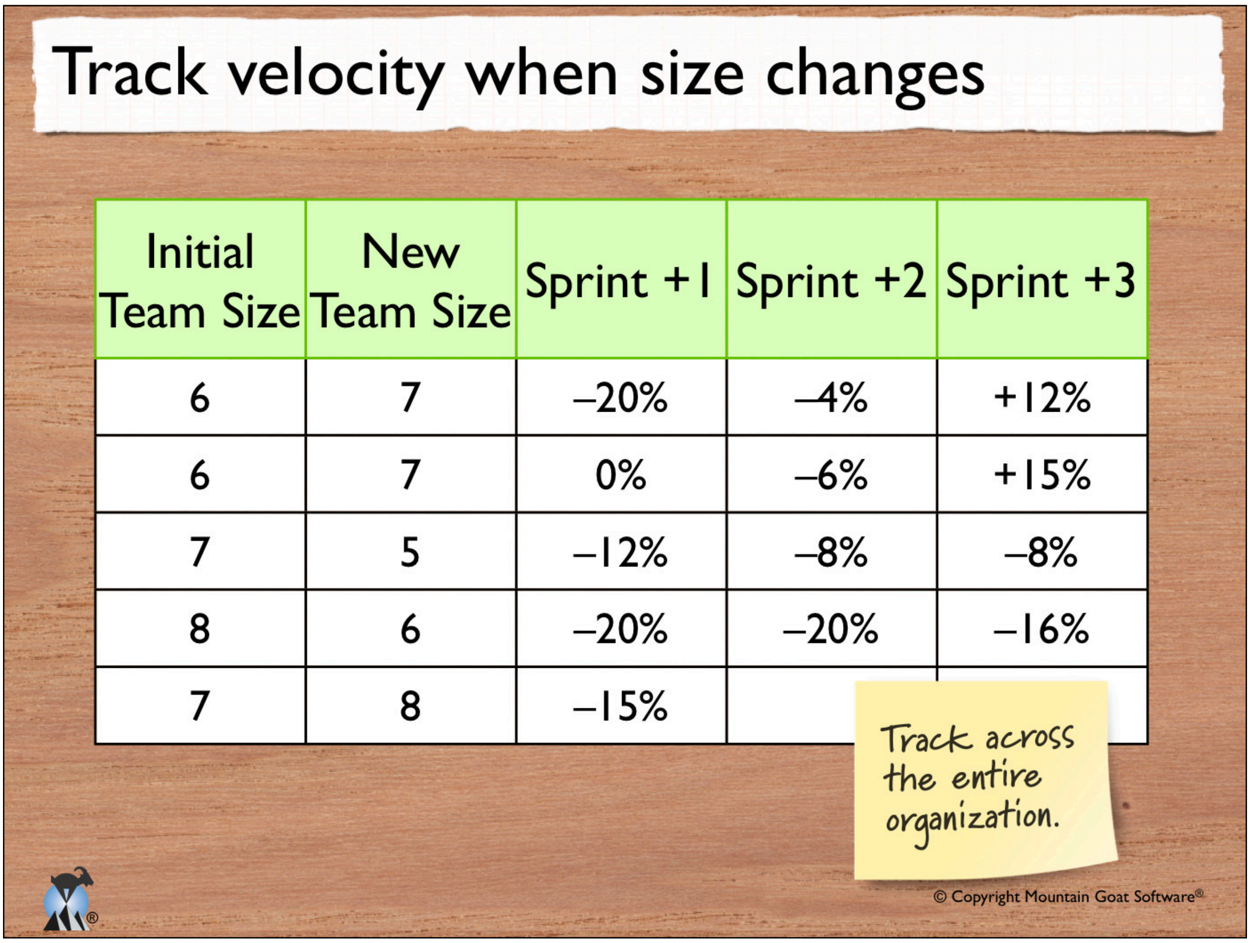
200÷15=13.3=14

"Before we start this project, our best estimate is from 10 to 14 sprints."



Scenario #4

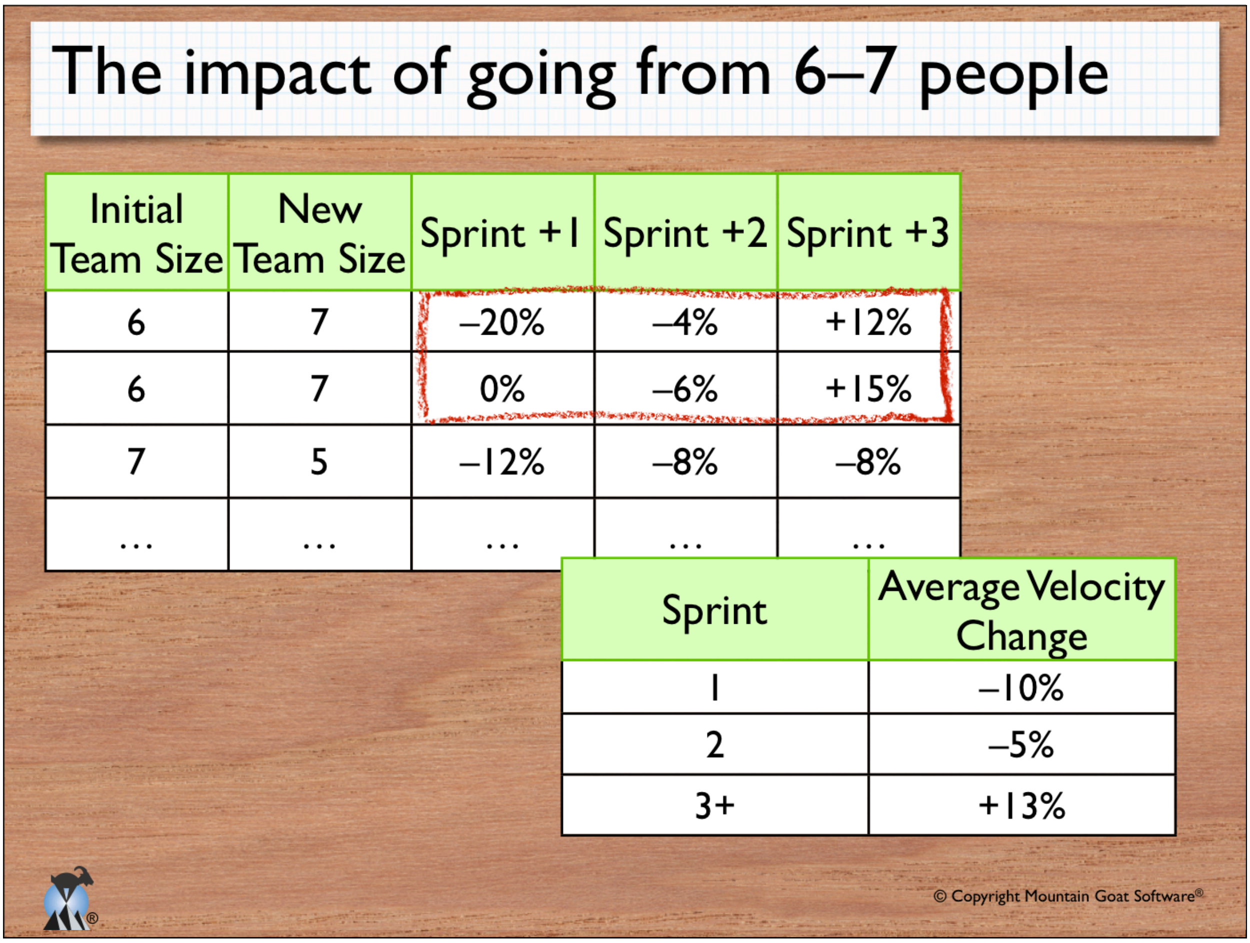
A team with historical data but that is about to add a new team member.



Track velocity when size changes

Initial Team Size	New Team Size	Sprint +1	Sprint +2	Sprint +3
6	7	–20%	–4%	+12%
6	7	0%	–6%	+15%
7	5	–12%	–8%	–8%
8	6	–20%	–20%	–16%
7	8	–15%		

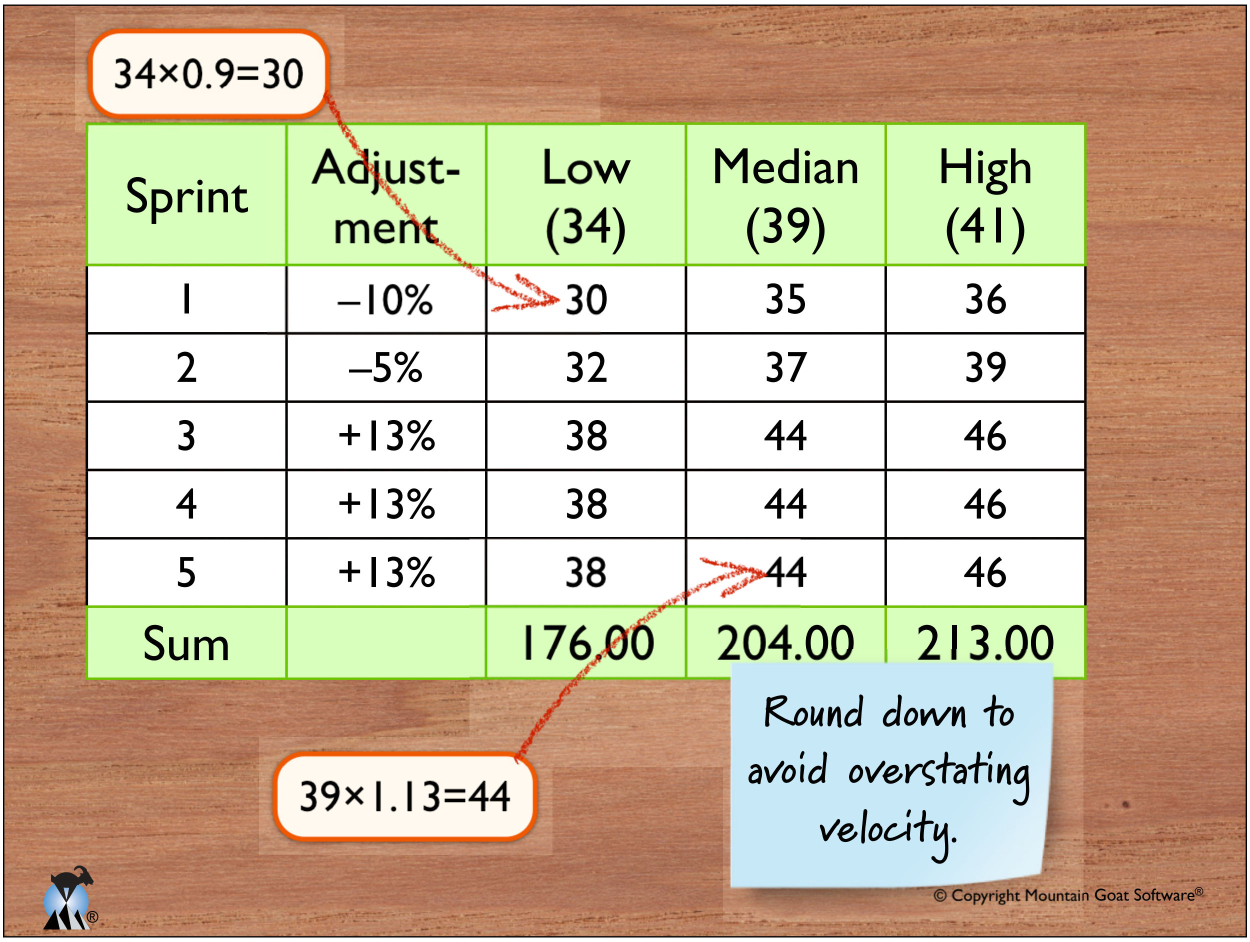
Track across the entire organization.



The impact of going from 6–7 people

Initial Team Size	New Team Size	Sprint +1	Sprint +2	Sprint +3
6	7	–20%	–4%	+12%
6	7	0%	–6%	+15%
7	5	–12%	–8%	–8%
…	…	…	…	…

Sprint	Average Velocity Change
1	–10%
2	–5%
3+	+13%

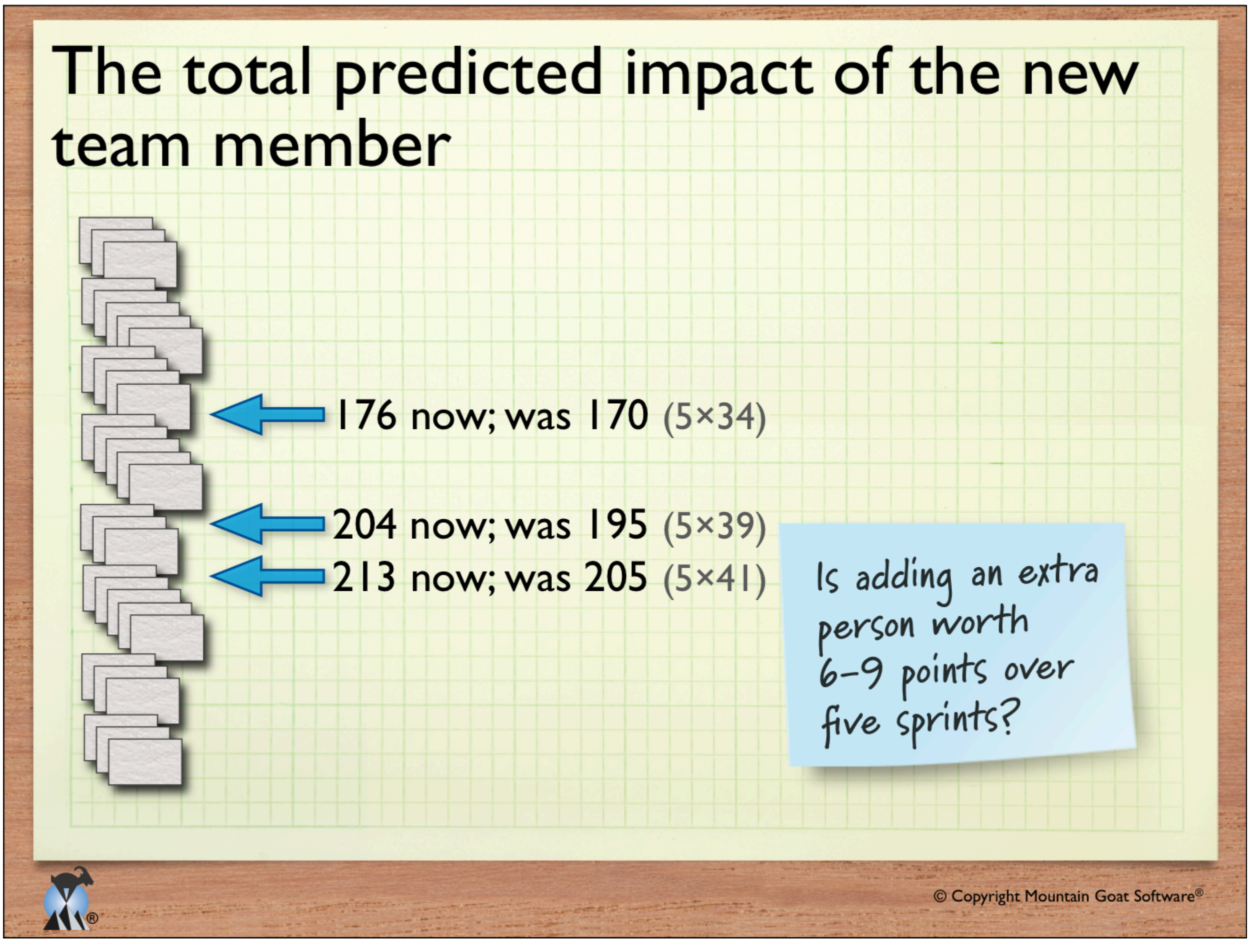


34×0.9=30

Sprint	Adjust-ment	Low (34)	Median (39)	High (41)
1	–10%	30	35	36
2	–5%	32	37	39
3	+13%	38	44	46
4	+13%	38	44	46
5	+13%	38	44	46
Sum		176.00	204.00	213.00

39×1.13=44

Round down to avoid overstating velocity.



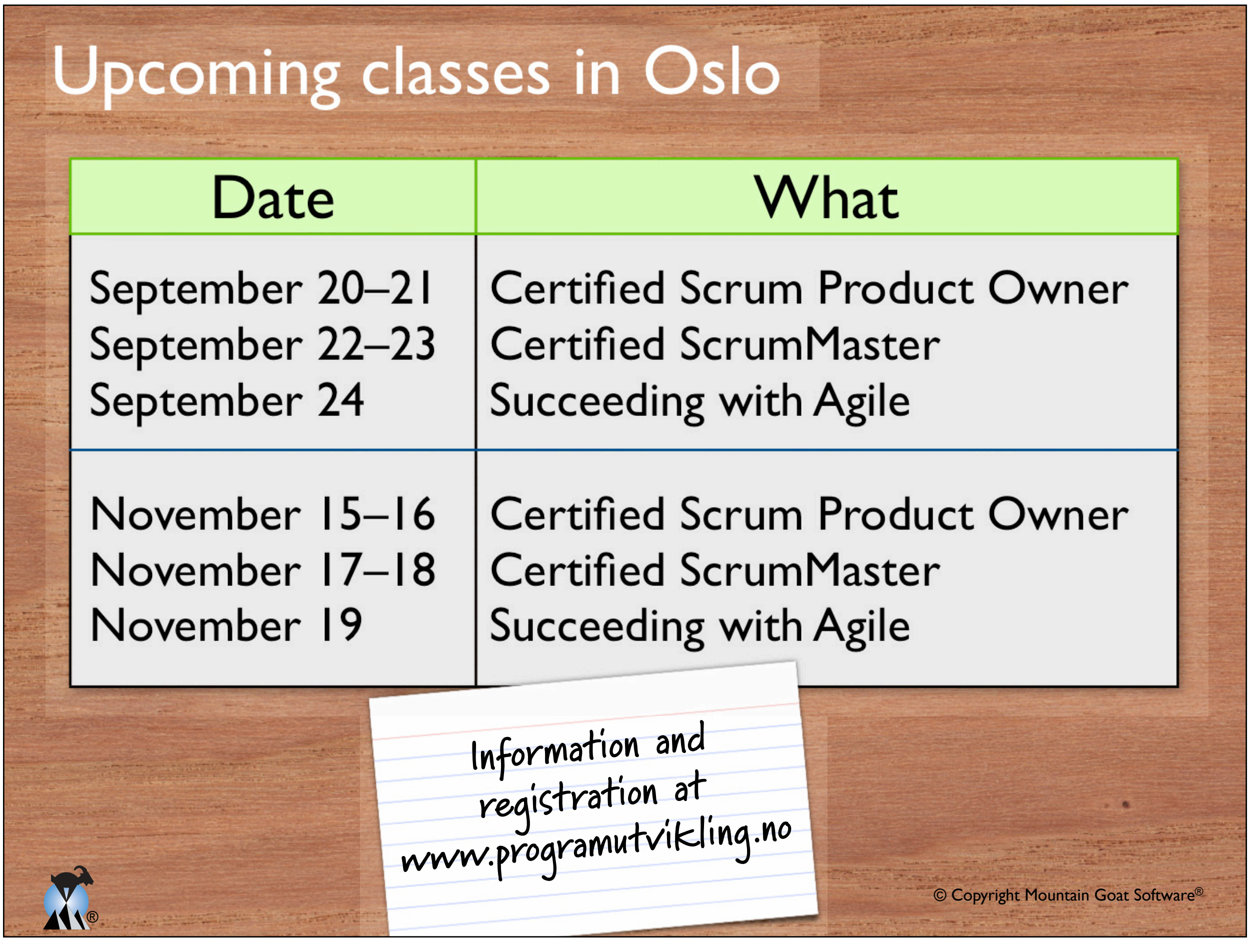
The total predicted impact of the new team member

176 now; was 170 (5×34)

204 now; was 195 (5×39)

213 now; was 205 (5×41)

Is adding an extra person worth 6–9 points over five sprints?



Upcoming classes in Oslo

Date	What
September 20–21 September 22–23 September 24	Certified Scrum Product Owner Certified ScrumMaster Succeeding with Agile
November 15–16 November 17–18 November 19	Certified Scrum Product Owner Certified ScrumMaster Succeeding with Agile

Information and registration at www.programutvikling.no

The Addison Wesley Signature Series
User Stories Applied
For Agile Software Development
Mike Cohn
Foreword by Kent Beck
A Kent Beck Signature Book
Robert C. Martin Series
Agile Estimating and Planning
Prentice Hall PTR
Forewords by Jim Highsmith, Agile P
Gabrielle ene ield , Director, Agile Pr
The Addison Wesley Signature Series
Succeeding with Agile
Software Development Using Scrum
Mike Cohn
A Mike Cohn Signature Book
Foreword by Tim Lister

Mike Cohn

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

twitter: mikewcohn

(720) 890−6110

MOUNTAIN GOAT SOFTWARE