

© Copyright Mountain Goat Software®

1

## 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



© Copyright Mountain Goat Software®

2

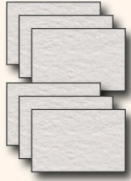




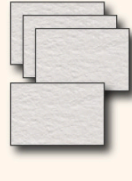
# Release and sprint planning

## Release Plan

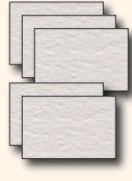
Sprint 1



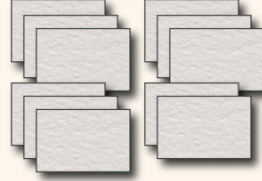
Sprint 2



Sprint 3



Sprint 4-7



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



© Copyright Mountain Goat Software®

3

## 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 February
  - We'll be done February 18th

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

John Maynard Keynes



© Copyright Mountain Goat Software®

4



# 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

A photograph of a wooden desk with a white bowl of colorful markers, a pink eraser, and two pencils. Two pieces of lined paper with handwritten notes are placed on the desk. The notes contain questions about agile planning, such as 'Estimating', 'How do I best use historical velocity data?', and 'How do I reliably commit to a fixed scope or fixed date?'.

- Estimating
- How do I best use historical velocity data?
- How do I estimate velocity for a new team?
- How do I estimate velocity if team size changes?

- How do I reliably commit to a fixed scope or fixed date?
- How do I create a plan when there is a tremendous amount of uncertainty in what we're building?
- When should I re-estimate?



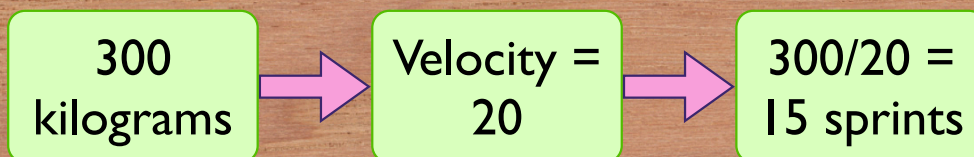
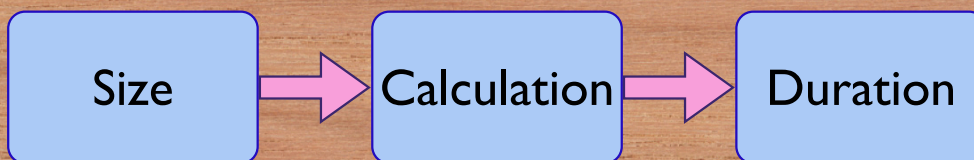


# How long will it take...

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



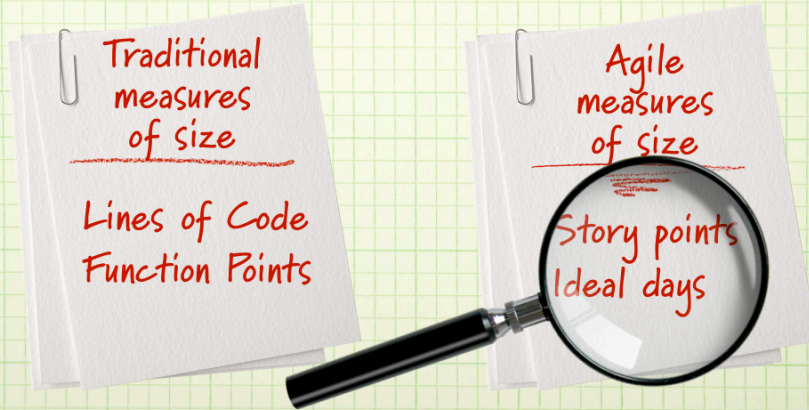
# Estimate size; derive duration





## Measures of size

- Traditional and agile measure size differently



## Story points

- How long a user story will take (effort)
- Influenced by complexity, uncertainty, risk, volume of work, etc.
- Relative values are what is important:
  - A login screen is a 2.
  - A search feature is an 8.
- Basic math properties should hold
  - $5+5 = 10$





How do I best use  
historical velocity data?



© Copyright Mountain Goat Software®

11

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

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

# of Historical Sprints	Sprints to throw out at each end
5	0
8	1
11	2
13	3
16	4
18	5
21	6
23	7
26	8

Use the online velocity range calculator at [www.mountaingoatsoftware.com/tools](http://www.mountaingoatsoftware.com/tools)



© Copyright Mountain Goat Software®

12



# Extrapolate from the velocity range

Assume:  
There are five sprints left.



← We'll almost certainly get here ( $5 \times 34$ )

← At our median velocity we'll get here ( $5 \times 39$ )

← The most we could realistically expect ( $5 \times 41$ )



How can I estimate  
velocity for a new  
team?





## 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

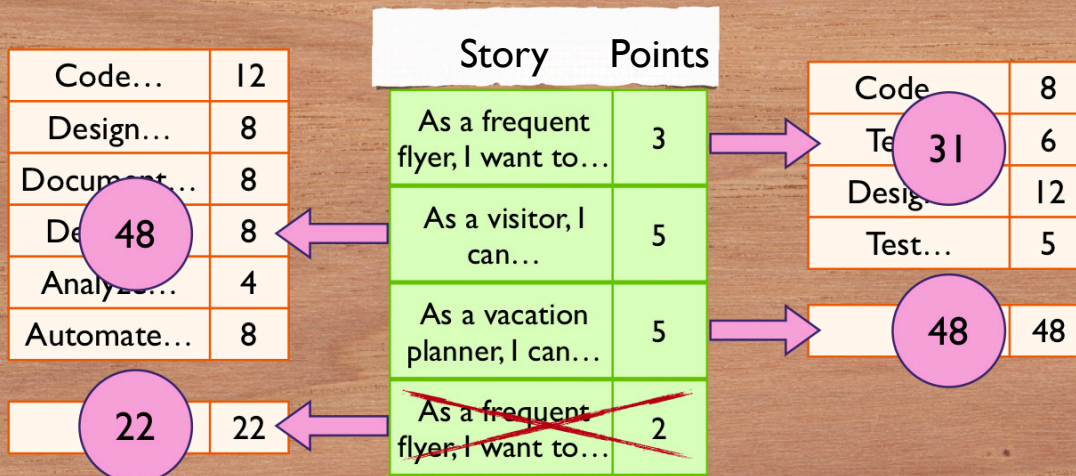
Person	Hours/Day	Hours / Sprint
Sergey	4–6	40–60
Yuri	5–7	50–70
Carina	2–3	20–30
Total		110–160





# An example

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



Sergey, Yuri and Carina have 110–160 hours available. What's their likely velocity?

## Story 1 (5 points)

Task 1a	4 hours
Task 1b	8 hours
Task 1c	16 hours
Task 1d	16 hours

## Story 3 (5 points)

Task 3A	4 hours
Task 3b	8 hours
Task 3c	12 hours

## Story 5 (8 points)

Task 5a	4 hours
Task 5b	8 hours
Task 5c	16 hours
Task 5d	16 hours

## Story 2 (8 points)

Task 2a	16 hours
Task 2b	8 hours
Task 2c	16 hours
Task 2d	8 hours
Task 2e	6 hours

## Story 4 (8 points)

Task 4a	16 hours
Task 4b	16 hours
Task 4c	16 hours
Task 4d	4 hours
Task 4e	4 hours
Task 4f	8 hours

## Story 6 (2 points)

Task 6a	4 hours
Task 6b	8 hours
Task 6c	2 hours
Task 6d	4 hours





## 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



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

1.19 → 21 →  $200 \div 21 = 9.5 = 10$

0.81 → 15 →  $200 \div 15 = 13.3 = 14$

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

© Copyright Mountain Goat Software®

21

How do I estimate velocity if team size will change during the project?

© Copyright Mountain Goat Software®

22



## Overall approach

- If future team size changes are likely to be similar to past team size changes:
  - Use historical velocity without any adjustments for team composition
- If future team size changes will be different:
  - Predict the impact of those changes with data from past team size changes



## 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%



$$34 \times 0.9 = 30$$

Sprint	Adjustment	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
<b>Sum</b>		<b>176</b>	<b>204</b>	<b>213</b>

$$39 \times 1.13 = 44$$

Round down to avoid overstating velocity.





# Predicted impact of a new team member



← 176 now; was 170 ( $5 \times 34$ )

← 204 now; was 195 ( $5 \times 39$ )

← 213 now; was 205 ( $5 \times 41$ )

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



How do I reliably  
commit to a fixed scope  
or fixed date?





# Fixed-date planning

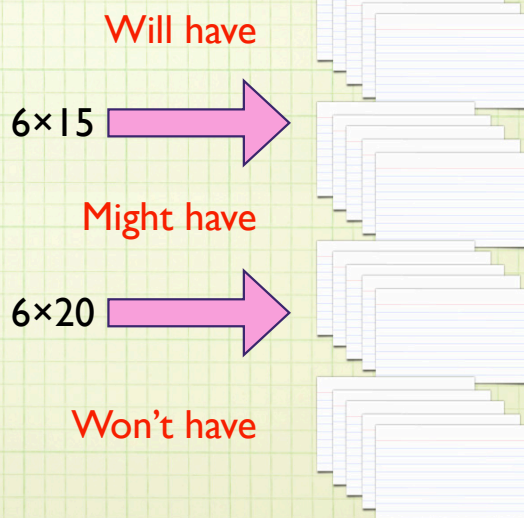
How much can I get by <date>?

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”



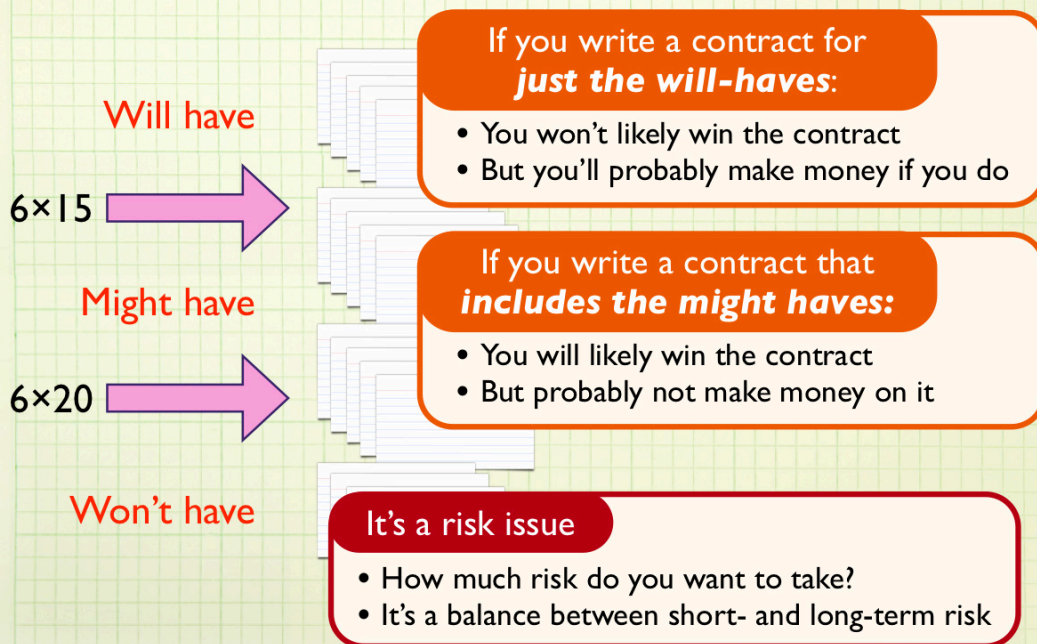
## Fixed-date planning: an example

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





# Fixed-date contracting



# 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

$120 \div 20 =$



$120 \div 15 =$



33

## 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



34



# 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





How do I create a plan  
when there is a  
tremendous amount of  
uncertainty in what  
we're building?



© Copyright Mountain Goat Software®

39

## My trip to the airport

Find  
keys  
1 5

Drive to airport  
45 30

Park  
5 10

Check-in  
7 30

Security  
7 30

50%  
estimate  
Time=1:05

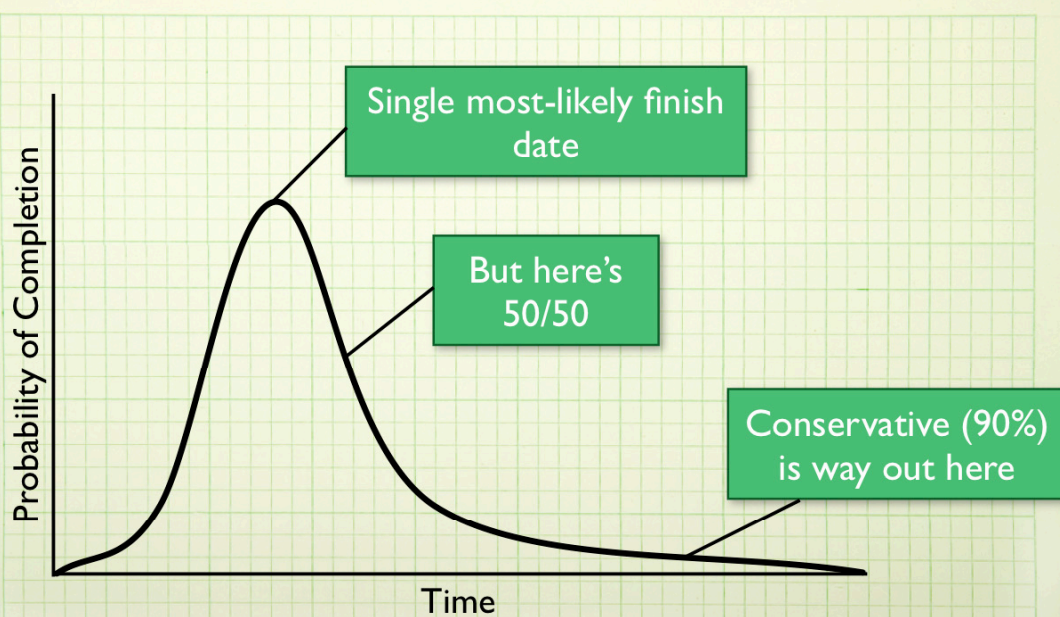
Buffer to  
90%  
Time=1:45



© Copyright Mountain Goat Software®

40

# Distribution of completion times



© Copyright Mountain Goat Software®

41

# Give 50% and 90% estimates

- 50% estimates
  - Remove all local safety: no “padding”
  - An estimate you should / will miss more than half the time
- 90% estimates
  - Not really a worst case
  - No lightning strikes or busses running over people
- Keep in mind that you’ll even exceed this estimate occasionally



© Copyright Mountain Goat Software®

42



# Calibration

		90% Confidence Interval	
		Low	High
1	When was Elvis Presley born?		
2	What is the latitude of Rome? (Hint: latitude is 0 at the equator at 90 at the North Pole.)		
3	How many airplanes did the U.S. military own in 1913?		
4	How many miles or kilometers long is the Nile?		
5	<i>Gone with the Wind</i> won the Oscar for best picture in what year?		
6	If you could jump 50 feet straight up into the air, how many seconds would you be airborne before you landed?		
7	In what year was the first European printing press invented?		
8	How many casualties did the French suffer in the Battle of Waterloo?		
9	How many soldiers were in a Roman legion?		
10	What is the wingspan (in feet or meters) of a 747?		



Source: *How to Measure Anything*, Doug Hubbard.

© Copyright Mountain Goat Software®

43

# Deriving the duration

- We can't add the 50% estimates together
  - That assumes everything goes smoothly
  - Overall schedule will be too short
- We can't add the 90% estimates together
  - That assumes that everything goes wrong
  - Overall schedule will be too long

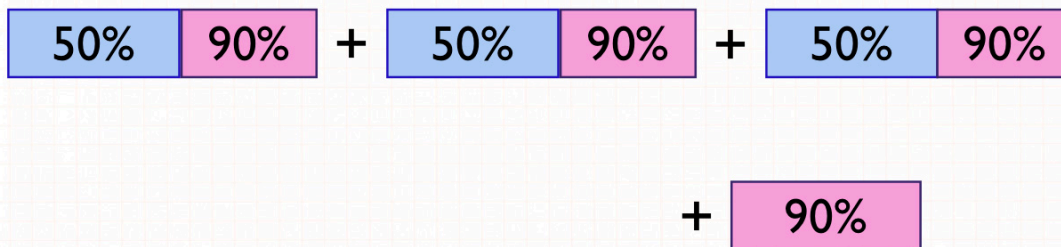


© Copyright Mountain Goat Software®

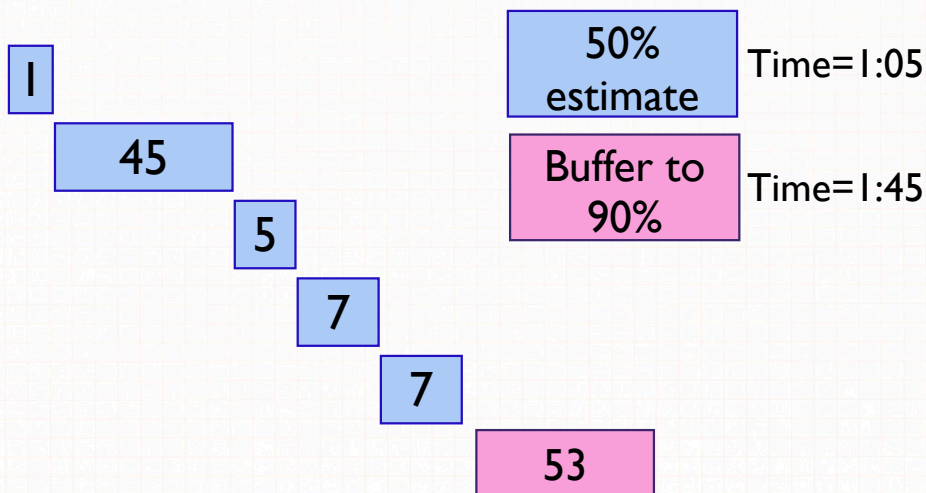
44

## The solution

- We add the 50% estimates
- And buffer the overall project, rather than the tasks



## My trip to the airport with a buffer





## A buffer isn't padding

- Padding is extra time you don't think you'll need but add to be safe
  - “Padding” implies it was unnecessary
- You will need the project buffer
- Even with the project buffer you're not guaranteed to be done on time



## The size of the buffer

- Simple rule
  - Use 50% of the unbuffered (50%) schedule
- More sophisticated, usually better
$$\sqrt{(h_1 - l_1)^2 + (h_2 - l_2)^2 + \dots + (h_n - l_n)^2}$$
  - $h$  = high estimate (90%)
  - $l$  = low estimate (50%)
- The more sophisticated approach considers the extreme variability of some work
  - “This will take 2 days if it goes well, or 20 if not.”



## When do you do this?

- When the cost of being wrong is significant
  - Highly political environment
  - A great deal of extra activity (press releases, ads, training, etc.) is planned for the release date
  - When there's a great deal of uncertainty or risk in the product backlog
  - When you're at risk (e.g., contract development)
- Don't bother in most cases
  - Trusting environment
  - When there's little impact to refining release date midway through the project



## Sample buffer calculation

Story	0.50	0.90	(90%-50%)
Story A	2	5	9
Story B	2	5	9
Story C	1	5	16
Story D	1	3	4
Story E	5	8	9
Story F	5	13	64
<b>Total</b>	<b>16</b>	<b>39</b>	<b>111</b>

$$\text{Schedule} = 16 + \sqrt{111} = 16 + 11 = 27$$





When should I re-estimate?

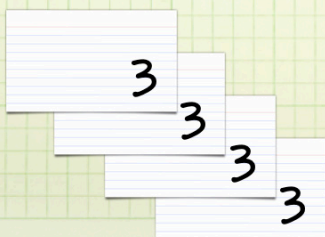
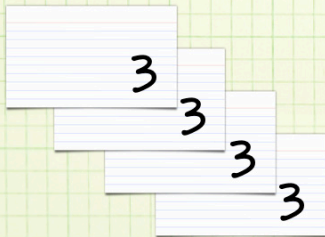


© Copyright Mountain Goat Software®

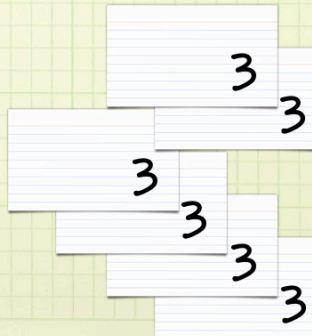
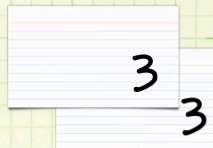
51

## Re-estimating

We expect velocity = 12



But we finish half that much

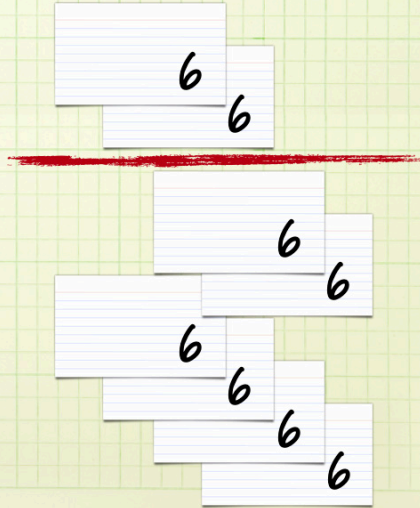
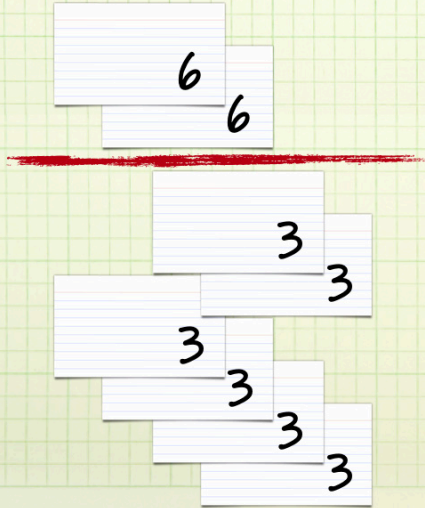


© Copyright Mountain Goat Software®

52

But a velocity of 6 doesn't seem right; so we double those stories

But if all stories are the same size, we need to double all of them



		Estimate	Planned	Finished
A	As a swimmer, I can see a line chart of my times for a particular event.	3	✓	✓
B	As a coach, I can have the system recommend who should swim in each event subject to restrictions about how many events a swimmer can participate in.	5	✓	✓
C	As a coach, I can see a line chart showing the progress over the season of all swimmers in a particular event.	5	✓	
D	As a swimmer, I can see a pie chart showing how many first, second, third, and lower places I've finished in.	3		
E	As a coach, I can see a text report showing each swimmer's best time in each event.	3		
F	As a coach, I can upload meet results from a file exported from the timing system used at the meet.	1		





## The current situation

- The team has just finished the first sprint
- They'd planned to do stories A, B, and C
- They only finished A and B
  - Largely because Story A was much harder than expected
  - "It should have been at least a six."
- The real problem is not just with Story A but with all stories related to graphing (A, C, and D)
  - Graphing is twice as hard as they've estimated



## The first two situations

1

### No re-estimating

- What is the team's velocity?
- Will they be able to maintain that velocity?

2

### Re-estimate the finished story

- What happens if Story A (only) is re-estimated as 6?
- What was the velocity of the finished sprint?
- What stories are planned for the next sprint?
- Will the team finish those stories?



## Re-estimating when relative size changes

3

- Stories A, B, and C are all graphing stories; graphing is twice as hard as thought so double all these
- What was the velocity of the first sprint?
- Can the team do that much in the second sprint?

What can you conclude about re-estimating from these three scenarios?



© Copyright Mountain Goat Software®

57

## Related, dependent stories

As a manager, I am sent an email when one of my employees requests time off. ?

As an employee, I am sent an email when my manager approves or rejects my time off request. ?

- There's some initial work before we can send the first email
- Once that is done, sending other emails will be easy
- What estimates should we put on these two stories?



© Copyright Mountain Goat Software®

58



## Which option is best?

- Assumptions
  - Five points for initial email support and sending first email
  - One point for each new email type to send

1

- Five points on each since we don't know which we'll do first

2

- Five points on either story and one point on the other

3

- Three points on each



## A Guiding Tip:

The sum of all items on the product backlog should always represent the full size of the entire project.





1	1935
2	41°54'
3	23
4	4,160 miles 6,694 km.
5	1939
6	3.525
7	1450
8	37,000
9	6,000
10	196 feet 59.74m



## My upcoming classes

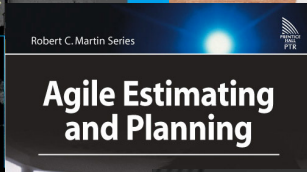
Date	What	Where
Aug 23–24 Aug 25–26	Certified ScrumMaster Succeeding with Agile	Dallas
Sept 13–14 Sept 15–16	Certified ScrumMaster Certified Scrum Product Owner	Silicon Valley
Oct 11 Oct 12–13 Oct 14	Effective User Stories Certified ScrumMaster Agile Estimating and Planning	Boulder
Nov 8–9 Nov 10–11	Certified ScrumMaster Succeeding with Agile	La Jolla

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

Classes also in  
London & Oslo







Mike Cohn  
mike@moungatsoftware.com  
www.moungatsoftware.com  
twitter: mikewcohn  
(720) 890-6110



© Copyright Mountain Goat Software®