

System Development and Project Organization (BSUP)  
*Paolo Tell*

# **Agile estimating and Advanced topics in agile planning**



# Outline

- Literature
  - n/a
- Recap
- Agile estimating
- Advanced topics in agile planning

# Recap

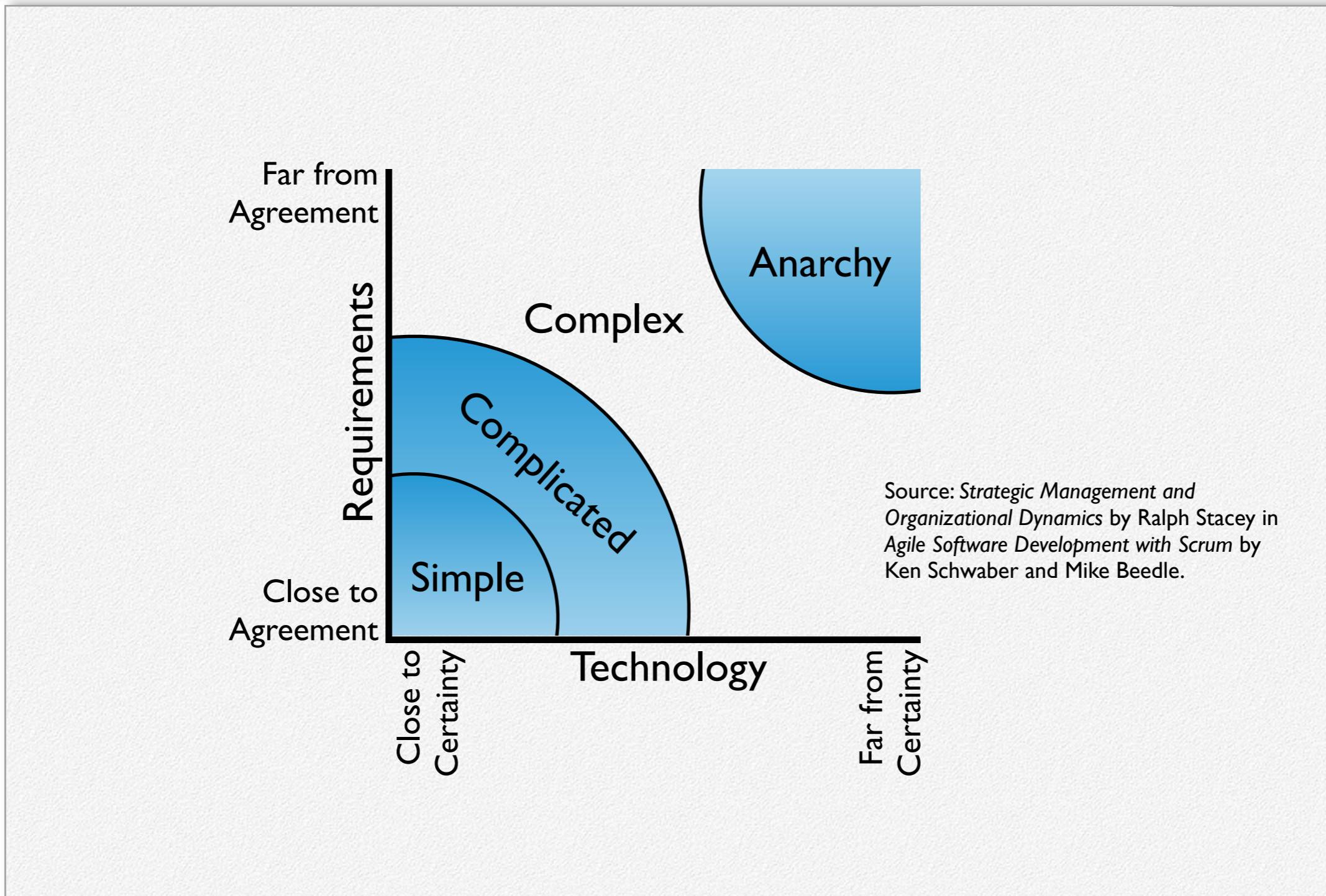
# We are loosing the relay race

“The... ‘relay race’ approach to product development...may conflict with the goals of maximum speed and flexibility. Instead a holistic or ‘rugby’ approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today’s competitive requirements.”

Hirotaka Takeuchi and Ikujiro Nonaka, “The New New Product Development Game”,  
*Harvard Business Review*, January 1986.



# Project noise level



# Scrum

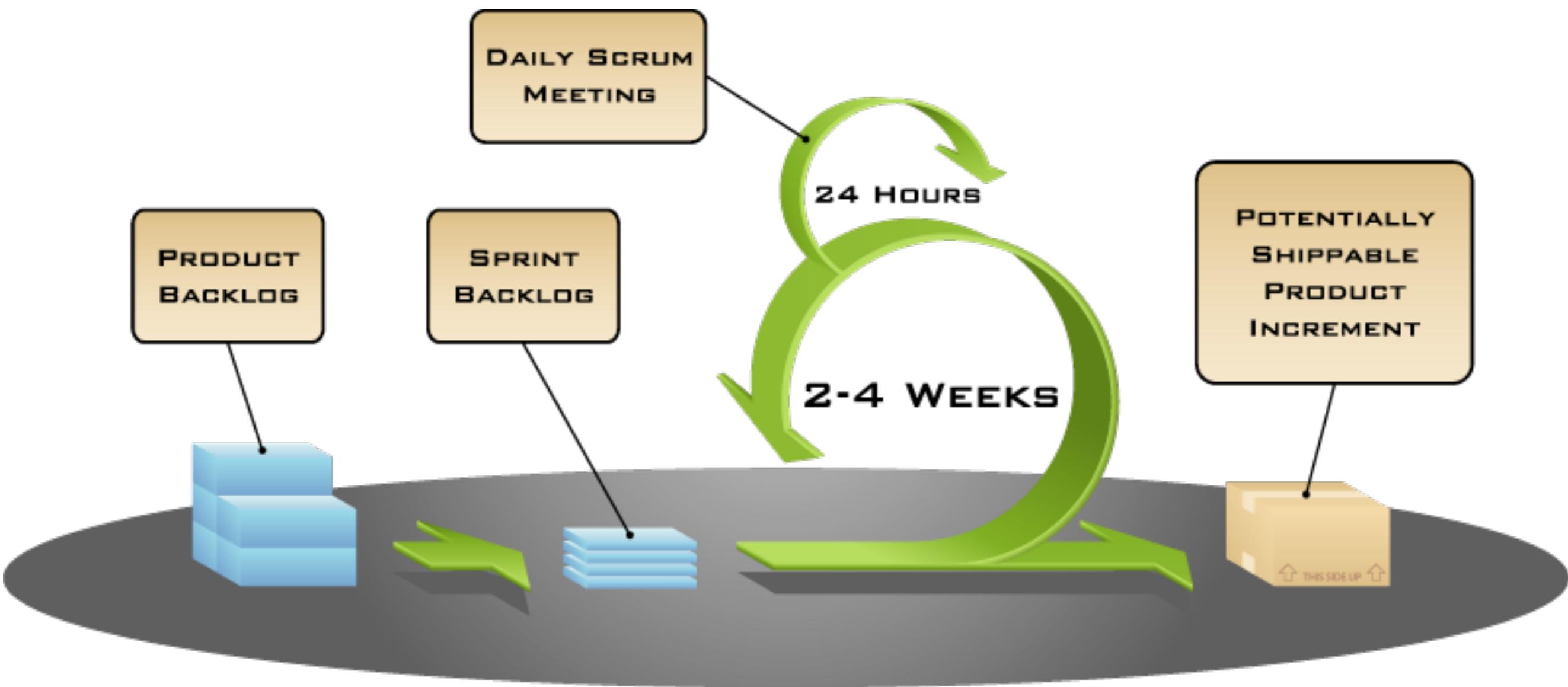


Image available at  
[www.mountaingoatsoftware.com/scrum](http://www.mountaingoatsoftware.com/scrum)

COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

# Scrum framework

## Roles

- Product owner
- ScrumMaster
- Team

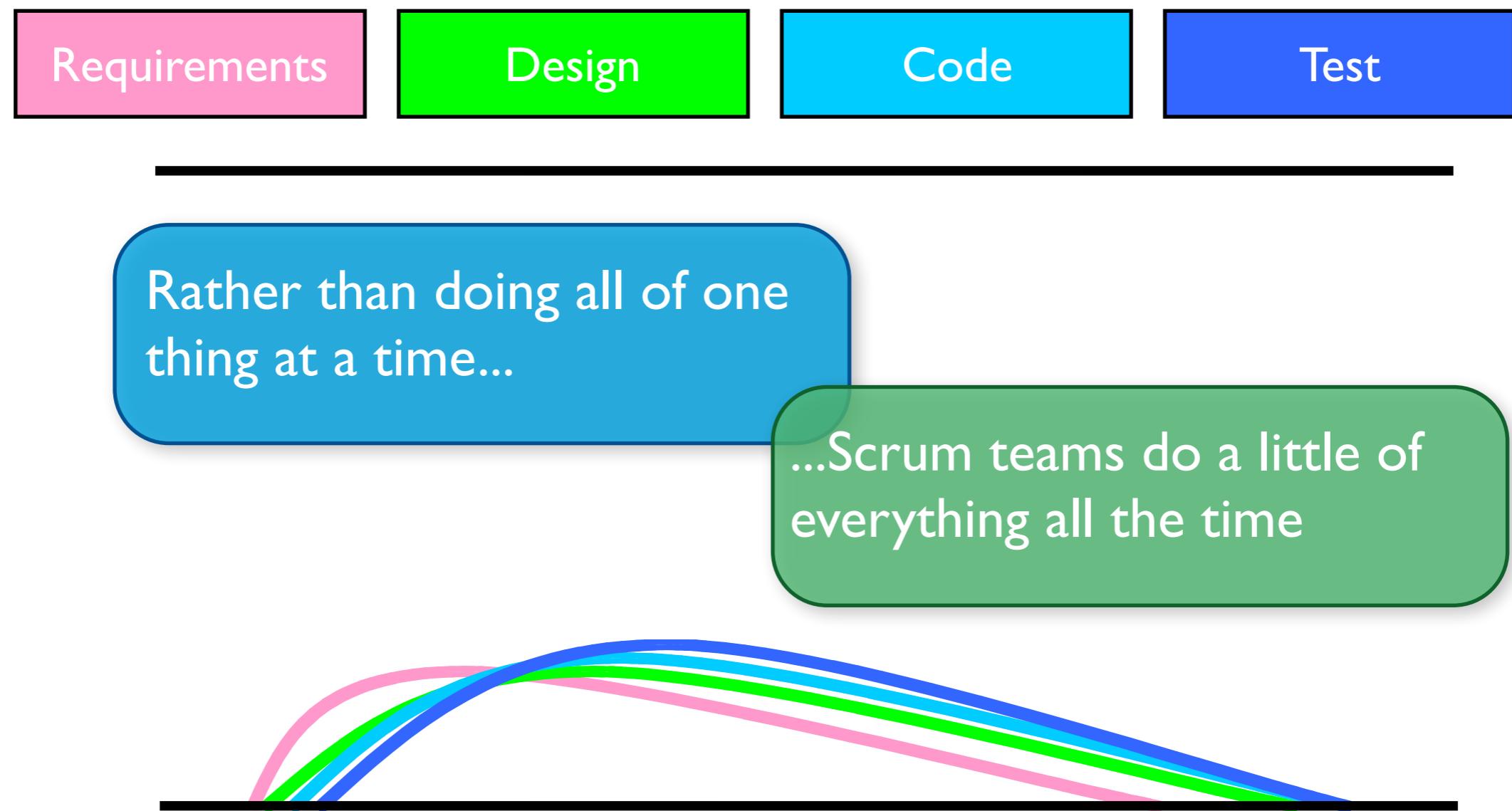
## Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

## Artifacts

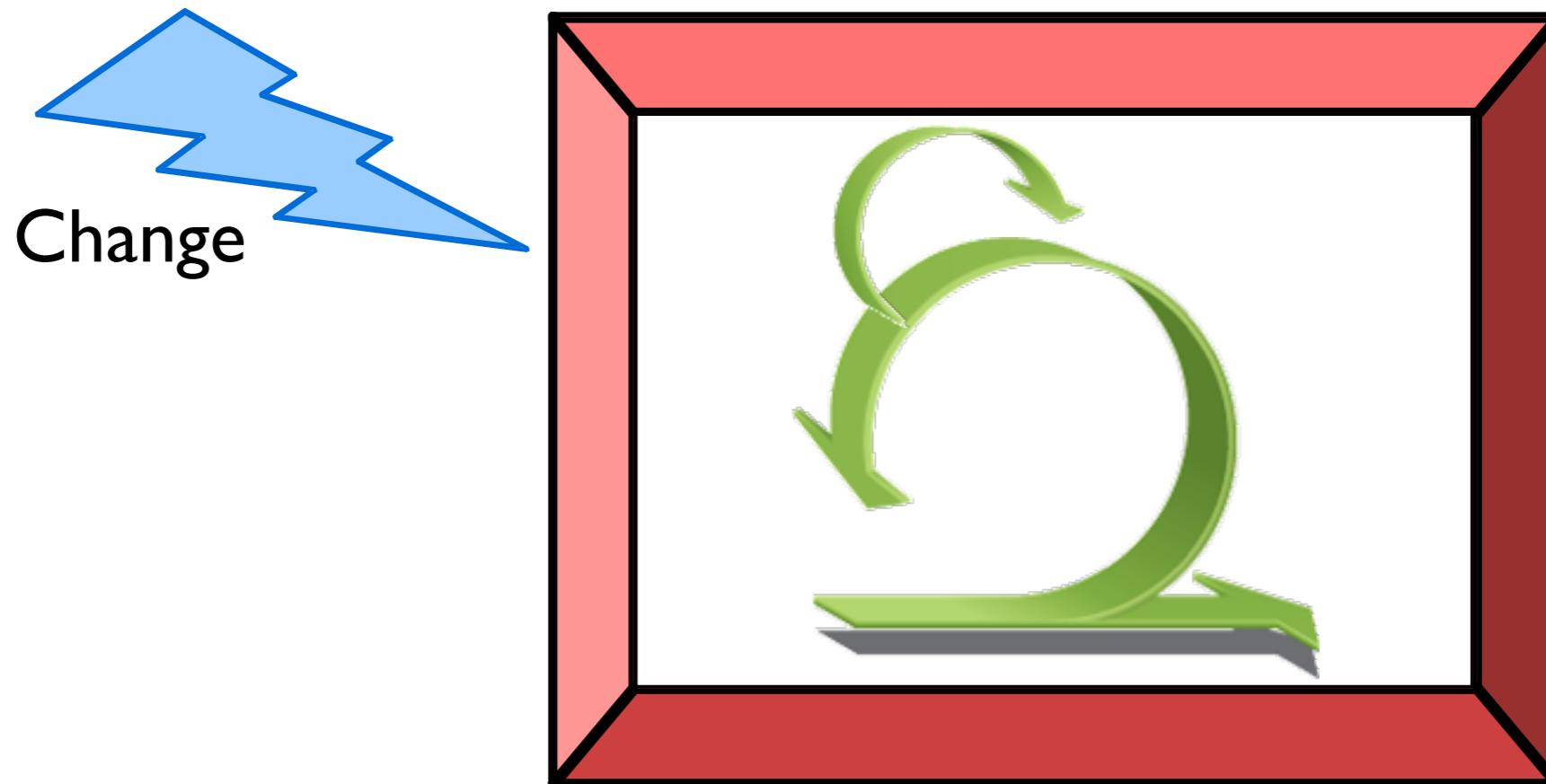
- Product backlog
- Sprint backlog
- Burndown charts

# Sequential versus overlapping development



Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review*, January 1986.

# No changes during a sprint



- Plan sprint durations around how long you can commit to keeping change out of the sprint

# A user story

“As a <user role>, I  
<want/need/can/etc> <goal>  
so that <reason>.”

# How do we deal with details

As a user, I want cancel a reservation.

- Verify that a premium member can cancel the same day without a fee.
- Verify that a non-premium member is charged 5% for the same-day cancellation.
- Verify that a confirmation email is sent.
- Verify that the hotel is notify of any cancellation.

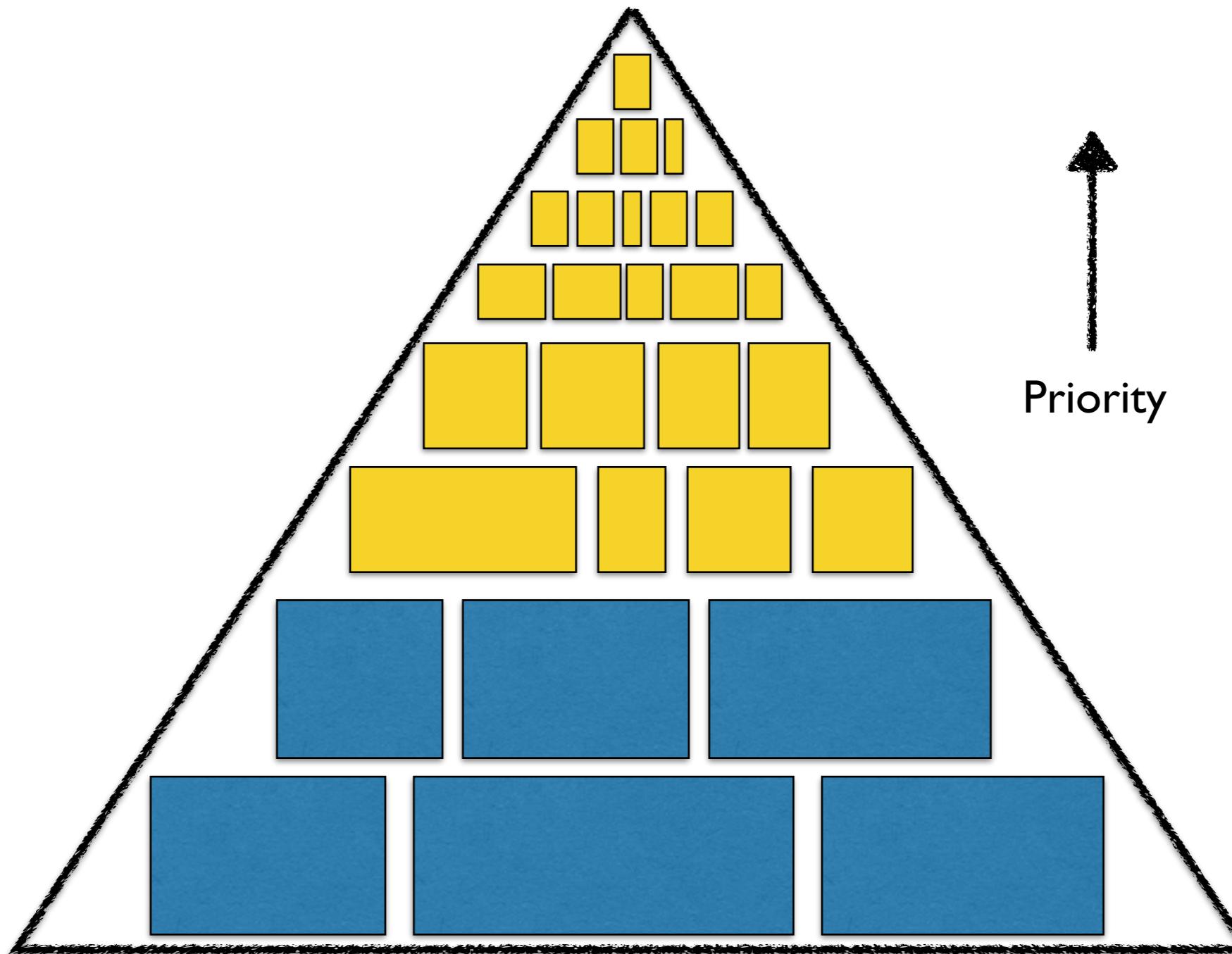
As a premium member, I can cancel a reservation up to the last minute.

As a non-premium member, I can cancel but I am charged 5%.

As a hotel, I am emailed a notification of any cancelled reservation.



# The product backlog iceberg



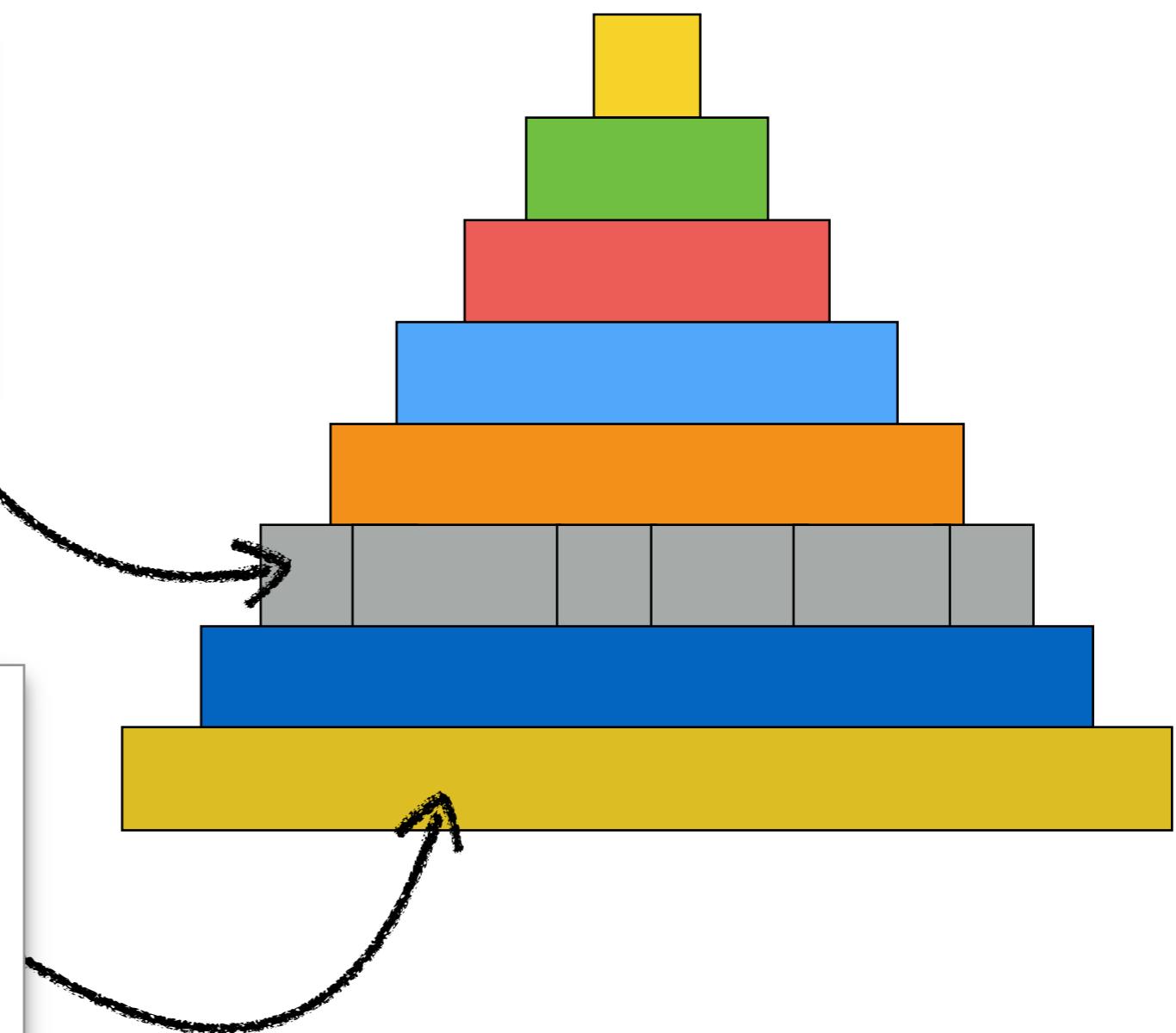
# Some additional useful terms

## Theme

- A collection of related user stories

## Epic/Saga

- A large user story



**Focus**

• ..

**Stories**

• ..

# User stories workshop [30']



# Agile estimating



Mountain Goat Software, LLC

# Agenda

- What is agile planning?
  - Story points
  - Ideal time
  - Estimating with Planning Poker

## Product Backlog

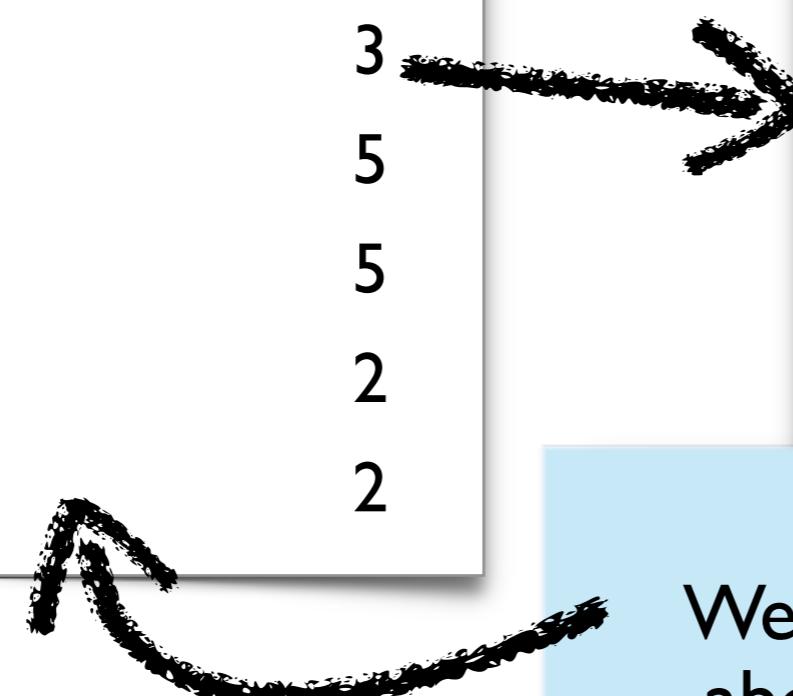
- As a user ... 3
- As a user ... 5
- As a user ... 5
- As a user ... 2
- As a user ... 2

## Sprint Backlog

- Code the UI 12
- Write tests 8
- Code middle tier 4
- Update docs 6

est

We're talking  
about these  
right now



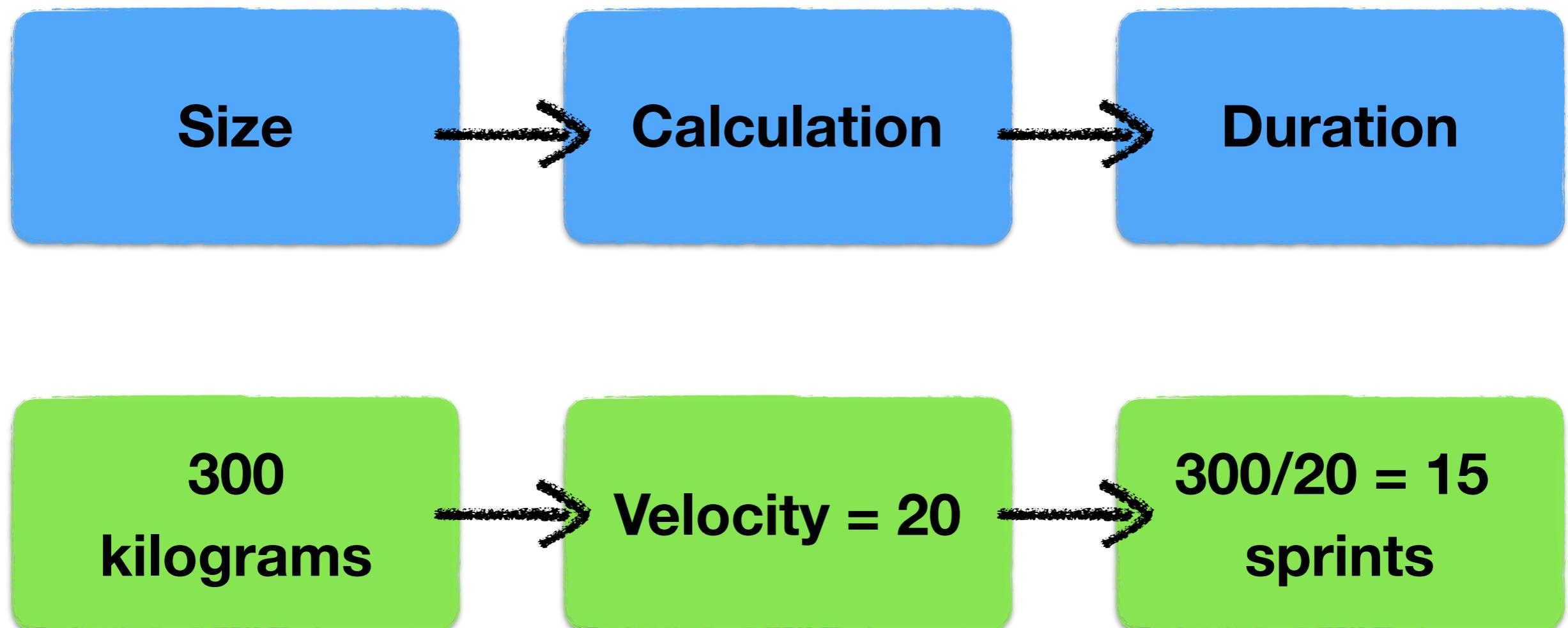
# How long will it take ... [10']

... to drive  
to Paris?

... to read  
the last Harry  
Potter book?



# Estimate size; derive duration



# Measures of size

Traditional  
measures of size

—  
Lines of Code  
Function Points

Agile measures  
of size

—  
Story point  
Ideal days



# 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
  - $5+5 = 10$

# Estimating in zoo points [10']

Lion

Kangaroo

Rhinoceros

Bear

Giraffe

Gorilla

Hippopotamus

Tiger

Blue whale



# Agenda

- What is agile planning?
  - Story points
  - Ideal time
- Estimating with Planning Poker

# Ideal time

How long something will take if:

- it's all you work on
- no one interrupts you
- and everything you need is available

# Ideal time

How long something will take if:

- it's all you work on
- no one interrupts you
- and everything you need is available



# Elapsed time versus ideal time

## Ideally

- Monday has 8 hours
- Each week has 40 hours

“How long will this take?”

- Are you answering what is being asked?

## But instead

Each day has something like:

- 2 hours of meetings
- 2 hours of email
- 4 hours left for the project



# Two big advantages to story points

1. Story points are additive; time-based estimate may not be
2. Story points help avoid problems with unit confusion

# Confusing units

## Product Backlog

- As a user ... 30
- As a user ... 50
- As a user ... 50
- As a user ... 20
- As a user ... 20

## Sprint Backlog

- Code the UI 12
- Write tests 8
- Code middle tier 4
- Update docs 6
- Automate tests 5



# Confusing units

## Product Backlog

- As a user ... 3
- As a user ... 5
- As a user ... 5
- As a user ... 2
- As a user ... 2



## Sprint Backlog

- Code the UI 12
- Write tests 8
- Code middle tier 4
- Update docs 6
- Automate tests 5

# Agenda

- What is agile planning?
  - Story points
  - Ideal time
- Estimating with Planning Poker

# 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 is discussed briefly
  - Each estimator selects a card that is his or her estimate
  - Cards are turned at the same time
  - Discuss differences (especially outliers)
  - Re-estimate until estimates converge

# Planning Poker — an example



Estimator	Round 1	Round 2
Ana	5	8
Trond	5	8
Maria	8	8
Johannes	20	13

# Estimate these [20']

Product Backlog Items	Estimate
Read (and understand) a high-level, 10-page overview of agile software development in a celebrity news magazine.	
Read (and understand) a densely written 5-pages research paper about agile software development in an academic journal.	
Your uncle owns a clock store and wants to sell clocks over the internet. Write a basic product backlog for him covering what he will need his website to do.	
Recruit, interview, and hire a new member of your team.	
Create a 60' presentation about agile software development for your coworkers.	
Wash and wax your boss' Porsche.	
Read (and understand) a 150-pages book on agile software development.	
Write a 5-page summary of these first three lectures on Scrum especially focusing on what you learned.	

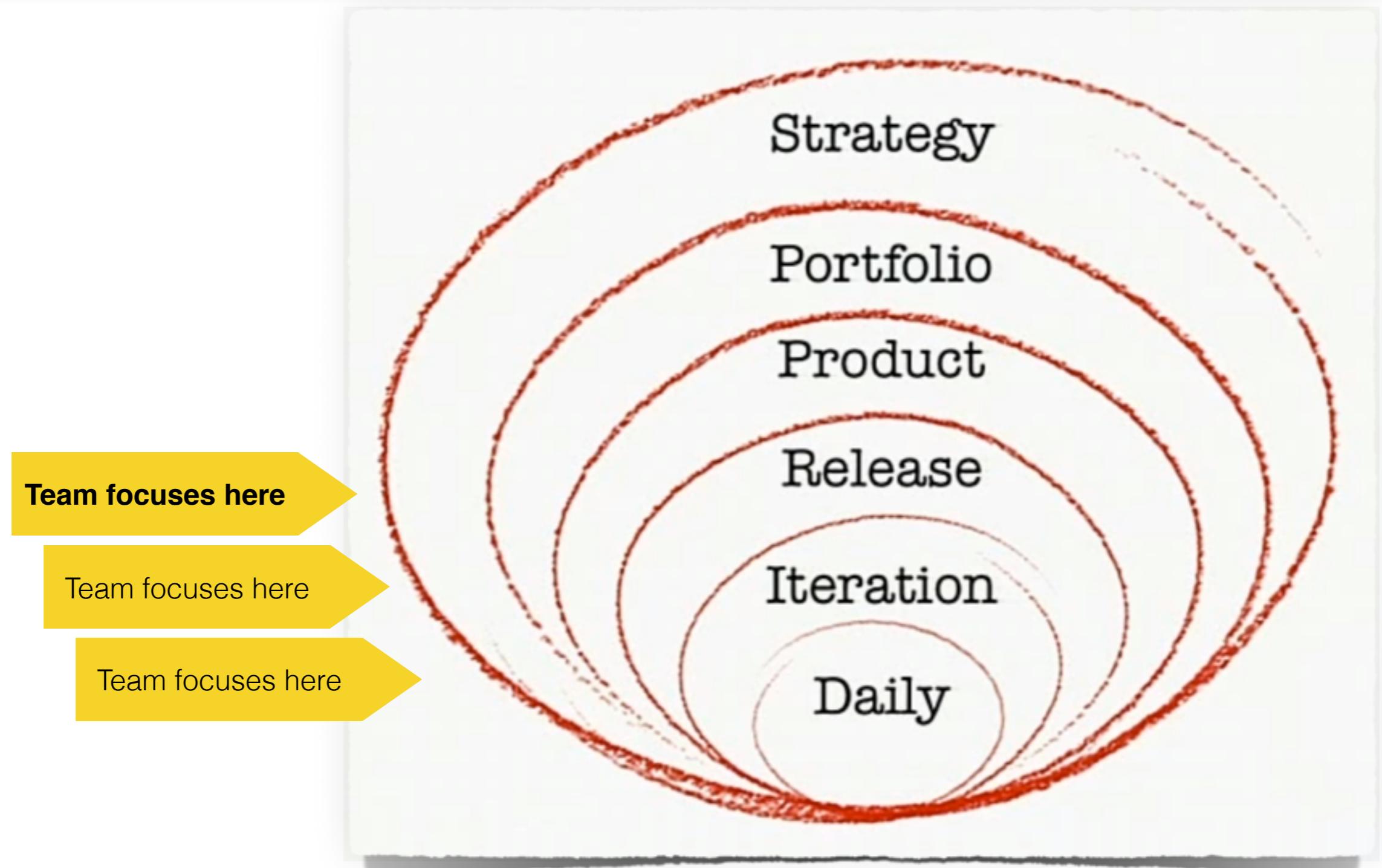


# Advanced Agile Planning



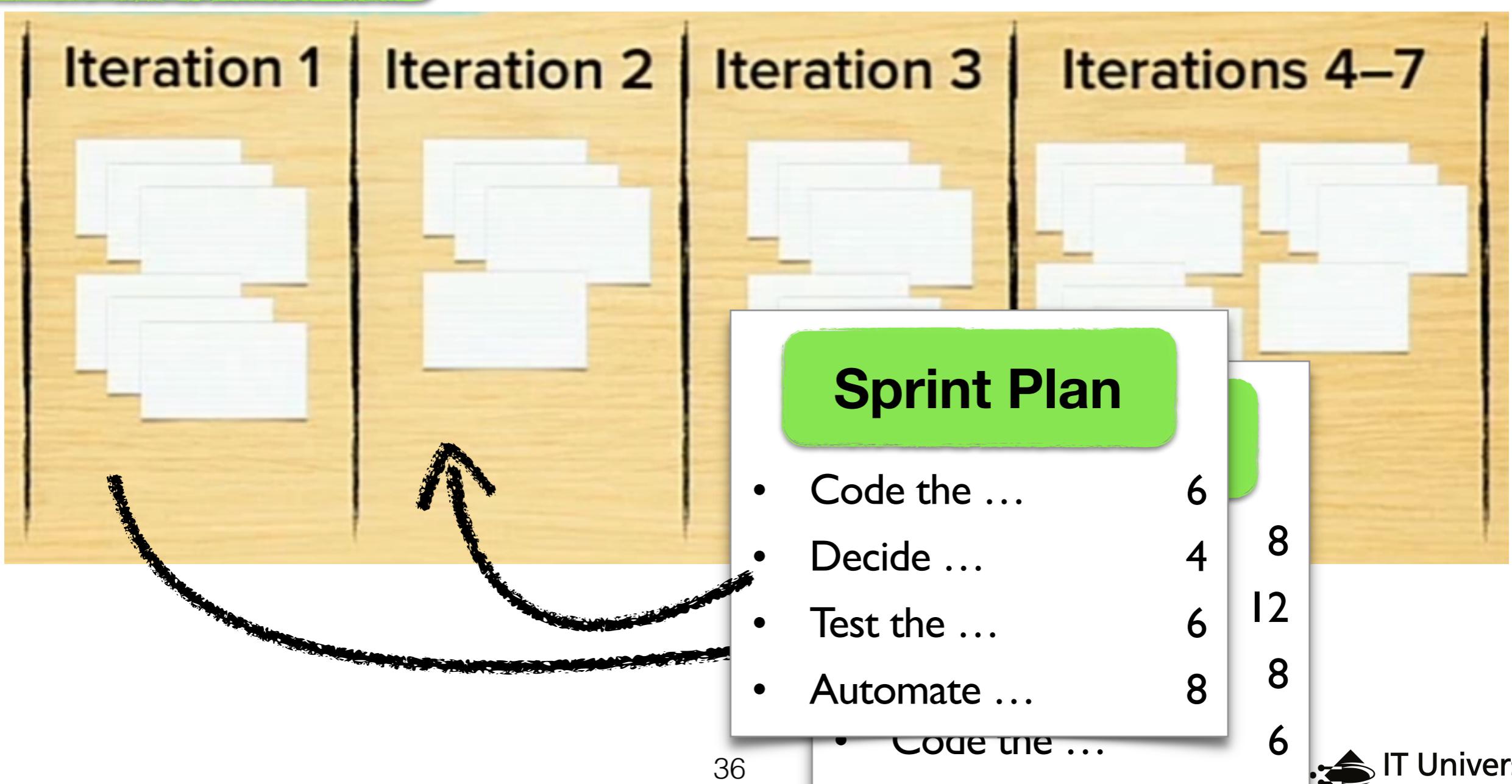
Mountain Goat Software, LLC

# The planning onion



# Release and sprint planning

## Release Plan



# What is a good plan?

- A good plan is one that supports reliable decision-making
- Will go from
  - We will be done in the forth quarter
  - We will be done in December
  - We will be done the 18th of December

“It’s better to be roughly right than precisely wrong.”

—J.M. Keynes

# Velocity

A Sprint

4

10

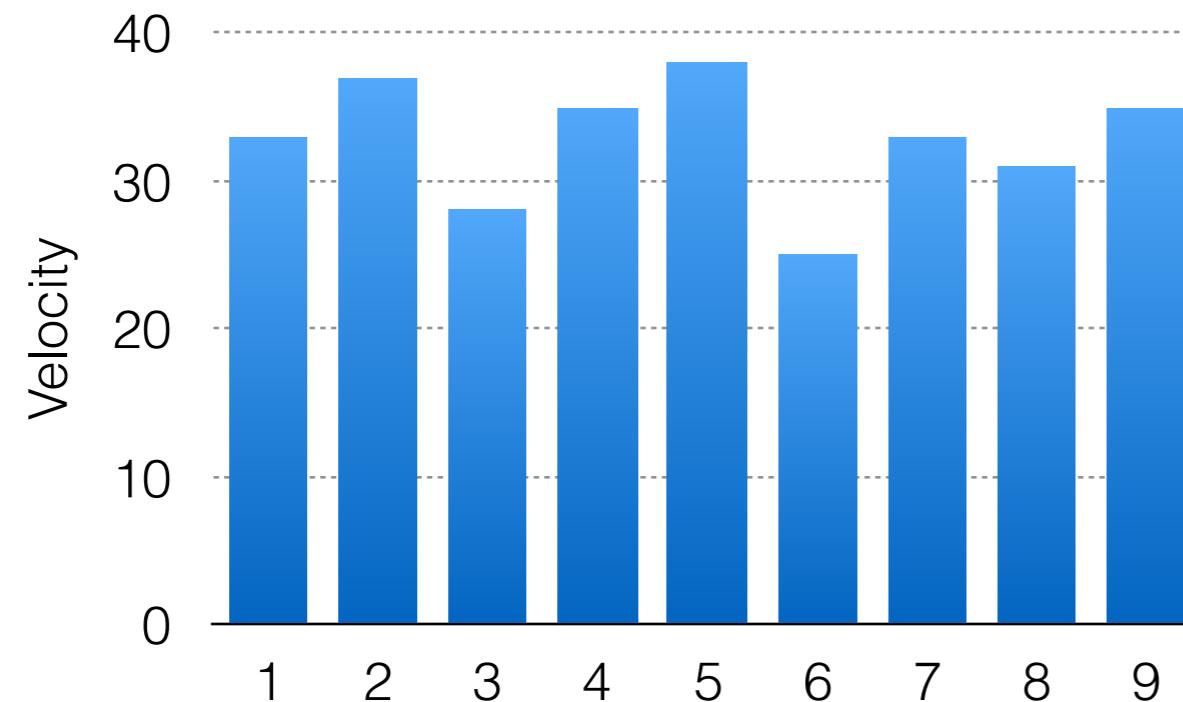
1

Velocity = 15

3



# Velocity



Velocity is measured in the units you use to estimate product backlog items.

- A useful long-term measure of the amount of work completed per iteration
- Most useful over at least a handful of iterations

# Five planning scenarios

- A team with historical data
- Fixed-date plans
- Fixed-scope plans
- A team with no velocity data
- A team changing size

# Calculate a confidence interval from historical data

27
34
35
38
39
40
40
41
45



90% confidence  
interval

Sorted  
Velocities

# of historical sprints	Sprints to throw out from each end
0-7	0
8-10	1
11-12	2
13-15	3
16-17	4
18-20	5
21-22	6
23-25	7
26+	8

# Extrapolate from the velocity range



We will almost certainly get here ( $5 * 34 = 170$ )

The most we can realistically expect ( $5 \times 41 = 205$ )

Assume there  
are 5 sprints  
left



### FREE AGILE PROJECT MANAGEMENT TOOLS



Velocity Range  
Calculator



Relative Weighting



Theme Screening



Theme Scoring



Planning Poker



Project Success Sliders

## Velocity Range Calculator

The velocity range calculator below is one of the free agile tools available from Mountain Goat Software. This agile project management tool is used to predict how much work a team will complete during a planned number of upcoming iterations, we are better off considering velocity as a range rather than a specific

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

five iterations. Enter them in the field at right in any order (lowest to highest, random, most recent first, etc.). You can optionally indicate how many iterations remain in the project. This number will be used to tell you how much work can likely be completed in that time.

Velocities From Completed Iterations

velocity numbers, comma-separated

Planned Iterations (optional)

**CALCULATE**

# Five planning scenarios

- A team with historical data
- Fixed-date plans
- Fixed-scope plans
- A team with no velocity data
- A team changing size

# Fixed-date planning

## Three steps

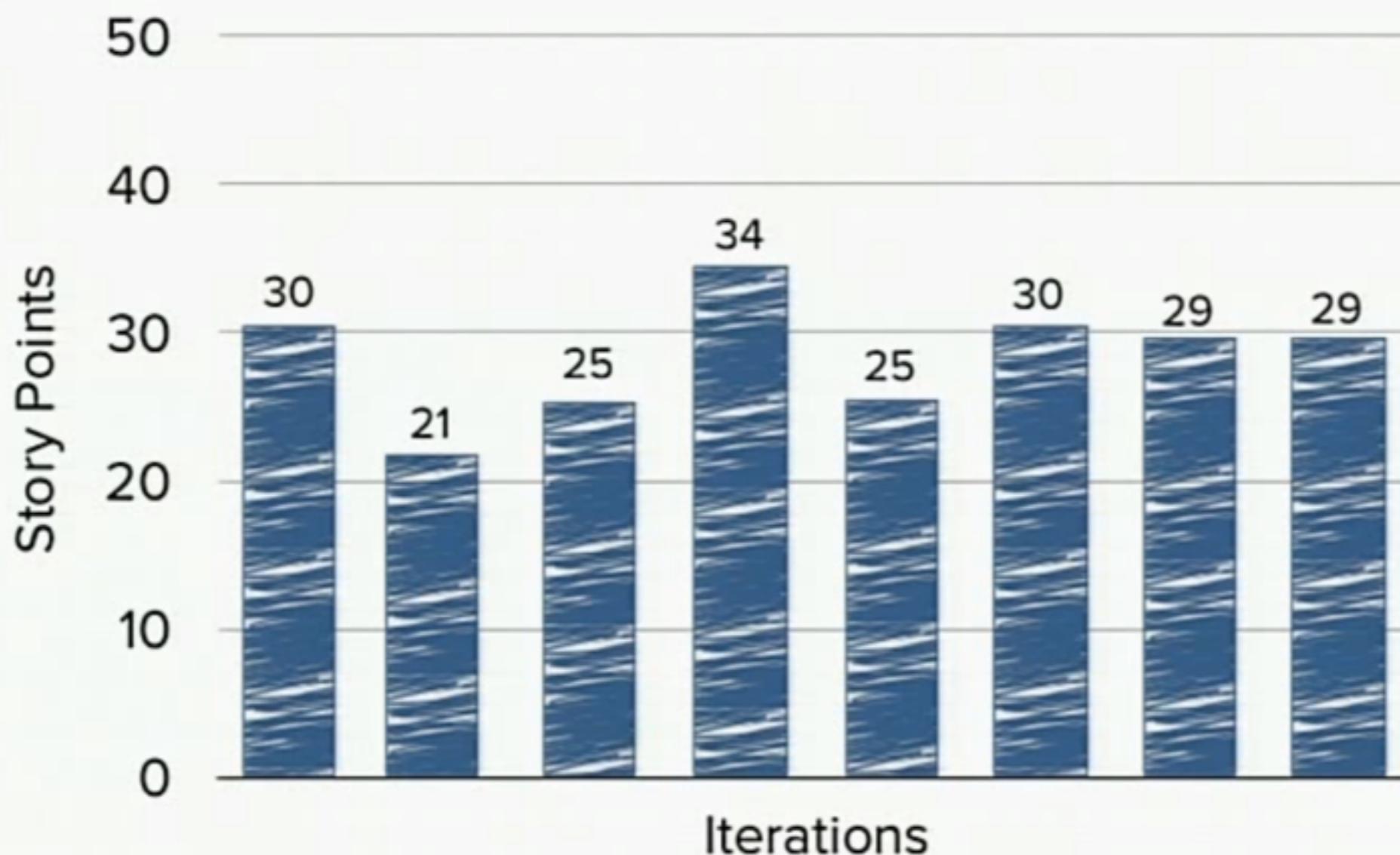
1. Determine how many sprints you have.
2. Estimate velocity as a range.
3. Use that range \* the number of sprints to partition the backlog into “will have”, “might have”, and “wont’ have”.



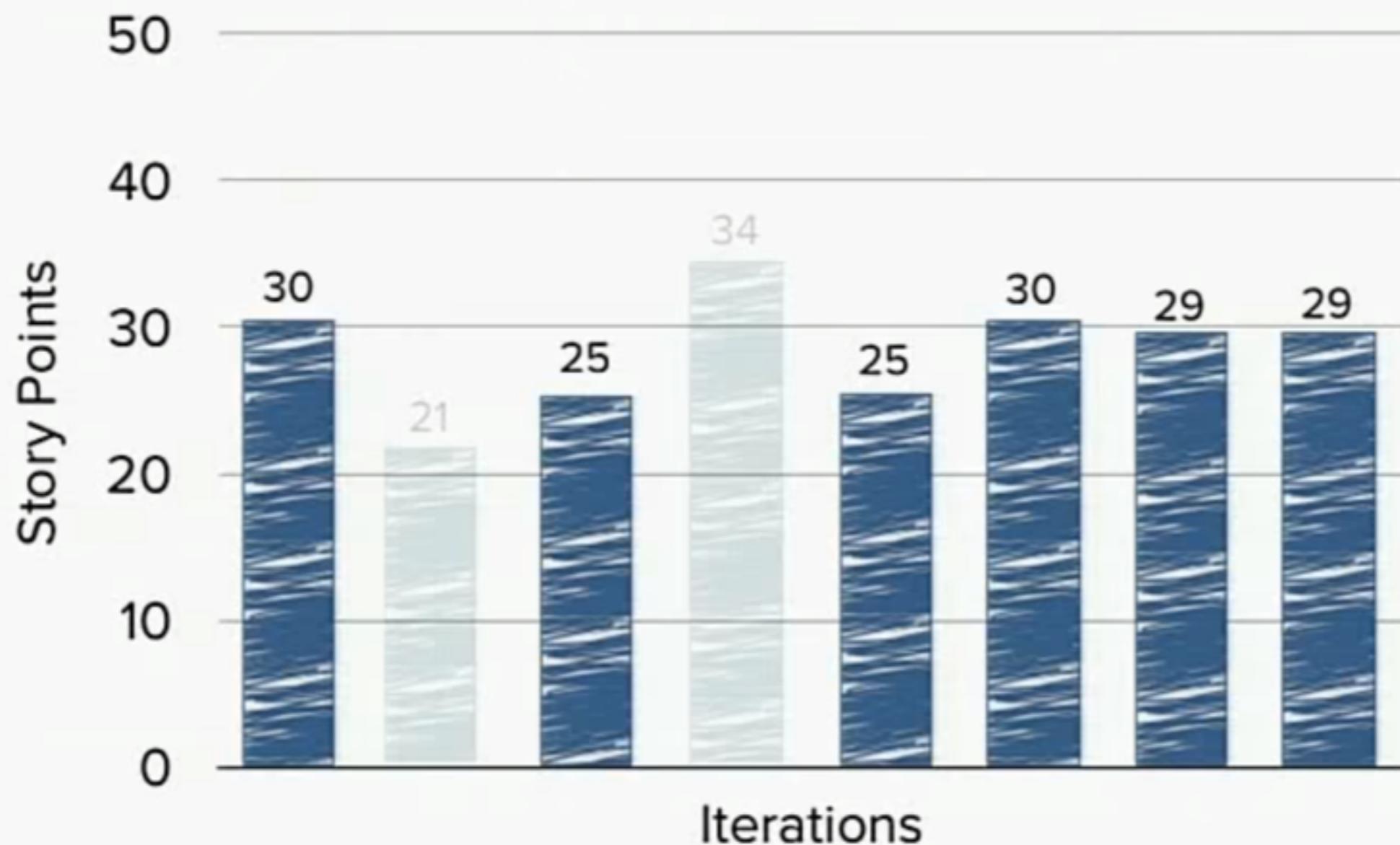
# Count the sprints



# Determine a velocity range



# Determine a velocity range



# Determine what to commit to

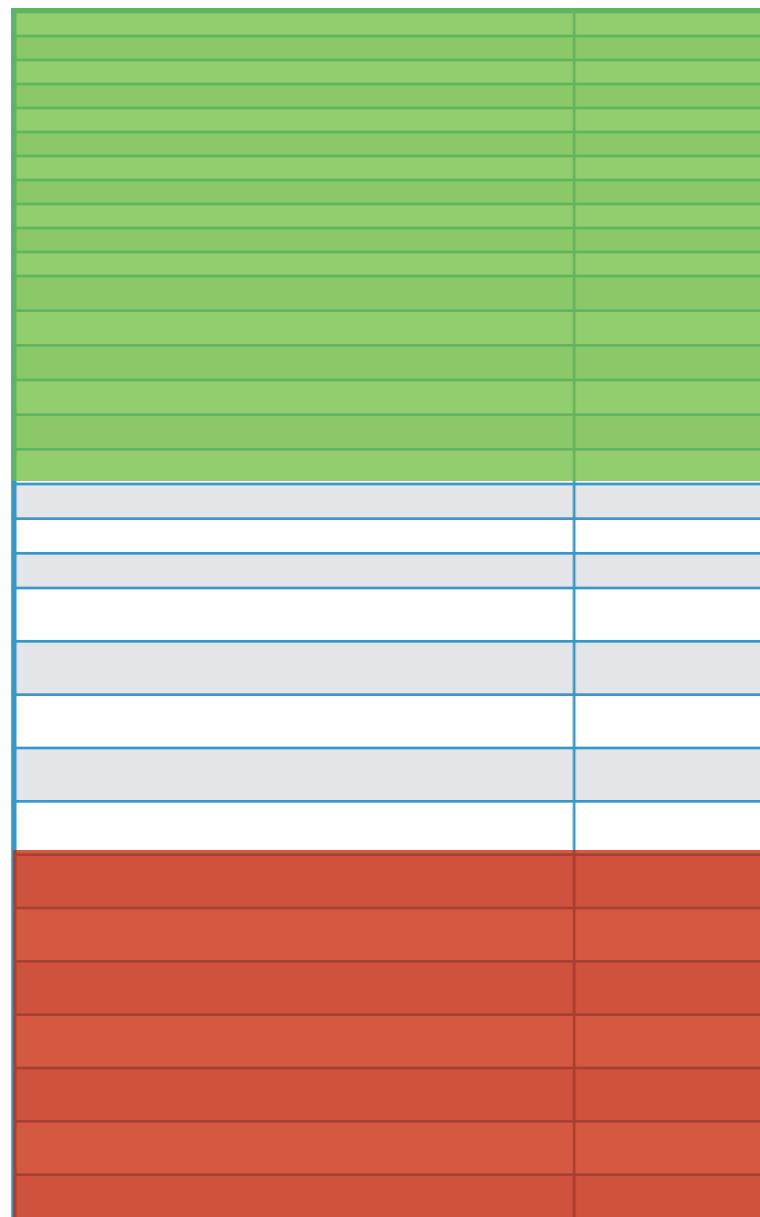
Will have

6\*25 ➔

Might have

6\*30 ➔

Won't have



Product Backlog

If you promise this

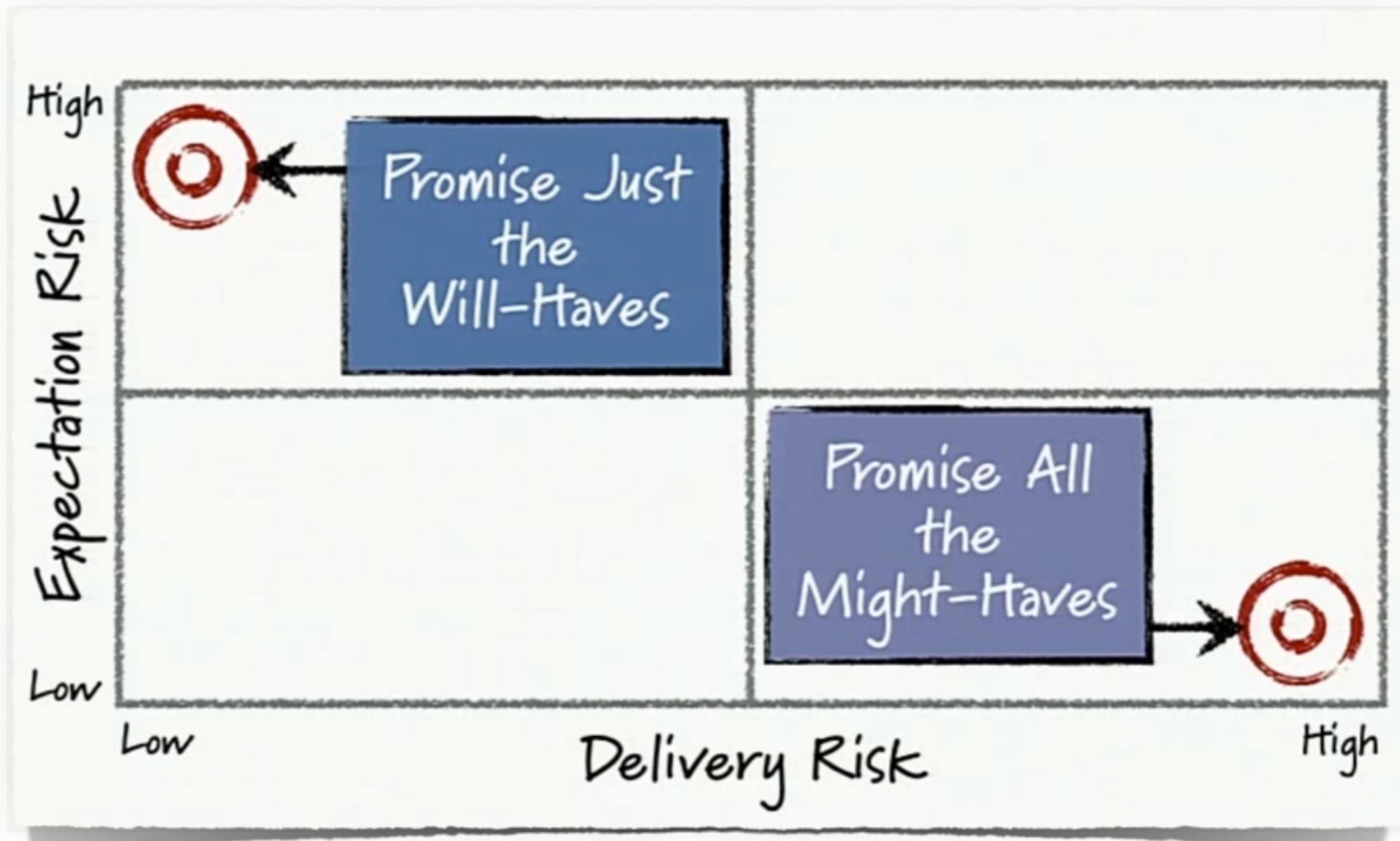
- You probably won't get the contract
- But will probably finish everything if you do

If you promise this

- You will probably win the contract
- But probably will not finish everything in time



# Balancing risk



# Five planning scenarios

- A team with historical data
- Fixed-date plans
- Fixed-scope plans
- A team with no velocity data
- A team changing size

# Fixed-scope planning

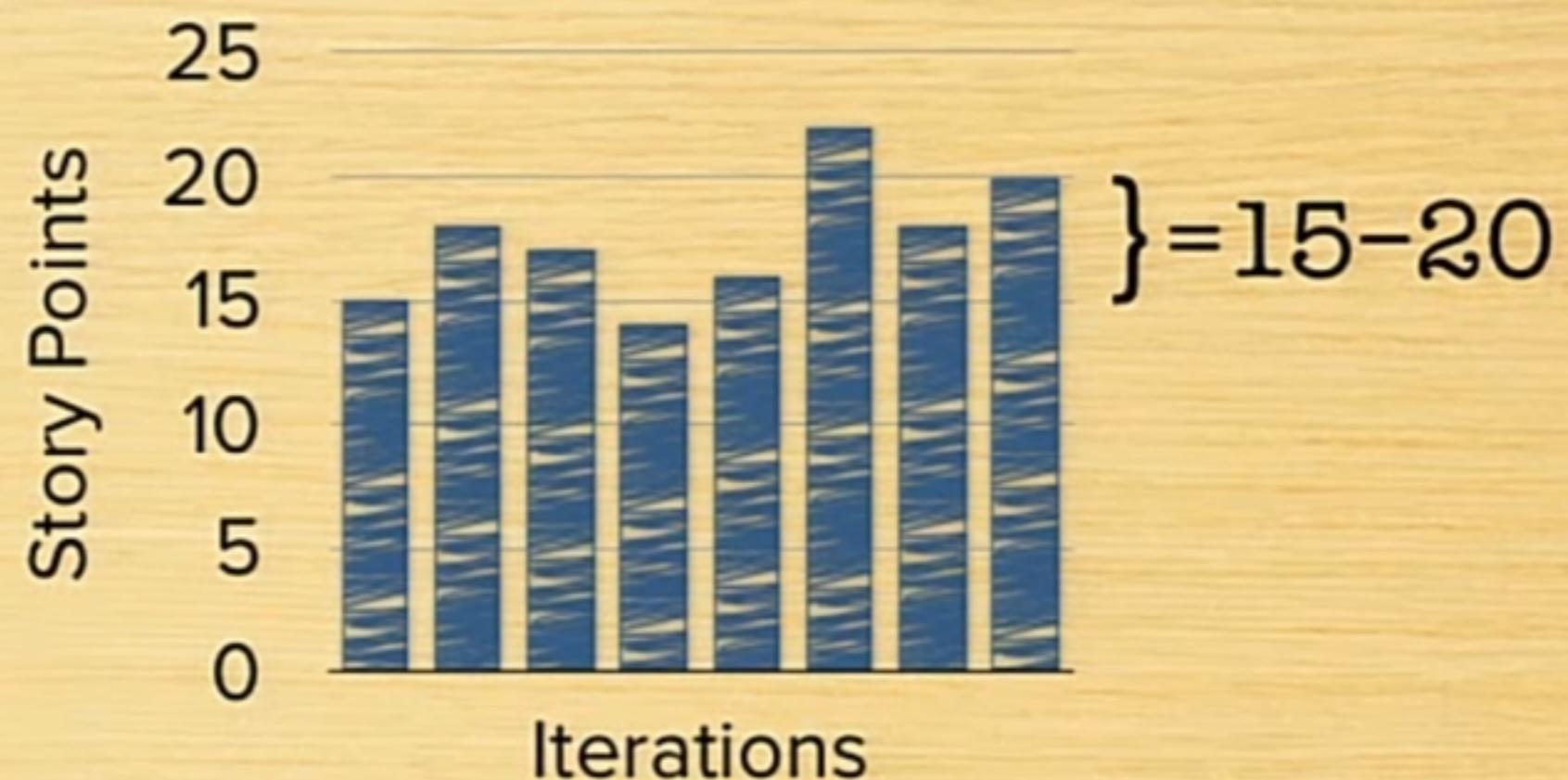
## Three steps

1. Sum the product backlog items.
2. Estimate velocity as a range.
3. Use that sum of the backlog divided by the velocity range to determine a date range.





= 120 story points



## If you promise the short duration

- You probably get the contract
- But may not finish everything in time

$$120 \div 20 =$$



$$120 \div 15 =$$



## If you promise the long duration

- You probably will not get the contract
- But it should be easy to finish everything



# Ranges

- Notice in both cases we had a range
- For a fixed-date project, use a scope range:
  - “By that date you will have all of these features and some of these.”
- For a fixed-scope project, use a date range:
  - “It will take use between 6 and 8 sprints to deliver all those features.”

# The impending tradeshow [15']

Your company develops tools for managing agile projects.

You have finished version 1.0 (on time, of course). Now the boss needs a 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?

# The impending tradeshow [15']

Your company develops tools for managing agile projects.

You have finished version 1.0 (on time, of course). Now the boss needs a 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	Velocity
1	20
2	14
3	23
4	18
5	25
6	30
7	12
8	22
9	15
10	23

## Your estimates

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



# Five planning scenarios

- A team with historical data
- Fixed-date plans
- Fixed-scope plans
- A team with no velocity data
- A team changing size

# Forecast an initial velocity

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

# Consider this team

Person	Hours/Day	Hours / Iteration
Sergey	4–6	40–60
Yuri	4–6	40–60
Carina	2–3	20–30
Total		100–150

# Establishing their velocity

## Capacity

100-150 hours per sprint

Code ...	12
Design ...	6
Test ...	8
Decide ...	8
Automate ...	12

46

Story	Points
As a frequent flyer ...	3
As a visitor ...	5
As a vacation planner ...	5
As a frequent flyer ...	2

3

5

5

2

Code ...	8
Test ...	6
Design ...	12
Test ...	5

31

48

22	22
----	----

22

# Turn the point estimate into a range

- If you do not have historical data
  - Take a wild guess, perhaps:
    - +/- 10% for a known team working together in a known application 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

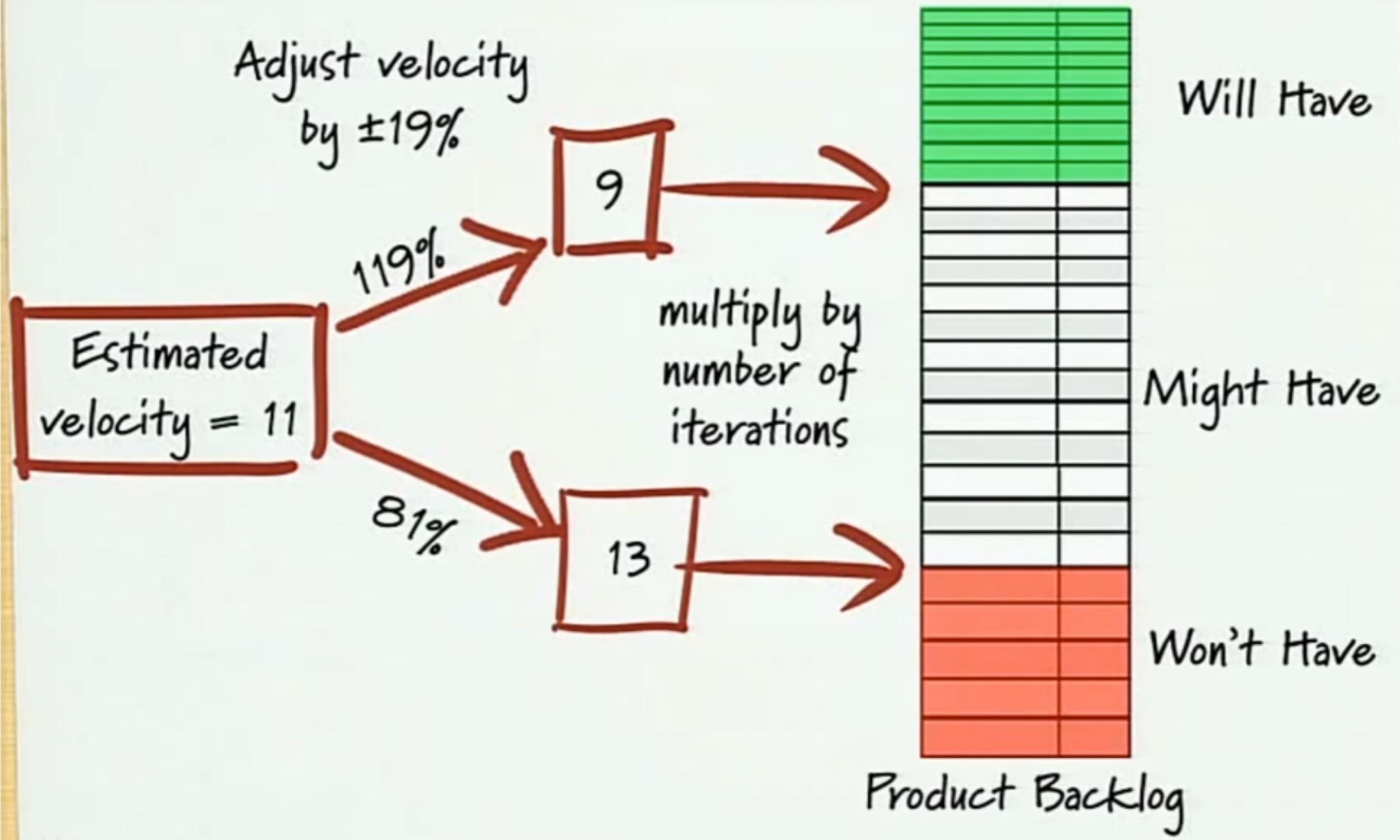


# Using data from other teams

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

Team A	
Mean	Standart deviation
22	3.8

Relative standard deviation  
 $3.8 / 22 = 17\%$



# Five planning scenarios

- A team with historical data
- Fixed-date plans
- Fixed-scope plans
- A team with no velocity data
- A team changing size

# 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 organisation.	

# Impact of going from 6 to 7 people

Initial Team Size	New Team Size	Sprint +1	Sprint +2	Sprint +3	Iteration	Average Velocity Change
6	7	-20%	-4%	+12%	1	-10%
6	7	0%	-6%	+15%	2	-5%
7	5	-12%	-8%	-8%	3	+13%
...	...	...	...	...		

# Concluding

# Outline

- Literature
  - n/a
- Recap
- Agile estimating
- Advanced topics in agile planning
- Next time: we will start with the book