Agile Estimation and Planning

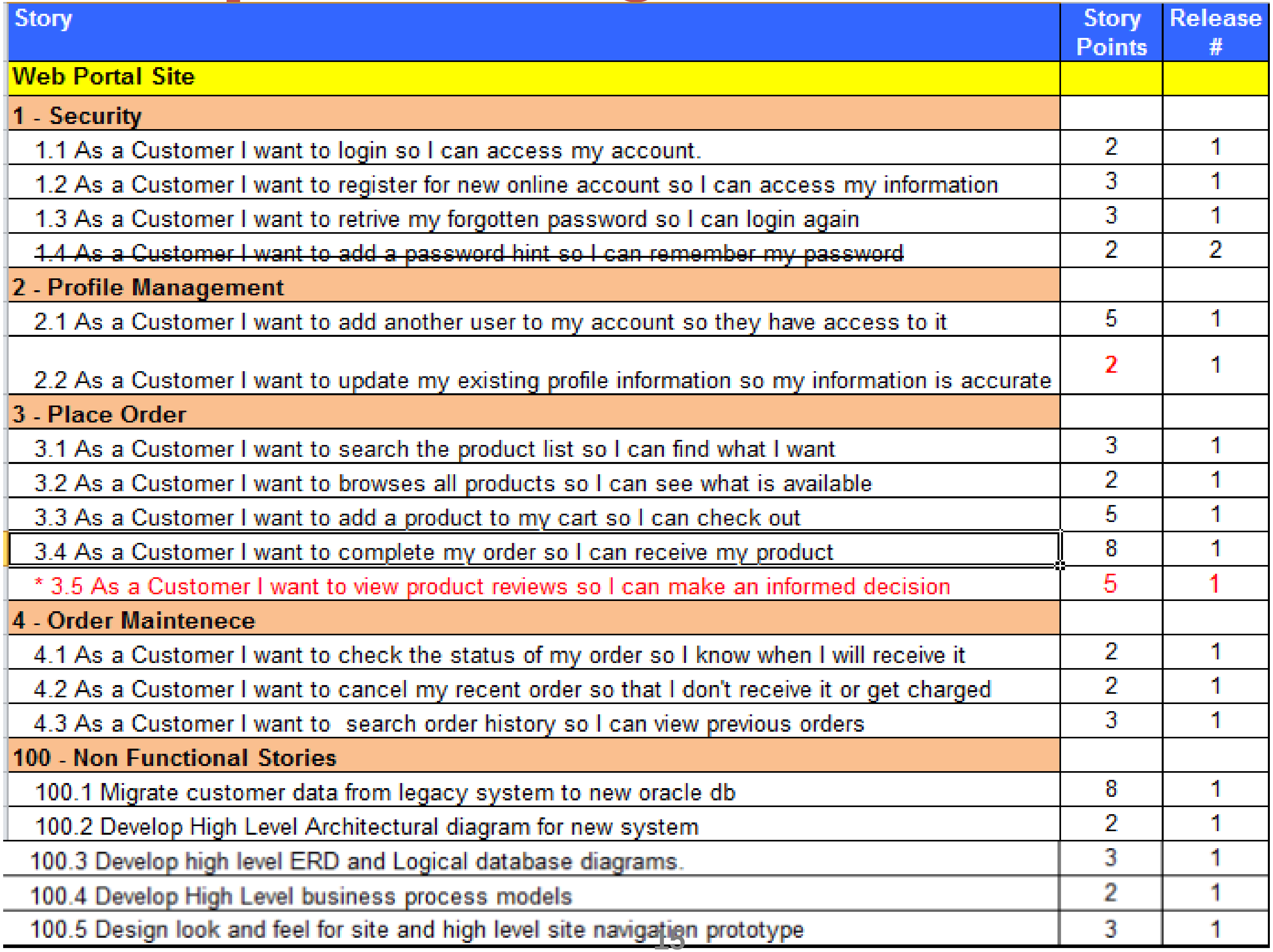
Based on Sally Elatta's lecture
(Agile Transformation Inc.)

Agile Planning

We will now cover how Agile addresses planning and estimation.

Please keep an open mind ☺

Example Backlog

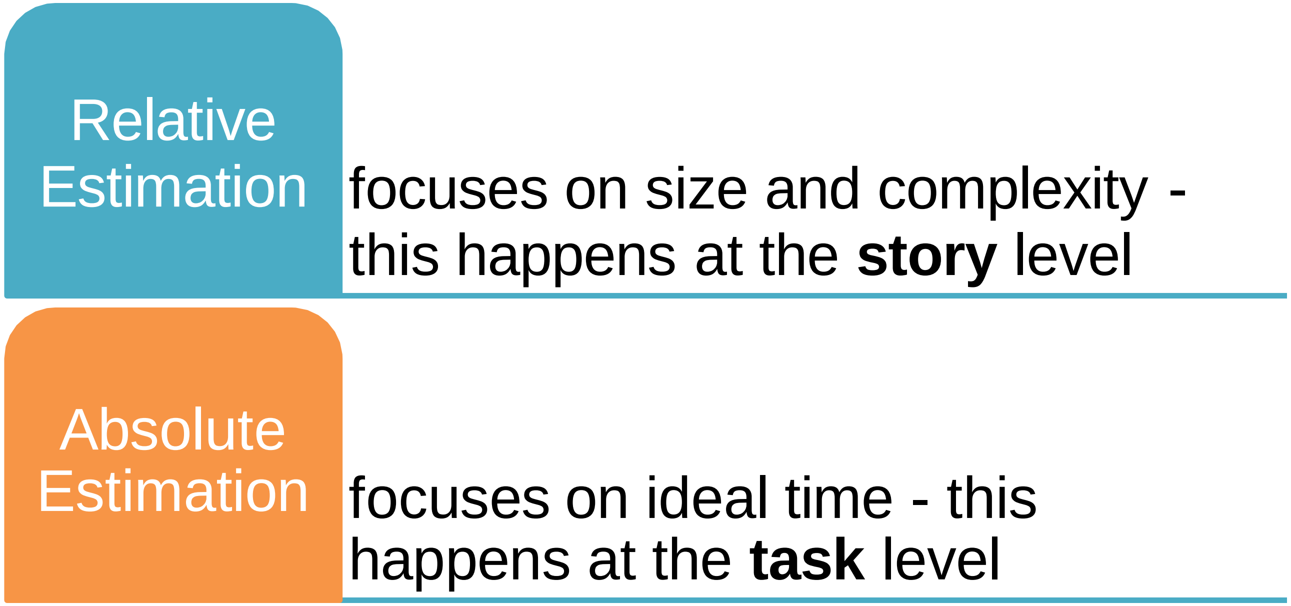


Story	Story Points	Release #
Web Portal Site		
1 - Security		
1.1 As a Customer I want to login so I can access my account.	2	1
1.2 As a Customer I want to register for new online account so I can access my information	3	1
1.3 As a Customer I want to retrive my forgotten password so I can login again	3	1
~~1.4 As a Customer I want to add a password hint so I can remember my password~~	2	2
2 - Profile Management		
2.1 As a Customer I want to add another user to my account so they have access to it	5	1
2.2 As a Customer I want to update my existing profile information so my information is accurate	2	1
3 - Place Order		
3.1 As a Customer I want to search the product list so I can find what I want	3	1
3.2 As a Customer I want to browses all products so I can see what is available	2	1
3.3 As a Customer I want to add a product to my cart so I can check out	5	1
3.4 As a Customer I want to complete my order so I can receive my product	8	1
* 3.5 As a Customer I want to view product reviews so I can make an informed decision	5	1
4 - Order Maintenece		
4.1 As a Customer I want to check the status of my order so I know when I will receive it	2	1
4.2 As a Customer I want to cancel my recent order so that I don't receive it or get charged	2	1
4.3 As a Customer I want to search order history so I can view previous orders	3	1
100 - Non Functional Stories		
100.1 Migrate customer data from legacy system to new oracle db	8	1
100.2 Develop High Level Architectural diagram for new system	2	1
100.3 Develop high level ERD and Logical database diagrams.	3	1
100.4 Develop High Level business process models	2	1
100.5 Design look and feel for site and high level site navigation prototype	3	1

Estimation

“It is better to be roughly right than precisely wrong.”
—John Maynard Keynes

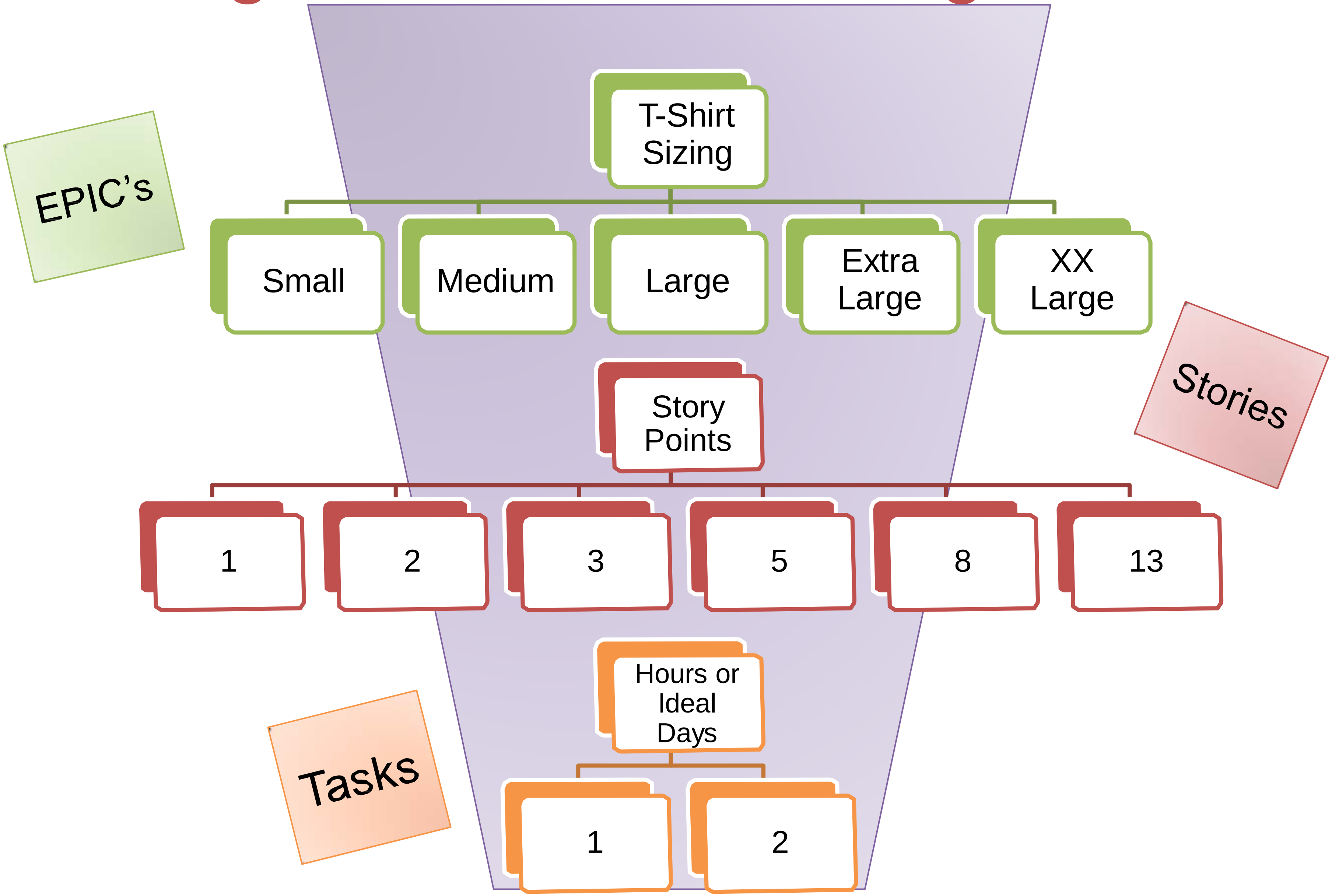
Relative vs. Absolute Estimating



Relative Estimation focuses on size and complexity - this happens at the **story** level

Absolute Estimation focuses on ideal time - this happens at the **task** level

The ‘Right Method’ for the ‘Right Time’



EPIC’s

T-Shirt Sizing

Small

Medium

Large

Extra Large

XX Large

Stories

Story Points

1

2

3

5

8

13

Tasks

Hours or Ideal Days

1

2

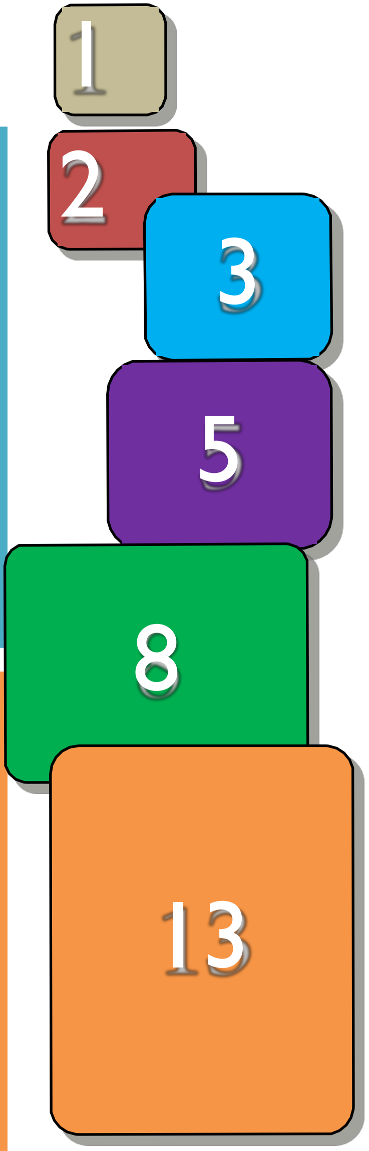
Common Sizing Units

Story Points

- A measure of the relative size and complexity of the story.
- How much effort and how hard is this story compared to others on our backlog?
- Avoids the need (and waste) behind precise estimates.

Ideal Days

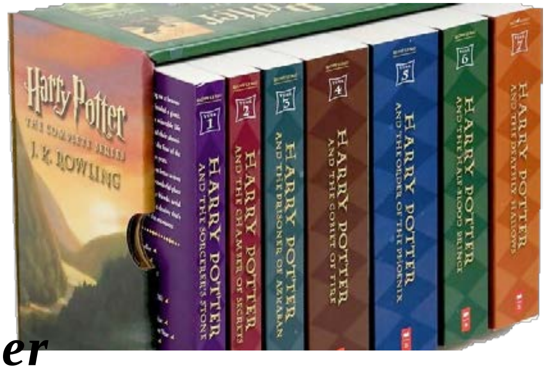
- The amount of time something is likely to take one person if they aren’t disrupted or distracted.
- If two people will work on it, their time is added.
- Often expressed in days (including ½ day, etc.)



Task 1

How long will it take…

…to drive to Adelaide?



…to read the last Harry Potter book?

Task 2

Estimation in zoo points

Lion

Kangaroo

Rhinoceros

Bear

Giraffe

Gorilla

Hippopotamus

Tiger

How To Size Points?

Planning Poker

- Team reviews each story then everyone shows their card with story points

Affinity Sizing

- Draw a line on the wall from smallest to largest
- Team reviews the stories then silently puts them up on the wall relative to each other
- Determine bucket values (points) at the end

Complexity Buckets

- Agree first on what buckets/criteria add complexity
- Review the story through each bucket to determine how complex it is
- Combine the points from the buckets then use consensus to decide final Story Point value to assign

Estimating Story Points Using Complexity Buckets

- *The Elatta Method* -

This approach provides a **consistent way** for teams to size stories by discussing each story in terms of pre-defined buckets of complexity before deciding on the final points. The steps are simple:

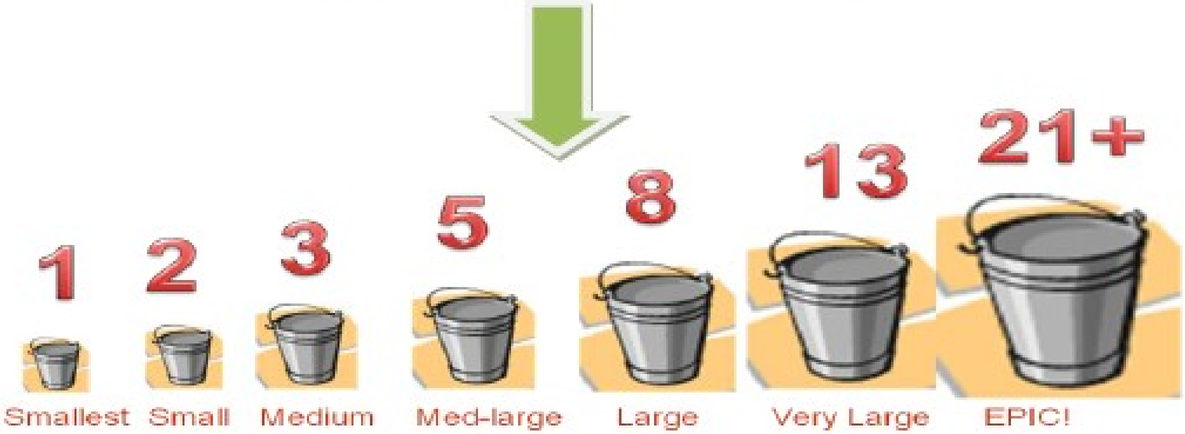
- **Decide on the buckets of complexity** you think match your project. For example, many software development efforts have the buckets used below, but a reporting or BI project could have different ones.
- **Discuss the story in each bucket** and determine if the team can agree if the work it has a **Light**, **Medium**, **High** or Complex level of complexity.
- **Add up the points** and see which Fibonacci Story Point bucket it falls into. If it falls between two buckets, have the team do a gut check and decide on which ones it falls into.

User Interface
L = 1
M = 2
H = 3
C = 4
Helpful Considerations:
- number of screen fields?
- screen validation logic?
- number of screens?

Business Logic
L = 1
M = 2
H = 3
C = 4
Helpful Considerations:
- number of business rules
- BR complexity

Data / Integration
L = 1
M = 2
H = 3
C = 4
Helpful Considerations:
- number of data stores
- complexity of StoredProc
- number of tables

Testing
L = 1
M = 2
H = 3
C = 4
Helpful Considerations:
- user testing complexity
- data setup complexity
- test automation



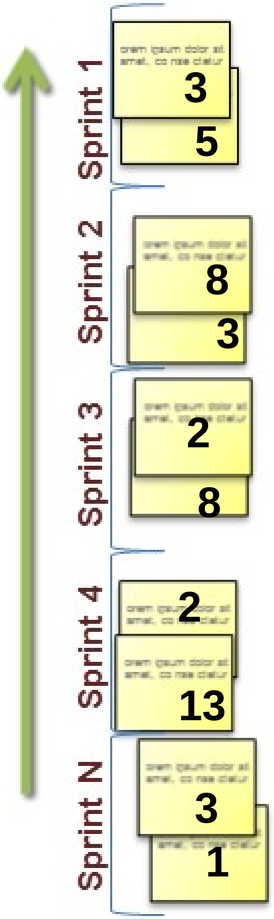
Example:
As a customer, I want to browse the list of products.
User Interface: M = 2
Business Logic: N/A
Data: L = 1
Testing: L = 1
Total is 4 points, which is between 3 and 5, team decides on 3.

What is Velocity?

Velocity is a measure of a team’s rate of progress used to estimate future commitments/capacity.

- It is measured by summing the number of story points ‘DONE’ in an iteration/sprint.
- No ‘partial points’ are allowed! Done or Not Done.
- Average velocity is an ESTIMATE, teams should use a range to represent uncertainty.
- Velocity is a critical measure used for planning.
- Cannot compare one team’s velocity to the other.

Velocity and Points



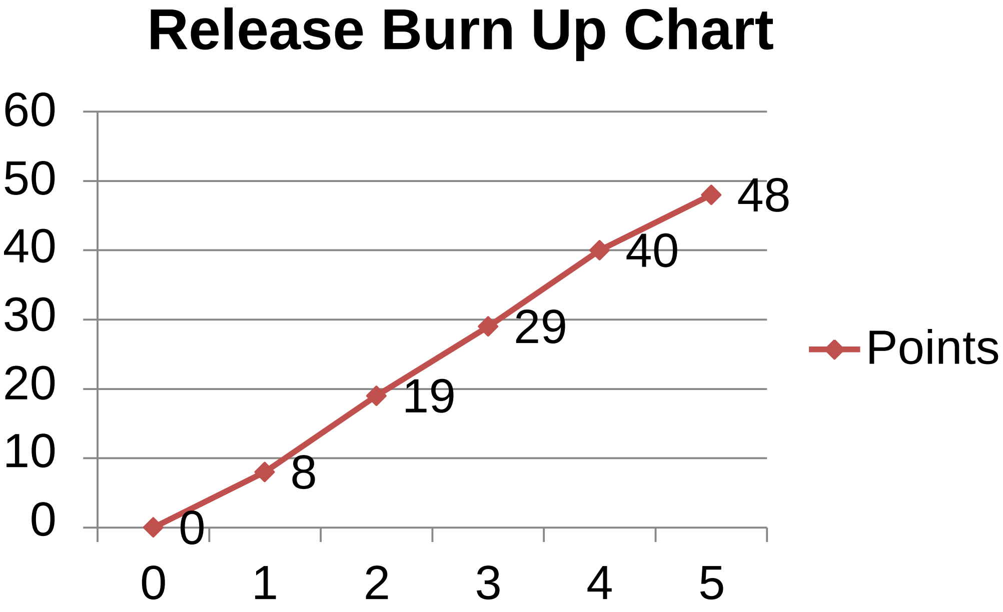
Velocity = 8 pts

Velocity = 11 pts

Velocity = 10 pts

Velocity = 15 pts

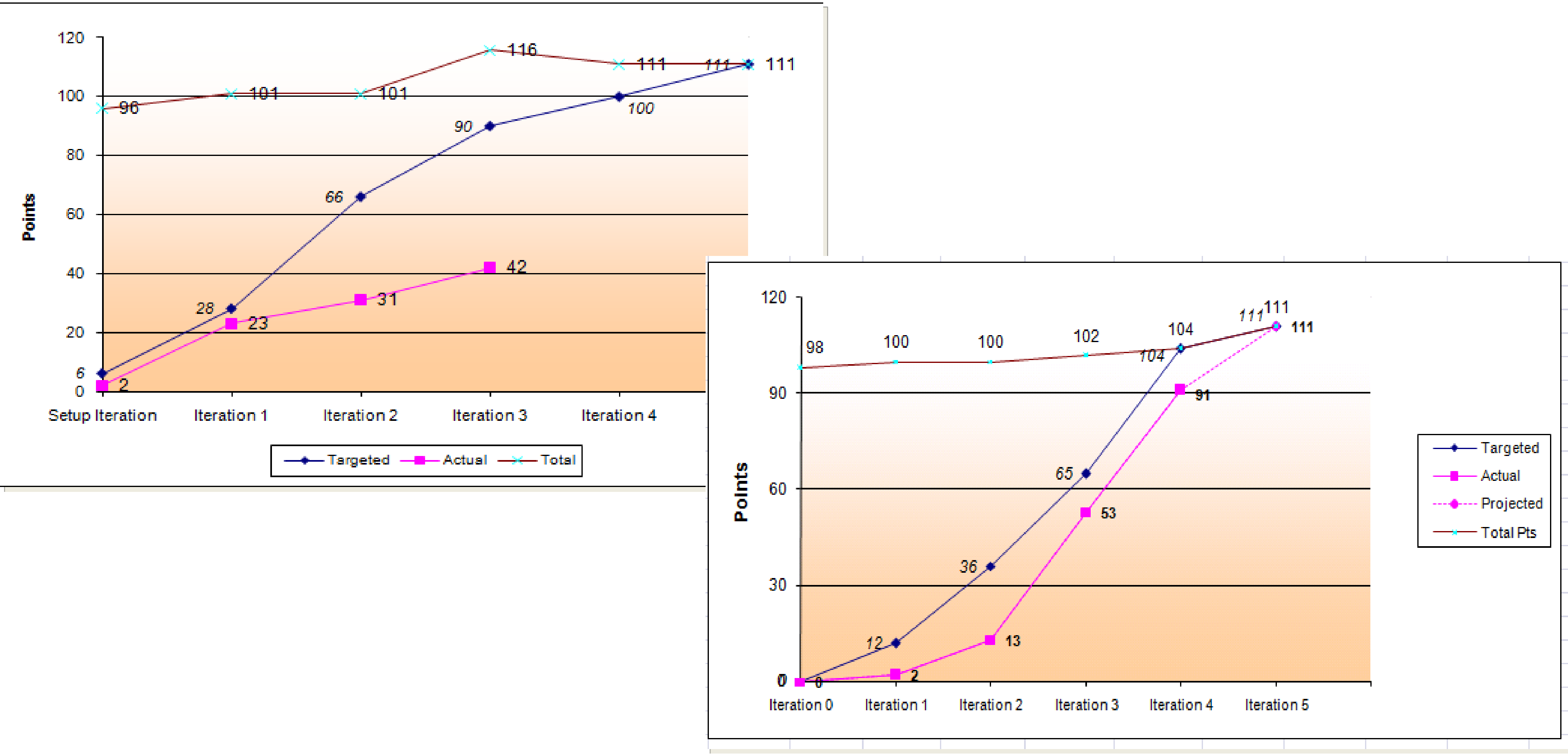
Velocity = 4 pts



Avg Sprint Velocity = 12 pts

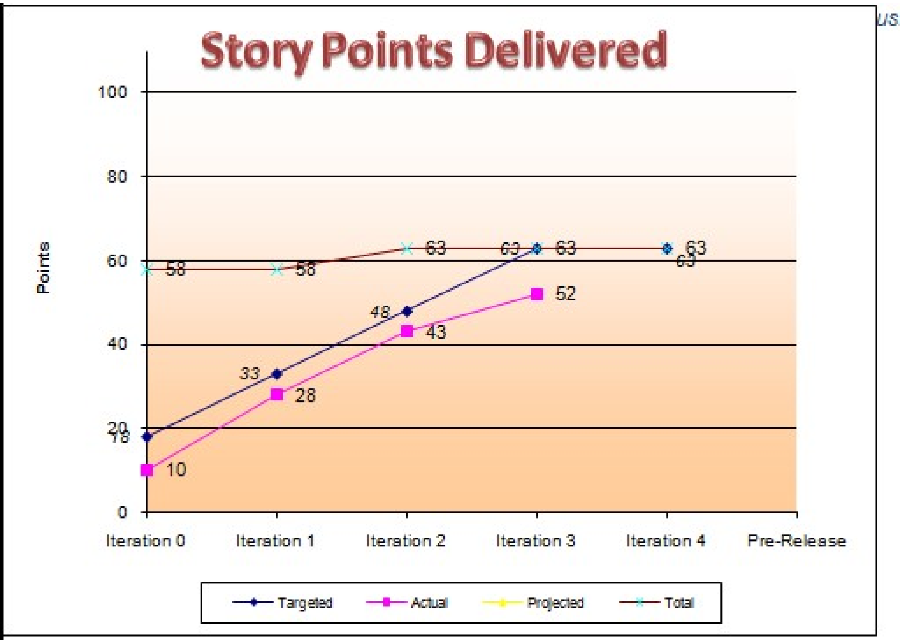
Backlog = 48 pts

Measuring Velocity

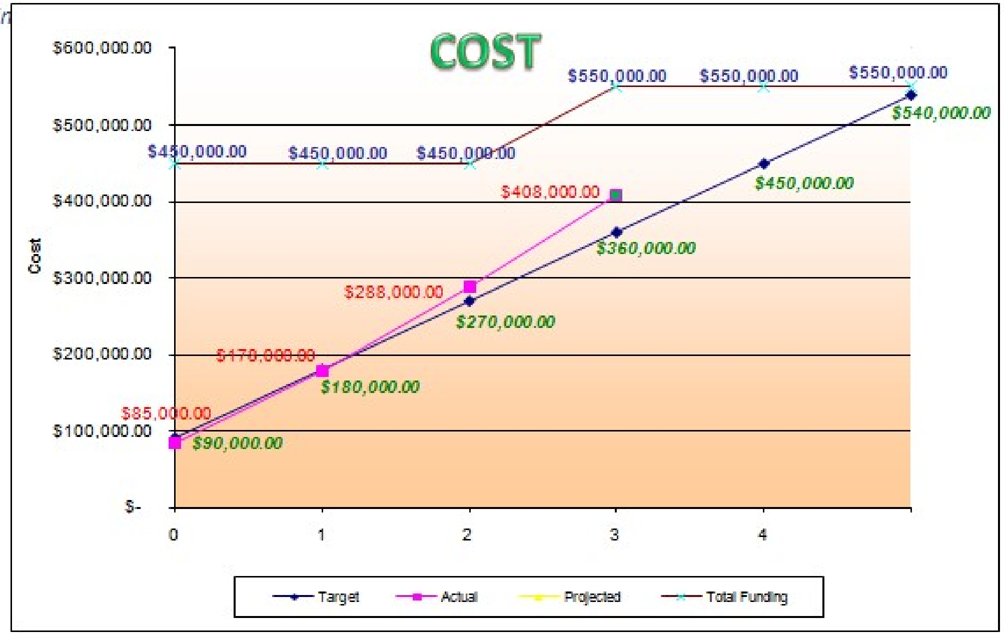


Sample Point and Cost Burn Up Chart

What is 'Done'? How much did it cost me?



usin

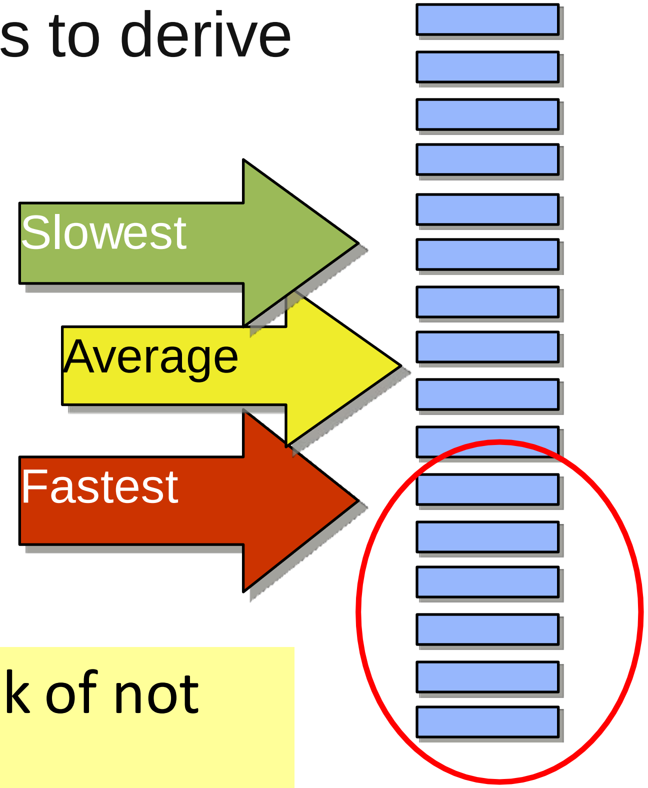


Current Project % Done 83%

Current Project % Burn 74%

How Velocity Helps with Planning

- Understanding the team’s velocity will enable better release planning and estimation.
- Over time utilize multiple data points to derive velocity:
  - Fastest pace
  - Average pace
  - Slowest pace



The features/stories at the bottom are at risk of not getting Done. Also called Feature Buffers

Task – Calculate Range Based on Historical Data

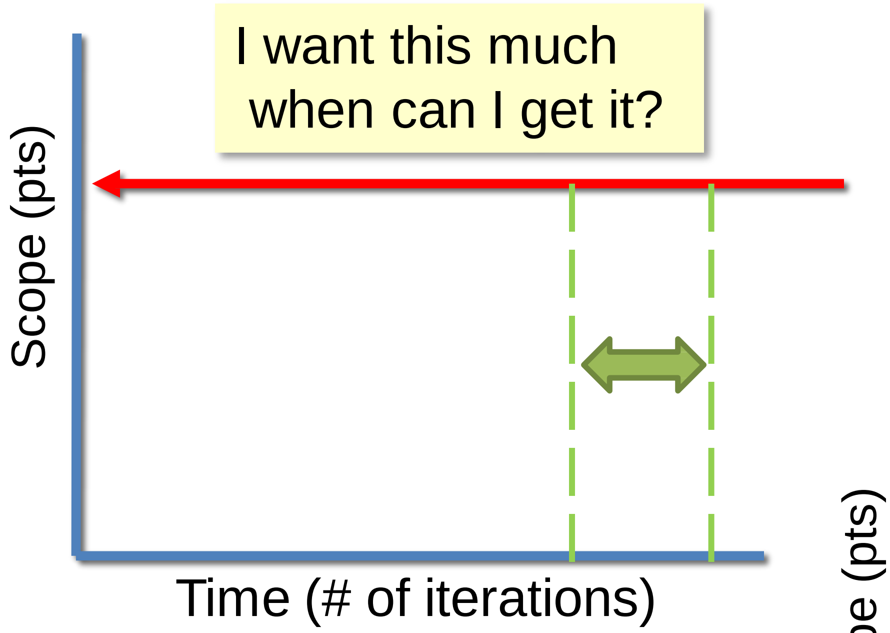
Iteration	Velocity
1	10
2	15
3	20
4	18
5	23
6	30
7	24
8	20

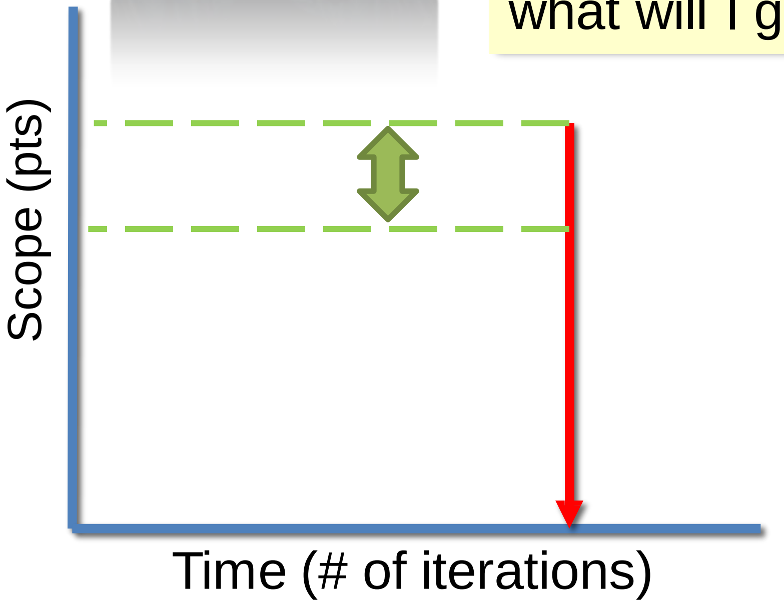
Release 2 has 200 pts. How many points can you confirm for the next 8 iterations? How many will likely be completed?

Factors Affecting Team Velocity

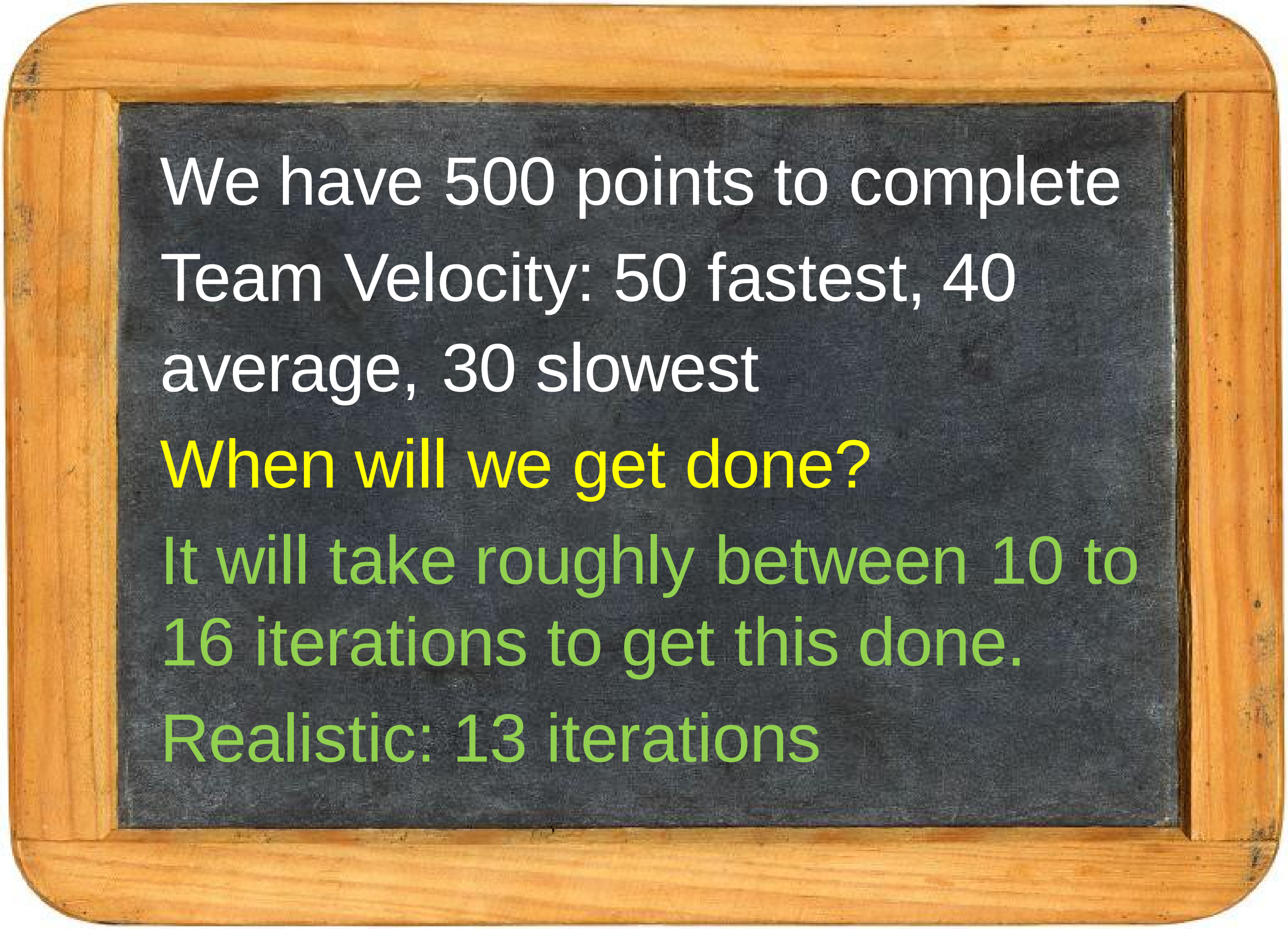
- Number of members
- Unresolved Impediments
- Unclear acceptance criteria
- Shifting priorities
- Interruptions
- Multi-tasking
- Skill level
- New members
- Team dynamics
- Vacation/sick time

Fixed Time .. Fixed Scope


I want this much when can I get it?


I need it by this date what will I get?

Example – Planning for Fixed Scope



We have 500 points to complete

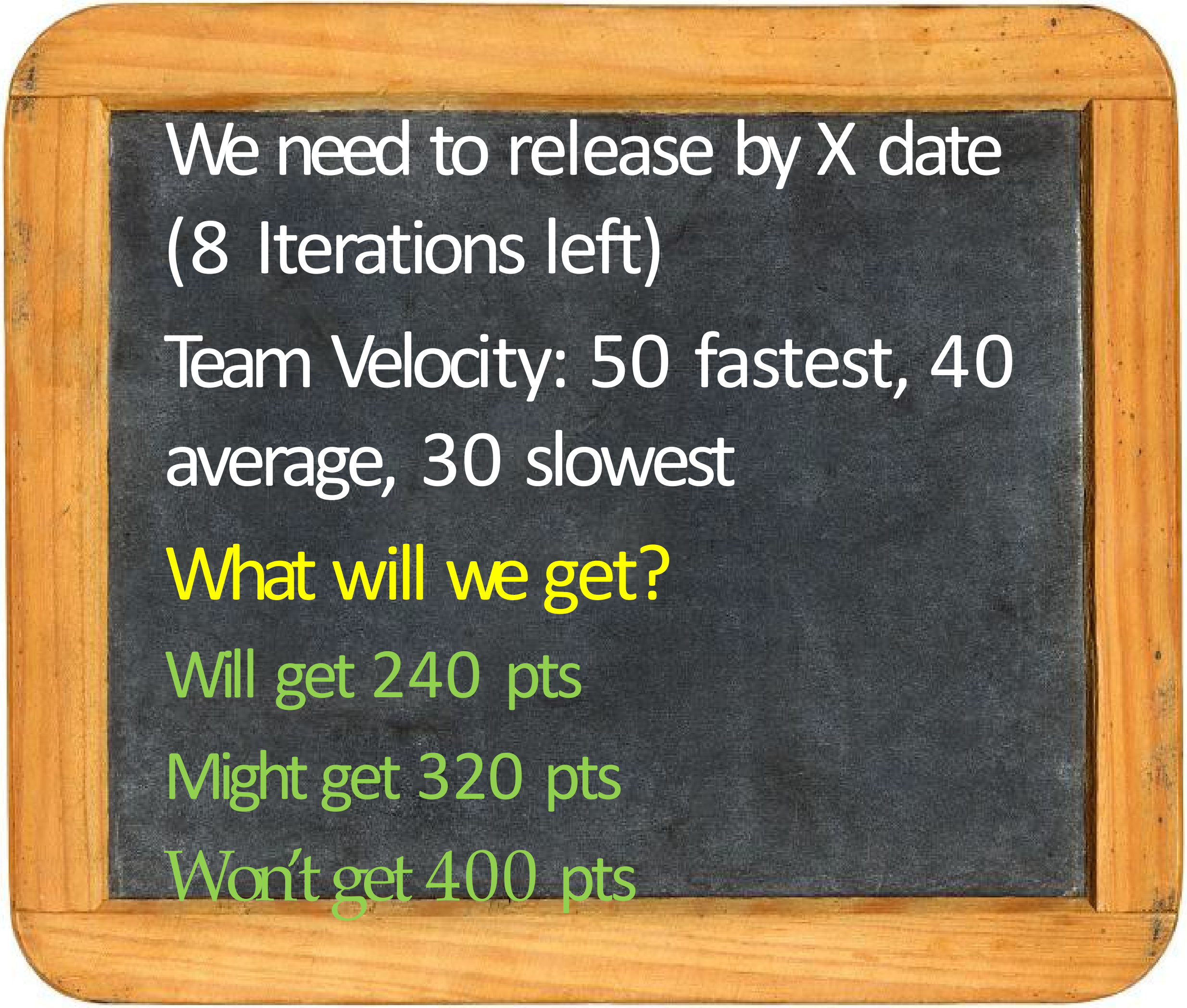
Team Velocity: 50 fastest, 40 average, 30 slowest

When will we get done?

It will take roughly between 10 to 16 iterations to get this done.

Realistic: 13 iterations

Example: Planning for Fixed Date



We need to release by X date (8 Iterations left)

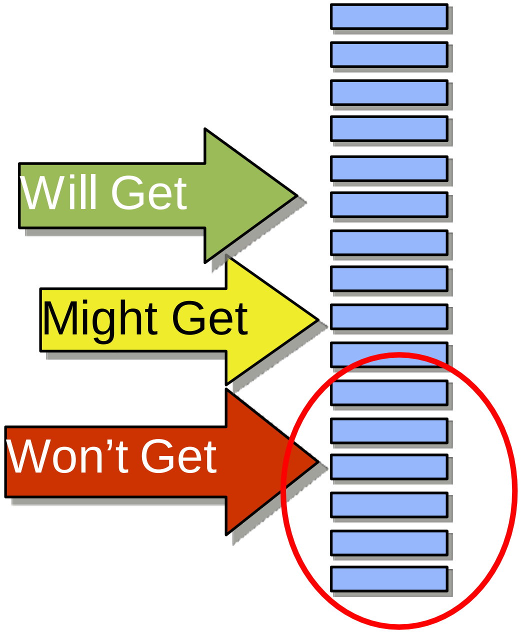
Team Velocity: 50 fastest, 40 average, 30 slowest

What will we get?

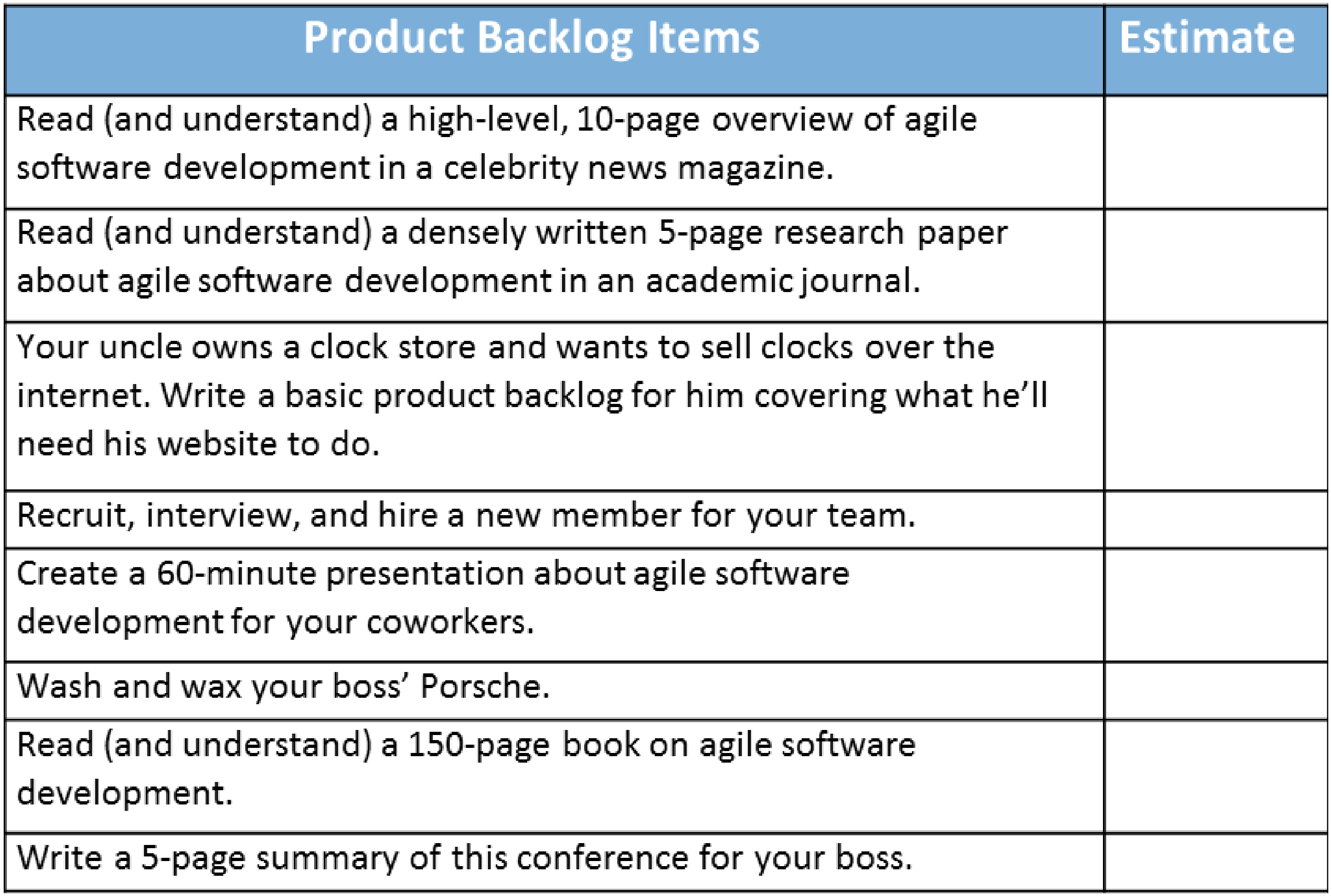
Will get 240 pts

Might get 320 pts

Won't get 400 pts



Agile Estimation and Planning Workshop Tasks



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-page 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'll need his website to do.	
Recruit, interview, and hire a new member for your team.	
Create a 60-minute presentation about agile software development for your coworkers.	
Wash and wax your boss' Porsche.	
Read (and understand) a 150-page book on agile software development.	
Write a 5-page summary of this conference for your boss.	

Resources

https://www.agilealliance.org/
http://agiletransformation.com
www.mountaingoatsoftware.com/scrum
https://www.scrumalliance.org/why-scrum/scrum-resources
http://www.scrumhub.com/
http://www.scrumguides.org/
http://www.scrumstudy.com/download-free-buy-SBOK.asp