

Agile Estimating and Planning

Mike Cohn

mike@mountaingoatsoftware.com

June 8, 2010

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



What makes planning agile?

Is more focused on planning than the plan

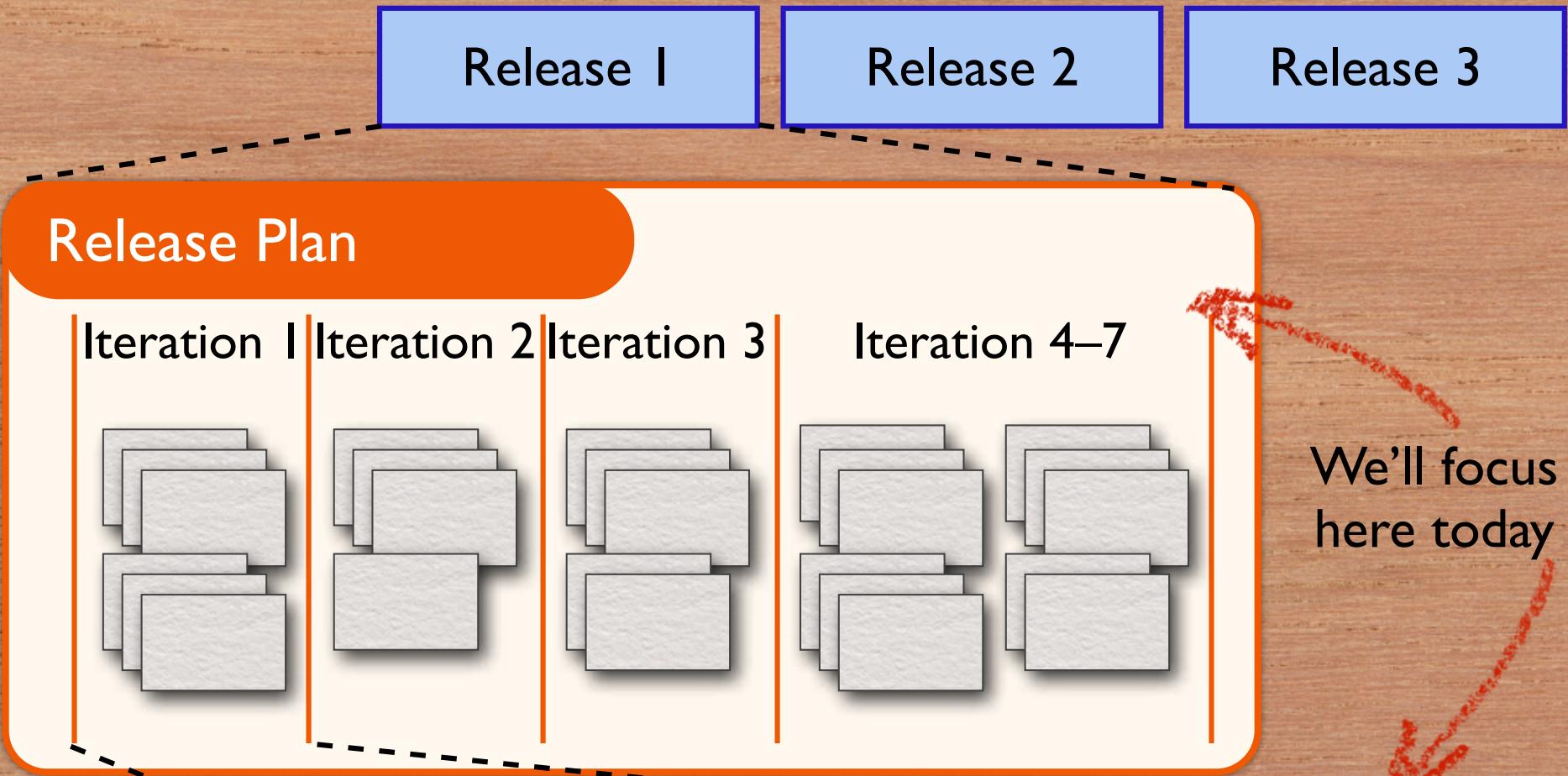
Encourages change

Results in plans that are easily changed

Is spread throughout the project



Product, release, iteration planning



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



Agenda

- Product backlog estimation
 - units
 - Story points
 - Ideal time
 - Techniques for estimating
 - Iteration planning
 - Release planning

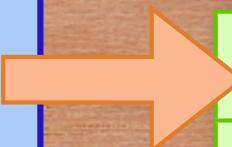


Product Backlog

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...	8
As a frequent flyer, I want to...	5

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

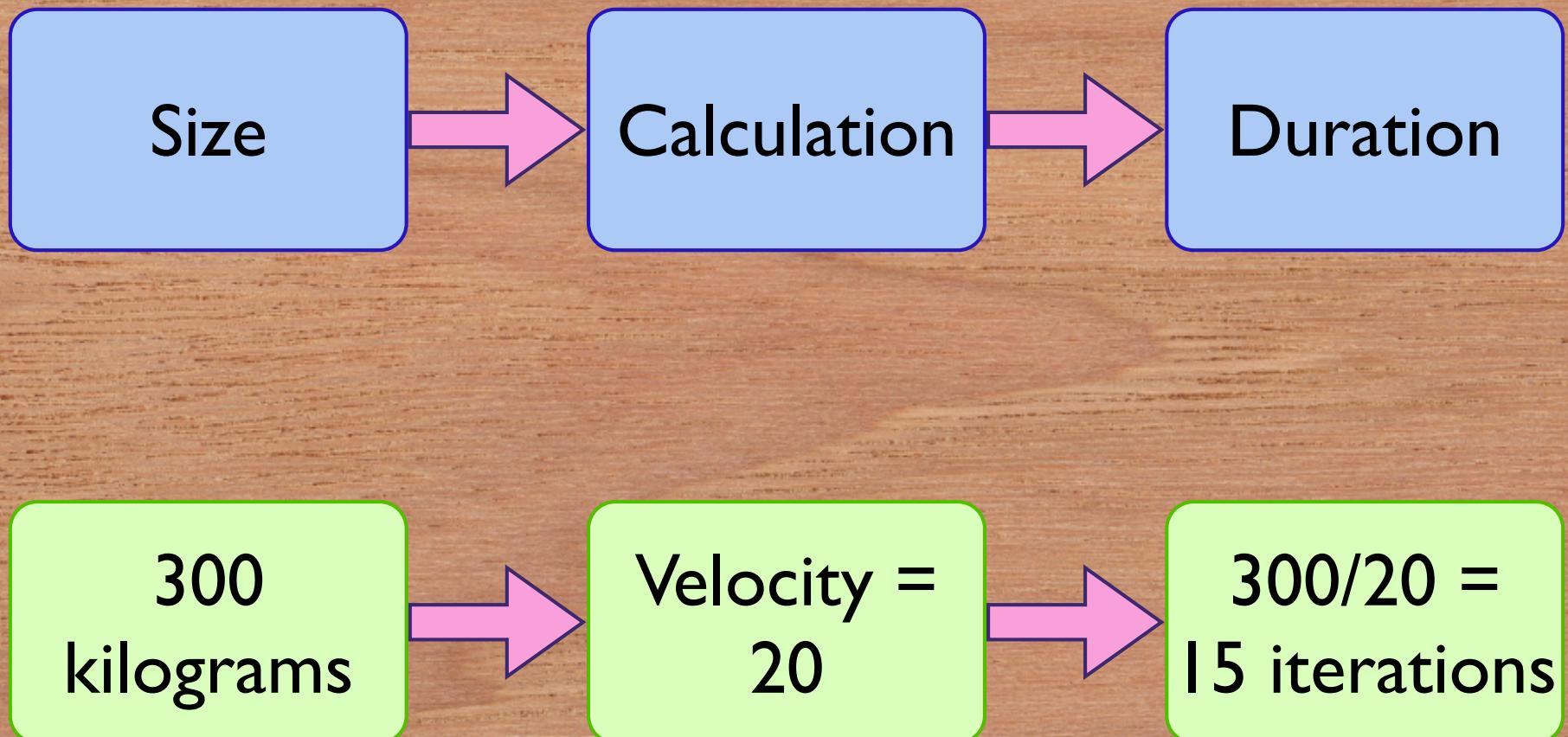


How long will it take...

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

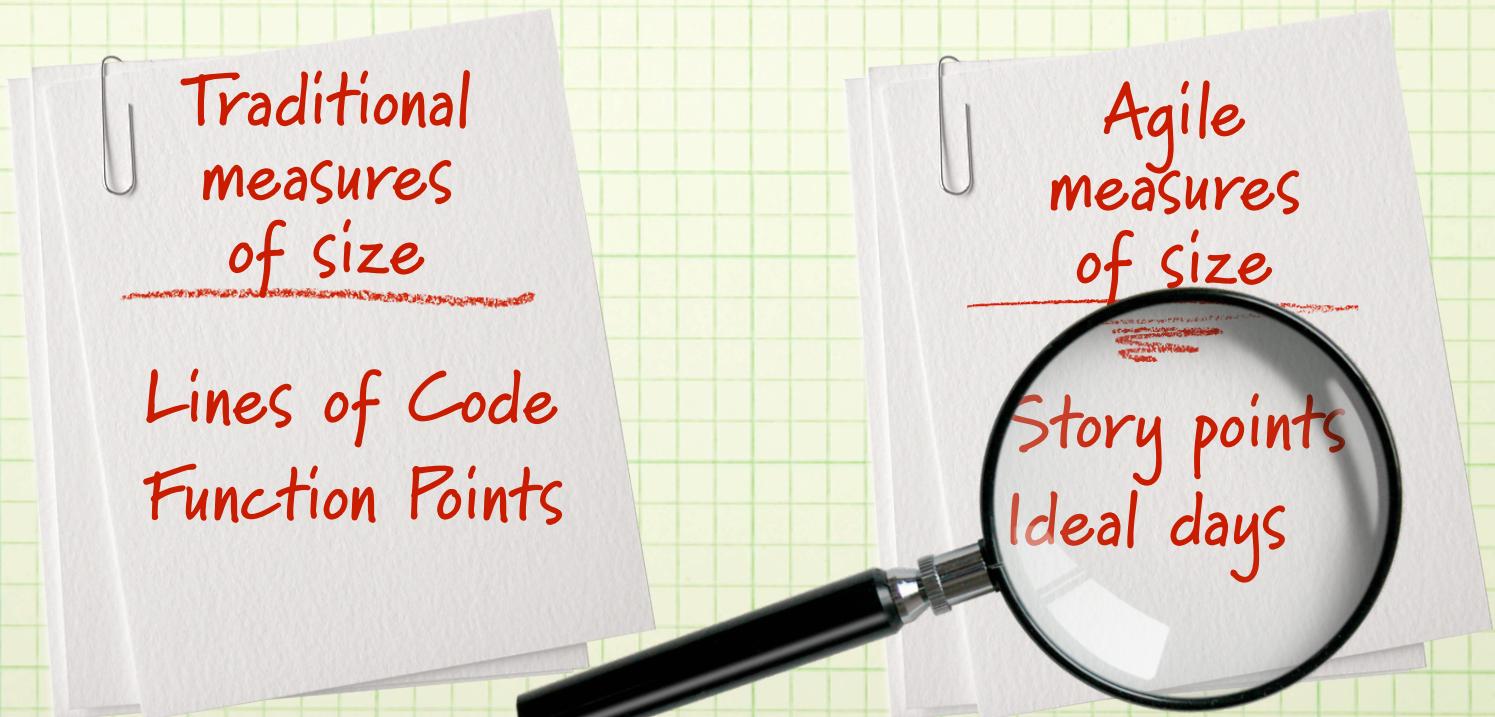


Estimate size; derive duration



Measures of size

- Traditional and agile measure size differently



Story points

- The “bigness” of a task
- Influenced by
 - How hard it is
 - How much there is
- Relative values are what is important:
 - A login screen is a 2.
 - A search feature is an 8.
- Points are unit-less
- Basic math properties should hold, e.g., $5+5 = 10$

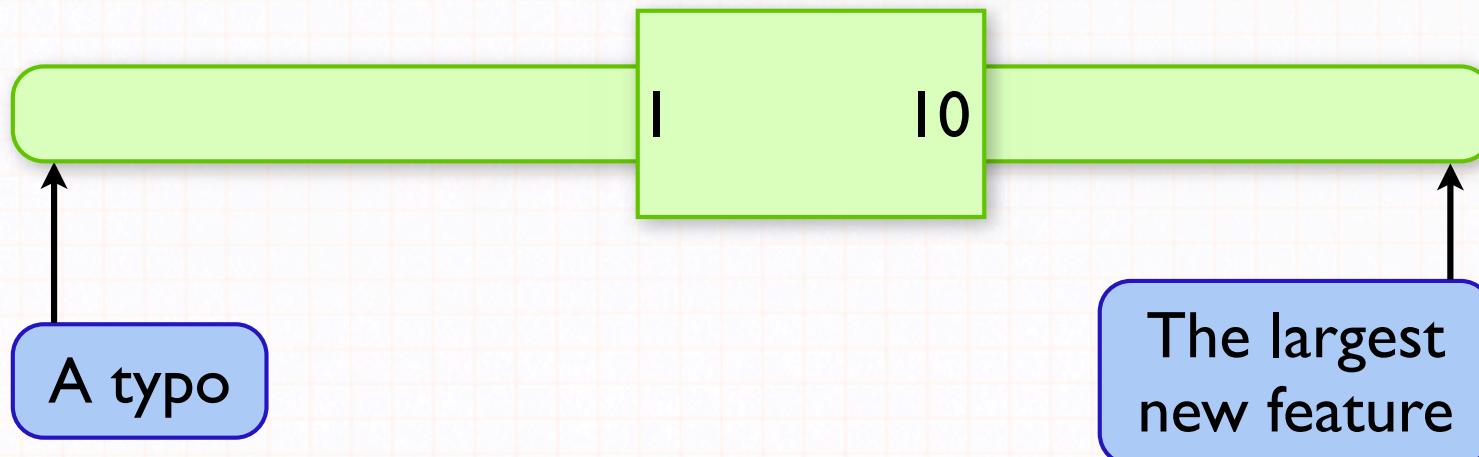
As a user, I want to be able to have some but not all items in my cart gift wrapped.

8



One order of magnitude

- We're good over one order of magnitude
- So think about where to place it on your product backlog



Agenda

- Product backlog estimation units
 - Story points
 - Ideal time
- Techniques for estimating
- Iteration planning
- Release planning



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)



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



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



Agenda

- Product backlog estimation units
 - Story points
 - Ideal time
- Techniques for estimating
- Iteration planning
- Release planning



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

3 points	Story A	
2 points	Story C	Story D
1 point	Story B	Story E

A hand-drawn red arrow points from the 'Story A' box in the top row to the 'Story D' box in the middle row, indicating a comparison between these two stories.



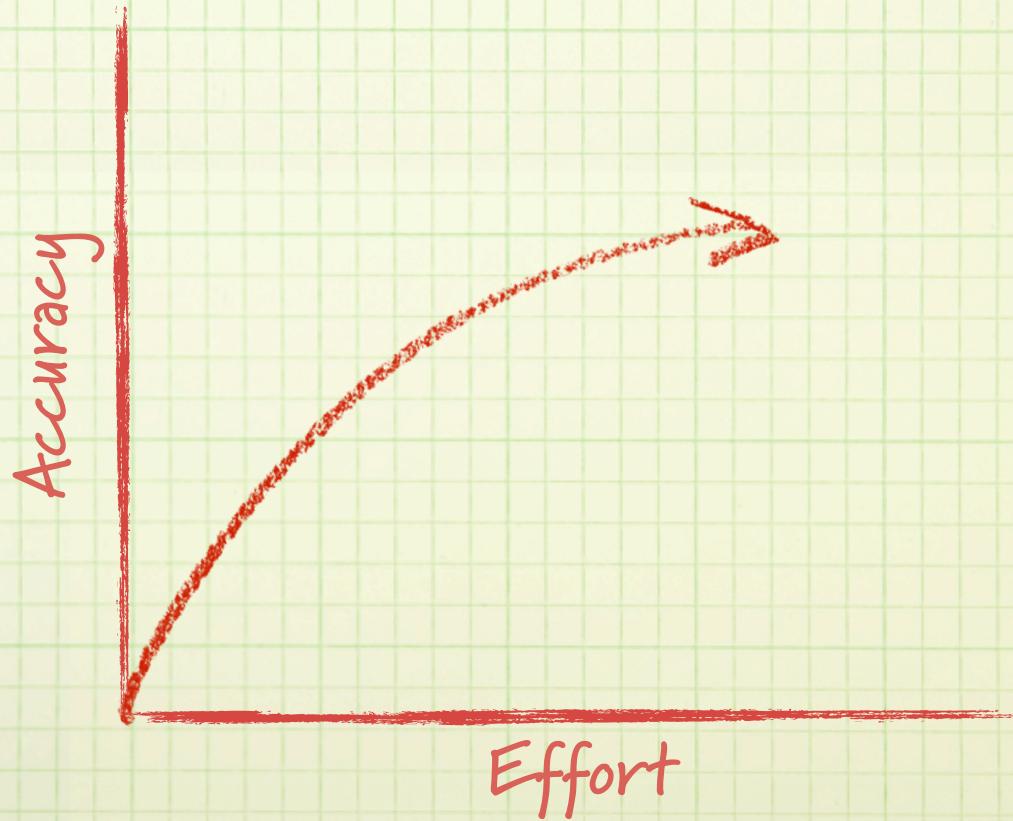
Disaggregation

- 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



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?
 - How about a 17 from an 18?
- Use a set of numbers that make sense; I like:
 - 1, 2, 3, 5, 8, 13, 20, 40, 100
- Stay mostly in a 1-10 range
- Nature agrees:
 - Musical tones and volume are distinguishable on a logarithmic scale

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



Why Planning Poker works

- Those who will do the work, estimate the work¹
- Estimators are required to justify estimates^{2, 3}
- Focuses most estimates within an approximate one order of magnitude^{4, 5}

¹Jørgensen, Magne. 2004. *A Review of Studies on Expert Estimation of Software Development Effort.*

²Hagafors, R., and B. Brehmer. 1983. *Does Having to Justify One's Decisions Change the Nature of the Decision Process?*

³Brenner, et al. 1996. *On the Evaluation of One-sided Evidence.*

⁴Miranda, Eduardo. 2001. *Improving Subjective Estimates Using Paired Comparisons.*

⁵Saaty, Thomas. 1996. *Multicriteria Decision Making: The Analytic Hierarchy Process.*



Why Planning Poker works

- Combining of individual estimates⁶ through group discussion⁷ leads to better estimates
- Emphasizes relative rather than absolute estimating
- Estimates are constrained to a set of values so we don't waste time in meaningless arguments
- Everyone's opinion is heard
- It's quick and fun

⁶Hoest, Martin, and Claes Wohlin. 1998. *An Experimental Study of Individual Subjective Effort Estimations and Combinations of the Estimates*.

⁷Jørgensen, Magne, and Kjetil Moløkken. 2002. *Combination of Software Development Effort Prediction Intervals: Why, When and How?*



www.PlanningPoker.com

The screenshot shows a web browser window for the URL <http://thijs.planningpoker.com/games/26>. The title bar says "Payroll system replacement". The main content area displays a list of user stories with their estimated values:

- 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: 3)
- As a/an admin I would like to add new users so that they can log in (Estimate: 2)

Below the stories, it asks "How are they going to get their username and password?". A row of cards shows the following estimates:

3	3	5	13	20
Thijs V.	Manfred S.	Mike C.	Giel N.	Angie

Below this, another row of cards shows the following estimates:

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

At the bottom, there is a "Complete" button and a note: "(Note: Completes automatically when all estimates are in)".

The right sidebar contains the following information:

- All games
- Estimator access ([Lock](#))
http://thijs.planningpoker.com/wy24mg
Estimators can join the game at the above URL. [Send it by email](#)
- Countdown timer
[Start timer](#)
Start the 2 minute countdown timer when you think we've talked long enough.
- Done playing?
[Complete game](#)
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,
or I
wouldn't
mention it



Agenda

- Product backlog estimation units
 - Story points
 - Ideal time
- Techniques for estimating
- Iteration planning
- Release planning



Product Backlog

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...	8
As a frequent flyer, I want to...	

Creating this list is iteration planning

Iteration Backlog

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

Tasks

Hours



Two approaches

I

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



2

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

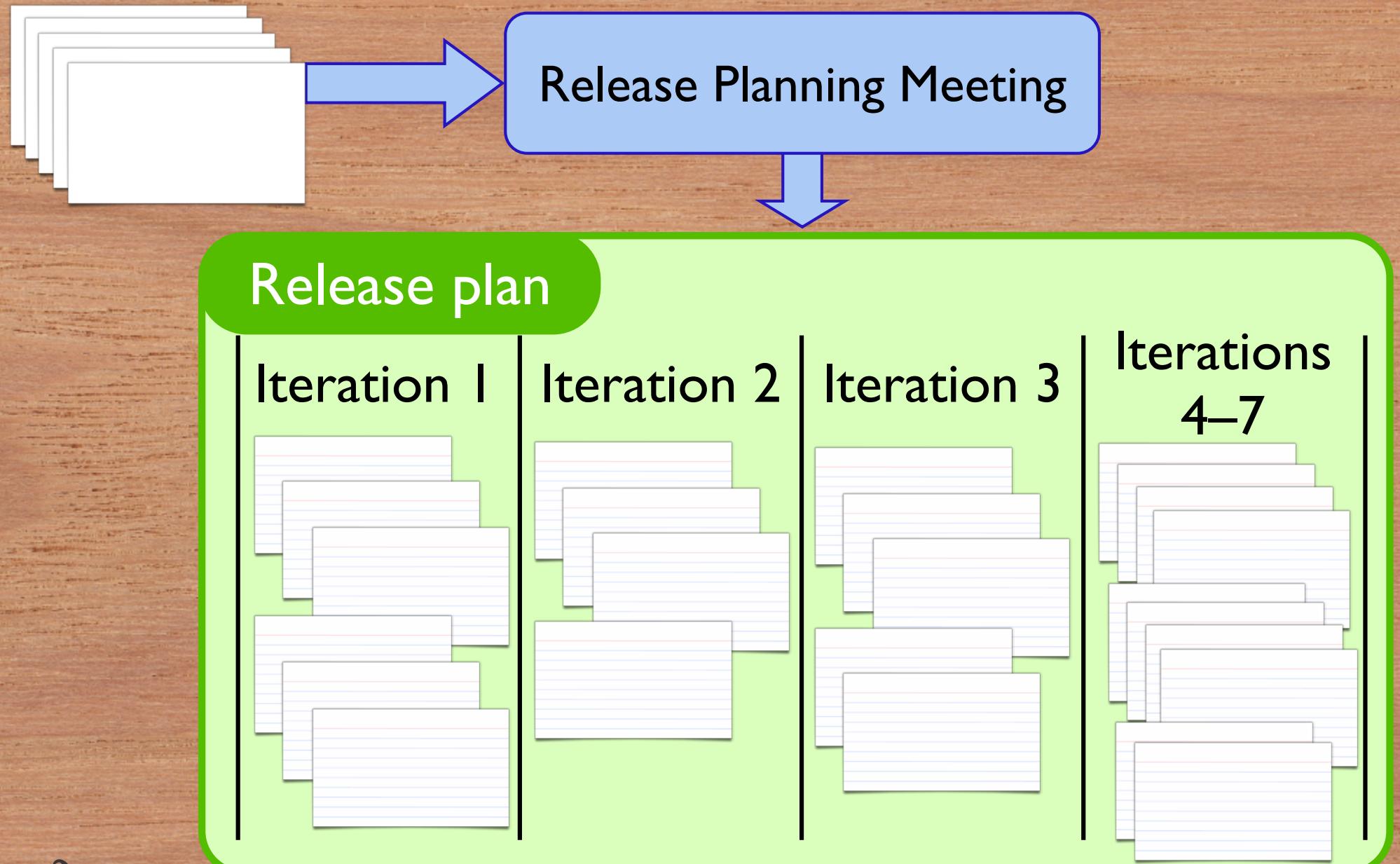


Agenda

- Product backlog estimation units
 - Story points
 - Ideal time
- Techniques for estimating
- Iteration planning
- Release planning



Release planning



Velocity

- To do a release plan, you need to know or have an estimate of velocity
- Three ways to get velocity:
 1. Use historical averages
 2. Run 1-2 iterations and see what you get
 3. Forecast it
- Should be expressed as a range
 - Size of range depends on familiarity of team, domain, and technologies



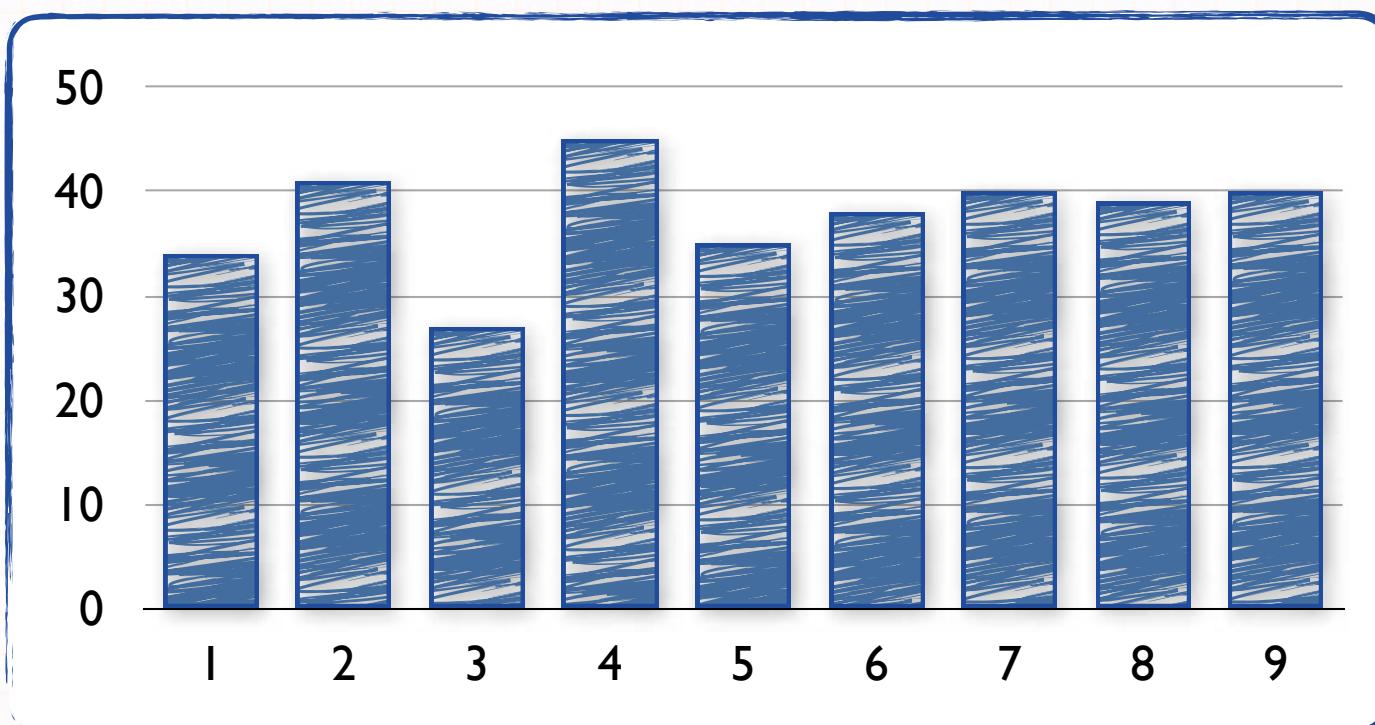
Forecasting velocity

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



Predicting release contents

- Determine your median velocity
- Put a 90% confidence interval around it
- Predicts “best case” and “worst case”



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

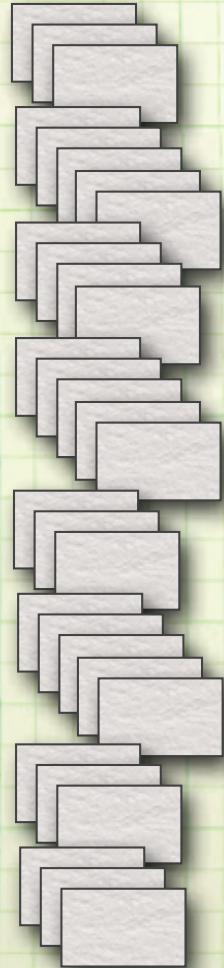
Use the next lower number
of iterations if you don't have
an exact number.

# of Historical Iterations	n th Highest & Lowest Iteration to Use
5	1
8	2
11	3
13	4
16	5
18	6
21	7
23	8
26	9

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



Extrapolate from the velocity range



Assume:

There are five
iterations 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)



Mike Cohn

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

twitter: mikewcohn

(720) 890-6110



MOUNTAIN GOAT
SOFTWARE



© Copyright Mountain Goat Software®