Agile Estimating and Planning

Mike Cohn - background

Agile coach and trainer

- Founding member and director of Agile Alliance and Scrum Alliance
- Founder of Mountain Goat Software
- Ran my first Scrum project back in 1995
- Typical programmer to manager etc. progression

Scrum

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

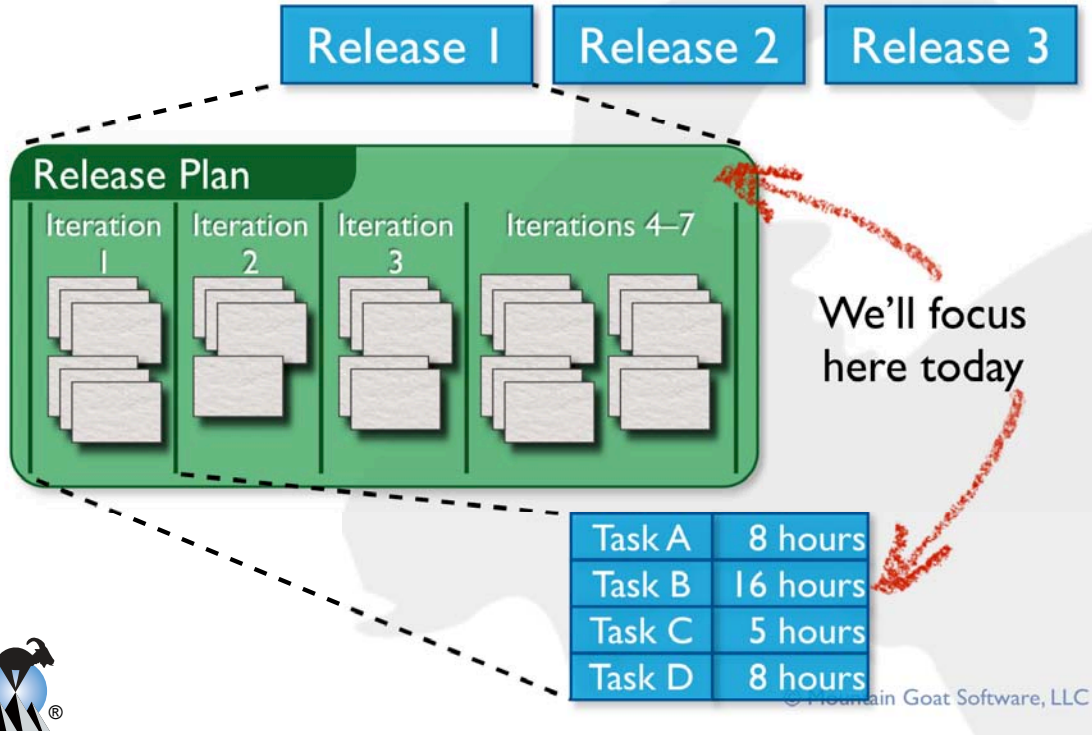
"It's better to be roughly right than precisely wrong."

~John Maynard Keynes

What makes planning agile?

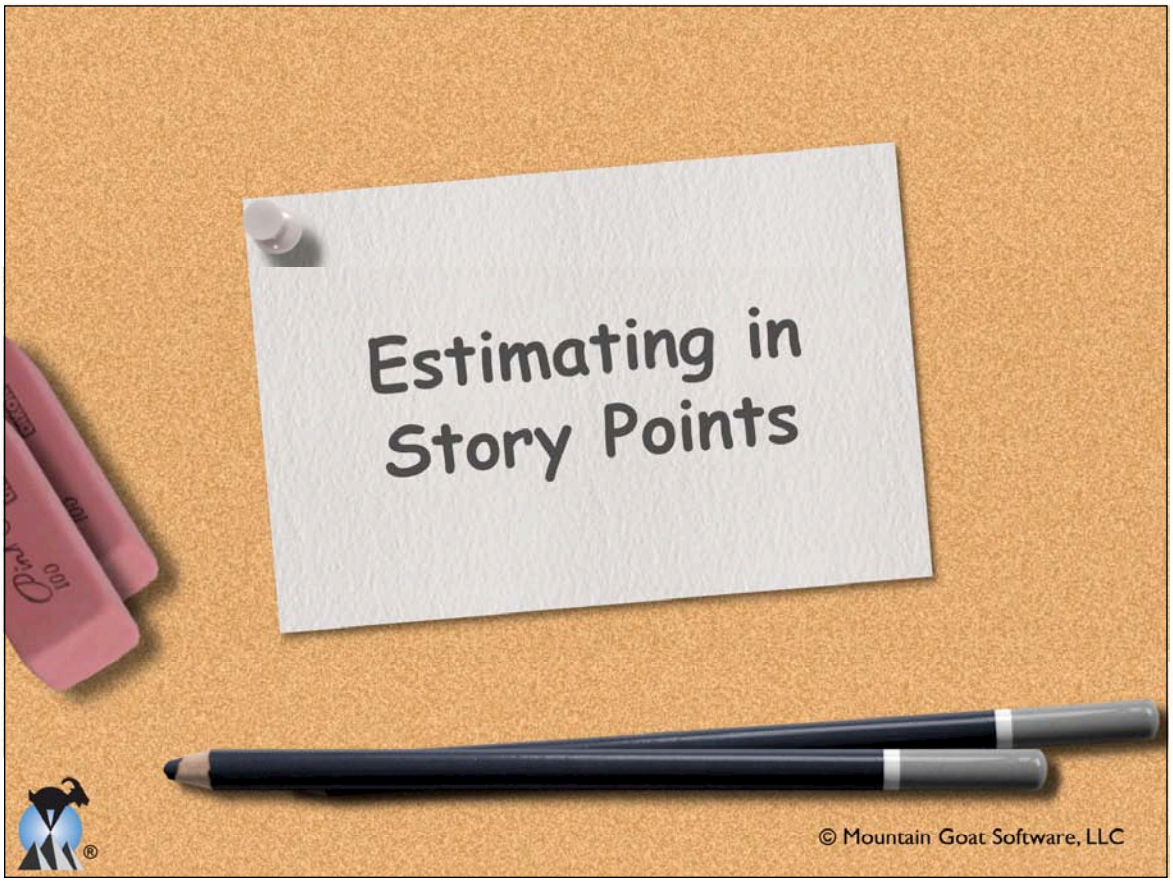
- Is more focused on planning than the plan
- Encourages change
- Results in plans that are easily changed
- Is spread throughout the project

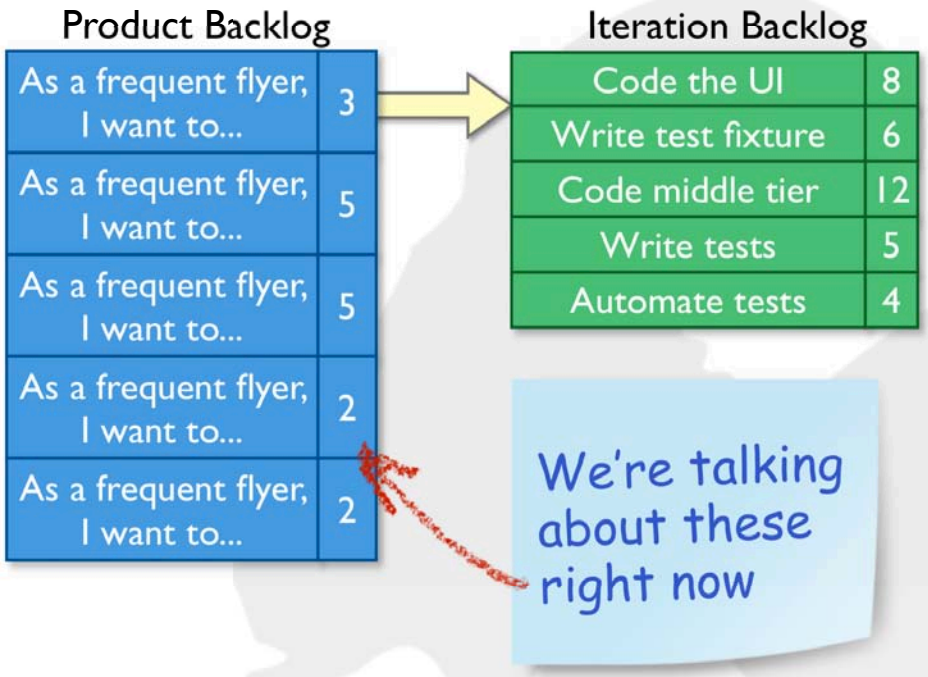
Product, release, iteration planning



Agenda

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

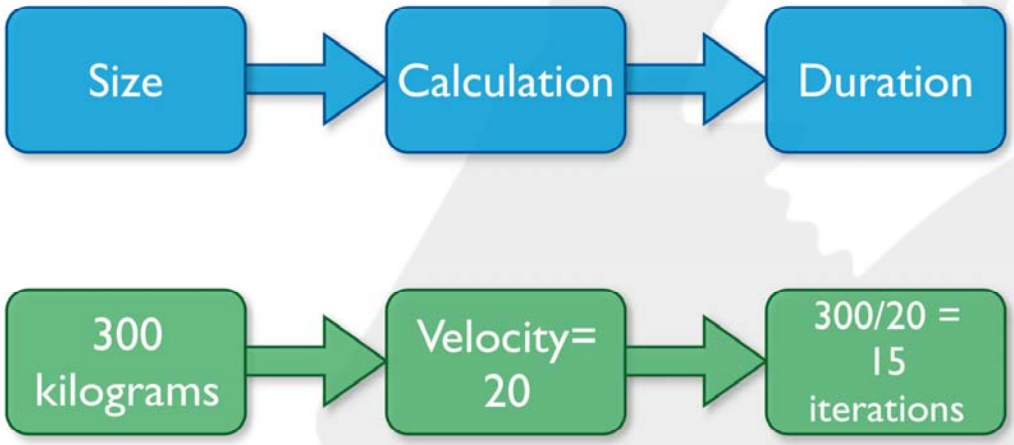




How long will it take...

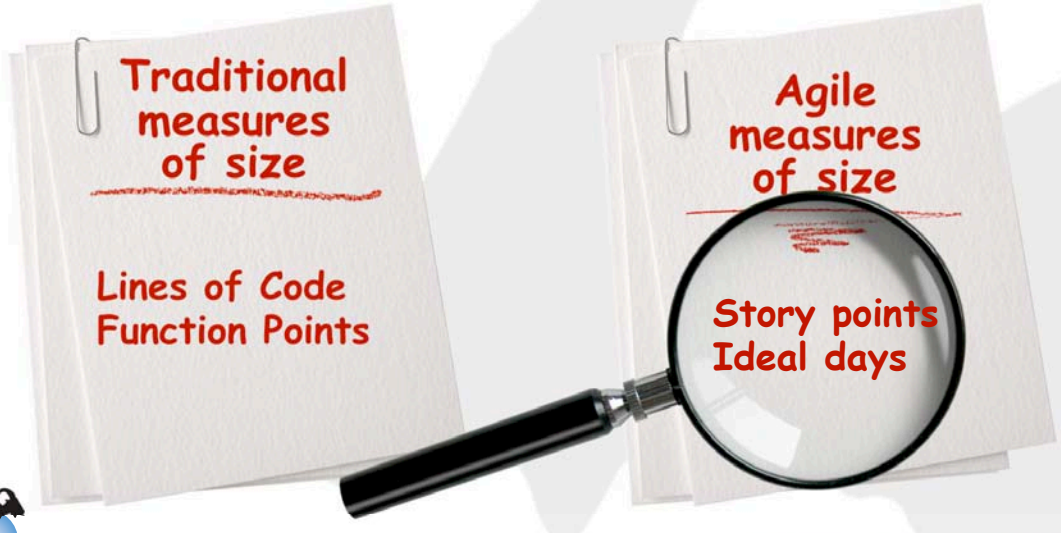
- ...to read the latest Harry Potter book?
- ...to drive to Minneapolis?

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 of it there is
- Relative values are what is important:
  - A login screen is a 2.
  - A search feature is an 8.
- Points are unit-less

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

5

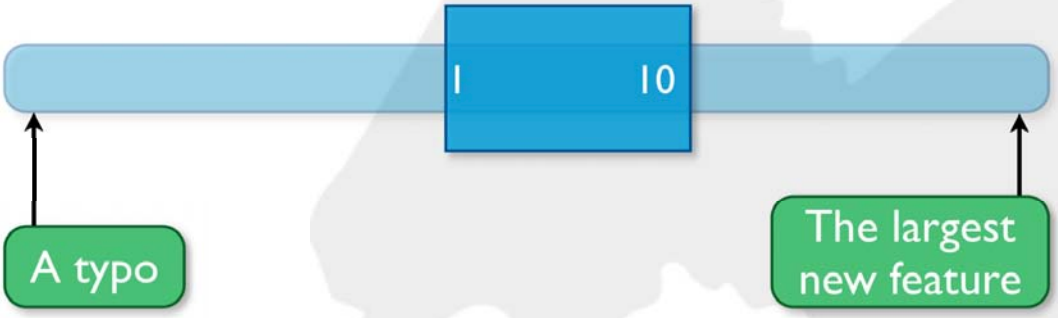
Dog points

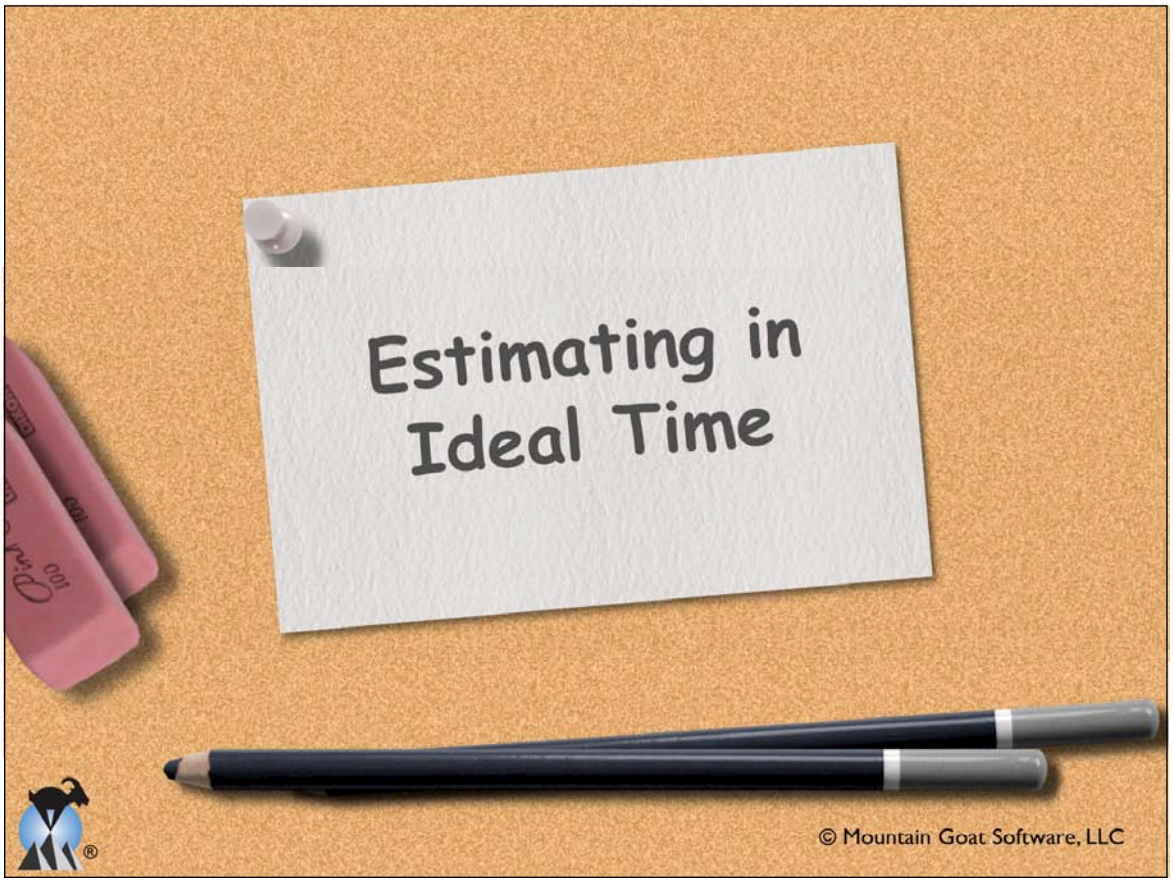
Assign "dog points" to the following breeds

Labrador retriever
Dachshund
Great Dane
Poodle
German Shepherd
Terrier
St. Bernard
Bulldog

One order of magnitude

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





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)

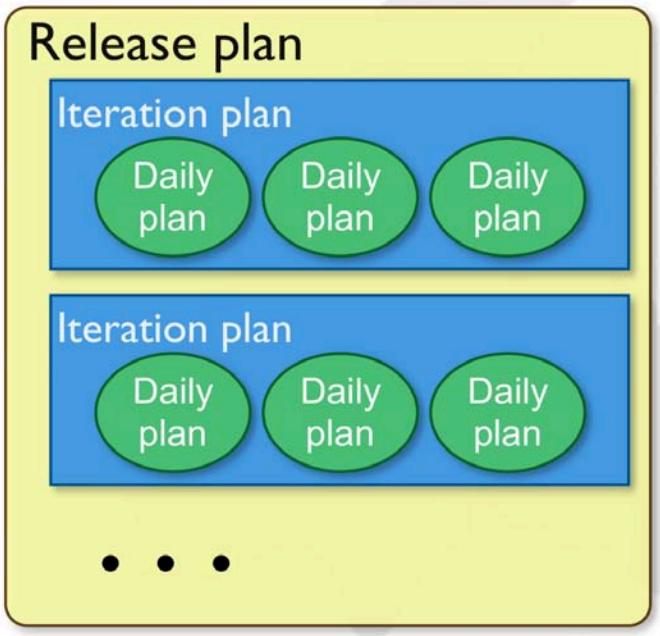
Ideal time vs. elapsed time

- 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

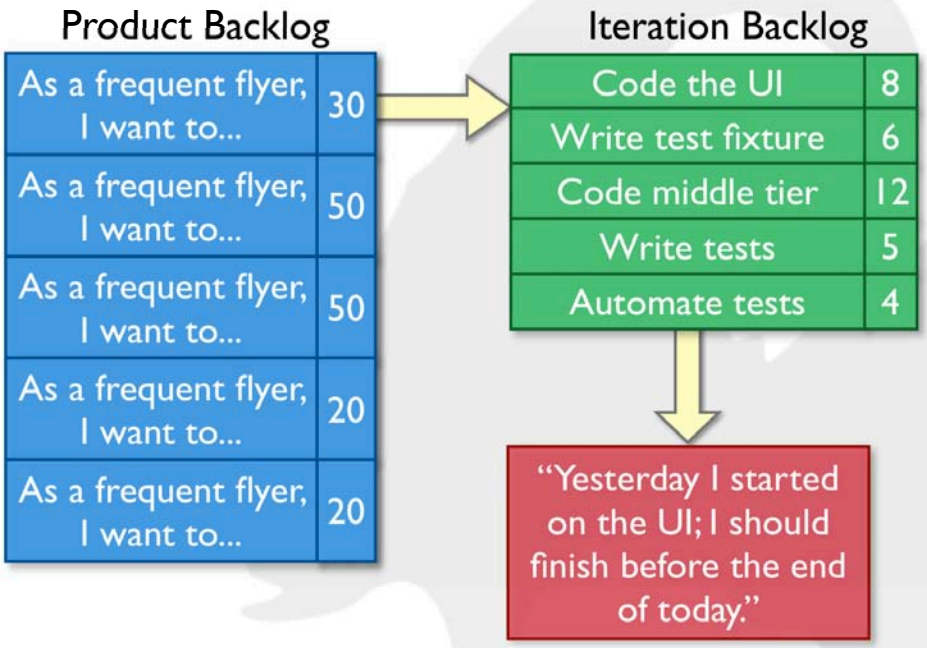
Comparing the approaches

- 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

Three levels of planning...

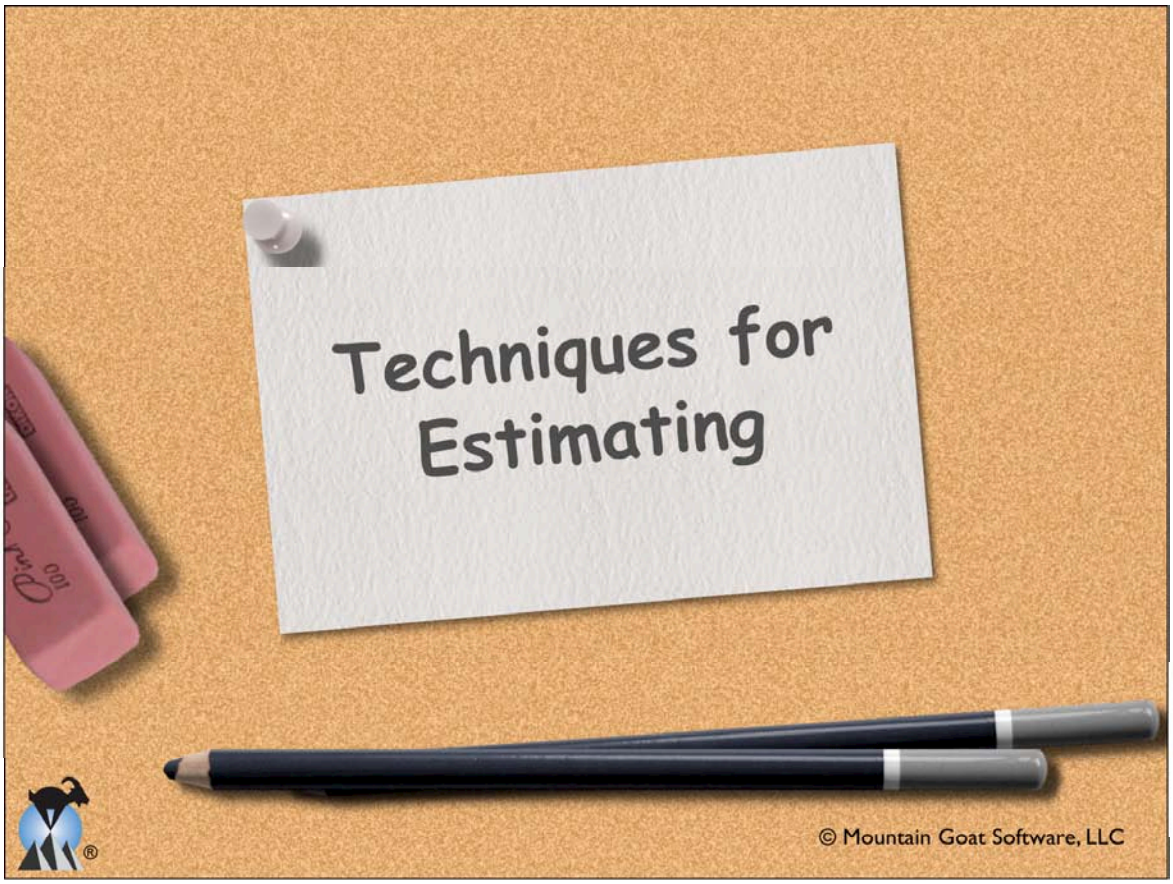


...three levels of precision



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

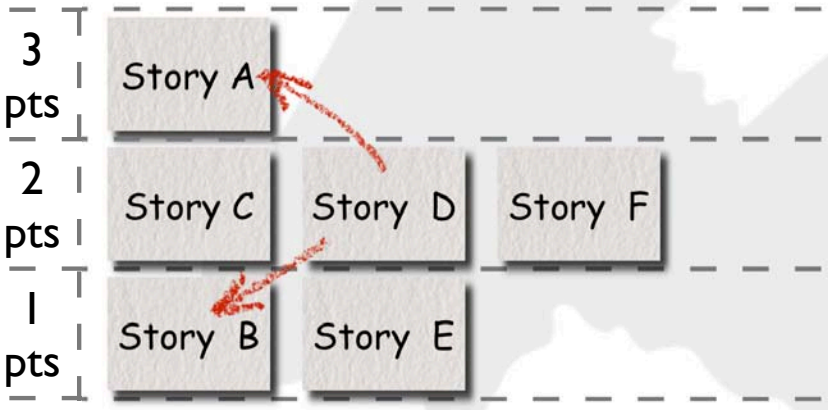


Estimate by analogy

- 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

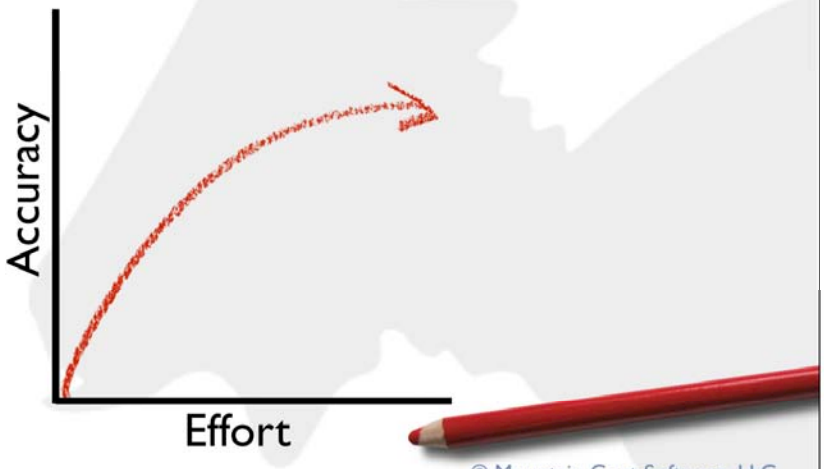


Disaggregation

- 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

Estimate these

Product backlog item	Estimate
Read a high-level, 10-page overview of agile software development in *People* magazine.	
Read a densely written 5-page research paper about agile software development in an academic journal.	
Write the product backlog for a simple eCommerce site that sells only clocks.	
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 a 150-page book on agile software development.	
Write an 8-page summary of this conference for your boss.	

Why planning poker works

- Those who will do the work, estimate the work[1]
- Estimators are required to justify estimates[2, 3]
- Focuses most estimates within an approximate one order of magnitude[4, 5]

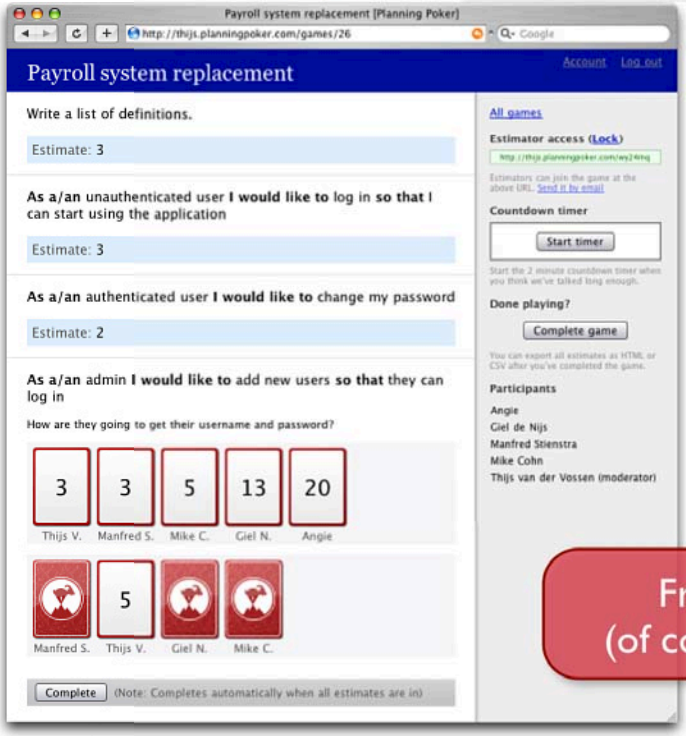
[1]Jørgensen, Magne. 2004. *A Review of Studies on Expert Estimation of Software Development Effort.*
[2]Hagafors, R., and B. Brehmer. 1983. *Does Having to Justify One's Decisions Change the Nature of the Decision Process?*
[3]Brenner, et al. 1996. *On the Evaluation of One-sided Evidence.*
[4]Miranda, Eduardo. 2001. Improving Subjective Estimates Using Paired Comparisons.
[5]Saaty, Thomas. 1996. *Multicriteria Decision Making: The Analytic Hierarchy Process.*

Why planning poker works

- Combining of individual estimates[6] through group discussion[7] 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

[6]Hoest, Martin, and Claes Wohlin. 1998. *An Experimental Study of Individual Subjective Effort Estimations and Combinations of the Estimates.*
[7]Jørgensen, Magne, and Kjetil Moløkken. 2002. *Combination of Software Development Effort Prediction Intervals: Why, When and How?*

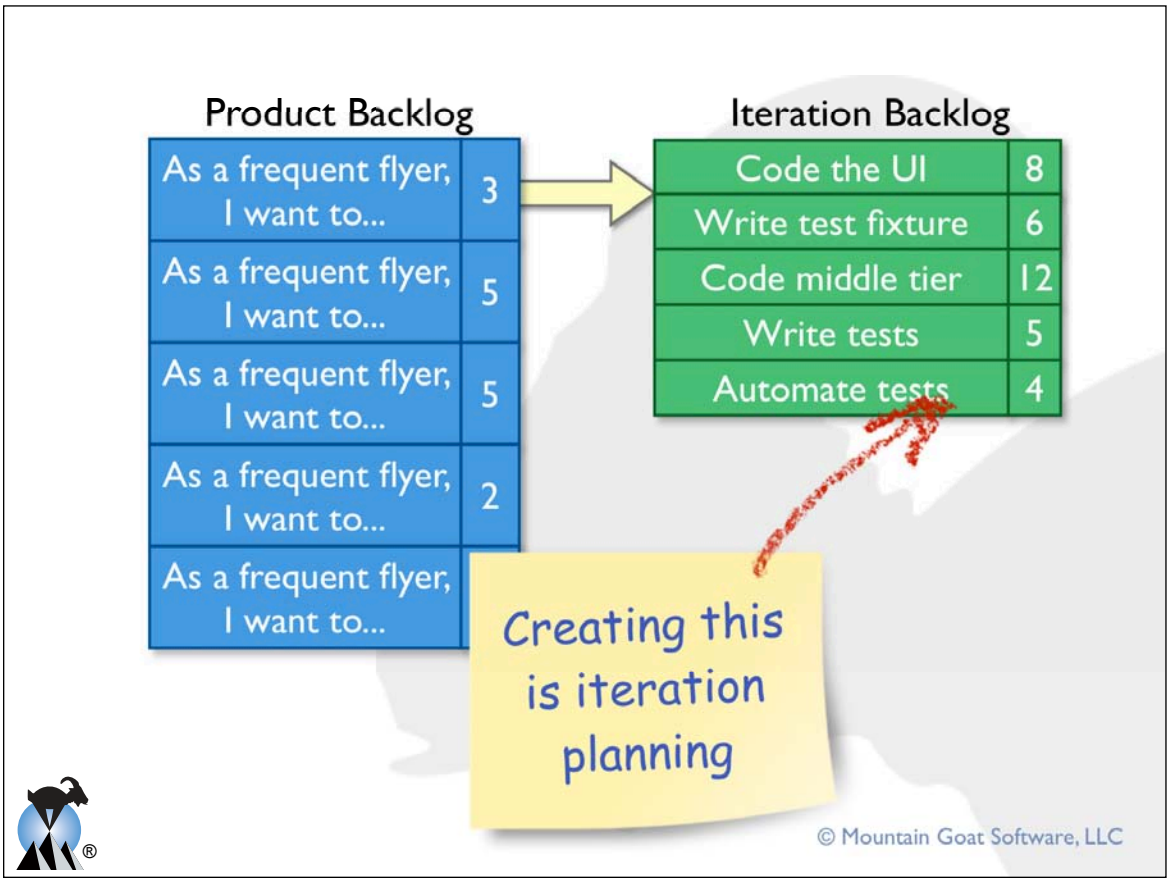
www.planningpoker.com



Free
(of course)


Iteration Planning
© Mountain Goat Software, LLC

35


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... 2
As a frequent flyer, I want to...
Iteration Backlog
Code the UI 8
Write test fixture 6
Code middle tier 12
Write tests 5
Automate tests 4
Creating this is iteration planning
© Mountain Goat Software, LLC

36

Two approaches

- 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
- Commitment-driven iteration planning

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

Estimate availability

Person	Hours per Day	Hours per Iteration
Sergey	4-6	40-60
Yuri	5-7	50-70
Carina	2-3	20-30
Total		110-160

It looks something like this

As a user, I want ...

2

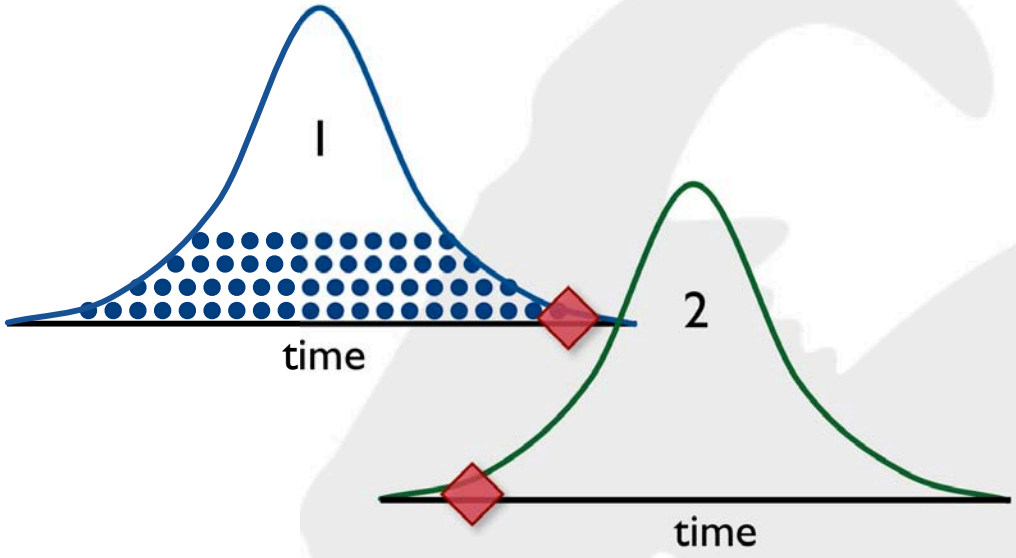
- Code the abc class (8 hours)
- Code the user interface (4)
- Write test fixtures (4)
- Code the xyz class (6)
- Update performance tests (4)

Team can commit, so they continue...

As a user, I want ...

3

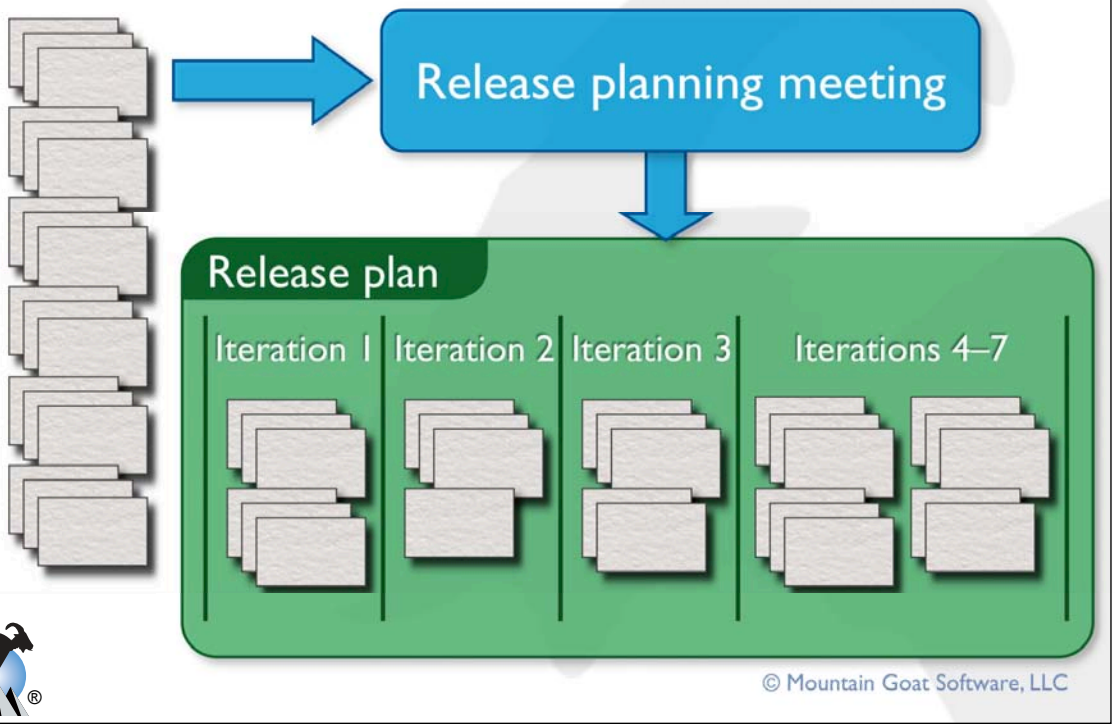
- Prototype the UI (8 hours)
- Demo UI to 3 outside users (3)
- Code new UI (12)
- Update documentation (3)



A caution

- The purpose of the iteration planning meeting is to arrive at a commitment to an iteration goal or set of product backlog items.
- The purpose of the meeting is not to come up with a list of tasks and hours.
- The tasks and estimates are a tool for determining what we can commit to.

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

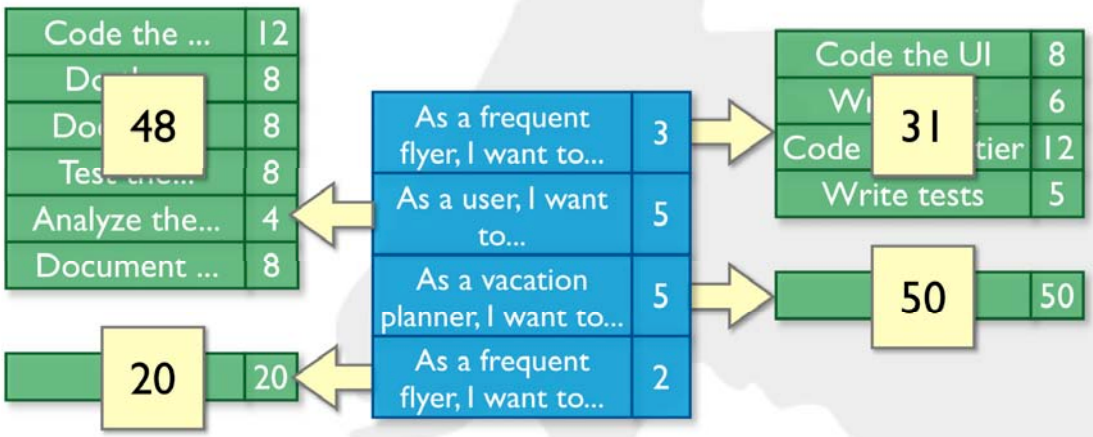
An example

- Estimating available hours

Person	Hours per Day	Hours per Iteration
Sergey	4-6	40-60
Yuri	5-7	50-70
Carina	2-3	20-30
Total		110-160

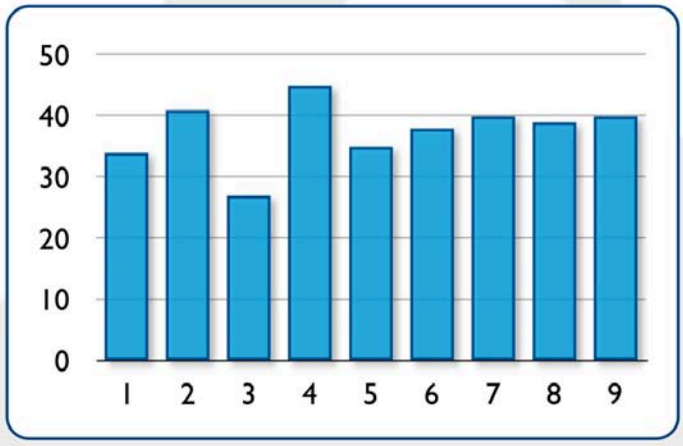
An example

At 110-160 available hours per iteration, what is the team's velocity?



Predicting release contents

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



An example

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

Extrapolate from velocity

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)

Upcoming public classes

Date	What	Where
Feb 1–2 Feb 3–4	Certified ScrumMaster Certified Scrum Product Owner	Dallas
March 1 March 2–3 March 4	User Stories for Agile Requirements Certified ScrumMaster Agile Estimating & Planning	Boulder
April 12 April 13–14 April 15	User Stories for Agile Requirements Certified ScrumMaster Succeeding with Scrum (new!)	San Diego

See mountaingoatsoftware.com for details

Mike Cohn

mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890−6110

twitter: mikewcohn

MOUNTAIN GOAT SOFTWARE