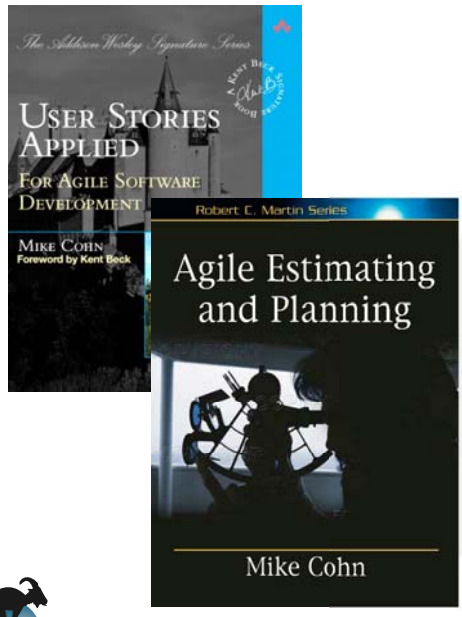
Planning and Tracking Agile Projects

Mike Cohn
August 15, 2007

1

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

2

Imagine...

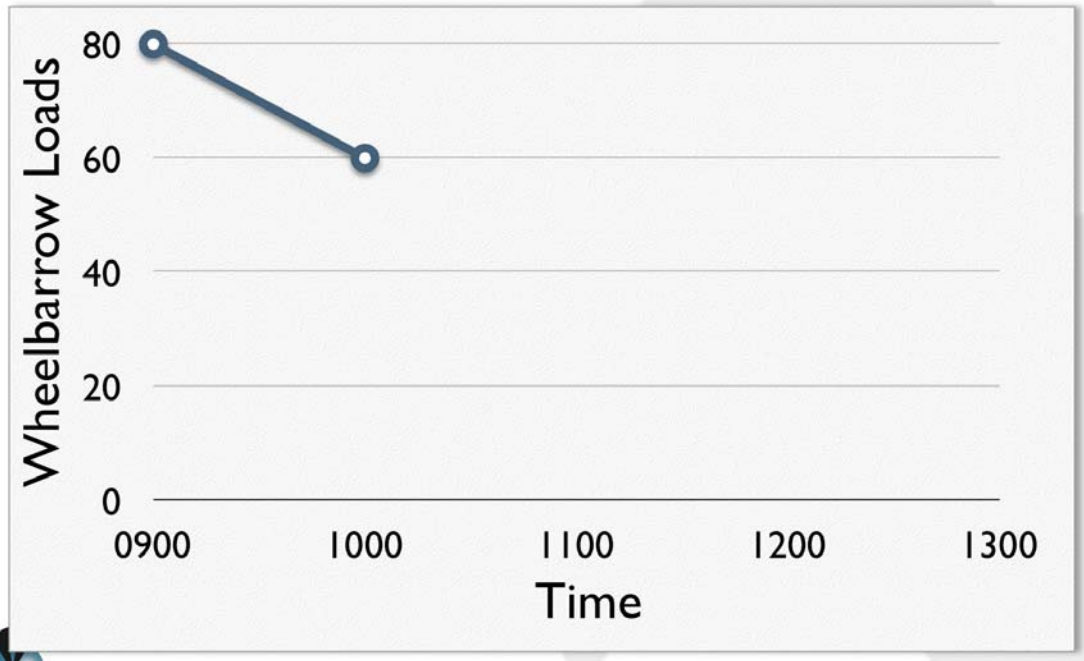
- That you're fed up with software development as a career
- And you decide to go into the landscaping business
- Your first job is moving this pile of rock from the front of my house to the back

How might you estimate this?

- One way:
  - Look at the pile of rock and estimate how many wheelbarrow loads it represents
- After an hour, see how many wheelbarrow loads you've moved then extrapolate the total duration

- I think that's 80 wheelbarrow loads
- After an hour I've moved 20 loads
- So, I'll be done in a total of 4 hours

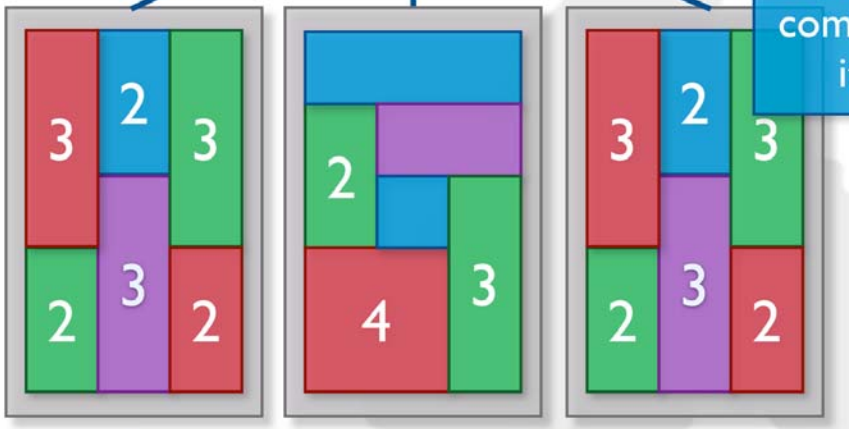
My landscaping



5

- An iteration is a short, constrained period of time
- Typically 1-4 weeks

Velocity is the amount of work planned or completed in an iteration.



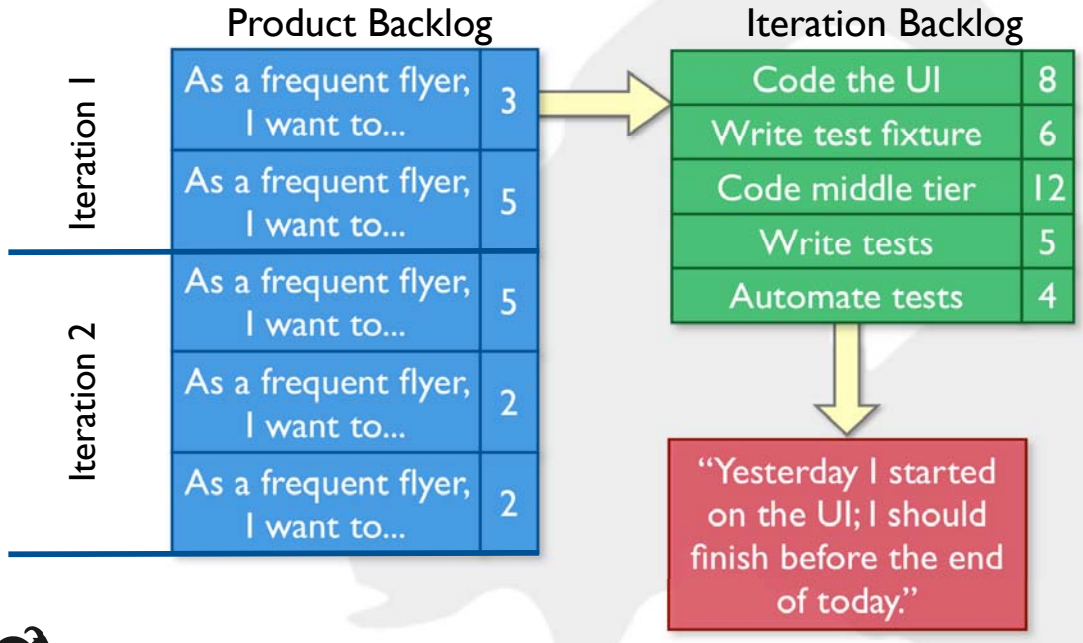
A release typically comprises more than one iteration

6

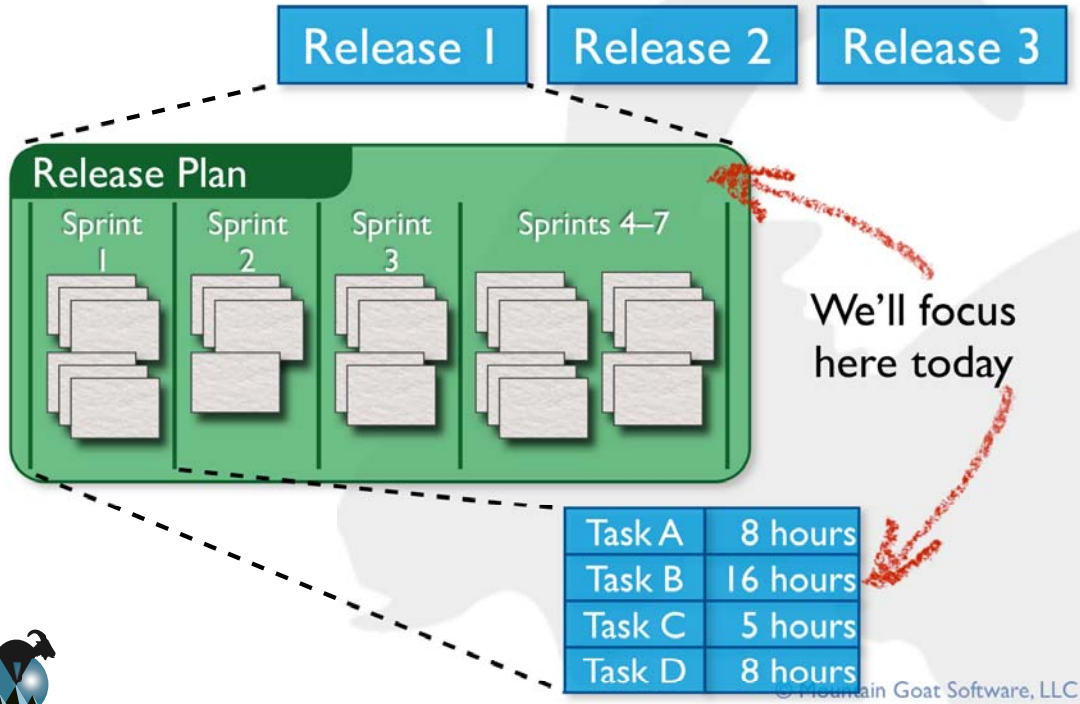
The planning onion

- Agile teams plan at the innermost three levels.
- Others (on the team in the company) plan at the outer levels.

Relating the different planning levels



Product, release, sprint planning



Agenda

- ☐ Estimating
- ☐ Release planning
- ☐ Burndown charts

Story points

- Probably the most commonly used estimating unit among agile teams today
  - Name is derived from agile teams commonly expressing requirements as "user stories"
- Based on a combination of the size and complexity of the work
- Unitless but numerically relevant estimates
  - A 10-point user story is expected to take twice as long as a 5-point user story

Consider these two piles of work

What story point values might we put on these?

Zoo points

Assign "zoo points" to the following breeds

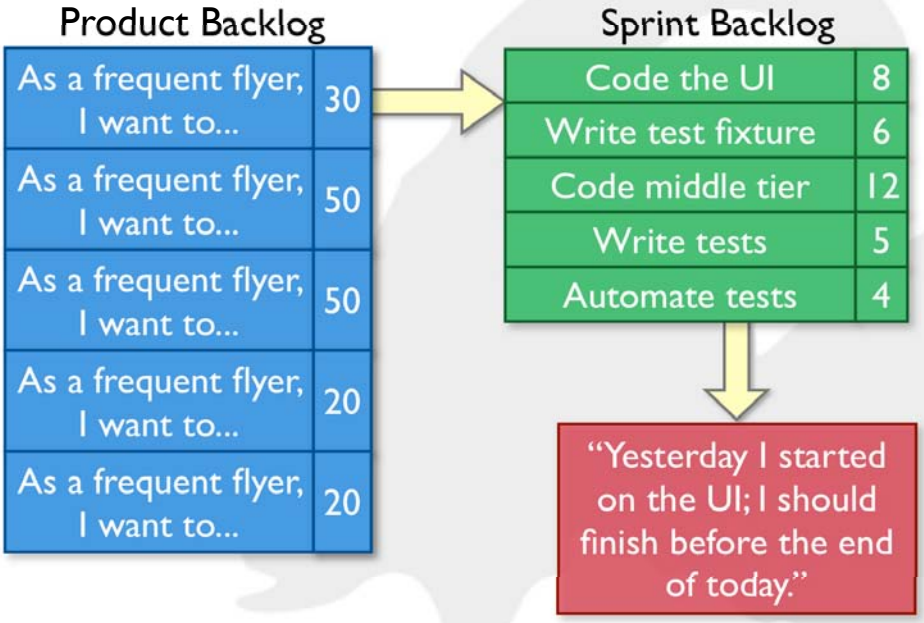
Lion
Kangaroo
Rhinocerus
Bear
Giraffe
Gorilla
Hippopotamus
Tiger

Three key advantages

- Estimating in story points:
  1. Forces the use of relative estimating
     - Studies have shown we're better at this†
  2. Focuses us on estimating the size, not the duration
     - We derive duration empirically by seeing how much we complete per iteration
  3. Puts estimates in units that we can add together
     - Time based estimates are not additive

†Lederer and Prasad, 1998. *A Causal Model for Software Cost Estimating Error* and Vicinanza et al., 1991. *Software Effort Estimation: An Exploratory Study of Expert Performance.*

Comparing apples to apples



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 programmer to join your 20-person startup.	
Create a 60-minute presentation about agile estimating and planning for your coworkers.	
Wash and wax your boss' Porsche.	
Read a 150-page book on agile software development.	
Write an 8-page summary of that book 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?*

Reduces likelihood of anchoring

Control group
- Given a product spec

→ 456 hours

High anchor group
- Given the same product spec
- Told the customer thinks 500 hours is a reasonable estimate but that
  - The customer knows very little about the implications of his spec on the estimate
  - You shouldn't let his number influence you

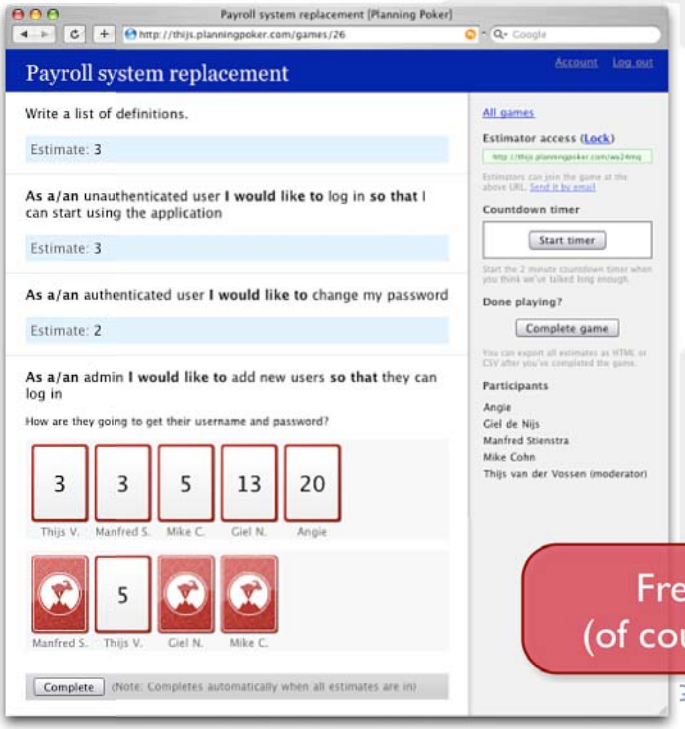
→ 555 hours

Low anchor group
- Same as high but customer thinks 50 hours

→ 99 hours

Source: *How to avoid impact from irrelevant and misleading information on your cost estimates*, Magne Jørgensen and Stein Grimstad, Simula Research Laboratory, Simula Research Labs Estimation Seminar, Oslo, Norway 2006.

www.planningpoker.com



Free (of course)

Agenda

- [x] Estimating
- [ ] Release planning
- [ ] Burndown charts

Release planning

Purpose

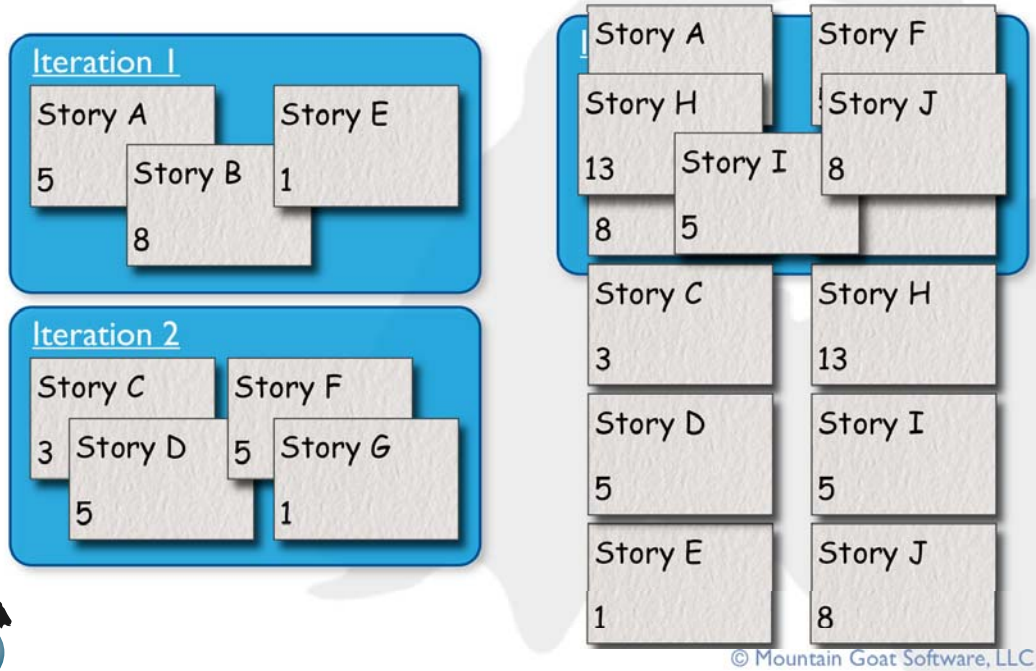
To answer questions such as:

- How much will be done by 30 June?
- When can we ship with this set of features?
- How many people or teams should be on this project?

Inputs

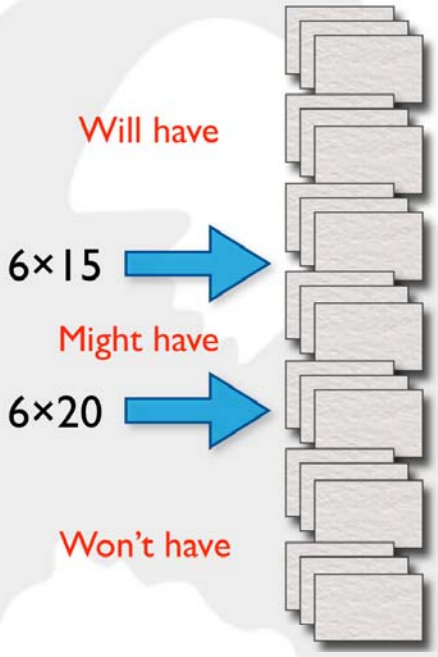
- Velocity
- The length of the project
- Prioritized product backlog

An example with velocity=14



What if I'm given a fixed date?

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



What if I'm given a fixed scope?

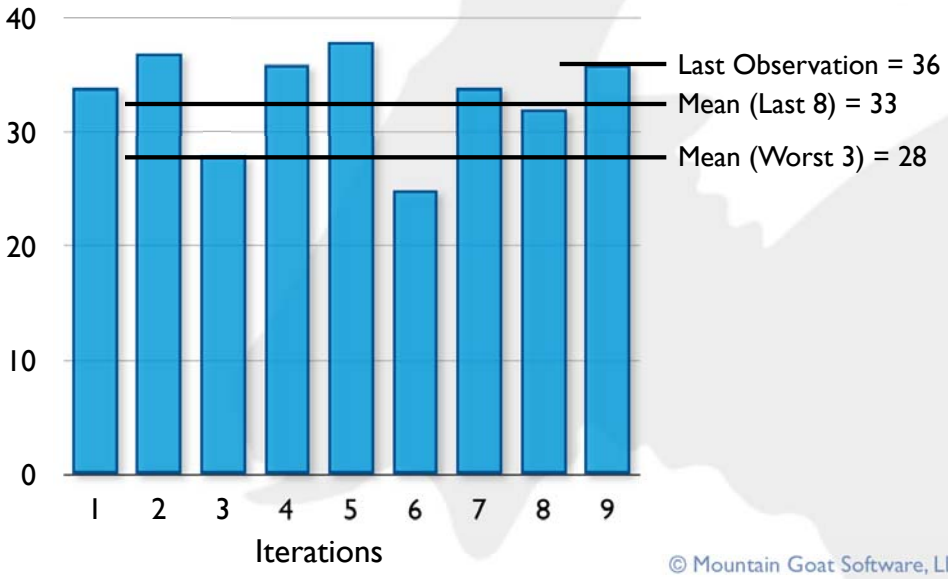
Total story points desired	120
Low velocity	15
High velocity	20

120÷20=

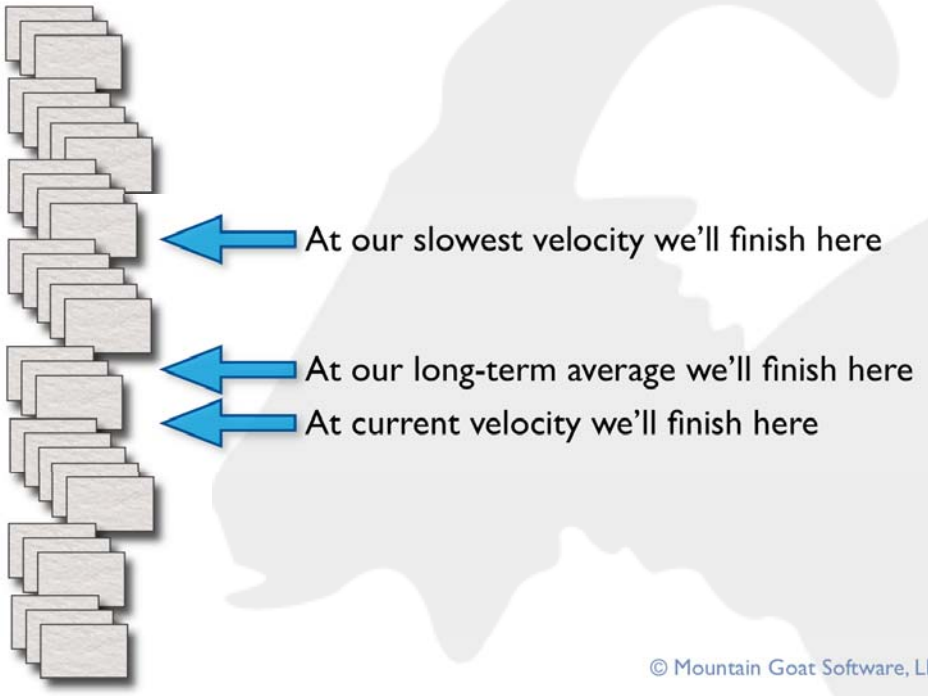
120÷15=

Updating the release plan

- Use multiple views of observed velocity



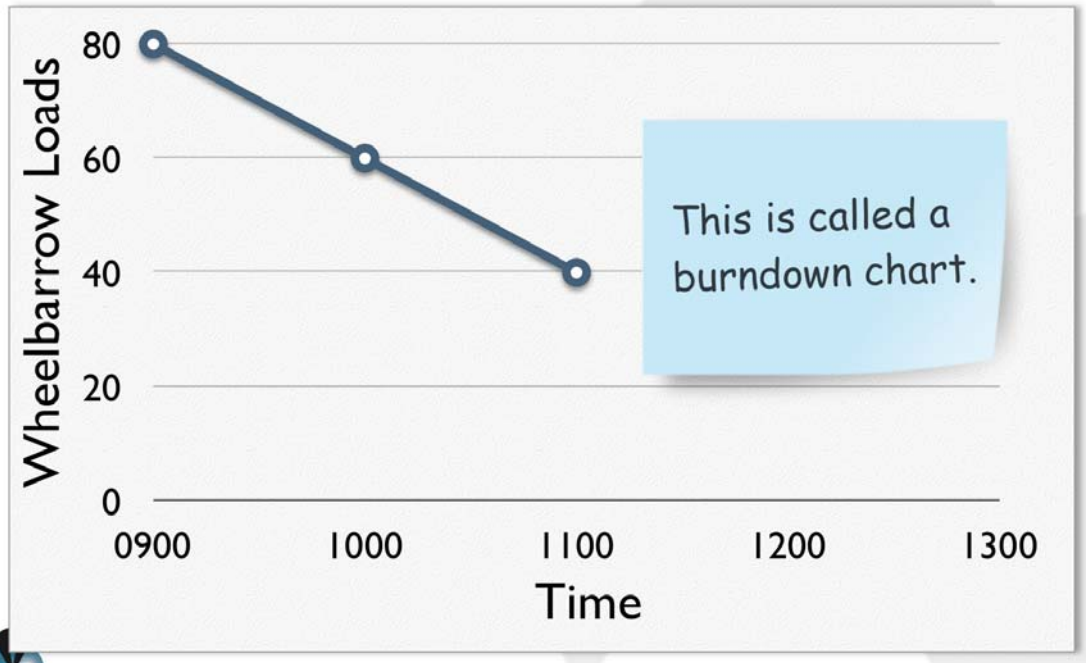
Extrapolate from velocity



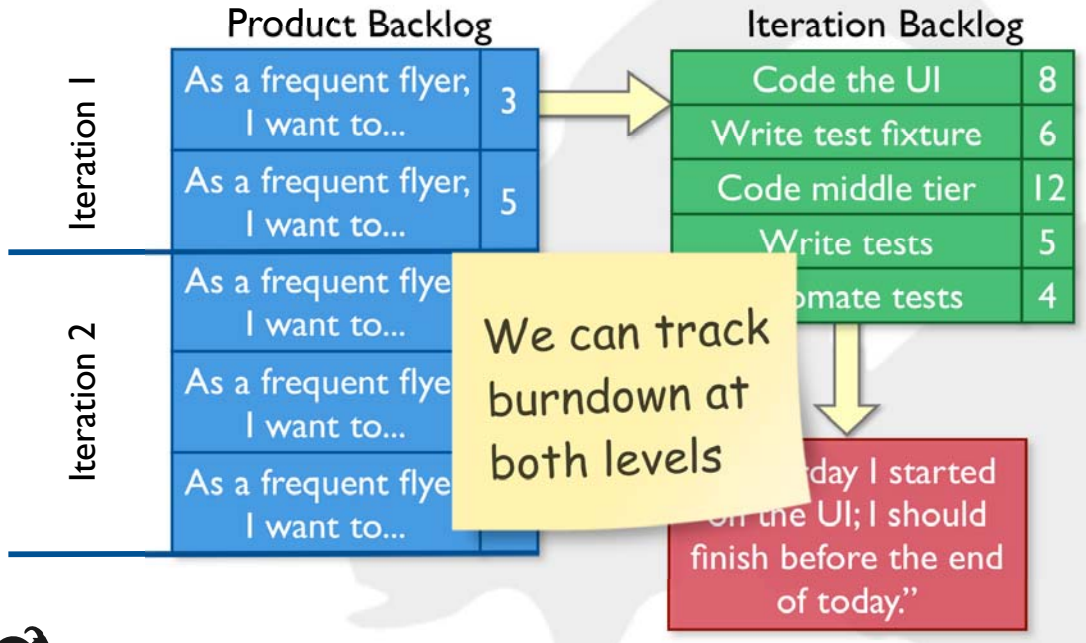
Agenda

- ☑ Estimating
- ☑ Release planning
- ☐ Burndown charts

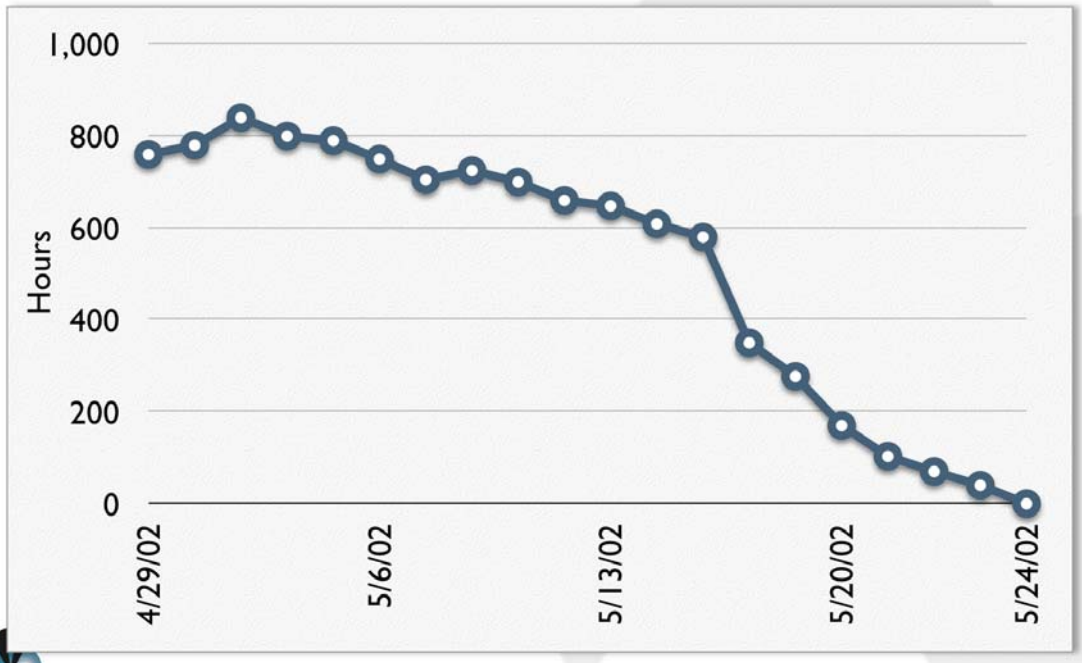
How's my landscaping coming?



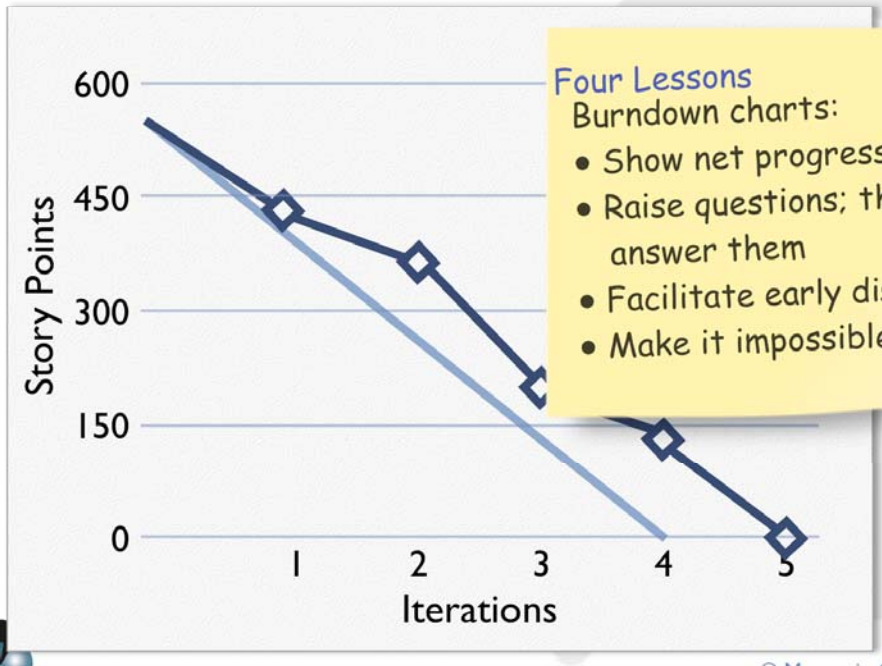
Remember the different levels?



An iteration burndown chart



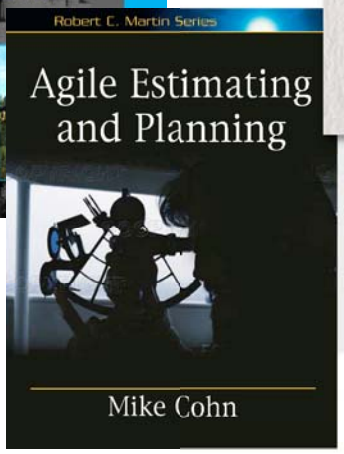
A release burndown chart



Four Lessons
Burndown charts:

- Show net progress
- Raise questions; they don't answer them
- Facilitate early discussions
- Make it impossible to lie

Mike Cohn contact info



mike@mountaingoatsoftware.com

www.mountaingoatsoftware.com

(720) 890-6110 (office)

(303) 810-2190 (mobile)

MOUNTAIN GOAT SOFTWARE