Agile Estimating and Planning

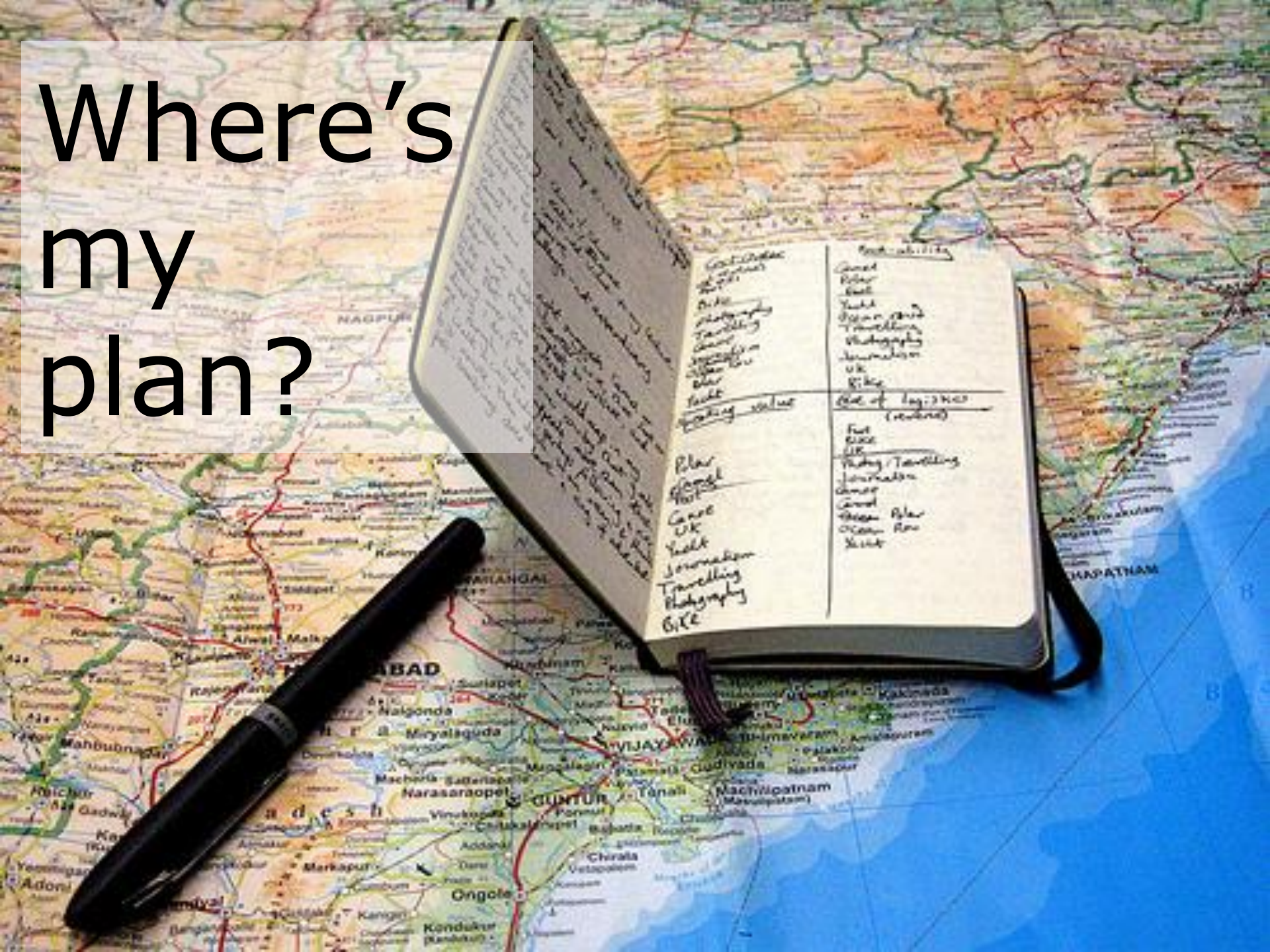
Special thanks to

Mike Cohn,

founder of Mountain Goat Software, who gave us permission to use this material.

mountaingoatsoftware.com

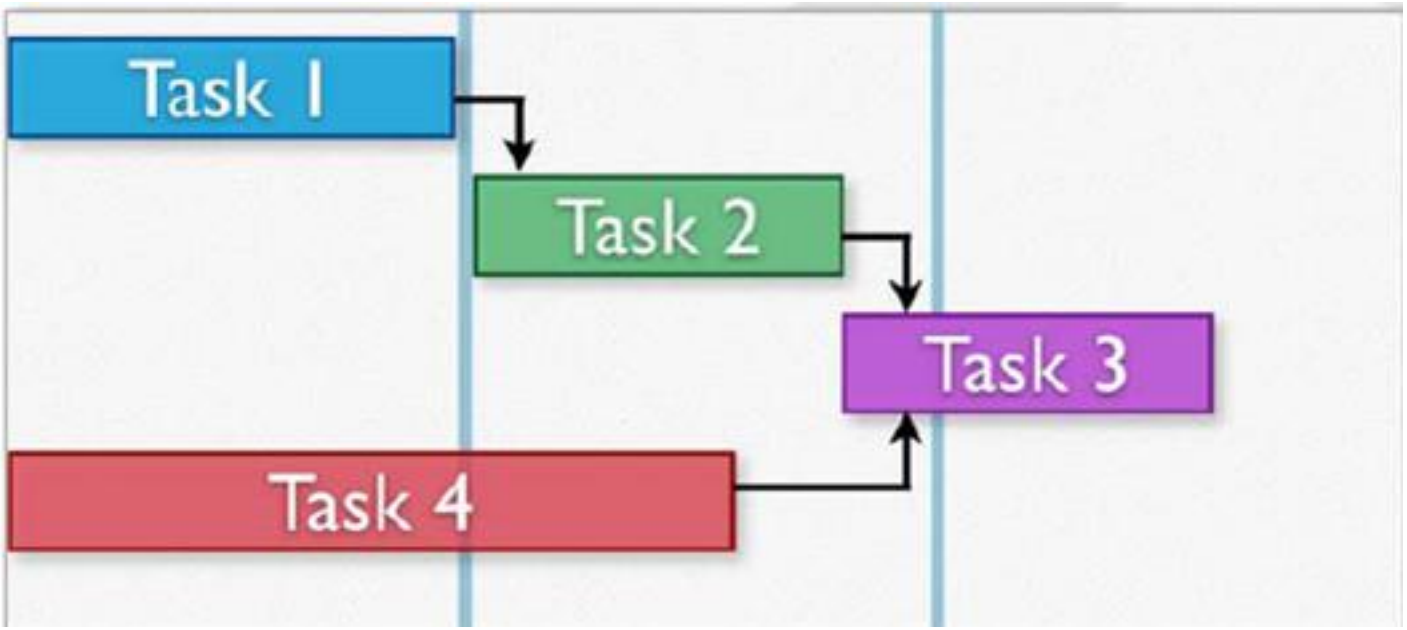
720-890-6110


Where's
my
plan?

Why plans go wrong…

Lateness passed down schedule

- Task 3 late when?

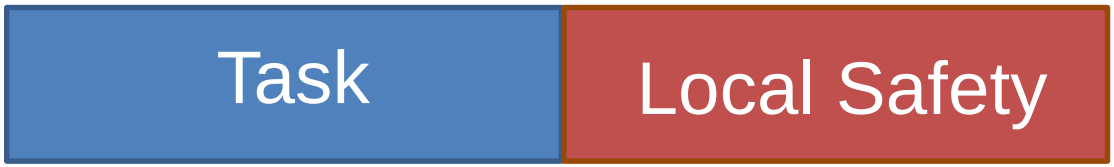


Student syndrome

- Have you ever started a term paper the night before it was due?

Student syndrome

- It is based on estimates like this:



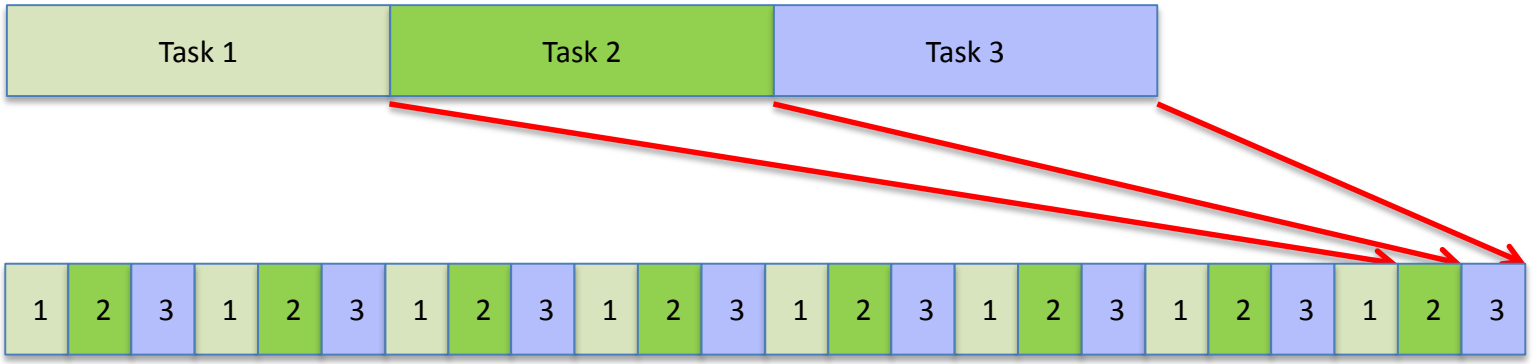
Student syndrome

- It is based on estimates like this:

- But we do this:

Multitasking

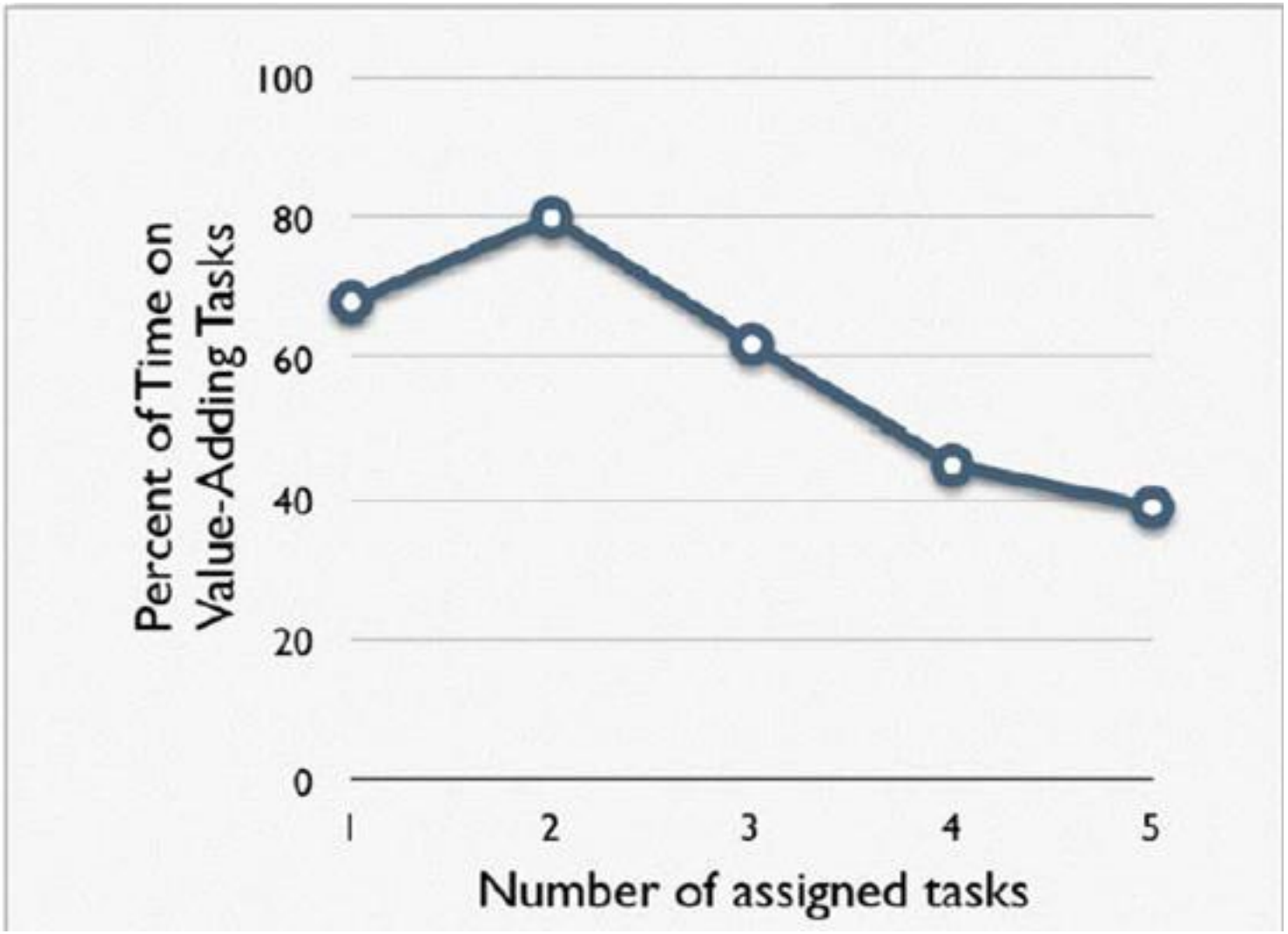
- 20-30% time increase to do each task
- Greater loss on complex tasks



To do two things at once is to do neither.

- Publilius Syrus

The Effects of Multitasking



What’s a good plan?

“It is a bad plan that admits to no modifications.”

-- Publilius Syrus (ca. 42 BCE)

What's a Good Plan?

- Supports reliable decision-making

Go from:

- Will be done in fourth quarter
- … in November
- …. November 7

A Plan is NOT a Commitment

It is a current view.

If one sticks to the idea that once set, a plan should not be changed, a business cannot exist for long.

--- Taiichi Ohno

In preparing for battle I have always found that plans are useless, but planning is indispensable.

--- Dwight D. Eisenhower

A Plan is NOT a Commitment

- If plans are commitments, then we are committing to decisions made **when we were the most ignorant** (recall the cone of uncertainty, 5%)
- Measuring conformance to plan is measuring the wrong thing because the plan will change

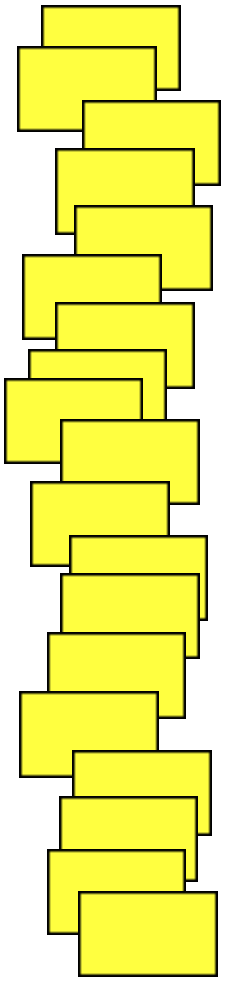
What makes planning Agile?

- More focused on planning than the plan
- Encourages change
- Plans are easily changed
- Done throughout the project

What's an Agile plan?

In the form of **three backlogs**:

- **Product** Backlog
  Epics and Themes for Product
- **Release** Backlog
  Release Theme and User Stories
- **Sprint** Backlog
  User Stories and tasks planned for the Sprint (iteration)



Product Planning

Product Planning

✓ Product Backlog:

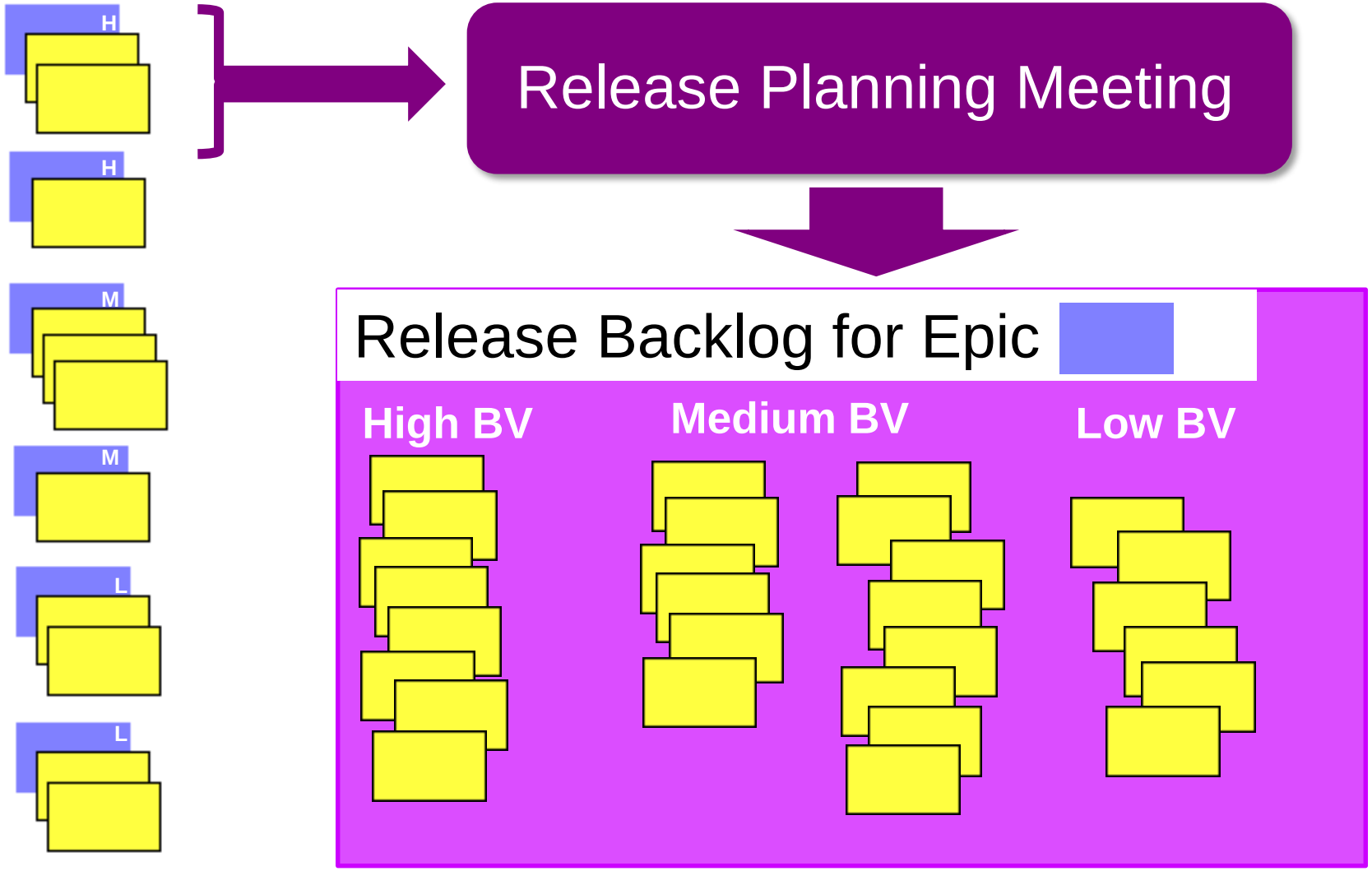
- Develop Epic User Stories
- Prioritize based on Business Value
- Define release themes
- Place Epics into releases

Who: **Stakeholders, Business and Team**

Release Planning

Release Planning

Product Backlog (Prioritized Epics and Themes)



Release Planning, part 1

✓ Release Backlog:

- Develop User Stories for ONE release
- Prioritize based on Business Value

Who: **Stakeholders, Business and Team**

Release Planning, part 2

- Estimate Story Points on User Stories

Who: **Delivery Team**

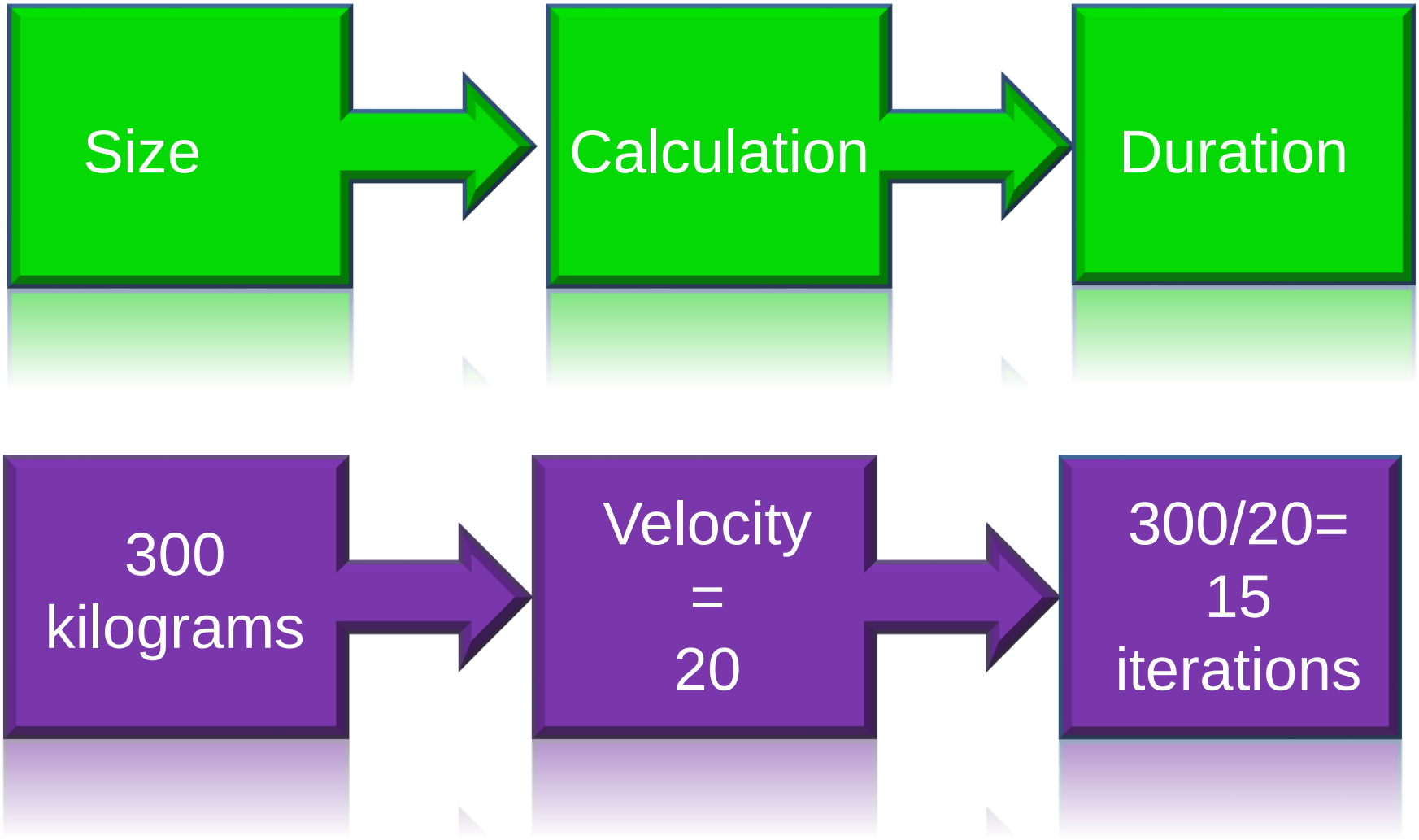
Agile Estimating

Exercise: How long will it take …

…. to **read** the latest Harry Potter book?

…. to **drive** to Austin, TX?

Estimate Size by Deriving Duration



measure of **size**

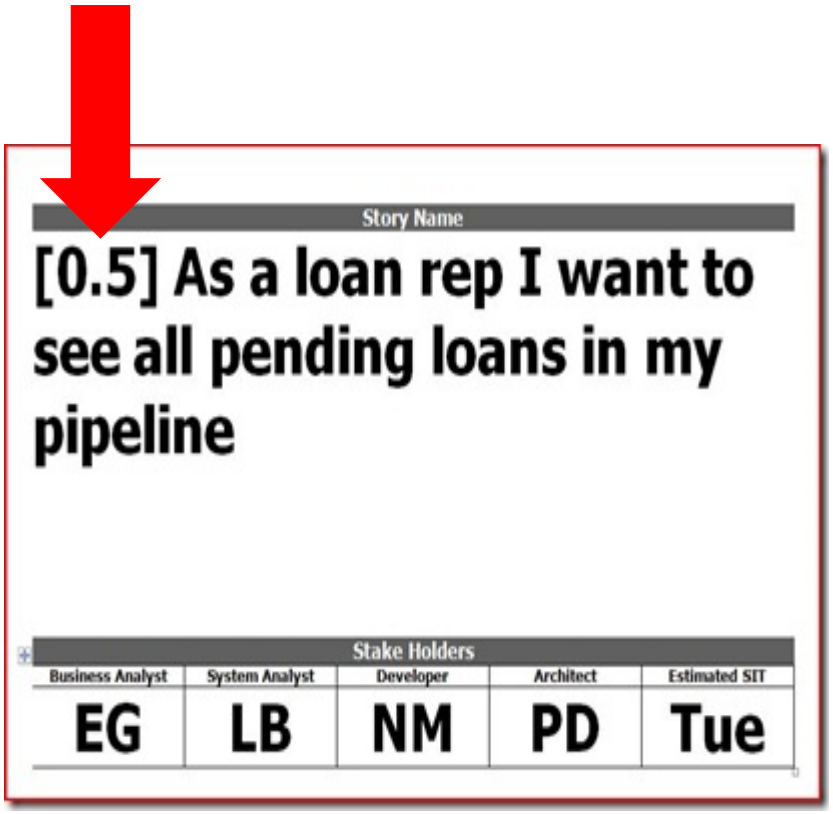
Traditional Measure of Size

Traditional measures of size:

Lines of Code

Function Points

Agile Measure of Size



Story Name
[0.5] As a loan rep I want to see all pending loans in my pipeline

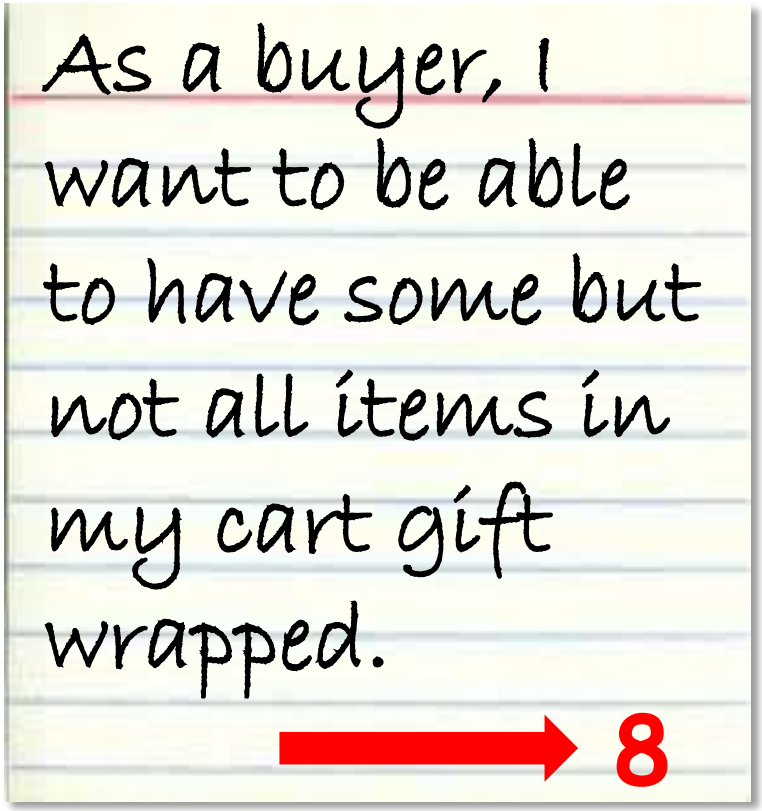
Stake Holders				
Business Analyst	System Analyst	Developer	Architect	Estimated SIT
EG	LB	NM	PD	Tue

Agile measures of size:

Story Points

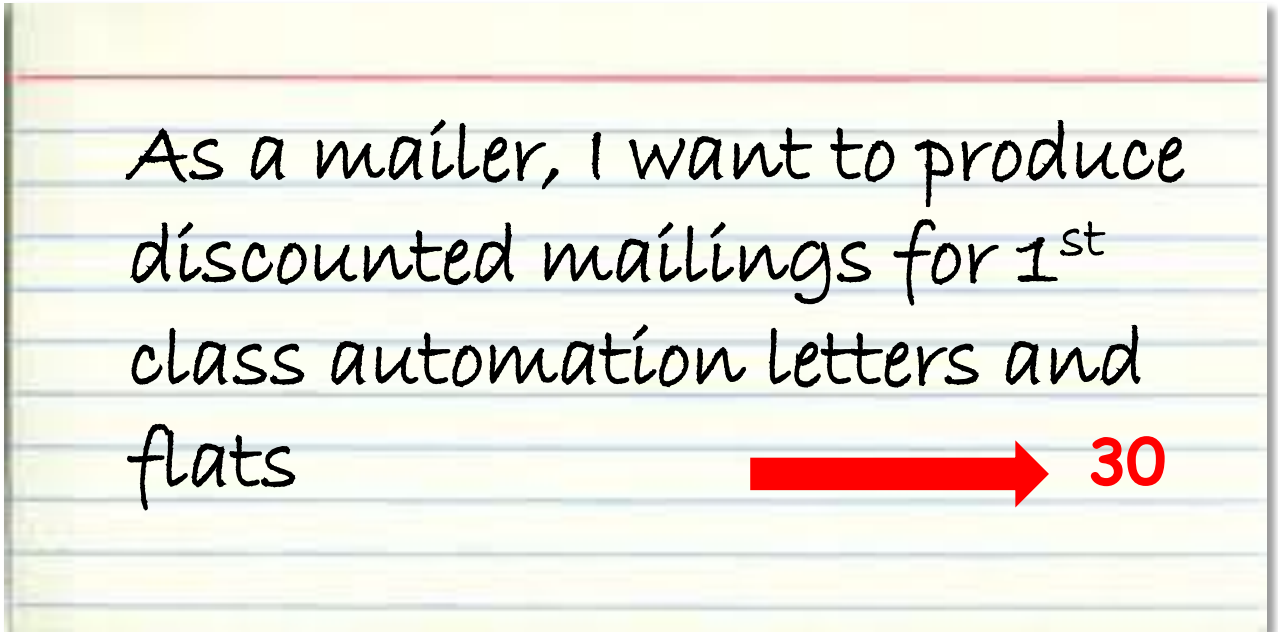
Story Points

- The "**bigness**" of a task
- Points are **unit-less**
- Influenced by
  - How hard it is
  - How much there is



Story Points

- Relative values are what is important:
  - A login screen is a 2.
  - A search feature is an 8.
- Basic math properties hold, e.g., 5+5 = 10



Exercise: Dog Points

Assign "dog points" to the following breeds:

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

Why Story Points

- Help drive cross-functional behavior
- Estimates do not decay
- Pure measure of size
- Estimating is typically faster

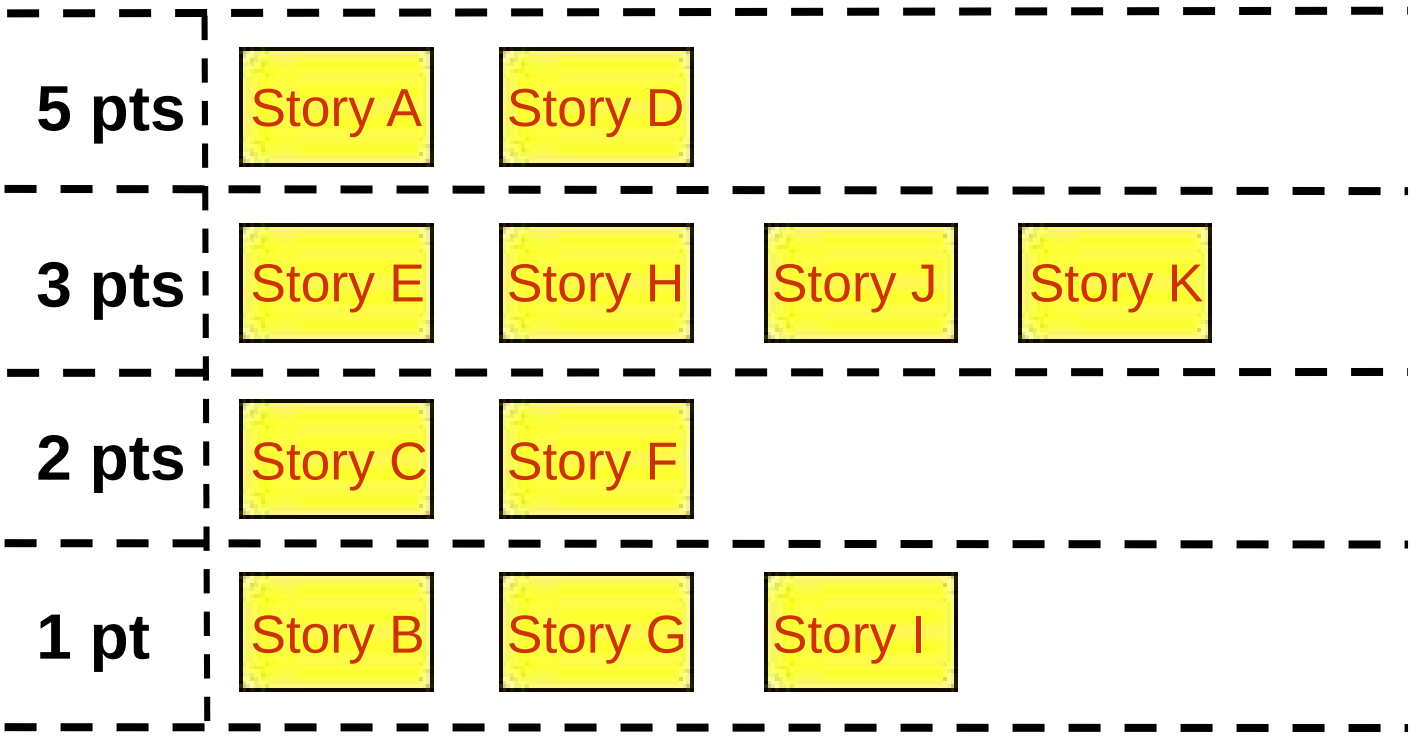
Estimating Tips and Techniques

Estimate by Analogy

- Compare one user story to another
  - “This story is like that story”
- Don’t use a single gold standard
  - Triangulate: Compare the story to be estimated to other estimated stories

Triangulation

- Compare a story to similar stories
- Group like-sized stories

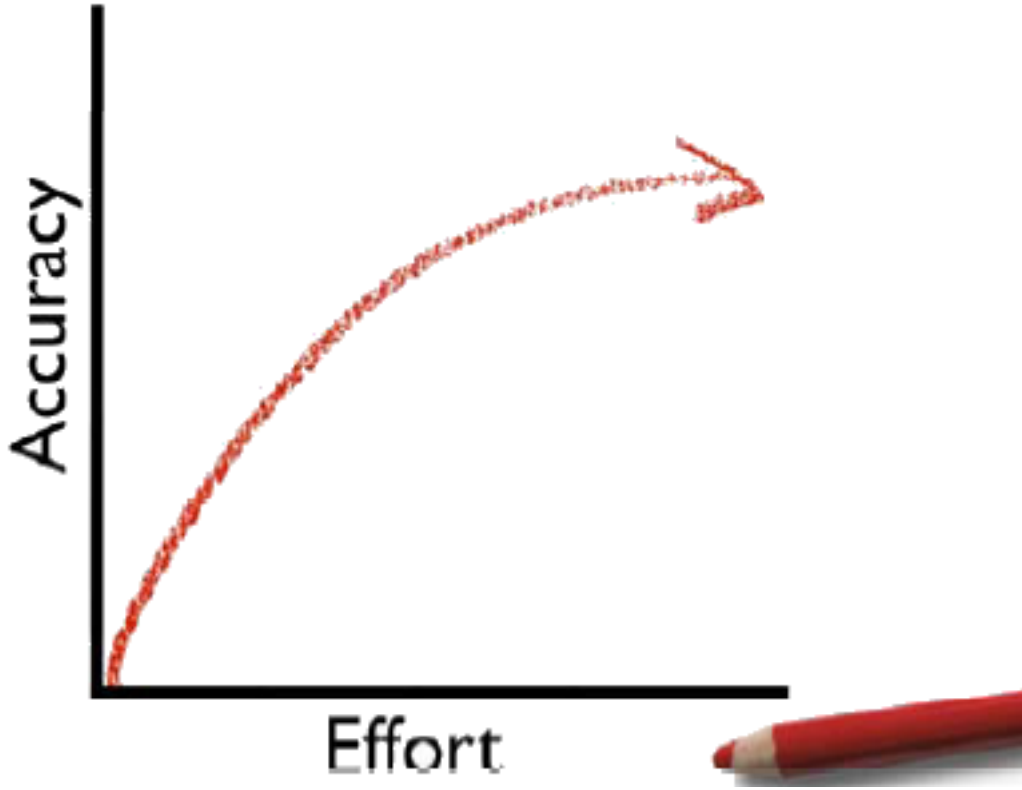


Disaggregation

- Breaking a big story into smaller stories
  - Can’t estimate the big story – don’t know enough
  - Break into smaller estimatable stories
- Don’t get **too** small
  - Small errors can add up
  - Tasks slip through cracks

How much effort?

- A little effort helps a lot
- A lot more effort only helps a little



Use the Right Units

- Can you distinguish a 1-point story from a 2?
- Can you distinguish a 17 from an 18?
- Use units that makes since, such as
  - 1, 2, 3, 5, 8, 13
  - 1, 2, 4, 8

Use Planning Poker Cards

Planning Poker

An iterative approach to planning:

- Team members use planning poker cards to make an estimate
- Product owner reads and discusses a story
- Each team member selects a card that's her estimate, placing it face down
- When all cards are face down, turn them over
- Outliers are discussed
- Continue estimating and discussing until consensus reached

Planning Poker Example

Estimator	Round 1		
Susan	3		
Vadim	8		
Ann	2		
Chris	5		

Planning Poker Example

Estimator	Round 1	Round 2	
Susan	3	5	
Vadim	8	5	
Ann	2	5	
Chris	5	8	

Planning Poker Example

Estimator	Round 1	Round 2	Round 3
Susan	3	5	5
Vadim	8	5	5
Ann	2	5	5
Chris	5	8	5

Exercise: Agile Estimating

Item #	Product Backlog Item	Estimate
1	Read a high-level 10-page overview of agile software development in *People* magazine.	
2	Read a densely written 5-page research paper about agile software development in an academic journal.	
3	Write the product backlog for a simple eCommerce site that sells only clocks.	
4	Recruit, interview, and hire a new member for your team.	
5	Create a 60-minute presentation about agile estimating and planning for your co-workers.	
6	Wash and wax your boss' Porsche.	
7	Read a 150-page book on agile software development.	
8	Write an 8-page summary of that book for your boss.	

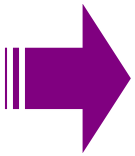
Planning Poker Guidelines

- Only **delivery team members** estimate their user stories
- Outliers explain their estimates
- Everyone's opinion is heard
- It's a **conversation**! Not a commitment.

Release Planning, part 2

Estimate Story Points on User Stories

Who: **Delivery Team**



Exercise: Estimate your user stories using story points.

Sprint Planning

Sprint Planning

- Determine user stories in sprint
- Define “done, done, done”
- Team commits
- Create an information radiator

Who: **Delivery Team**

Sprint Duration

How long can you keep business changes out of the sprint?

Sprint Planning Session

- Held at the beginning of a new Sprint
- Chaired by the Scrum Master
- Attended by all including Key Stakeholders
- Update the Product Backlog with new user stories
- Select highest priority items in the backlog based on Business Value And Optimization of team resources

Creating a Sprint Backlog

Team may break down stories into tasks (all tasks must be done to demo User Story)

As a card holder, **I want to** be able to withdraw funds from my account **so I can** have cash available to me

- Make a withdrawal with sufficient funds
- Make a withdrawal with insufficient funds (exception)
- Make a withdrawal when specified demonization not available (exception)

Team Defines ‘Done3’

Consider:

- No showstoppers
- All errors that the team has not agreed to are removed
- Code is unit tested, function tested, system tested, performance tested, tested end-to-end
- A meaningful stakeholder review has been conducted

Can ‘Done3’ really be done?

This puts a high premium on:

- Valuable, maintained, nested automation
- Appropriate coverage (e.g. 80%)
- True test-driven development
- Avoiding technical debt
- Continuous integration
- Really understanding what quality looks like

Example: ‘Done4’

patientkeeper®

Jeff Sutherland (co-creator of Scrum), said while @ PatientKeeper:

“It took us four years to get *done, done, done, done*.”

Patientkeeper provides safety critical patient management software

Example: ‘Done4’

patientkeeper®

What does “Done”, “Done”, “Done” , “Done” mean?

- It is fully developed
- It is fully tested
- It has no Severity 1s or 2s
- It has been deployed before a release in a client environment

Example: ‘Done4’

patientkeeper®

They ship

- A major release every 3 months
- A minor release every month
- And minor updates once a week

Consider the competitors, teams of 600-700 developers and they cannot achieve the work flow Patientkeeper does.

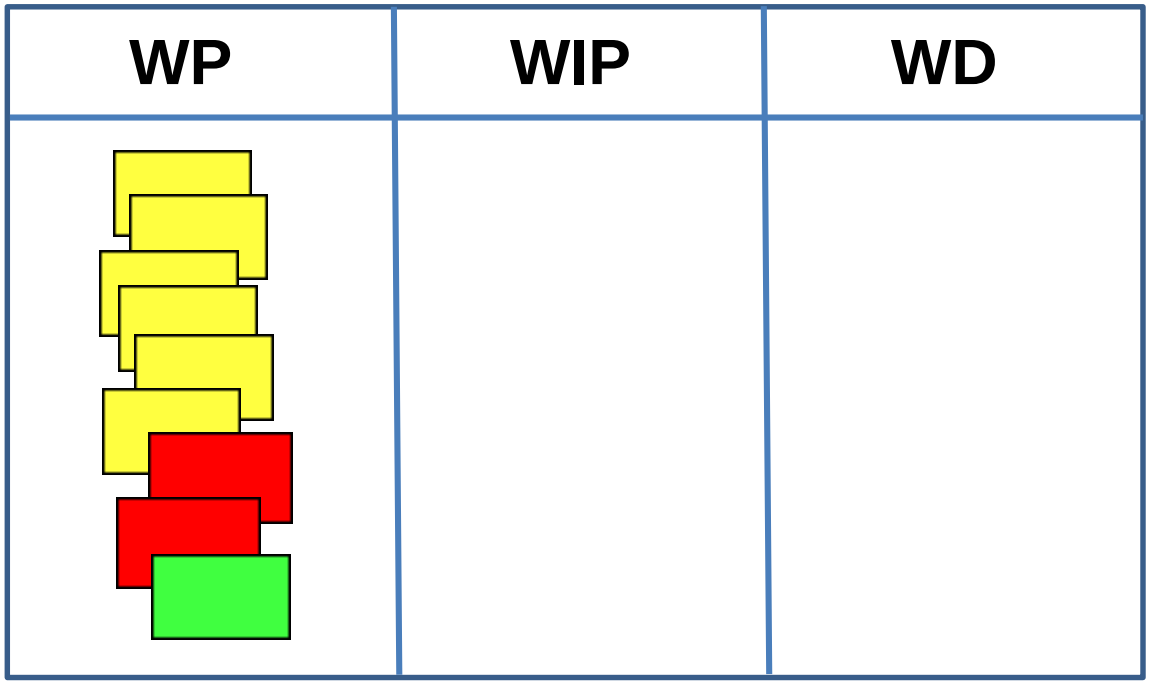
Information Radiator

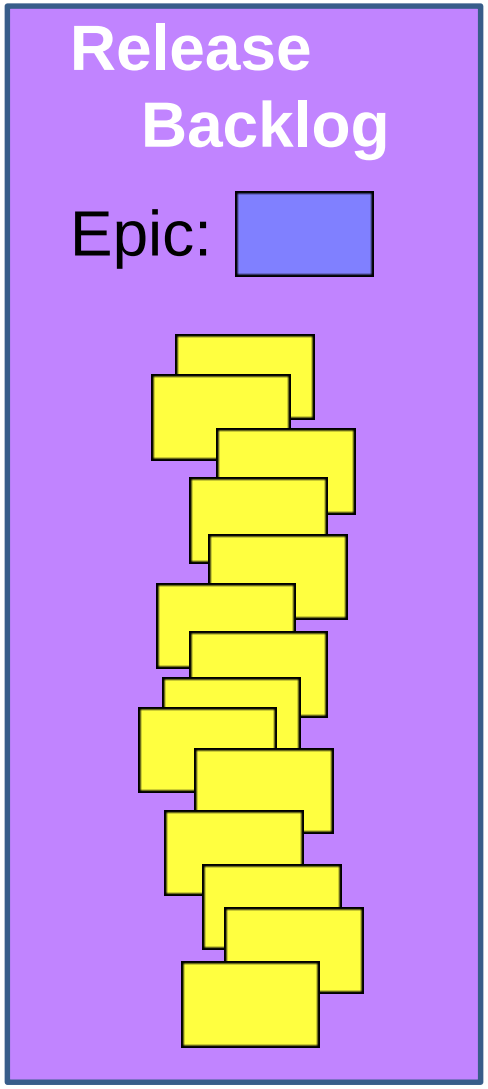
Information Radiator

- Visual representation of progress
- Display of:
  - **Work Planned** (Product, Release and Sprint)
  - **Work in Progress**
  - **Work Done**
  - User Stories to mitigate risk
  - User Stories to gather information to make future decisions

Information Radiator

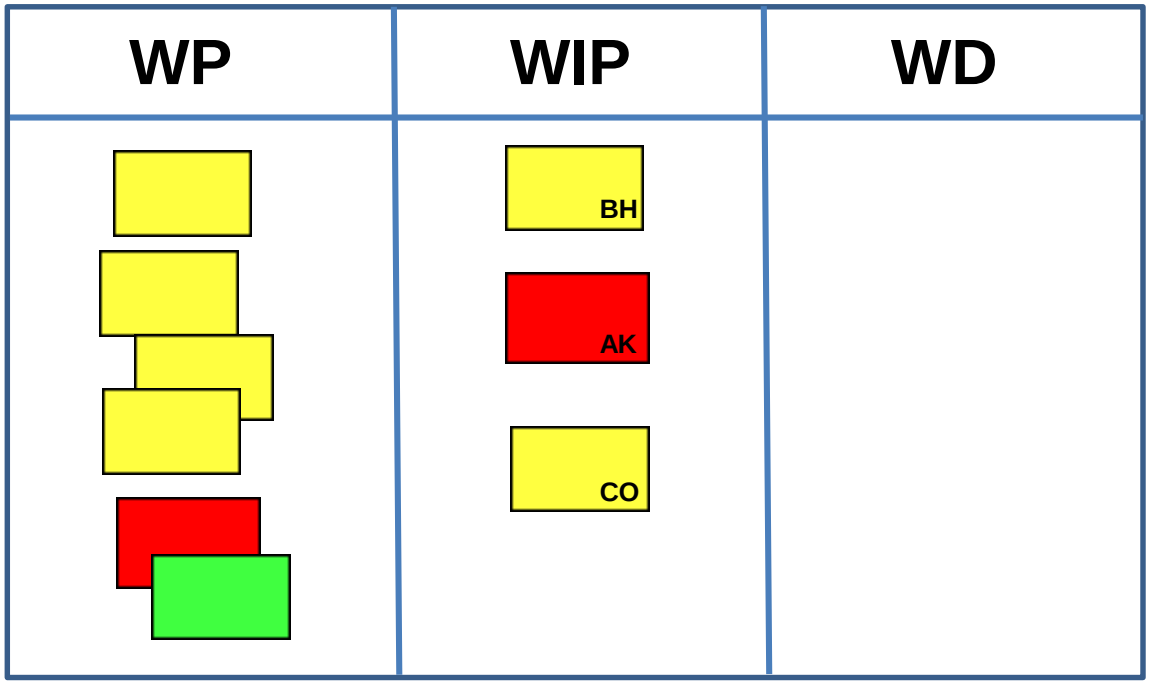
Sprint 2

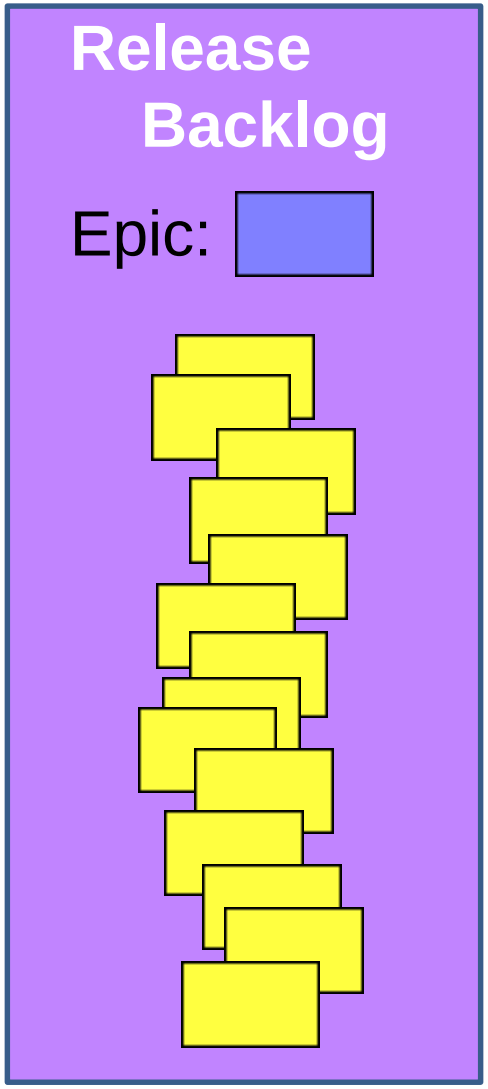




Information Radiator

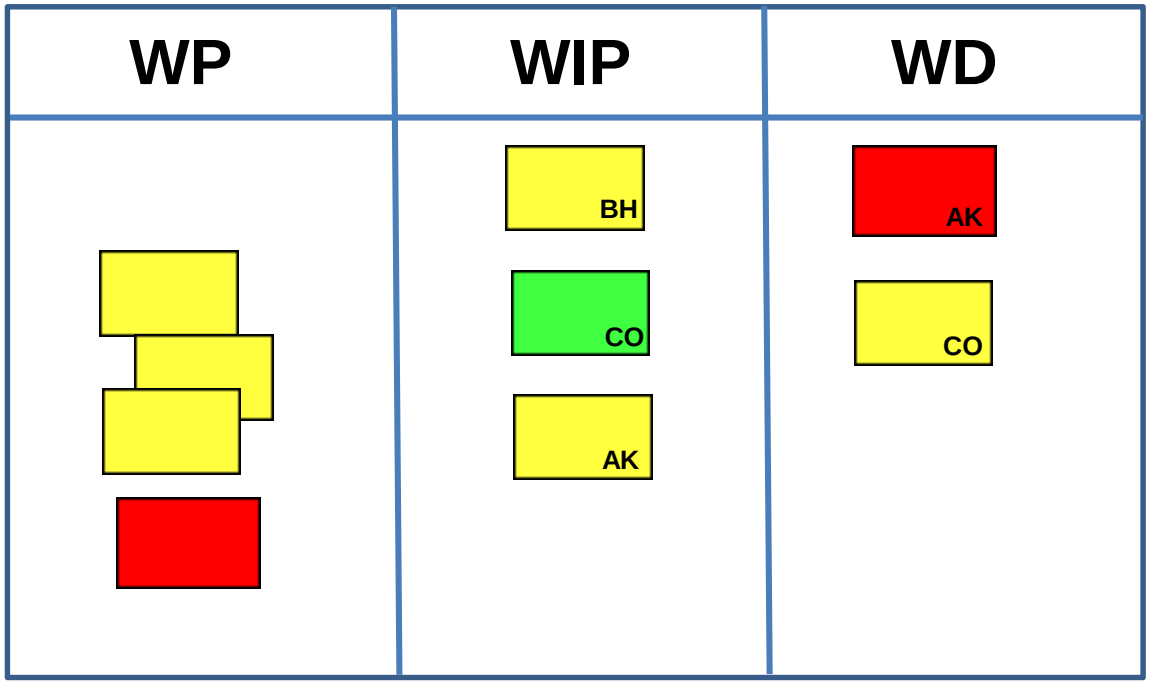
Sprint 2

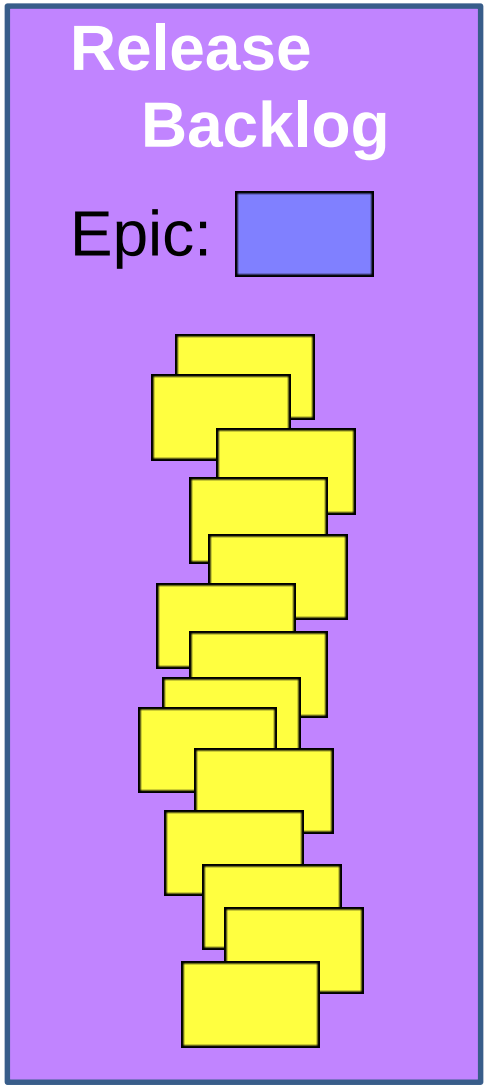




Information Radiator

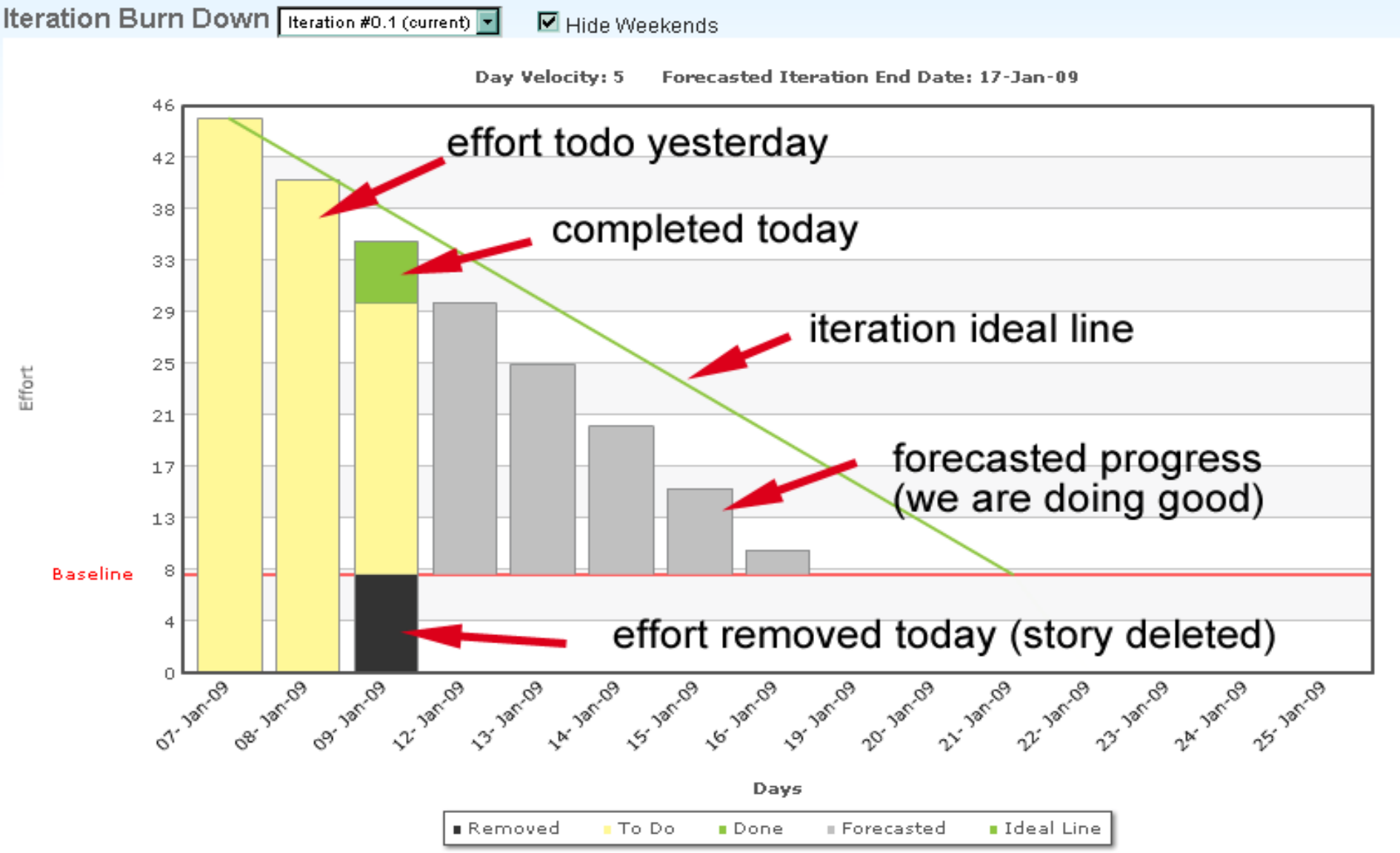
Sprint 2





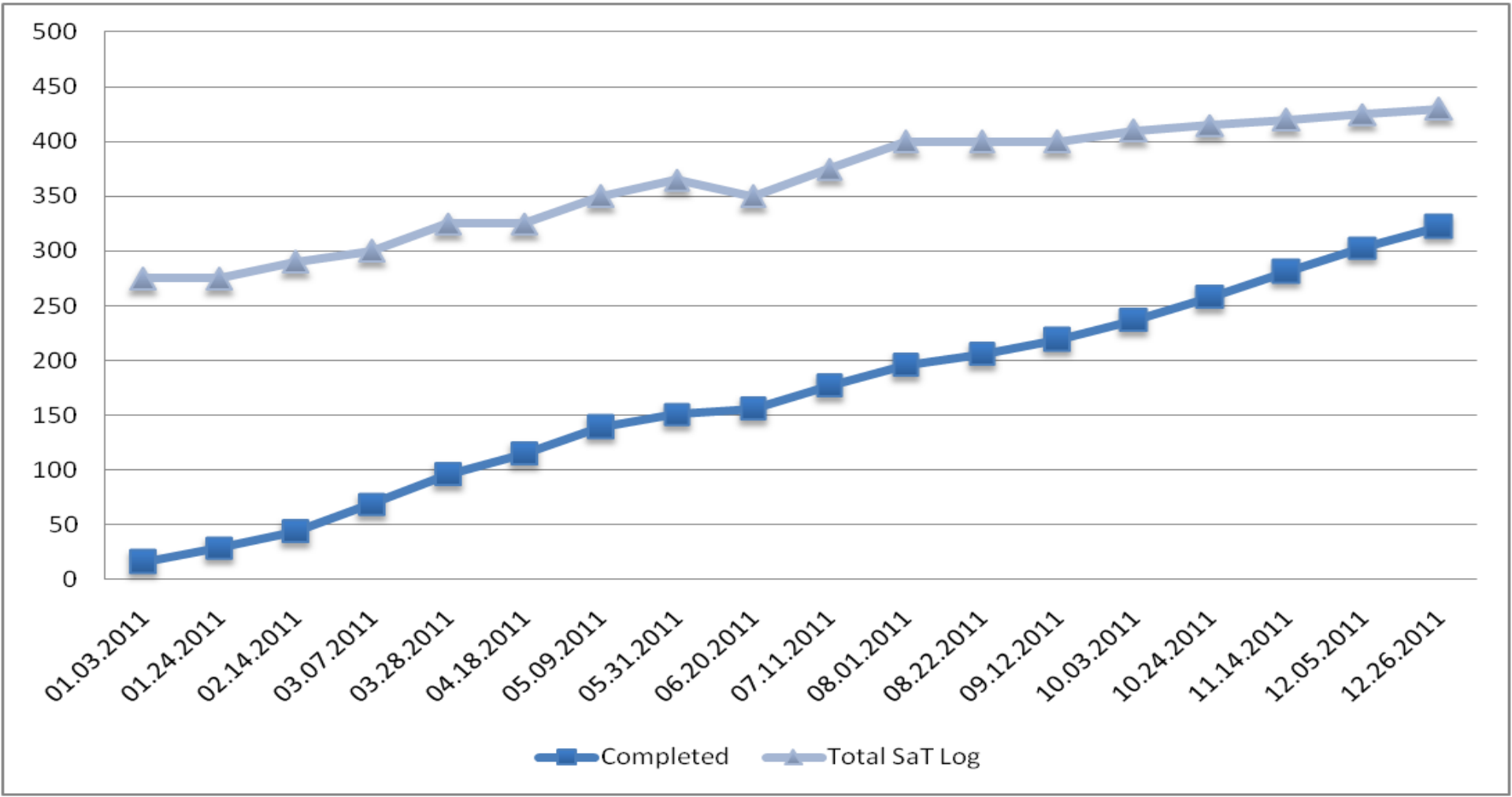
Example: Information Radiator

Sprint Burndown Chart



Release Burn Up Chart

Done for sprint results against total release backlog.



Exercise: Information Radiator

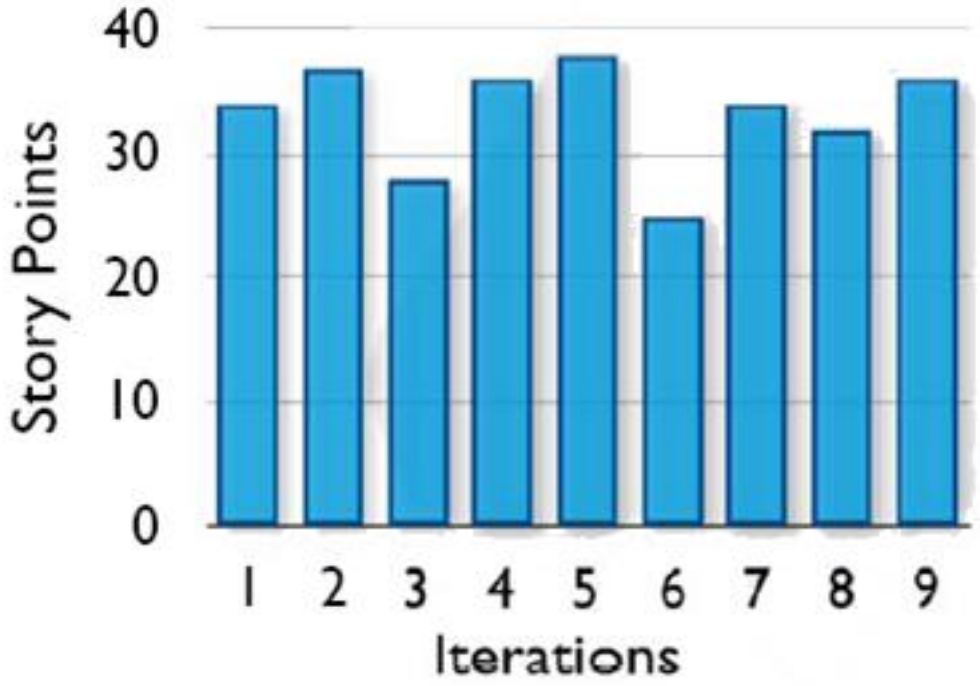
Create an information radiator

- Place all sprint stories onto the Information Radiator under Work Planned column
- Add a section for “New User Stories”
- Decide what ‘Done’ is

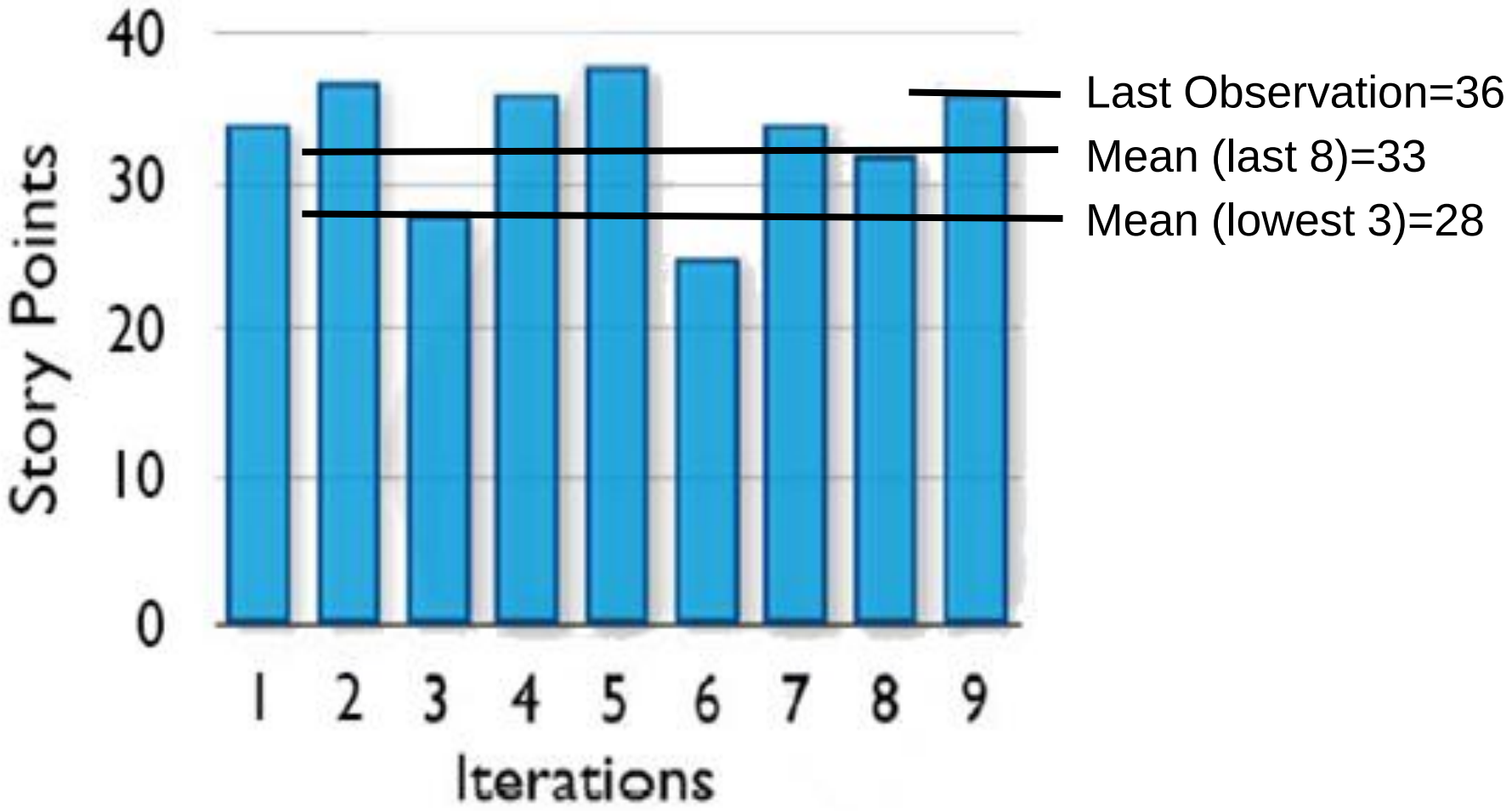
Velocity

Velocity

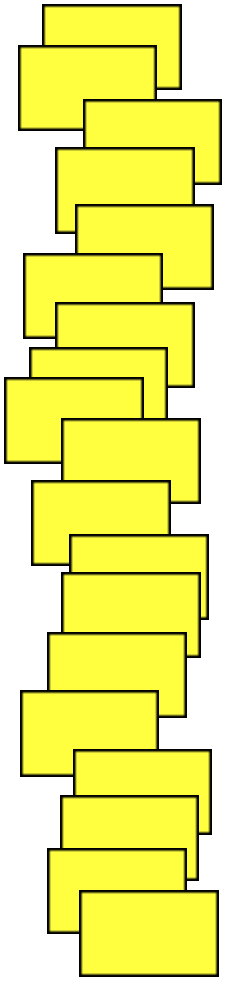
- **Long-term** measure of work completed in iterations
- A **GUIDE** not a goal**.**



Track Velocity Multiple Ways



Extrapolate from Velocity



Assume five iterations left

Finish here at **lowest** velocity: 5x28

Finish here at **average** velocity: 5x33

Finish here at **current** velocity: 5x36

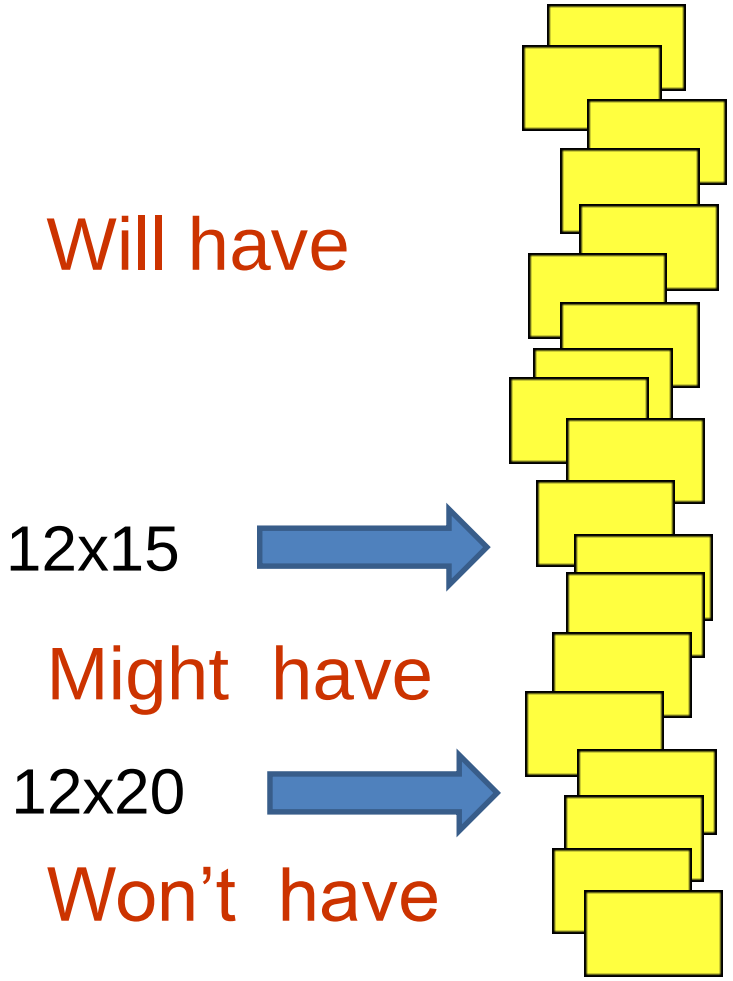
How much can I get by <date>?

Fixed Date ‘Planning’

- Determine how many iterations you have
- Estimate high and low velocity
- Low velocity times iterations = ‘will have’ stories
- High velocity times iterations = ‘might have’
- All the rest = ‘will not have’

Fixed Date ‘Planning’ Example

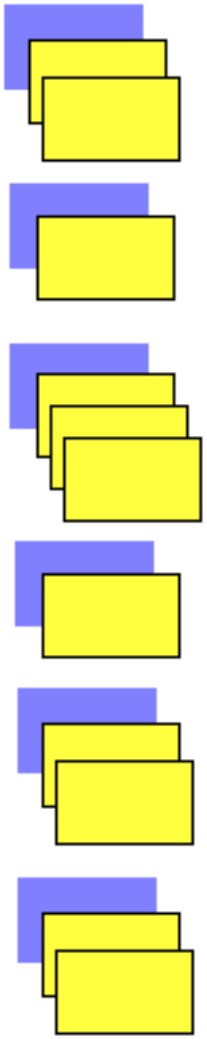
Desired Release Date	30 June
Start Date	1 January
Number of Iterations	12 (two weeks)
Low Velocity	15
High Velocity	20



Agile Estimating and Planning Summary

Product Planning

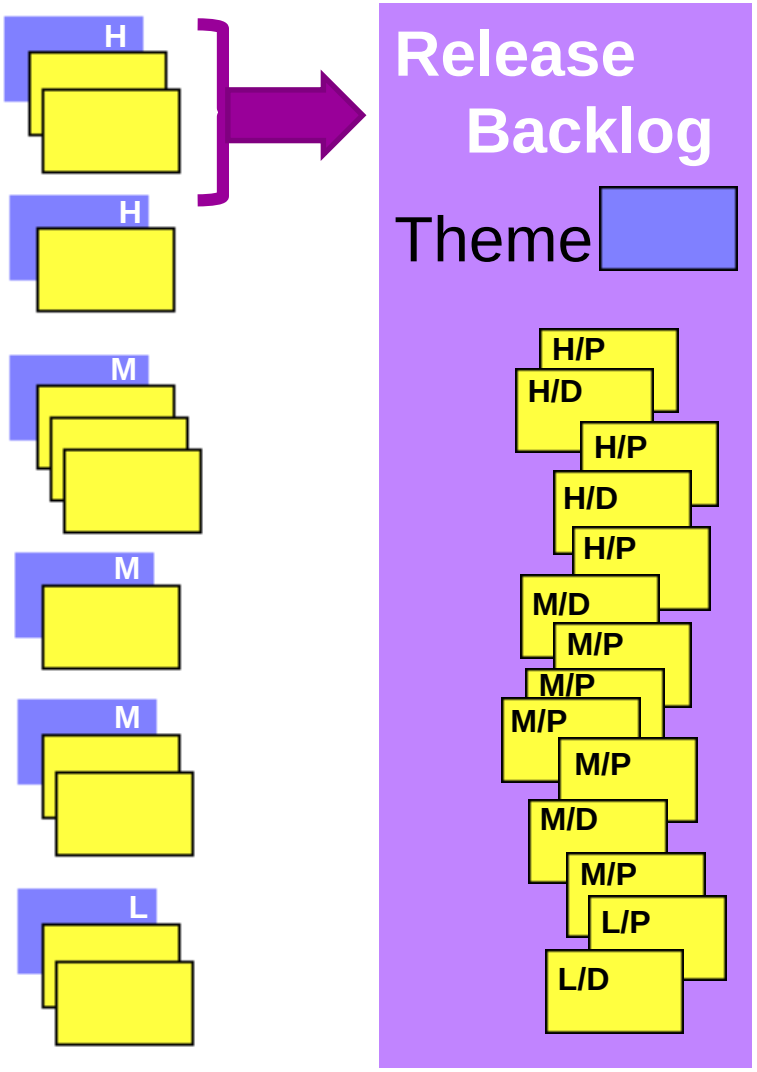
Product Backlog



Input	SOW, Purpose
Output	**Prioritized Product Backlog** (Epics and Release Themes)
Who	Business, Product Owner, Dev Team representatives

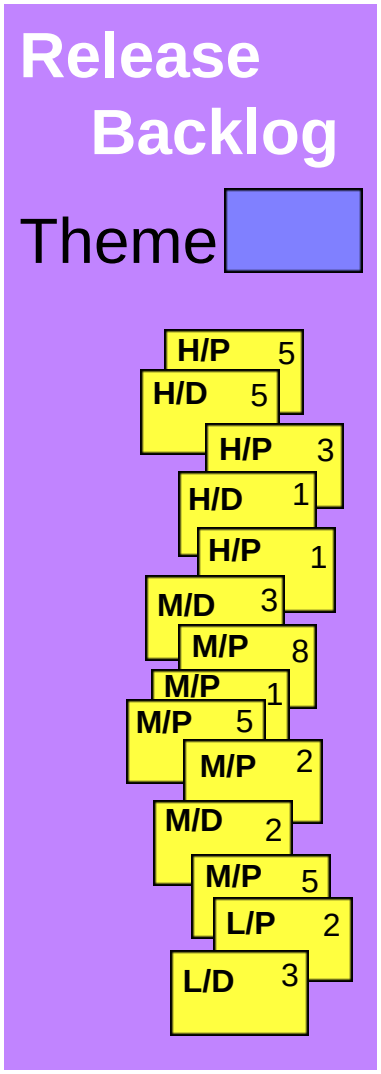
Release Planning, part 1

Product Backlog



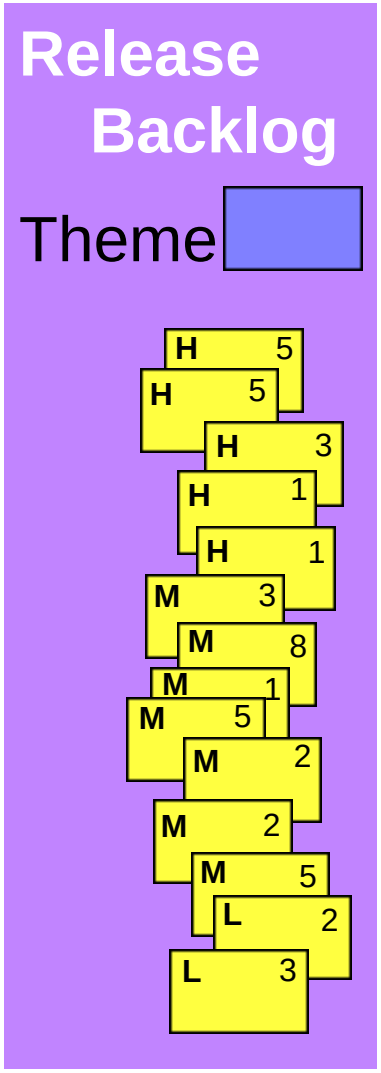
Input	Product Backlog
Output	**Prioritized Release Backlog** (User Stories) marked as differentiating or parity
Who	Business, Product Owner, Dev Team

Release Planning, part 2



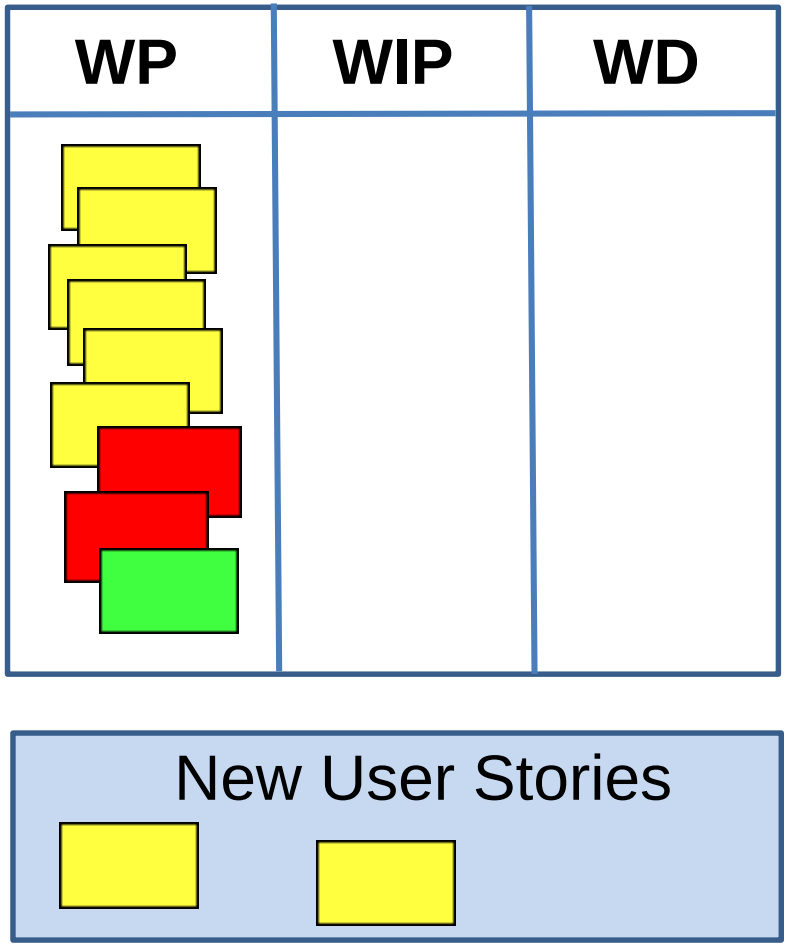
Input	Release Backlog
Output	**Estimated and Prioritized Release Backlog** (story points)
Who	Delivery Team using planning poker

Release Planning, part 2



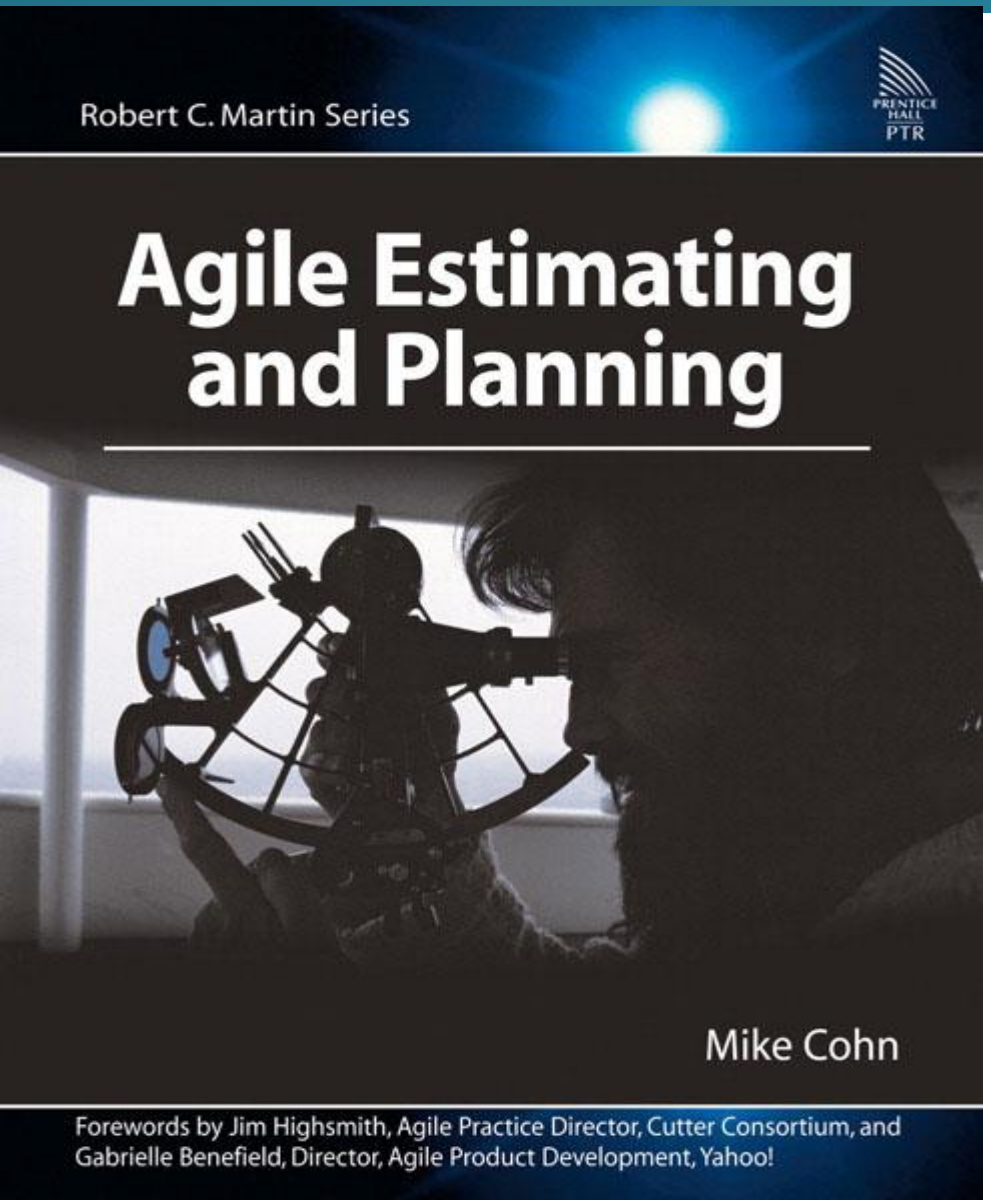
Input	Release Backlog
Output	**Estimated and Prioritized Release Backlog** (story points)
Who	Delivery Team using planning poker

Sprint Planning



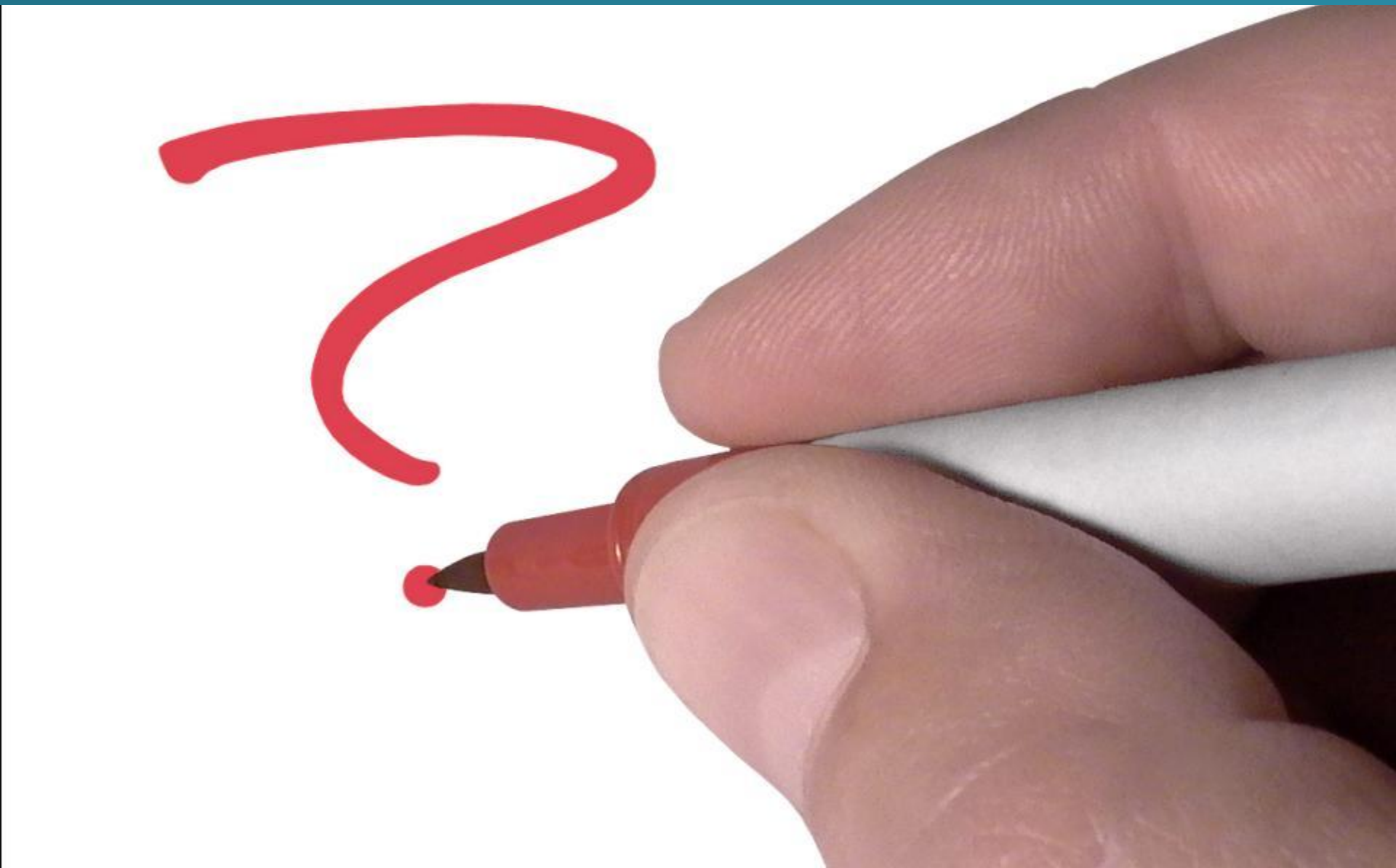
Input	Release Backlog
Output	**Sprint Backlog, Definition of Done, Architecture Spike, Information Radiator**
Who	Delivery Team

References



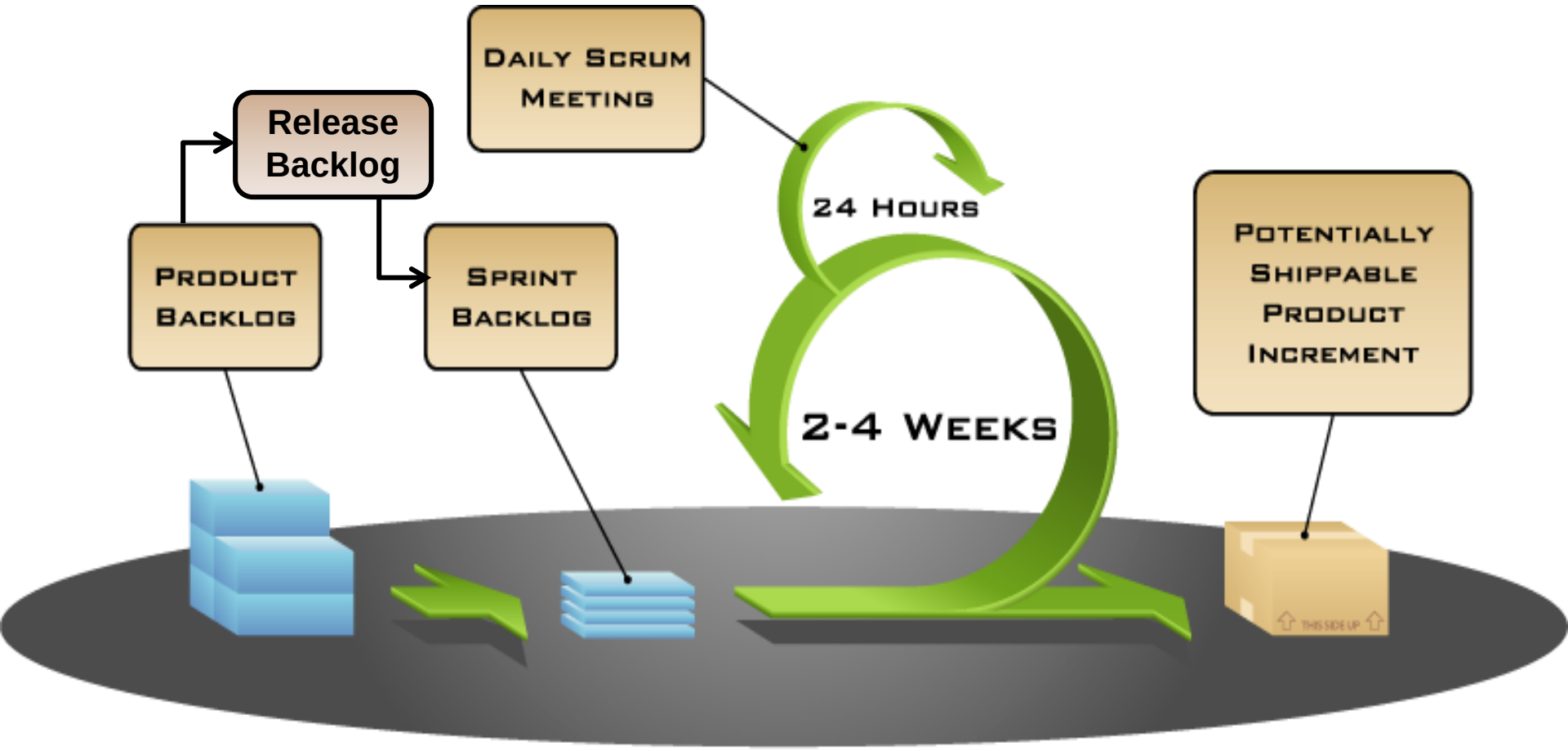
Refer to Mike Cohn's book for details on how to estimate and plan.

Estimating & Planning



For every sprint …

Scrum



Copyright © 2005, Mountain Goat Software

Daily Standup Scrum Meetings

- Daily 15 minute status meeting
- Same place and time every day
- Chaired by Scrum Master
- Attended by entire sprint team
- Others can attend
- Chickens and pigs (only the deliverers speak)

Daily Scrums

Each team member answers:

- What did you do yesterday?
- What are you doing today?
- What are your blocking issues?

No problem solving!

➡ **Leave after 15 minutes!**

Daily Scrum Outcome

Records

- Sprint Backlog up to date
- Scrum Master updates the blocks list

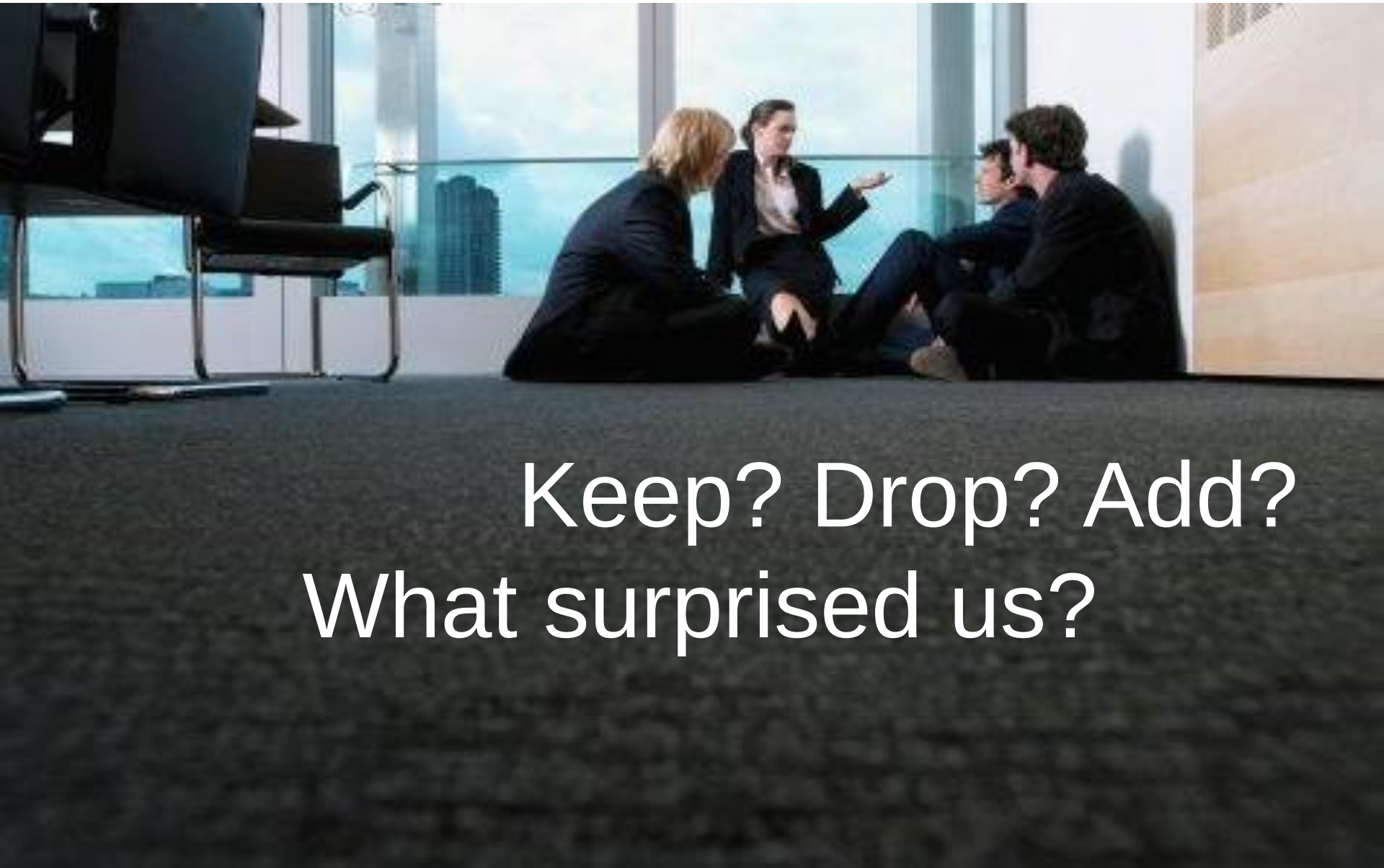
Sprint Review Meeting

- Held the last day of the sprint
- Attended by team
- Team demos “done” user stories to stakeholders
  - Requests feedback
- Team holds retrospective
  - Updates the process for the next sprint

Demonstration

- Only DONE DONE working user stories.
- Ask for attendance from the following:
  - Executives
  - Internal users
  - Stakeholders
  - Customers
- Early iterations may be unsuitable for customers
- Add or Update Stories on the Release Backlog

Retrospective



Other Retrospective Questions

- What was supposed to happen?
- What actually happened?
- Why were there differences?

Take Action

What process changes will we make for the next iteration?

The Team owns the learning from the retrospective.

They do not have to share it with the rest of the organization..

Scrum Exercise

Develop a Brochure in a 3-day Sprint

Complete Sprint Planning Meeting -10min

00

- Select at least 5 Product Backlog Items
- Identify 2 to 3 Tasks per Item

Day 1

00

- 8 minute day

Day 2

00 00

- 2 minute Daily Scrum
- 8 minute day

Day 3

00 00

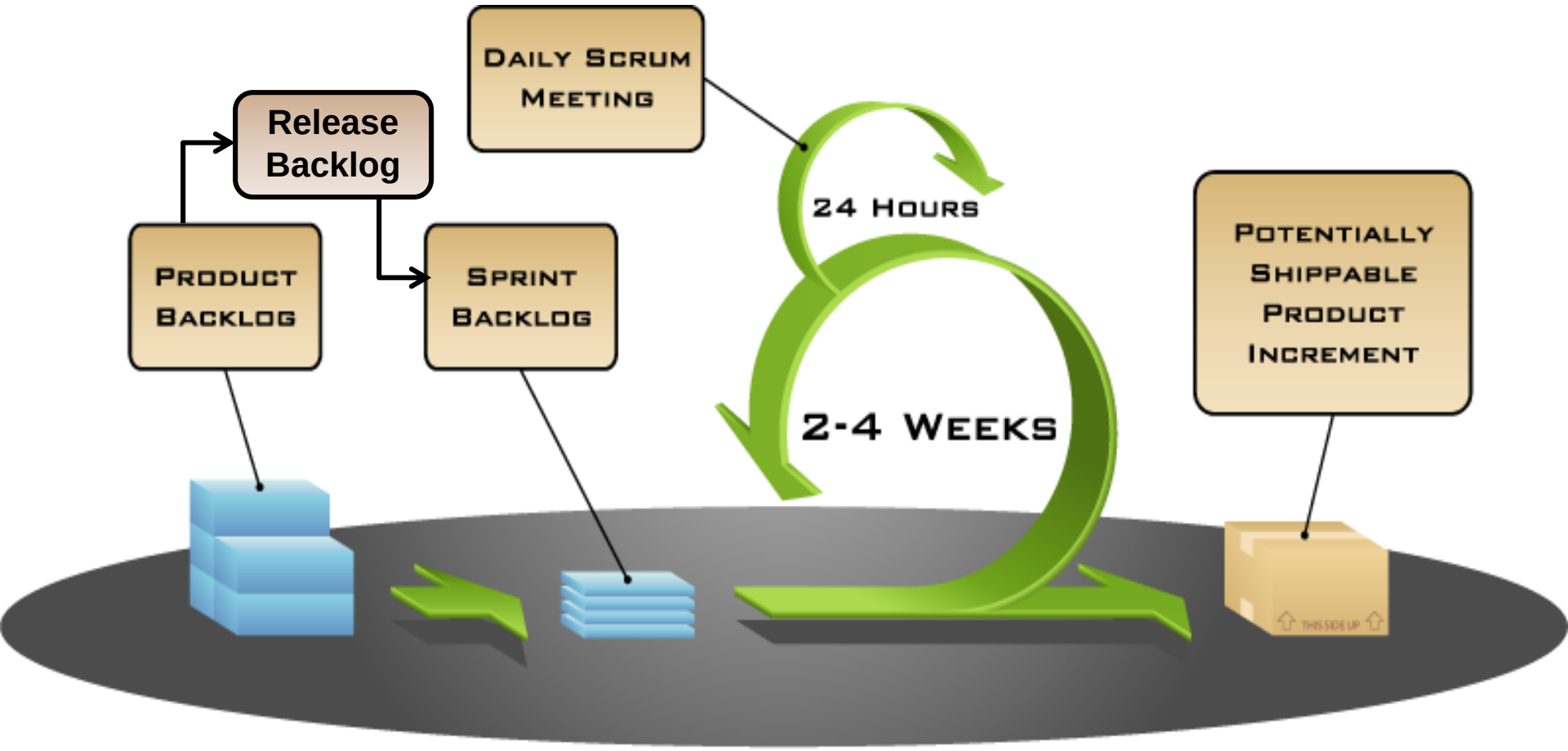
- 2 minute Daily Scrum Meeting
- 8 minute day

Demo & Reflection

Scrum Exercise
Retrospective

Agile Summary

Scrum



Copyright © 2005, Mountain Goat Software

Scrum on a Page

Roles

Stakeholders

Product Owner

Scrum Master

Team

Artifacts

Product Backlog

Release Backlog

Sprint Backlog

Blocks List

Information Radiator

Meetings

Product Planning

Release Planning

Sprint Planning

Daily Scrum

Sprint Review Meetings

Concept inspired by William Wake's "Scrum on a Page," http://xp123.com/xplor/xp0507/index.shtml

Agile is continuous learning and adaptive planning.

- M. Buckingham

A good agile project will build something that meets customers needs but may be different from original plans.

- Jim Collins

Agile Highlights

- Make decisions together to avoid handoffs
- Dev Team decides how – nothing technical in user stories
- Cover all types of stakeholders
- It’s all about learning
- Pace yourselves
- Don’t accumulate technical debt
- Beware of chicken subversion

Getting Started Tips

- Expect the teams to over estimate in the first few sprints
- It will take about 5 sprints to develop a cadence and velocity
- Teams may take on too much after some time
- Watch out for anti-bodies

Things to Consider

- Management Support
- Strong and Experienced Leader(s)
- Picking the right project as a proof point
- Providing the right education, tooling and governance
- Ability to allow change to occur
- Keep it Simple

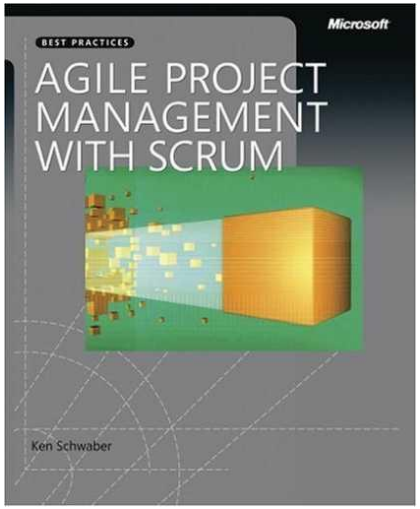
Scrum has been used by:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- Lockheed Martin
- Philips
- Siemens
- Nokia
- IBM
- Capital One
- BBC
- Pitney Bowes
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting

References

- http://scrumalliance.org/pages/what_is_scrum
- *Scrum and XP from the Trenches,* Henrik Kniberg http://www.infoq.com/minibooks/scrum-xp-from-the-trenches
- http://c2.com/cgi/wiki?PairProgrammingBenefits
- *The Elegant Solution*, Matthew May
- *Outside-in Software Development*, Carl Kessler and John Sweitzer

References



Agile Project Management with Scrum, Ken Schwaber
(for scrum novices)

Agile Software Development with Scrum, Ken Schwaber and Mike Beedle
(for experienced scrum types)

Feedback

1. **WWW**: What Went Well
2. **WCBI**: What Could Be Improved
3. **NPS**: Net Promoter Score

"Would you recommend this class to a peer?"

10-9	Yes!
8-7	Neutral
0-6	No way!