Agile Program Management: Measurements to See Value and Delivery

Johanna Rothman

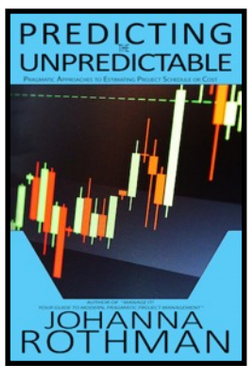
Agile and Lean Program Management: Scaling Collaboration Across the Organization

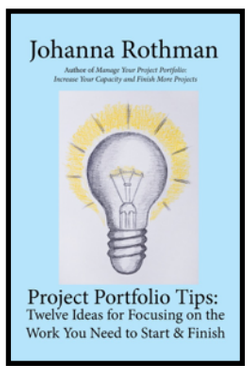
@johannarothman

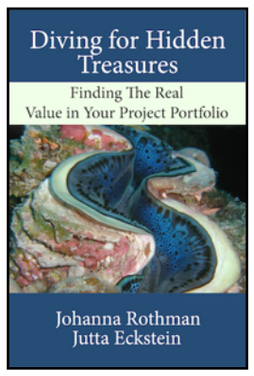
www.jrothman.com

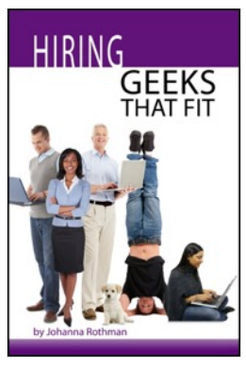
jr@jrothman.com

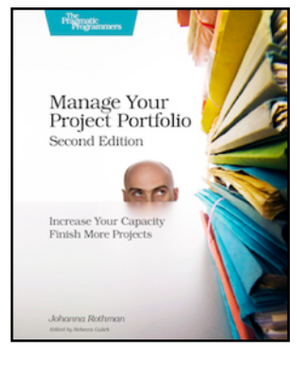
781-641-4046

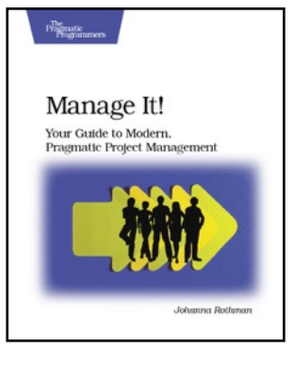


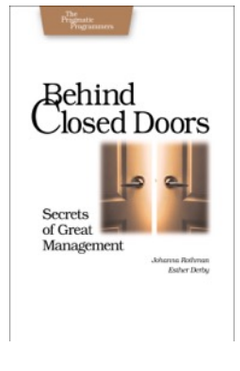










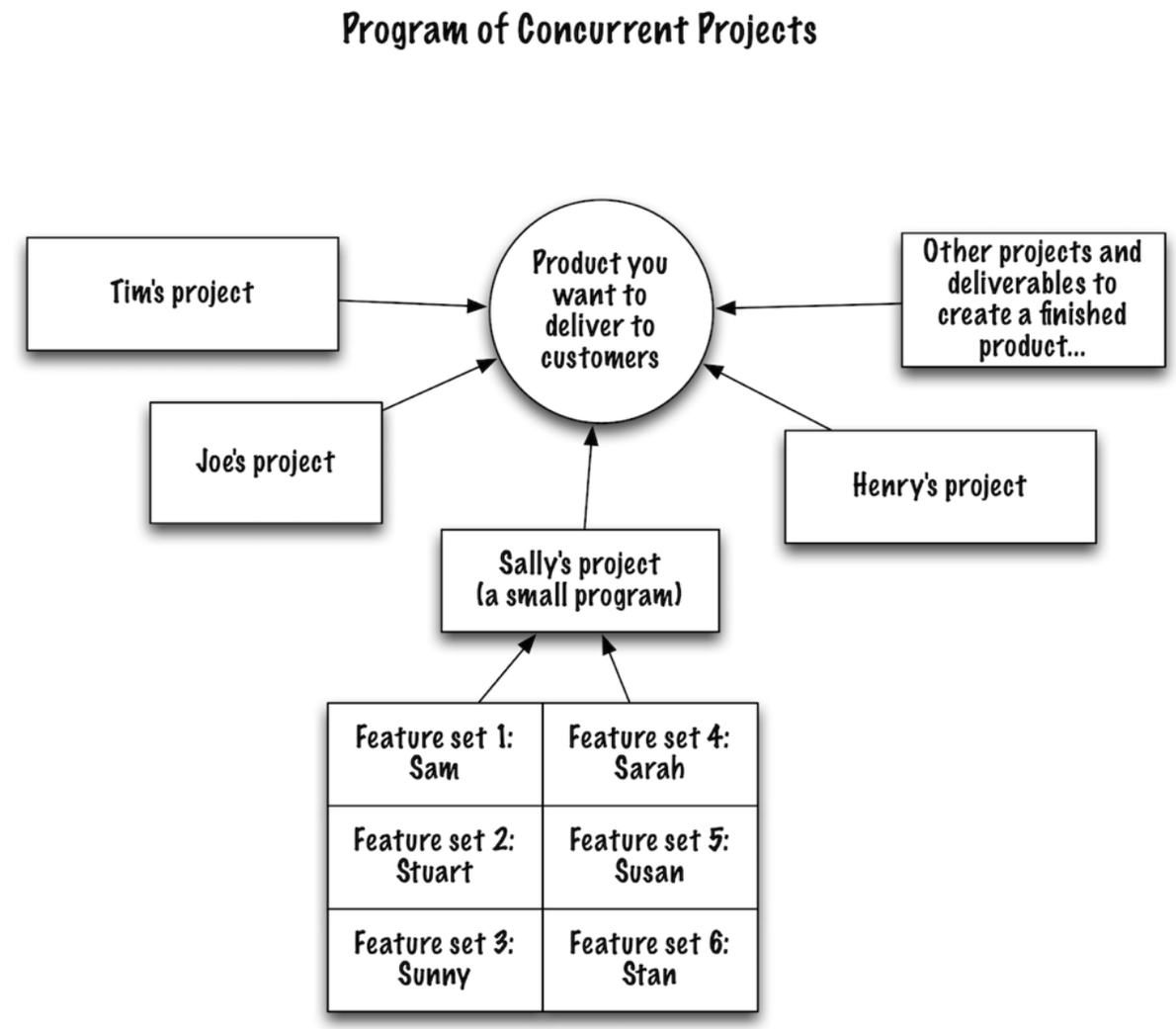


Measurement Problems in Programs

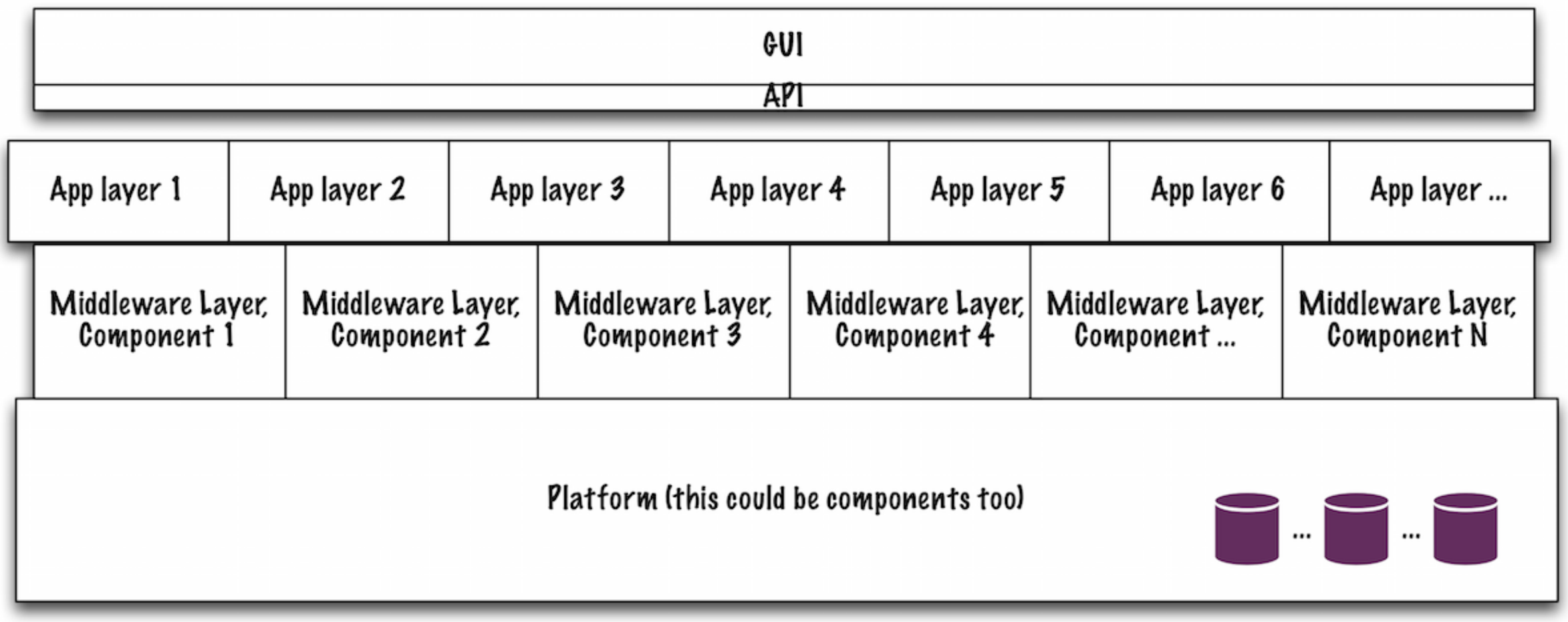
- Management wants to understand the big picture
- Teams see their own piece
- Program management needs to see the whole

What A Program Is

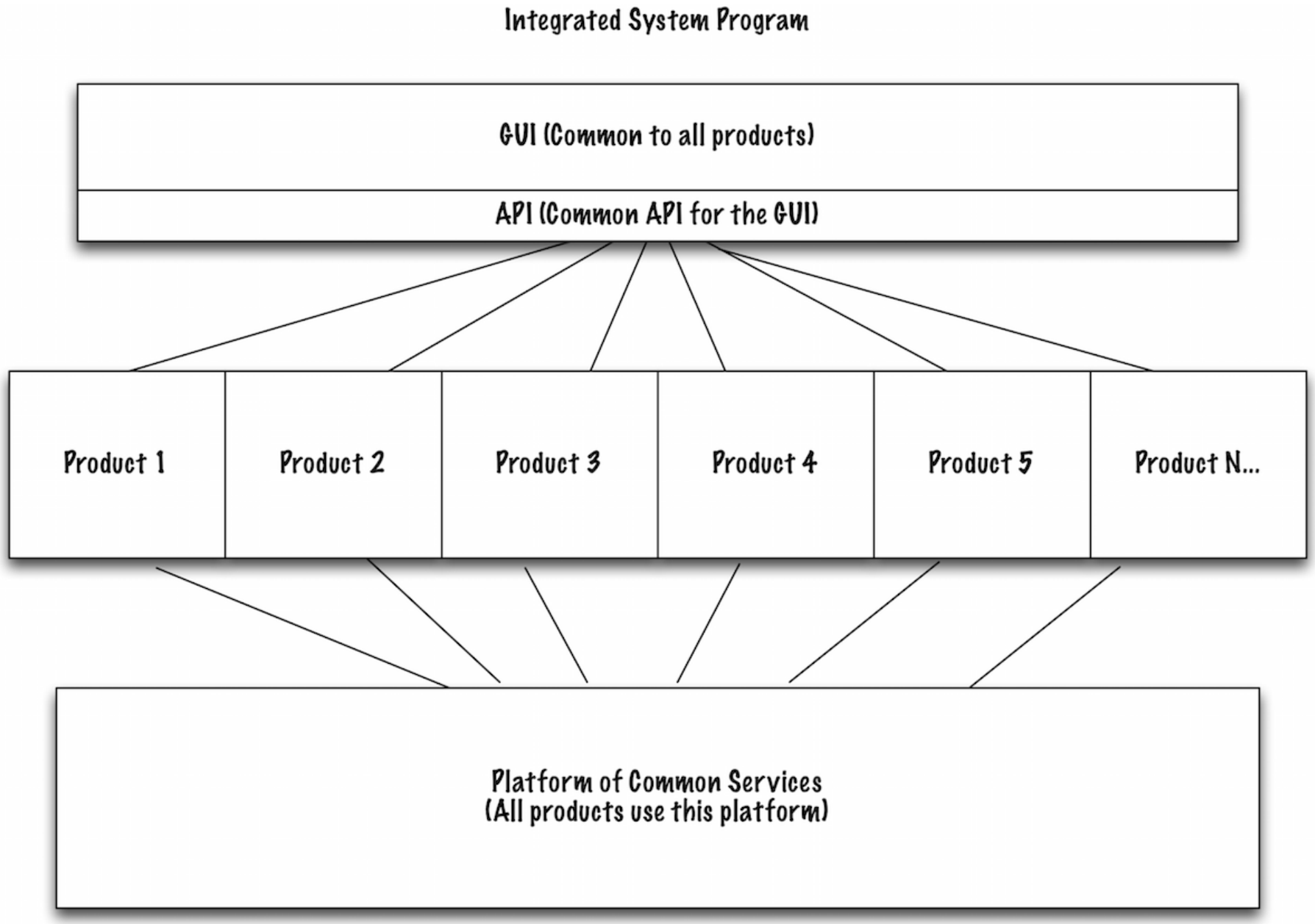
- Program: strategic collection of several projects that fulfill one business objective
- What some people think of as "scaling agile"



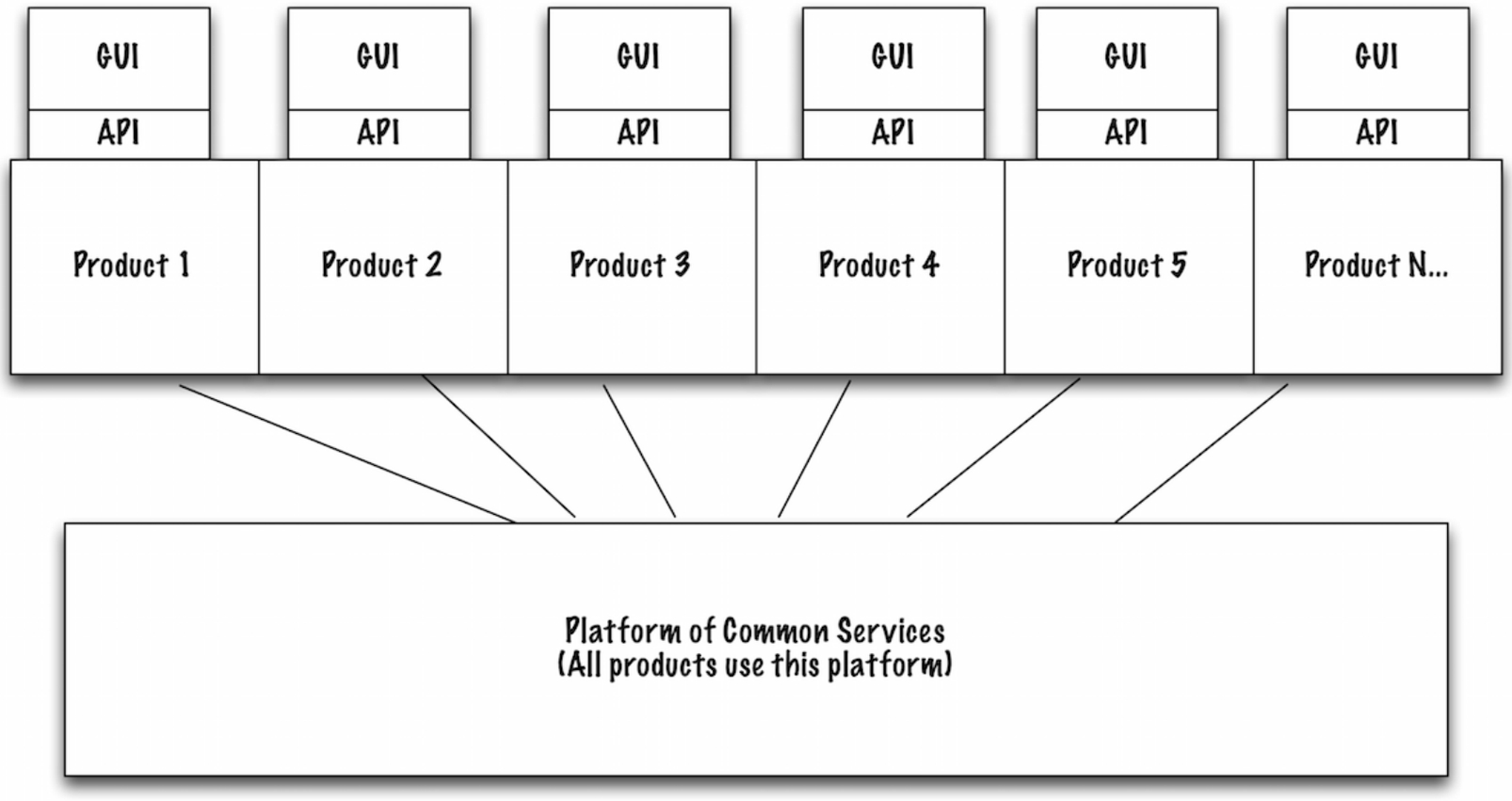
One Coherent Product Program (Web Product)



Integrated System Program (Integrate other products)



Inter-Related Product Program (Smartphone)



People ask for predictive measures when they don't trust you to deliver

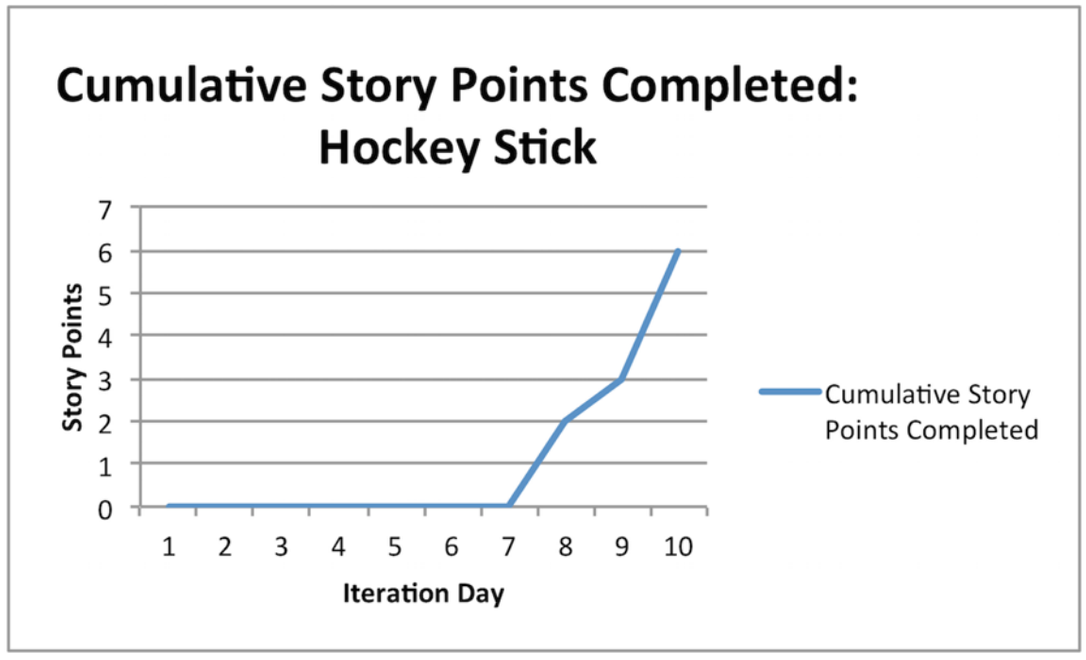
What Managers Care About

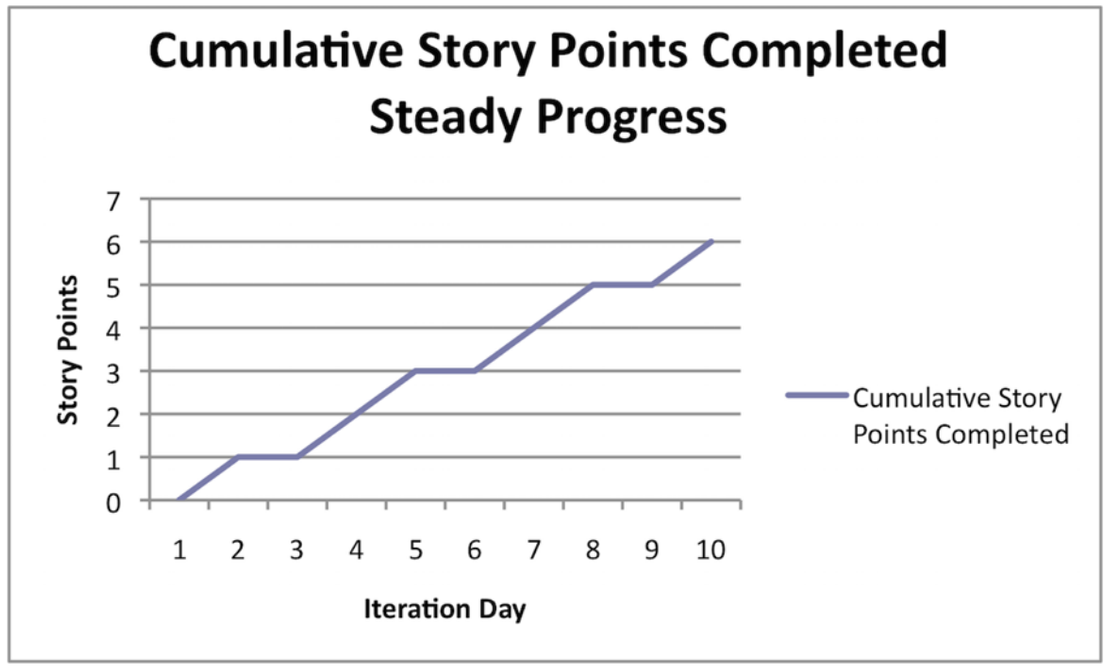
- Customer acquisition
- Customer retention
- Revenue
- Customer experience (so you can acquire/retain customers)

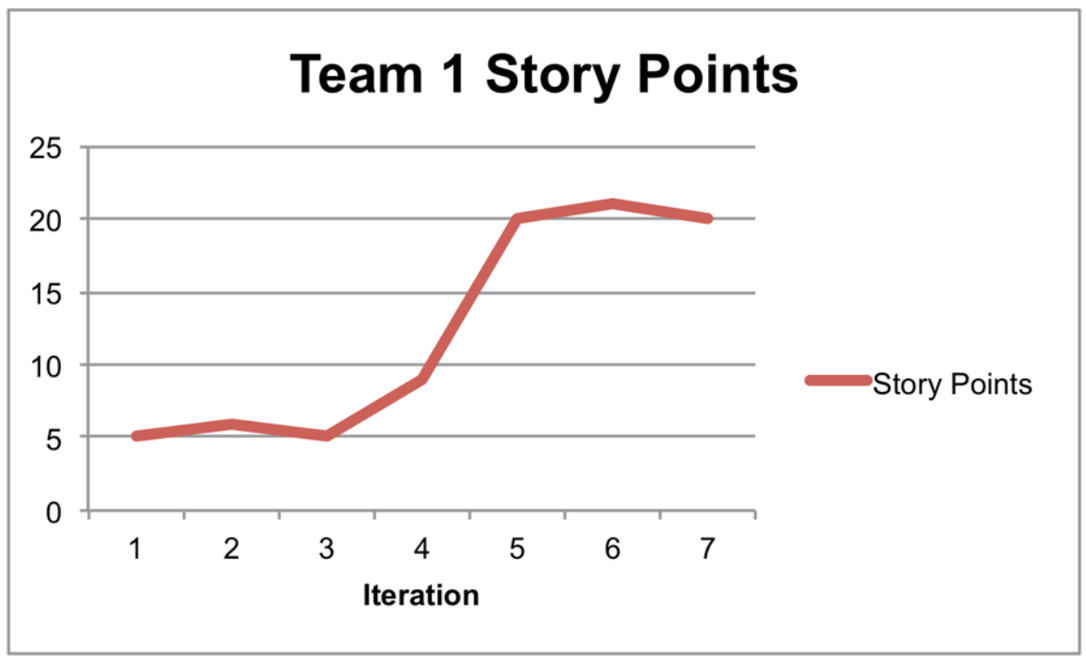
Change Measurement Thinking and Measurements

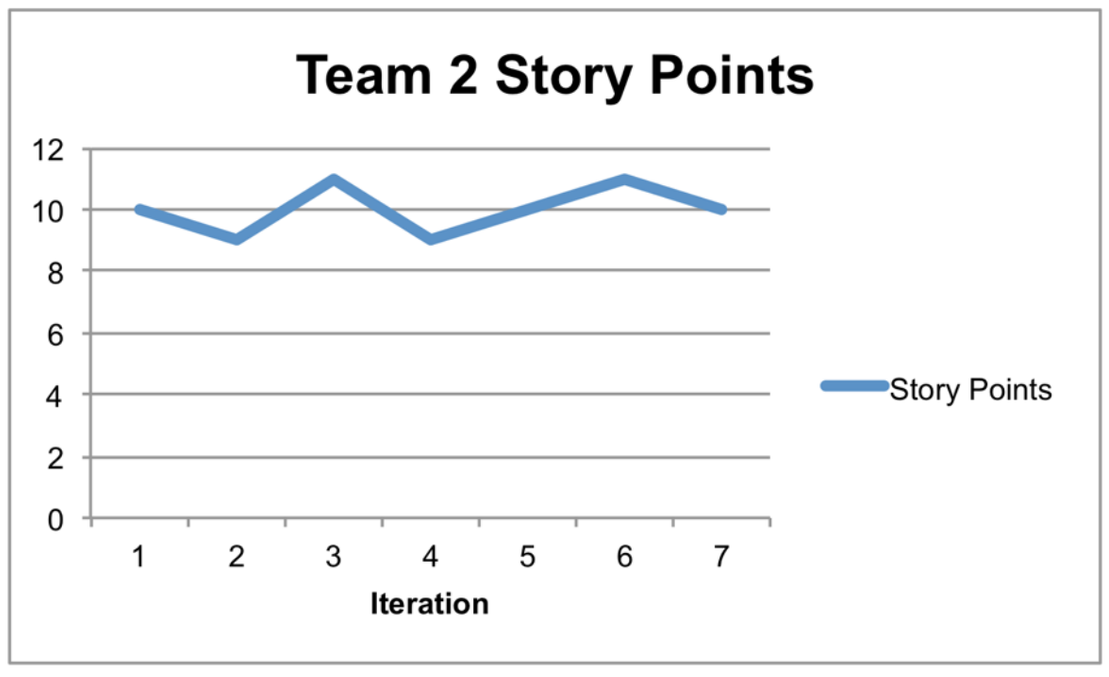
From (Predictive or Surrogate)	To (Empirical)
When will you be done? or How much will it cost?	How much are you willing to invest? Do you have a target date?
Are you on track? or What's the Earned Value?	Let us show you working product.
When will we see revenue?	We can show you value now. We can release now.
What do the customers think?	We can show progress against release criteria and We have customer satisfaction data

Several Velocity Stories









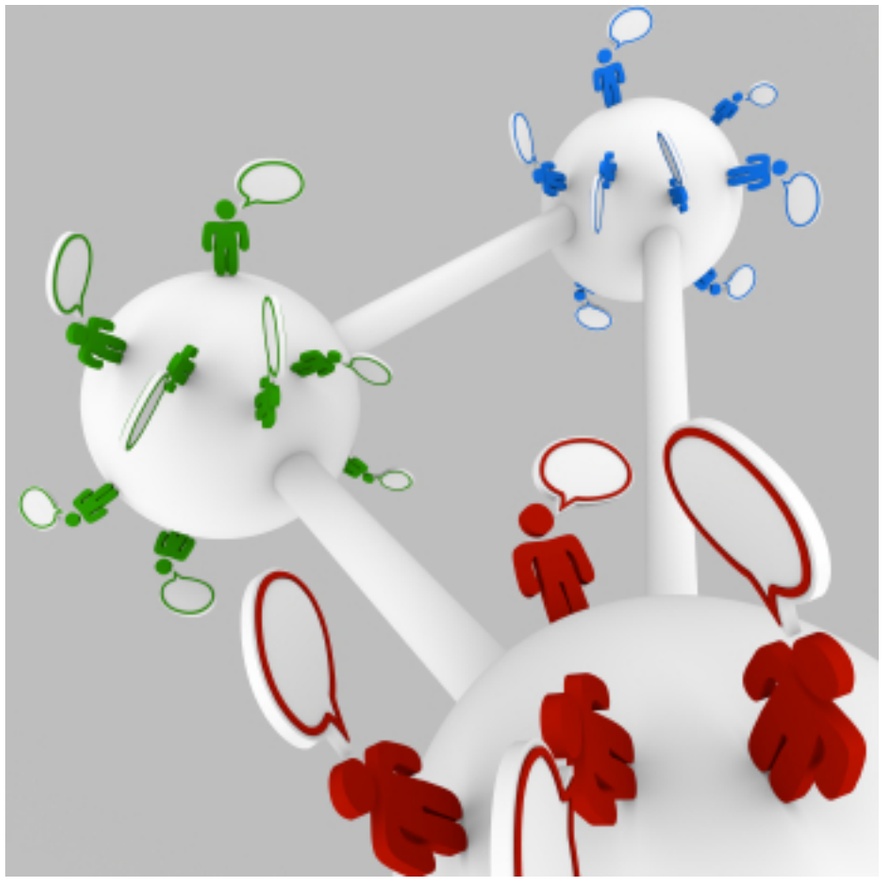
"Do you want story points or do you want working features?"
—a courageous team

“Problems” with Velocity

- Velocity is a measure of capacity, not productivity — Ryan Ripley
- Not always predictable
- Individual to each team, and can vary with domain

Scaling Team Measurements Doesn't Work

- Team measurements are personal, not additive
- Teams have different cycle and lead times because they do different work
- We don't normalize stories inside or across teams

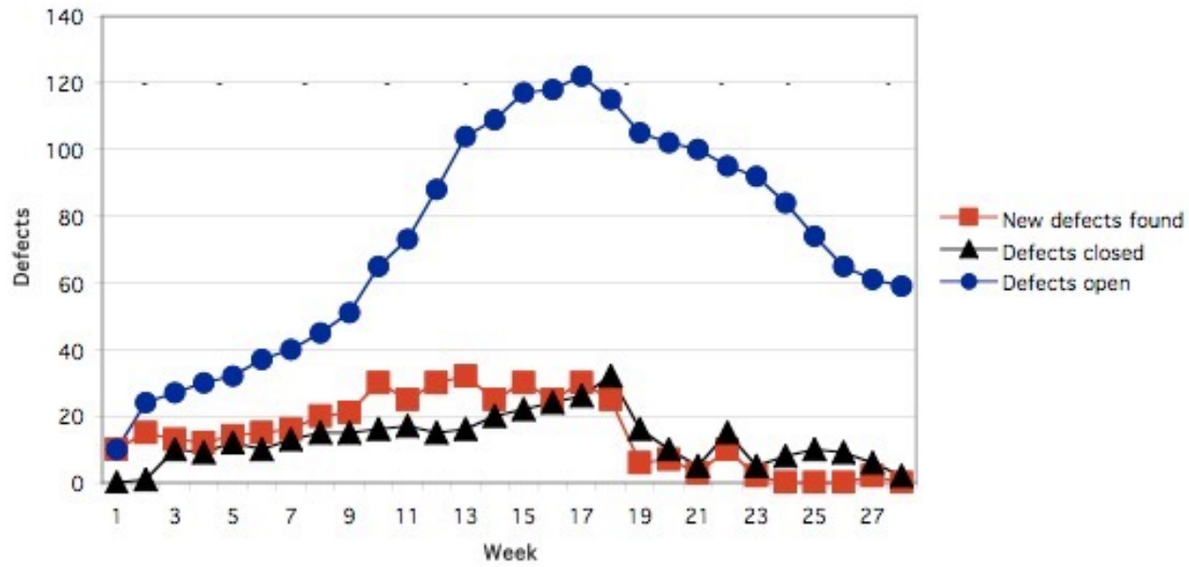


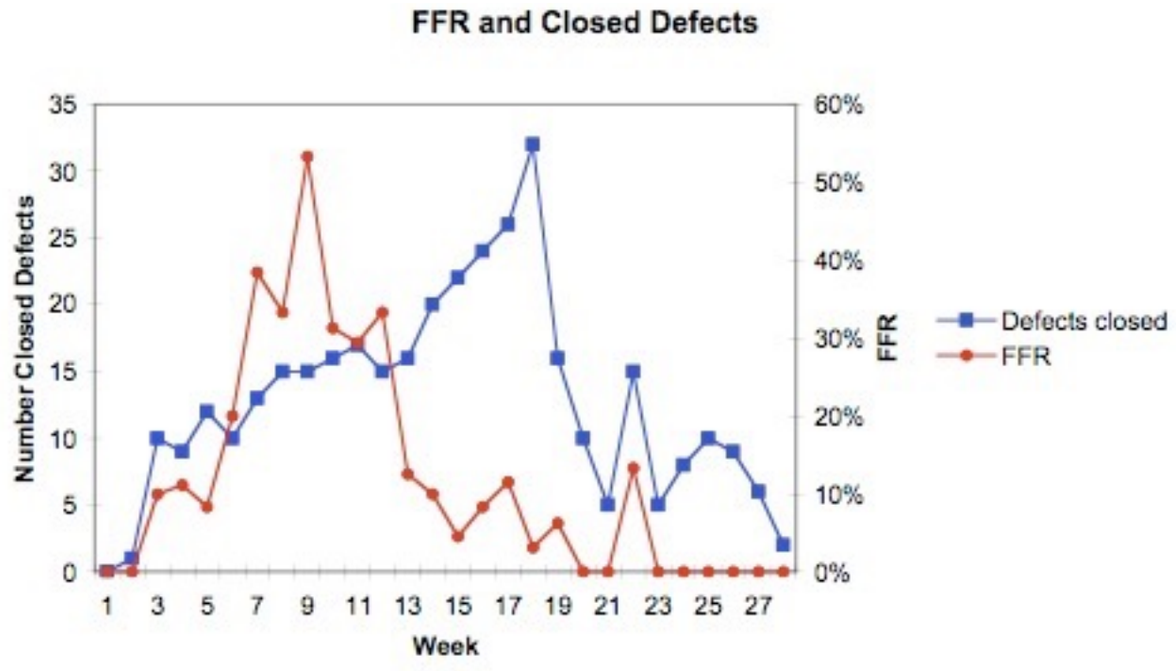
Can We Measure Learning?

- Software is learning, innovation
    - Agile helps us learn when we deliver some value and then have the choice to change what is next
        - Discover with spikes and prototypes
        - Implement to address risks but not have a walking skeleton
- We learn to build momentum

Note About Measurements

- Use trend data
- Snapshots do not provide sufficient information





Agile Programs Provide Empirical Measurements

- Measure what you want to see and where you want to go
- Agile teams provide working features at regular intervals

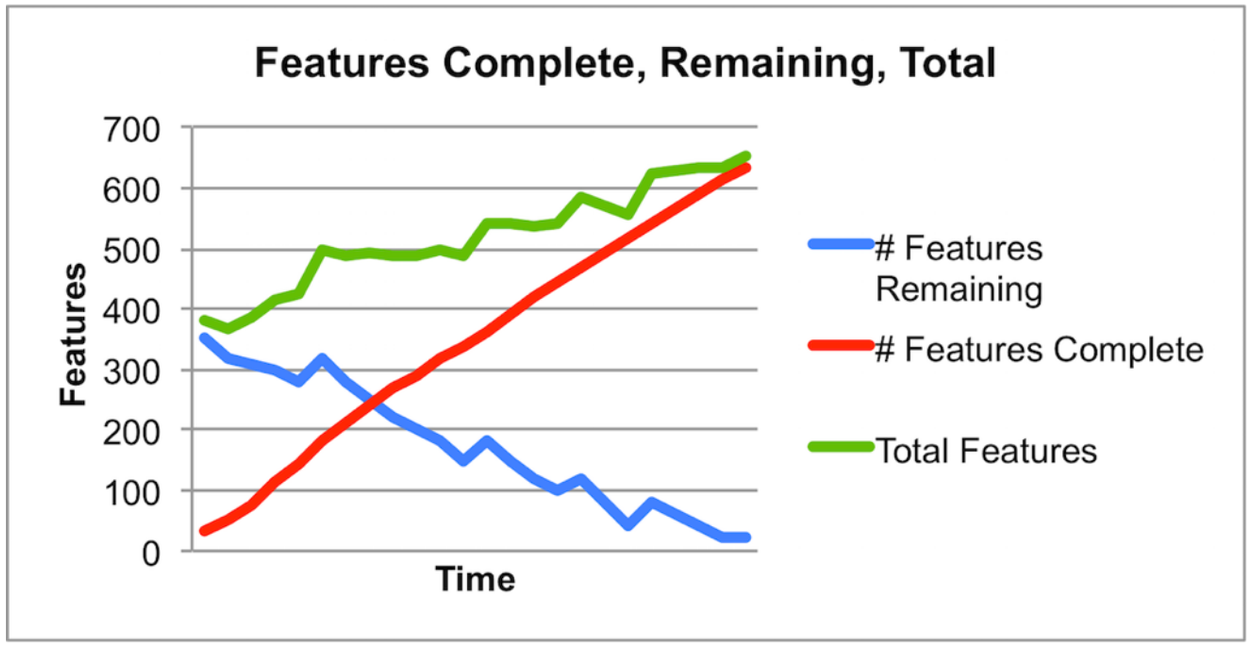
Measure What You Want to See

- “Tell me how you'll measure me and I'll tell you how I'll behave!” — Goldratt

- The more direct the measurement, the more you will get what you want

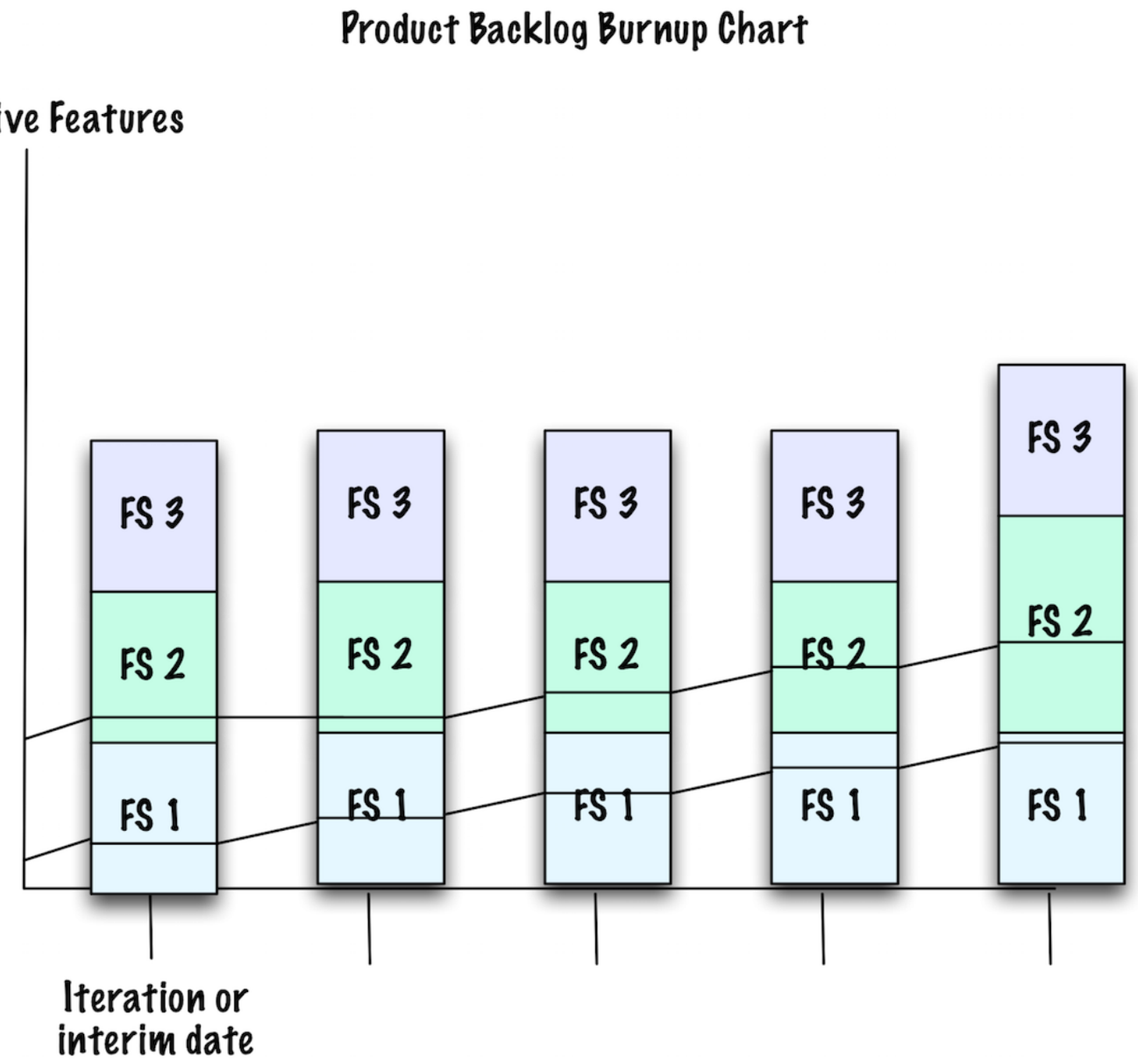
Measure Completed Features

- Completed features (running, tested features):
  - Your customers use them
  - You can release them
  - They are valuable
- Include total and remaining features so we have a sense of where we are
- Depends on deliverables, not epics or themes



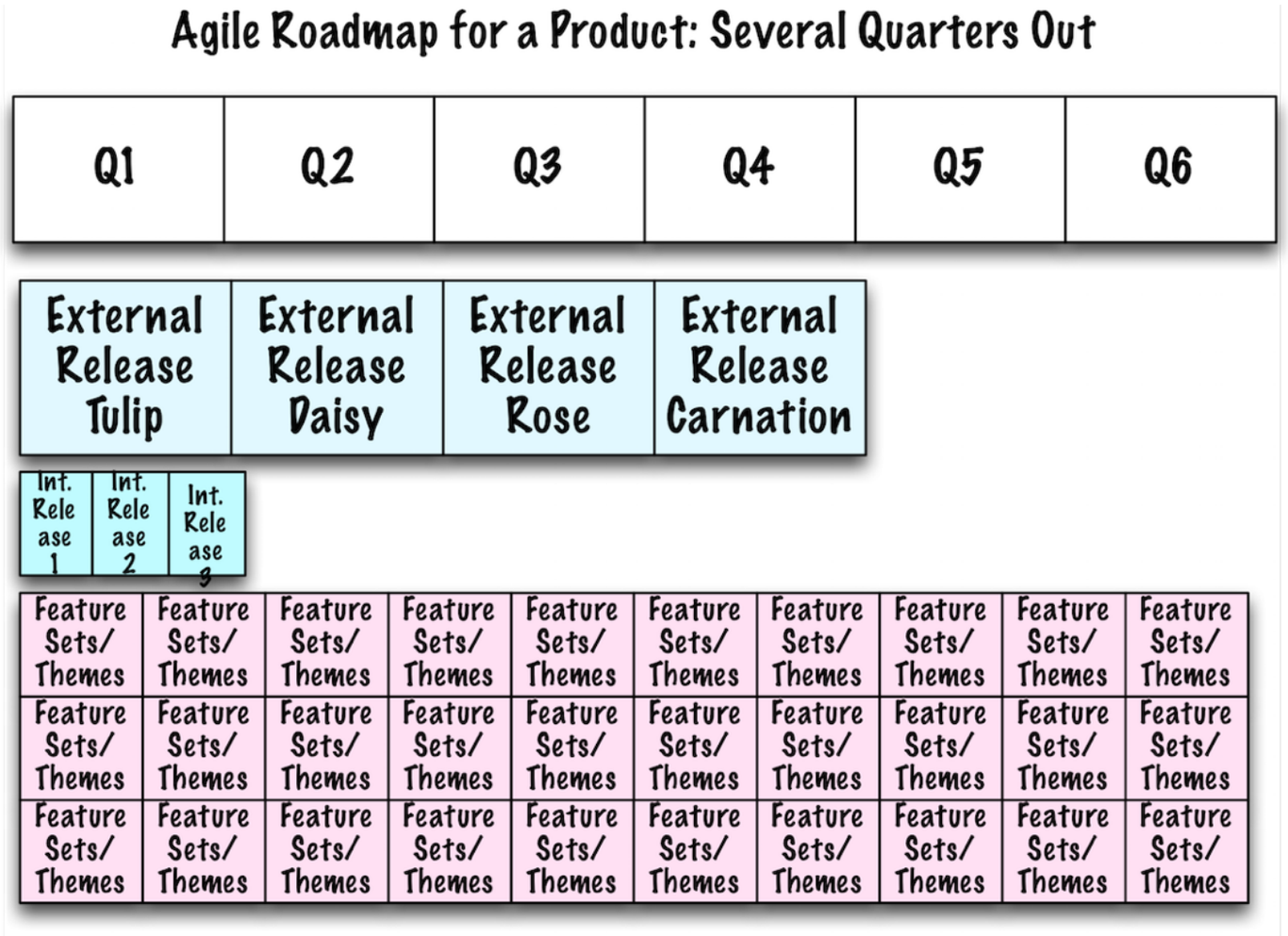
Product Backlog Burnup

- Real earned value
- Partial answer to "Where are we?"
- Shows value feature-by-feature
- Shows when features grow



Agile Roadmap in the Large

- "Big Idea" of what the product will be
- Interesting and not sufficient
- Deliverables often too large and not specific

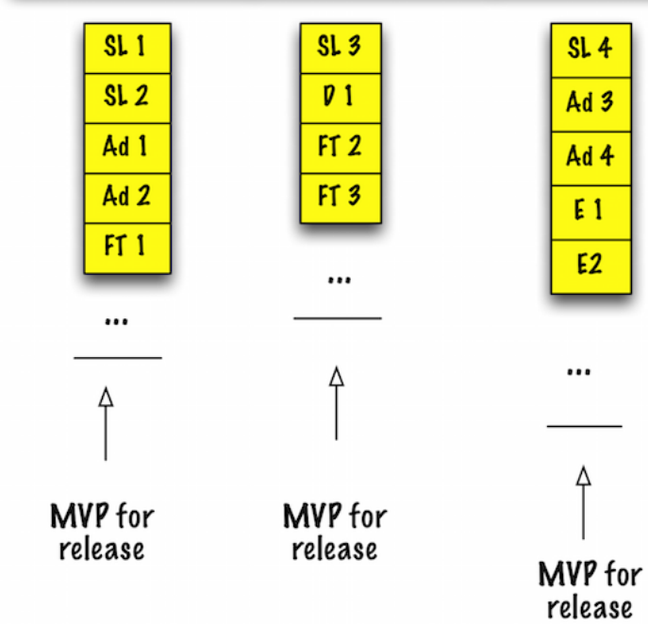


Agile Roadmap in the Small

- Deliverable-based planning (small slices through the architecture)
- Specifies value for different users
- Use as a rolling wave plan

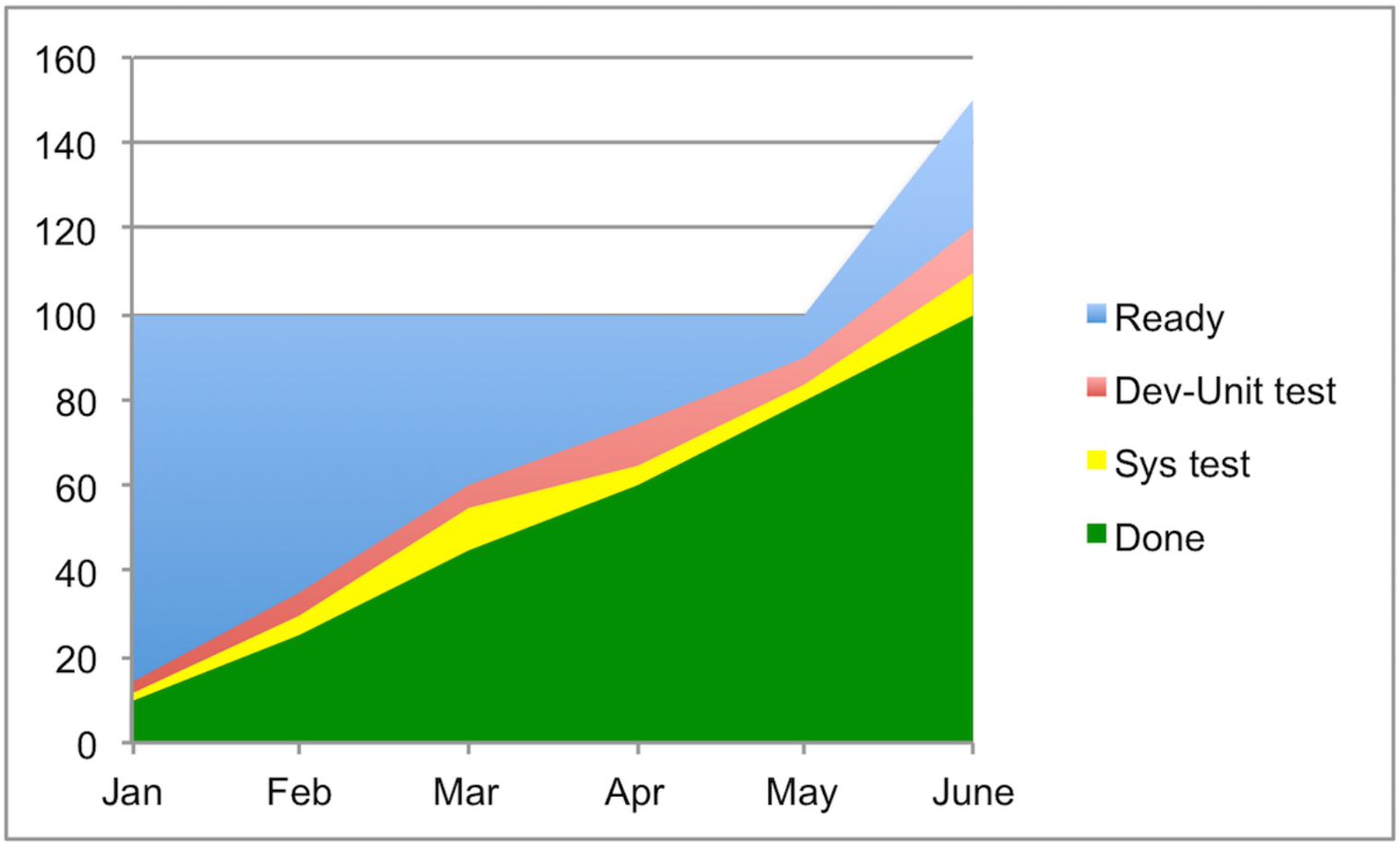
Product Example: One Quarter Agile Roadmap

Internal Release 1		Internal Release 2		Internal Release 3	
Secure Login, Part 1	Secure Login, Part 1	Secure Login, New ID	Text Transfer, Part 1	Text Transfer, Part 1	Secure Login, Part 3
Admin, Part 1	Diagnostics, Part 1	Admin, Part 2	Admin, Part 2	Admin, Part 2	Admin, Part 2
File Transfer, Part 1	File Transfer, Part 1	Engine, Part 1	Engine, Part 1	Engine, Part 2	Engine, Part 2



What Do You Want Less of?

- Work In Progress (across entire program)
- Defects
- Other "Less of":
  - Multitasking
  - ?



People respond to what you measure.

Measure what you want less of and more of.

Tell people which is which.

More Program Measurements

- Run rate (money are we use every week/month?)
  - Check against the target or investment question
- Program-level WIP as feedback for the program:
  - Product Owner Value Team
  - Feature teams' momentum

Release Frequency Possibilities

Software as a Service	Boxed Software	Product with Firmware	Product with Hardware or Mechanical components
Continuous			**Infrequently**
Continuous Deployment: As often as several times a day	Often: But the cost of release is high	Less Often: The cost of release is high	Infrequently: Every release might be a major release

How Often Can You Release?

- Internal releases
  - Can you use continuous delivery inside your org?
  - If not, how often can you release?
- External releases: business decision and as often as you can afford to release

Release "Measurements"

- How often do you release, internal or external
- How long does it take to move from build to release?
  - Build time
  - Cycle time (interdependencies show up here)

Product Measurements

- Create scenarios for what's important to your product
  - Performance
  - Reliability
  - Quality attributes of the product
- Can be a function of release criteria

Qualitative Measurements

- Customer feedback/ happiness
- How often you get feedback from customers
- Obstacle report
- How long it takes to make decisions

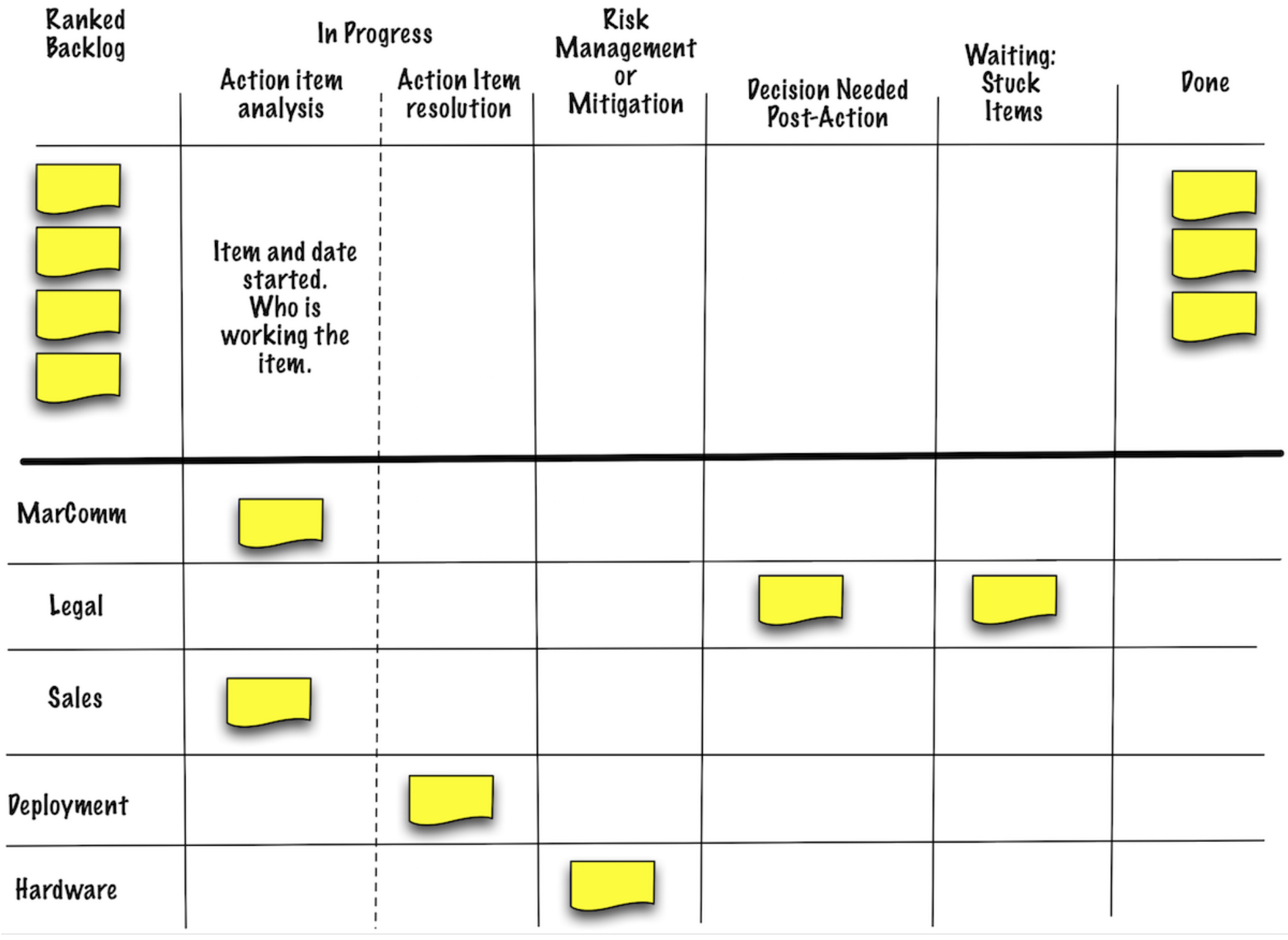
Obstacle Report

- For the program, not management

Obstacle Report

Rank	Obstacle	Request Date	Days Since Request Date
1	Chair for Jim	Feb 1	15
2	Need tester full time	Jan 1	44
3	Standup meeting area with whiteboard	Jan 7	38

Decision Times



Measurement Traps

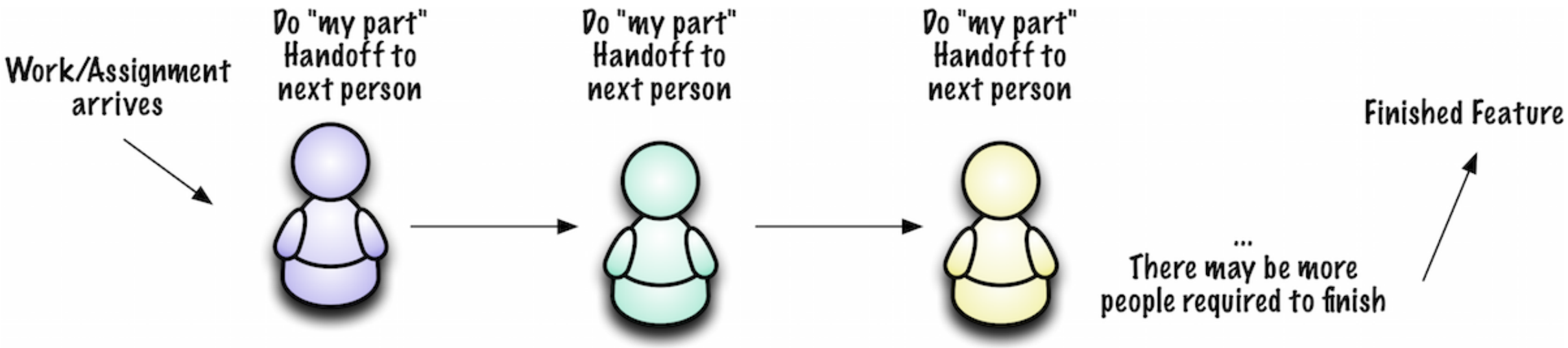
- Never attempt to measure anyone or any team's *productivity*
- Never compare people as individuals
- Never compare teams against each other

What is Productivity in an Agile Environment?

- Features over time
- Teams take features, people don't take features
  - If people collaborate, swarm, or mob, personal "productivity" is irrelevant
  - We don't normalize features between teams

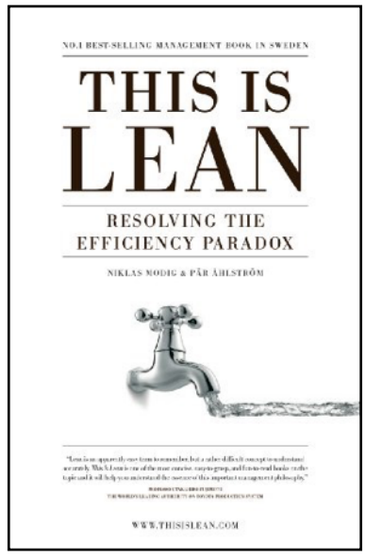
Resource Efficiency

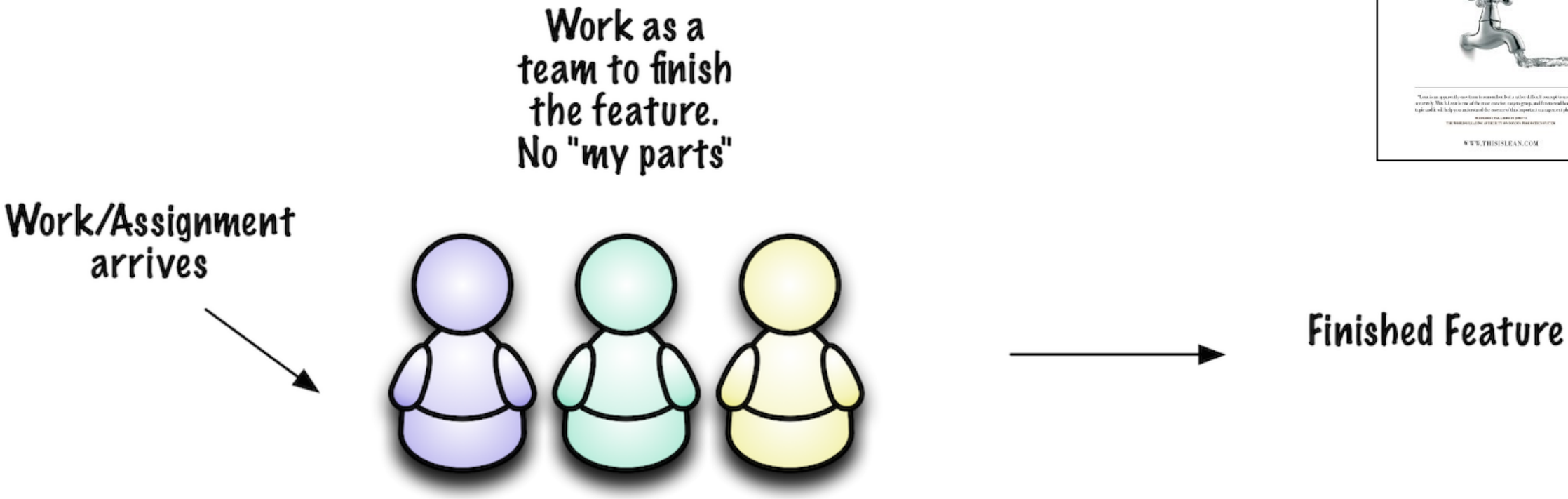
- Are your managers concerned with utilization?
- Resource efficiency is based on experts, not throughput



Flow Efficiency

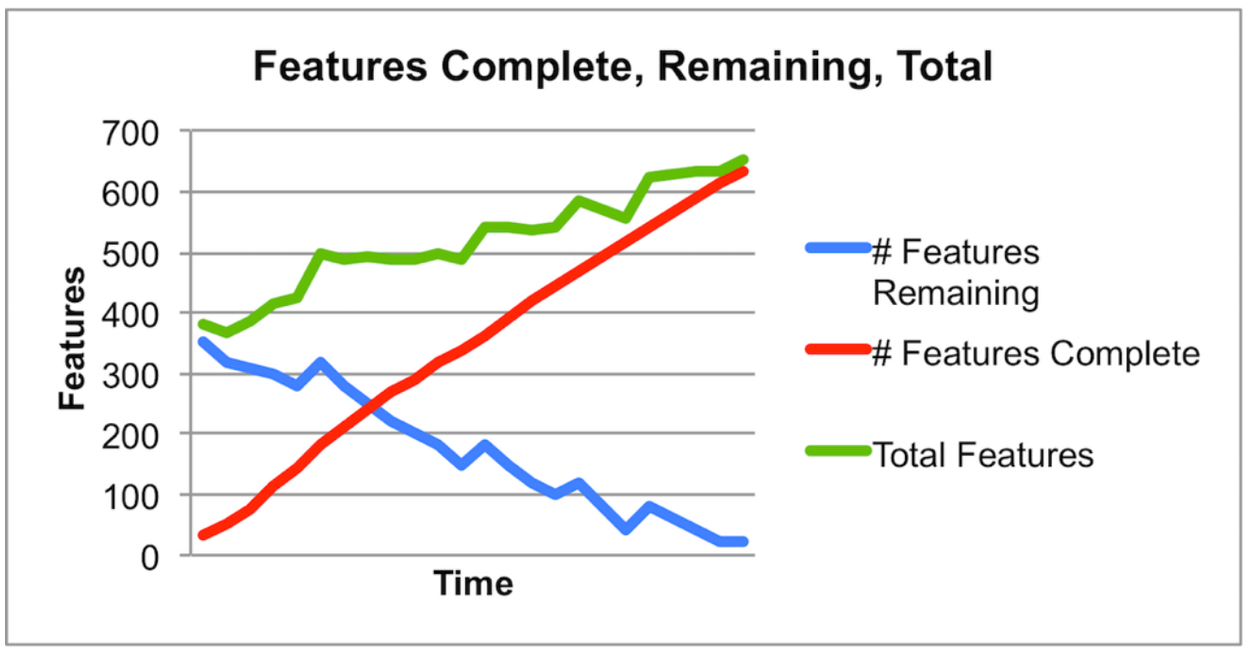
- Based on throughput
- Throughput provides program momentum

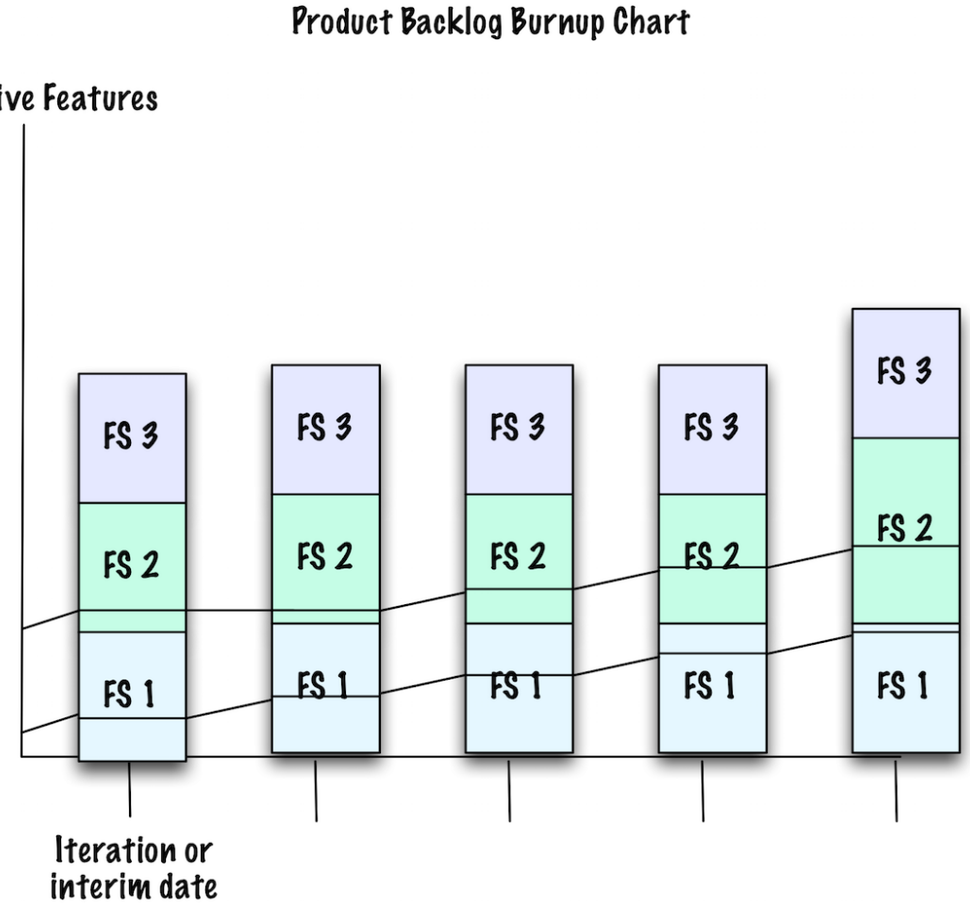




We Want Flow Efficiency

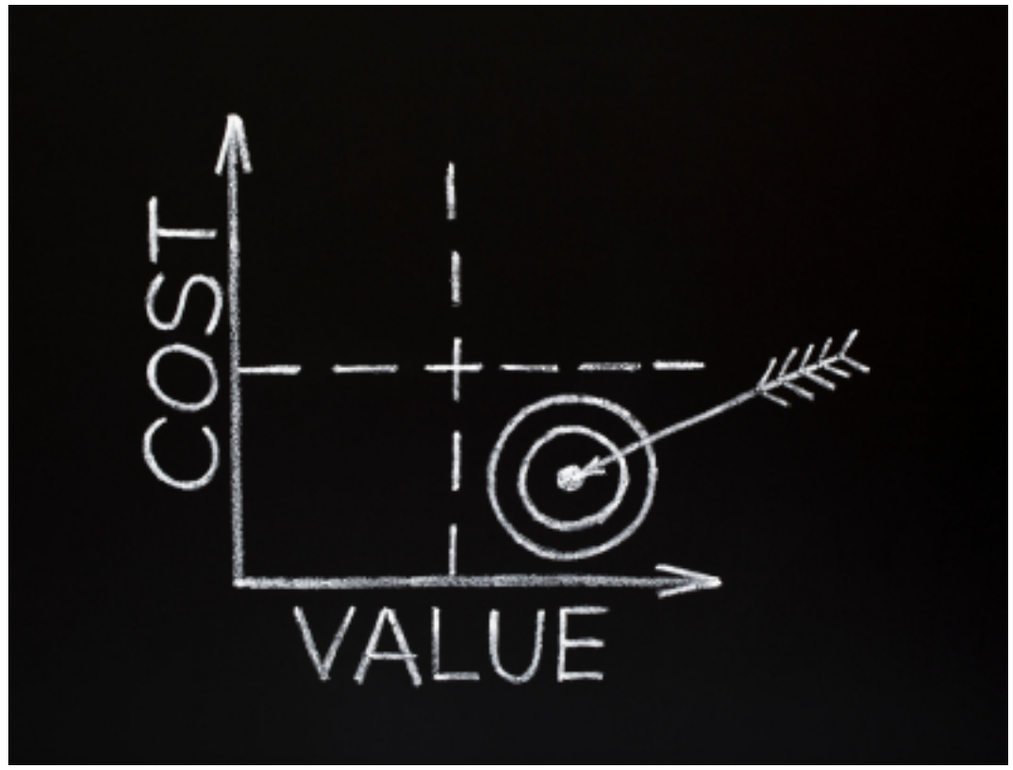
- Increases throughput
- Decreases WIP





Cost or Date Questions

- Do you want resilience or prediction?
- Ask these questions:
  - How much would you like to invest in time, money, or learning before we stop?
  - Are you ready to watch the product grow as we build it so you can stop us when you don't want to invest anymore?
  - What is the value of this project or program to you?



If You Must Estimate

- 3-point estimate, SWAG with percentage confidence, or spiral in on a date. Provide "next estimate" date
- Deliver often to build trust and maybe release earlier

Measure What's Valuable to You

- Features, because your customers buy/adopt features
- How often you release because you can recognize revenue/satisfy customers with completed features
- Progress against your expectations or against a target date
- Other indicators that help you know if you are succeeding or have trouble

Let's Stay in Touch

- Please link with me on LinkedIn:
  - www.linkedin.com/in/johannarothman
- Subscribe to the Pragmatic Manager newsletter:
  - http://www.jrothman.com/pragmaticmanager/