Software and Cyber Solutions Symposium 2018

Agile Metrics at Scale

Will Hayes

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM18-0354

Software and Cyber Solutions Symposium 2018

Metrics Used by Teams, Programs & Enterprises

Many Methods Generally Termed “Agile”

focused on team technical practices

focused on team management practices

Encourages risk-based selection of practices; different patterns for different contexts

Technical and management practices focused on writing the test that proves acceptance, then coding to that

Originally derived from Rational Unified Process, designed to scale

Pull-based approach particularly favored for services like security, systems engineering

Merger of lean, Kanban, and other Agile methods to support large scale projects

Agile is principles-based, which means there isn't only "one way" to do it correctly

Key Elements of Scrum

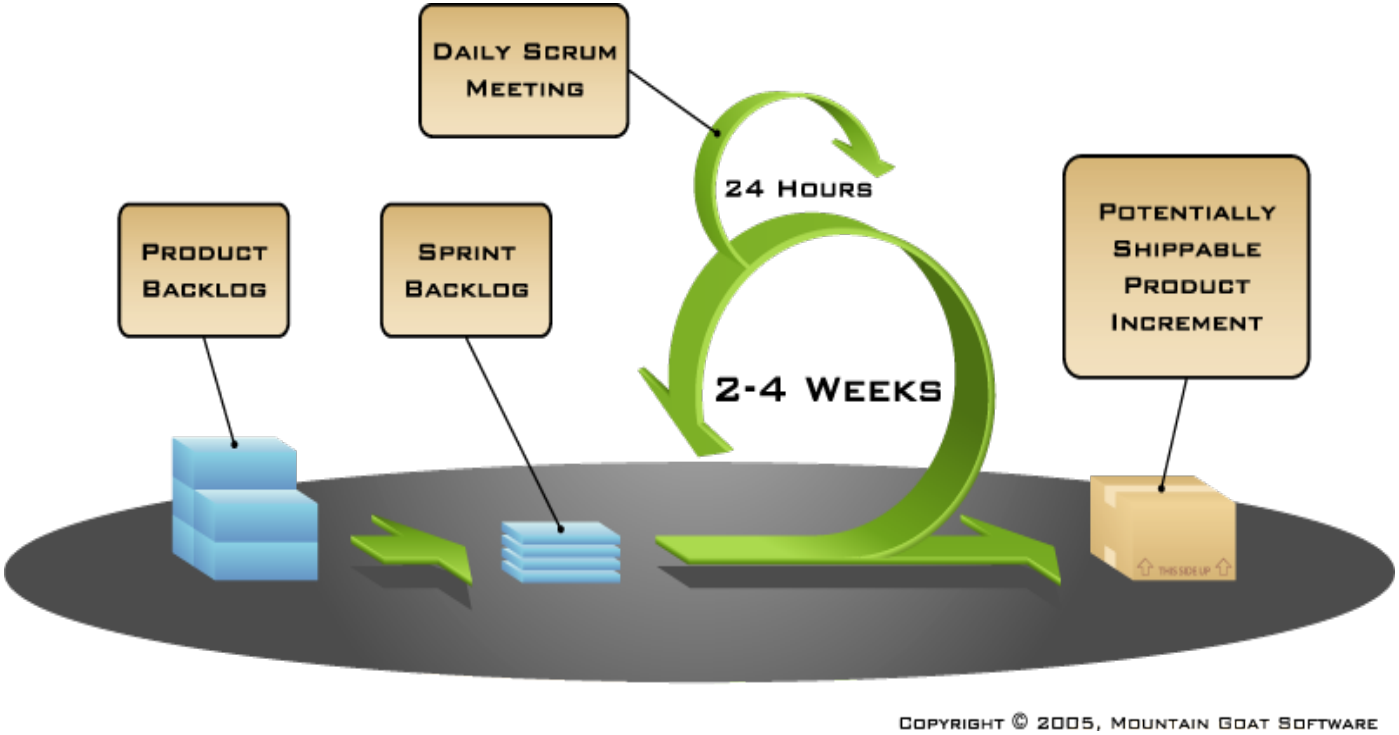
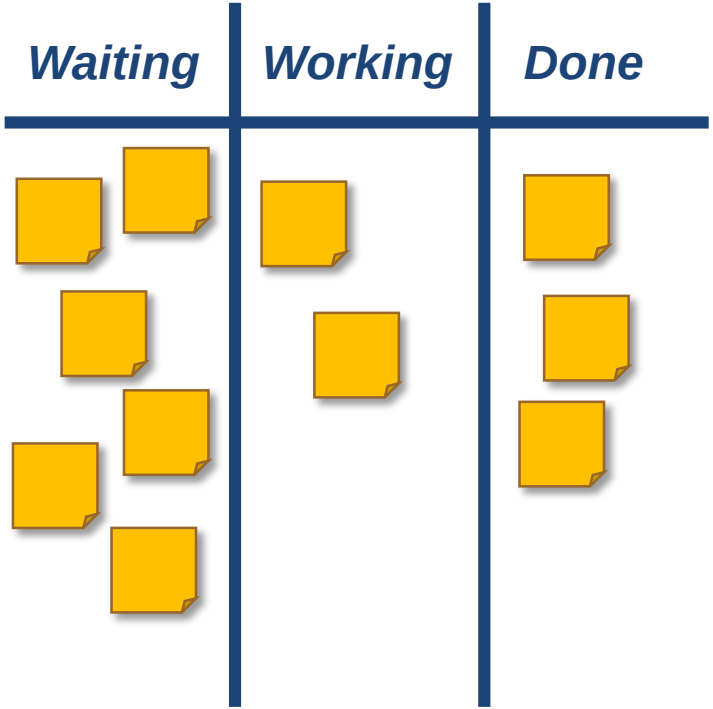


Image available at www.mountaingoatsoftware.com/scrum

Focused Work by Individuals

Visualizing work in process is a common building block of agile methodologies and frameworks/



Common Metrics: Teams

Notice Anything Missing?

Teams Using Scrum:

User Story Size: Story Points, Ideal Days, Hours…

Team Velocity: Story Points (or Stories) delivered per sprint

Sprint Burndown: Trend line for story points completed during a sprint

Release Burnup: Trend line for accumulation of finished work across sprints

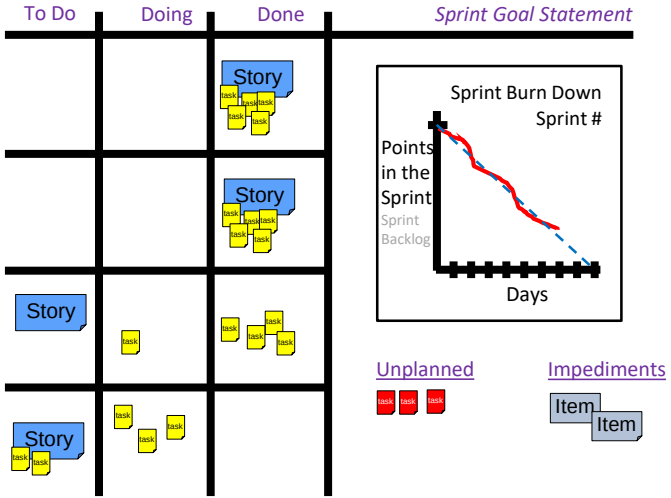
Teams Using Kanban:

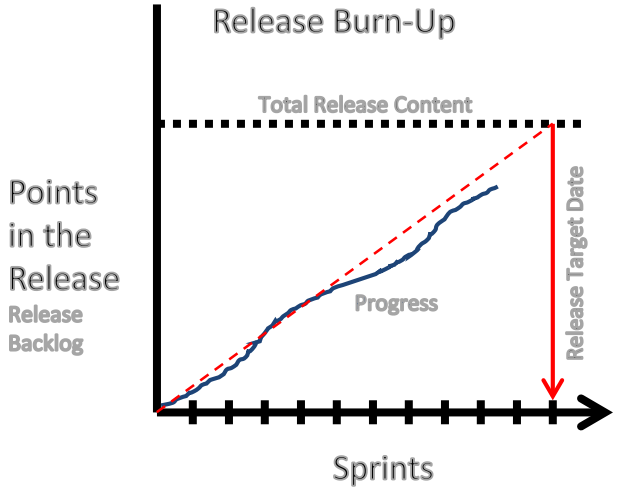
Work In Process: Number of work items are being worked at a given point in time

Throughput: Work items completed per unit of time

Cycle- or Lead-Time: The duration of time a work item spends in a particular state

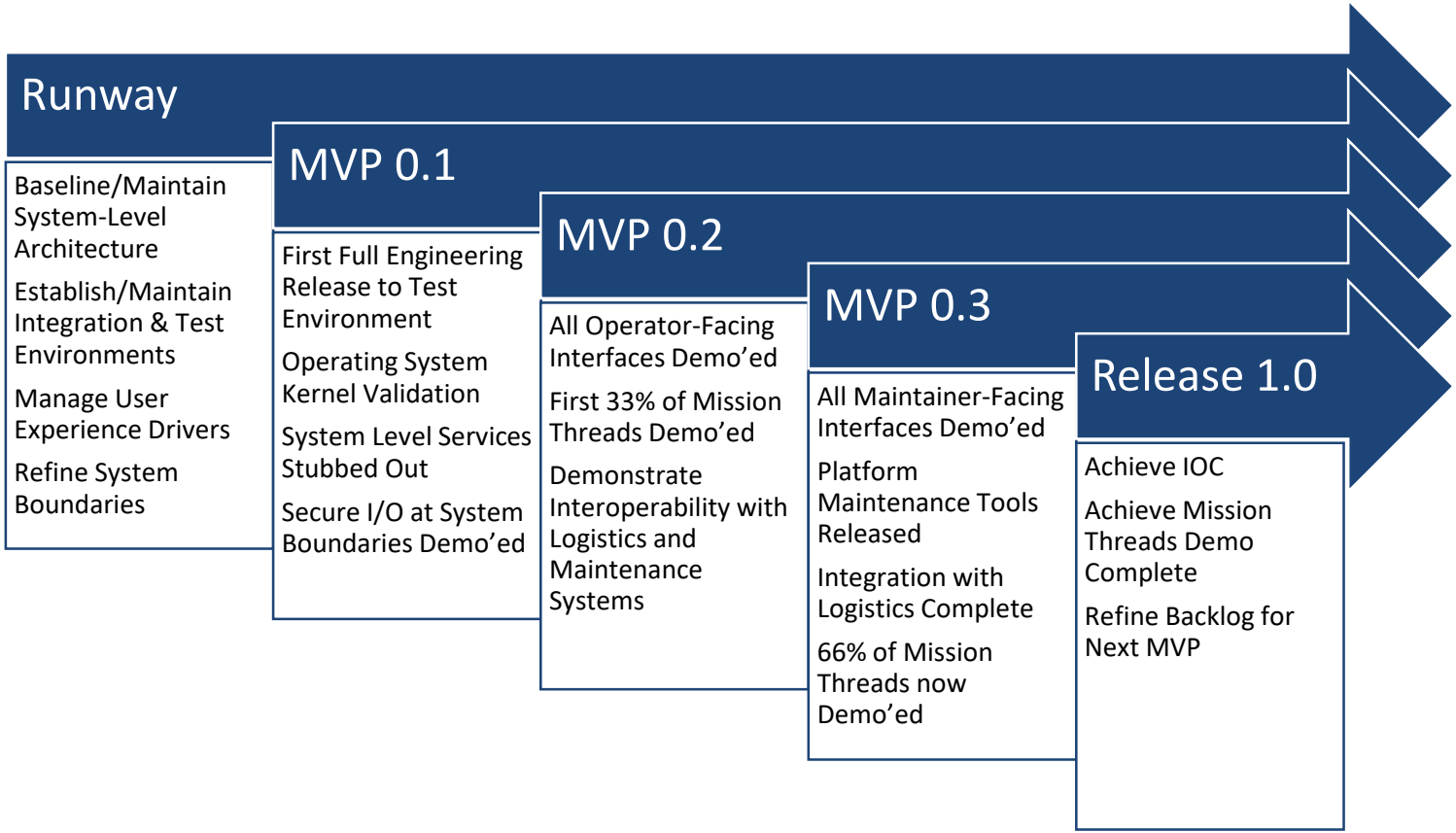
Things You Can Expect to See Teams Measuring





This is the Team's Data

Roadmap-Driven Accomplishments

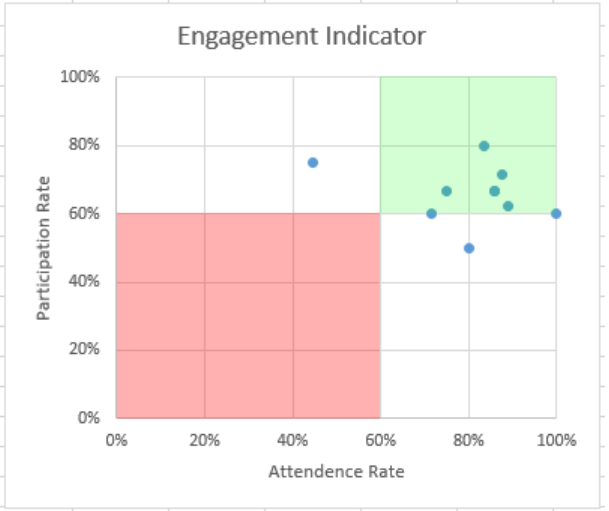


Software and Cyber Solutions Symposium 2018

Agile Adoption Metrics

Simple Indicator, Powerful Analysis

Ceremony	Expected	Attended	Engaged	%Attend	%Engage
1	5	5	3	100%	60%
2	7	6	4	86%	67%
3	8	7	5	88%	71%
4	4	3	2	75%	67%
5	9	4	3	44%	75%
6	6	5	4	83%	80%
7	9	8	5	89%	63%
8	7	5	3	71%	60%
9	7	6	4	86%	67%
10	5	4	2	80%	50%



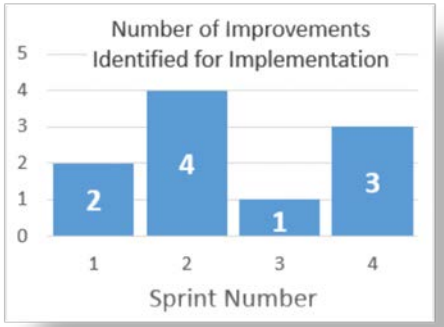
Subset/aggregate data to look for trends across:

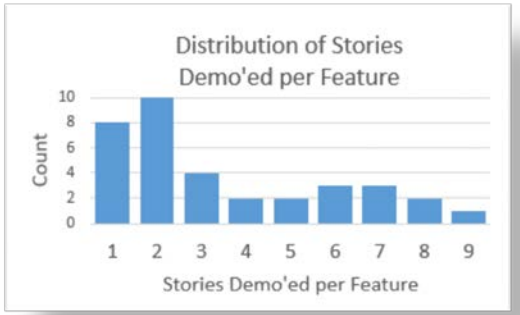
- Particular event types
  - Are ‘standups’ not working?
- Pockets of staff
  - Have we alienated ‘release managers’?

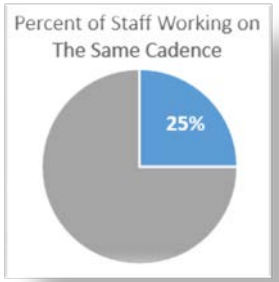
Indicator Examples$_1$

Essential Process Attributes

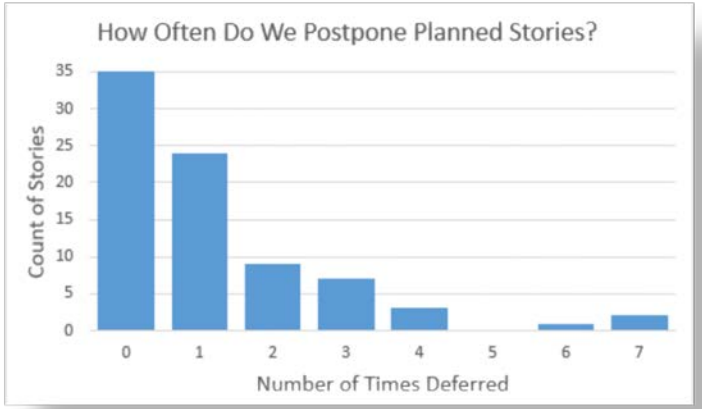
- Cadence
- Synchronization
- Short Learning Cycles
- Reduction in Batch Size
- Iterative and Incremental Delivery

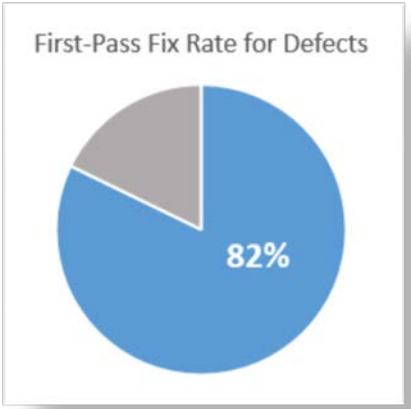


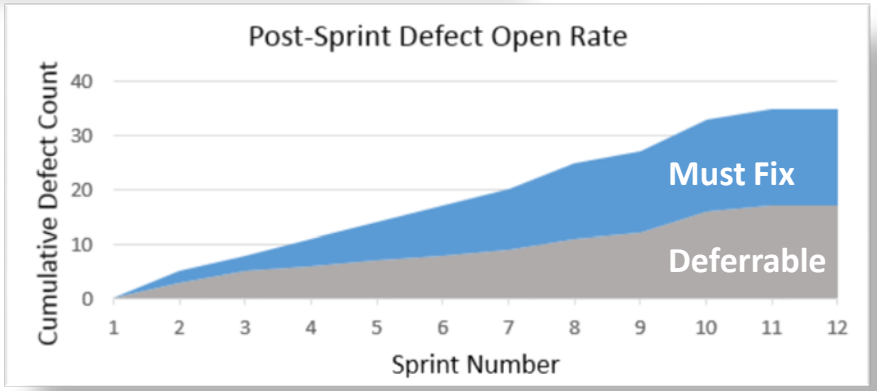




Indicator Examples$_2$

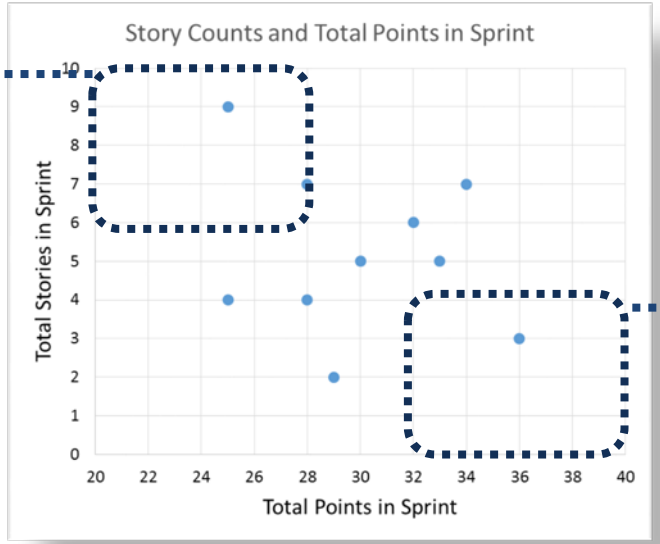






Potential Story Granularity Indicator?

Sprints with many small stories



Sprints with a few large stories

Metrics for Flow-based Product Development

Queues

- Design-in-Process Inventory
- Queue Size
- Trends in Queue Size
- Cost of Queues
- Aging of Items in Queues

Batch Size

- Batch Size
- Trends in Batch Size
- Transaction Cost per Batch
- Trends in Transaction Cost

Cadence

- Processes Using Cadence
- Trends in Cadence

Capacity Utilization

- Capacity Utilization Rate

Feedback

- Feedback Speed
- Decision Cycle Time
- Aging of Problems

Flexibility

- Breadth of Skill Sets
- Number of Multipurpose Resources
- Number of Processes with Alternate Routes

Flow

- Efficiency of Flow
- DIP Turns

Page 235: Principles of Product Development Flow: Don Reinertsen

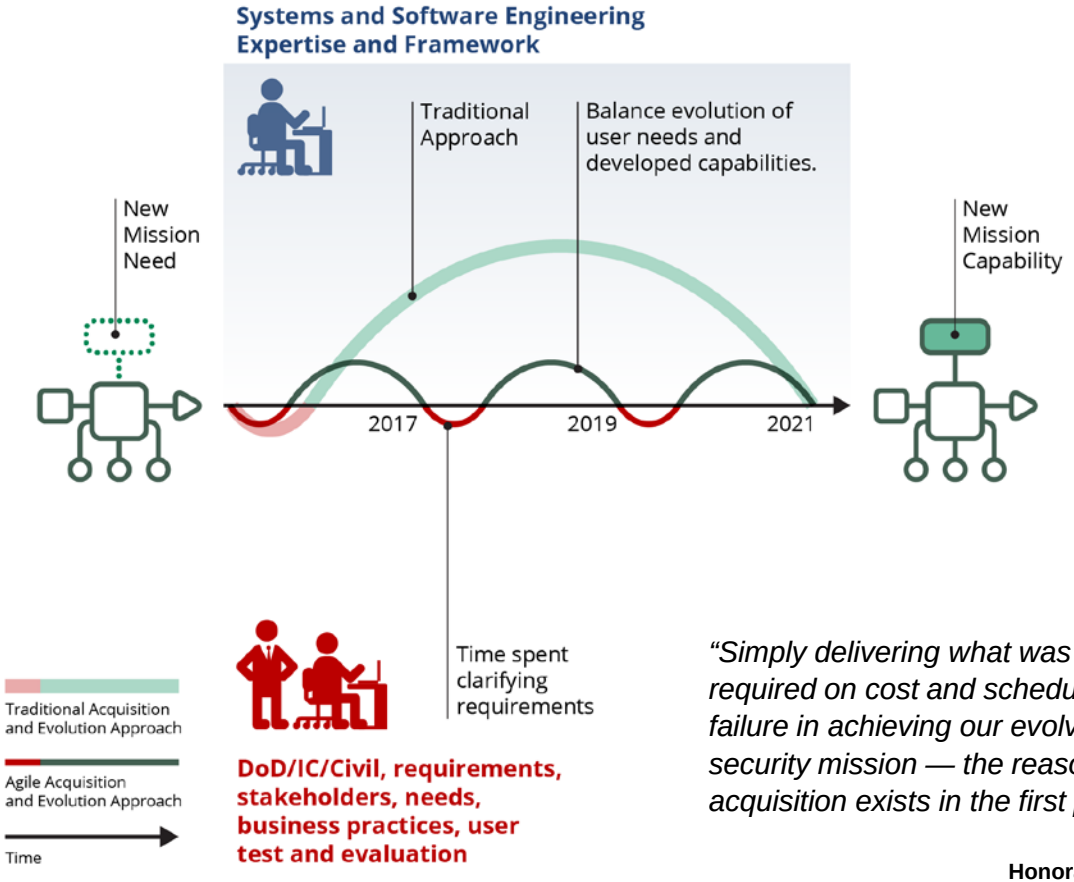
Agile In Government

Why does the DoD care?

Deliver performance at the speed of relevance

Streamline rapid, iterative approaches from development to fielding

National Defense Strategy Summary
Jan 2018



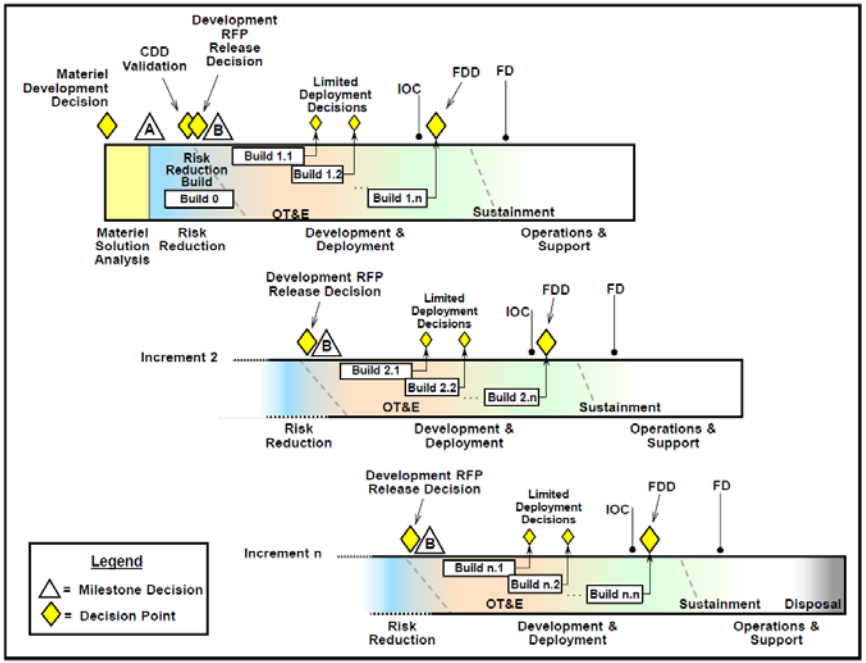
"Simply delivering what was initially required on cost and schedule can lead to failure in achieving our evolving national security mission — the reason defense acquisition exists in the first place."

Honorable Frank Kendall
Under Secretary of Defense (AT&L)
2015 Performance of The Defense Acquisition System

Multiple Intersecting Systems

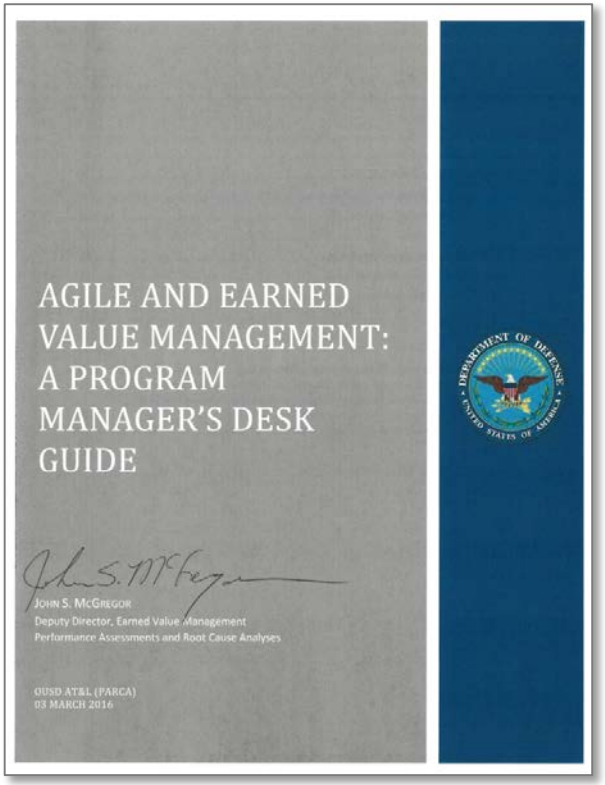
Incremental Deployment in DoD 5000.02

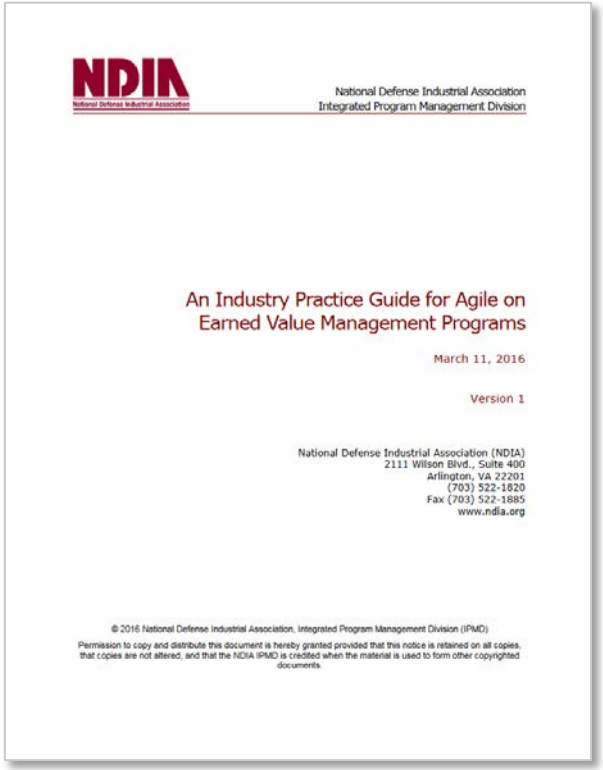
Figure 5. Model 3: Incrementally Deployed Software Intensive Program



Recognizing the growing prevalence of Agile development, government is adjusting to accommodate new approaches.

Earned Value Management for Programs Using Agile





Capability-Based Work Breakdown Structure

1.1	Prime Mission Subsystem				
1.1.1		Computer Software Configuration Item A			
1.1.1.1			CSCI Requirements Analysis		
1.1.1.2			CSCI Design		
1.1.1.3			CSCI Code and Unit Test		
1.1.1.4			CSCI Integration and Test		
1.1.2		Computer Software Configuration Item B			
1.1.3	High level Integration, Assembly, Test, and Checkout				
1.1.4	...				

Figure 1 - SW Development MIL-STD-881C Appendix K WBS breakout

1.1	Prime Mission Subsystem				
1.1.1		Capability A			
1.1.1.1			Feature A1		
1.1.1.2			Feature A2		
1.1.1.3			Feature A3		
1.1.1.4			Feature A4		
1.1.2		Capability B			
1.1.3	High level Integration, Assembly, Test, and Checkout				
1.1.4	...				

Figure 2 - Possible Agile SW Development MIL-STD-881C WBS breakout. Not prescriptive.

Figures 1&2 from Page 5, PARCA Desk Guide

The WBS reported for EVM can align to a workflow-based waterfall development oriented hierarchy as found in MIL-STD-881C* (see Figure 1). However, an outcome-based Agile structure that focuses on customer driven deliverables (see Figure 2) is also acceptable. Both WBS hierarchies are product-based and supported by the DoD EVMSIG and MIL-STD- 881C.

*DEPARTMENT OF DEFENSE STANDARD PRACTICE WORK BREAKDOWN STRUCTURES FOR DEFENSE MATERIEL ITEMS," October 2011

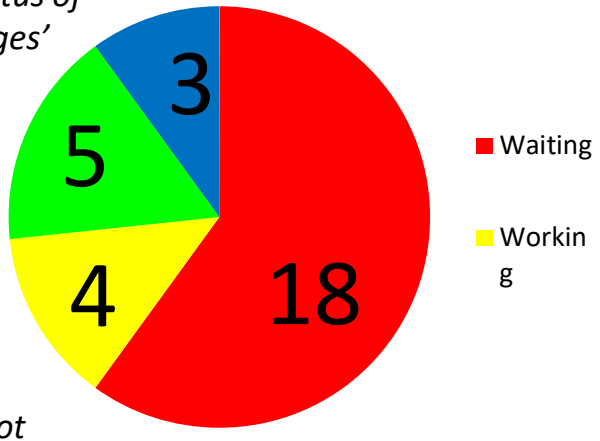
Algebra and Earned Value Management

Software and Cyber Solutions Symposium 2018

Cumulative Flow Diagram

Constructing a Cumulative Flow Diagram[1]

Here we have a Pie Chart showing the status of 30 'work packages'



This is a snapshot for a single point in time.

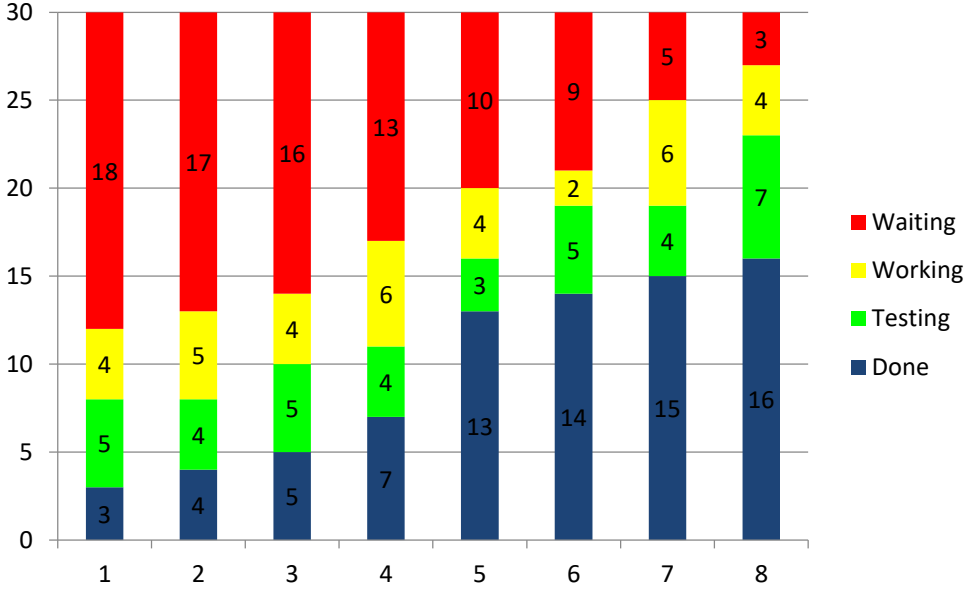
Constructing a Cumulative Flow Diagram$_2$

Same data, but presented in a stacked column chart

For a single point in time.

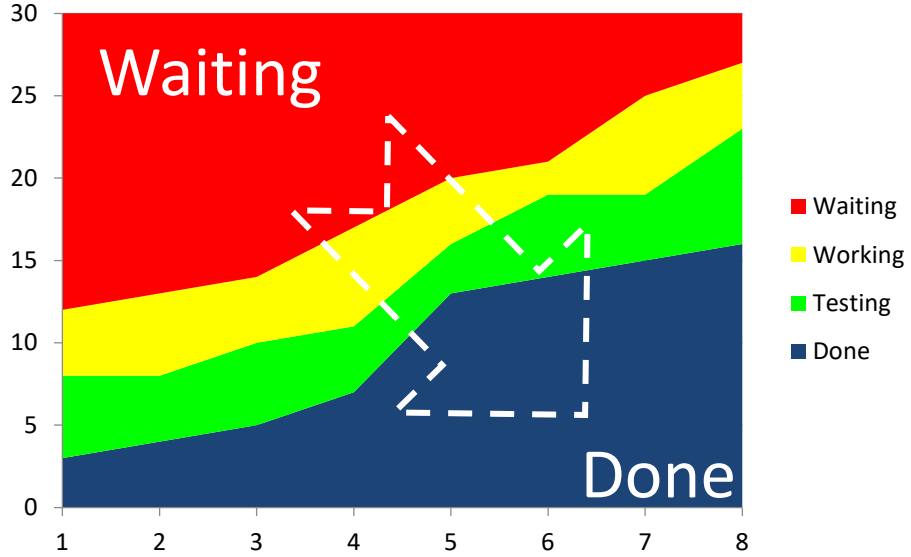
Constructing a Cumulative Flow Diagram$_3$

… adding the next 7 times



Constructing a Cumulative Flow Diagram$_4$

… now we are looking at the flow from "Waiting" to "Done"…
This view starts to show patterns a little easier…

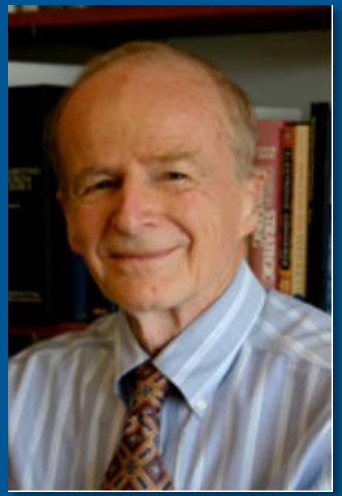


Theoretical Basis

Little's Law

$$L = \lambda W$$

…the long-term average number L of customers in a stationary system is equal to the long-term average effective arrival rate λ multiplied by the average time W that a customer spends in the system…



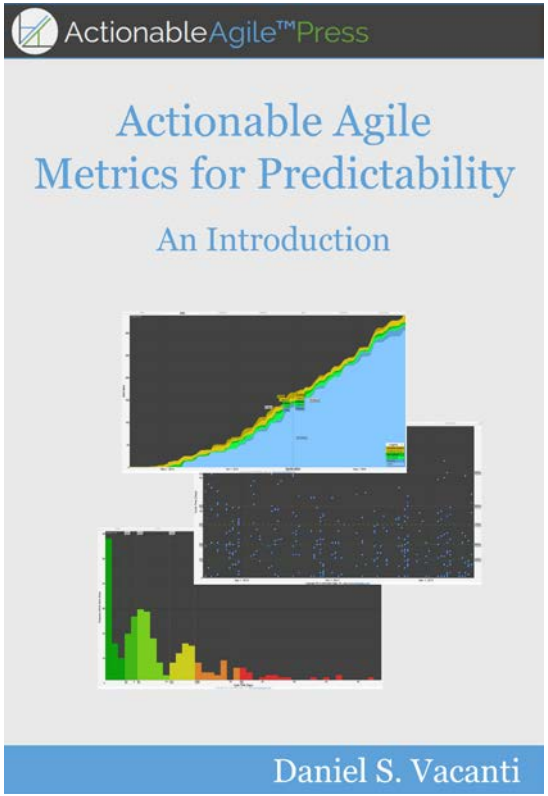
http://mitsloan.mit.edu/faculty-and-research/faculty-directory/detail/?id=41432

Little's Law in Agile Metrics

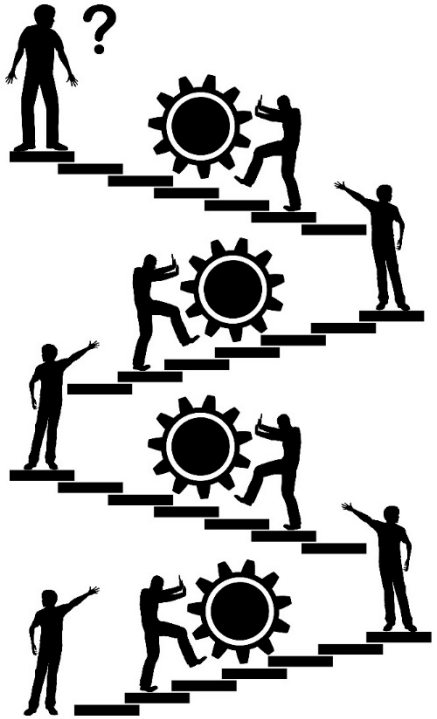
Three Metrics Emphasized*:

1. **Work In Progress** (the number of items that we are working on at any given time),
2. **Cycle Time** (how long it takes each of those items to get through our process), and
3. **Throughput** (how many of those items complete per unit of time).

** Excerpted from page 13 of the book depicted on the right.*



Barriers to Actionable Metrics



Metrics often focus exclusively on:

- Appeasing an authority role
- Demonstrating competence
- Validating the chosen path

This may engender trust concerns, and often conflicts with the concept of an empirical process – one where we learn from looking at facts that inform tactical/strategic options.

Software and Cyber Solutions Symposium 2018

Why Agile Metrics? Are there “Waterfall Metrics?”

Opportunities Created by Agile Approaches

Greater user involvement – even if through a ‘Product Owner’ proxy

- Focus on quality attributes valued by the user – as user stories are written
- Confirm level of utility in what is demonstrated at the end of each iteration

More frequent delivery of products – even if early drops are to a simulation environment

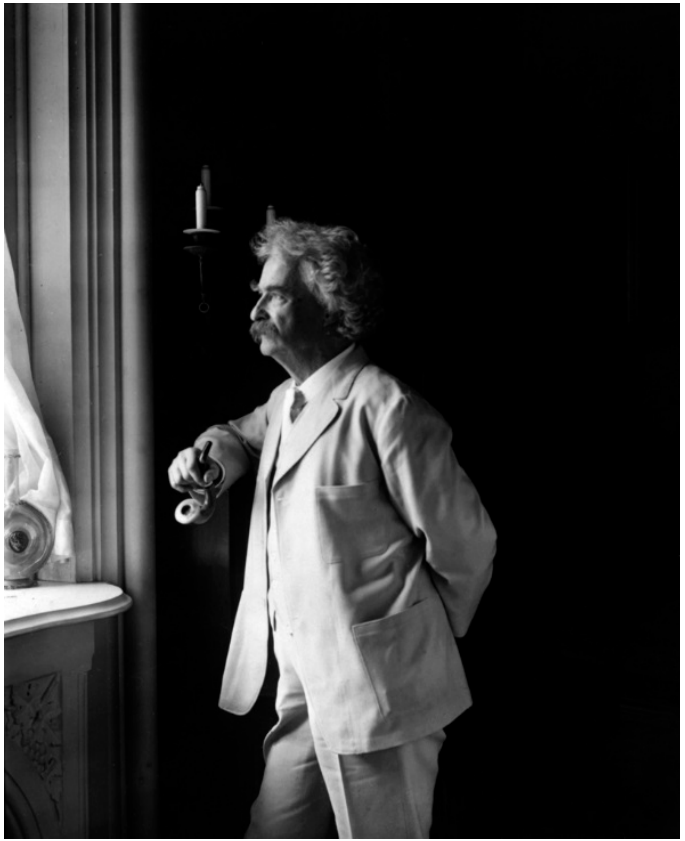
- Verify requirements earlier, with more opportunities to recover when needed
- Integration events of smaller scope help mitigate risk of quality-driven delays
- Operational benefit from new capabilities can be validated incrementally as well

Experience suggests that simply asking “is it going according to plan?” is inadequate

Deterministic Thinking

A World Full of Choices

According to Mark Twain…



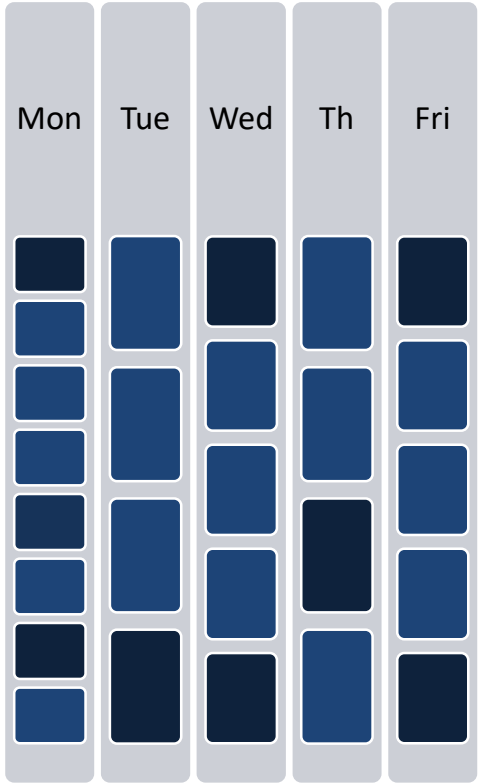
It ain't what you don't know that gets you in trouble. It's what you know for sure that just ain't so.

Samuel Langhorne Clemens

Software and Cyber Solutions Symposium 2018

Utilization vs. Flow

Value Flow: Utilization is the Wrong Goal



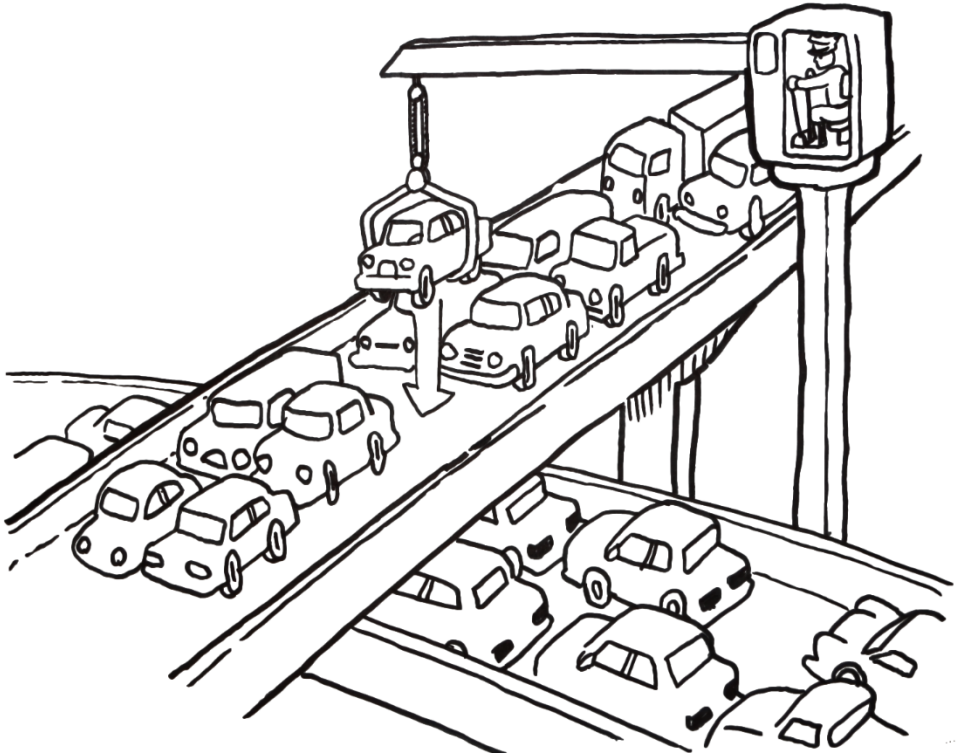
100% Utilization:

- Magnifies the impact of variation
- Maximizes task-switching overhead
- Assures slower overall progress

Change is inevitable, plan to learn

Multi-tasking is a myth we don't accurately comprehend

Maximum Utilization is Counterproductive



Program Management

Large scale program management involves on-going balancing of resources brought to bear against priorities.

- Pushing for resolution of estimates and actuals below a certain level of detail reveals the boundaries of interpretability for the Quantifiable Backup Data used to manage
- Large grained estimates for program budgets and short-term forecasts for team workload do not share the same estimation basis, nor do they respond to the same type of corrective actions

Software and Cyber Solutions Symposium 2018

Precision vs. Consistency

Build Incrementally, Adjusting Along the Way

Hone the approach as we iterate

Build from initial minimum result to robust capabilities

Plan to learn and adjust to increase value of outcomes

Work Prioritization & Size Estimation – Iterative Development

User Value Assessment

The relative value of features in the backlog can be assessed by assigning “business value” measures.

These measures are used to adjudicate among features competing to “go next.”

Features are accepted into the workflow one at a time, or in groups.

We need a way to consistently determine which feature (or group of features) is more important.

Size Estimation

Teams using story point estimation are taught to focus on relative magnitudes, not absolute volume of work being estimated.

Fibonacci points (1, 2, 3, 5, 8, 13, 21) often used, as well as “ideal hours/days.”

User Stories are selected for work with the question “does this fit in the time I have?”

We need to commit to accomplishable work scope each iteration, and make reasoned tradeoffs in scope when necessary.

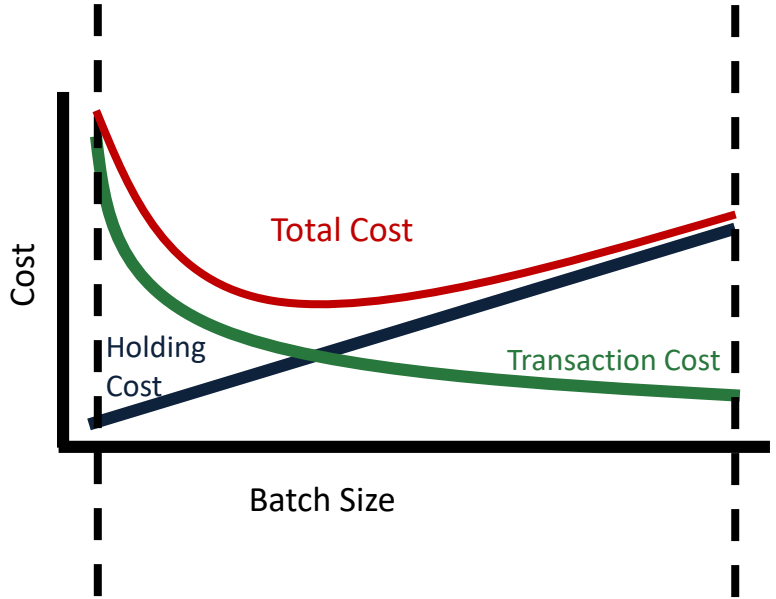
Software and Cyber Solutions Symposium 2018

Holding Costs and Transaction Costs

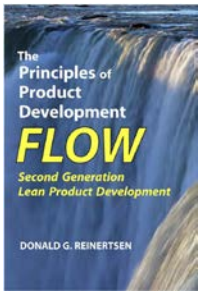
Economies of Batch Size

Specify, build test & ship a **SINGLE** requirement

Specify, then implement, then test & then ship **ALL** requirements



U-Curve optimization problem as described in *Principles of Product Development Flow*, by Don Reinertsen



Inventory of Partially Complete Work

Traditionally, *projective documents* like requirements and design specifications represent the backlog of software inventory.

These reflect decisions and constraints which have not yet been validated.

Holding costs for these documents are not well understood.

Transaction Costs are Visible

The cost of personnel and infrastructure to perform the work can be readily assessed

- In fact, we are biased toward amortizing these effectively because they are visible

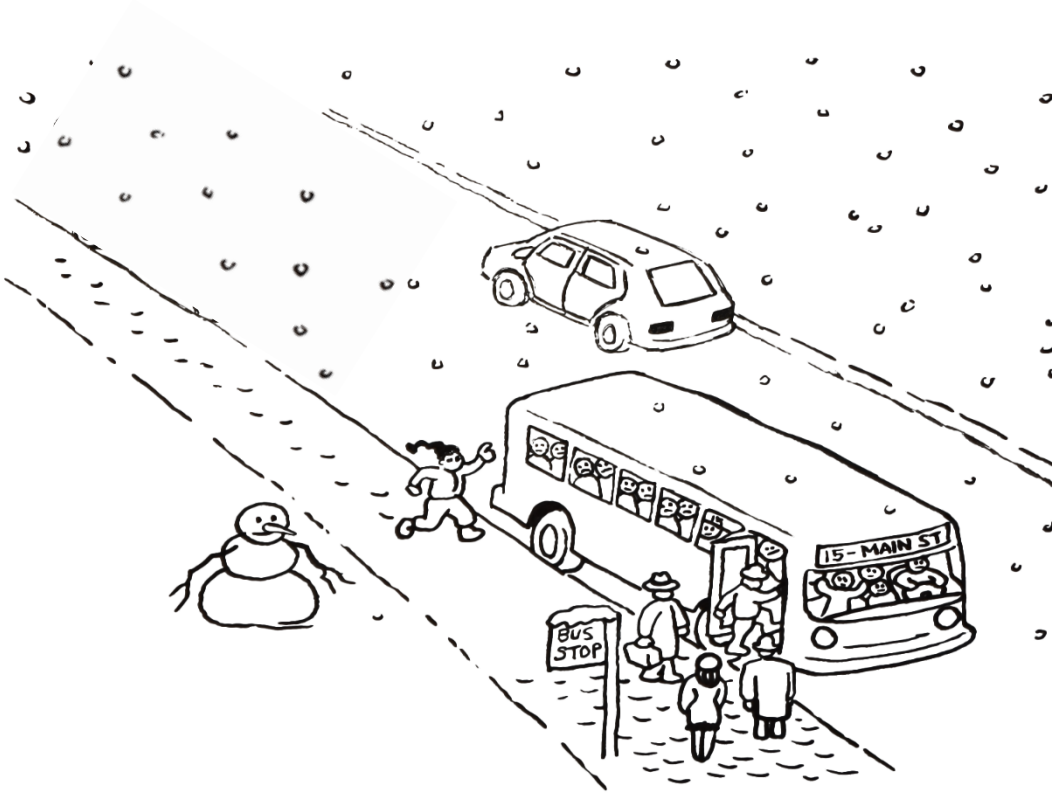
DevOps offers many opportunities for making economic gains by addressing transactions

- Collaborate with developers and operations staff for continuity
- Use tools to assure reliability and consistency

Software and Cyber Solutions Symposium 2018

Optimizing for Continuous Integration & Continuous Delivery

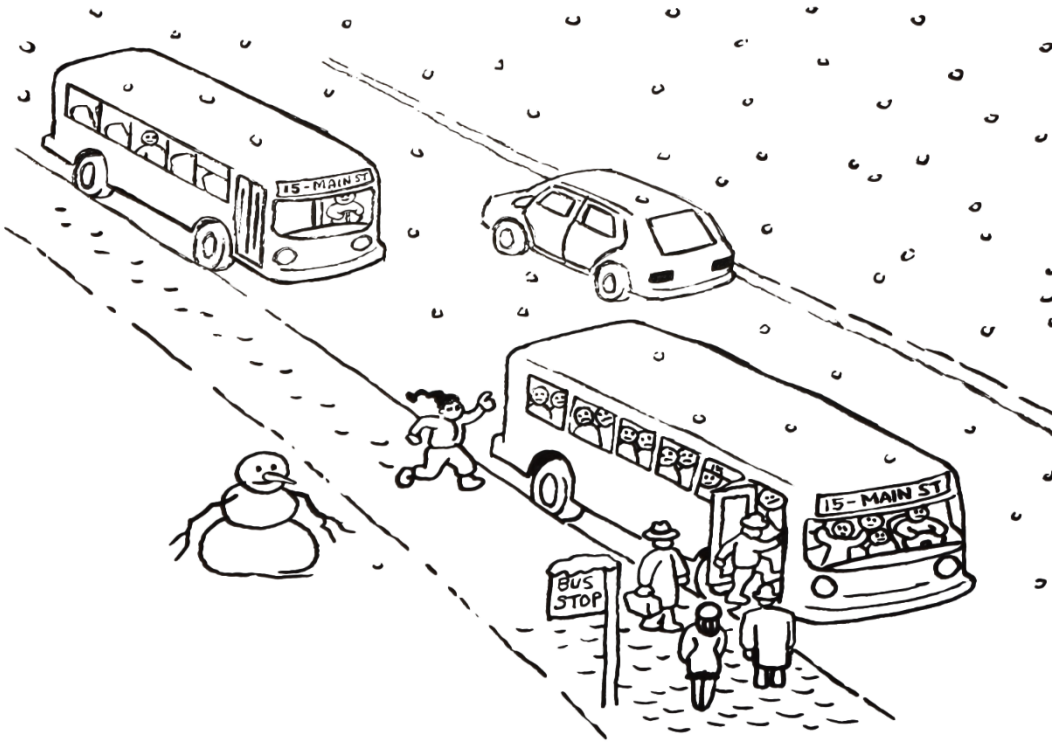
Cadence Enhances Predictability



A Late Bus:

- Makes people scramble to get aboard
- They don't know when the next one will get here

Cadence Enhances Predictability

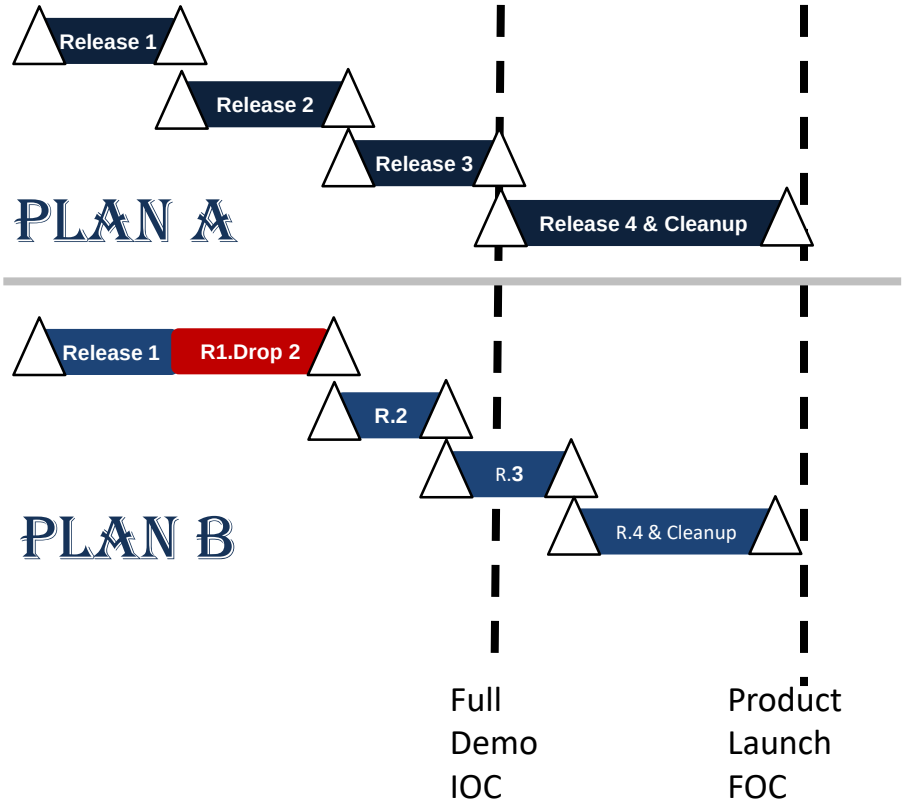


A Late Bus:

- Makes people scramble to get aboard
- They don't know when the next one will get here

Then the next bus comes along empty

Late Releases Become “Feature Magnets”



As things start to slip

- Influential people get ‘their priorities’ moved up, rather than deferred
- Pressure increases on early releases
- Functions slated for final release can’t be guaranteed…

Discipline is Required

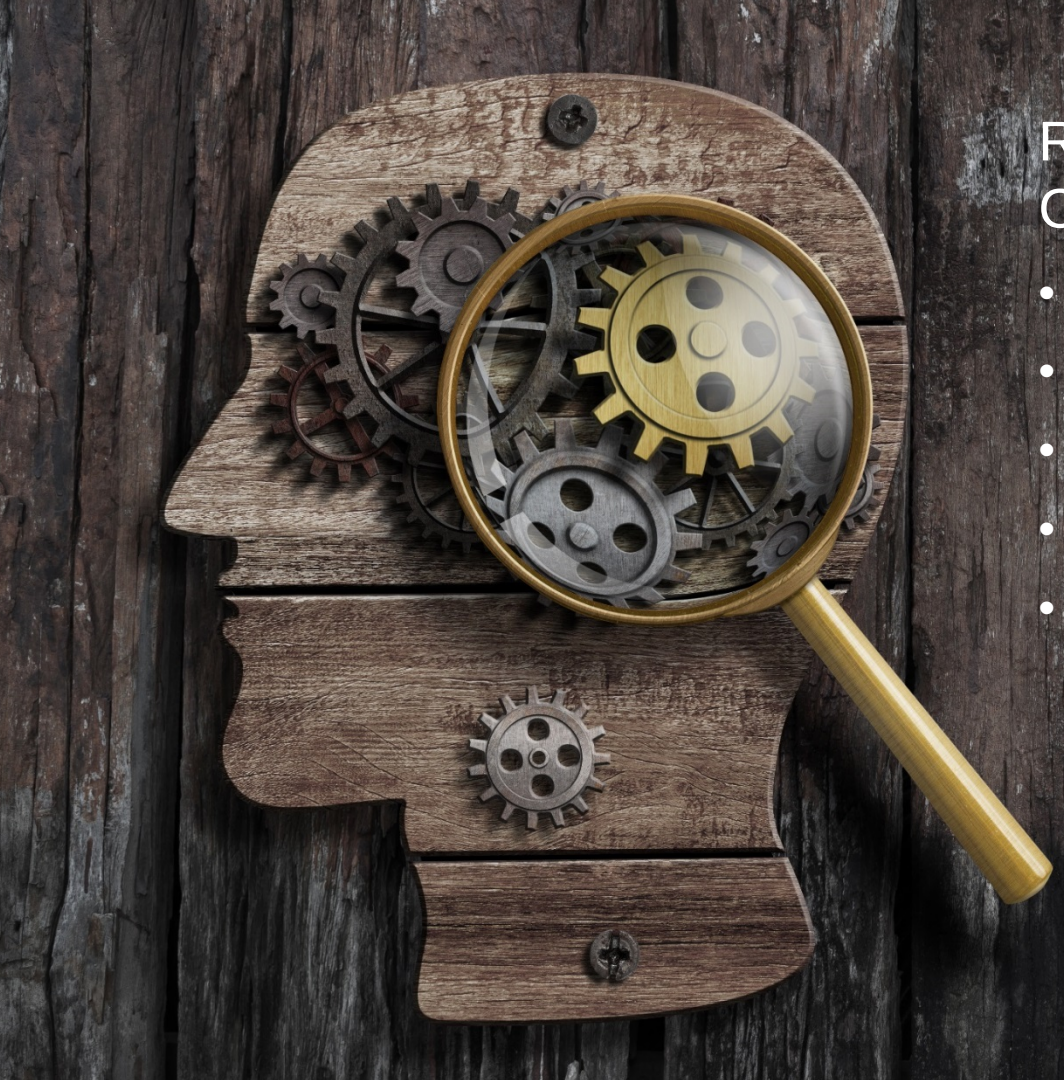
Economic Framework

Prioritizing based on the ***Cost Of Delay*** enables a focus on flow.

- Attributes of value derived from the delivered work contribute to decision-making
  - Desirability
  - Time Criticality
  - Risk Reduction
  - Opportunity Enablement
- Priorities chosen in to balance varying work item size and importance.

Software and Cyber Solutions Symposium 2018

Conclusions



Rapid Incremental Delivery Changes Priorities

- Optimize Proximity of Talent
- Simplicity is Robust
- Test is Documentation
- Agile is not Local
- Innovation is a Forcing Function

Call To Action

Understand the Full Value Stream

- Metrics can lead to local optimization at global expense, if you are not careful
- Many traditional metrics focus on accuracy of forecasts, not performance

Set Goals

- Aspire to succeed for your users, not just the funding authority
- Measure successive approximations of the desired end state

Experiment and Learn

- An empirical approach means we look at outcomes and act

Performance Matters Most