

Lesson 2

Applying SAFe Principles: A Scrum Master's Perspective

1. Exploring the Scrum Master Role in the SAFe Enterprise
2. Applying SAFe Principles:
A Scrum Master's Perspective
3. Exploring Agile and Scrum Anti-Patterns
4. Facilitating Program Execution
5. Improving Flow with Kanban and XP
6. Building High-Performing Teams
7. Improving Program Performance with Inspect and Adapt

SAFe® Course Attending this course gives learners access to the
SAFe® Advanced Scrum Master exam and related preparation materials.

SAFe Lean-Agile principles

#1 - Take an economic view

#2 - Apply systems thinking

#3 - Assume variability; preserve options

#4 - Build incrementally with fast, integrated learning cycles

#5 - Base milestones on objective evaluation of working systems

#6 - Visualize and limit WIP, reduce batch sizes, and manage queue lengths

#7 - Apply cadence, synchronize with cross-domain planning

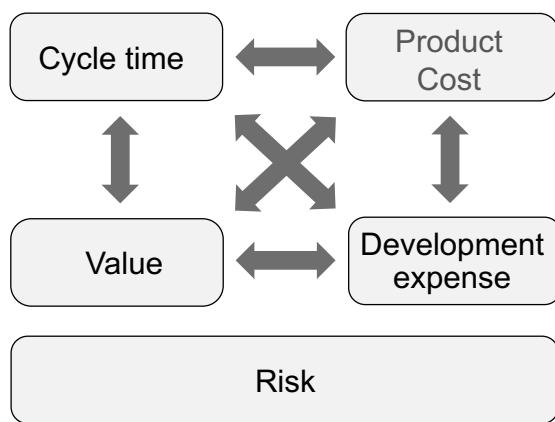
#8 - Unlock the intrinsic motivation of knowledge workers

#9 - Decentralize decision-making

#1 Take an economic view

Base decisions on economics

Many decisions at the Team Level impact development economics.

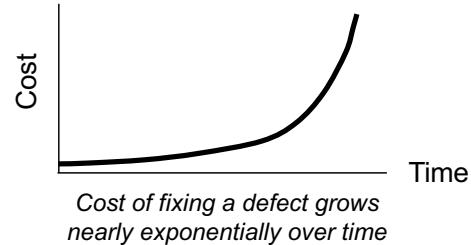


- ▶ Validate key decisions based on how they impact the five variables
- ▶ Team makes thousands of decisions during the PI
- ▶ Effective team process leverages more economically-feasible ways of product development
- ▶ Taking an economic view does not always require knowing 'dollarized value' but is rather a general thinking tool

Consider examples ...

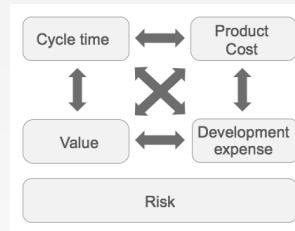
Following are examples of indicators that may have surprisingly high economic impact in the long run:

- ▶ Cost of late defect fixing
- ▶ Cost of branching with late merge
- ▶ Cost of delayed performance testing
- ▶ Cost of large batch of cross-team dependencies
- ▶ Economic value of test automation
- ▶ Economic value of 'Enablers,' such as research spikes, refactors, etc.



Exercise: Take an economic view

1. Cost of late defect fixing
2. Cost of branching with late merge
3. Cost of delayed performance testing
4. Cost of large batch of cross-team dependencies
5. Economic value of test automation, Enablers, such as research spikes, refactors, etc.



#2 Apply systems thinking

Systems thinking

- ▶ Complex systems development requires disciplined, systematic systems thinking
- ▶ Optimizing a component does not optimize the system
- ▶ The value of a system passes through its interconnections
- ▶ A system can evolve no faster than its slowest integration point
- ▶ Understand and optimize the full Value Stream



The Enterprise that builds systems is, itself, a system.



The systems the Enterprise builds are systems, too.

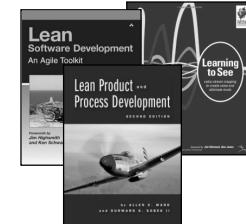
Take a systems view on value delivery

All we are doing is looking at the timeline, from when the customer gives us an order to when we collect the cash. And we are reducing the timeline by reducing the non-value added wastes.

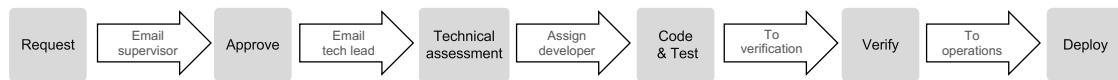
—Taiichi Ohno

Understand the full Value Stream:

- Most inefficiencies and impediments in your process will surface themselves as *delays* in value delivery
- Consider all steps as part of the Value Stream, including definition, analysis, validation, deployment, and release
- Customers and suppliers are part of your Value Stream
- Establish a culture of relentless improvement of the *full* Value Stream



Poppoendieck, Ward et al,
Rother, Shook, Womack

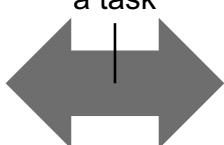


SCALED AGILE® © Scaled Agile, Inc.

2.9

What are cycle times?

Time to finish
a task



Reduce 'cycle time'

Make decisions that result in a **reduction of cycle times**
(total time to complete something and get value from it)

Time to finish a Story



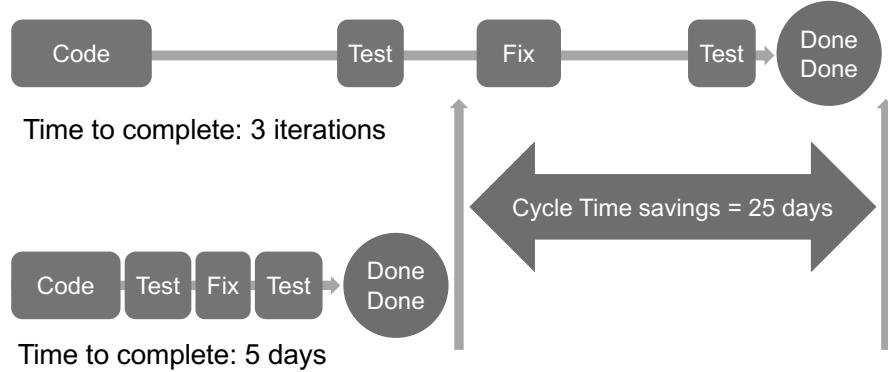
Time to deploy to production

SCALED AGILE® © Scaled Agile, Inc.

2.10

Reduce cycle times/improve flow!

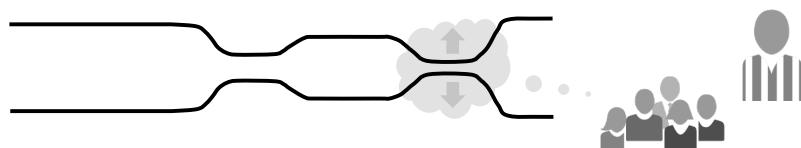
A tale of two Stories



Move from bottleneck to bottleneck

I say an hour lost at a bottleneck is an hour out of the entire system. I say an hour saved at a non-bottleneck is worthless. Bottlenecks govern both throughput and inventory.

—Eliyahu M. Goldratt, *The Goal*



- ▶ The Scrum Master is an Enabler who helps the team identify and remove bottlenecks.
- ▶ Every system has only one or two bottlenecks that significantly constrain performance.
- ▶ Once you have identified and removed the current bottleneck, there will be another one, but the system is already operating at a higher level of performance!

Exercise: More than a team...

- ▶ Select a part of your development process (could be any part, example below)



- ▶ Reason about ways you could increase flow by reducing WIP and eliminating bottlenecks

- ▶ Pair and share



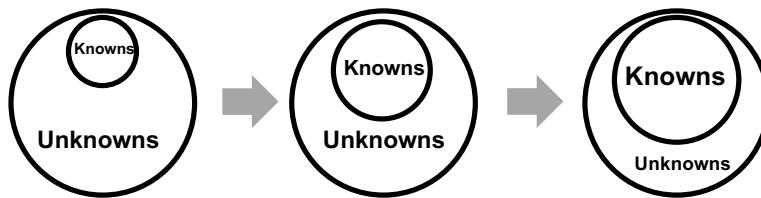
#3 Assume variability; preserve options

Development occurs in an uncertain world

Aggressively evaluate alternatives. Converge specifications and solution sets.

—Allen Ward

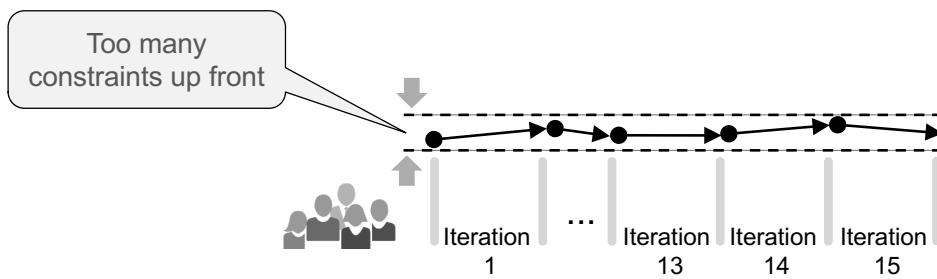
- ▶ You cannot possibly know everything at the start
- ▶ Requirements and designs must be flexible to build an optimum solution
- ▶ Iterative and incremental development aims at reducing uncertainty over time



Common problem of many organizations

When Agile practices are adopted on top of a traditional, phase-gate mindset, teams end up with a typical problem:

- ▶ They follow an iterative and incremental development model while committing to a specific Solution option early in the process
- ▶ As a result, the power of Agile is significantly underused and reduced to applying only minor adjustments



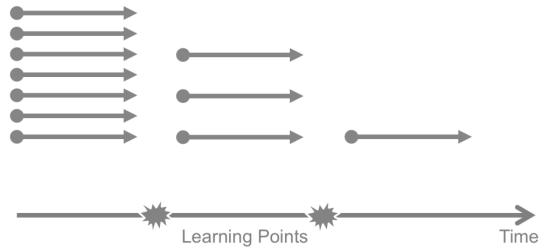
Different thought process is needed

'Just-in-Time' elaboration of requirements and design

- ▶ Not everything should be defined at once
- ▶ Better requirements and implementation options will emerge over the course of Iterations
- ▶ Up-front thinking is not enough; 'learning by doing' must extend the paradigm

Set-based design

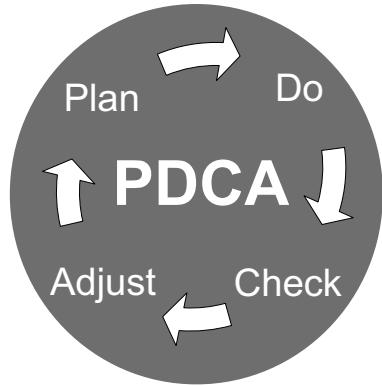
- ▶ If you can't be right early on, preserve multiple options until you have more certainty
- ▶ Narrow them down over the course of the learning points and Iterations



#4 Build incrementally with fast, integrated learning cycles

Apply fast learning cycles

Fast feedback accelerates knowledge.



The iterative learning cycle

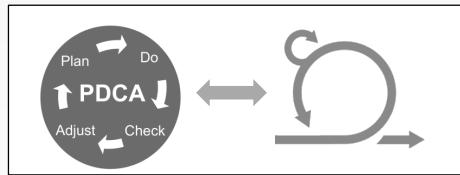
Principles of Product Development Flow, Don Reinertsen
Plan. Do. Check. Act. (Adjust), W. Edwards Deming, Walter Shewhart, et al.

- ▶ Improves learning efficiency by decreasing the time between action and effect
- ▶ Reduces the cost of risk-taking by truncating unsuccessful paths quickly
- ▶ Facilitated by small batch sizes
- ▶ Requires increased investment in development environment

The shorter the cycles, the faster the learning

Exercise: PDCA cycle in Scrum

- ▶ Create mapping between the PDCA cycle and the Scrum flow on a flip chart or whiteboard
- ▶ What aspects of your process contribute the most to learning?
- ▶ What would you like to improve in your Scrum process to facilitate learning?

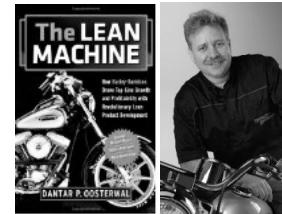
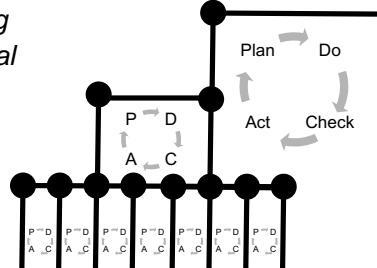


Apply fast learning cycles

Product development is the process of converting uncertainty to knowledge —Dantar P. Oosterwal

Integration points control product development

- ▶ Integration points accelerate learning
- ▶ Development can proceed no faster than the slowest learning loop
- ▶ Improvement comes through synchronization of design loops and faster learning cycles

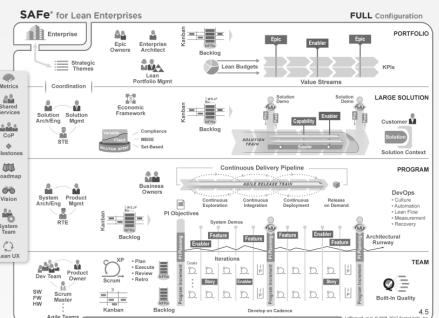


The Lean Machine:
How Harley Davidson Drove
Top-Line Growth and
Profitability with
Revolutionary Lean
Product Development

—Dantar P. Oosterwal

Exercise: PDCA on the Big Picture

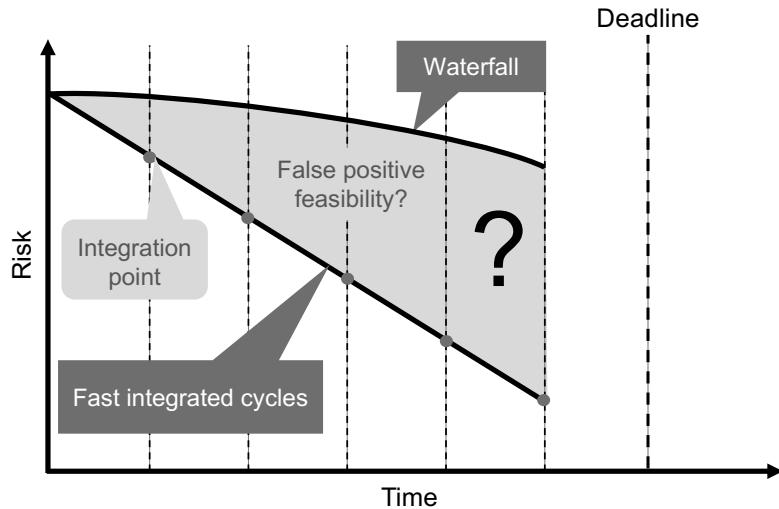
- ▶ As a group, discuss where on the Big Picture you have PDCA cycles.
- ▶ Which cycles do you think your team should participate in directly?



PREPARE
5 min

SHARE
2 min

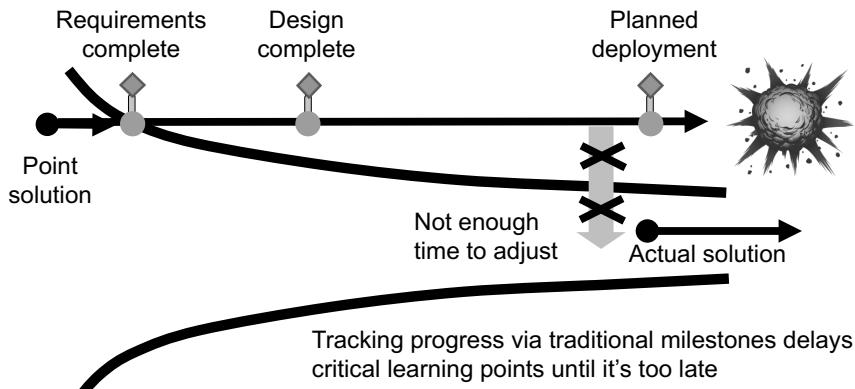
Integration points reduce risk



#5 Base milestones on objective evaluation of working systems

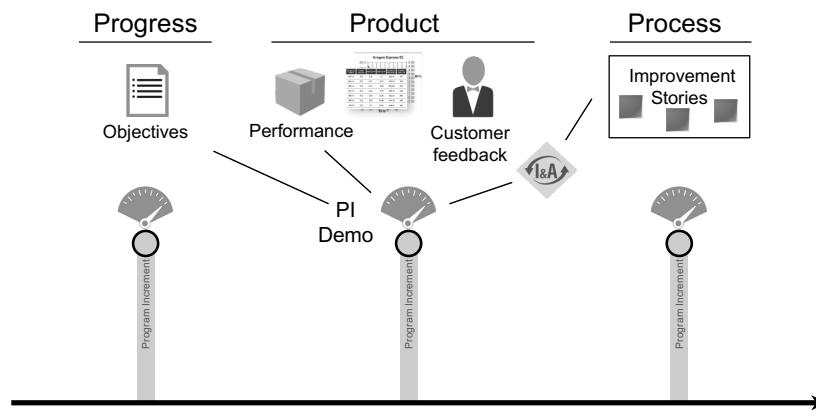
The problem of phase gate milestones

Phase gates fix requirements and designs too early, making adjustments costly and late as new facts emerge.



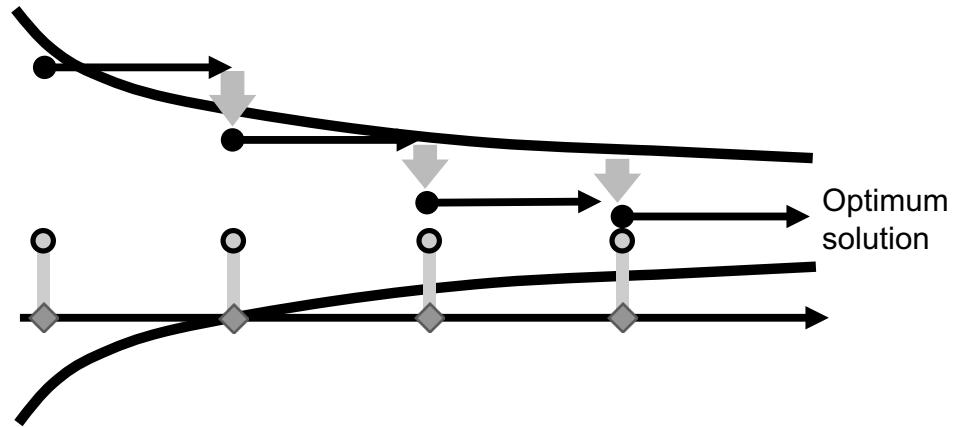
Apply objective Milestones

PI Demos are orchestrated to deliver objective progress, product, and process metrics.



Iterate to the optimum solution

Objective Milestones facilitate learning and allow for continuous, cost-effective adjustments towards an optimum solution.



#6 Visualize and limit WIP, reduce batch sizes, and manage queue lengths

Exercise: Cycle time

1. Fold the Paper
2. Stuff the Envelope
3. 'Seal' the envelope (simulate)
4. Initial the envelope



How can we reduce cycle times?

- ▶ Reduce size of work
- ▶ Reduce bottlenecks
- ▶ Reduce waiting
- ▶ Increase swarming
- ▶ Improve quality

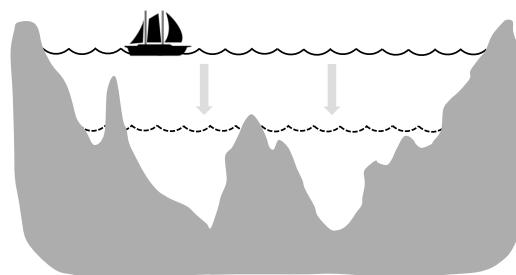




Reduce WIP!

Work in process (WIP) and flow

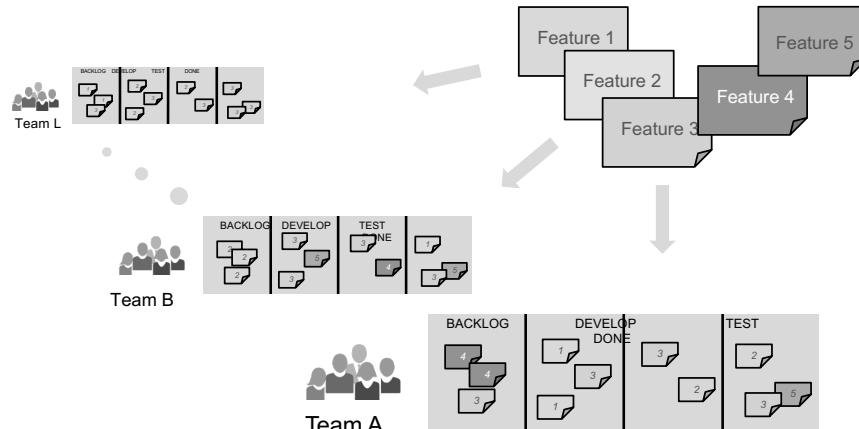
When there's too much WIP, there's no visibility into the bottlenecks and the system is usually highly inefficient.



At the surface, however, it all looks smooth and promising.

Too much WIP slows down the enterprise

WIP levels of the team and the program are connected. Understanding the connection creates improvement opportunities.

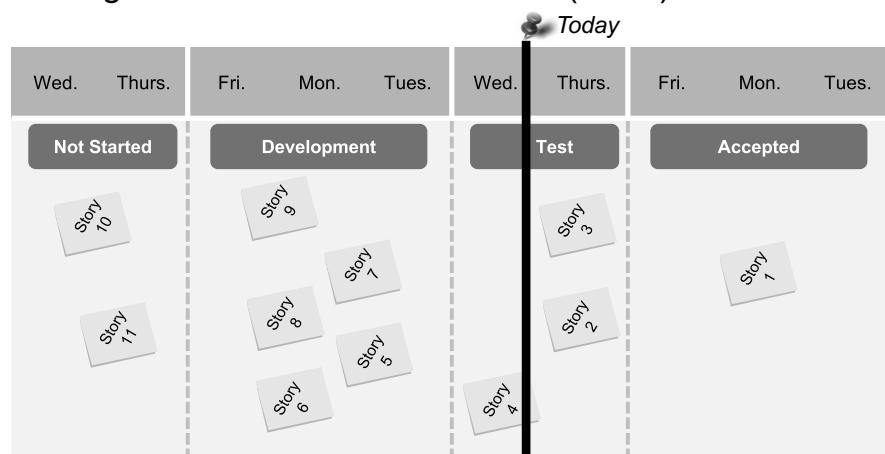


SCALED AGILE® © Scaled Agile, Inc.

2.33

Visualize and limit WIP

One team's Big Visible Information Radiator (B VIR)



How is this team doing? How do you know that?

SCALED AGILE® © Scaled Agile, Inc.

2.34

Exercise: WIP improvement opportunities

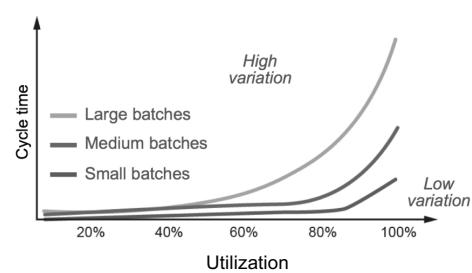
- ▶ Identify the improvement opportunities on the previous slide
 - Look carefully and determine how can we better manage WIP?
 - Did you identify the improvement opportunity Development?
 - What behavior would adding a WIP limit for Development cause?



The importance of small batches

Small batches go through the system faster, with lower variability.

- ▶ Large batch sizes increase variability
- ▶ High utilization increases variability
- ▶ Severe project slippage is the most likely result
- ▶ Most important batch is the transport (handoff) batch
- ▶ Proximity (co-location) enables small batch size
- ▶ Good infrastructure enables small batches

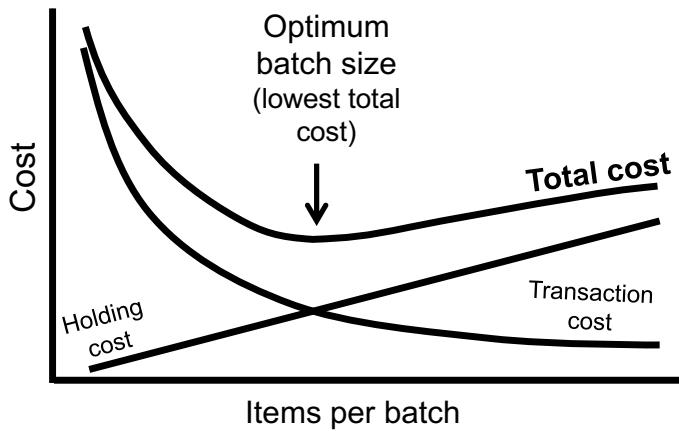


Implementing Lean Software Development,
Mary Poppendieck

Principles of Product Development Flow,
Don Reinertsen

Finding optimum batch size

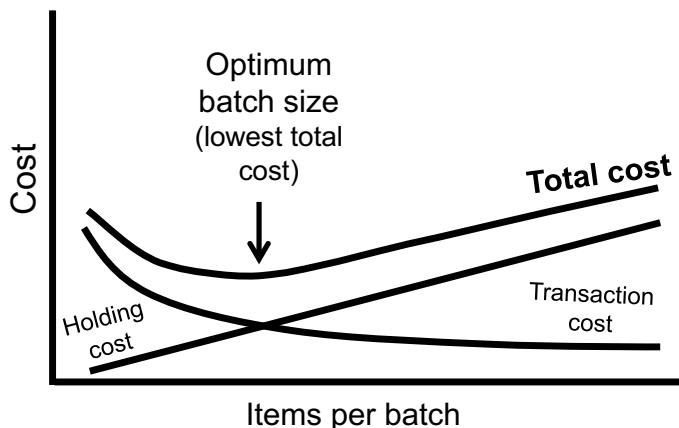
Optimum batch size is an example of a U-curve optimization.



- ▶ Total costs are the sum of holding costs and transaction costs
- ▶ Higher transaction costs shift optimum batch size higher
- ▶ Higher holding costs shift batch size lower

Reducing optimum batch size

Reducing transaction costs reduces total costs and shifts optimum batch size lower.



- ▶ Reducing batch size:
 - Increases predictability
 - Accelerates feedback
 - Reduces rework
 - Lowers cost
- ▶ Batch size reduction probably saves twice what you think



Reducing transaction costs example
https://youtu.be/RRy_73ivcms
2:09

Manage queue lengths

Email from a client service organization:

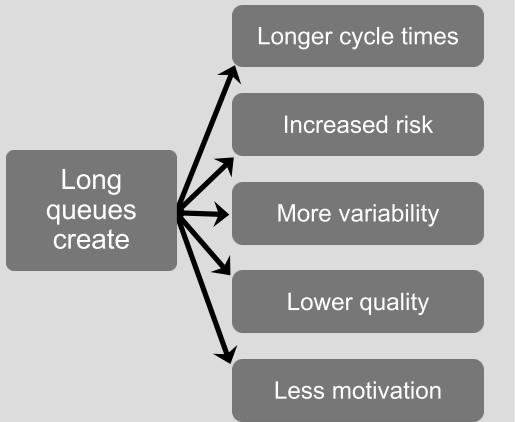


Thank you for contacting us.

*We are experiencing increased volumes
and apologize in advance for the delay.*

Our goal is to contact you within . . .

Long queues: All bad



Principles of Product Development Flow, Don Reinertsen

Reduce queue lengths

- ▶ Understand Little's Law
- ▶ Faster processing time decreases wait
- ▶ Control wait times by controlling queue lengths

$$W_q = \frac{L_q}{\lambda}$$

Average wait time = average queue length/average processing rate

Example: given average processing speed of 10 Features per quarter and a committed set of 30 Features, a new Feature will experience approximate wait time of:

$$\frac{30 \text{ items}}{10 \text{ items/Q}} = 3Q$$

#7 Apply cadence, synchronize with cross-domain planning

Cadence and synchronization

Cadence

- ▶ Transforms unpredictable events into predictable events
- ▶ Makes waiting times predictable
 - If you can't predict delivery, existing programs become *Feature magnets*
- ▶ Facilitates planning; provides more efficient use of resources
- ▶ *Delivering on cadence requires scope or capacity margin*



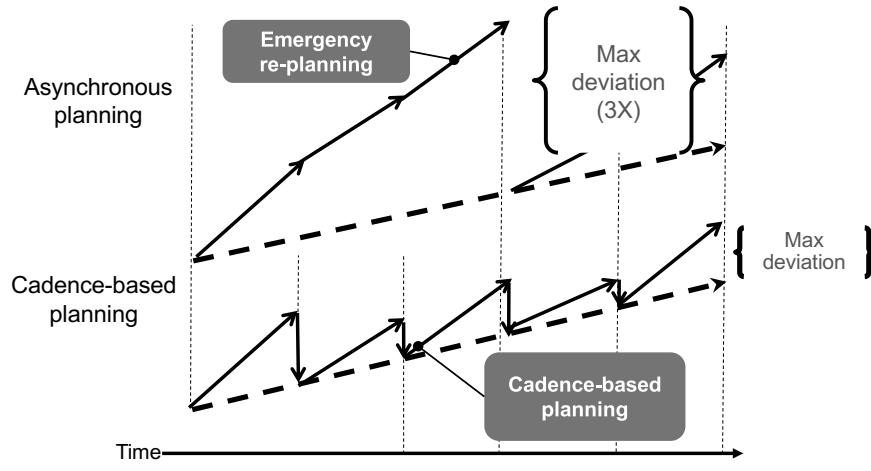
Synchronization

- ▶ Synchronization causes multiple events to happen at the same time
- ▶ Sync events facilitate cross-functional tradeoffs of people and scope
- ▶ Periodic resynchronization limits variance to a single time interval
- ▶ Regular integration provides high fidelity tests and objective assessment
- ▶ *To work effectively, design cycles must be synchronized*

Principles of Product Development Flow, Don Reinertsen
The Lean Machine, Dantar Ootserwall

Control variability with planning cadence

Cadence-based planning limits variability to a single interval.



Synchronize with cross-domain planning

Future product development tasks can't be pre-determined. Distribute planning and control to those who can understand and react to the end results.

— Michael Kennedy, *Product Development for the Lean Enterprise*

- ▶ All stakeholders face-to-face, (but typically multiple locations)
- ▶ Management sets the mission, with minimum possible constraints
- ▶ Requirements and design happen
- ▶ Important stakeholder decisions are accelerated
- ▶ Teams create—and take responsibility—for plans



#8 Unlock the intrinsic motivation of knowledge workers

Drive: The puzzling puzzles of Harry Harlow

The 1949 experiment

- ▶ Eight rhesus monkeys for a two-week experiment on motivation and learning
- ▶ Puzzles were placed in their cages



Results

- ▶ Unprompted, the monkeys solved the puzzles on their own
- ▶ An interesting and not understood phenomenon
- ▶ As a motivator, raisins were added as rewards
- ▶ Result: the monkeys made more errors and solved the problems less frequently

It appears that the performance of the task provides its own intrinsic reward ... this drive ... may be as basic as the others —The Surprising Truth About What Motivates Us, Daniel H. Pink

#9 Decentralize decision-making

Decentralize decision-making

Any inefficiency of decentralization costs less than the value of faster response time
—Principles of Product Development Flow, Don Reinertsen

- ▶ Centralize decisions that are infrequent, long-lasting, and have significant economies of scale
- ▶ Decentralize all others:
 - Frequent decisions
 - Time-critical decisions
 - Decisions that require local information
- ▶ Define the economic logic behind a decision; empower individuals and teams to actually make them



Exercise: Decentralized decision-making

- ▶ Identify two to three currently-centralized decisions that affect your team and would benefit from decentralization
- ▶ Pick one of them and justify the benefits of decentralization
- ▶ Identify the stakeholder(s) who would help you apply a decentralized model to decision-making in this case

PREPARE | SHARE

5 min | 2 min

Exercise: Building a SAFe Principles poster

- ▶ Each table will be building a poster using flip charts and colored markers (no whiteboards)
- ▶ The poster will contain four out of nine SAFe Principles that you think are the most important
- ▶ Every Principle (out of the selected 4) contains:
 - Principle statement (ex: “Take an economic view”)
 - At least one picture that exemplifies the Principle
 - A concise statement about how the Scrum Master can apply the Principle
- ▶ Posters should fit on a single flip chart sheet



PREPARE | SHARE

17 min | 3 min

Principles are great, but...

*Clarity on how to think, without clarity on how to act,
leaves people unmoved.*

—Dan Pink

... it's time to put this thinking to work.
Let's move to doing.

Lesson summary

#1-Take an economic view

#2-Apply systems thinking

#3-Assume variability; preserve options

#4-Build incrementally with fast, integrated learning cycles

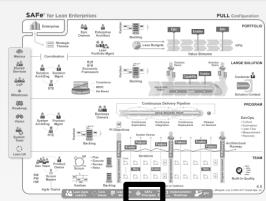
#5-Base milestones on objective evaluation of working systems

#6-Visualize and limit WIP, reduce batch sizes, and manage queue lengths

#7-Apply cadence, synchronize with cross-domain planning

#8-Unlock the intrinsic motivation of knowledge workers

#9-Decentralize decision-making



Suggested Scaled Agile

Framework reading:

"SAFe Principles" article