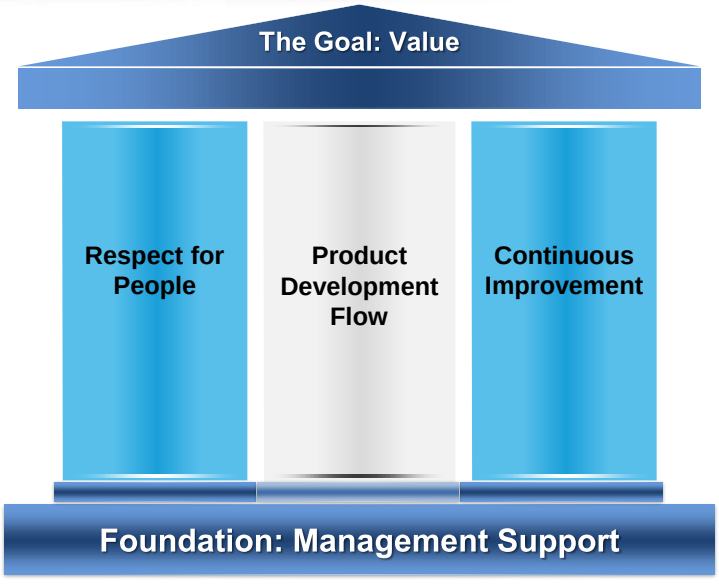
SAFe Lean Thinking

House of Lean &
Product Development Flow

Scaled Agile Academy
July 2012

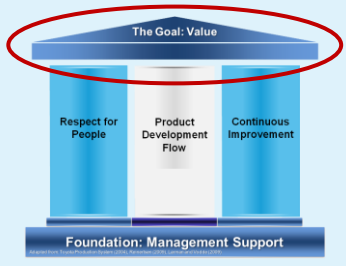
Scaled Agile Framework™

Lean Perspectives: House of Lean



Lean Perspectives: House of Lean

Lean Goal: Speed, Value, Quality



The Goal

- Sustainably shortest lead time
- Best quality and value (to people and society).
- Most customer delight, lowest cost, high morale, safety.

- *All we are doing is looking at the timeline, from the moment the customer gives us an order to the point where we collect the cash. And we are reducing the time line by reducing the non-value added wastes.*
  – Taiichi Ohno
- *We need to figure out a way to deliver software so fast that our customers don't have time to change their minds.*
  – Poppendieck
- *Focus on the baton, not the runners.*
  – Larman and Vodde
- Minimize delays, handoffs and non-value added activities

Lean Pillar 1: Respect for People

Summary

- Develop Teams, then build products
- Empower Teams to continuously improve
- Build Partnerships

- Your customer is whoever consumes your work
- Don't trouble your customers
- Don't force people to do wasteful work
- Don't overload them
- Don't make them wait
- Don't impose wishful thinking
- Equip them with problem-solving tools
- Form long-term relationships based on trust

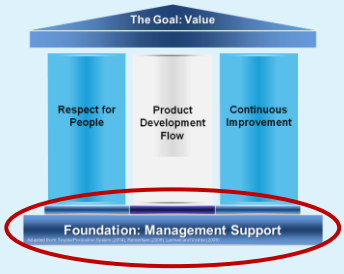
Lean Pillar 2: Continuous Improvement

Become relentless in:

- Reflection (hansei)
- Continuous improvement (kaizen)

- Consider all data then implement rapidly
- Protect the org knowledge base by developing stable personnel and careful succession systems
- Reflect at key milestones to identify and improve shortcomings
- Use tools like retrospectives, root cause analysis, and value stream mapping

Lean Foundation: Management Support

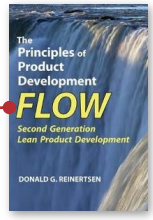


Develop Lean-Thinking, Kaizen management...

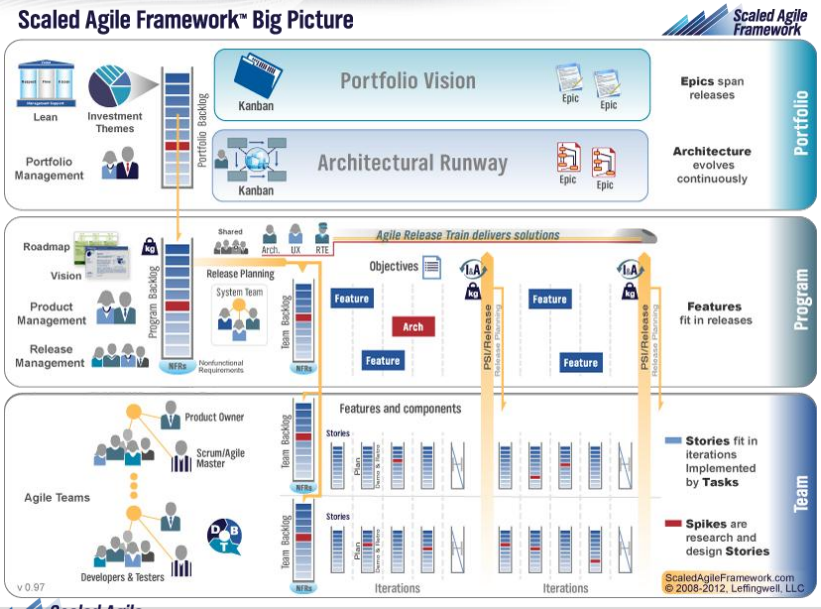
- Your job is to do the job AND improve the job...
- Endlessly!

- Be Lean-Thinking Manager-Teachers!
- Management is trained in lean thinking - bases decisions on this long term philosophy
- Management is trained in the practices and tools of continuous improvement
- Management leads, coaches and drive teams to continuously improve

Principles of Product Development Flow

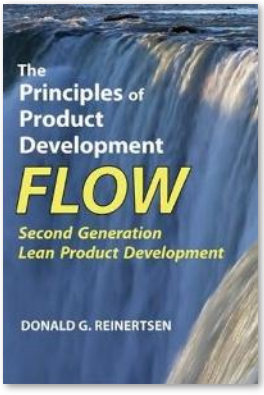


SAFe Implements Lean Principles Throughout



Principles of Product Development Flow

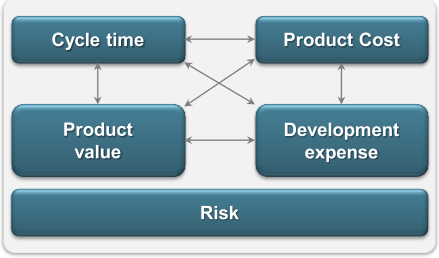
1. Take an economic view
2. Actively manage queues
3. Understand and exploit variability
4. Reduce batch sizes
5. Apply WIP constraints
6. Control flow under uncertainty: cadence and synchronization
7. Get feedback as fast as possible
8. Decentralize control



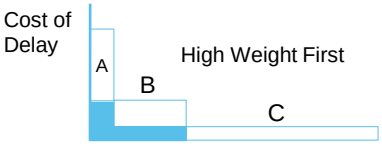
Reinertsen, Don. *Principles of Product Development Flow*, 2009.

#1 – Take an Economic View

Establish the economic framework for decision making

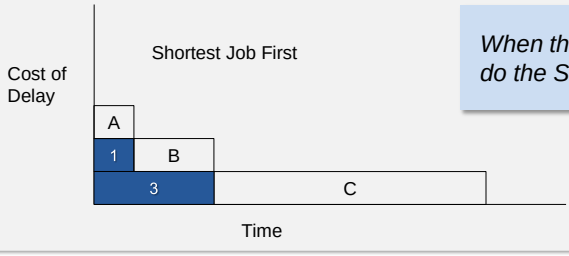


- Empower *local* decision making
- Do not consider money already spent
- Understand the full value chain
- Sequence high payoff, low cost activities first
- If you only quantify one thing, quantify the cost of delay

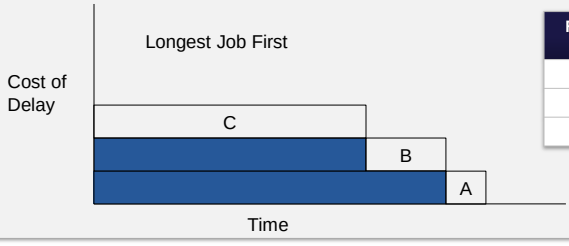


Reinertsen, Principles of Product Development Flow, 2009.

Prioritization: Shortest Job First



When the cost of delay is equal, do the Shortest Job First

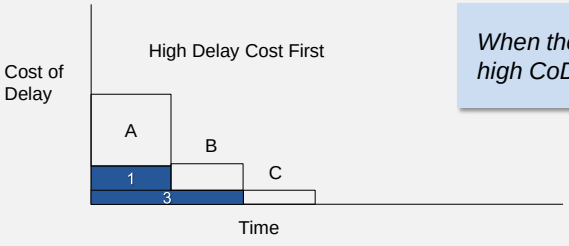


Feature	Time (Effort)	Cost of Delay
A	1	3
B	3	3
C	10	3

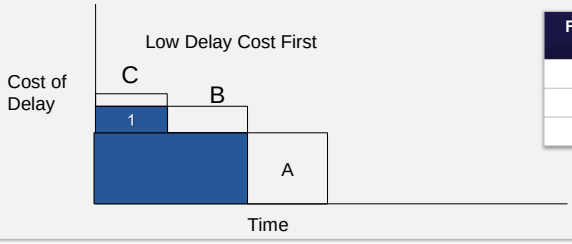
Delay Cost

From "The Principles of Product Development Flow," by Donald G. Reinertsen. Celeritas Publishing: 2009. Copyright 2009, Donald G. Reinertsen

Prioritization: High Delay Cost First



When the effort is equal, do the high CoD job first

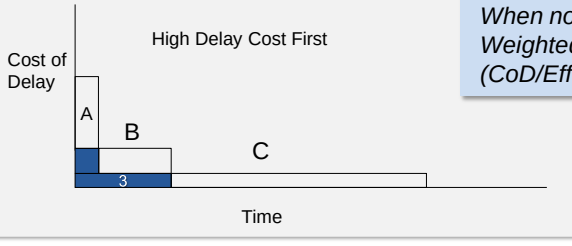


Feature	Time (Effort)	Cost of Delay
A	3	10
B	3	3
C	3	1

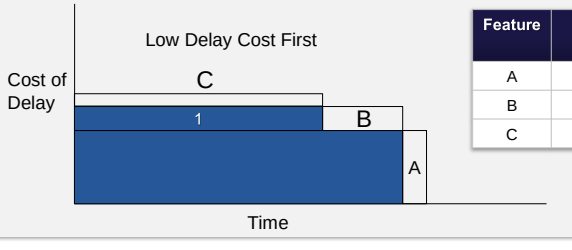
Delay Cost

From "The Principles of Product Development Flow," by Donald G. Reinertsen. Celeritas Publishing: 2009. Copyright 2009, Donald G. Reinertsen

CoD Economics: Weighted Shortest Job First



When nothing is equal, do the Weighted Shortest Job First! (CoD/Effort)



Feature	Time (Effort)	Cost of Delay	Weight= CoD/Effort
A	1	10	10
B	3	3	1
C	10	1	0.1

Delay Cost

From "The Principles of Product Development Flow," by Donald G. Reinertsen. Celeritas Publishing: 2009. Copyright 2009, Donald G. Reinertsen

WSJF Prioritization Matrix

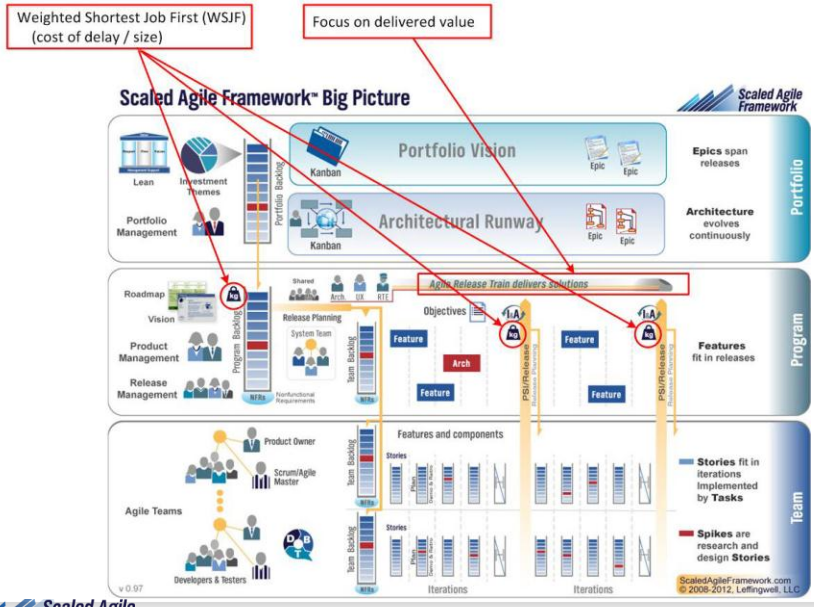
In general, do the Weighted Shortest Job First!
(Cost of Delay/Size)

$$\text{WSJF} = \frac{\text{User value + Time Value + RR|OE Value}}{\text{Job Size}}$$

Feature	User Value	Time Value	RR|OE Value	Job Size		WSJF

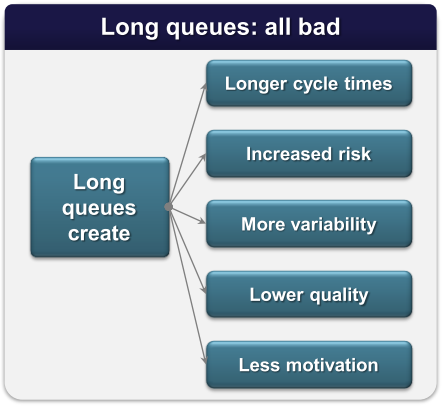
Scale for each parameter: 1, 2, 3, 5, 8,13, 20
Highest WSJF = highest priority

#1 – Take an Economic View



#2 – Actively Manage Queues

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…..



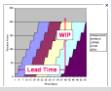
Reinertsen, Principles of Product Development Flow, 2009.

- Understand Little's Law

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

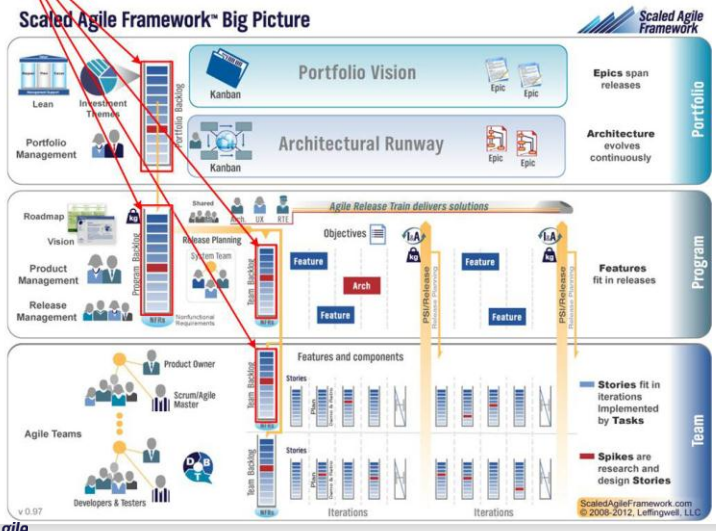
Wait time = Queue length/Processing rate

- Control queue sizes to control wait times
- Operating at high levels of utilization increases variability
- Measure cycle time and queue length with cumulative flow diagrams



#2 – Actively Manage Queues

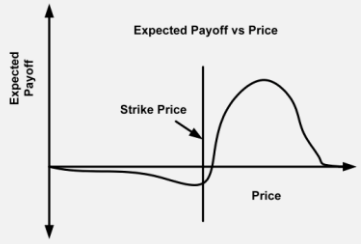
Short queues are good based on Little's law



#3 – Understand and Exploit Variability

Risk-taking is central to value creation in product development

Option theory: Why we can afford to take risks

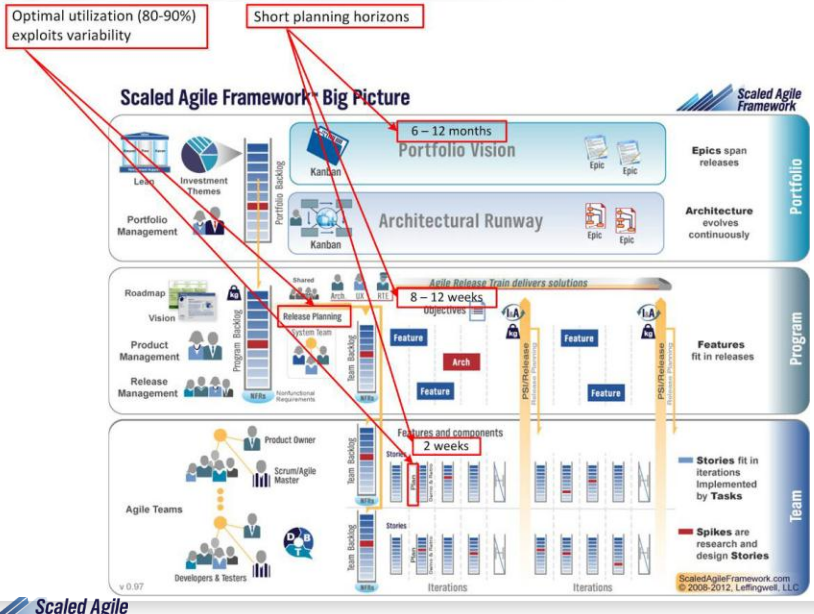


The expected payoff occurs because the left side has less area than the right side

- We cannot add value without adding variability
- Development variability can *increase economic value*
- Buffers trade money and time for variability reduction
  - Schedule buffers convert uncertain earliness to certain lateness
- Managing variability
  - Planning is exponentially easier in short-term horizons
  - Requirements are exponentially more difficult over long horizons

Reinertsen, Principles of Product *Development Flow, 2009.*

#3 – Understand and Exploit Variability

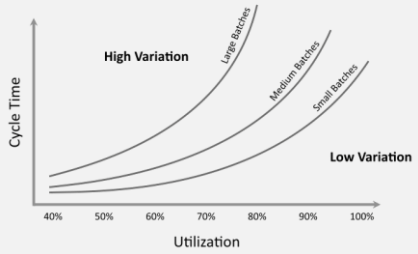


#4 – Reduce Batch Size

Small batches go through the system faster, with lower variability

Principles

- Large batch sizes increase variability
- High utilization increases variability
- Severe project slippage is the most likely result



Slippage in projects rises exponentially with duration

Fig. Source: [Poppendieck and Poppendieck 2007].

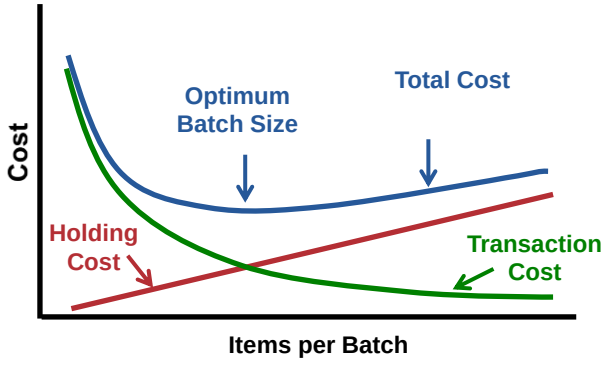
Cause and effect

- Reducing batch size
  - Reduces cycle time; faster feedback
  - Decreases variability and risk
- Most important batch is the transport (handoff) batch
- Proximity (co-location) enables small batch sizes
- Good infrastructure enables small batches (build and test automation, continuous integration, etc)
- Loose architectural coupling enables small batches

Reinertsen, Principles of Product Development Flow, 2009.

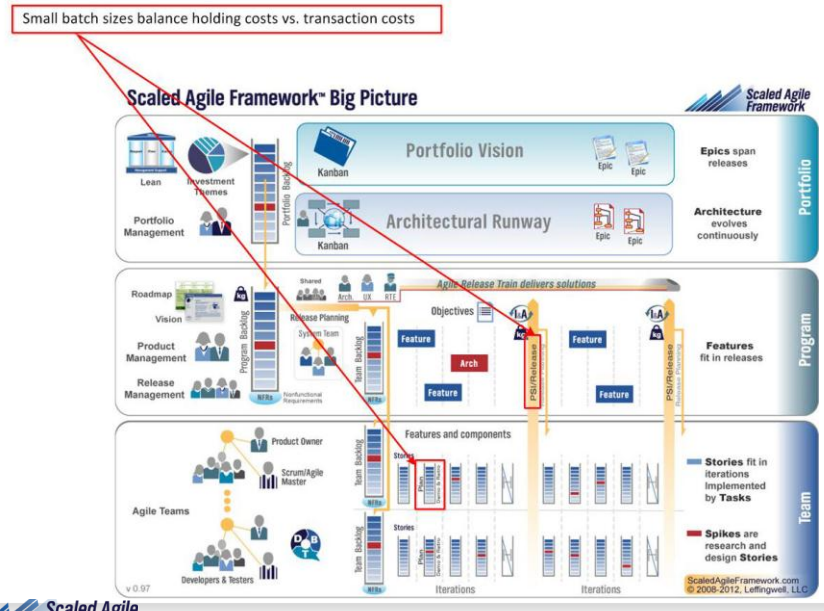
Finding Economic Batch Size

Optimum batch size is an example of a "U-Curve optimization"



- Higher transaction costs shift optimum batch size higher
- Higher holding costs shift it lower
- Batch size reduction probably saves 2X what you think!

#4 – Reduce Batch Sizes



#5 – Apply WIP Constraints

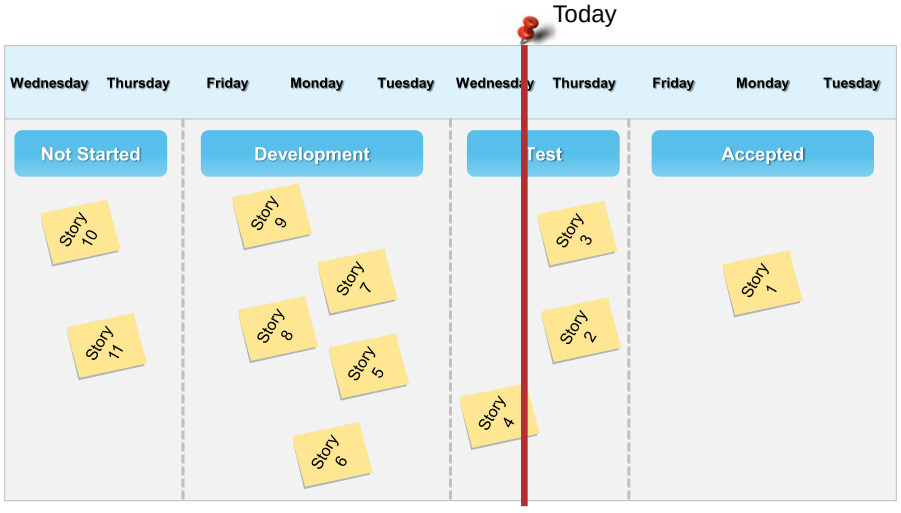
WIP Constraints force capacity matching and increase flow

Principles	Economic Benefits
Apply WIP constraints	Force capacity matching Accelerate delivery
Timebox	Prevent uncontrolled expansion of work Make waiting times predictable
Purge lower value projects when WIP is too high	Increase efficiency and throughput of remaining work
Constrain local WIP pools	Constrains global WIP pools
Make WIP continuously visible	1) Understand 2) take action

It is far easier to start a project than it is to finish one.

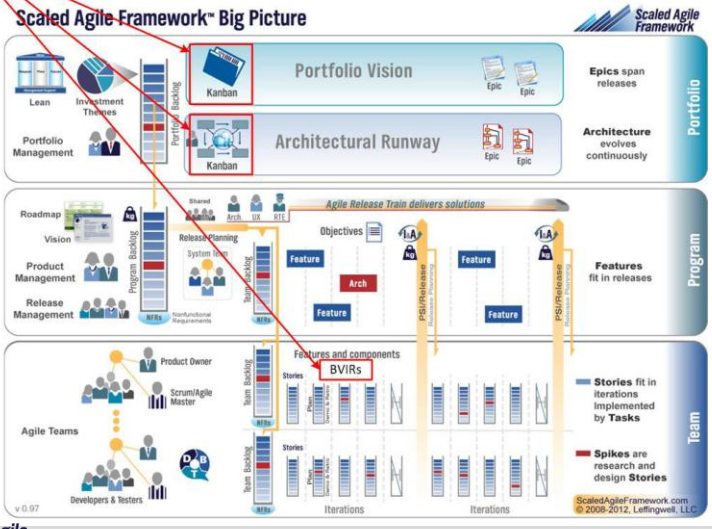
Reinertsen, Principles of Product Development Flow, 2009.

WIP Constraints with Kanban



#5 – Apply WIP Constraints

Constrain WIP and make it visible (Kanban – pull system)



#6 – Control Flow Under Uncertainty

Cadence

- Transforms unpredictable events into predictable events
- Requires scope or capacity margin
- Makes waiting times predictable
  - If you can't predict delivery, existing programs become "feature magnets"
- Manages load: When load and utilization become too high, you will see a sudden and catastrophic reduction in throughput!

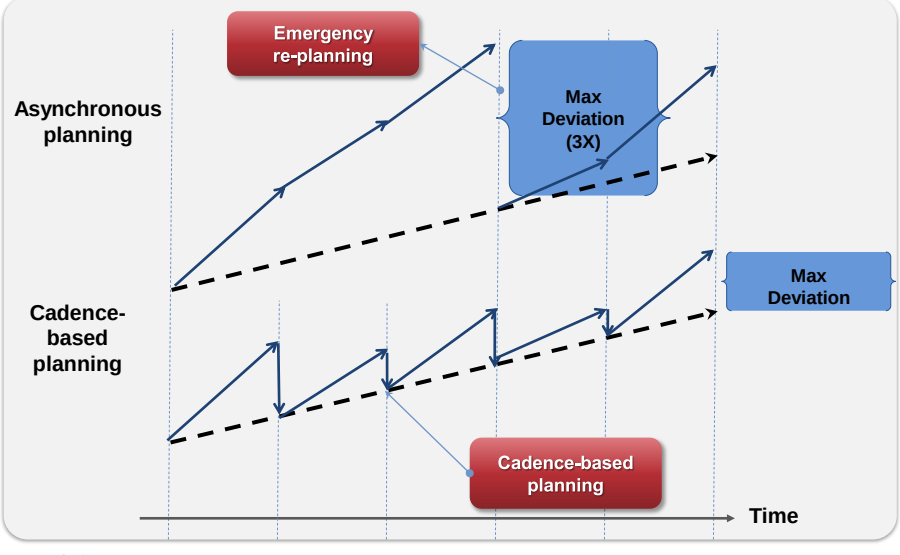
Synchronization

- Synchronization causes multiple events to happen at the same time
- Scheduled, periodic resynchronization limits variance to a single time interval
- Regular, system wide integration provides higher fidelity tests and objective assessment
- Synch events facilitate cross functional tradeoffs of resources, scope

Reinertsen, Principles of Product Development Flow, 2009.

Scaled Agile Framework

Control Variance with Planning Cadence



Scaled Agile Framework

Planning for Alignment

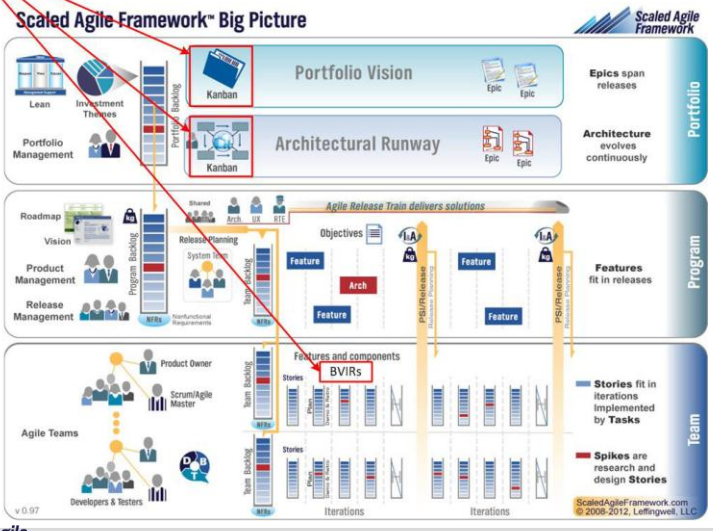
Principle: There is more value created with overall alignment than with local excellence

- Planning is done to create alignment, NOT to measure conformance
  - Align mass - concentrate force in one area
  - Economy of force – allocate minimum resources to other areas
- Principle of Mission: Specify the end state, its purpose, and minimum possible constraints
- Principle of Opportunism: Exploit unexpected opportunities by increasing deviation from plan when it makes economic sense

Reinertsen, Principles of Product Development Flow, 2009.

#6 – Control Flow Under Uncertainty

Constrain WIP and make it visible (Kanban – pull system)



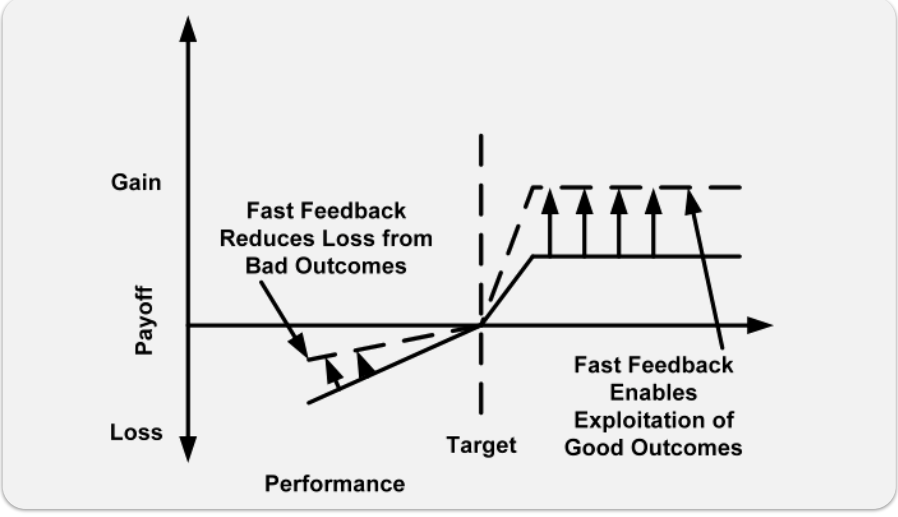
#7 – Get Feedback As Fast As Possible

Fast feedback manages risk and facilitates innovation

- Truncates unsuccessful paths quickly, reducing the cost of failure in risk taking
- Improves the efficiency of learning by reducing the time between cause and effect
- Facilitated by small batch sizes
- Requires increased investment in development environment to extract smaller signals
- Local feedback loops are inherently faster than global feedback loops; assists decision making

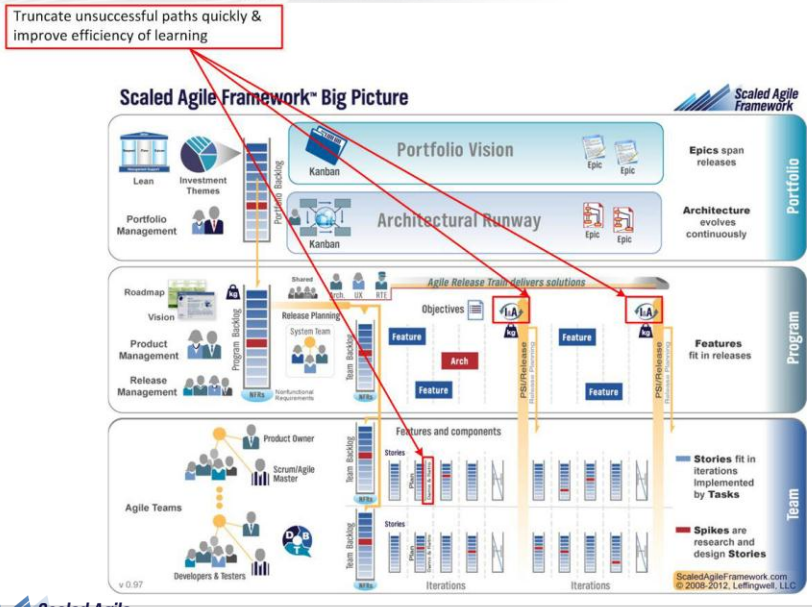
Reinertsen, Principles of Product Development Flow, 2009.

Fast Feedback Optimizes Payoff



Reinertsen, Principles of Product Development Flow, 2009.

#7 – Get Feedback As Fast As Possible



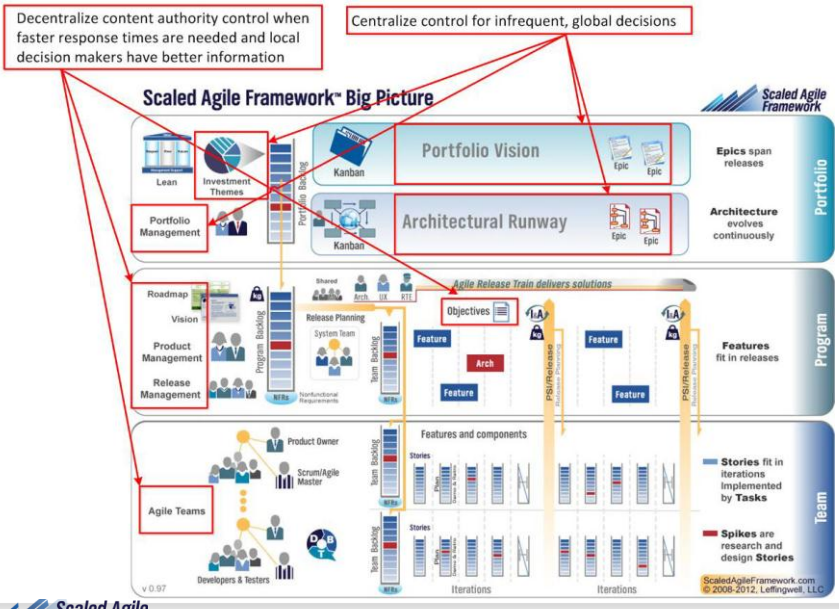
#8 – Decentralize Control

Optimize control for each level of decision making

- Centralize control for decisions that
  - Are infrequent
  - Can be applied globally; have significant economies of scale
  - Age well
- Decentralize control for all others
  - Inefficiency of decentralization costs less than the value of faster response time
  - Local decisions have better local information
- Control the economic logic behind a decision, not the entire decision
  - Set the framework, empower others to make the decisions

Reinertsen, *Principles of Product Development Flow, 2009.*

#8 – Decentralize Control



Summary

SAFe is fundamentally based on Lean principles

- Supports House of Lean culture
  - Value
  - Respect for people
  - Continuous improvement
  - Management support
- Implements Reinertsen's eight principles of product development flow
  - Take an economic view
  - Actively manage queues
  - Understand and exploit variability
  - Reduce batch sizes
  - Apply WIP constraints
  - Control flow under uncertainty: cadence and synchronization
  - Get feedback as fast as possible
  - Decentralize control

Reinertsen, *Principles of Product Development Flow, 2009.*