The Continuous Engineering Experts

INCOSE Healthcare Working Group

4th Annual Systems Engineering in Healthcare Conference

Scaling Agile to the Enterprise – Pitfalls and Practices addressed with the Scaled Agile Framework

Doug Stewart

Director of Services

321Gang

Overview

- Agile principles are the foundation
- Team-based agile processes solve part of the problem
- Enterprises add scaling practices
  - Many patterns harvested and included in SAFe

Pain Points of Scaling Agile

- The Team of Teams becomes too large a team
- New corporate initiatives are large and may take a long time to complete
- Meeting regulatory compliance
- Emergent design does not provide for a stable architecture
- Agile is restricted to IT
- Can not set and meet expectations with Stakeholders and Customers
- No clear way to handle highly specialized roles
- Subcontractors and Vendors

Lean Agile Principles

Why the focus on principles?

A common disease that afflicts management the world over is the impression that "Our problems are different". They are different to be sure, but the principles that will help to improve quality of product and service are universal in nature. *—W. Edwards Deming*

100 books

Principles over Practices

100s implementations

- A Lean-Agile transformation will deliver substantial benefits
- But it is a significant change and every implementation is different
- Leaders should understand why the practices work; it's part of "knowing what it is they must do"
- If a practice needs to change, understanding the principles will assure the change moves the enterprise in the right direction

Apply 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

Team of Teams becomes too large a team

- The problem: Scrum of Scrums becomes too large
  - 200 agile teams means 200 Scrum Managers to sync
  - 7 ± 2 is an ideal team size, for Scrum of Scrums as well

How Do Scrum and Agile Help Teams?

Fosters decentralized decisions

Better aligned to deliver value faster

Reduced WIP drives focus

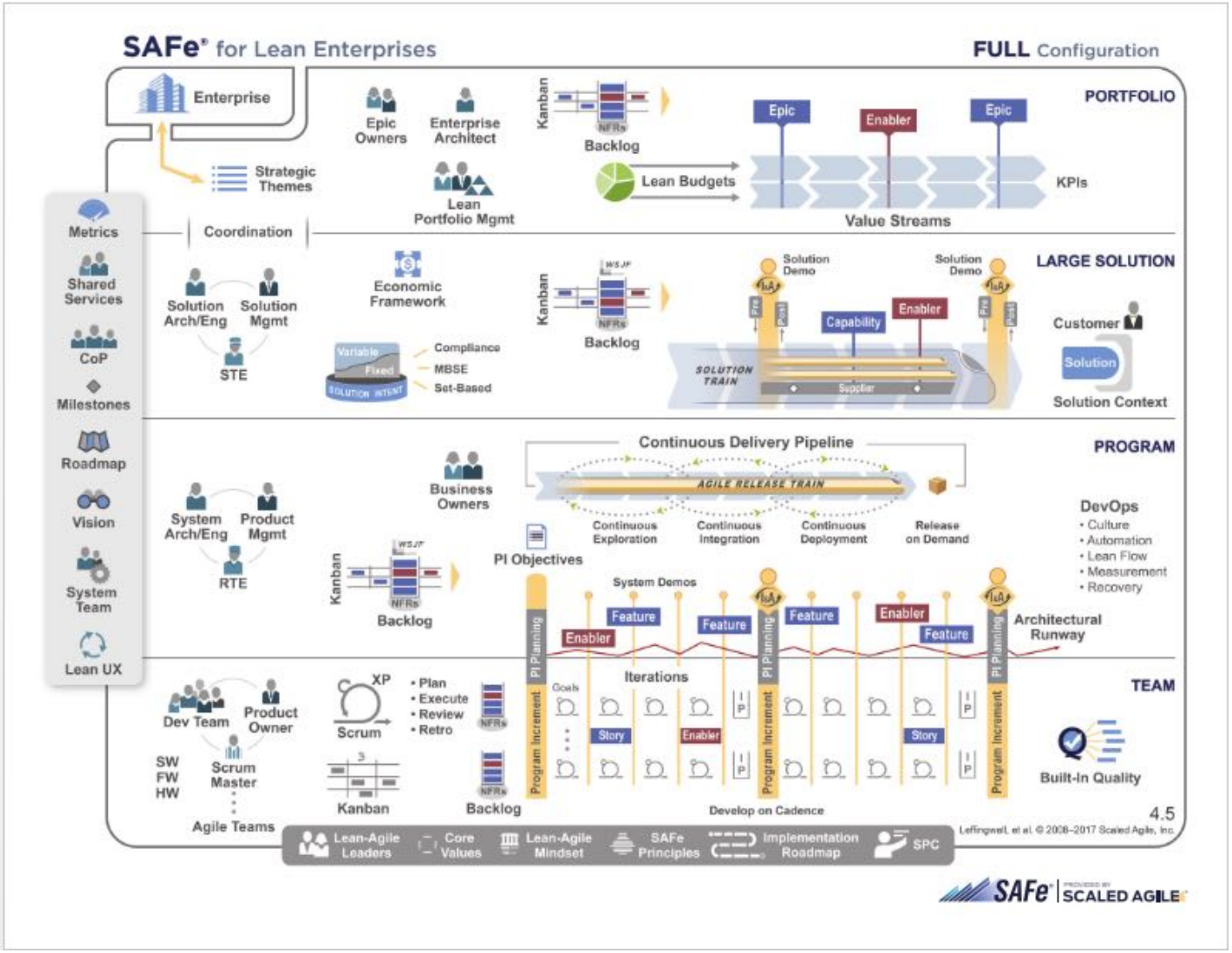
Synchronized around common ceremonies

Smaller batches move through system faster

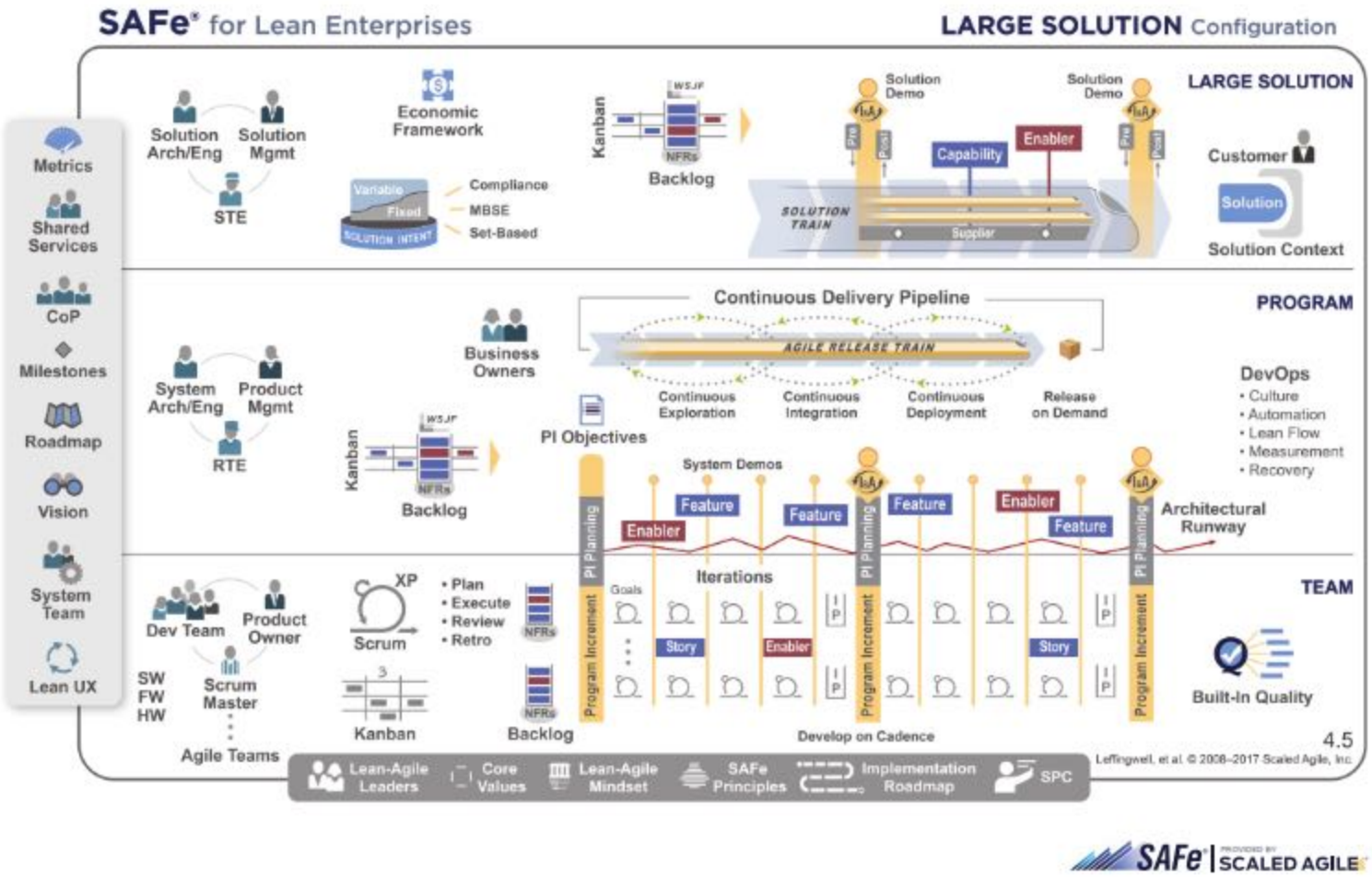
Regular cadence limits variability to single Sprint

Incremental development provides fast feedback

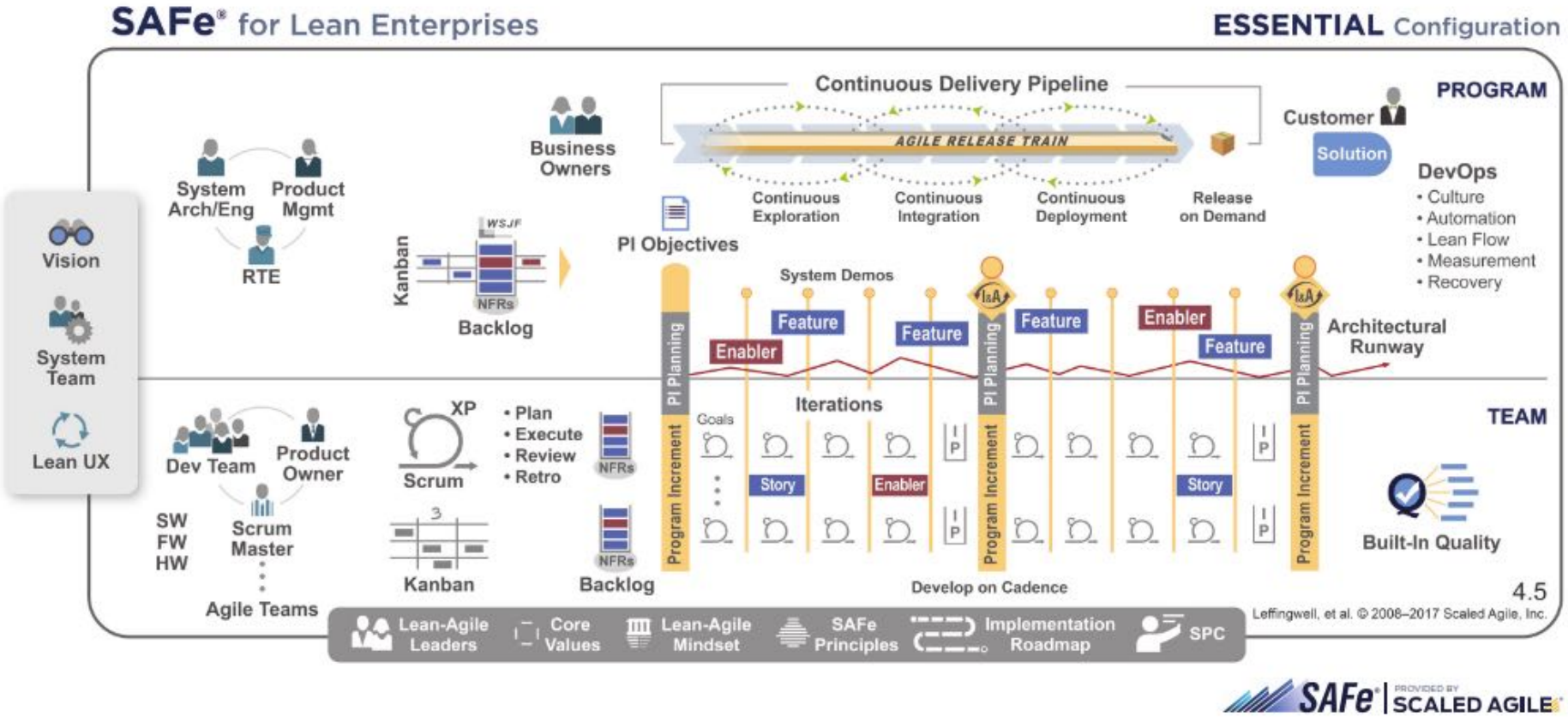
SAFe Provides a Structure to Scale Teams



Large Solution SAFe



Essential SAFe

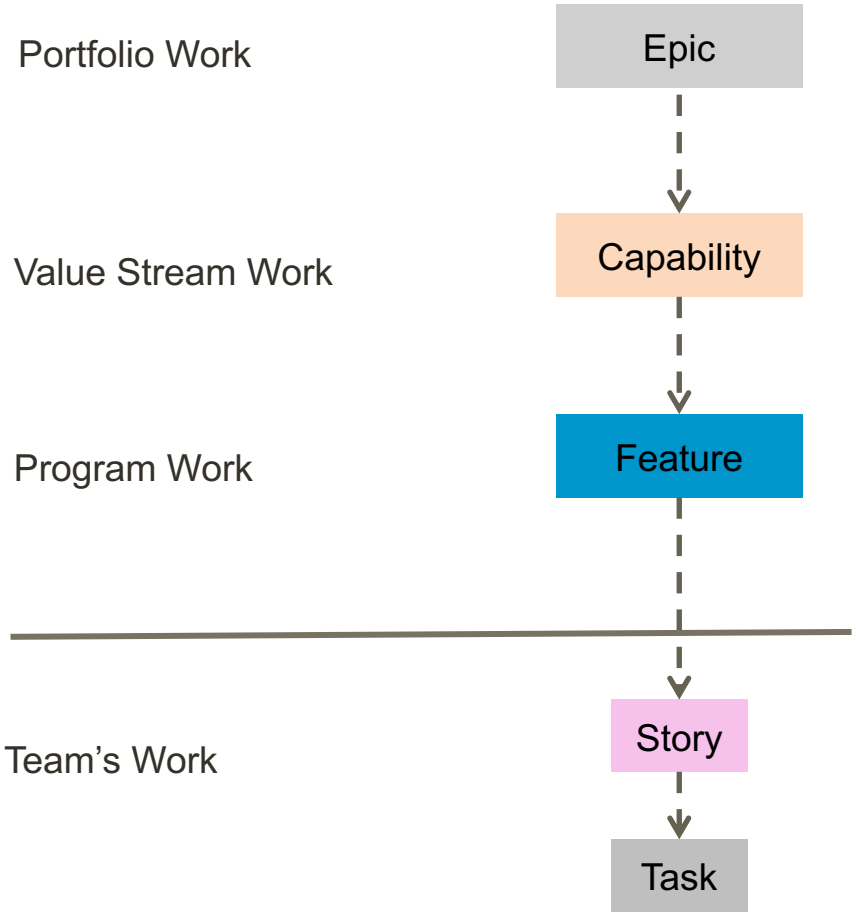


SAFe® PROVIDED BY SCALED AGILE

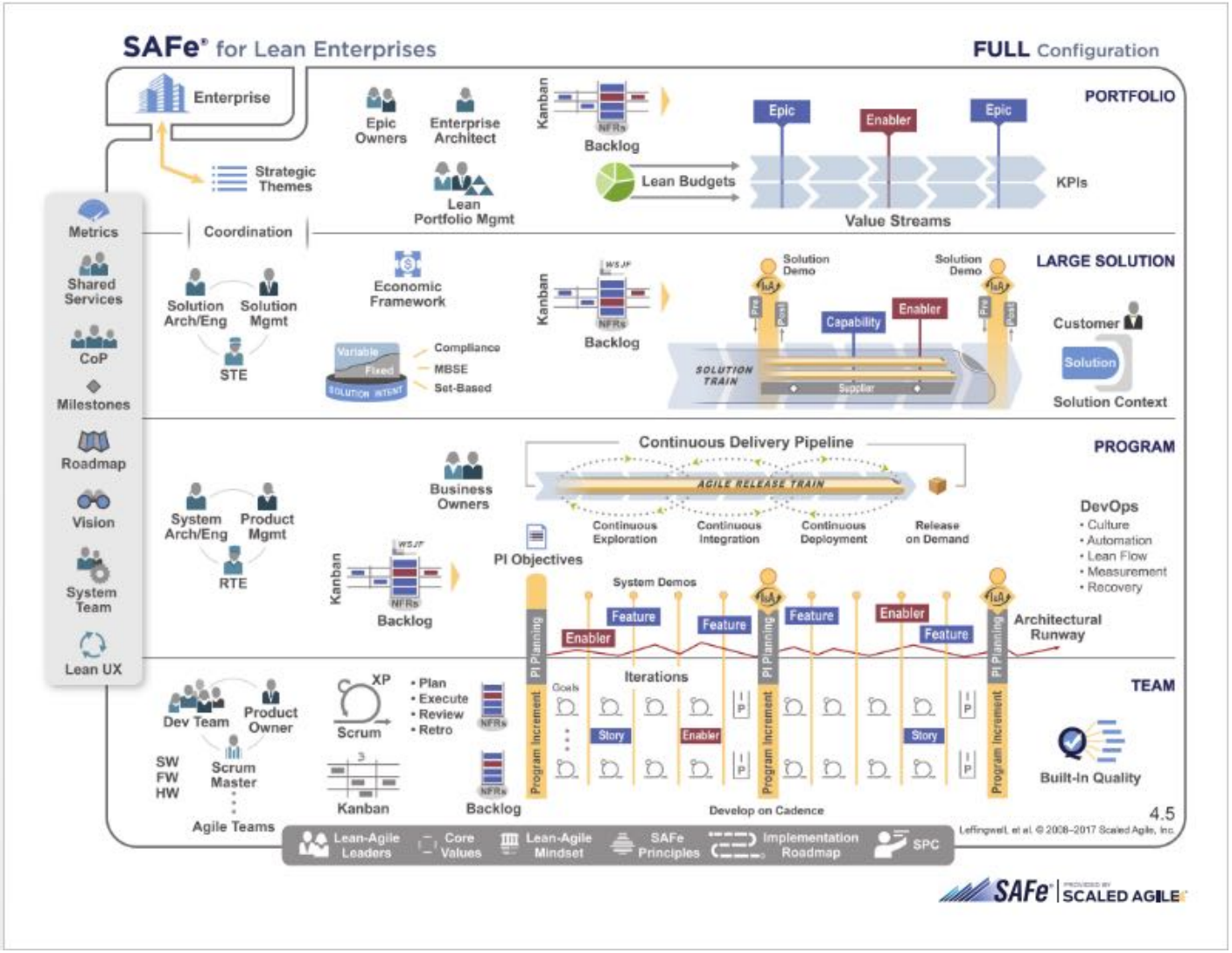
New corporate initiatives are large and may take a long time to complete

- The problem: It may take a year or longer to implement a large initiative
  - May cross product lines
  - May involve vendors
  - In traditional agile team-based practice, epics split into stories and big stories split into smaller stories. Once stories are split, the original story is no longer relevant
  - Can not track where we are
  - Difficult to explore multiple options in set-based designs

Structure for Evolving Solution Intent



SAFe Provides a Structure to Scale Requirements



Meeting regulatory compliance

- The problem: Being agile does not remove any compliance obligations
  - Still must prove we are meeting the regulations
  - Compliance traditionally measured with waterfall artifact inspections

Compliance meets lean-agile

Regulatory/Compliance

ensure health and welfare

- Quality, safety, security, etc.
- Verification and validation
- Inspections, audits, sign-off
- Metrics – defects, req coverage, code coverage, etc.
- Agile metrics - % tests automated, # new tests, # refactors

Lean-agile

shortest sustainable lead-time

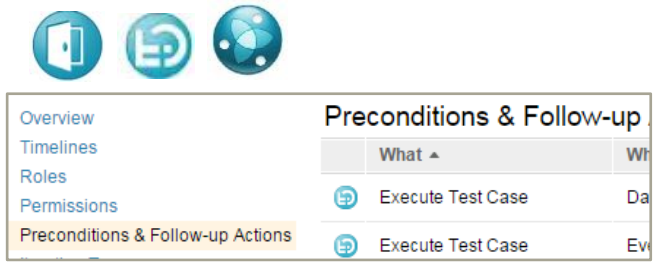
- Organize around value
- Apply cadence and synchronization
- Build quality in
- Deliver business value earlier and continuously
- Make work and progress visible

Define scalable Definition of Done (DoD)

- Testing and compliance are continuous, part of process

Story (within Sprint)	Feature (within Increment)	Increment/Release
• Acceptance passed (auto) • DBT tests passed (auto) • Standards met (SW, HW) • Peer reviews (SW, HW) • Accepted by PO	• Deployed to V&V/QA for end-end system testing • QA/Customer/etc. review • Included in build/deploy process • Documentation updated • Accepted by Prod Mgr	• Deployed to IV&V testing • Customer signoff • Assess compliance status via report generation (RPE)

- Enforce DoD with process actions



Ensure unit test pass before delivering code

Ensure reviewer viewed all changes

Cannot deliver code until peer review

Workspace must be caught up before delivery

Cannot change work item state after delivery to test

Only Quality role can set a Feature to done

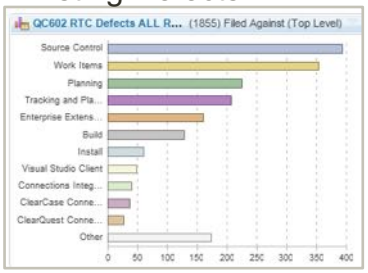
E-Sig required for sign-off

Cannot accept Story until all issues from review are resolved

Measure progress towards compliance

- Architect CE products with the end in mind – compliance
- Define and enforce overall schema (dashboard w/ anti-queries)
- Ensure signatures where necessary
- Automate – coverage, quality, etc.

Existing Defects



Defect Fix Rate

RTC 6.0.2 Feature Defects

Show Parameters

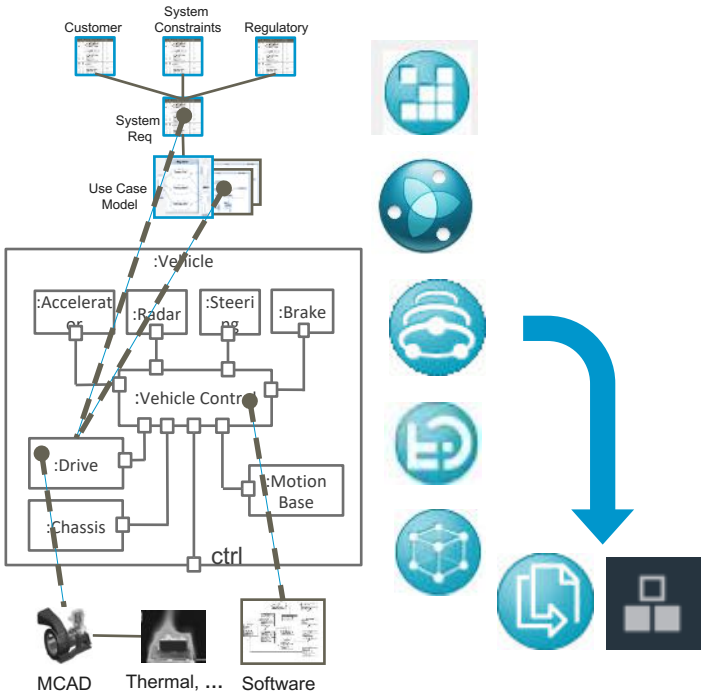
Description	Found	Fixed	Fix Rate
Build: 376391: 6.0.2 Buil	0	0	NaN
dotNet: 376208: [CCM] 6	68	51	75.0%
EE: Ant invoked by Trans	14	14	100.0%
EE: Build log improveme	9	9	100.0%
EE: Multi-threading with I	2	1	50.0%
EE: RDz integration impr	0	0	NaN
EE: test automation fram	0	0	NaN
EE: other	121	106	87.6%
TAP: All 6.0.2 Plan Items	137	106	77.4%
376322: [SCM] Code Re	47	25	53.2%
286826: [SCM] Compone	128	113	88.3%
Total	526	425	80.8%

Feb 19, 2016 10:12 AM

Requirements Coverage

System Requirement	System Requirement	Software Requirement	Software Requirement	Test Case ID	Test Case
178	Donor Dividend Allocation	188	Donation by Amount	13	Donation amount limit
		319	Dividend allocation by percent	18	Dividend Allocation by
				19	Allocate Dividends to
		74	Frequency of dividend transfer	1	Verify dividend transfe
181	Organizations can apply	238	Requests sent in form of hard	4	Process hard copy re

Compliance Documents



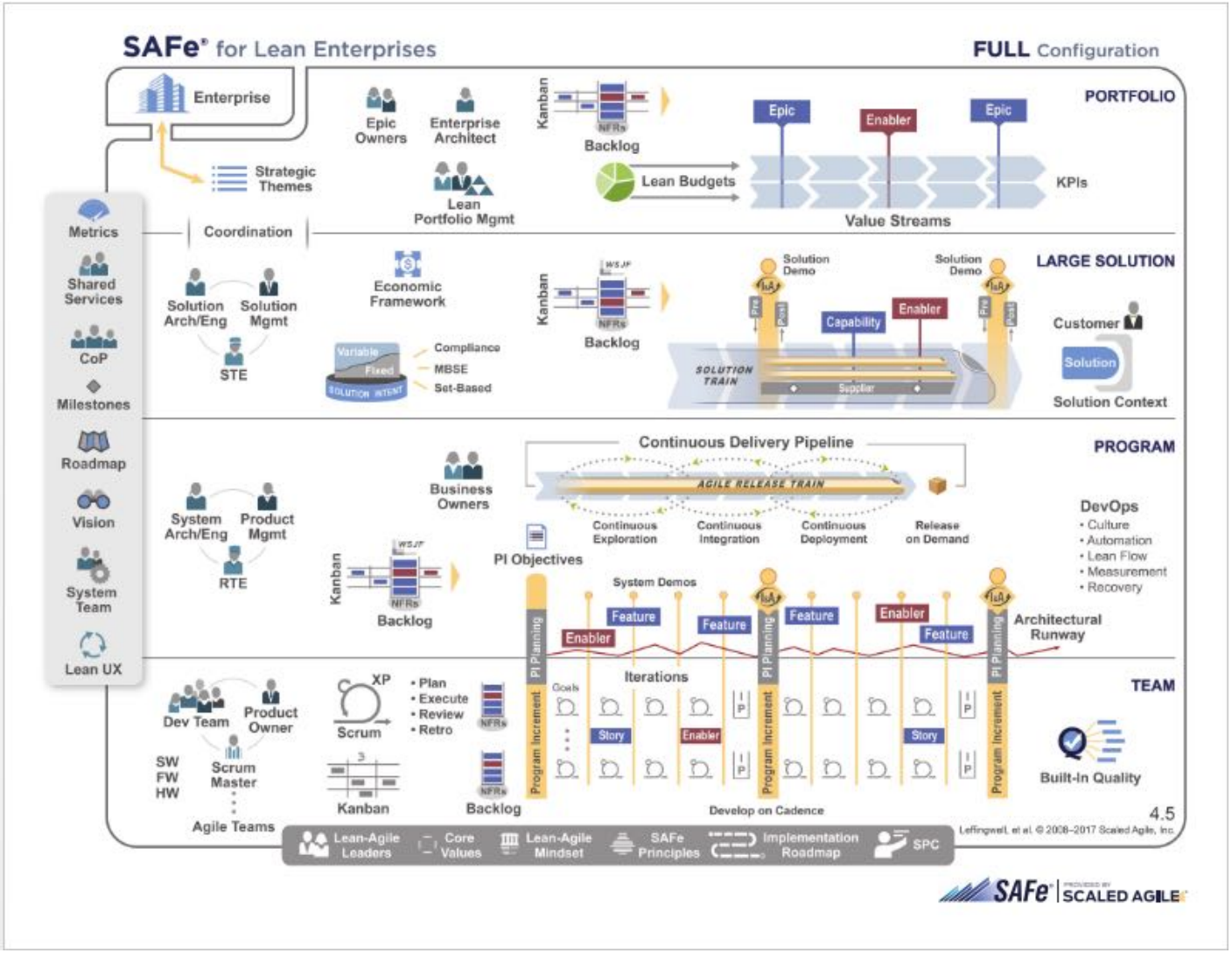
Continuous review and approval process

- Reviews are part of the process (Definition of Done)
- Strive for lean process – attack delay, waste, WIP

Emergent design does not guarantee a stable architecture

- The problem: Emergent design becomes brittle architecture without
  - a well communicated vision
  - A common structure to emerge in to

Architectural Runway Provides Structure for Emergent Design

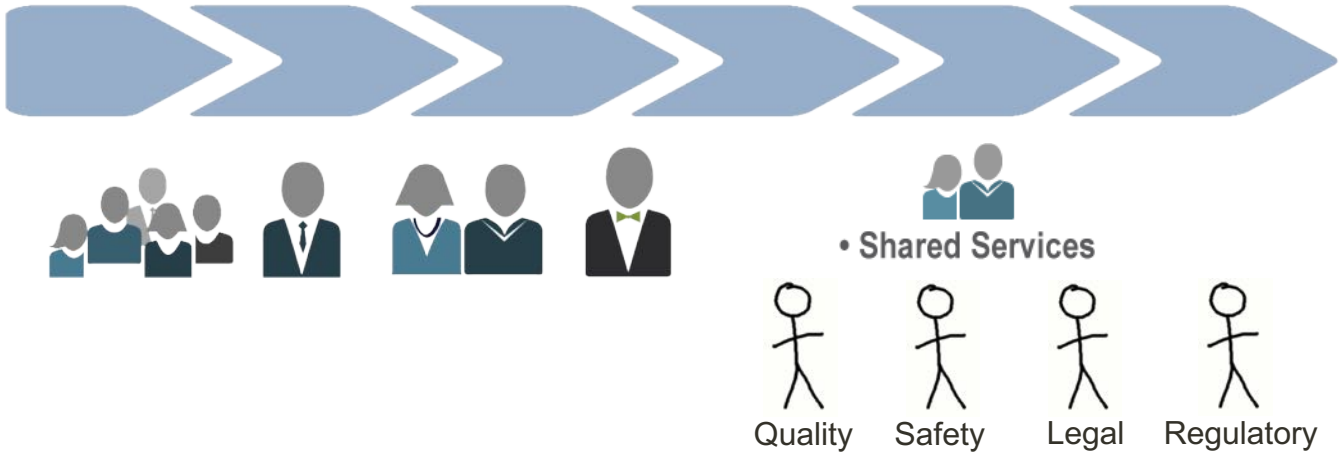


Agile is Restricted to IT and Engineering

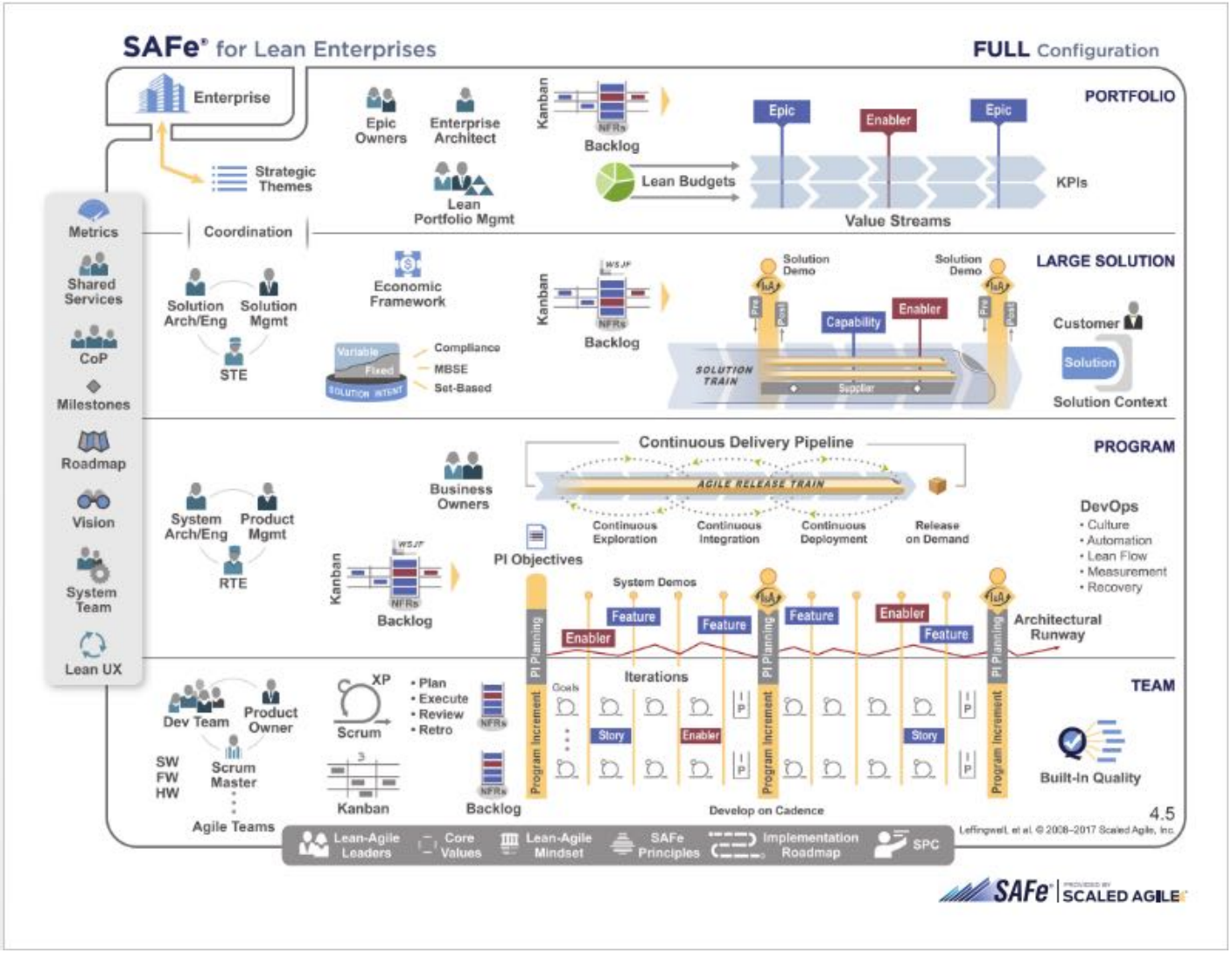
- The problem: Agile is seen as an IT or Engineering practice
  - Executives are learning Lean
  - Testing, Compliance, Business Controls, Finance think Waterfall

Ensure EVERYONE is on the value stream

- Goal is reduced waste – waiting, delays, hand offs, batch sizes, WIP
- SAFe identifies stakeholders, customer, suppliers
- But, also includes anyone who reviews, approves, signs-off
  - QA, safety, customer, regulatory, legal, procurement, etc.



A Place for Everyone in SAFe



Can not set and meet expectations with Stakeholders and Customers

- The problem: Businesses need commitments in order to operate
  - It is not enough to work in a priority backlog and to deliver capability as soon as possible, though it is essential
  - Product sales, market opportunities, contract funding require time-based commitments

1) Organize around value

- Optimize around value delivery
- Deliver quickly by minimizing handoffs, delays, waiting, and eliminating waste
- Facilitate early and incremental verification and validation, reviews, sign-off, etc.

> *"There is nothing so useless as doing efficiently that which should not be done at all."*
>
> —Peter F. Drucker

2) Apply cadence and synchronization

- Align and focus on short term vision
- Make unpredictable events predictable
- Reduce delay and batch size for compliance activities
- Deliver value sooner; perform quality activities continually, not at end

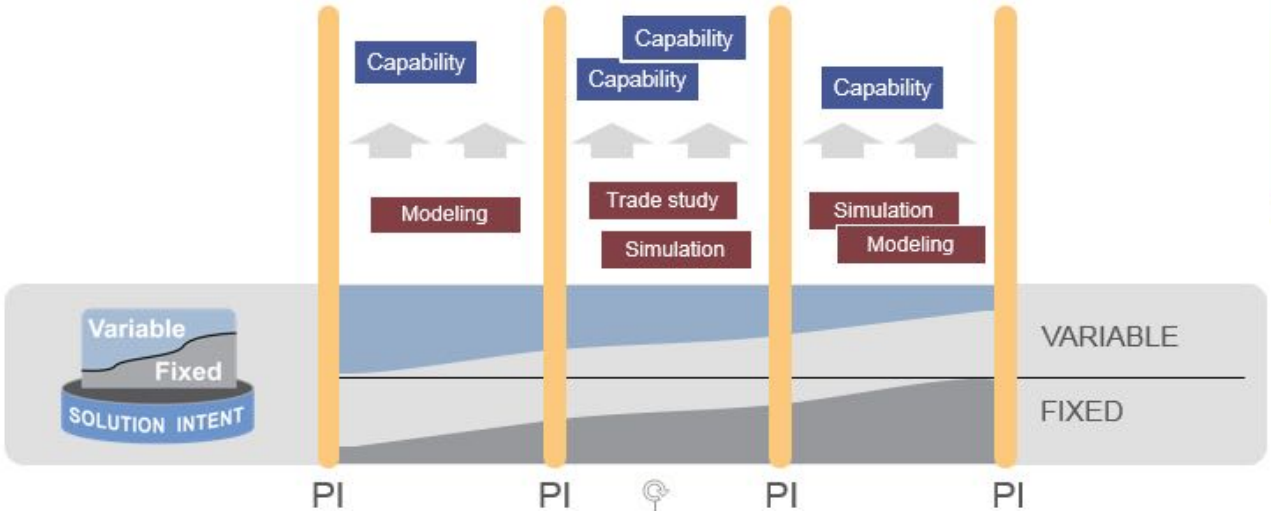
> *"Cadence and synchronization limit the accumulation of variance"*
> *—Don Reinertsen*

4) Deliver business value earlier and continuously

- Exploration is continuous, not all up front
- Faster feedback on req and des decisions
- Everyone contributes to learning, not "leads"
- Specifications complete at end; evolve with learning

"The more detailed we made our plans, the longer our cycle times became"
—Don Reinertsen,

"The best architectures, requirements, and designs emerge"
-- Agile Manifesto



5) Make Work and Progress Visible

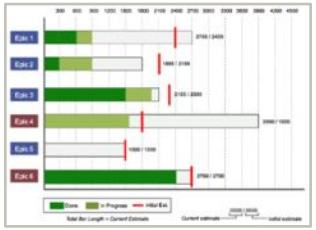
- Quality and compliance are continuous processes
  - Visualize progress towards completion
  - Generate compliance reports (don't bolt on at end)
- Encourages quality behavior and (eventually) culture
- Increase trust with business, customer, regulatory
- Do **NOT** plan to maximize utilization

> *"All is visible. All is known."*
> *— From SAFe Core Values*

Measure progress at each System Demo

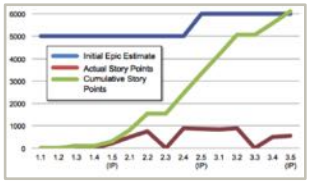
- Towards release, improved process, and compliance

Epic Progress

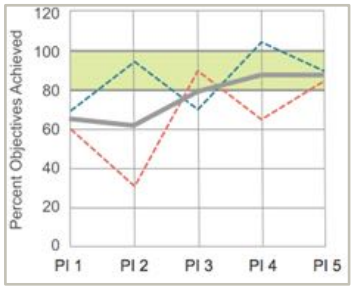


Progress towards release

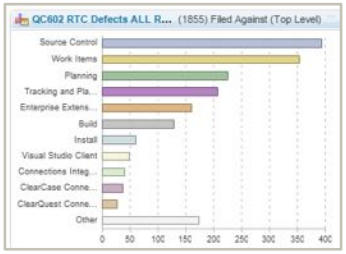
Epic Burnup



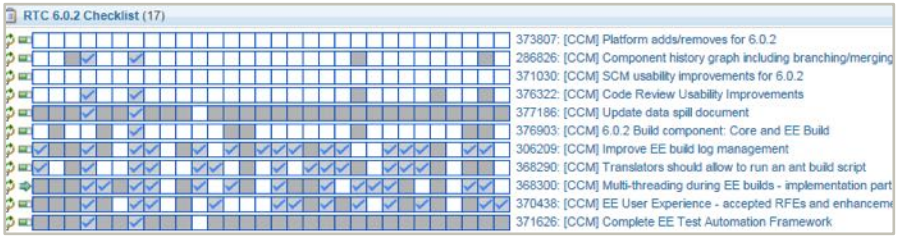
PI Predictability



Defects

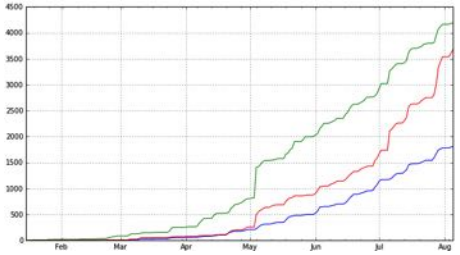


Team's Feature DoD



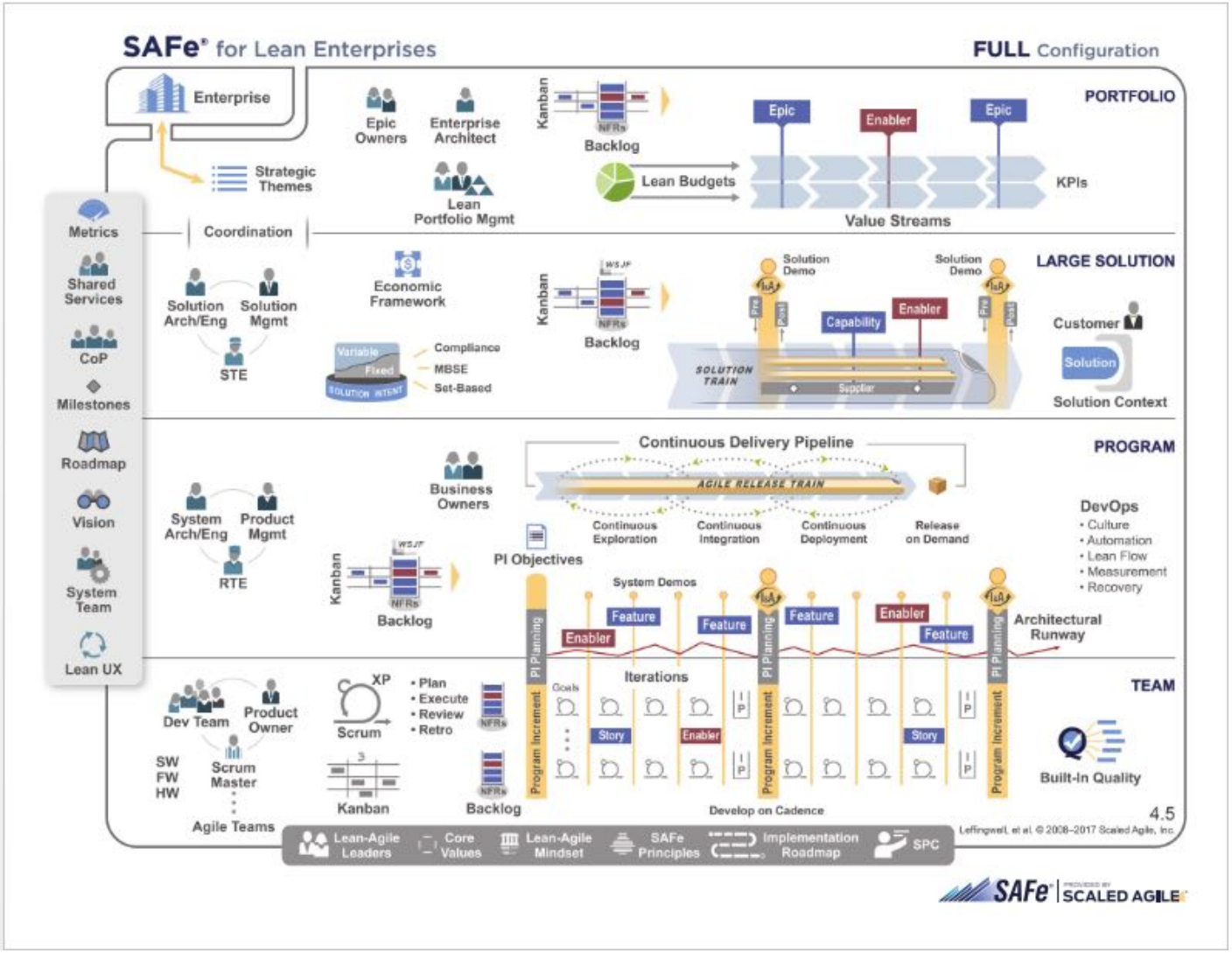
Progress following process (WIP constraints)

Cumulative Flow



Kanban Board

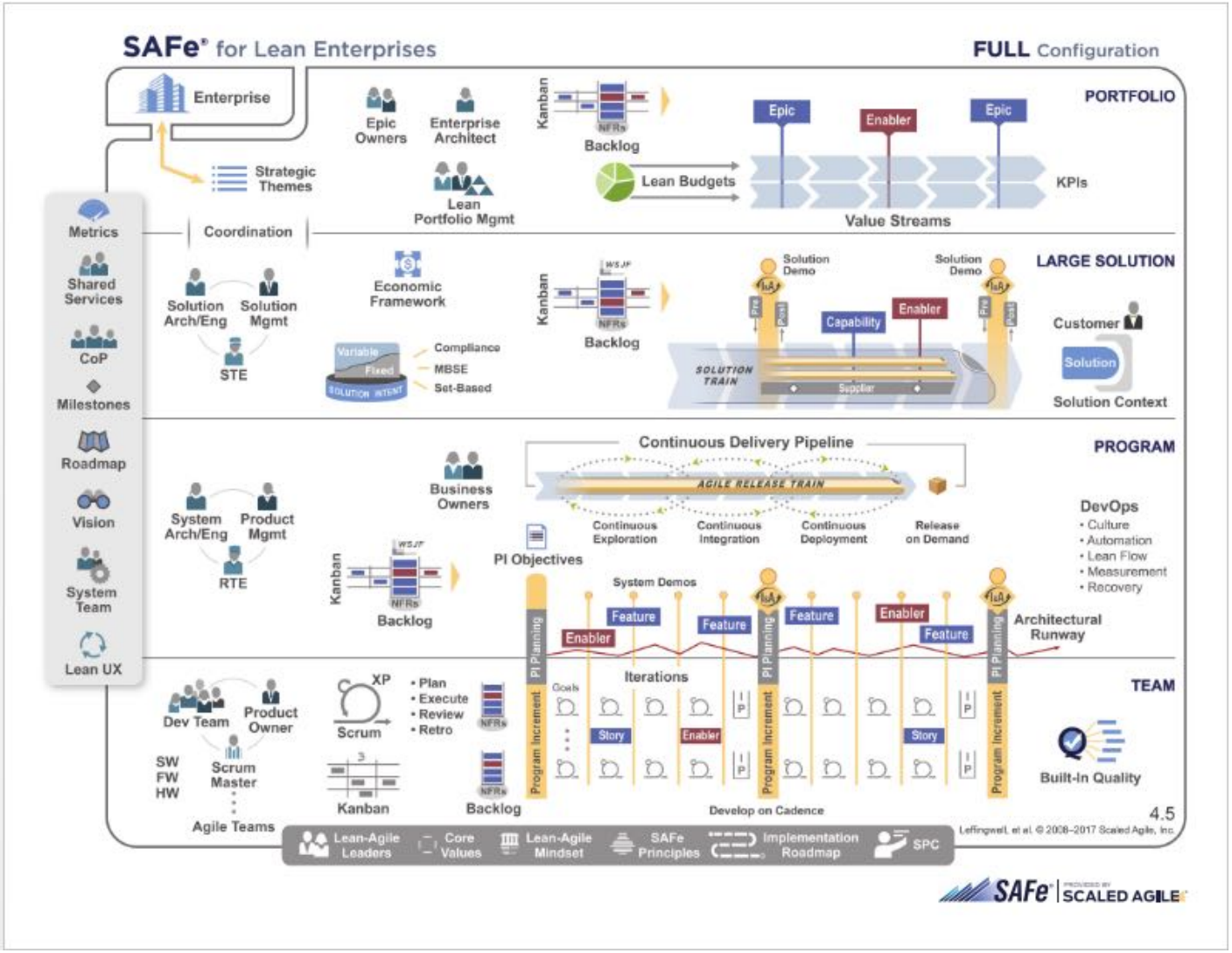
Release Any Time



No clear way to handle highly specialized roles

- The problem: We often have specialized teams that do not fit well in a team-based agile cadence
  - System Team
  - DevOps
  - Formal Acceptance Testing
  - Release Management
  - User Experience
  - Database
  - Shared Services

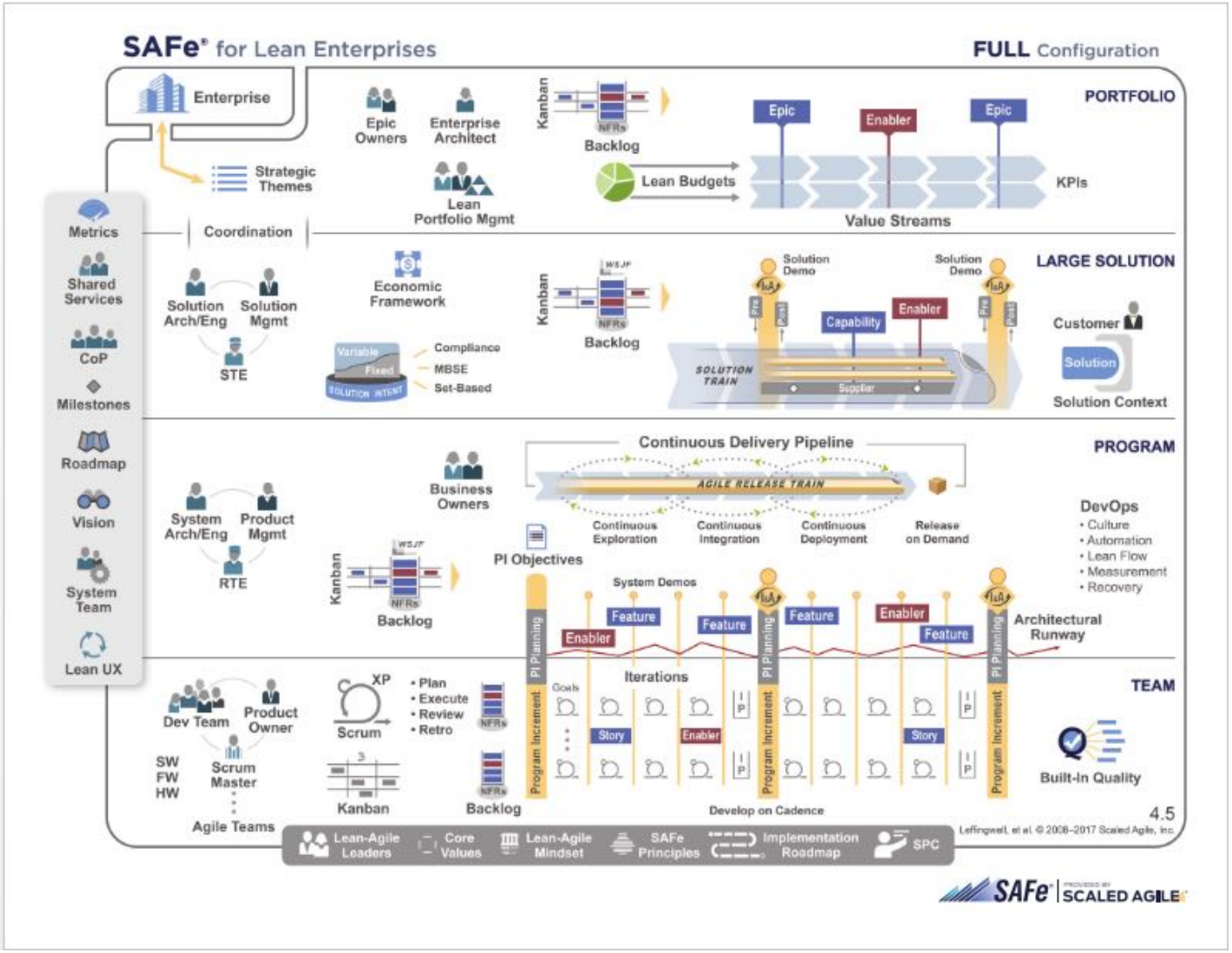
Specialized Roles



Subcontractors and Vendors

- The problem: Another group may be delivering part of the product being developed
  - Subcontracted company
  - Vendor packaged solution customization
  - Manufacturer, such as for hardware

Subcontractor Release Trains



Doug Stewart

Director of Services

doug@321gang.com

#hwgsec @321gang

INCOSE Healthcare Working Group

4th Annual Systems Engineering in Healthcare Conference

321 Gang — The Continuous Engineering Experts