Essential SAFe®

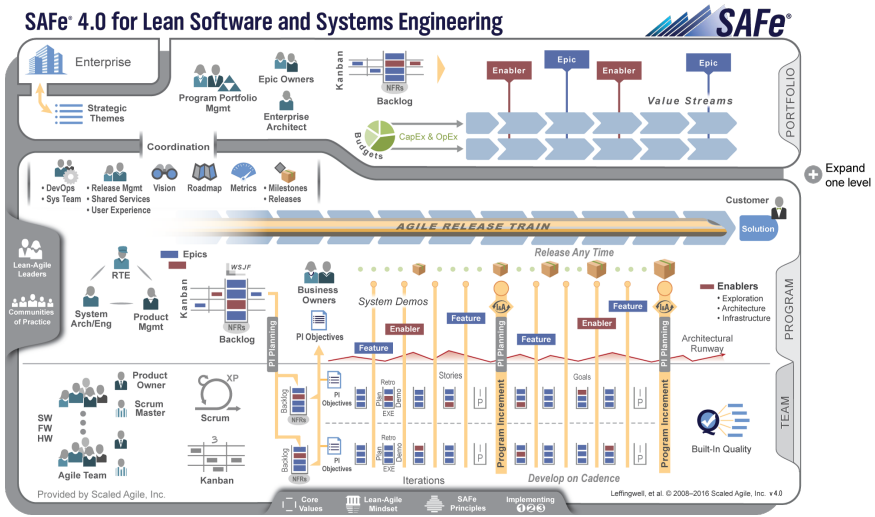
By Dean Leffingwell

Milwaukee Scaled Agile Meetup
October 18, 2016

SCALED AGILE®

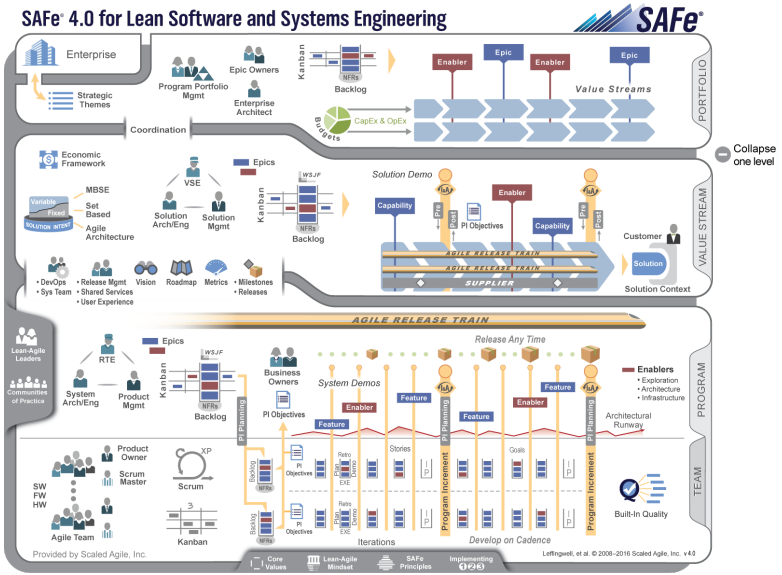
SAFe has grown to support larger enterprises

Three-level SAFe



Well suited for small to medium sized, independent ARTs and Solutions

Four-level SAFe



Designed for the largest, most complex solutions

For some, it's too much to begin with

As a result, some are skipping critical elements

Findings from the field

Successful rollout, but still struggling, root causes

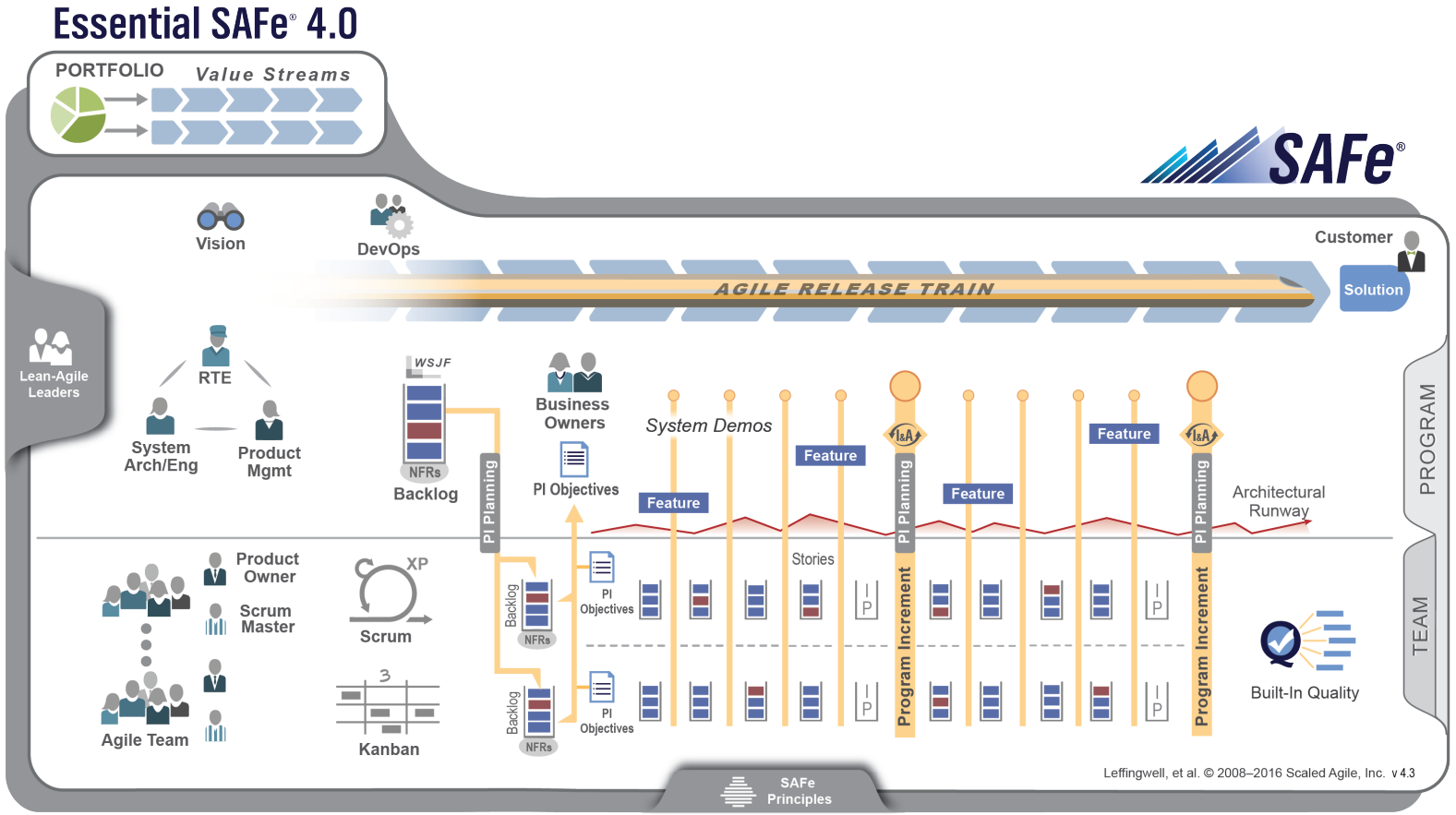
- Not doing Inspect and Adapt
- No Innovation and Planning iteration
- Individual ART teams were not themselves, cross-functional
- No system demo

Heard in the field

- “ SAFe is a flexible framework, we’ve adopted what we liked, but we don’t use Agile Release Trains”
- “SAFe is flexible, were adopting it, but we’ve decided not to affect the way the teams are working, so we didn’t include the teams in training”
- “Our leaders don’t have time for training”

What is Essential in SAFe?

The Essential SAFe Little-Big-Picture



Ten elements of Essential SAFe

#1 – Lean-Agile Principles

#2 – Agile Teams and Release Trains

#3 – Cadence and Synchronization

#4 – Critical Team and Program Roles

#5 – PI Planning

#6 – System Demo

#7 – Inspect and Adapt

#8 – IP Iteration

#9 – Architectural Runway

#10 – Lean-Agile Leaders

#1 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

Principles over Practices

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

100 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

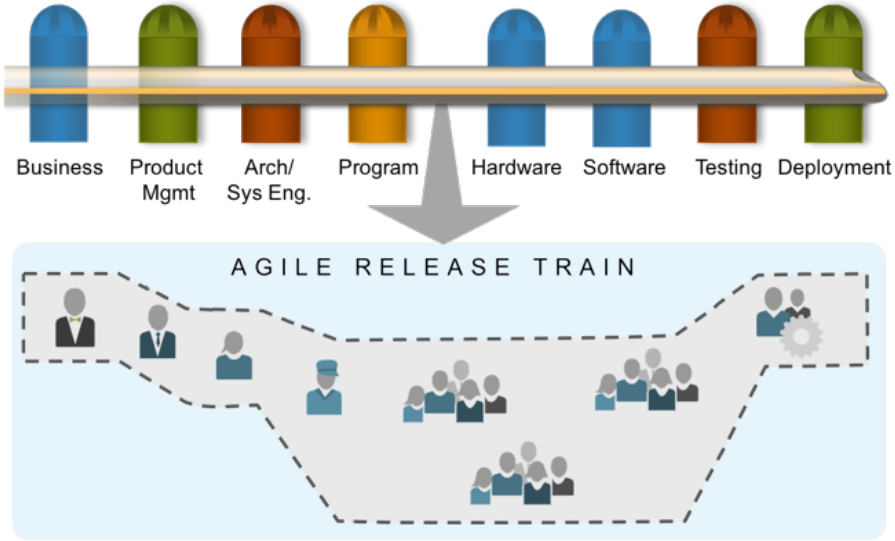
Symptoms of unprincipled development

- Every portfolio, value stream and program is different, but change agents, management and other practitioners are unable to adapt practices
- Improvement of business outcomes over time is impaired – practices and measures that used to be beneficial become constraints
- Lean-Agile mindset is unachievable; Agile practices deployed on a wrong-minded “mental platform” produces serious problems
- Misalignment, conflict and disagreement on processes and practices are impossible to resolve

#2 Agile Teams and Release Trains

Agile Release Trains are cross functional teams of Agile Teams and other stakeholders that deliver solutions. They have all the people they need to define and deliver value every two weeks.

Agile Team

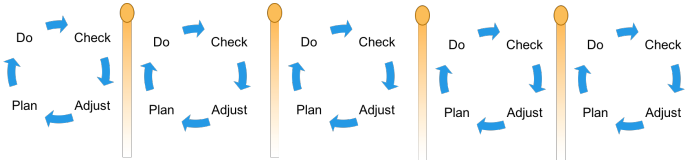


<u>Agile Teams power the train:</u>
Cross-functional teams apply Scrum, XP and Kanban and built-in quality practices to produce working system increments every iteration.

Without Agile Teams and ARTs

- Teams are isolated, uncoordinated, can’t deliver end to end value
- Lack of collaboration
- Over-specialization. Bottleneck constraints.
- Teams are not aligned to a common mission; solutions show it
- Releasing is late, problematic and big bang
- No architectural and user experience integrity
- Too much WIP, multiplexing, low productivity

#3 – Cadence and Synchronization

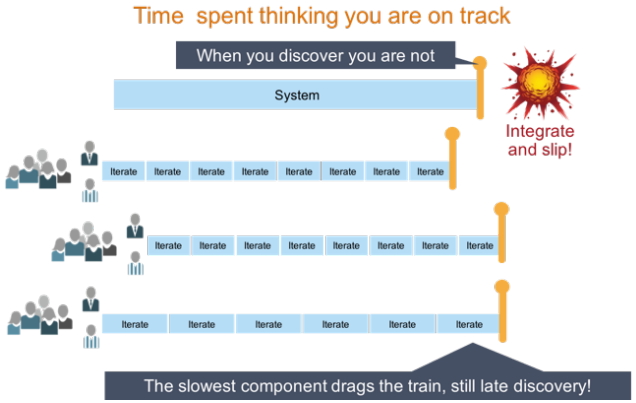


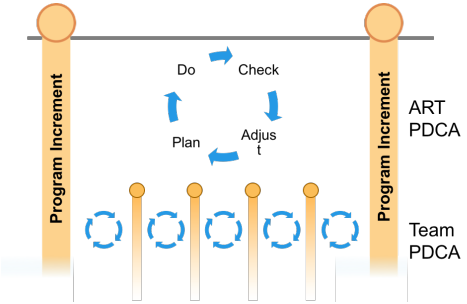
Cadence

- Transforms unpredictable events into predictable events
- Makes waiting times predictable
- Facilitates planning; provides more efficient use of resources

Synchronization

- Synchronization causes multiple events to happen at the same time
- Sync events facilitate cross-functional tradeoffs of people and scope





Without Cadence and Synchronization

- Entropy constantly increases
- Getting the right people to meetings is impossible
- Integration comes late
- Everyone shoves everything into the current timebox
- No forced integration and evaluation points
- Individual teams may be Agile, but the system is not iterating.
- Enterprise Agility is not achievable

#4 Critical Team Roles

Scrum Master facilitates team events, drives Agile behavior and coaches the team

---

Product Owner acts as the customer for team and prioritizes their work. Defines and accepts requirements

---

Development Team is everyone needed to define, build, and test an increment of value

Critical Program Roles

Release Train Engineer acts as the Chief Scrum Master for the train

Product Management is responsible for identifying Customer needs. Owns the vision and product backlog, and sequences features for optimum economics

System Architect/Engineering align ARTs to a common technological and architectural vision

Customer consumes the work of an ART. They are the ultimate arbiters of value

Business Owners are a small group of stakeholders who have the ultimate fiduciary, governance, efficacy, and ROI responsibility for the value delivered by an ART

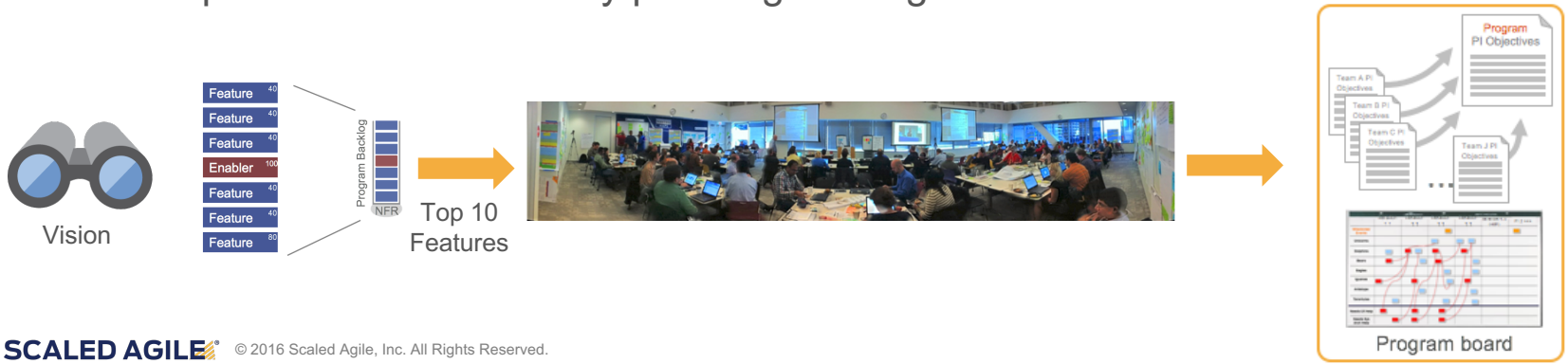
Without these clear role responsibilities?

- Responsibilities for requirements, design, architecture, implementation and deployment are unclear; no systems thinking
- Meetings are a waste of time and end without clear outcomes
- Teams find it hard to integrate due to incompatible components
- Vision and requirements are not clear; lack of prioritization
- Deliverables do not meet stakeholder expectations
- Impossible to improve ad hoc processes
- Programs must be centrally defined and managed

#5 PI Planning

PI Planning achieves alignment to a common mission.

- Teams create, and take responsibility, for plans
- Two days every 8 – 12 weeks (10 weeks is typical)
- Everyone attends, in person or remotely
- Product Management owns feature priorities
- Development teams own story planning and high-level estimates

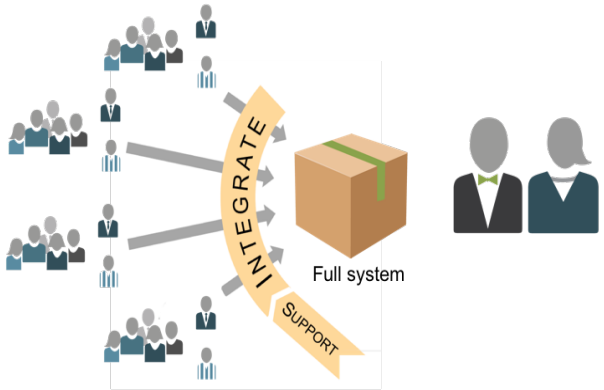


Without PI Planning

- Stakeholders and teams don’t have a clear understanding of the vison and roadmap
- Teams don’t know the business context and the most important objectives
- Lack of alignment between business and technology. “us” vs “them”.
- Dependencies are discovered too late
- Planning must be centralized
- Teams are not committed

#6 System Demo

Demonstrate the full Solution increment to stakeholders every Iteration.



- An integrated Solution demo
- Happens after the teams' demo (may lag by as much as one Iteration, maximum)
- Demo from the staging environment, or the nearest proxy

Without the System Demo?

- Waterfalled PIs. Late discovery of integration problems.
- Business Owners unclear about progress. Management intervenes.
- Management applies unrepresentative proxy metrics
- Quality is uncertain. Velocity is false.
- Little to no meaningful feedback
- No forcing function for continuous integration and test automation.
- Teams are sprinting, but the system isn't.

…WITHOUT EXECUTION, THERE IS NO TRUST

#7 Inspect and Adapt

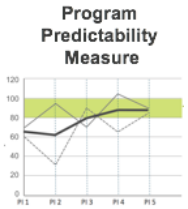
Inspect & Adapt provides a venue for comprehensive review of the PI outcomes and systematic improvements.

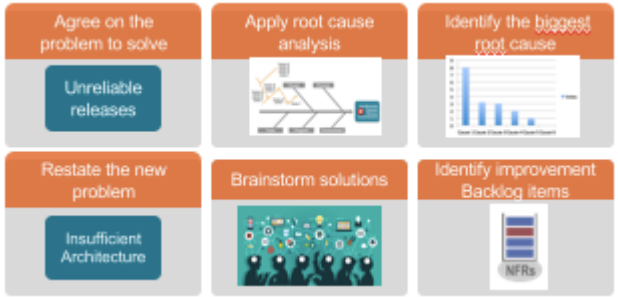
Three parts:

1. The PI System Demo
2. Quantitative measurement
3. The problem-solving workshop

Objective	BV	Actual BV
- Proof of concept with mock sounds	10	9
- Help with Radar POC	4	0
- Decide buy/make engine noises	3	0
===== **Stretch Objectives** =====	===	===
- Proof of concept with real sounds	7	7
Total	17	16

Achievement: 94%



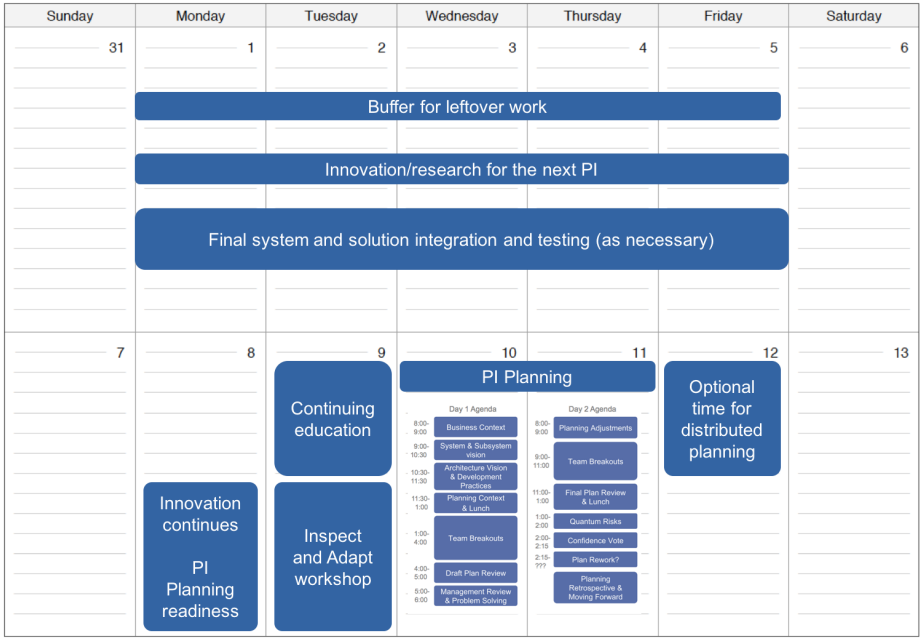


Without Inspect and Adapt?

- Programs do not predictably delivery committed objectives
- No way to improve systemically
- Improvement efforts address symptoms, not root causes
- Centralized improvement mandates don’t reflect actual development problems
- Management stakeholders are not involved in changing the system

#8 IP Iteration

The IP Iteration is like extra oxygen in the tank: without it the train may start gasping under the pressure of a plan that forgives no mistakes, nor provides dedicated time for innovation

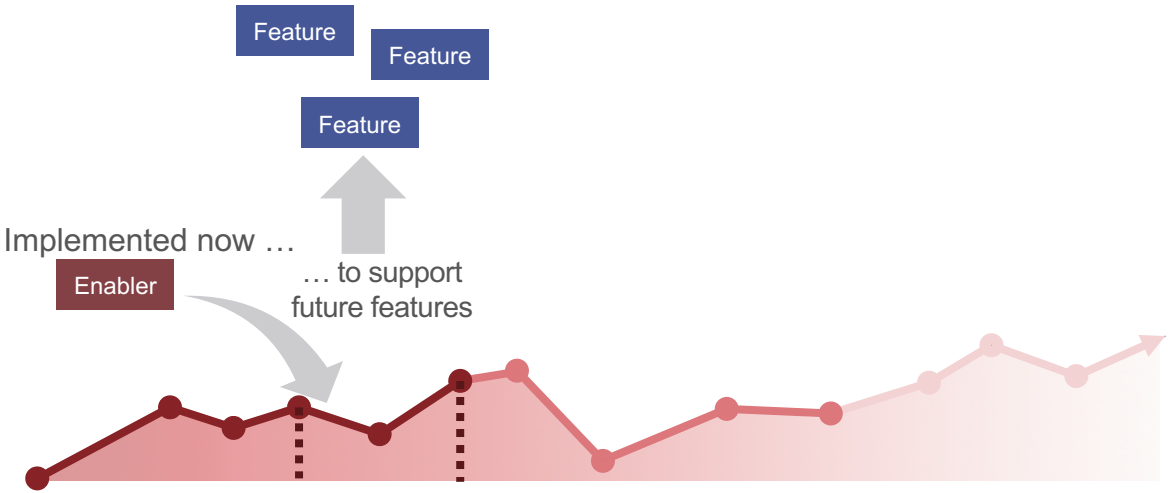


Without Innovation and Planning?

- No capacity buffer, ARTs are not predictable
- There is no time for innovation due to delivery urgency
- Technical debt just grows and grows
- People burn out
- Cadence and schedule challenges: no allocated time for teams to plan together, demo together and improve together
- No time for continuing education
- Real velocity slows

#9 The Architectural Runway

Architectural runway provides “just enough” technical enablement to keep program velocities high and avoid excessive redesign



Without Architectural Runway?

- Architecture and infrastructure is deferred to new business requirements
- Development pace is unsustainable
- Architecture deteriorates under the pressure of ”now”
- Velocity peaks for a while, then falls off
- Solution robustness, maintainability and quality decay
- Solutions obsolesce quickly

#10 Lean-Agile Leadership

Successful transformations are based on educating management first. They become lean-thinking manager-teachers", who lead, rather than follow, the transformation

People are already doing their best; the problems are with the system.

Only management can change the system.

—W. Edwards Deming

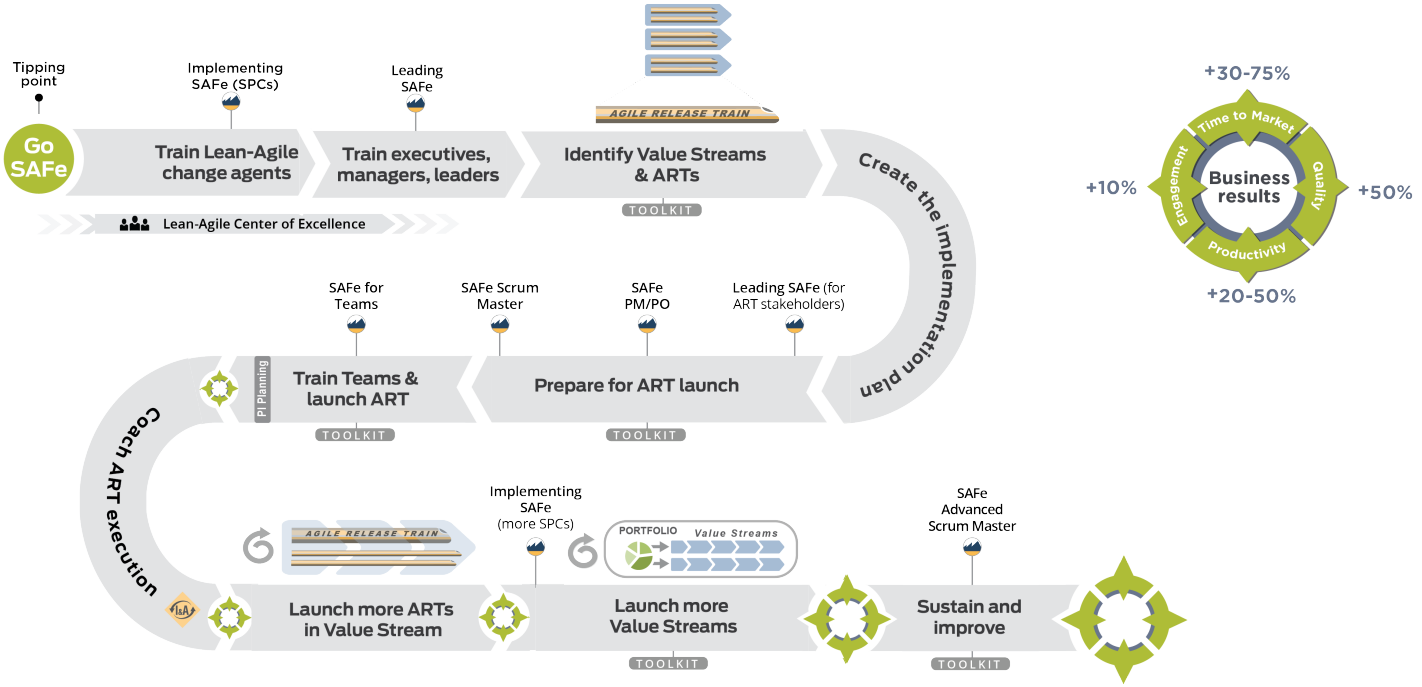
- Lead the change
- Know the way; emphasize life-long learning
- Develop people
- Inspire and align with mission; minimize constraints
- Decentralize decision-making
- Unlock the intrinsic motivation of knowledge workers

Without Lean-Agile Leadership

- Who do teams learn from?
- The development processes cannot continuously improve
- “Agile” development with non-Agile governance: Iron triangle of scope, time and budget. Centralized planning and commitments.
- Lead time is long as decisions must be escalated
- Insufficiently powerful coalition for change. Lean-Agile transformation wanes; no dedicated change management team
- People cannot experiment, fail and learn
- People are over-controlled, underutilized, demotivated

Implementing Essential SAFe

SAFe Implementation Roadmap



Beyond Essential SAFe

Trigger	Where to next?	Go To
Desire to improve the throughput of the ART	Implement a program kanban system	http://www.scaledagileframework.com/program-and-value-stream-kanbans/
Need to provide longer term visibility to stakeholders	Create a three-PI roadmap	http://www.scaledagileframework.com/roadmap/
Trouble with integrating the system for the system demo	Build a System Team	http://www.scaledagileframework.com/system-team/
Measure the performance of the ART	Establish Lean-Agile metrics	http://www.scaledagileframework.com/metrics/
Synchronization between ARTs is hard to manager	Implement the Value Stream Level	http://www.scaledagileframework.com/value-stream-level/
Need to add financial governance	Train portfolio and establish a Portfolio Level	http://www.scaledagileframework.com/portfolio-level/
Require traceability for high-assurance systems	Build a Solution Intent	http://www.scaledagileframework.com/solution-intent/

Questions?