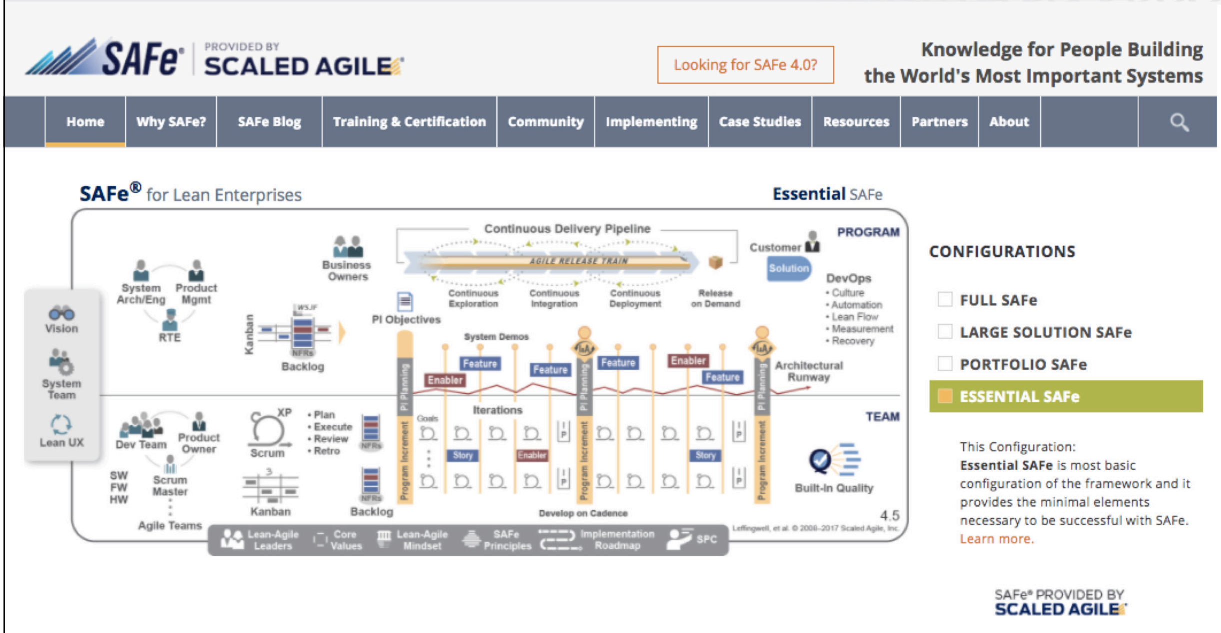
Essential SAFe® 4.5
Overview and Assessment

SCALED AGILE®

Introducing the Essential SAFe configuration...



Knowledge for People Building the World's Most Important Systems

CONFIGURATIONS

- □ FULL SAFe
- □ LARGE SOLUTION SAFe
- □ PORTFOLIO SAFe
- ■ ESSENTIAL SAFe

This Configuration:
Essential SAFe is most basic configuration of the framework and it provides the minimal elements necessary to be successful with SAFe. Learn more.

Why focus on the essentials?

Because there's a danger in skipping critical elements

Findings from the field

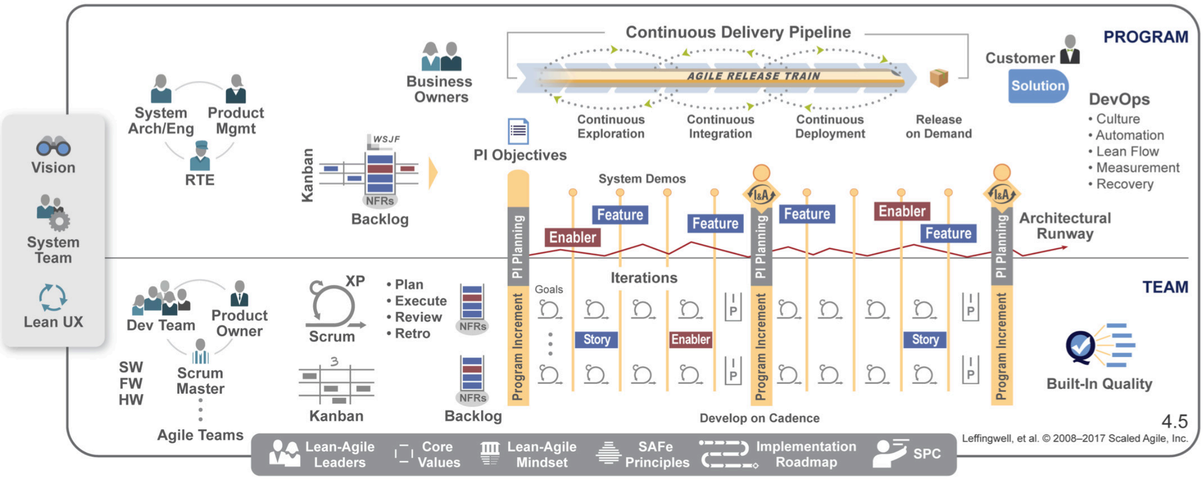
Successful rollout, but still struggling. Root causes:

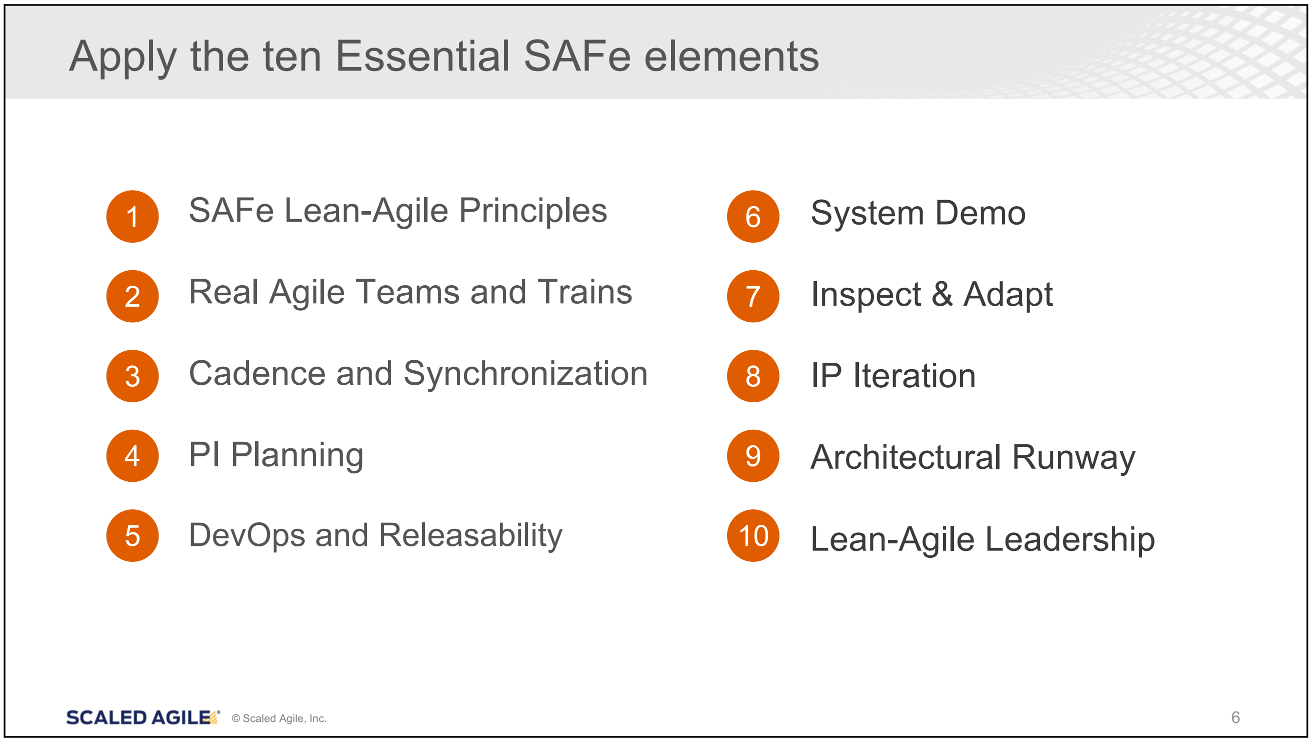
- Not doing Inspect & Adapt
- No Innovation and Planning iteration
- Individual Agile Teams were not actually cross-functional
- No routine System Demo

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. We're 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."*

Start with the basics of Essential SAFe ...





Apply the ten Essential SAFe elements

1 SAFe Lean-Agile Principles

2 Real Agile Teams and Trains

3 Cadence and Synchronization

4 PI Planning

5 DevOps and Releasability

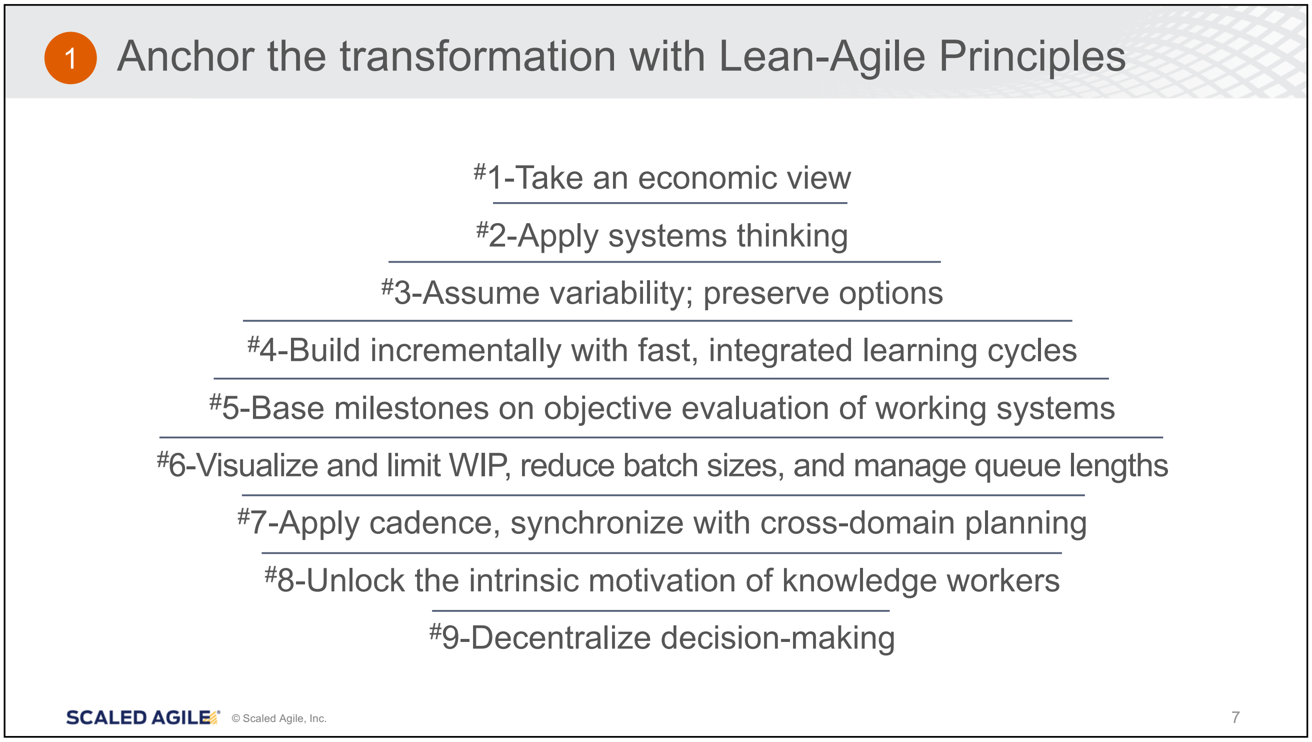
6 System Demo

7 Inspect & Adapt

8 IP Iteration

9 Architectural Runway

10 Lean-Agile Leadership



1 Anchor the transformation with 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

Without a shared understanding of principles ...

□ There is no systematic way to adapt practices to local context

□ Business outcomes do not significantly improve

□ Practices and measures that were once beneficial become problematic

□ Lean-Agile Mindset is unachievable

□ Conflict and disagreement on processes and practices are difficult to resolve

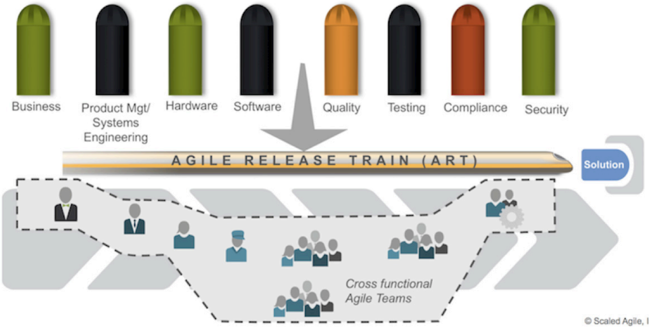
2 Implement Real Agile Teams and Trains

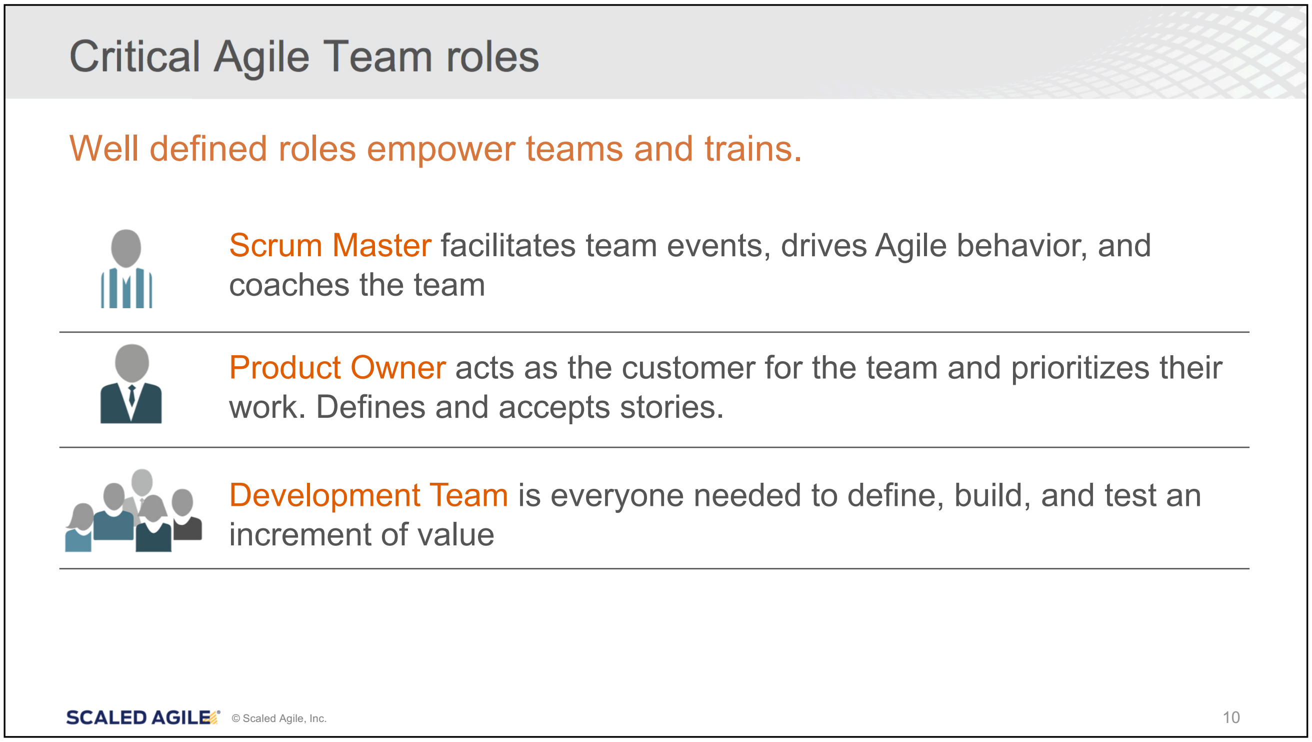
Cross-functional Agile Teams and trains work towards a common mission and operate with architectural and Lean UX guidance.

Agile Team

Agile Teams power the train:
Cross-functional teams apply Scrum, XP and Kanban and Built-in Quality practices to produce working system increments every iteration.

Agile Release Trains (ARTs) apply systems thinking and build a cross-functional organization optimized to facilitate the flow of value from idea to release.


Business
Product Mgt/ Systems Engineering
Hardware
Software
Quality
Testing
Compliance
Security
AGILE RELEASE TRAIN (ART)
Solution
Cross functional Agile Teams



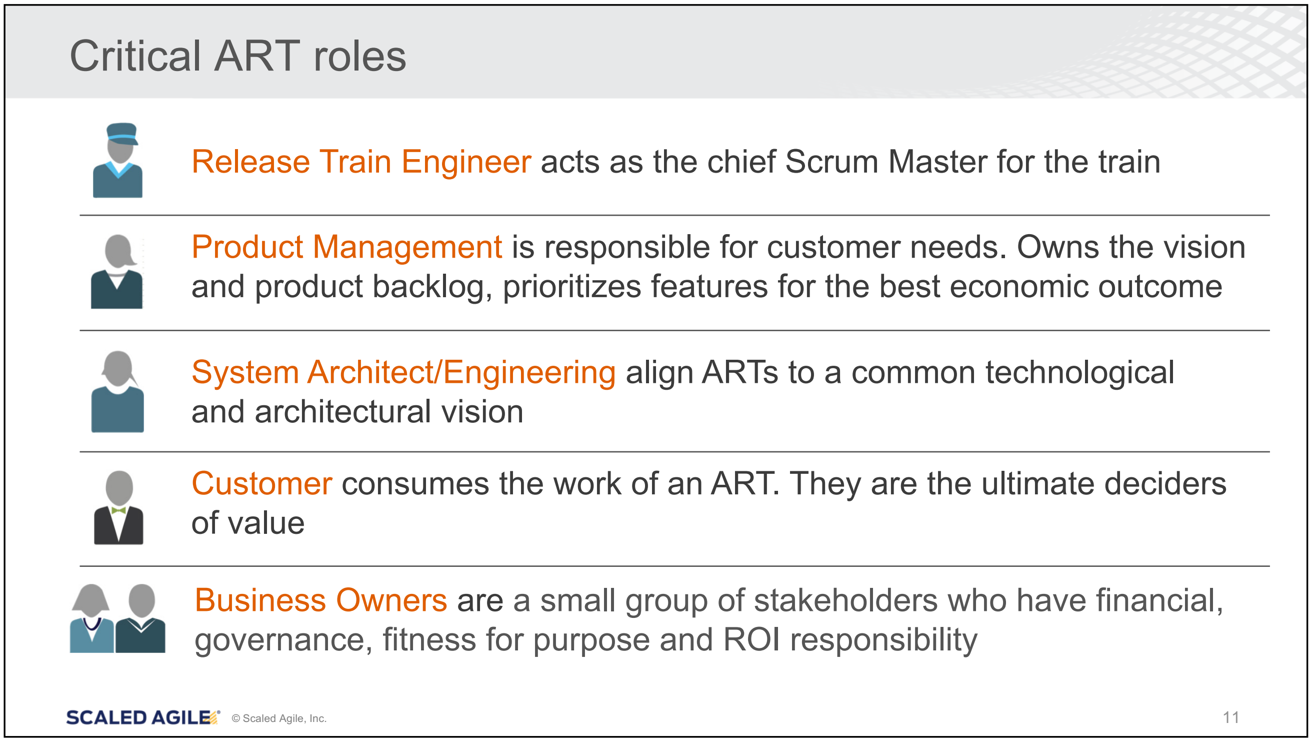
Critical Agile Team roles

Well defined roles empower teams and trains.

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

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

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



Critical ART roles

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

Product Management is responsible for customer needs. Owns the vision and product backlog, prioritizes features for the best economic outcome

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

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

Business Owners are a small group of stakeholders who have financial, governance, fitness for purpose and ROI responsibility

Without Real Agile Teams and Trains ...

☐ Responsibilities are unclear; delayed decision-making

☐ The skills needed to define, build, test and deliver value are not fully present and accountable. Over-specialization and bottlenecks inhibit flow.

☐ Teams locally optimize and can’t deliver end-to-end value

☐ No architectural and user experience integrity; solution features and components evolve incompatibly

☐ Vision and requirements are not clear and prioritization is extremely difficult

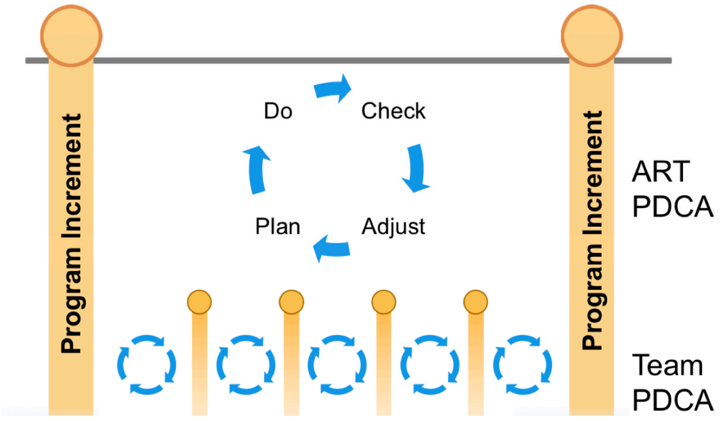
3 Apply Cadence and Synchronization

Cadence

- Transforms unpredictable events into predictable events
- Makes wait 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 ...

□ No development rhythm

□ Gradual decline into disorder and lack of predictability

□ It's hard to schedule planning, retrospectives, demos and other key events

□ Difficult to adjusting to changing priorities

□ Teams are constantly overloaded

4 Create alignment with PI Planning

No event is more powerful than PI planning. It's the magic in SAFe—the alignment and energy created in just two days saves months of delays.

- All stakeholders face-to-face, whenever possible
- Management sets the mission with minimum possible constraints
- Important stakeholder decisions are made immediately
- Requirements and design emerge
- Teams create and take responsibility for plans

See the short PI planning example video
https://www.youtube.com/watch?v=ZZAtl7nAB1M

Without PI Planning ...

□ Stakeholders, teams, and management are not aligned

□ Demand doesn't match capacity; no predictability; excess WIP

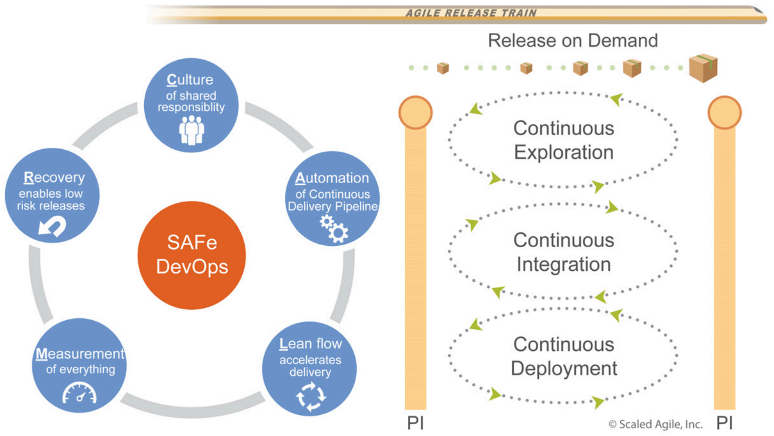
□ Lack of trust between stakeholders and teams

□ Late discovery of dependencies cause delays

□ Low commitment, ownership, and employee engagement

5 Improve DevOps and Releasability

DevOps improves collaboration and flow between Development and IT Operations with a continuous delivery pipeline.


Culture of shared responsiblity
Automation of Continuous Delivery Pipeline
Lean flow accelerates delivery
Measurement of everything
Recovery enables low risk releases
SAFe DevOps
AGILE RELEASE TRAIN
Release on Demand
Continuous Exploration
Continuous Integration
Continuous Deployment
PI
PI
© Scaled Agile, Inc.

Without DevOps and Releasability...

□ Value delivery is seriously delayed

□ Reduced quality of deployments and high production defects

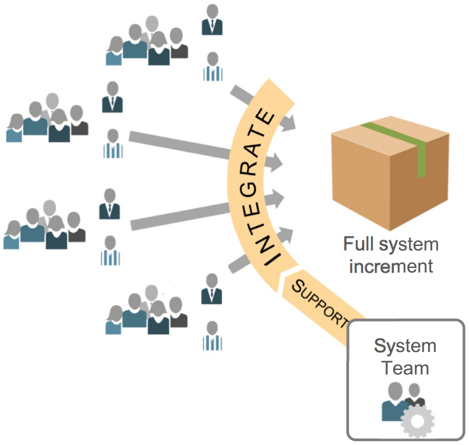
□ More frequent releases are not possible, increasing time to market

□ Large batches of code are pushed to production, resulting in production errors, and emergencies

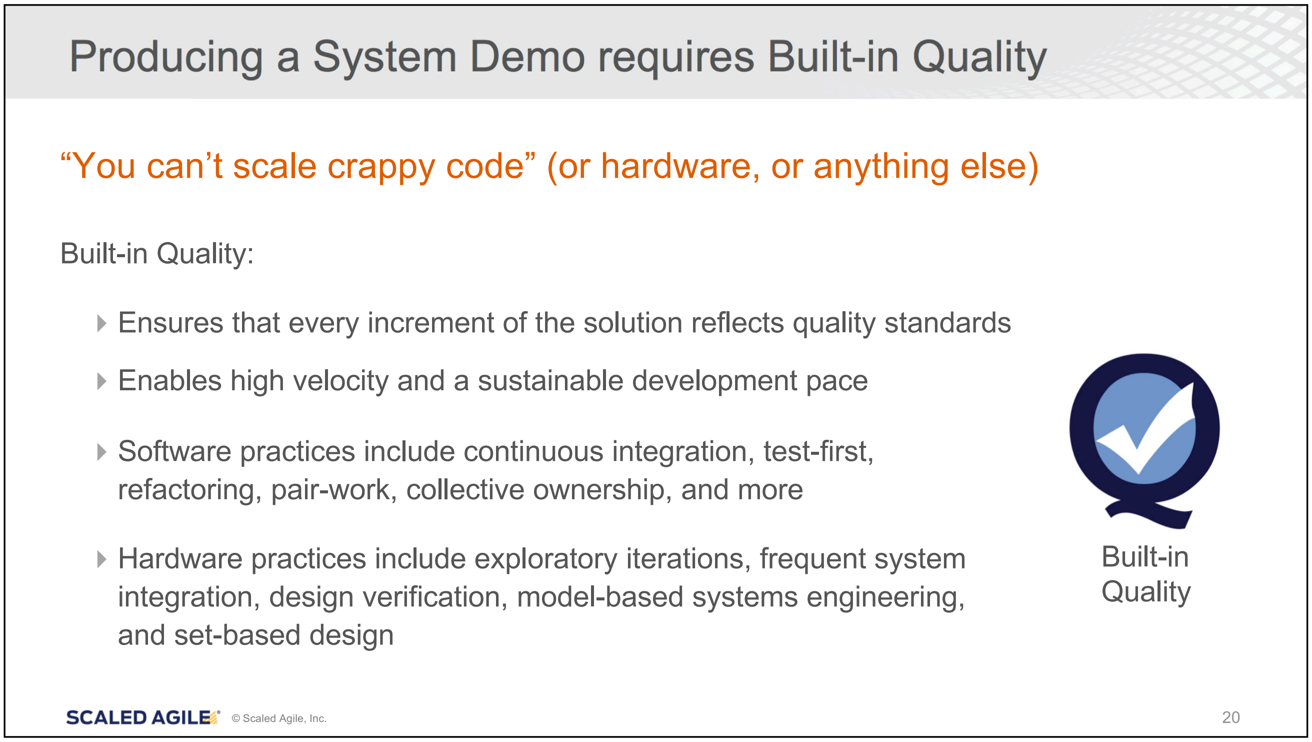
□ Friction between development and operations limits collaboration, learning, and cultural change

6 Get fast feedback with the System Demo

Demonstrate the full system increment to stakeholders every iteration



- Features are either functionally complete or ‘toggled off'
- New features work together, and with existing functionality
- Demo from a production-like staging environment



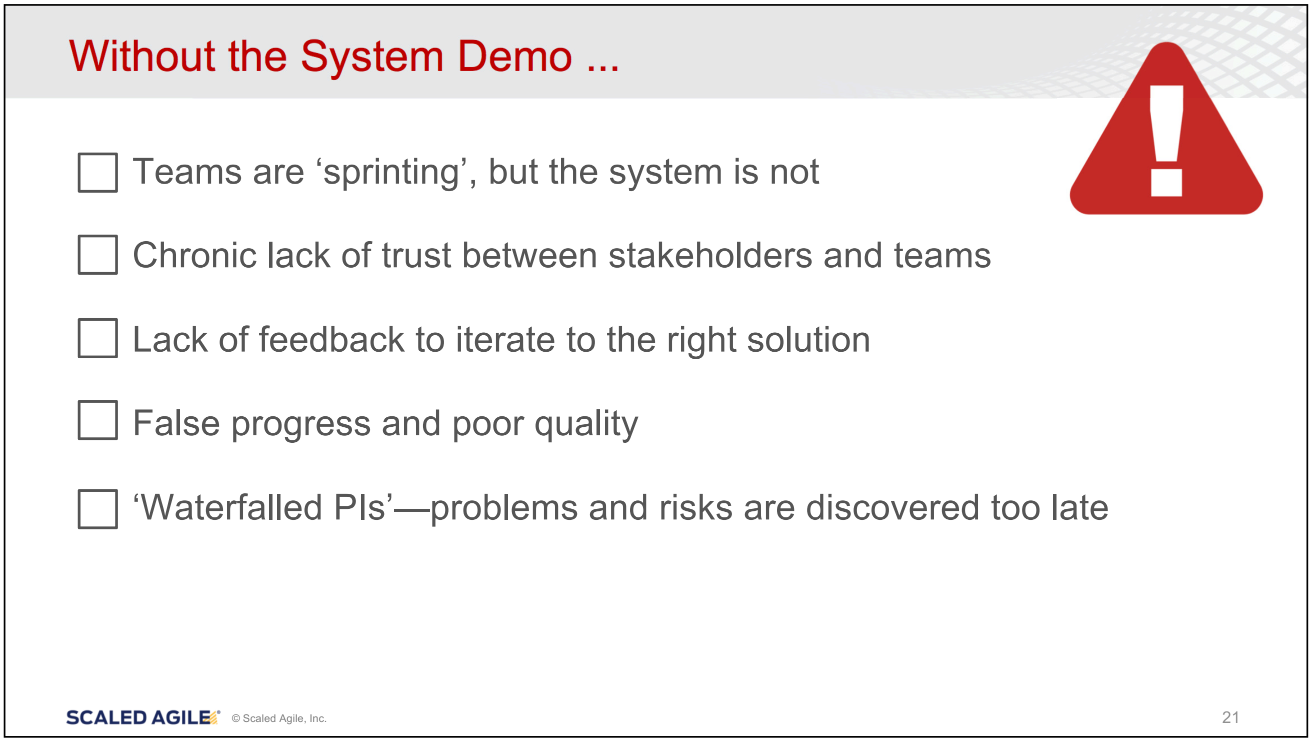
Producing a System Demo requires Built-in Quality

“You can’t scale crappy code” (or hardware, or anything else)

Built-in Quality:

- Ensures that every increment of the solution reflects quality standards
- Enables high velocity and a sustainable development pace
- Software practices include continuous integration, test-first, refactoring, pair-work, collective ownership, and more
- Hardware practices include exploratory iterations, frequent system integration, design verification, model-based systems engineering, and set-based design

Built-in Quality



Without the System Demo ...

□ Teams are ‘sprinting’, but the system is not

□ Chronic lack of trust between stakeholders and teams

□ Lack of feedback to iterate to the right solution

□ False progress and poor quality

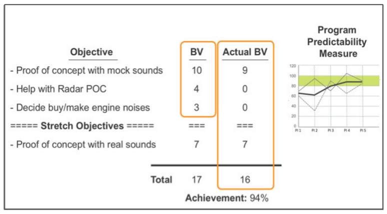
□ ‘Waterfalled PIs’—problems and risks are discovered too late

7 Relentlessly improve with Inspect & Adapt

Inspect & Adapt (I&A) supports systematic review of PI outcomes and continuous improvement.

PI System Demo

Quantitative measurement

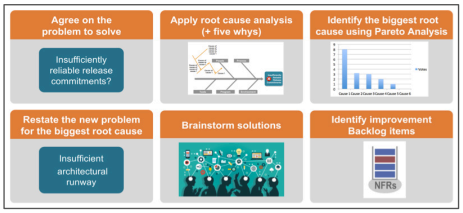


Objective	BV	Actual BV
- Proof of concept with mock sounds	10	9
- Help with Radar POC	4	0
- Decide buy/make engine noises	3	0
===== **Stretch Objectives** =====	===	===
- Proof of concept with real sounds	7	7
Total	17	16

Achievement: 94%

Program Predictability Measure
120
100
80
60
40
20
0
PI 1
PI 2
PI 3
PI 4
PI 5

Problem-Solving Workshop



Agree on the problem to solve

Insufficiently reliable release commitments?

Apply root cause analysis (+ five whys)

Identify the biggest root cause using Pareto Analysis

Restate the new problem for the biggest root cause

Insufficient architectural runway

Brainstorm solutions

Identify improvement Backlog items

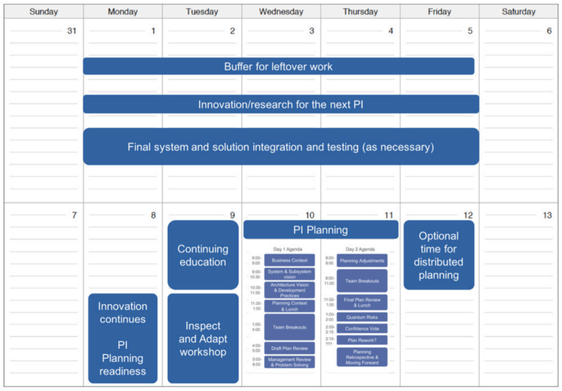
Without Inspect & Adapt ...

- □ No systemic improvement; problems persist
- □ No means to measure or establish delivery predictability
- □ Improvement efforts address symptoms, not root causes
- □ Leaders who could change the system are not engaged
- □ Low morale

8 Dedicate time for Innovation and Planning

The IP Iteration provides an estimating buffer for meeting PI objectives, and dedicated time for innovation, education, PI planning and I&A events.

IP
Program Increment



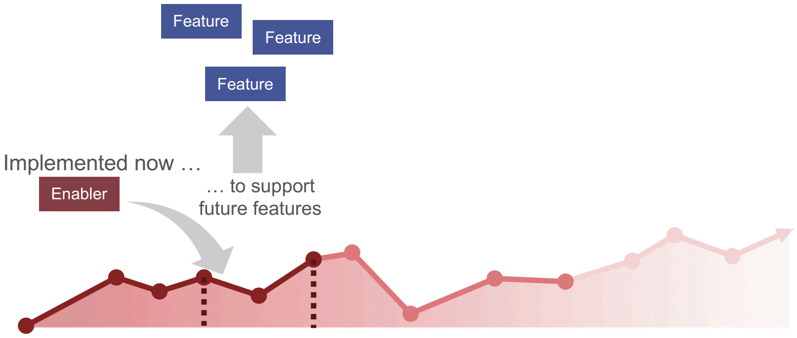
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
31
1
2
3
4
5
6
Buffer for leftover work
Innovation/research for the next PI
Final system and solution integration and testing (as necessary)
7
8
9
10
11
12
13
Continuing education
PI Planning
Optional time for distributed planning
Day 1 Agenda
8:00-9:00 Business Context
9:00-10:30 System & Subsystem vision
10:30-11:30 Architecture Vision & Development Practices
11:30-1:00 Planning Context & Lunch
1:00-4:00 Team Breakouts
4:00-5:00 Draft Plan Review
5:00-6:00 Management Review & Problem Solving
Day 2 Agenda
8:00-9:00 Planning Adjustments
9:00-11:00 Team Breakouts
11:00-1:00 Final Plan Review & Lunch
1:00-2:00 Quantum Risks
2:00-2:15 Confidence Vote
2:15-??? Plan Rework?
Planning Retrospective & Moving Forward
Innovation continues
PI Planning readiness
Inspect and Adapt workshop

Without the IP Iteration ...

- □ Lack of estimating buffer and poor predictability
- □ ‘Tyranny of the urgent’ iteration inhibits innovation
- □ Technical debt grows uncontrollably
- □ Lots of overtime and people burn out
- □ No time for teams to plan together, demo together, and improve together

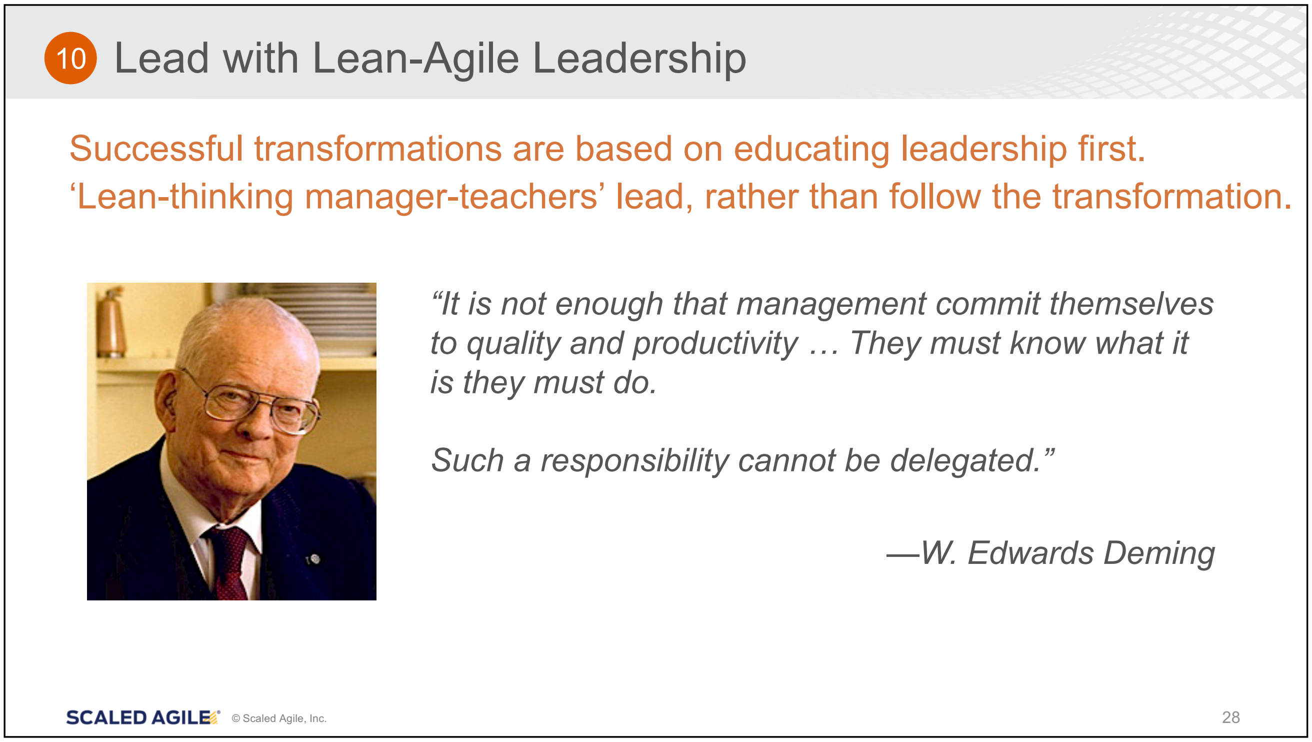
9 Enable fast feature delivery with Architectural Runway

Architectural runway provide ‘just enough’ technical enablement to keep velocity high and avoid excessive redesign and delays.



Without Architectural Runway ...

- □ Architecture progressively decays under the ‘urgency of now’
- □ Velocity peaks for a while, then falls off
- □ Infrequent and irregular releases
- □ Solution robustness, maintainability, and quality decay
- □ Unsustainable development pace

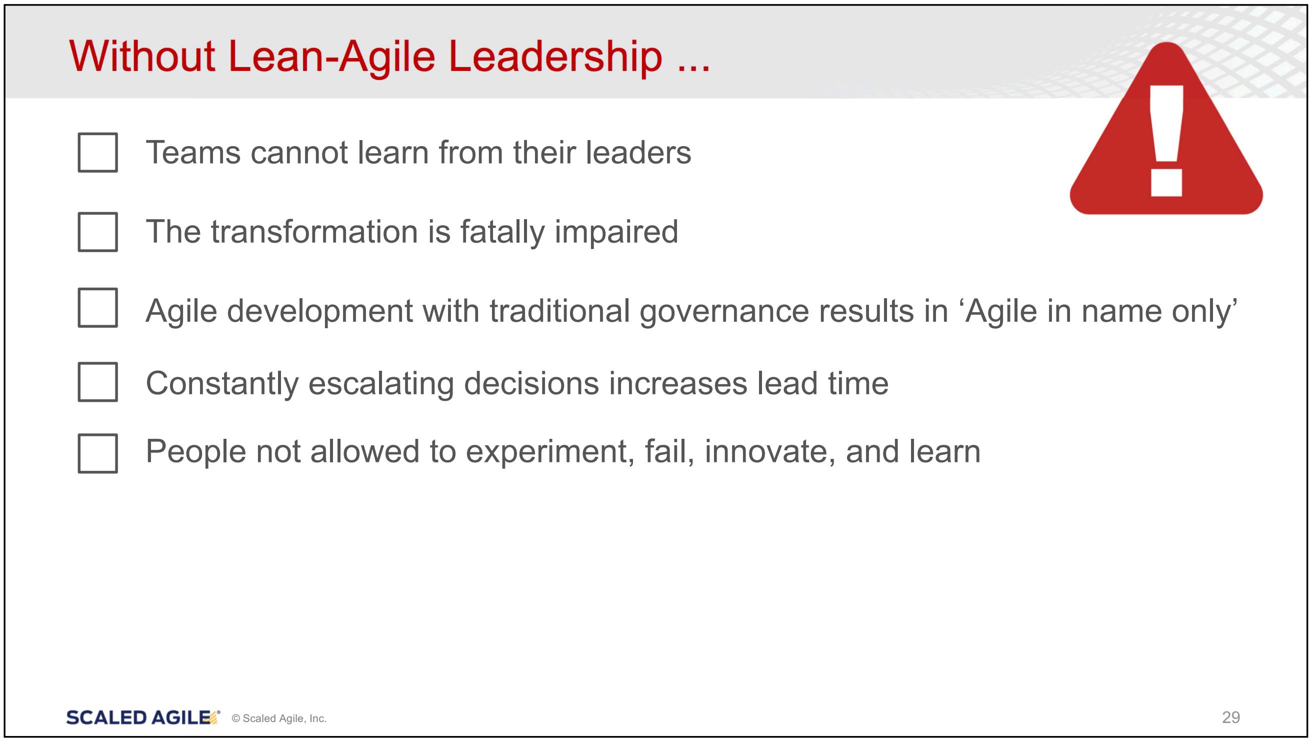


10 Lead with Lean-Agile Leadership

Successful transformations are based on educating leadership first. ‘Lean-thinking manager-teachers’ lead, rather than follow the transformation.

> *“It is not enough that management commit themselves to quality and productivity … They must know what it is they must do.*
>
> *Such a responsibility cannot be delegated.”*
>
> *—W. Edwards Deming*

SCALED AGILE

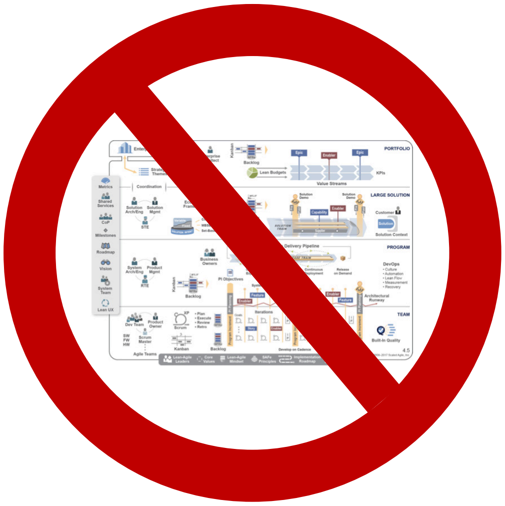


Without Lean-Agile Leadership ...

- □ Teams cannot learn from their leaders
- □ The transformation is fatally impaired
- □ Agile development with traditional governance results in ‘Agile in name only’
- □ Constantly escalating decisions increases lead time
- □ People not allowed to experiment, fail, innovate, and learn

SCALED AGILE

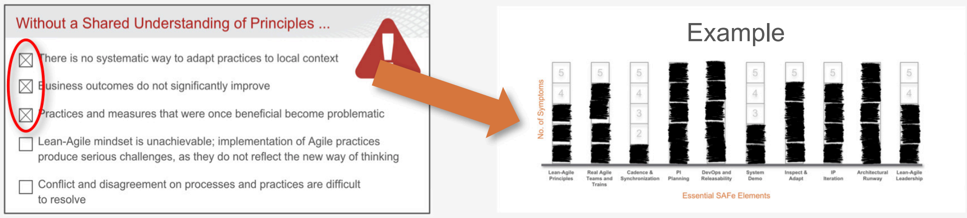
One more thing, without Lean-Agile Leadership ...



SAFe will not work!

Exercise: Essential SAFe Self-Assessment

- Fill out the Essential SAFe Self-Assessment on the next page
- Use the previous slides to identify the symptoms that exist in your enterprise
- For each Essential SAFe element, shade one box for every symptom identified



Without a Shared Understanding of Principles ...

☒ There is no systematic way to adapt practices to local context

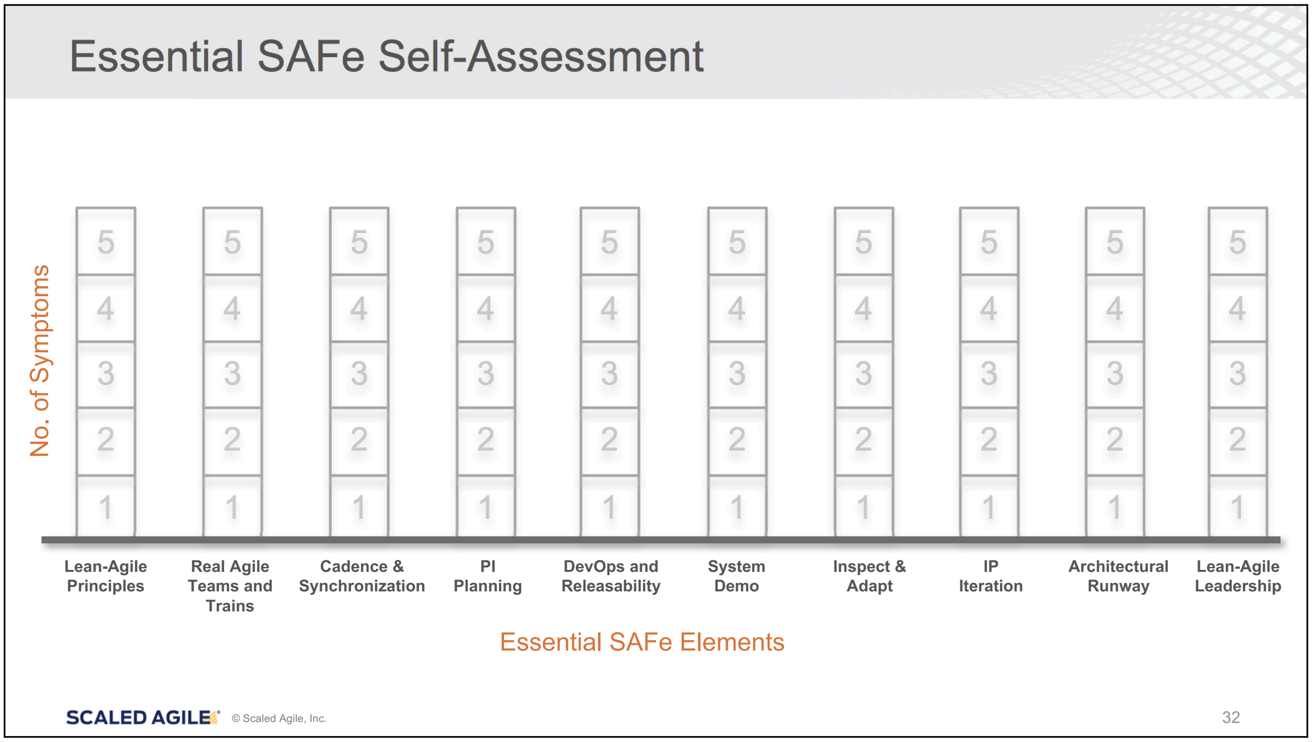
☒ Business outcomes do not significantly improve

☒ Practices and measures that were once beneficial become problematic

□ Lean-Agile mindset is unachievable; implementation of Agile practices produce serious challenges, as they do not reflect the new way of thinking

□ Conflict and disagreement on processes and practices are difficult to resolve

10 min



Essential SAFe Self-Assessment

No. of Symptoms

Lean-Agile Principles	Real Agile Teams and Trains	Cadence & Synchronization	PI Planning	DevOps and Releasability	System Demo	Inspect & Adapt	IP Iteration	Architectural Runway	Lean-Agile Leadership
5	5	5	5	5	5	5	5	5	5
4	4	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1

Essential SAFe Elements

Questions?