Agile Management Change:

Why Enterprise Agile is Fragile

Agile Transformation

Agile transformation improves when policy, collaboration, and decision-making guidance has direct Agile experience

The focus is to enable those who:

- Create policy
- Reorganize the company
- Create job roles
- Design compensation and rewards
  - For the individual
  - For the team
  - Across departments
- Budget
- Finance

The deck reviews:

1. Impact of Experience and Training on enterprise guidance
2. Transformation Models:
   1. Agile Fluency model
   2. Scaled Agile Framework Implementation Roadmap
3. Agile Change Management:
   - Develop
   - Enable
   - Adapt
   - Own

Impact of Experience and Training on Enterprise Guidance

The trouble with training:

1. Learning without practice
2. Training and forgetting
3. Experience without context

Agile Enterprise needs Experienced Agile Leadership

Theory without practice or experience leaves a gap to understand both Agile software development and Lean-Agile organization transformation

Learning without practice remains theory

70:20:10 is a learning and development model*:

- 10% of learning comes from from structured training or formal courses
- 20% of learning comes from interaction with others
- 70% of learning comes from job-related or workplace learning (on-the-job)

Does management have on-the-job, Agile development experience or theoretical experience?

Does this learning matter?

*70:20:10 Model for Learning and Development and Why 70 is a key metric for learning and development

… or, for the visually stimulated 70:20:10

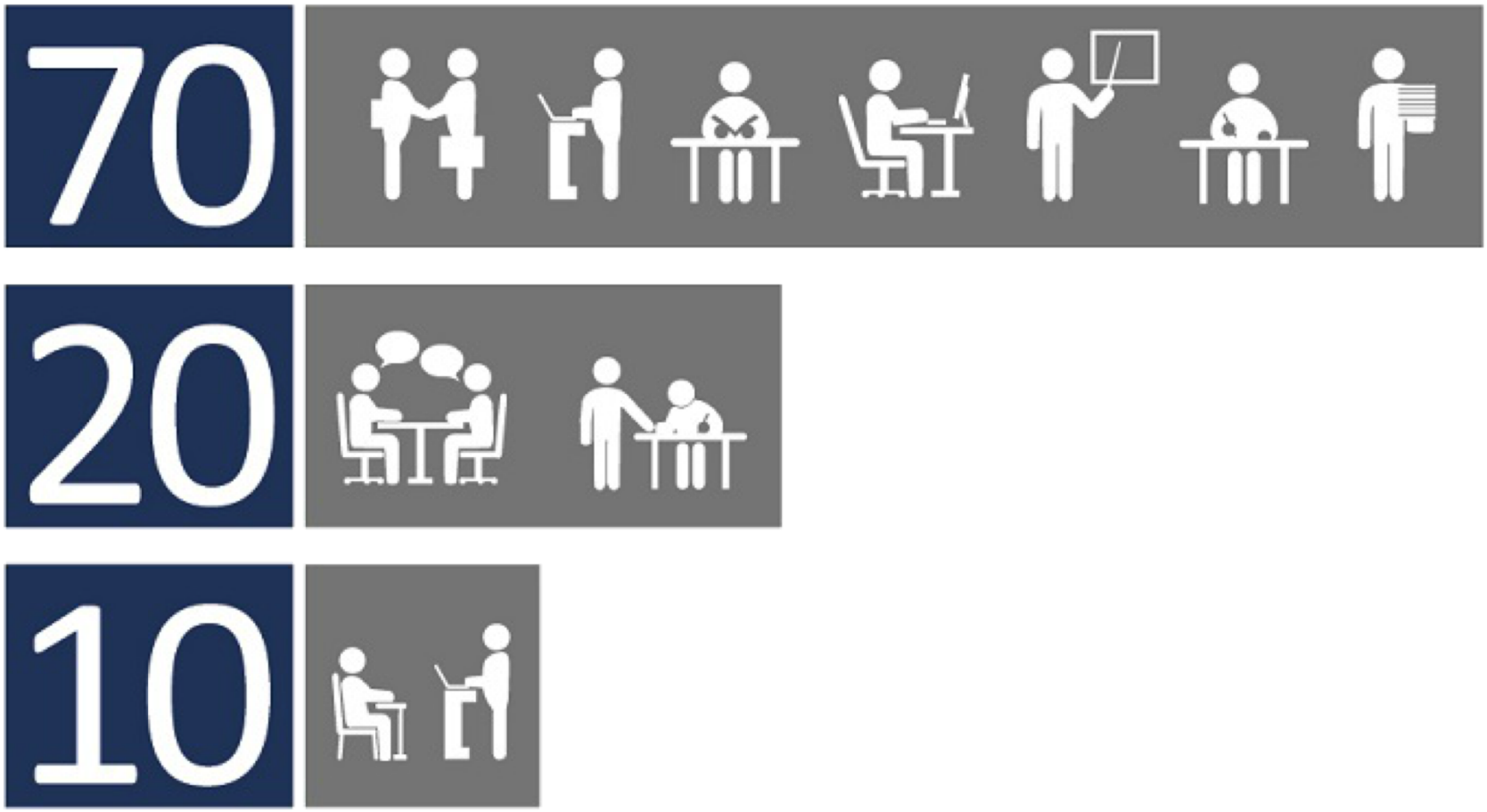


Image source: https://ethoscrs.com.au/702010/

Agile Adoption needs Agile practice experiences

Agile transformation requires whole-system learning and practice

Training without practice decays

- 20 minutes after training you forget ~40%*
- One day after training you forget more than 60%

To improve memory and learning design training with meaning to the participant as well as understand the impact stress on learning

Improve training with study and repetition:

- Experience as a developer within an Agile development team
- Experience as a Scrum Master or Agile Coach with a development team

*[Ebbinghaus forgetting curve]

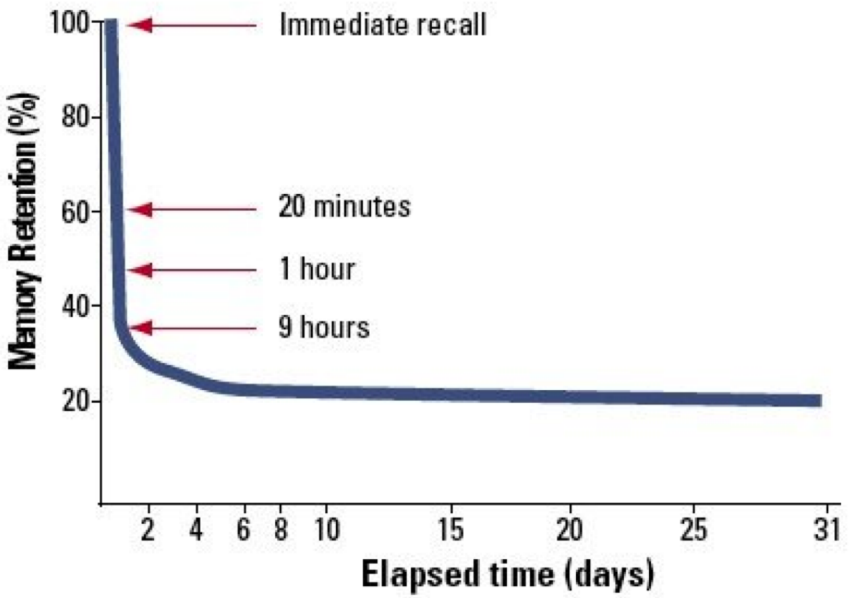
Do managers have on-the-job Scrum Master or Agile Coach experience in a Scrum Team?

Does this experience matter?

Image source: dailymail.co.uk

Or, for the visually stimulated: the forgetting curve

FIGURE 1.
The forgetting curve



The "forgetting curve" was developed by Hermann Ebbinghaus in 1885. Ebbinghaus memorized a series of nonsense syllables and then tested his memory of them at various periods ranging from 20 minutes to 31 days. This simple but landmark research project was the first to demonstrate that there is an exponential loss of memory unless information is reinforced.

Stahl SM, Davis RL, Kim D, et al. *CNS Spectr.* Vol 15, No 8. 2010.

Image source: https://elearningindustry.com/

Agile Adoption needs experiences to practice agile

Agile transformation requires whole-system learning and practice

Policy without experience is conjecture to impact decisions for:

1. Project method:
   - Sequential, Hybrid, Agile, Scaled Agile Framework, EPMO
2. Development frameworks:
   - Extreme Programming, Lean Development, Acceptance Test Driven Development, Scrum
3. Human Capital:
   - Job title, management role, performance incentives, staffing
4. Finance:
   - Budget, Enterprise Resource Planning/Utilization, Sales, Revenue, Contracts

Does executive-level, management, or team training provide enough context to set policy or organization design?

Does context matter?

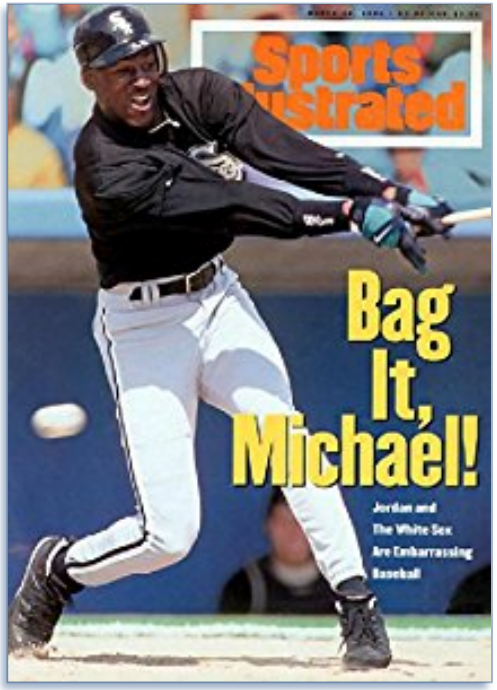


Image source: https://si.com

Transformation models

Scale with system:

1. Scaled Agile Framework Transformation Roadmap
2. Agile Fluency Model
3. Agile Organization Adoption

Enterprise Agility challenges many large, global organizations

Many frameworks exist, the following are two frameworks* for context. The Agile Management Change solution will address any Agile framework.

Agile Fluency Model

- Describes an agile team's pathway and helps coaches, consultants, and change agents put model insights into practice
- Goal is for every team to work at the level of fluency that best fits their business' needs
- The model can:
  - Chart a course for the team,
  - Create alignment with management, and
  - Secure organizational support for improvement
- More about the Agile Fluency Project

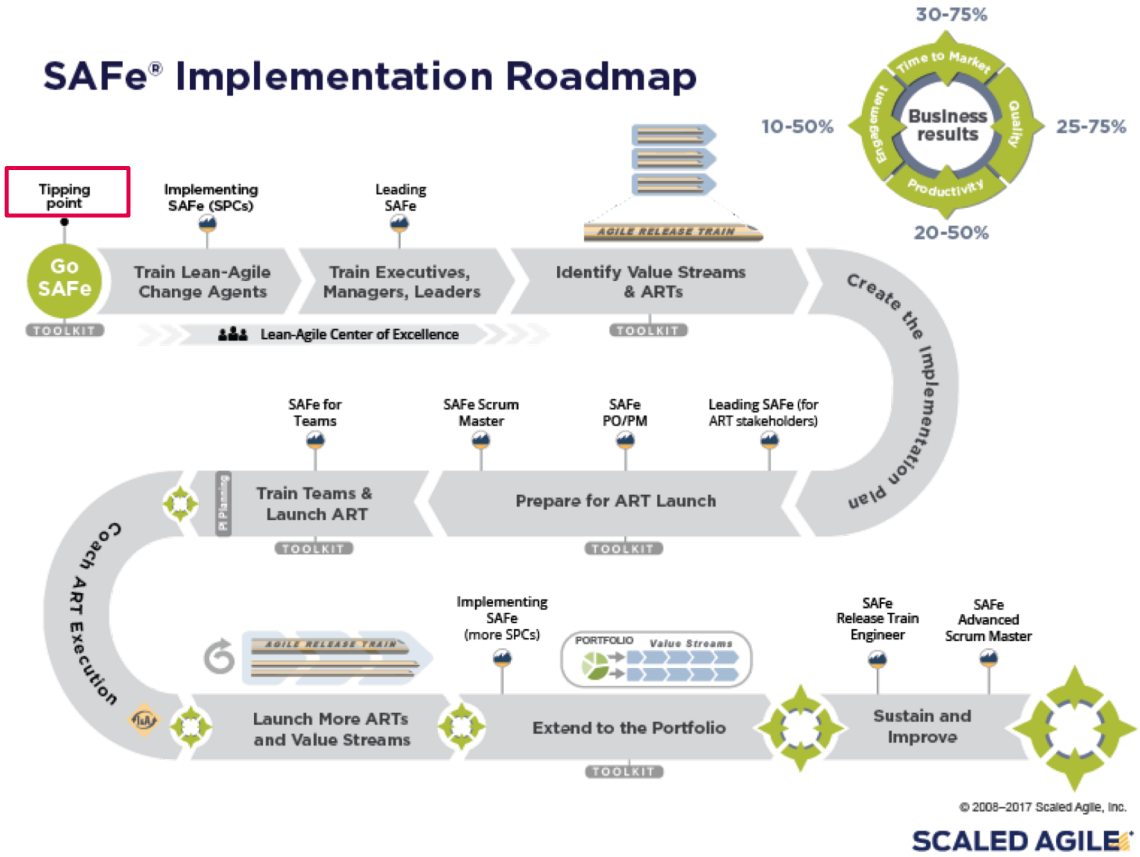
Scaled Agile Framework (SAFe)

- A leading framework for enterprise agility
- Mission includes: to improve business outcomes of the enterprises who build and depend on the world's most important systems
- Fourth iteration and has been adopted by 70% of the Fortune 100
- More about the SAFe

**Recent financial technology engagement uses SAFe and Agile Fluency to enable global, enterprise Agile transformation*

SAFe implementation roadmap: focus on training

Agile impacts the team, to scale Agile, both the safe enterprise and the fluent team rely on each other



Tipping Point:

1. Train Lean-Agile Change Agents
2. Train Executives, Managers, Leaders

If 10% of learning through classroom is retained and within 24 hours 60% is forgotten, part of leadership and learning must include:

- Train early and often,
- Assure practice,
- Provide context, and
- Key in on experience

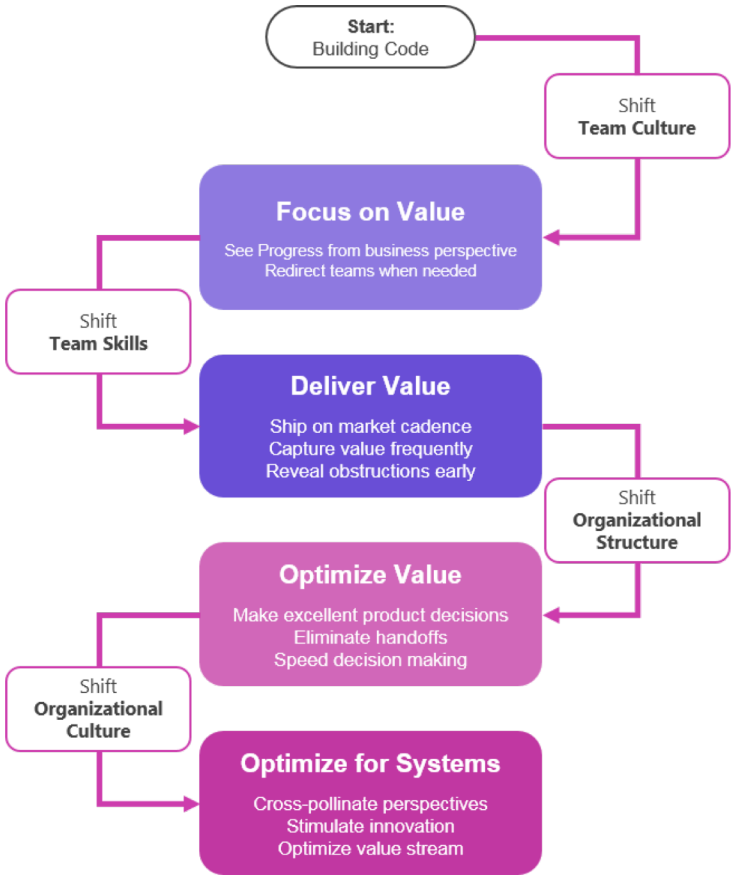
More with SAFe on slides 24 - 28

Image source: http://www.scaledagileframework.com/implementation-roadmap/

Agile Fluency Model

How Agile teams typically progress as they develop new capabilities

- A successful team begins as a collection of individuals with complementary technical skills.
- As the team adopts agile practices, a *team culture shift* occurs: instead of planning in terms of technical considerations, such as software layers or modules, the team now plans in terms of business, customer, or user benefit, exhibiting **Focus on Value** fluency.
- Mastery of technical practices like test driven development requires greater investment and, usually, more time. Once a *team skills shift* occurs that eliminates technical limitations to delivering working software, the team exhibits **Deliver Value** fluency.
- Where circumstances require, the team may internalize the capability to understand and address market needs. When an *organizational structure shift* moves key business capabilities inside the team, the team may exhibit **Optimize Value** fluency.



Agile Fluency Model

Fluency requires dedication and support

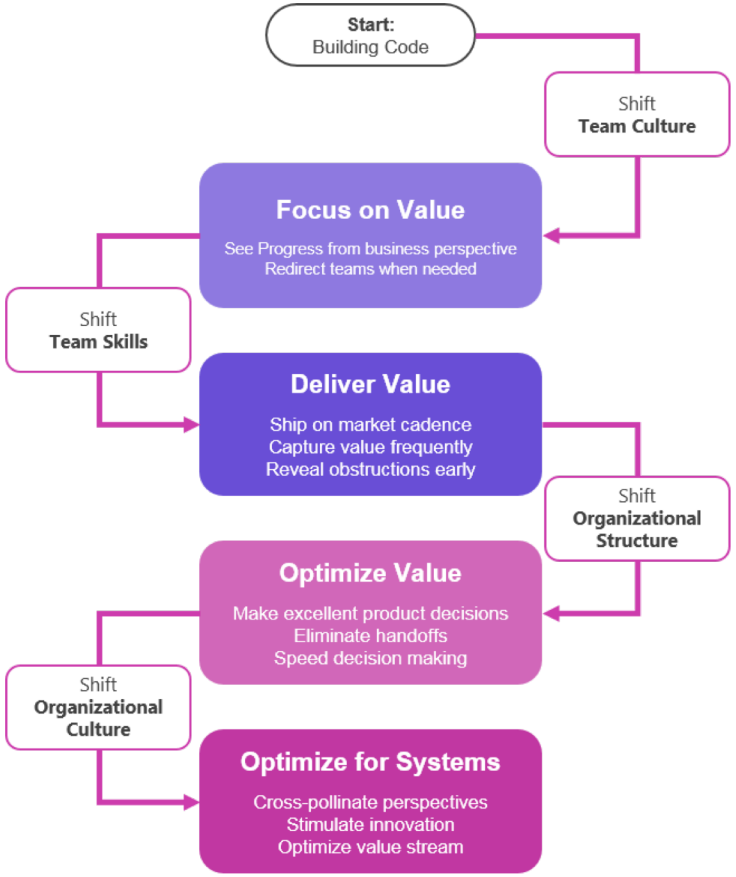
Skillful practice under pressure

- Fluency is routine practice mastery that persists under stress

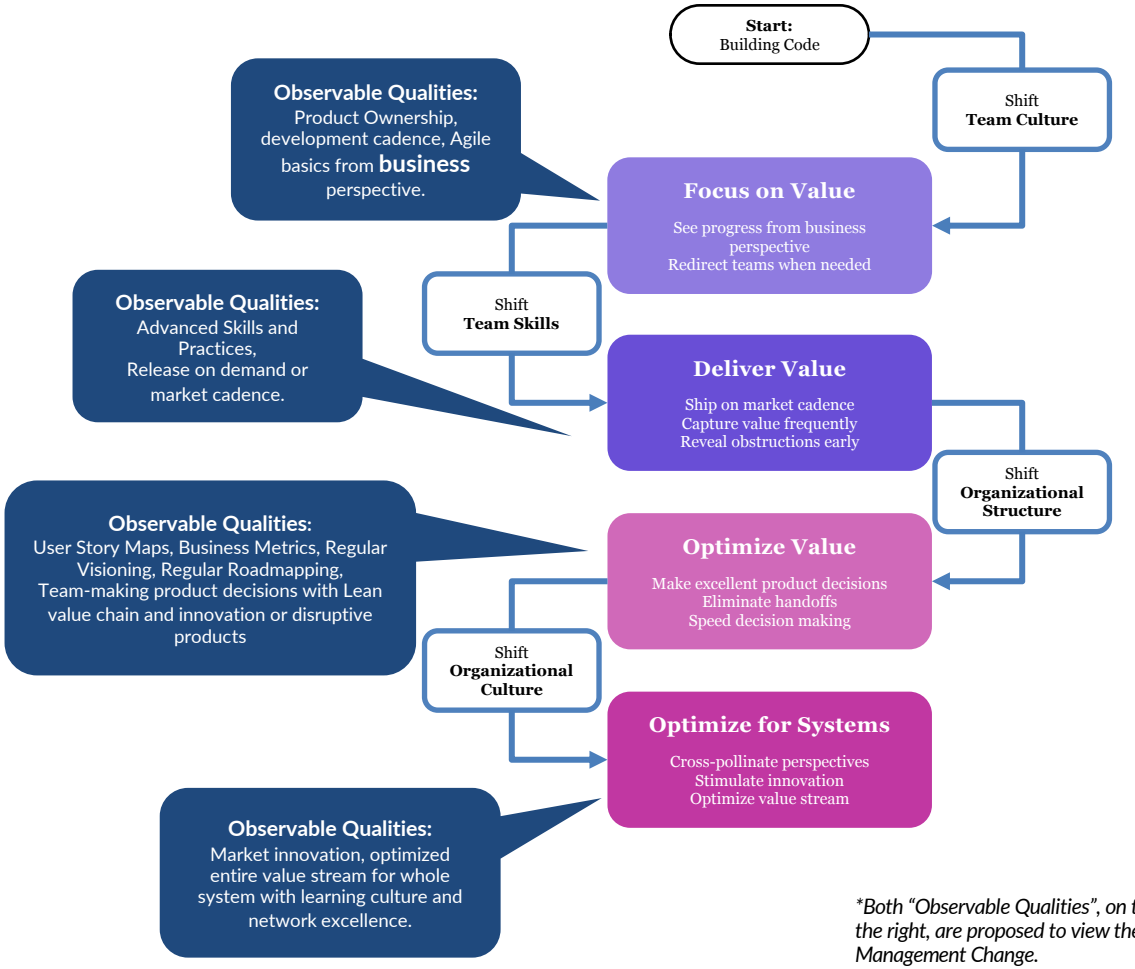
Deliberate, consistent practice

- Deliberate practice involves regularly and consistently practicing a skill with increasing levels of challenge and the intention of mastering that skill

Agile Fluency = Team Fluency



Agility (Fluency) collaboration and management change:



**Both "Observable Qualities", on the left, and Shared Investment, on the right, are proposed to view the organization systemic need for Agile Management Change.*

Transform with Agile organization adoption

Agile change management blends Agile development with organization change management as a repeatable change series

Every enterprise is part of a system. The phases blend as well as influence each other.

Develop	Enable	Adapt	Own
Training	Management	Cross-functional alignment	Key Performance Indicators
Development	Process	Structure	Business Metrics
On-the-job	Workplace	Engineering	Job Family
Support	Logistics	Architecture	Shippable Code
Work structure	Roles	Policy	Cycle Time

Each phase within the focus needs:

1. Principles – immutable
2. Competence/quality – observable, measurable
3. Method – tool/technique, enterprise application, architecture

Systemic change relies on who and what success enablers are for both:

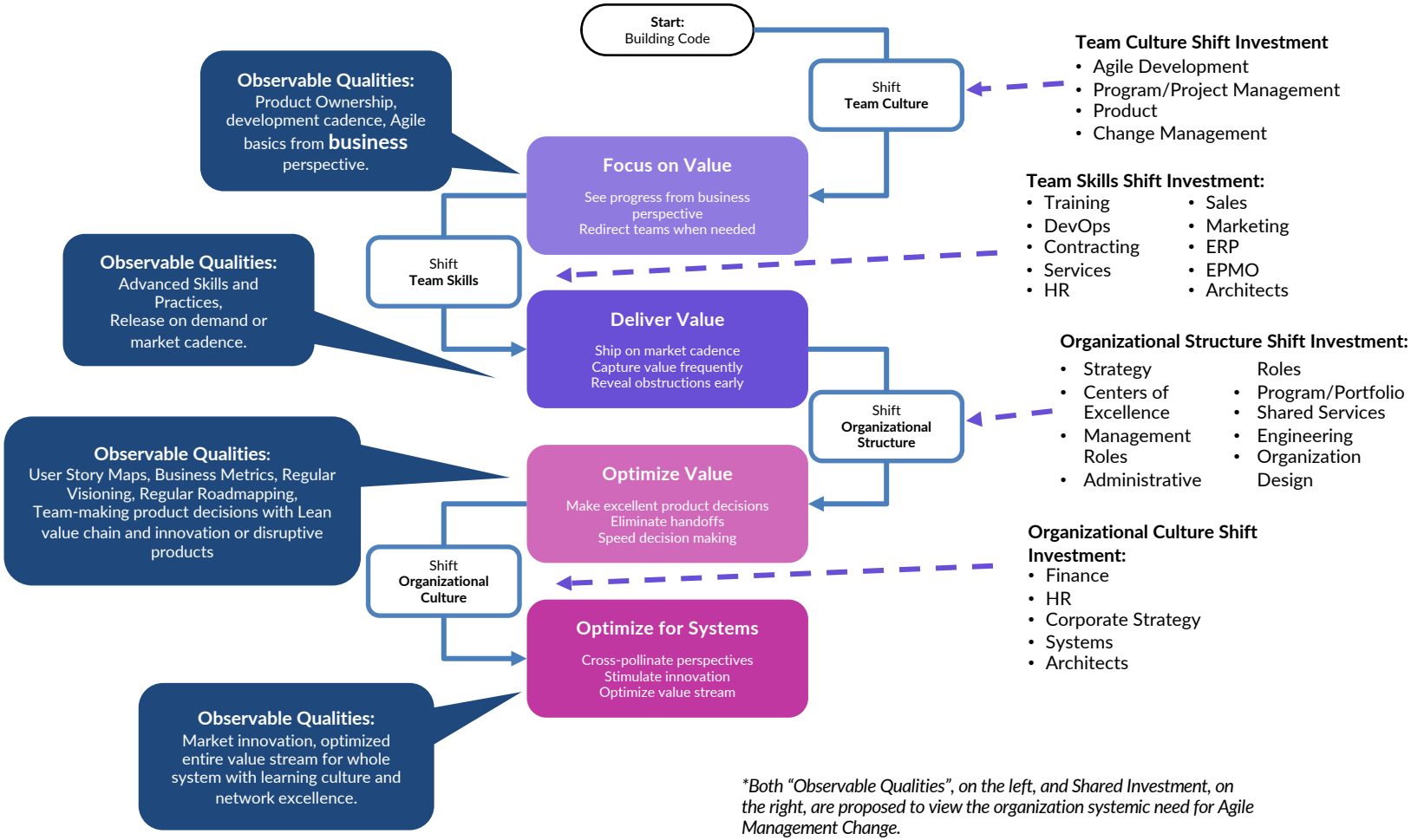
1. Individual and
2. Shared

Agile Change Management

Organization Adoption with Agile Development Principles

1. Agile Fluency Model with Develop > Enable > Adapt > Own
2. Scaled Agile Framework 4.5 models

Agility (Fluency) collaboration and management change:



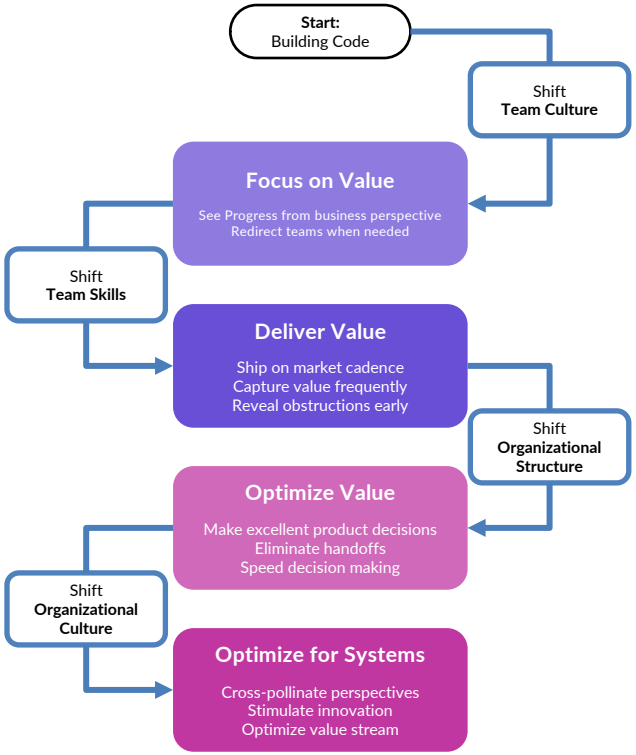
**Both "Observable Qualities", on the left, and Shared Investment, on the right, are proposed to view the organization systemic need for Agile Management Change.*

Agile Change Management

Organization Adoption with Agile Development Principles

1. **Agile Fluency Model with Develop > Enable > Adapt > Own**
2. Scaled Agile Framework 4.5 models

Fluency relies on collaboration



- Adoption and ownership is a source for an Agile transformation environment
- The framework model is not the key
  - Could use any transformation method *this model comes from a recent environment I worked in*
- Agile Fluency* provides a good compliment to SAFe Implementation Roadmap:
  - The Agile Fluency goal is every team works at the level of fluency that best fits their business' needs

**Like many organizations, this represents Fluency essence, but modified in practice. This modification, like SAFe, happens many times and we roll with the options. Still, the Agile Management Change approach works.*

Transform with Agile organization adoption

Agile change management blends Agile development with organization change management as a repeatable change series

Training	Management	Cross-functional Alignment	Key Performance Indicators
Development	Process	Structure	Business Metrics
On-the-job	Workplace	Engineering	Job Family
Support	Logistics	Architecture	Shippable Code
Work structure	Roles	Policy	Cycle Time

Transform with Agile organization adoption

Every enterprise is part of a system. The phases blend as well as influence each other

Training	Management	Cross-functional Alignment	Key Performance Indicators
Development	Process	Structure	Business Metrics
On-the-job	Workplace	Engineering	Job Family
Support	Logistics	Architecture	Shippable Code
Work structure	Roles	Policy	Cycle Time

Each phase within the focus needs:

1. Principles – immutable
2. Practice – method, tool/technique, enterprise application, architecture
3. Competence/quality – observable, measurable

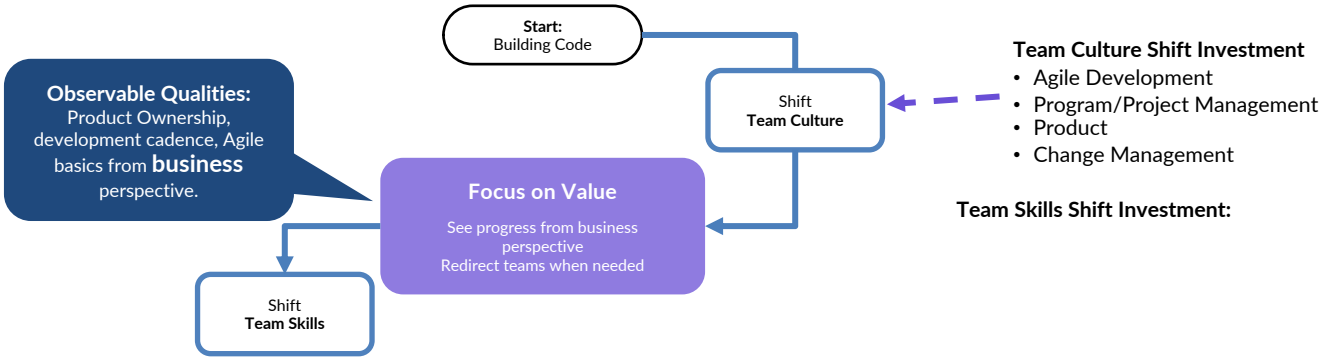
Systemic change relies on who and what success enablers are for both:

1. Individual and
2. Shared

Level 1 fluency relies on: Agile Basics, Product, and Team

Agile transformation: Develop > Enable > Adapt > Own

Sample: **Level 1** fluency relies on: Agile Basics, Product, and Team competence



Agile Basics	Product	Team
Transparent	User Stories	Autonomous
Focus on Customer Value	Backlog Grooming	Have Definition of Done
Trust	Regular Release Planning	Estimation by team
Scrum or Kanban or Scrumban	Regular Sprint Planning	Self Organizing
Transparent	Regular Daily Planning	Cross Functional
	Sprint Review	Retrospectives
	Usable Mission (Sprint/Release Goals)	Co-located

Collaborate on individual and shared success of team

Visualize change with the phase, the success, and the enable requirements

1. Start with needs of each phase:

1. Principles – immutable
2. Competence/quality – observable, measurable
3. Method – tool/technique, enterprise application, architecture

2. Then build success need for:

1. Individual level
2. Shared level

3. Then move to **Enable** and repeat step 1. and step 2. above, **Adapt**, and **Own**

DEVELOP – Agile Basics		
Principles	Competence/quality	Method
•	•	•
•	•	•
•	•	
•		

Team Culture shift to Level 1, Focus on Value: Agile Basics

Sample: Level 1 fluency relies on: Agile Basics, Product, and Team investment

Develop **Enable** **Adapt** **Own**

Each phase within the focus needs:

1. Principles
2. Competence/quality
3. Method (tool)

1. Agile Basics	Who and what is needed to help:	Develop	Enable	Adapt	Own
Transparent	Individual				
	Shared (enterprise, portfolio, program)				
Focus on Customer Value					
Trust					
Scrum or Kanban or Scrumban					

2. Product	Who and what is needed to help:
User Stories	Individual
	Shared (enterprise, portfolio, program)
Backlog Grooming	
Regular Release Planning	
Regular Sprint Planning	
Regular Daily Planning	
Sprint Review	
Usable Mission (Sprint/Release Goals)	

3. Team	Who and what is needed to help:
Autonomous	Individual
	Shared (enterprise, portfolio, program)
Have DOD	
Estimation by team	
Self Organizing	
Cross Functional	
Retrospectives	
Co-located	

Note: This sample is Level 1 Agile Fluency. You can carry forward to Levels 2 through 4, as easily

Team Culture shift to Level 1, Focus on Value: Product

Develop Enable Adapt Own

Each phase within the focus needs:

1. Principles
2. Practice
3. Competence/quality

2. Product	Who and what is needed to help:
User Stories	Individual Shared (enterprise, portfolio, program)
Backlog Grooming	
Regular Release Planning	
Regular Sprint Planning	
Regular Daily Planning	
Sprint Review	
Usable Mission (Sprint/Release Goals)	

1. Agile Basics	Who and what is needed to help:	Develop	En		
Transparent	Individual Shared (enterprise, portfolio, program)				
Focus on Customer Value					
Trust					
Scrum or Kanban or Scrumban					

3. Team	Who and what is needed to help:
Autonomous	Individual Shared (enterprise, portfolio, program)
Have DOD	
Estimation by team	
Self Organizing	
Cross Functional	
Retrospectives	
Co-located	

Team Culture shift to Level 1, Focus on Value: Team

Develop **Enable** **Adapt** **Own**

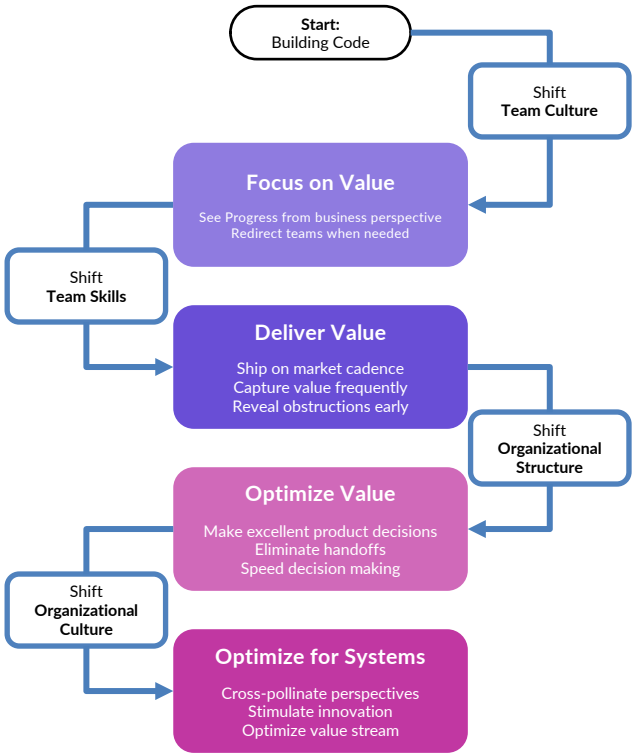
Each phase within the focus needs:

1. Principles
2. Competence/quality
3. Method (tool)

1. Agile Basics	Who and what is needed to help:
Transparent	Individual Shared (enterprise, portfolio, program)
Focus on Customer Value	
Trust	
Scrum or Kanban or Scrumban	

2. Product	
User Stories	
Backlog Grooming	
Regular Release Planning	
Regular Sprint Planning	
Regular Daily Planning	
Sprint Review	
Usable Mission (Sprint/Release Goals)	

3. Team	Who and what is needed to help:
Autonomous	Individual Shared (enterprise, portfolio, program)
Have DOD	
Estimation by team	
Self Organizing	
Cross Functional	
Retrospectives	
Co-located	

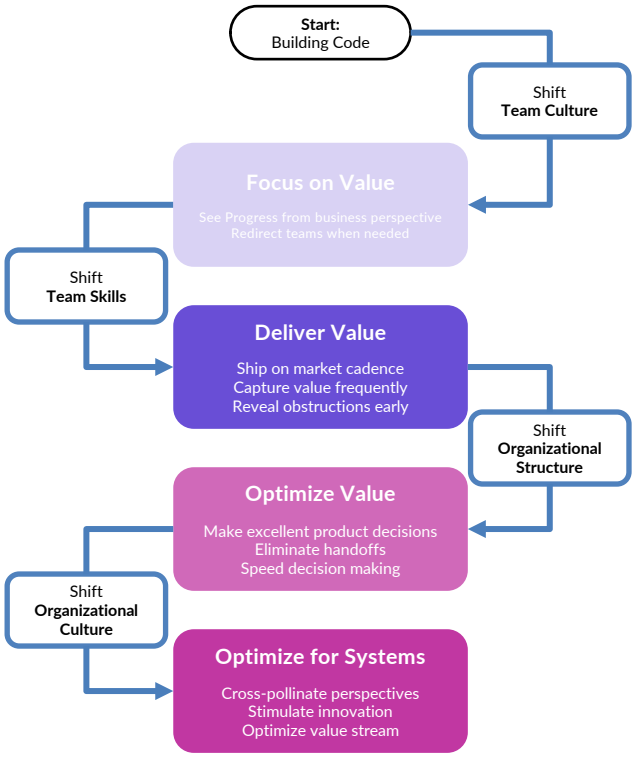


Agile change is frame for each successive level

Any change would visualize phase, the success, and the enable requirements

Develop **Enable** **Adapt** **Own**

1. Start with needs of each phase
2. Then build success need for both individual and shared levels
3. Repeat for scale enablement framework



[*... map any set of needs against ...*]		
People	Process	Practice
•	•	•
•	•	•
•	•	•
•		

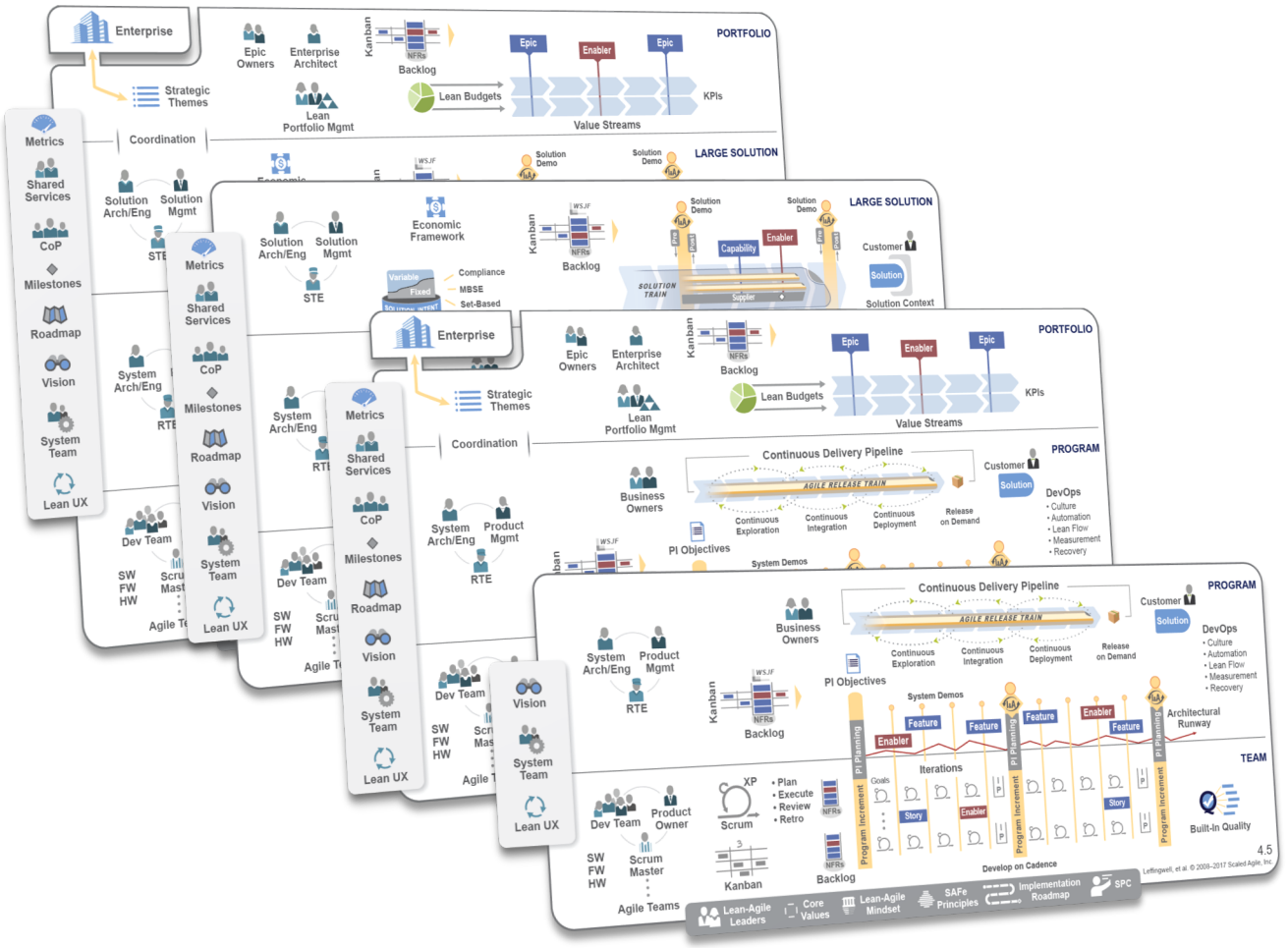
Agile Change Management

Organization Adoption with Agile Development Principles

1. Agile Fluency Model with Develop > Enable > Adapt > Own
2. **Scaled Agile Framework 4.5 models**

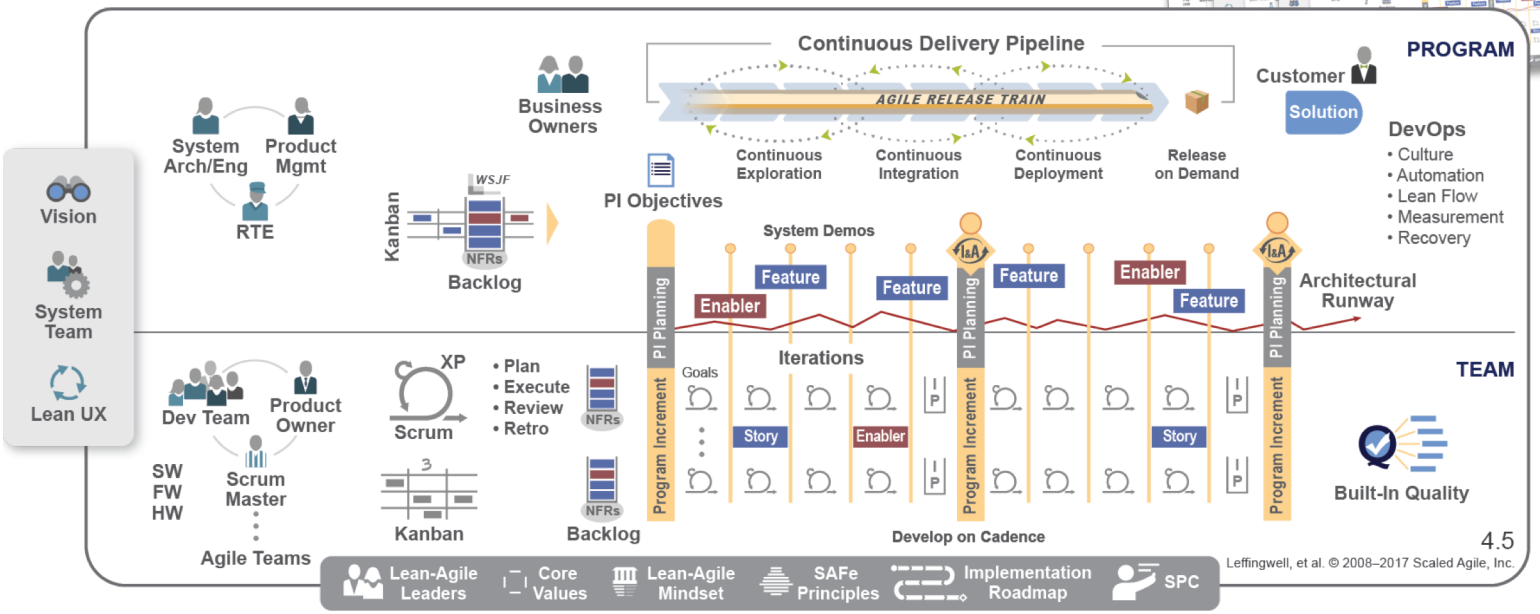
Scaled Agile relies on whole system change

SAFe 4.5 reveals Agile ownership at all levels with accountability to plan for development team excellence.



Essential SAFe AND systems change

- With SAFe 4.5 to align ownership at levels senior to development
- This senior transparency in the next four slides reveals dedication to the Agile system/network



SAFe reflects the latest in Lean-Agile thinking, more visibly incorporating scalable DevOps and the Continuous Delivery Pipeline. It demonstrates advancements in configurability, implementation guidance, and enhanced capabilities for improving the user experience and accelerating time-to-market.

Read more at: http://www.scaledagileframework.com/#

Portfolio SAFe AND systems change

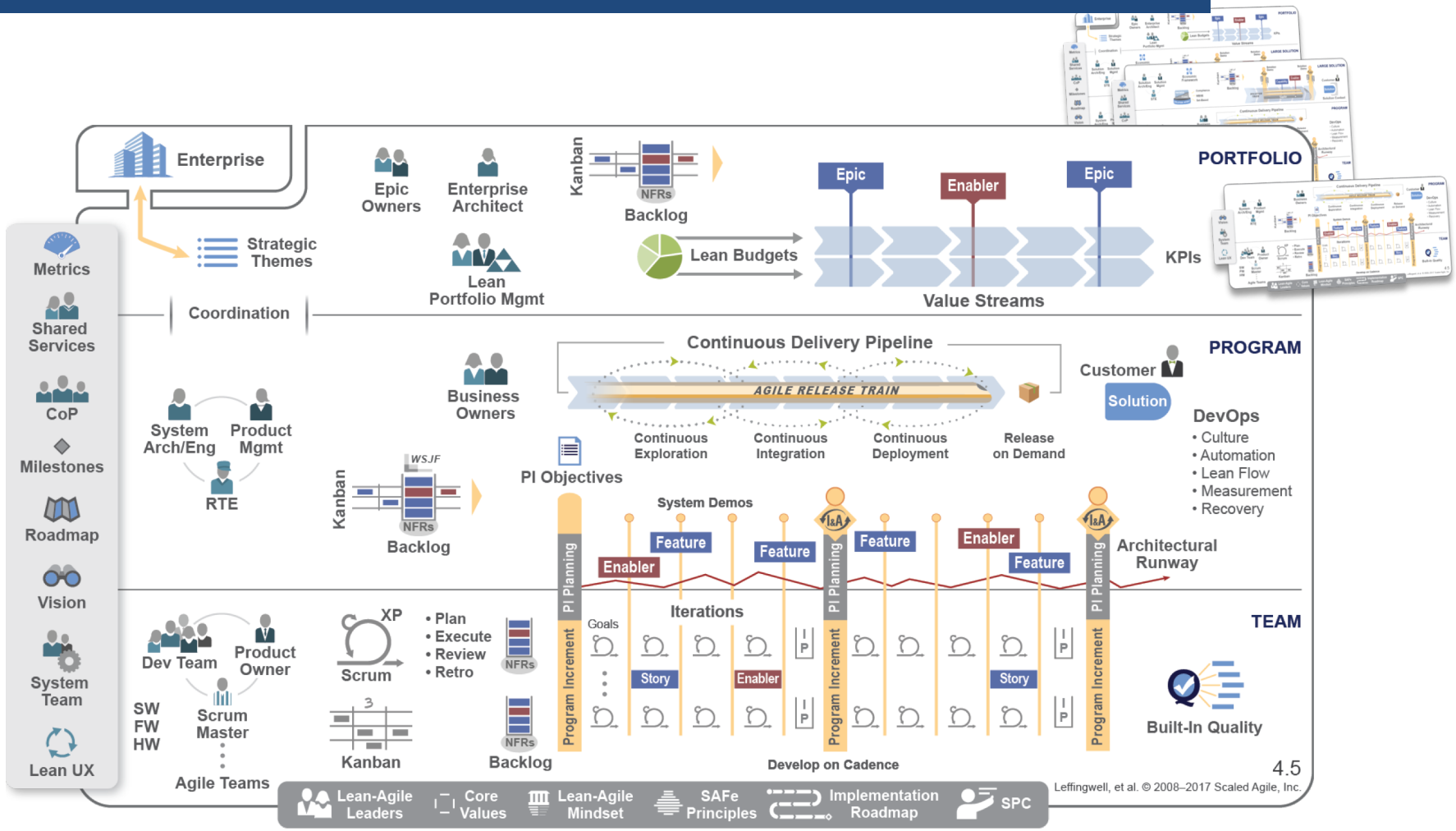


Image source: http://www.scaledagileframework.com

Large solution SAFe AND systems change

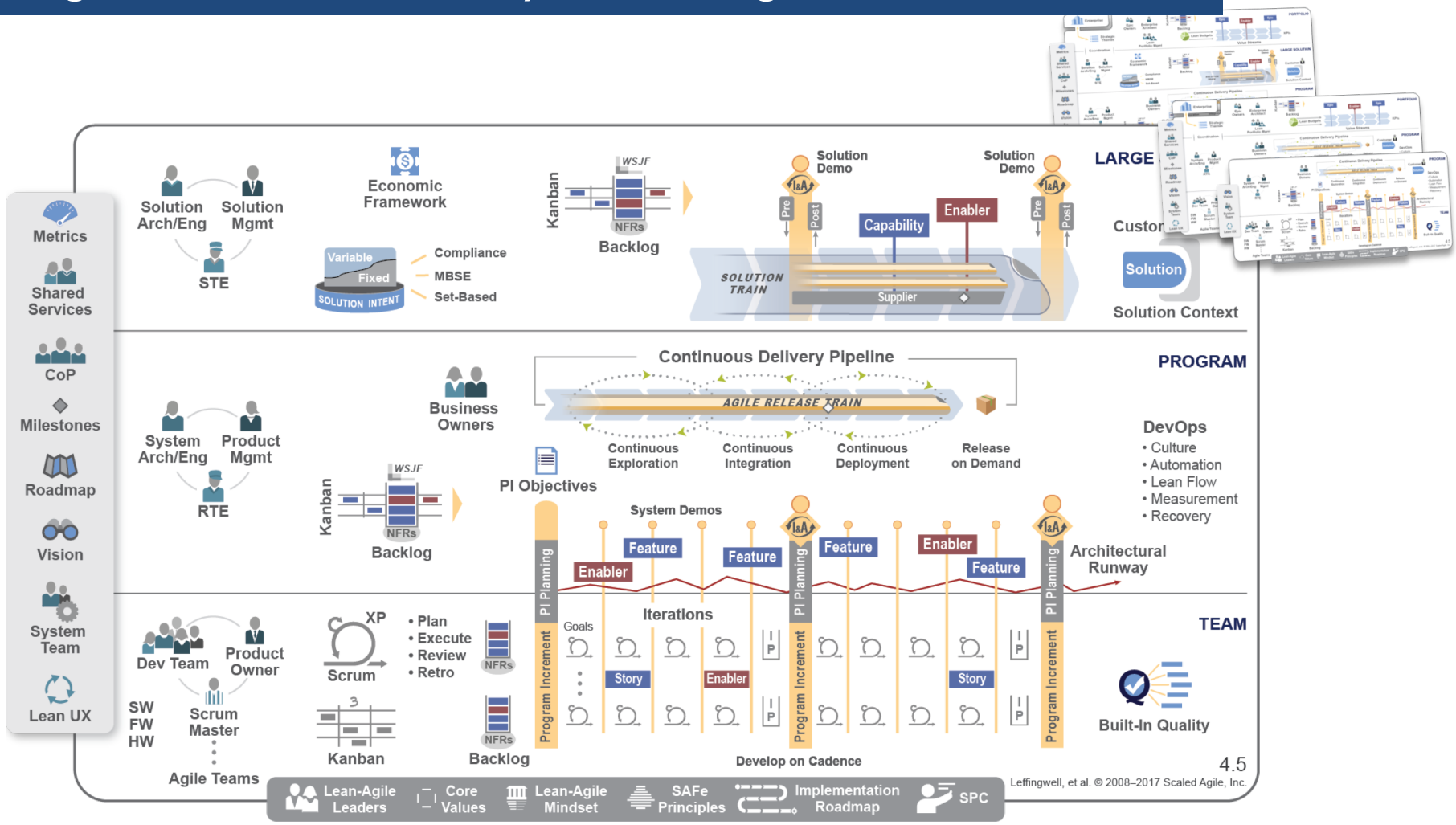


Image source: http://www.scaledagileframework.com

Full SAFe AND systems change

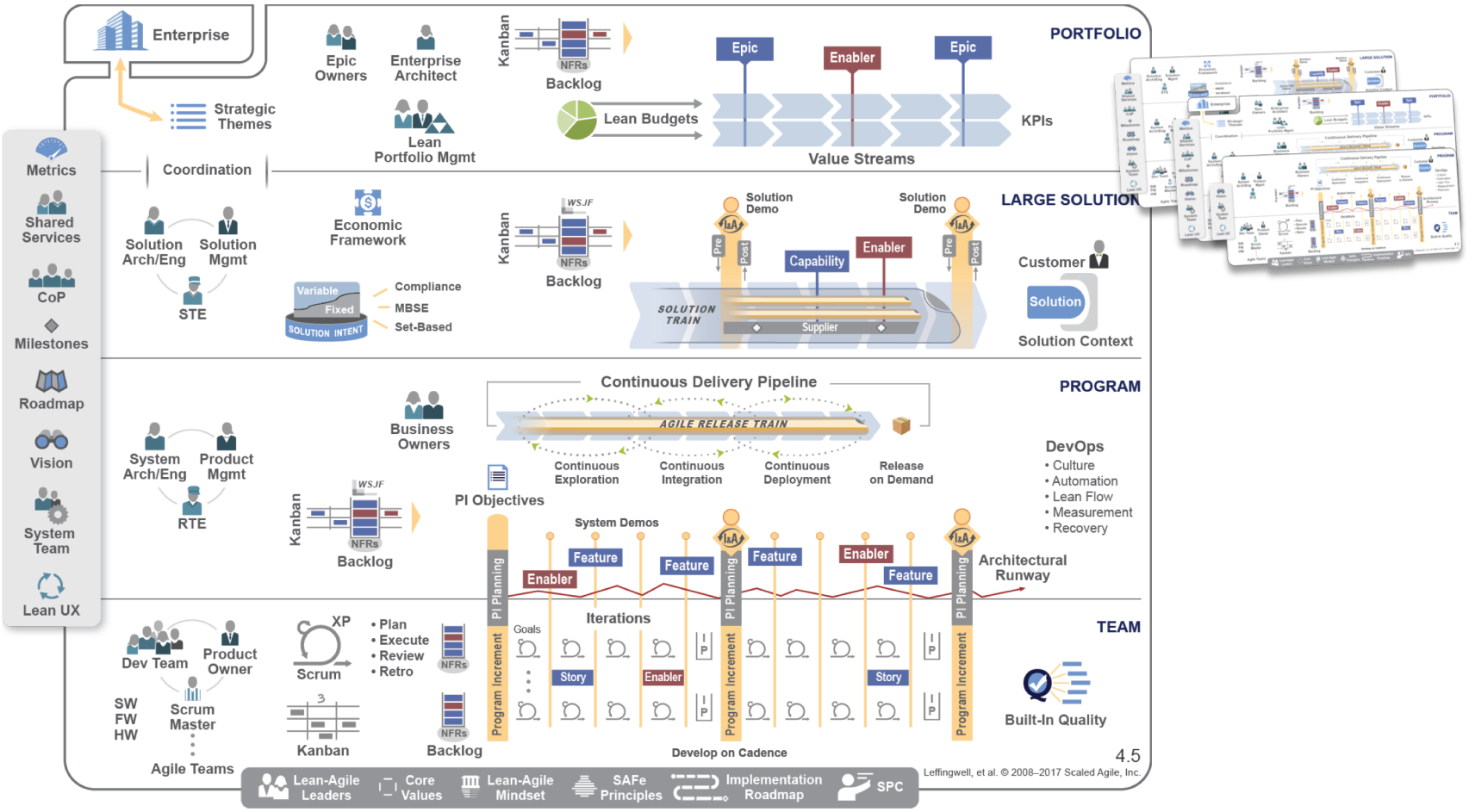


Image source: http://www.scaledagileframework.com

Agile management change builds bridges

Focus on how to enable development teams to do Agile

Image source: http://www.comicbox.eu/

- Mind the gap from learning to doing
  - ❑ Leverage Lean Six Sigma with executives and managers to relate to Agile
  - ❑ Leverage Project Management Institute commitment to Agile iterative project methods
- Seeing is deceiving, but doing is believing
  - ❑ Get teams to focus on practical, not theoretical
- People and teams do not learn at the same pace
  - ❑ Adults learn differently, Agile progress never advocates team comparison as competence, complexity, constraints, and communication are never the same
- Commit to Agile enablers
  - ❑ Executive and management input is required to set Agile development teams up for success
- Rely on Agile learning to learn fast while you inspect and adapt
  - ❑ Avoid one-size-fits all methods and templates, allow customization within Agile principles

Thank You

Further Learning and Development resources available

email@TobyElwin.com

https://TobyElwin.com

@TobyElwin on Twitter