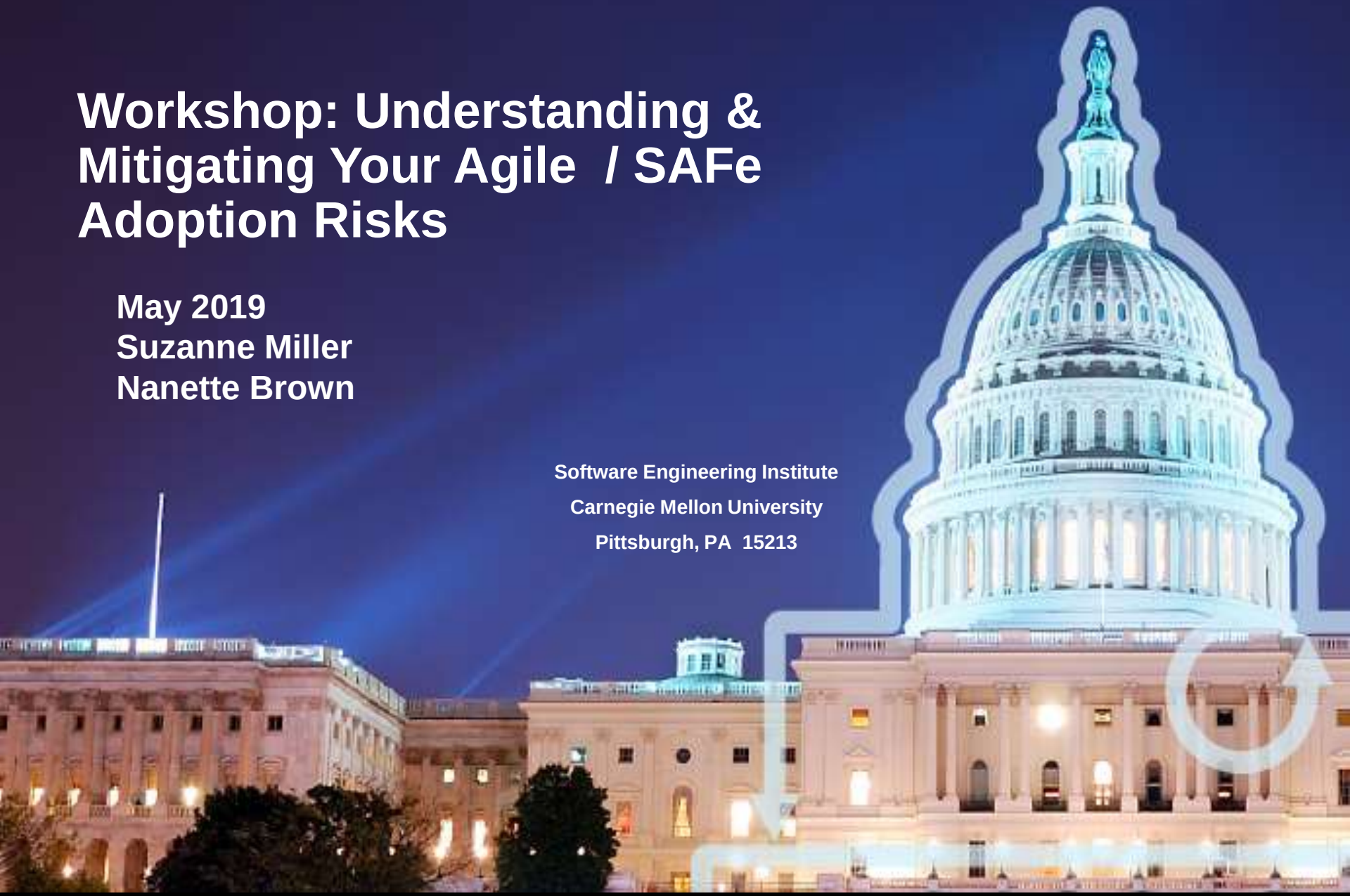
Workshop: Understanding & Mitigating Your Agile / SAFe Adoption Risks

May 2019

Suzanne Miller
Nanette Brown

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-0430

Workshop Topics

- Workshop Expectations
- Key Concepts of Agile
- Scaling up Agile
- Key Concepts of SAFe
- AMPD Infrastructure & BMP ART Overview
- Key Concepts of Technology Transition
- Key Concepts of the SEI's Readiness and Fitness Analysis (RFA)
- Identifying Agile / SAFe Adoption Risks with RFA
- Summary/Next Steps

Exercise: Wants & Offers

What outcomes would make this workshop a successful one for you? *(What do you WANT?)*

What skills/abilities/knowledge/attitudes are you willing to contribute to help achieve what you want? *(What do you OFFER?)*

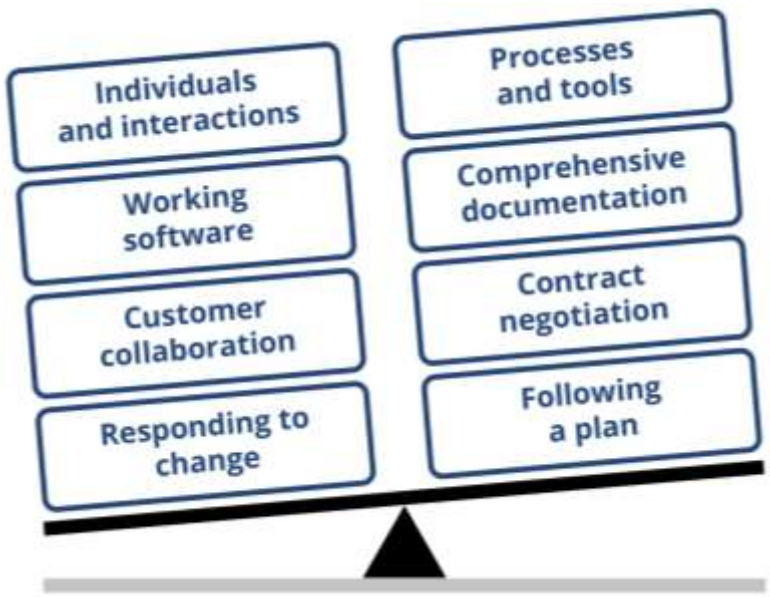
Write down one per sticky note (3-4 max) and post them on the ***WANTS/OFFERS*** flip chart.

The facilitator will group them and comment on which are achievable.

Key Concepts of “Agile”

Agile Manifesto

Through this work we have come to value:



That is, while there is value in the items on the right, we value the items on the left more.

"The Agile movement is not anti-methodology. In fact, many of us want to restore credibility to the word methodology. We want to restore a balance."

Jim Highsmith – Short Post on the History of the Agile Manifesto

http://www.agilemanifesto.org/

Agile Principles-1

1. Highest priority is satisfy the customer through early and continuous delivery of software.
2. Welcome changing requirements, even late in development…
3. Deliver working software frequently, from a couple of weeks to a couple of months...
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Provide environment and support they need…
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Agile Principles – 2

7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development…a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Adapted from http://agilemanifesto.org/principles.html

Working Definition of Agile

Agile An *iterative* and *incremental* (evolutionary) approach to software development which is performed in a *highly collaborative manner* by *self-organizing teams* within an *effective governance framework* with *"just enough" ceremony* that produces *high quality software* in a *cost effective and timely* manner which *meets the changing needs of its stakeholders*. [Ambler 2013]

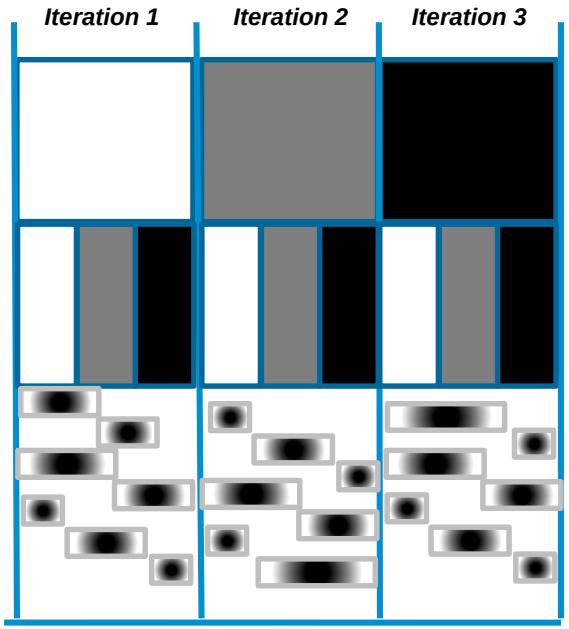
[Ambler 2013] Ambler, Scott. *Disciplined Agile Software Development: Definition*. http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm

Taking an Iterative Approach

Single batch – one process step per iteration

Multiple batches - complete each batch at the end of an iteration; siloed process steps within each iteration

Multiple batches - decompose each batch into small packages, with multiple start-to-finish cycles in each iteration



Scaling up Agile

Scaling Up by Adding Teams

Agile Principles were Designed for Small Teams (7 + or – 2)

You can only get so much done with 5 to 9 people.

In Agile the idea of small, tightly communicating, self-managing teams is seminal.

The Agile team is the basic building block.

In Agile, we don't scale up by adding people to the teams.

We scale up by adding teams.

Scaling Issues

But scaling up by adding teams causes other issues:

- How to divide up the work?
- How to keep everyone informed?
- How to synchronize releases?
- How to manage dependencies?
- How to incorporate system level attributes and specialty disciplines that cut across teams?

Addressing these issues is the focus of SAFe.

Agile is a Team Approach

Agile can't succeed in a vacuum. Different roles to play by:

- **Developers**
- **Testers**
- **End Users**
- **Customer Representative**
- **Subject Matter Experts**
- **Program Office**
- **Contracts**
- **Finance**
- **Certifiers**
- **….**

Key Concepts of SAFe

SAFe 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

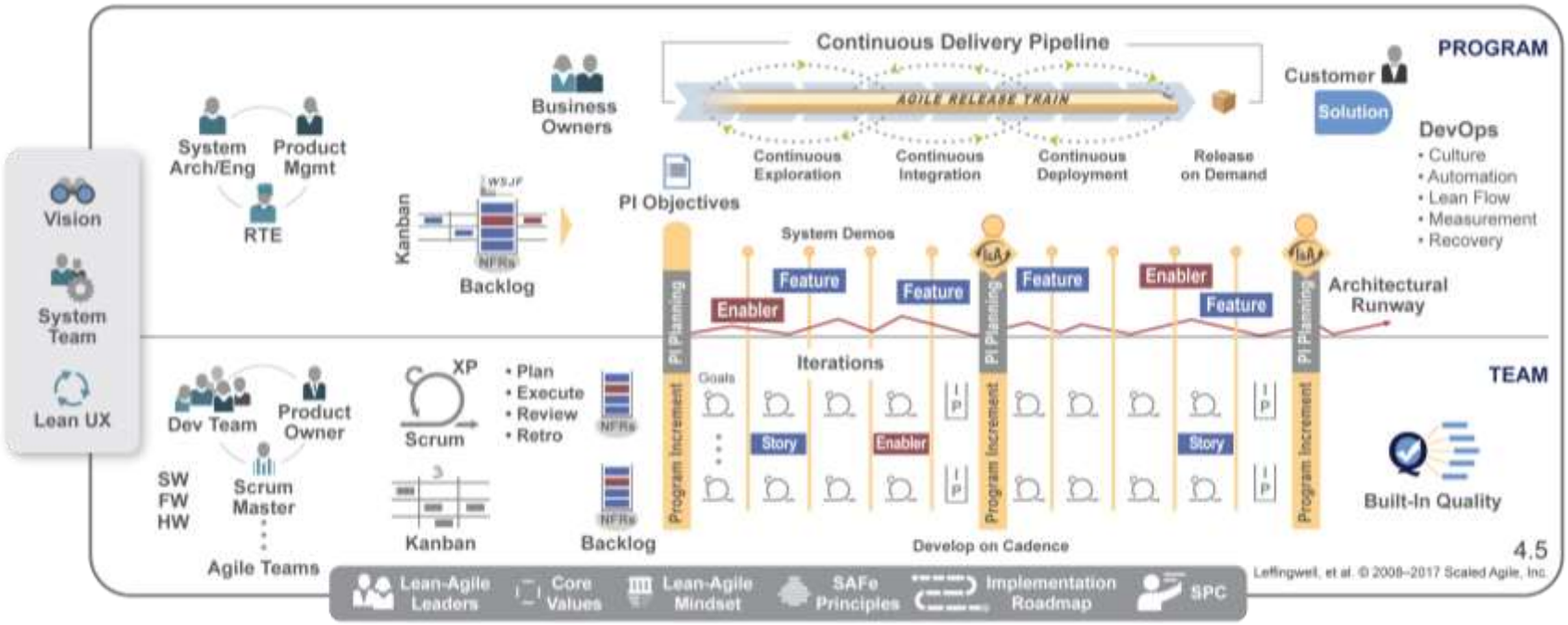
#6 - Visualize and limit WIP, reduce batch sizes, and manage queue lengths

#7 - Apply cadence, synchronize with cross-domain planning

#8 - Unlock the intrinsic motivation of knowledge workers

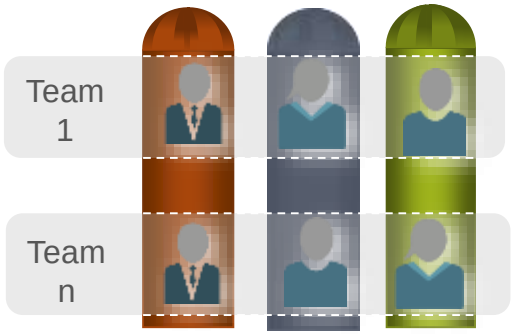
#9 - Decentralize decision-making

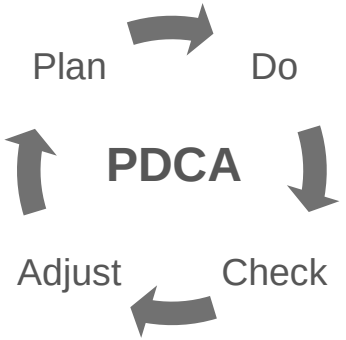
Essential SAFe provides the basis for success



Nothing beats an Agile Team

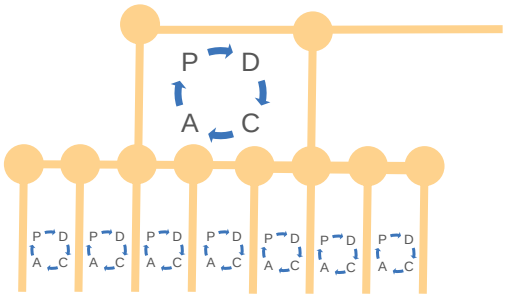
- Cross-functional, self-organizing entities that can define, build and test a thing of value
- Applies basic scientific practice: Plan—Do—Check—Adjust
- Delivers value every two weeks

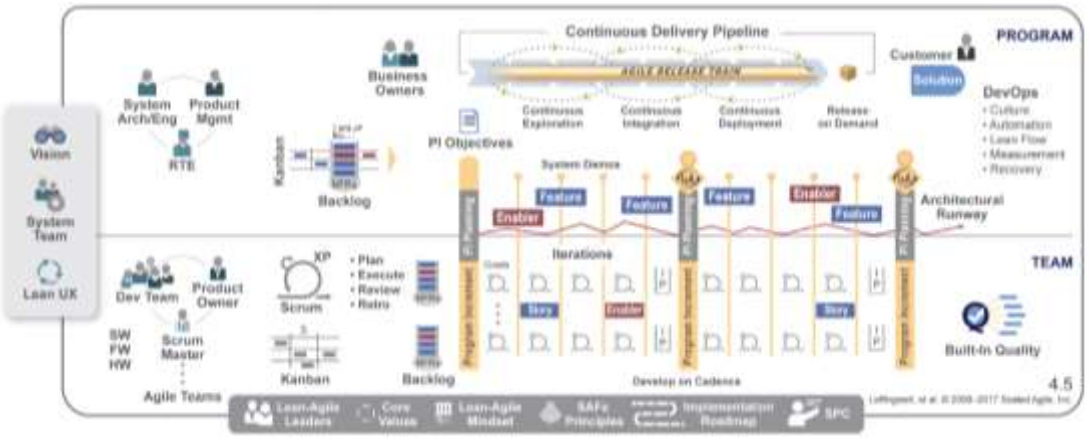




Except a team of Agile Teams

- Align 50-125 practitioners to a common mission
- Apply cadence and synchronization, Program Increments every 6-12 weeks
- Provide Vision, Roadmap, architectural guidance



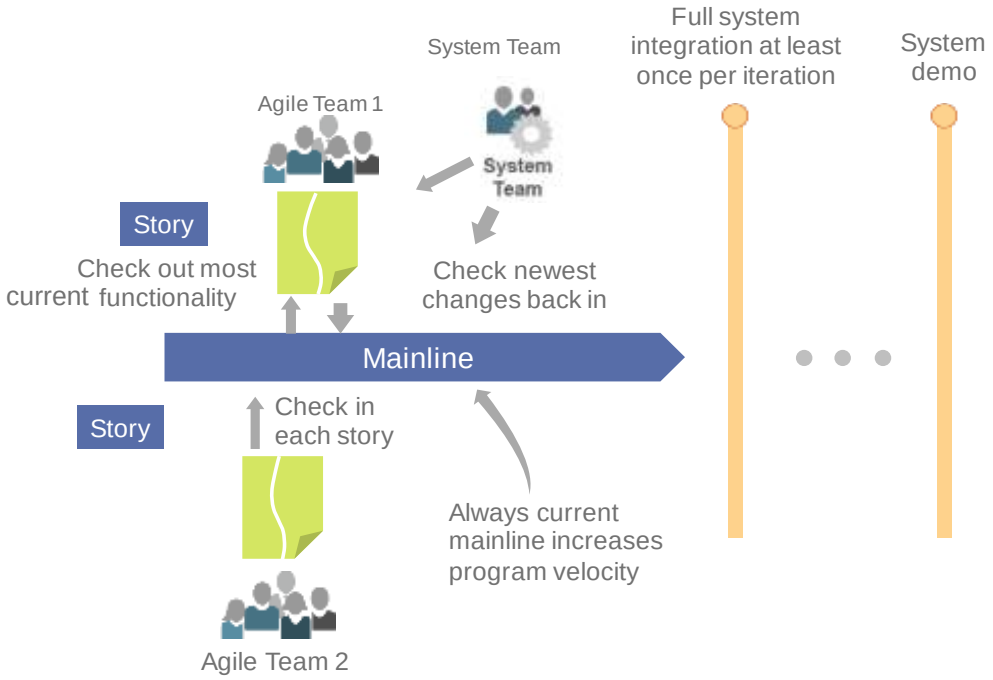


That integrates frequently

Integration points control product development.

— Dantar Oosterwal, The Lean Machine

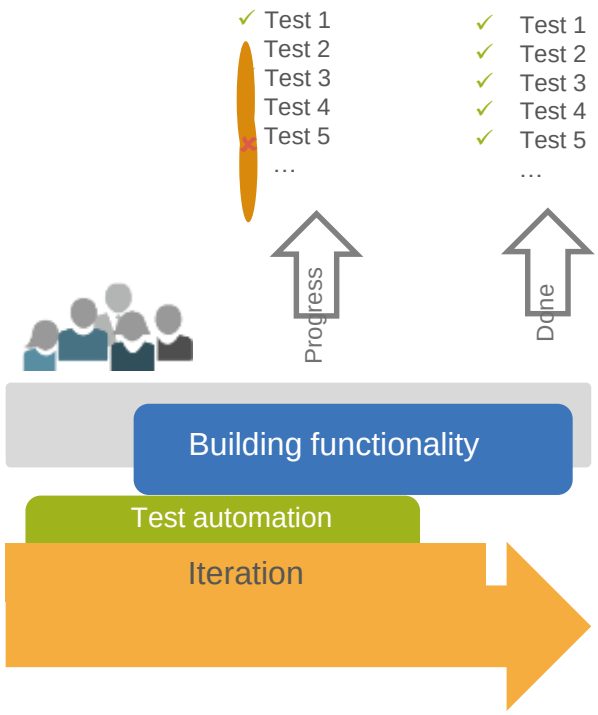
- Avoid physical branching for software
- Frequently integrate hardware branches
- Use development by intention in for inter-team dependencies



Applies test automation

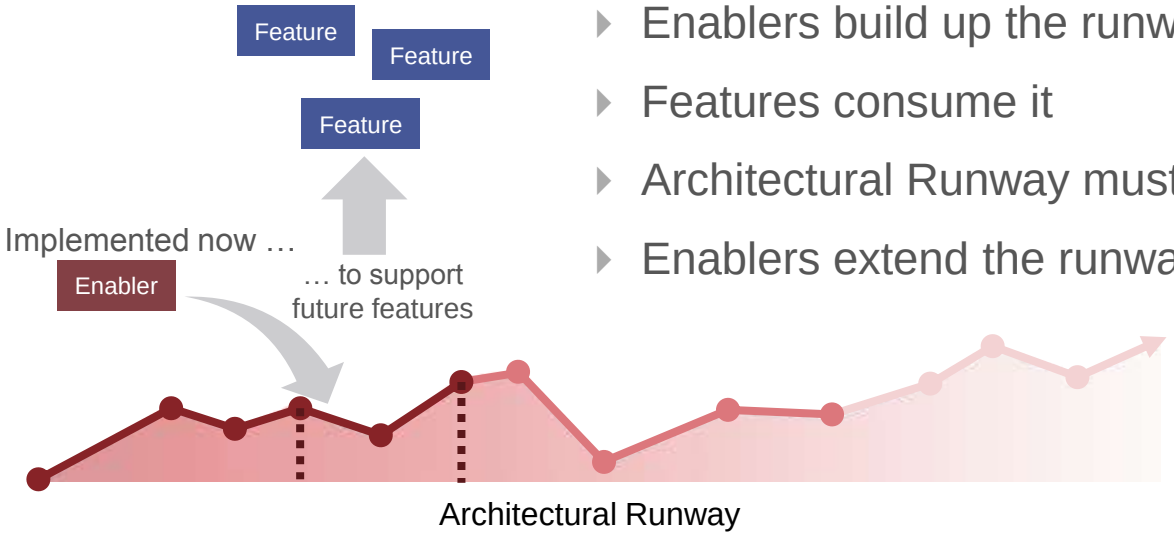
Test automation supports rapid regression testing

- Implemented in the same iteration
- Maintained under version control
- Passing vs. not-yet-passing and broken automated tests are the *real* iteration progress indicator



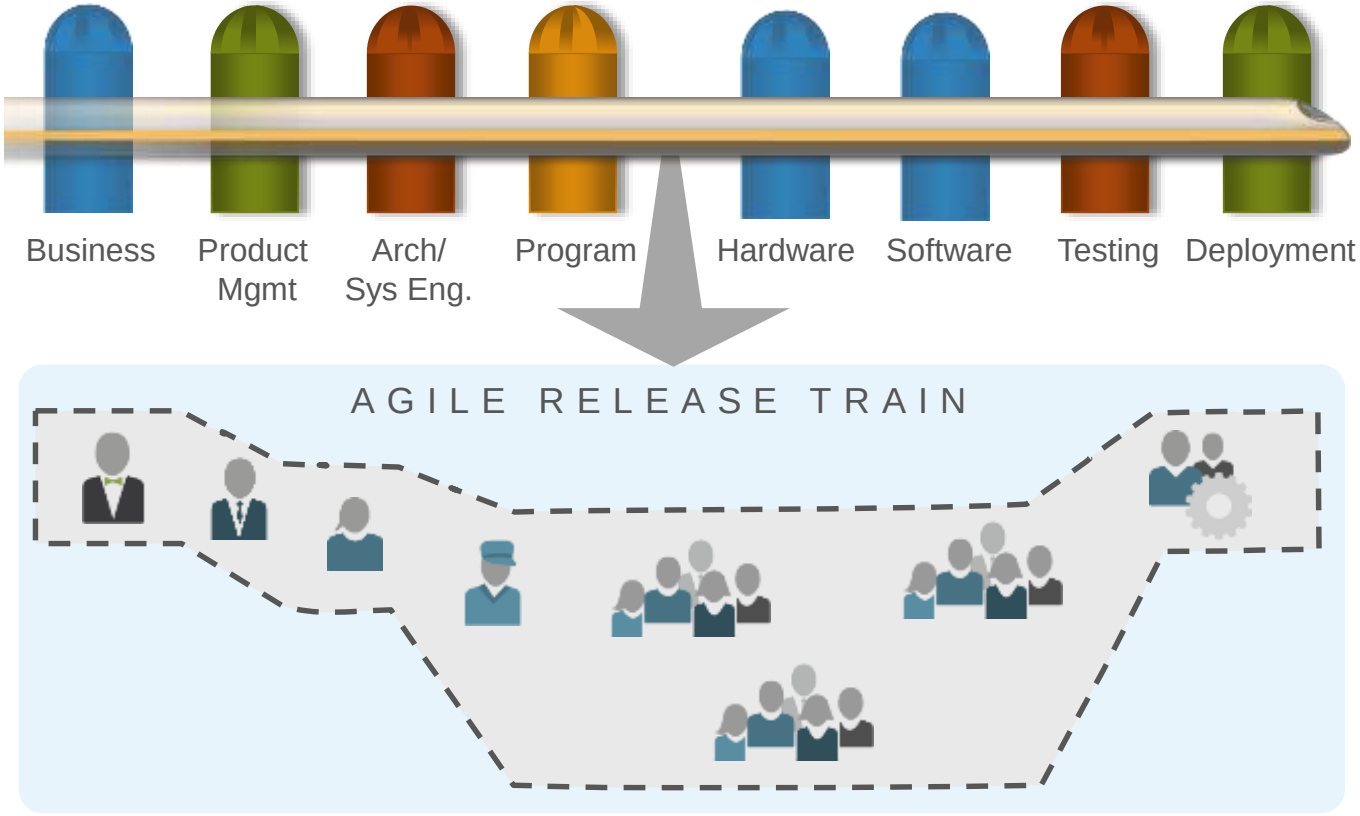
With some Architectural Runway

Architectural Runway—existing code, hardware components, etc. that technically enable near-term business features



- Enablers build up the runway
- Features consume it
- Architectural Runway must be continuously maintained
- Enablers extend the runway

Bringing together the necessary people



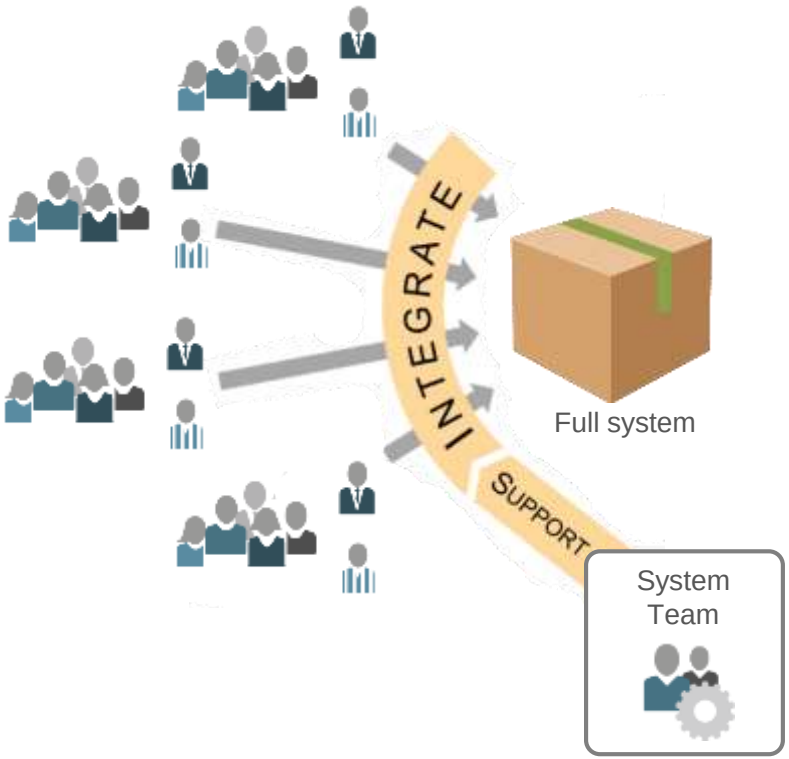
Synchronizes with PI Planning

Future product development tasks can't be pre-determined. Distribute planning and control to those who can understand and react to the end results. — Michael Kennedy, Product Development for the Lean Enterprise

- All stakeholders face-to-face (but typically multiple locations)
- Management sets the mission, with minimum possible constraints
- Requirements and design emerge
- Important stakeholder decisions are accelerated
- Teams create—and take responsibility for—plans

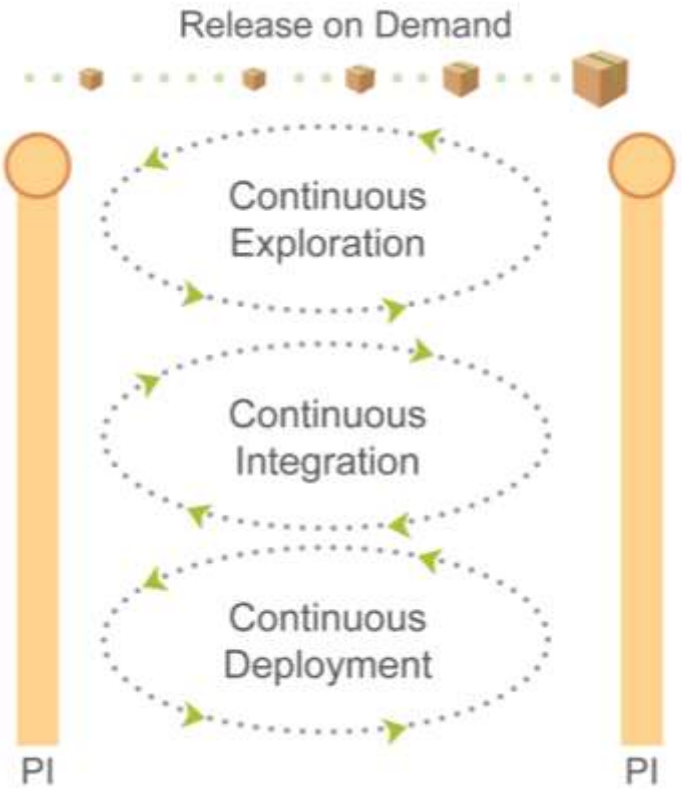
For a short video PI planning example, see: https://youtu.be/ZZAtl7nAB1M

Demonstrates the full system every two weeks



- An integrated solution demo
- Objective milestone
- Demo from the staging environment, or the nearest proxy

Continuously delivers value to customers with DevOps



Inspects and Adapts every PI

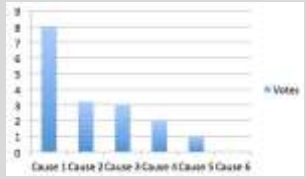
Every PI, teams systematically address the larger impediments that are limiting velocity.

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

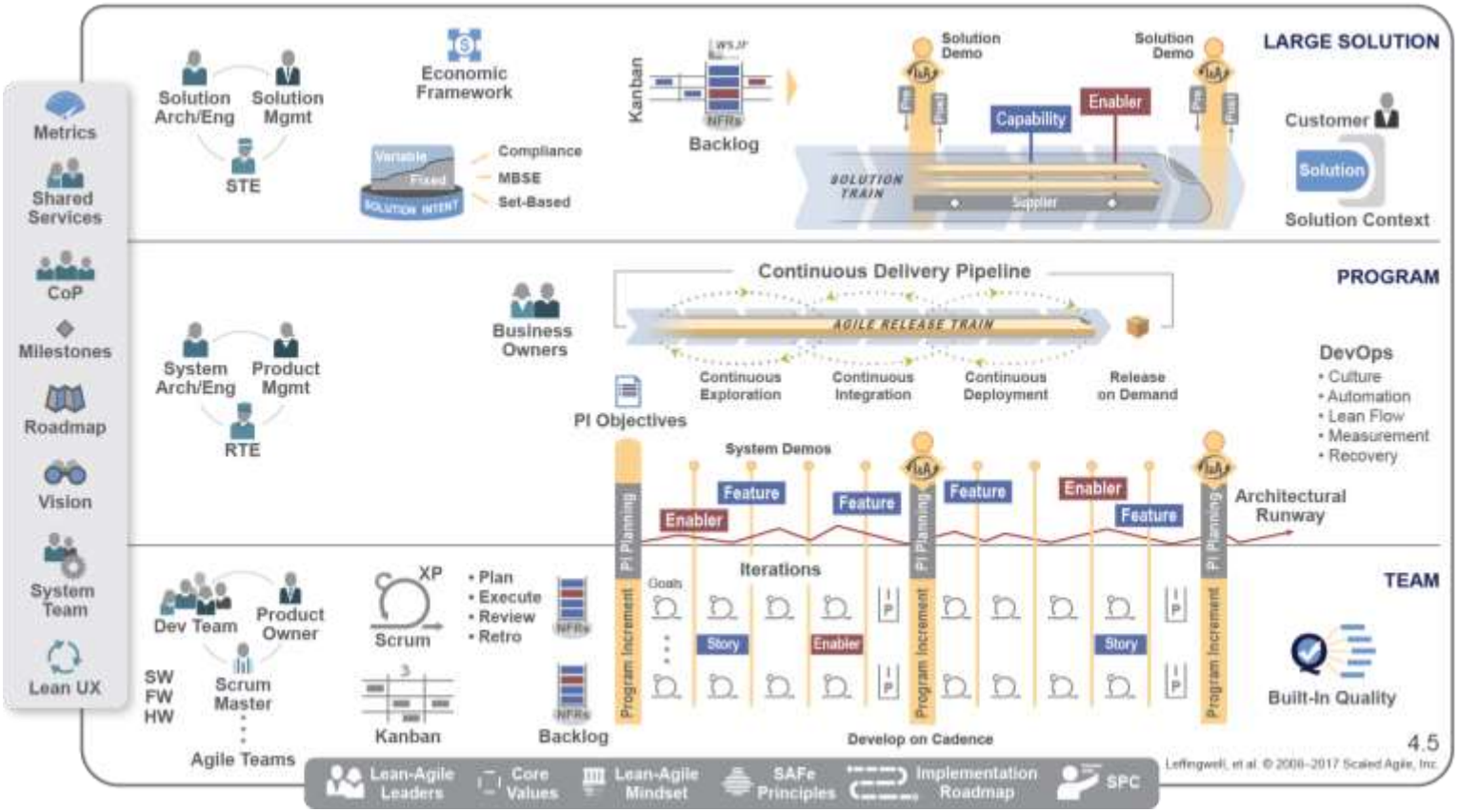
Identify improvement Backlog items

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

Scaled Agile Inc: Essential SAFe Toolkit v4.5.0

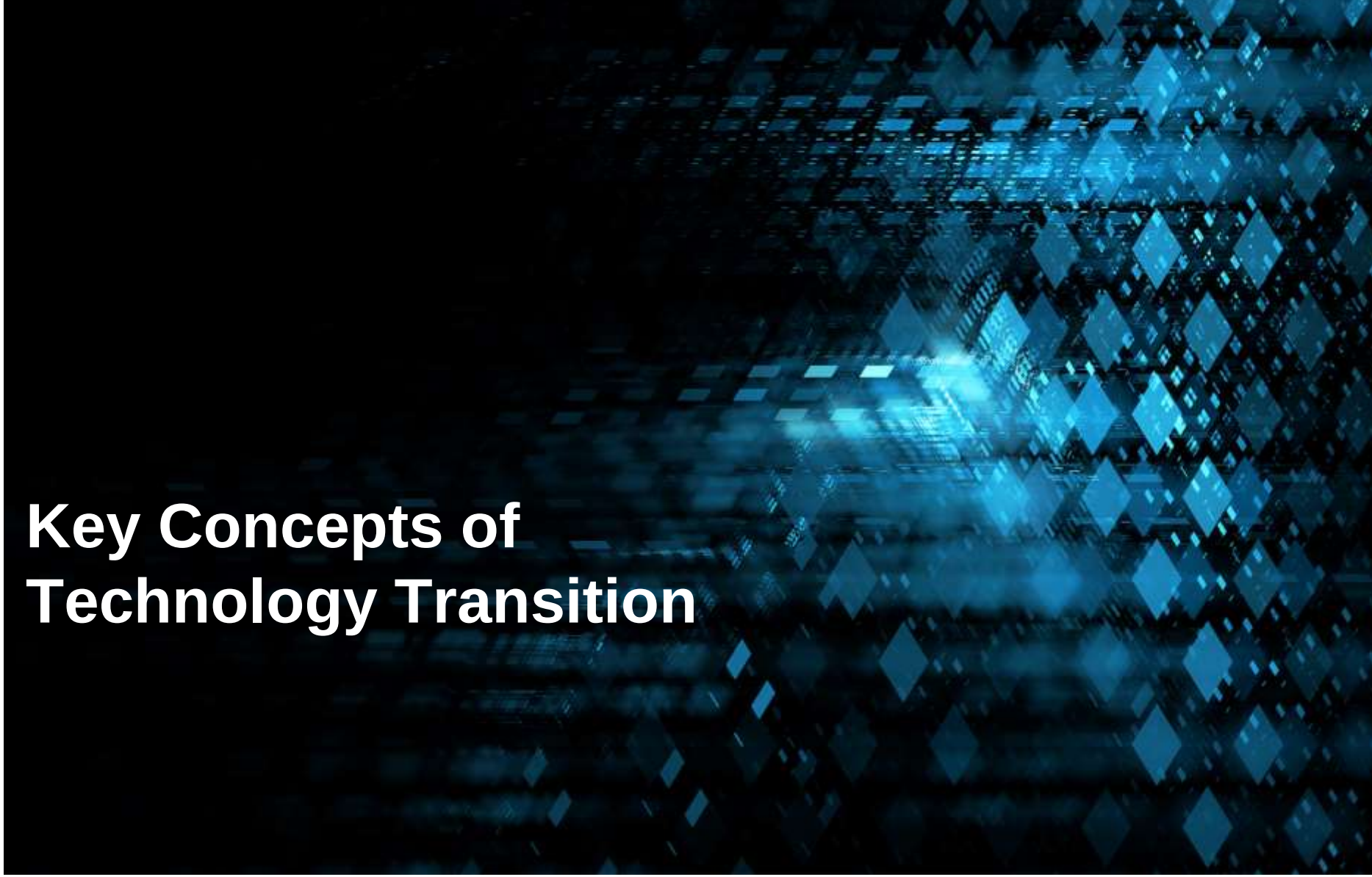
Large Solution SAFe coordinates ARTs with a Solution Train



AMPD Infrastructure & BMP ART Overview

Overview on AMPD Infrastructure and BMP ART

To be presented by AMPD Infrastructure & BMP ART Leadership



"Radical innovation is the process of introducing something that is new to the organization and that requires the development of completely new routines, usually with modifications in the normative beliefs and value systems of organization members."

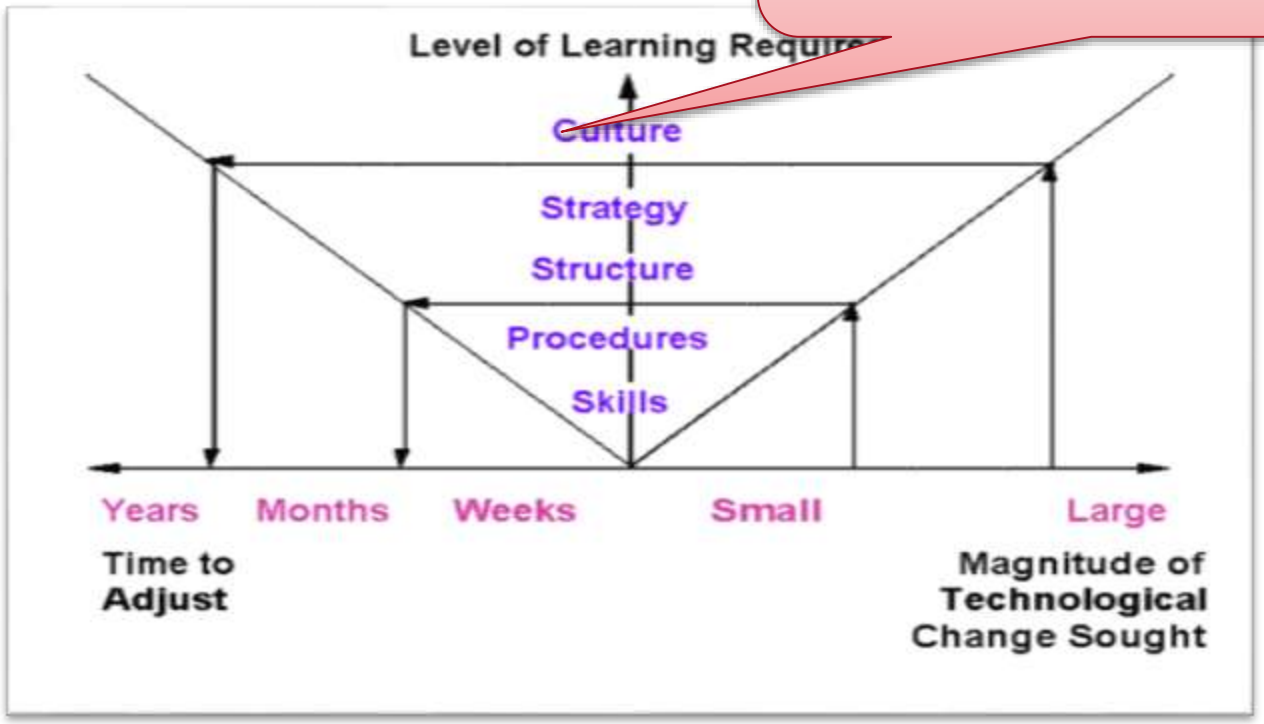
- Nord and Tucker, Routine and Radical Innovations, 1987

Observations from Technology Transitions SEI Has Witnessed

Multi-dimensional	Challenging to Duplicate Success	Multi-Audience	Non-linear
• Multiple dimensions have to be addressed simultaneously to achieve success, not just the technology content	• Most organizations are very poor at transferring what they've learned from one technology transition effort to another	• Different audiences respond differently as they are introduced to the technology	• Acceptance of a new technology does not happen in a linear, predictable fashion

Adler model for Practice/Technology Change Magnitude

Agile and SAFe adoption (especially at scale) reflect a cultural shift. That means all the levels below culture will have changes to make.



From Adler, Paul. "Adapting Your Technological Base: The Organizational Challenge", Sloan Mgmt Review, 1990.

Note: This model has been successfully used for both "hard" technology changes and "soft" (e.g. practices and methodologies) technologies for over 25 years.

Key Concepts of the SEI's Readiness & Fit Analysis (RFA)

Readiness and Fit Analysis (RFA):
An Approach to Understanding Adoption Risks

The purpose of RFA is to support *adoption*, not just installation, of a product/technology

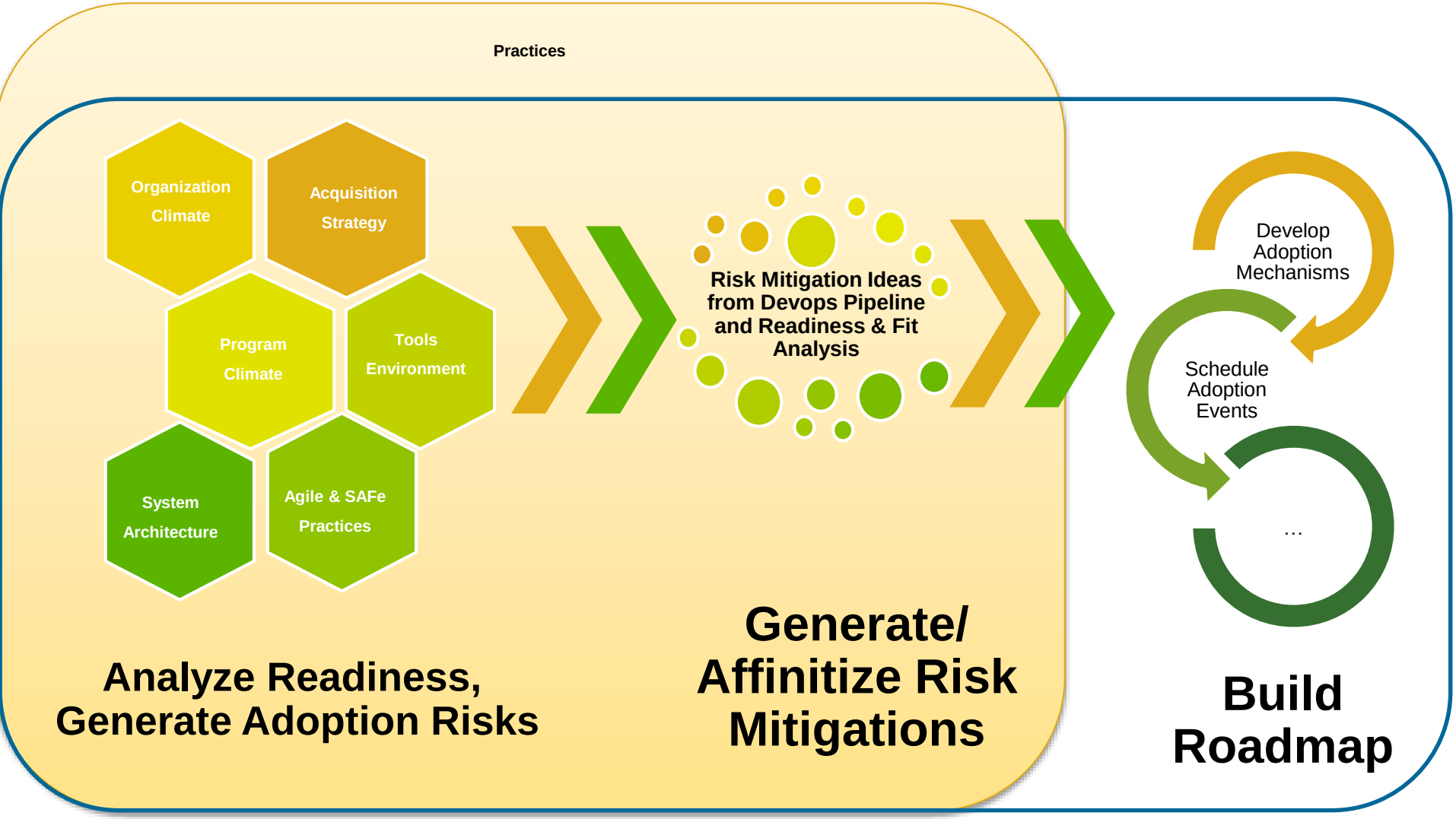
It helps by supporting identification of adoption risks and assisting in developing mitigation strategies for them

RFA Objectives

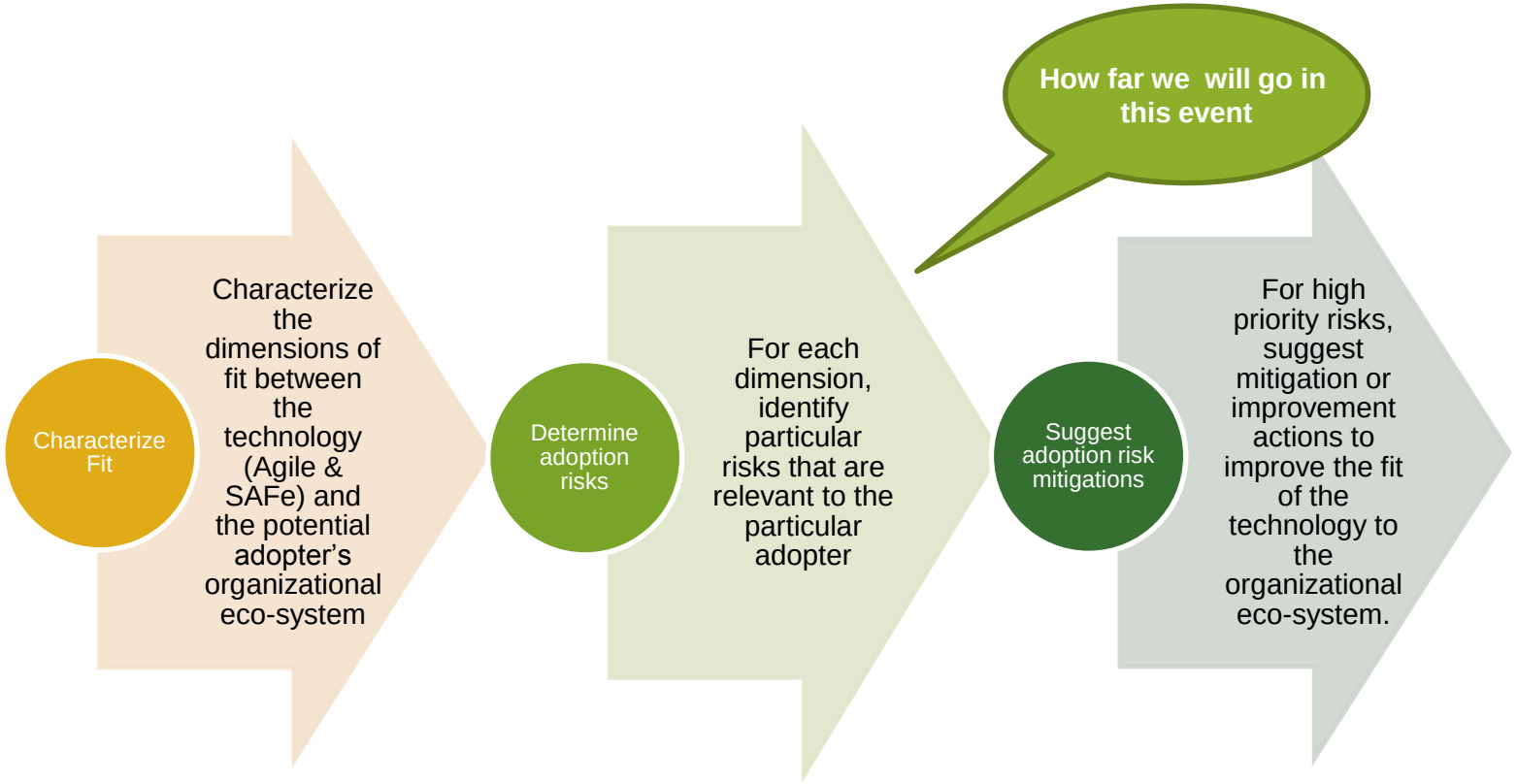
Evaluate the transition fit between the proposed technology and the specified organization.

Specify adaptations required to improve the fit to the point where the transition can be managed.

One Page View of Adoption Planning Process



General RFA Approach



Dimensions of Fit for Agile / SAFe Adoption

- Acquisition Strategy
- Organizational Climate
- Program Climate
- System Architecture
- Tools Environment
- Agile and SAFe Practices

RFA Exercise:

Identifying Agile / SAFe Adoption Risks for AMPD

For each of the RFA Dimensions

- Identify (with the help of SEI-provided “joggers”)
  - **Risks** (things that might happen) to adoption
  - **Issues** (things that are happening now) related to adoption
  - **Enablers** that could facilitate or support adoption
- Create 1 sticky note for each Risk, Issue or Enabler
  - Label with “R”, “I” or “E”
  - A useful form to express risks is *“Given that (condition), there is a risk that (consequence)”*
- Approximately 10 minutes per dimension, after which stickies will be collected and we’ll move on to the next
- “Things to think about” are statements expressed as though the Program is successfully executing in an Agile eco-system

Things to Think About for: *Acquisition Strategy Fit*

- Mechanisms are in place in the contract and acquisition strategy to support the use of agile / lean / SAFe practices
- Mechanisms are in place in the contract and acquisition strategy to allow close collaboration between organic developers, vendor developers and end users
- Mechanisms are in place in the contract and acquisition strategy that allow for interim demonstration and delivery between official releases.
- Acquisition and contracting mechanisms account for the extra time and effort and cost to ramp up agile / lean / SAFe practices and tooling

Things to Think About for: *Organizational Climate Fit*

- Senior stakeholders openly, actively, and explicitly support the use of Agile/Lean methods, including accepting the ramp up required to effectively use these methods.
- Sponsorship support for agile/lean/SAFe methods is explicit and cascades throughout management levels and the acquisition chain
- Environment supports data-based, de-centralized decision-making rather than HIPPO-based decision-making. (Highest Paid Person's Opinion)
- Organization's change history for introducing new engineering and management approaches is recently positive
- Organization supports early and frequent delivery of potentially-shippable software to customers and end users
- Organization accepts that: learning occurs throughout the development process, resulting in ongoing course corrections; accelerating learning is a major agile / lean risk reduction strategy.

Things to Think About for: *Program Climate Fit*

- Program goals and success strategies (e.g. roadmaps, product portfolios) have been clearly communicated and reflect stakeholders concerns
- There is a high level of early and ongoing cross-functional (e.g. dev, security, test, CM, operations) participation at both the team and the ART level.
- The full range of PI activities (planning, sustainable team execution, demos and reviews, etc.) are supported and are aligned with oversight and management review mechanisms.
- Program stakeholders and processes support the agile approach to requirement (i.e. - beginning with incomplete and abstract requirements which are elaborated as the product evolves).
- All roles involved in the ART have the skills and competencies needed to perform their role and / or have relevant training and coaching available.
- The Program operates under a people-centric ethos emphasizing high-trust, high bandwidth communications and sustainability and providing supportive physical and social environments.
- Program incentives and organizational climate supports a "many hats" culture and close collaboration among all ART members.

Things to Think About for: *System Architecture Fit*

- Product architecture allows for components to be produced independently (architecture reflects loose coupling).
- System architecture is able to accommodate "develop on cadence, deliver on demand“
- The architecture supports incremental build-out using an "architectural runway" approach
- Mission- and safety-critical components and dependencies are identified and accounted for in the Program strategy. Mission and safety criticality are supported by the selected development practices.
- Security drivers are identified and accounted for in the Program strategy and architecture. Security needs of the program are supported by the selected development practices.
- Required technology and support for architectural development and test supplied by IT and interfacing systems
- Early system and software artifacts reflect a small number of fixed requirements/design decisions, and a larger number of variable requirements/design trade spaces.

Things to Think About for: *Tools Environment Fit*

- Project management tools that support SAFe are available and integrated into the program's tool environment.
- Automated testing tools are available and implemented for unit, integration, functional and performance test
- Automated testing tools are available and implemented for unit, integration, functional and performance test
- Highly skill support is available for customizing and extending tools and training development, test and operations on their usage
- Teams have access to environments that correspond closely to production
- Support exists (tools and personnel) to ensure that the work of all teams within the ART can be integrated and demoed throughout the course of the PI

Things to Think About for: *Agile / SAFe Practices Fit*

- All teams within the ART are working to a common sprint cadence
- Agile teams are cross-functional, self-organizing and self-managing with dedicated members and assigned Product Owner and Scrum Master
- All ART level Roles (RTE, Product Mgmt, System Arch / Engr, Business Owners) are filled with trained individuals with sufficient skills, time and decision-making authority
- All ART members have been trained on the SAFe principles and are able to translate that understanding into day to day practices and decision-making.
- All SAFe ceremonies and cadences are implemented (e.g. - PI Planning, Innovation and Planning Iteration; Inspect and Adapt Workshop; per-iteration planning and reviews

Things to Think About for: *Agile / SAFe Practices Fit-2*

- A Program Backlog is maintained , consisting primarily of Epics and Features and prioritized by Product Management after consultant with relevant stakeholders.
- Features from the Program Backlog are split into User stories which are presented to the Agile teams
- ART members are trained in feature and story splitting techniques. Story splitting is informed by an understanding of both user needs and the underlying architecture and technology of the system.
- Solutions Engineering is conducted as a collaborative effort between the Agile teams, the Architecture team and Product Management.

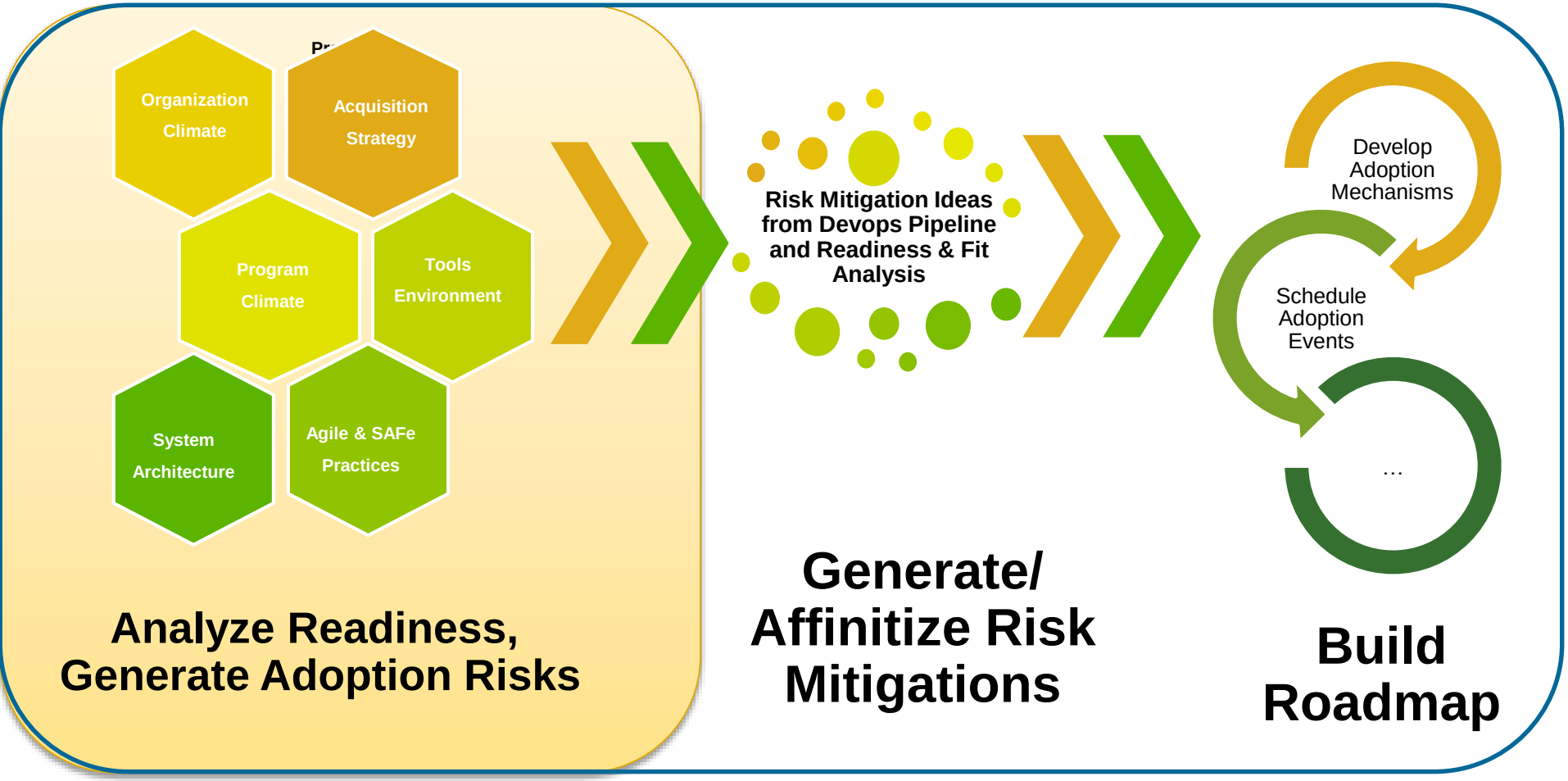
What Do We Do Next?

Now, it's time to:

Aggregate and analyze the themes of our risks and issues

Next Steps

Still Work to Do to Plan and Execute Successful Adoption!



Contact Information

Facilitator 1 - Point of Contact
Suzanne Miller
Principal Researcher
Telephone: +1 412.268.9143
Email: smg@sei.cmu.edu

Facilitator 2
Nanette Brown
Senior Member of the Technical Staff
Telephone: +1 203.231.3999
Email: nb@sei.cmu.edu