Agile Awareness Workshop for NC3

Module 1: Introduction

July 2019

SEI Continuous Lifecycle Solutions Initiative

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Carnegie Mellon University
Software Engineering Institute

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.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM19-0696

Topics the SEI will address in this course include:

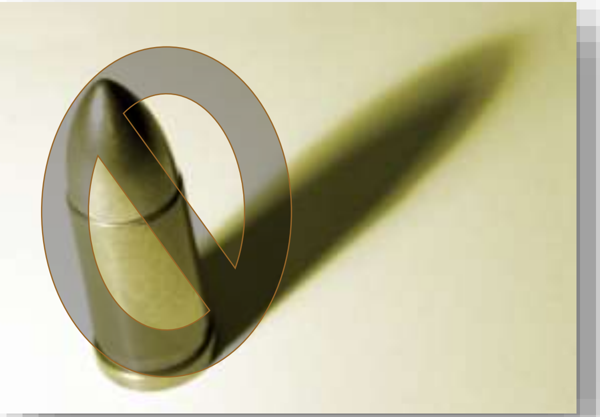
- ***Introduction***
- **Basic Agile/Lean Concepts**
  - ➢ Today’s Landscape
  - ➢ Agile Tenets, Principles and Concepts
  - ➢ Scaling Agile
  - ➢ SAFe Foundations
- **Enabling an Agile/Lean Culture in Regulated Settings**
- **Summary**

Course Purpose

This course is an early step in building your skills and knowledge of software acquisition issues and solutions in a regulated setting using Agile methods

We will accommodate your knowledge needs throughout the course to the best of our ability and will point to resources where time does not permit us to address your needs in the context of this training

BLUF—Bottom Line Up Front



Agile is not a silver bullet for software acquisition, but it can be used effectively in a regulated setting, like government, when appropriate

“Agile” isn’t one approach or method – it’s an umbrella term and is best thought of as reflecting the Agile tenets and principles of the Agile Manifesto

How the program office tasks for oversight are performed in a program using Agile is different from a traditional acquisition, but the responsibility of the PMO remains

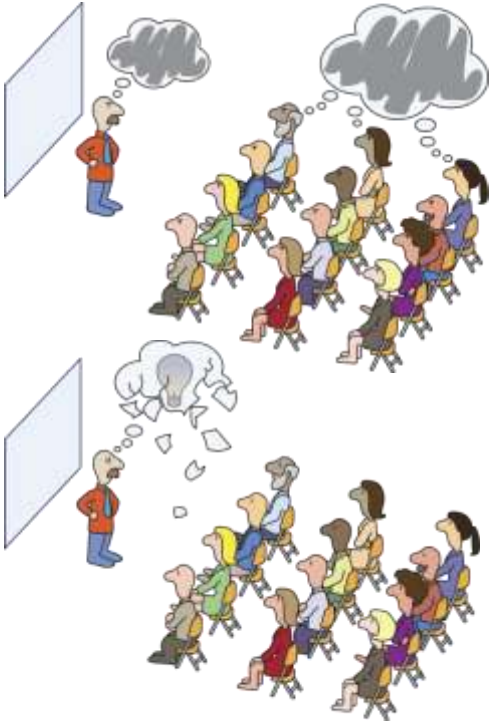
Acquisition/Program Office staff typically are not “developing” the product

- This course provides chances to see and try some of the things that will be different when planning and engaging in an acquisition using Agile approaches

Materials

Slide prints plus exercise handouts

Audience—Who are You?



What role do you have in software acquisition?

Engineer

Program Management

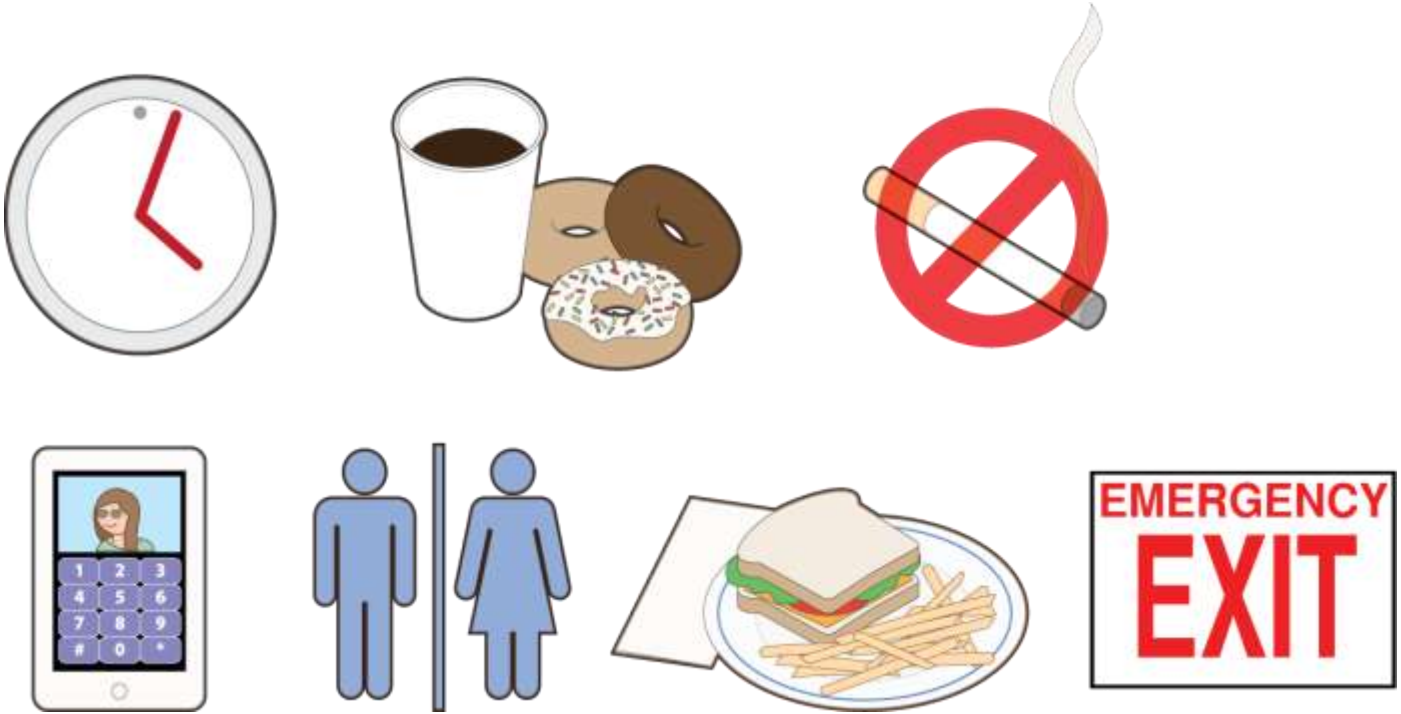
Test

Finance Staff

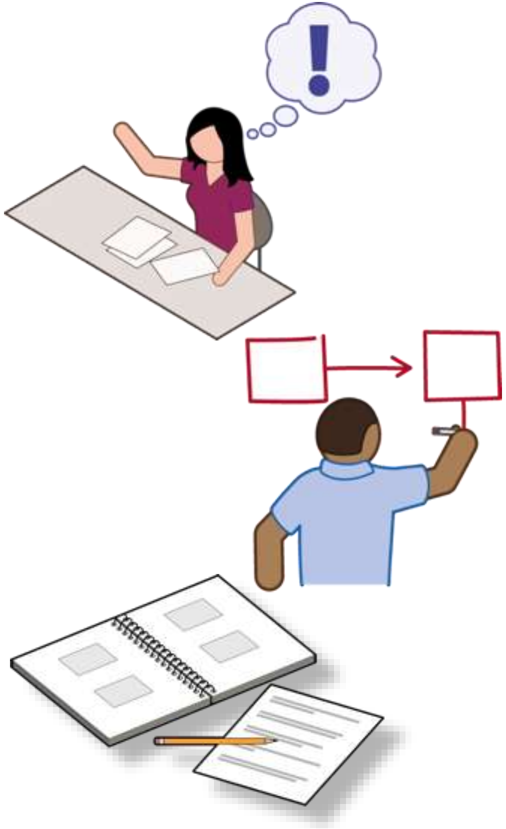
Contracting Staff

[your role here]…

Course Logistics



Approach



Mix of

- exercises,
- discussions
- presentations

What Else do You Need to Learn?

Break into pairs and discuss what one thing **BEYOND** what we've discussed will be included that you want to get out of the course (3 min)

Each pair provides two sticky notes, each with one idea, to instructor's flip chart

Instructor will quickly group them and tell group which ones are in and which are out of scope for the course.

Agile Awareness Workshop for NC3

Module 2: Basic Agile/Lean Concepts

July 2019
SEI Continuous Lifecycle Solutions Initiative

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.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM19-0696

Topics the SEI will address in this course include:

- **Introduction**
- ***Basic Agile/Lean Concepts***
  - ➢ Today’s Landscape
  - ➢ Agile Tenets, Principles and Concepts
  - ➢ Scaling Agile
  - ➢ SAFe Foundations
- **Enabling an Agile/Lean Culture in Regulated Settings**
- **Summary**

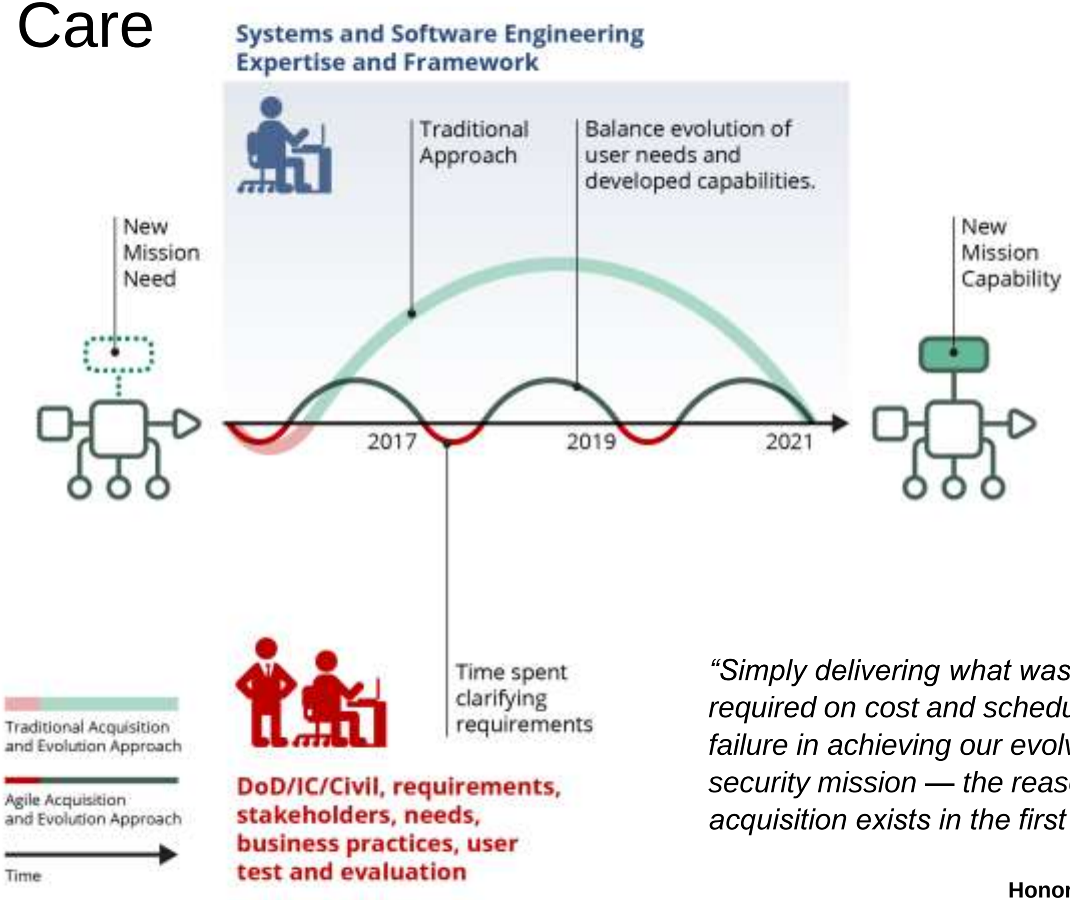
Today’s Landscape

Why Does the DoD Care about Agile?

Deliver performance at the speed of relevance

Streamline rapid, iterative approaches from development to fielding

National Defense Strategy Summary
Jan 2018

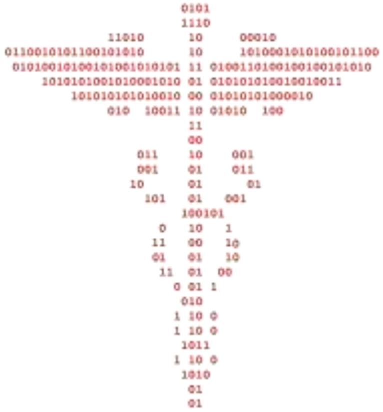


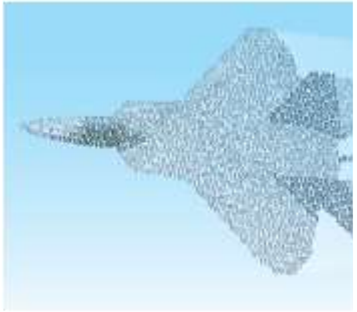
Systems and Software Engineering Expertise and Framework
Traditional Approach
Balance evolution of user needs and developed capabilities.
New Mission Need
New Mission Capability
2017
2019
2021
Time spent clarifying requirements
DoD/IC/Civil, requirements, stakeholders, needs, business practices, user test and evaluation
Traditional Acquisition and Evolution Approach
Agile Acquisition and Evolution Approach
Time

"Simply delivering what was initially required on cost and schedule can lead to failure in achieving our evolving national security mission — the reason defense acquisition exists in the first place."

Honorable Frank Kendall
Under Secretary of Defense (AT&L)
2015 Performance of The Defense Acquisition System

Why Does the DoD Care about Software?

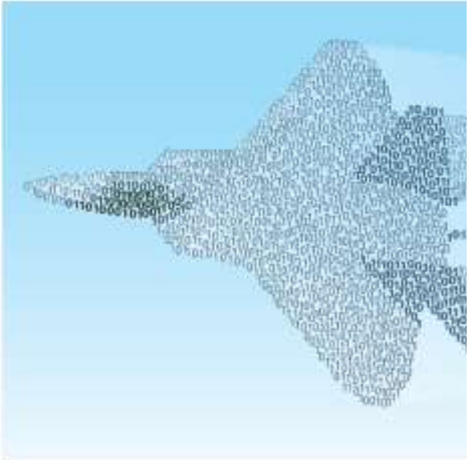




If the world isn’t software-defined yet, it is certainly software-reliant!

Software is often part of a larger System Development Activity

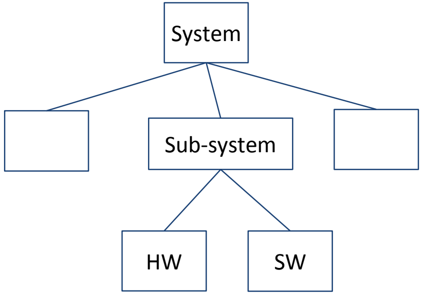
- Major system engineering endeavors in aircraft vehicles, satellites, economic infrastructure, energy management…
    - Often these are contracted systems, where the software part of the system is being acquired by the owner of the overall system



Many Project Management Practices Assume a Hardware-Centric System

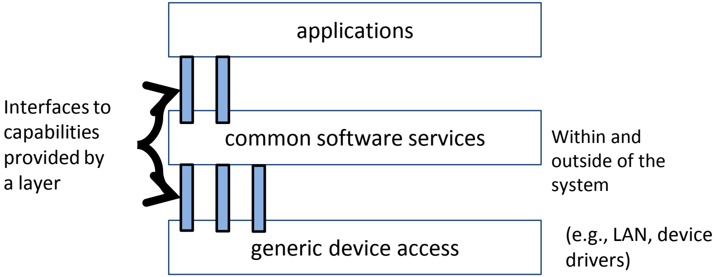
Hardware Sys Development Assumptions

- Systems can be decomposed into discrete, independent, and hierarchically related components (or subsystems).
- Component interaction can largely be derived from the original decomposition or BOM
- Quality attributes can be allocated to specific components.



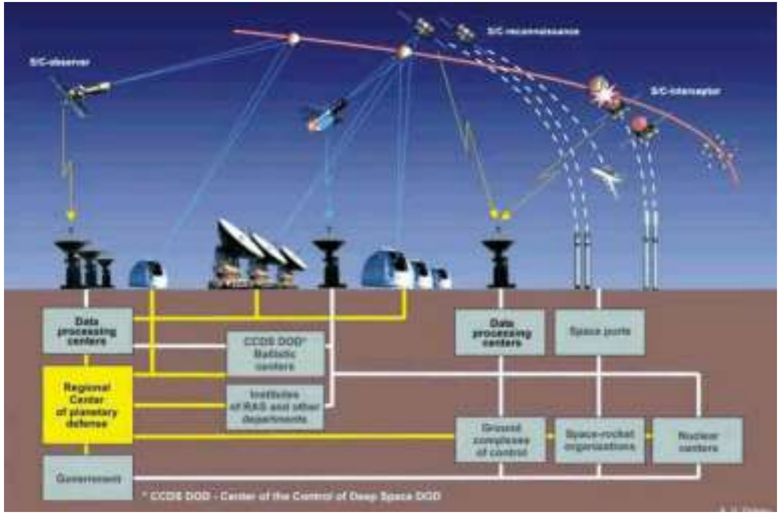
Software Realities

- Complex interactions among components and across layers
- Interactions can take many different paths.
- Quality attributes are achieved by component interactions (*not* by individual components).


applications
Interfaces to capabilities provided by a layer
common software services
Within and outside of the system
generic device access
(e.g., LAN, device drivers)

Software is a Primary Connector Among Systems (of Systems)

“The value of a system passes through its interconnections” [1]


S/C-reconnaissance
S/C-observer
S/C-interceptor
Data processing centers
CCDS DOD* Ballistic centers
Regional Center of planetary defense
Institutes of RAS and other departments
Government
Data processing centers
Space ports
Ground complexes of control
Space-rocket organizations
Nuclear centers
* CCDS DOD - Center of the Control of Deep Space DOD

[1] scaledagileframework.com\apply-systems-thinking

Integrated functionality often means overlapping accountability.

Who “owns” the responsibility for software in a complex system of systems?

DoD Defense Science Board

The Defense Science Board serves as the Federal Advisory Committee chartered to provide Department of Defense leadership with "independent advice and recommendations on science, technology, manufacturing, acquisition processes, and other matters of special interest to the DoD..."

Currently, the Board's authorized strength is forty-eight members and seven *ex officio* members, including the chairs of the Army, Navy, and Air Force advisory committees, and the Defense advisory committees on Policy, Business, Health, and Innovation. The Board's forty-eight members are appointed for terms ranging from one to four years …"

As of February 1, 2018, the newly formed Office of the Under Secretary of Defense for Research and Engineering (USD(R&E)) will serve as the formal sponsor of the Defense Science Board.

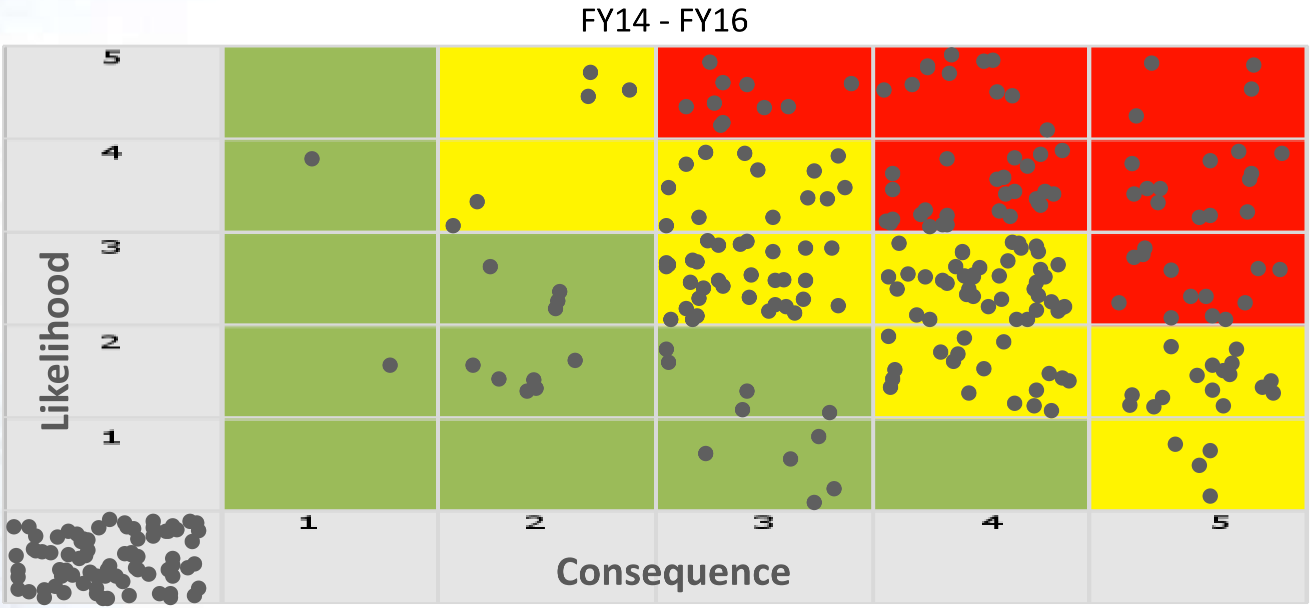
Task Force Findings and Recommendations on "Design and Acquisition of Software for Defense Systems" issued October, 2017.

https://www.acq.osd.mil/dsb/history.htm

Importance of Software in Defense Systems

- Software is a crucial and growing part of weapon systems/national security mission
  - "The DoD is experiencing an explosive increase in its demand for software-implemented features in weapon systems…in the meantime, defense software productivity and industrial base capacity have not been growing as quickly." –Institute for Defense Analyses, 2017
- Software never dies. It will require DoD to update continuously and indefinitely

Software Risk Assessed by DoD Program Offices



Software not in top program risks

Software assessed among most frequent and most critical challenges, driving program risk on ~ 60% of acquisition programs

Defense Science Board Task Force Findings

Commercial development practices allow rapid development, deployment and adaptability.

The DoD is behind the commercial sector as are DoD contractors

DoD contracting must change to incentivize changes in suppliers

- Current contracts and incentives are built around waterfall processes used 20 years ago. This creates a self-reinforcing loop that stifles change

DSB Task Force Recommendations

DoD and its contractors need to adopt continuous iterative development best practices for software

Software acquisition is not about buying a black box end result

- The offeror's software factory should be a key source selection evaluation criterion
- Ensure software sustainment is considered in RFPs
- Contract for an ongoing stream of delivered value; not for a fixed point in time product

DSB Task Force Recommendations

Transition Strategy

- Ongoing development programs should plan to transition to a software factory and continuous iterative development
- Legacy programs to perform a business case on whether to transition the program

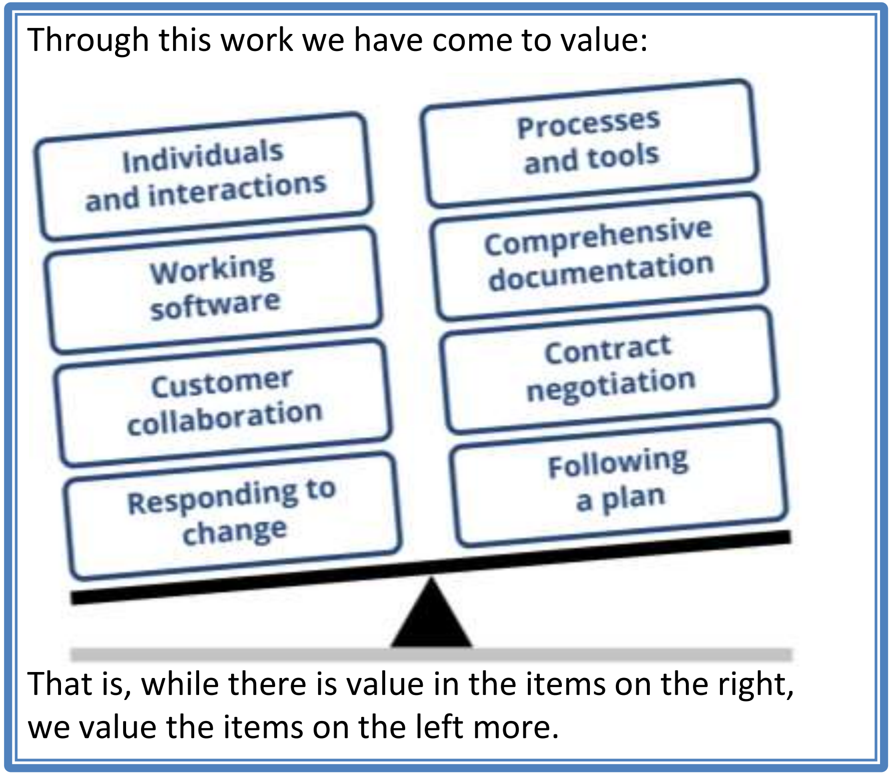
Build metrics based on Agile best practices

Accept that there may be initial short term costs

Develop competency in the workforce. (It takes 2 to tango)

Agile Tenets, Principles, and Concepts

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

Discussion Exercise

Which of the above principles and tenets

- Are supported by or compatible with current acquisition practice in regulated settings? (+)
- Are **un**supported by or **in**compatible with current acquisition practice in regulated settings? (-)

Teams 1 and 2: Agile principles 1-6

Teams 3 and 4: Agile principles 7-12

Useful Interpretation of Agile Principles for Government Settings

(1/3)

Agile Principle	Useful Interpretations in Government Settings
The highest priority is to satisfy the customer through early and continuous delivery of valuable software.	In government, the “customer” is not always the end user. The customer includes people who pay for; people who use; people who maintain; as well as others. These stakeholders often have conflicting needs that must be reconciled
Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.	Rather than saying “competitive” advantage, we usually say “operational” advantage. This principle causes culture clash with the “all requirements up front” perspective of many large, traditional approaches.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.	What it means to “deliver” an increment of software may well depend on context. With large embedded systems, we are sometimes looking at a release into a testing lab. Also, for some systems, the operational users are not able to accept all :”deliveries” on the development cadence – because there are accompanying changes in the workflow supported by the software that require updates.
Business people and developers must work together daily throughout the project.	In government settings, we interpret “business” people to be end users and operators, as well as the other types of stakeholders mentioned in Principle 1, since in many government settings, the business people are interpreted as the contracts and finance group.

Source: SEI Congressional testimony July 14, 2016 to House Ways and Means Committee.

Useful Interpretation of Agile Principles for Government Settings
(2/3)

Agile Principle	Useful Interpretations in Government Settings
Build projects around motivated individuals. Give them environment and support they need, and trust them to get the job done.	A frequent challenge in government is to provide a suitable technical and management environment to foster the trust that is inherent in Agile settings. Allowing teams to stay intact and focused on a single work stream is another challenge.
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	In today's world, even in commercial settings, this is often interpreted as "high bandwidth" rather than only face-to-face. Telepresence via video or screen-sharing allows more distributed work groups than in the past.
Working software is the primary measure of progress.	Our typical government system development approaches use *surrogates* for software – documents that project the needed requirements and design – *rather than the software itself*, as measures of progress. Going to small batches in short increments allows this principle to be enacted, even in government setting, although delivery may well to be a test environment or some internal group other than users themselves.
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely	This principle is a caution against seeing agility just as "do it faster." Note that this principle includes stakeholders outside of the development team as part of the pacing.

Source: SEI Congressional testimony July 14, 2016 to House Ways and Means Committee.

Useful Interpretation of Agile Principles for Government Settings
(3/3)

Agile Principle	Useful Interpretations in Government Settings
Continuous attention to technical excellence and good design enhances agility	This is a principle that often is cited as already being compatible with traditional government development.
Simplicity– the art of maximizing the amount of work not done– is essential.	One issue with this principle in government setting is that our contracts are often written to penalize the development organization if they don't produce a product that reflects 100% of the requirements. This principle recognizes that not all requirements we think are needed at the onset of a project will necessarily turn out to be things that should be included in the product.
The best architectures, requirements, and designs emerge from self-organizing teams.	Note that the principle does not suggest that the development team is necessarily the correct team for requirements and architecture. It is however, encouraging teams focused in these areas to be allows some autonomy to organize their work. Another complication in many government settings is that we are often re-architecting and re-designing existing systems.
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	This principle is an attempt to ensure that “lessons learned” are actually learned and applied rather than just being “lessons written”

Source: SEI Congressional testimony July 14, 2016 to House Ways and Means Committee.

Working Definition of Agile

Business Team
To succeed, all levels of an organization (& the customer) need to support the agile process
Technical Team
customer

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

Some Observable Characteristics of Agile Implementations

Iterative—elements are expected to move from skeletal to completely fleshed out over time, not all in one step

Incremental—delivery doesn’t occur all at once

Collaborative—progress is expected to be made by stakeholders and the development team working collaboratively throughout the development timeframe

Loosely-coupled Architecture—multiple self-organizing, cross-functional teams work concurrently on multiple product elements (e.g., requirements, architecture, design, and the like) for multiple loosely coupled product components

Dedicated—team members are allowed to focus on the tasks within an iteration/release as opposed to multi-tasking across multiple projects

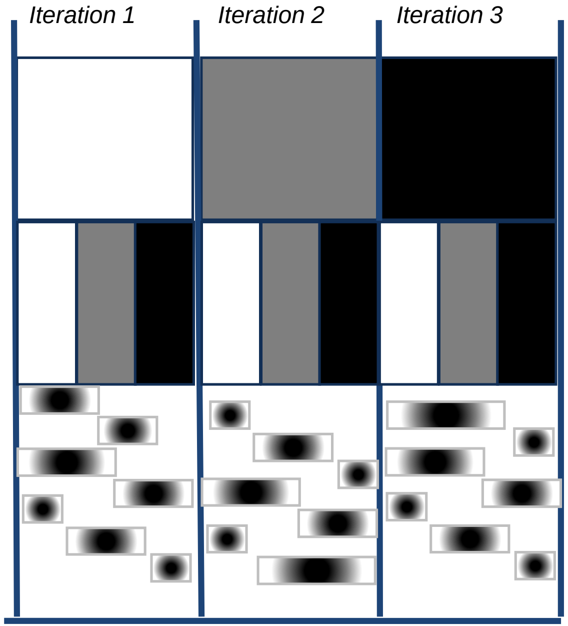
Time-boxed or Flow-based—relatively short-duration development cycles that permit changes in scope rather than changes in delivery time frame

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



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
- ….

Business Team
To succeed, all levels of an organization (& the customer) need to support the agile process
Technical Team
customer

Traditional vs. Agile Approaches

Agile approach:

- Supports an environment of innovation, change & uncertainty
  - Programs with volatile requirements and environment
  - Programs where solutions are sufficiently unknown that significant experimentation is likely to be needed
  - Programs for which the technology base is evolving rapidly
- Requires an environment that can support collaboration
  - Programs with stakeholders who can engage with developers in ongoing, close collaboration

Nidiffer, K. Miller, S. & Carney, D. *Potential Use of Agile Methods in Selected DoD Acquisitions: Requirements Development and Management* (CMU/SEI-2013-TN-0006), September 2013.

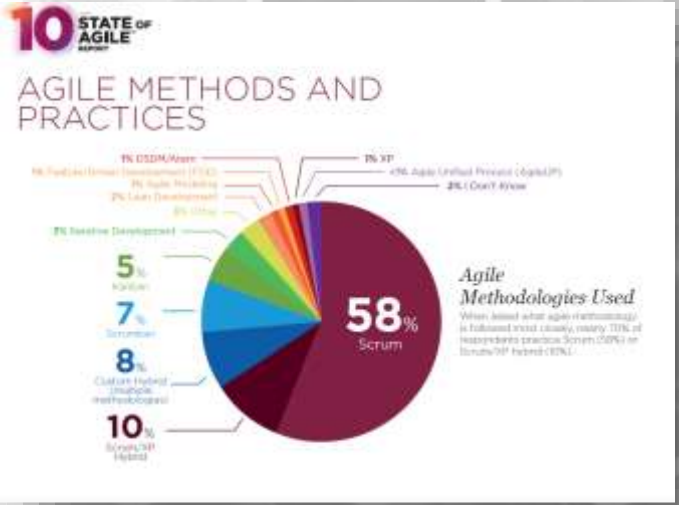
Traditional vs. Agile Approaches

Traditional approach

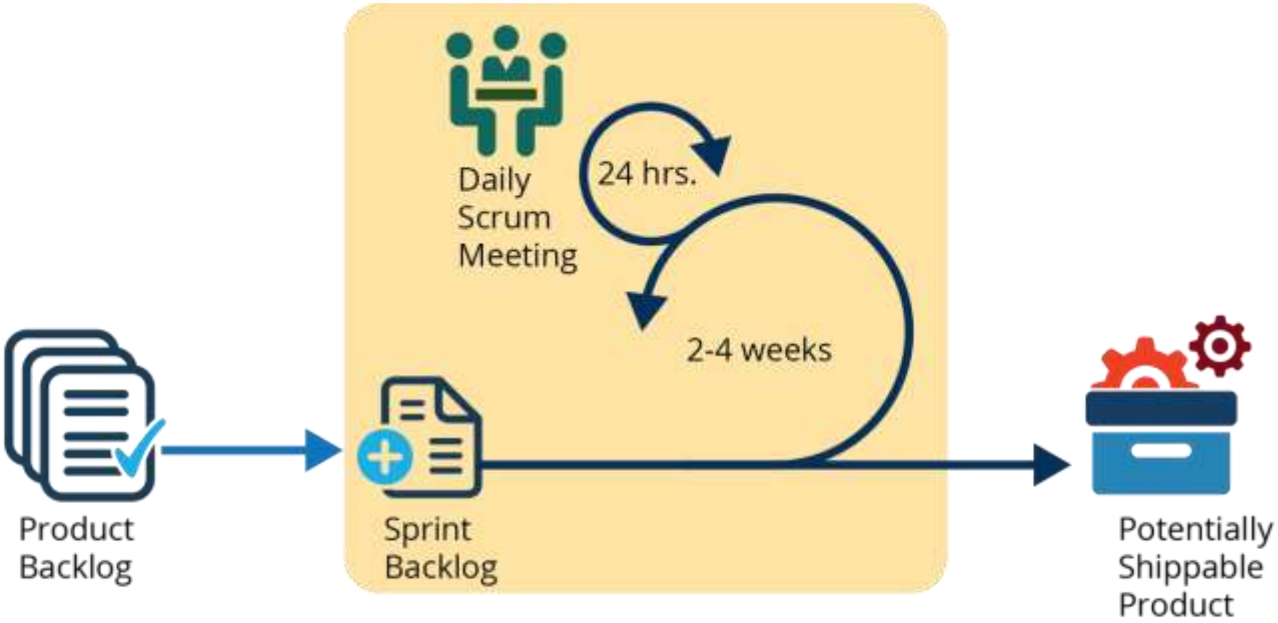
- Is consistent with the acquisition lifecycle guidance provided in typical acquisition guidance
- Assumes stability:
  - Programs with stable requirements and environment, with known solutions to the requirements
  - Programs for which the technology base is evolving slowly (technology is not expected to be refreshed/replaced within the timeframe of the initial development)
- Requires stakeholders who:
  - Know what they want before they see it
  - Communicate well via documents

Nidiffer, K. Miller, S. & Carney, D. *Potential Use of Agile Methods in Selected DoD Acquisitions: Requirements Development and Management* (CMU/SEI-2013-TN-0006), September 2013.

Scrum— A Time-Boxed Process Framework for Small AgileTeams



Key Elements of Scrum



Scrum Framework

Roles

- Product owner
- Scrum Master
- Team

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts
- Potentially Shippable Product Increment

Other Concepts Used with Scrum

- Epics
- User Stories

Ceremonies

- Release Planning
- Sprint planning
- Daily scrum meeting
- Sprint review
- Sprint retrospective

New Terms - Roles

Product Owner:

- the “voice of the customer,” accountable for ensuring business value is delivered by creating customer-centric items (typically stories (or user stories), prioritizing them, and maintaining them in the product backlog

Scrum Master:

- the process facilitator for a development team who works with the team to identify and find solutions to impediments to progress, as well as maintaining the information radiators that show progress to stakeholders and representing the team in management activities related to the project

Team:

- cross-functional self-organizing team of 5-10 members including Agile Team Lead, Product Owner or Capability Owner, and developers, testers and SMEs

New Terms - Artifacts

Product Backlog:

- a prioritized list of (user) stories and defects ordered from the highest priority to the lowest

Sprint Backlog:

- a list of tasks that the team believes need to be completed to satisfy the user stories for that sprint and meet the sprint goal

Burndown Charts:

- a visual tool displaying progress via a simple line chart representing remaining work (vertical axis) over time (horizontal axis)

Potentially Shippable Product Increment:

- the result of a sprint—even though it is unlikely that the project would be cancelled after a particular sprint, the idea is that the product is implemented in such a way that some value could be derived no matter when the project stops

New Terms – Ceremonies$_1$

Release Planning: planning activities across a defined number of sprints that implement a desired set of features to the point of delivery to the next customer (could be external, often is internal, e.g. system test)

Sprint Planning: planning activities that occur at the beginning of a sprint, including determining the capacity for the sprint, establishing a sprint goal, performing relative estimation on the candidate stories for the sprint, and creating the task list (sprint backlog) the team will use to self-manage the work of the sprint

Daily Scrum Meeting: daily standups that are used to communicate what was accomplished yesterday, what will be accomplished today and identify any impediments to the Scrum Master in order for action to be taken to eliminate the issue.

New Terms – Ceremonies$_2$

Sprint Review:

- the activity at the end of a sprint that demonstrates the code that has been completed and verifies, with the product owner, that the sprint goal has been met

Sprint Retrospective:

- an activity at the end of the sprint review/demo where the team and the Scrum Master review the processes and practices used in the sprint to identify improvements to be tried in the next or future sprints—Agile’s way of “inspect and adapt” for processes

New Terms - Other

Epics:

- user stories that are too large to directly implement. There is no official threshold to differentiate between an epic and a user story

User Stories:

- used in several Agile methods, derive from Extreme Programming; used as the basis for defining the functions a system must provide, and include a written sentence or two and a series of conversations about the desired functionality, to shift the focus from writing about requirements to talking about them

Definition of Done[1]:

- A checklist of value-added activities that must be performed before a story can be declared to be Done. The checklist is determined by the Agile Team.

Sprint:

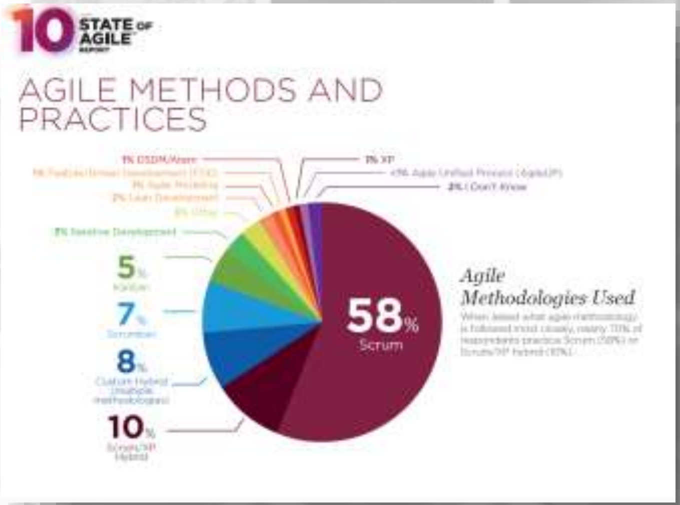
- an iteration of a defined, consistent time span (2-4 weeks is typical) during which the backlog items selected for the iteration are planned, designed, implemented, tested, and demonstrated to the product owner/customer

Information Radiator:

- a physical or virtual display of tasks and progress that is accessible to all stakeholders throughout the project

[1] https://www.scrumalliance.org/community/articles/2008/september/what-is-definition-of-done-(dod)

Kanban— A Flow-based Practice for Continual Delivery



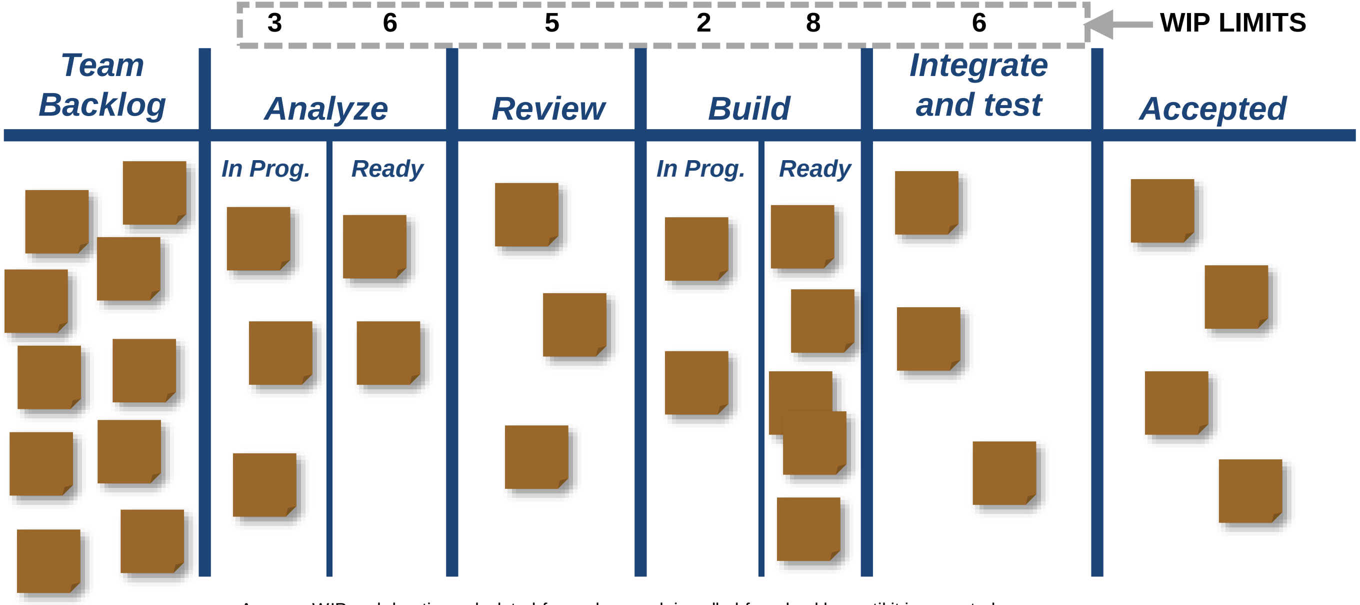
When asked what agile methodology is followed most closely, nearly 70% of respondents practice Scrum (58%) or Scrum/XP hybrid (10%).

5 Core Properties of Kanban

- Visualize the workflow
- Limit WIP
- Manage Flow
- Make Process Policies Explicit
- Improve Collaboratively (using models & the scientific method)

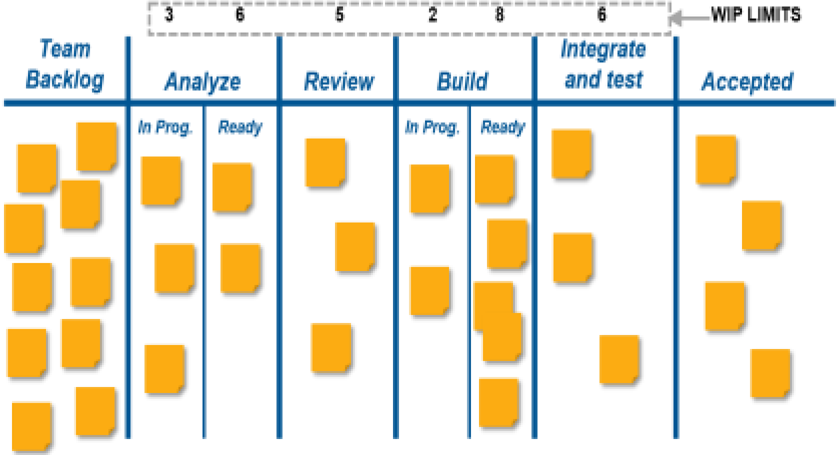
http://www.djaa.com/principles-kanban-method-0

Visualize and Limit WIP, Reduce Batch Sizes, and Manage Queue Lengths



Average WIP and duration calculated from when work is pulled from backlog until it is accepted

Key Concepts-Useful for Tasks that Don’t Easily Conform to a Time Box



Average WIP and duration calculated from when work is pulled from backlog until it is accepted

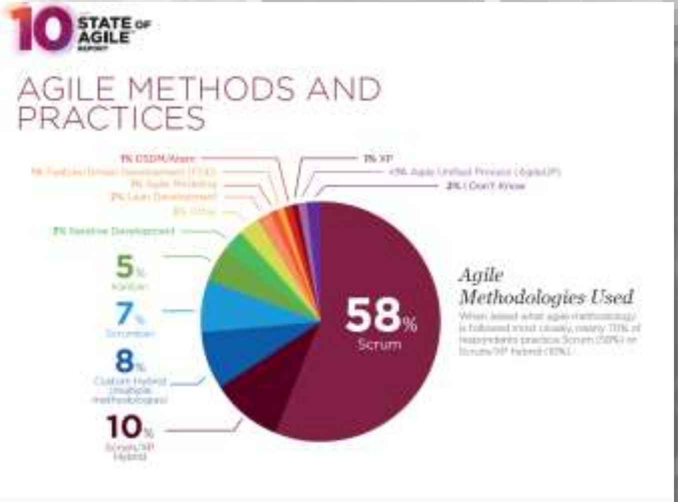
Defined States the Work Progresses Through

Rules about Limiting Work in Process for Each State

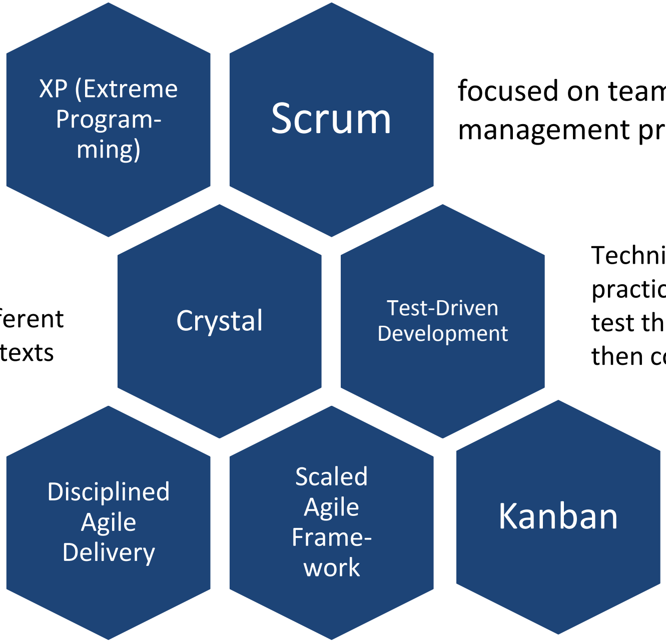
Definitions of “Special” Classes of Tasks (Due Date, Expedited, or others as defined by the project)

Explicit Acceptance Criteria

Additional Agile Methods



Methods Generally Termed “Agile”



XP (Extreme Programming)

focused on team technical practices

Scrum

focused on team management practices

Crystal

Encourages risk-based selection of practices; different patterns for different contexts

Test-Driven Development

Technical and management practices focused on writing the test that proves acceptance, then coding to that

Disciplined Agile Delivery

Originally derived from Rational Unified Process, designed to scale

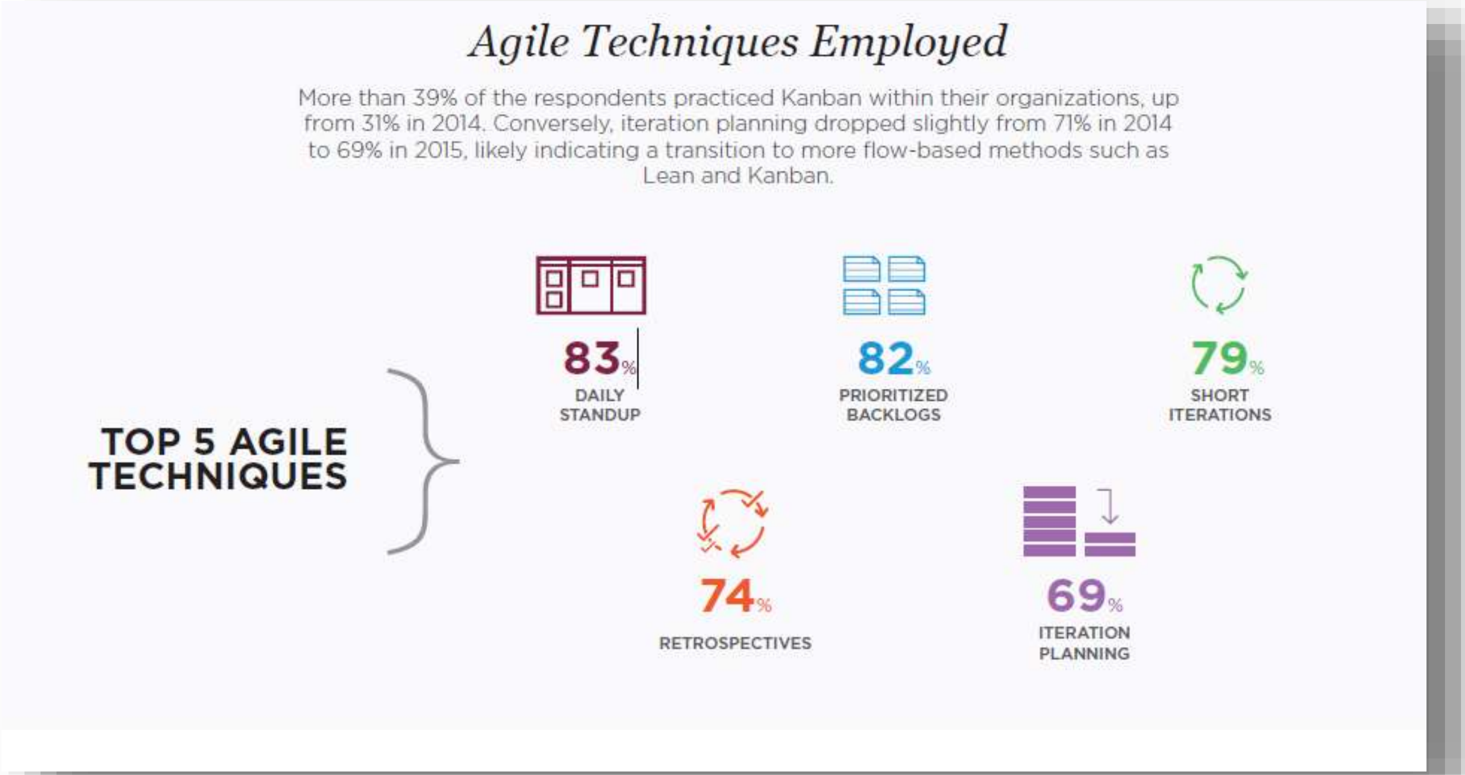
Scaled Agile Framework

Merger of Lean, Kanban, and other Agile methods to support large scale projects

Kanban

Pull-based approach particularly favored for services like security, systems engineering

Top 5 Agile Techniques Used in Industry



Agile Techniques Employed

More than 39% of the respondents practiced Kanban within their organizations, up from 31% in 2014. Conversely, iteration planning dropped slightly from 71% in 2014 to 69% in 2015, likely indicating a transition to more flow-based methods such as Lean and Kanban.

TOP 5 AGILE TECHNIQUES

- 83% DAILY STANDUP
- 82% PRIORITIZED BACKLOGS
- 79% SHORT ITERATIONS
- 74% RETROSPECTIVES
- 69% ITERATION PLANNING

Source: 10th Annual Survey on State of Agile, Version One, April 2016

Scaling 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.

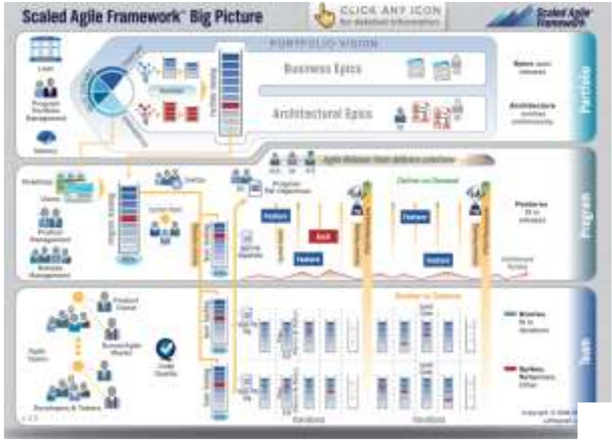
Agile is a Team Approach

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
- ….

Business Team
To succeed, all levels of an organization (& the customer) need to support the agile process
Technical Team
customer
Software Engineering Institute

Multiple Commercial Scaling Frameworks to Choose From

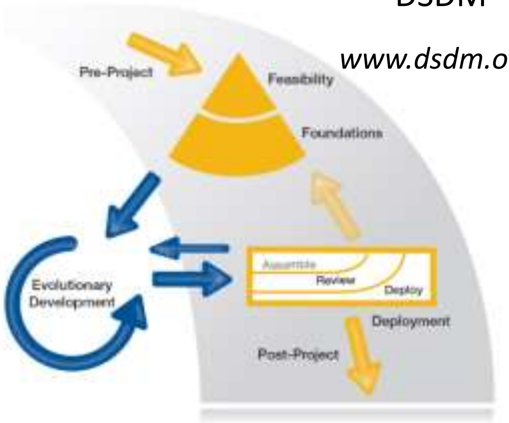


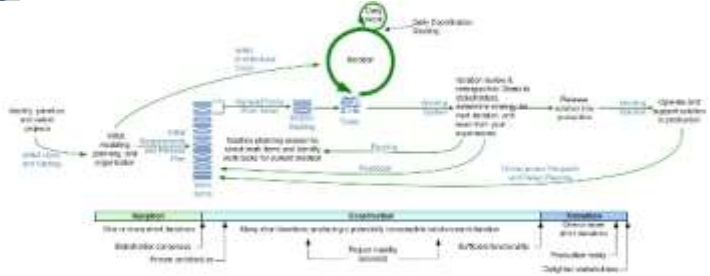
SAFe

www.scaledagileframework.com

DSDM

www.dsdm.org

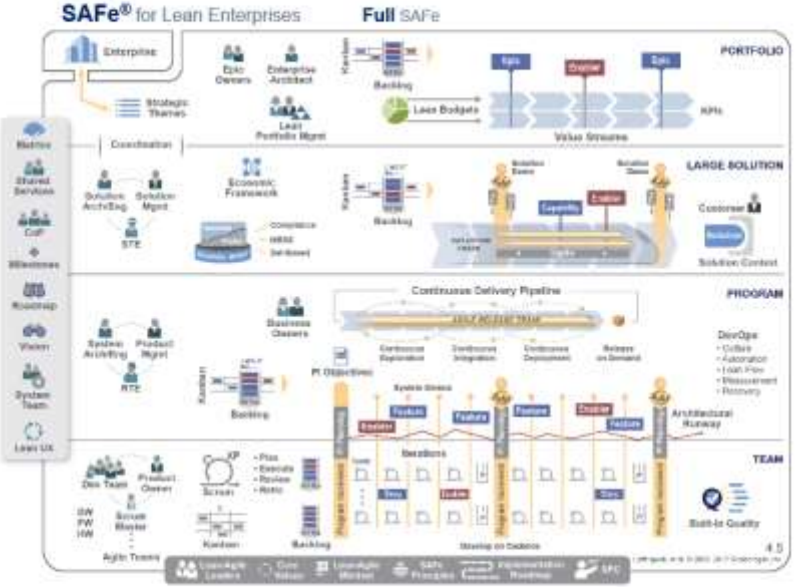




DAD

https://www.ibm.com/developerworks/community/blogs/ambler/entry/disciplined_agile_delivery_dad_lifecycle14?lang=en

Scaled Agile Framework



Version One survey lists SAFe as the most popular scaling technique

SAFe is based on:

- Scrum Team Practices
- XP Technical Practices
- Lean Thinking and Kanban

SAFe “big-room” Program Planning sessions, are a key differentiator.

SAFe framework allows for incorporation of enterprise roles.

Why does the SEI Teach / Consult on the Scaled Agile Framework

As an FFRDC, SEI does not recommend particular tools or techniques

SEI Technical Note, *Scaling Agile Methods for Department of Defense Programs,* provides criteria for government programs to use in determining a scaling framework that fits their needs.

- The TN is found at: https://resources.sei.cmu.edu/library/asset-view.cfm?assetID=484635

SAFe is currently the most adopted scaling framework by DoD contractors.

Having SAFe SPC qualifications allows SEI Agile in Government staff to provide government programs with advice that is fully informed by SAFe concepts while preserving a government perspective.

Knowledge for people building the world's most important systems

SAFe® is a freely revealed knowledge base of integrated, proven patterns for enterprise Lean-Agile development.

scaledagileframework.com

SAFe Foundations

Foundations of the Scaled Agile Framework® (SAFe®) 4.5

V4.5.0

SCALED AGILE®

We thought we'd be developing like this.

But sometimes it feels like this.

Library of Congress

And our retrospectives read like this:

No way to improve systematically

Too little visibility

Too early commitment to a design that didn’t work

Late delivery

Problems discovered too late

Under-estimated dependencies

Massive growth in complexity

Hard to manage distributed teams

Phase gate SDLC isn't helping reduce risk

Poor morale

Management's challenge

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

"… and if you can't come, send no one."

—Vignette from *Out of the Crisis*, Deming,1986

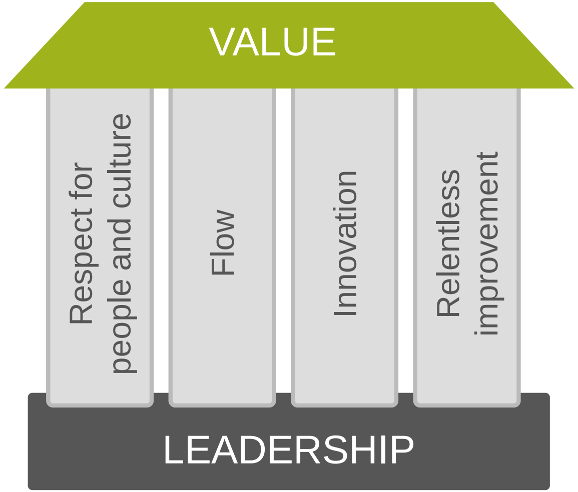
What it is they must do

- Embrace a Lean-Agile mindset
- Implement Lean-Agile practices
- Lead the implementation
- Get results

Embrace a Lean-Agile mindset

Embrace Lean-Agile values

House of Lean



Value in the shortest sustainable lead time

Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

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

SAFe House of Lean Pillars

Flow

- The heart of Lean thinking
- Reduce time to value by eliminating waste
- SAFe – Principle of Systems Thinking

Innovation

- Ability to pivot
- Build slack into the system to support innovation
- SAFe - Innovation and Planning iteration

Relentless Improvement

- “A constant sense of danger” – don’t get complacent
- Everyone engaged
- SAFe - Retrospectives and Inspect & Adapt Workshop

Respect for People and Culture

- “Individuals and interactions over processes and tools”
- “Customer collaboration over contract negotiation”
- SAFe – “To change the culture, you have to change the organization”

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

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

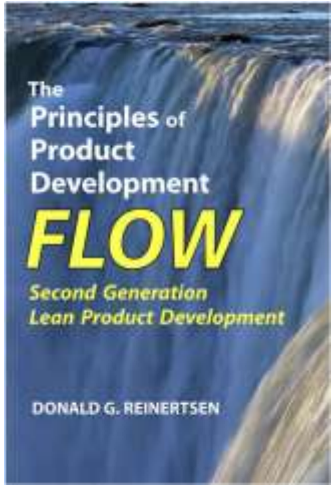
100 implementations

- 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

The Principles of Product Development Flow

Asks the question - “What lessons can we take from other disciplines that will assist us in the endeavor of Product Development?”

- Economics
- Queuing Theory
- Network Design
- Manufacturing
- Control System Engineering
- Maneuver Warfare
- Computer Operating System Design
- Probability and Statistics
- …


The Principles of Product Development
FLOW
Second Generation Lean Product Development
DONALD G. REINERTSEN

175 Principles across
8 categories

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

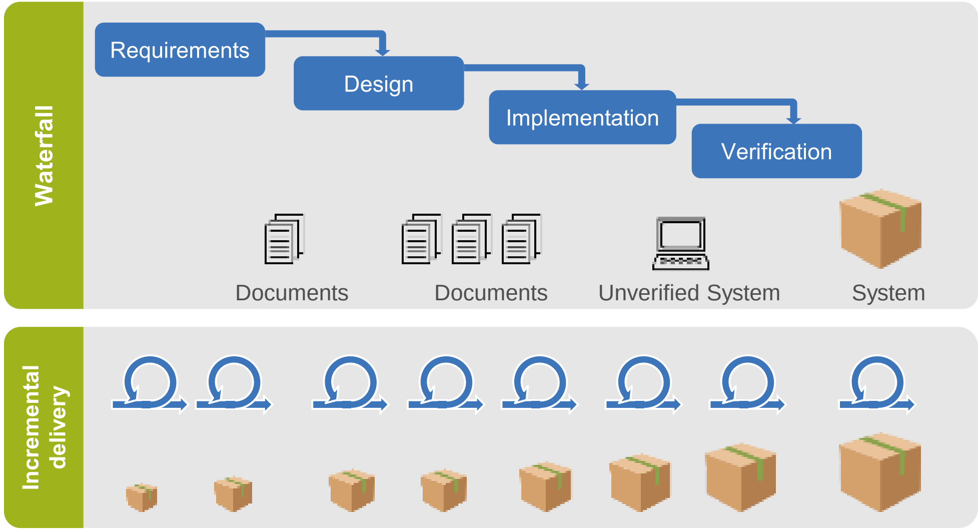
#6 - Visualize and limit WIP, reduce batch sizes, and manage queue lengths

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

#8 - Unlock the intrinsic motivation of knowledge workers

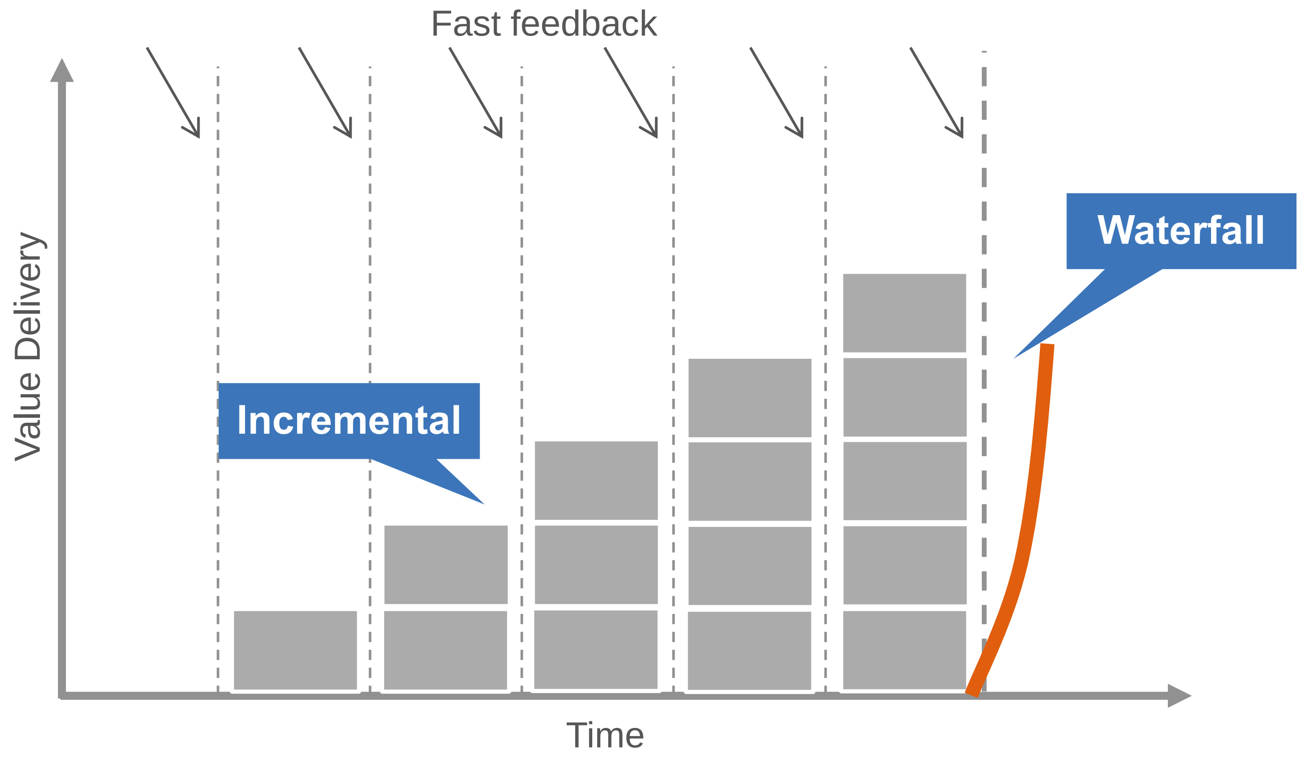
#9 - Decentralize decision-making

Deliver incrementally



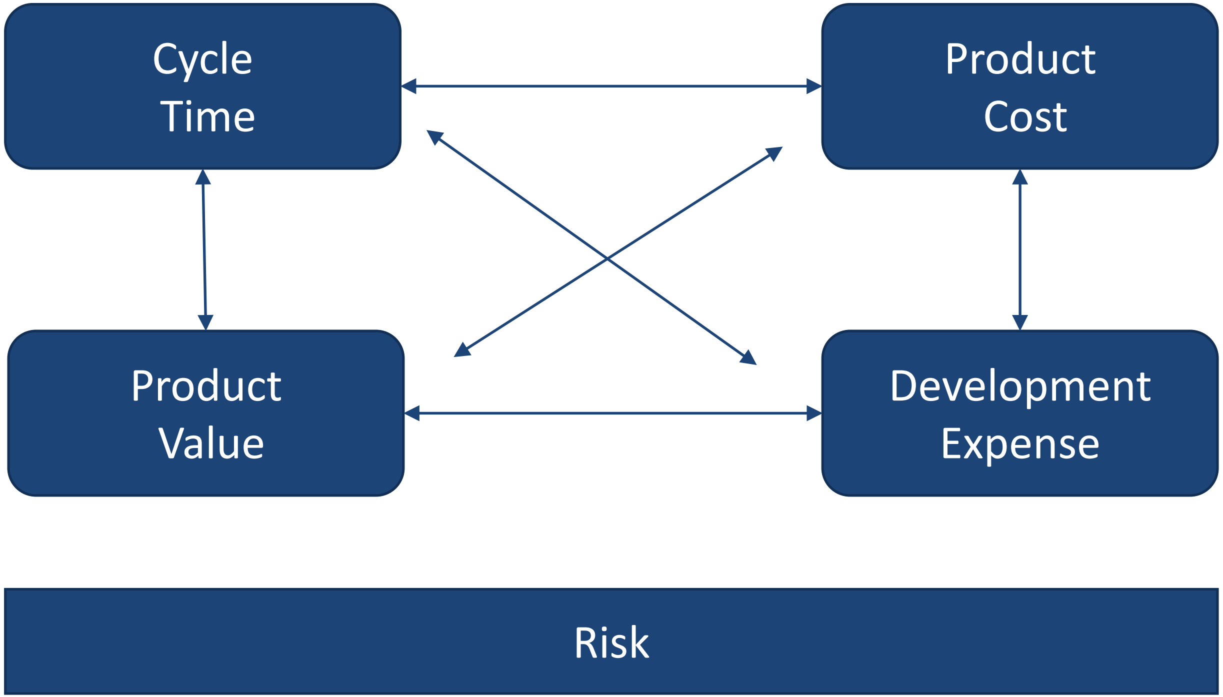
And delivers better economics

Early delivery provides fast value with fast feedback



Economic Decisions Can’t be Made in Isolation

1 of 9



Must consider how changes in one factor impact others

Principles of Product Development Flow, Don Reinertsen, P.30

1 of 9

Economic Decisions Must be Continuously Reassessed

“E9: The Principle of Continuous Economic Tradeoffs: Economic Choices Must be Made Continuously”

- In product development we’re continuously learning and receiving new information
- Our decisions should reflect the current economic reality – otherwise, we’re making decisions based on outdated data

Principles of Product Development Flow,
Don Reinertsen

Focus on Value

1 of 9

Spend time doing only valuable work

Sequence work according to value

- What has the greatest value?
- What has the shortest “time to value”?

Ignore sunk costs

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

Apply Systems Thinking

2 of 9

“Systems thinking has been defined as an approach to problem solving that attempts to balance holistic thinking and reductionist thinking. By taking the overall system as well as its parts into account systems thinking is designed to avoid potentially contributing to further development of unintended consequences.”

https://en.wikipedia.org/wiki/Systems_thinking

Systems thinking

A system must be managed. It will not manage itself.

Left to themselves, components become selfish, independent profit centers and thus destroy the system…

The secret is cooperation between components toward the aim of the organization.

—W. Edwards Deming

Apply Systems Thinking

2 of 9

Consider the Full Value Stream

- All stages from initial concept to final realization of value
- Optimize the flow across all stages, not the productivity of one stage/department over the others
- Systemic thinking should permeate all aspects of product development
  - What’s being built is a system
  - The enterprise undertaking the effort is a system
  - So is the problem space or domain addressed by the solution being built

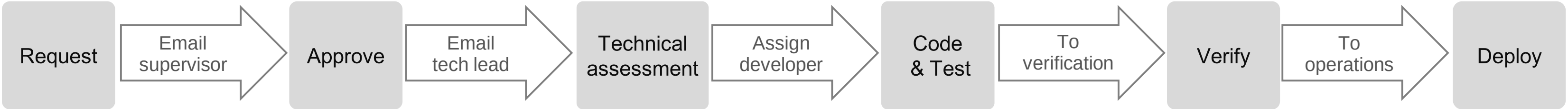
Stage 1
Stage 2
Stage 3

Optimize the full value stream

> *All we are doing is looking at the timeline, from when the customer gives us an order to when we collect the cash. And we are reducing the timeline by reducing the non-value added wastes.*
>
> *—Taiichi Ohno*

- Most problems with your process will surface as *delays*
- Most of the time spent getting to market is a result of these delays
- Reducing delays is the fastest way to reduce time to market

Focus on the delays!



	Request	Email supervisor	Approve	Email tech lead	Technical assessment	Assign developer	Code & Test	To verification	Verify	To operations	Deploy	
Value	1 hour		1 hour		1 hour		1 hour		1 hour		1 hour	**6 hours**
Wait		1 week		2 weeks		2 weeks		1 week		1 week		**7 weeks**

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

Accommodating & Enabling Change & Learning

#3 - Assume variability; preserve options

#4 - Build incrementally with fast, integrated learning cycles

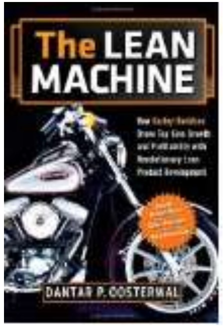
#5 - Base milestones on objective evaluation of working systems

Learning is Central to Product Development

“Learning is the engine that powers product development.

Specifically, experimental learning is the predominate form of learning in a product development environment.

Thus experiential learning cycles are the building blocks for the experimentation and discovery necessary for effective product development.”



“Product development is the process of converting uncertainty to knowledge”

Dantar P. Oosterwal

Assume Variability; Preserve Options

3 of 9

Key Sources of Variability

- External Change
- Imperfect Knowledge of
  - What we want the system to do
  - How to implement the system

Options Strategies

- Small Batches
- Spikes
- Set-Based Design
- Architectural “Hooks”
- Slack

Apply Options Strategies with an Economic View

4 of 9

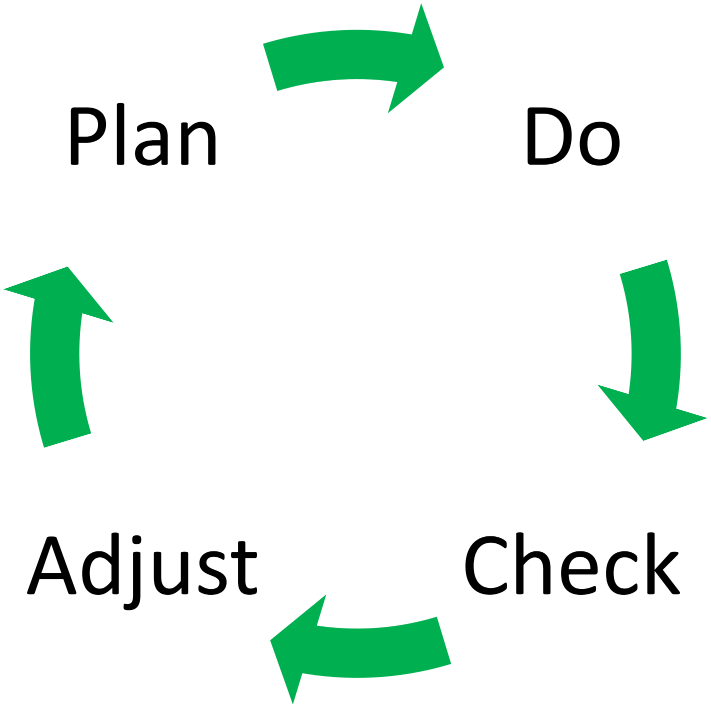
Build incrementally with fast, integrated learning cycles

Most of the risks in product development comes from lack of knowledge; especially from assuming we know more than we do

We need to act in order to learn

PDCA describes a learning cycle

Small batch sizes allow more frequent iterations through the PDCA cycle



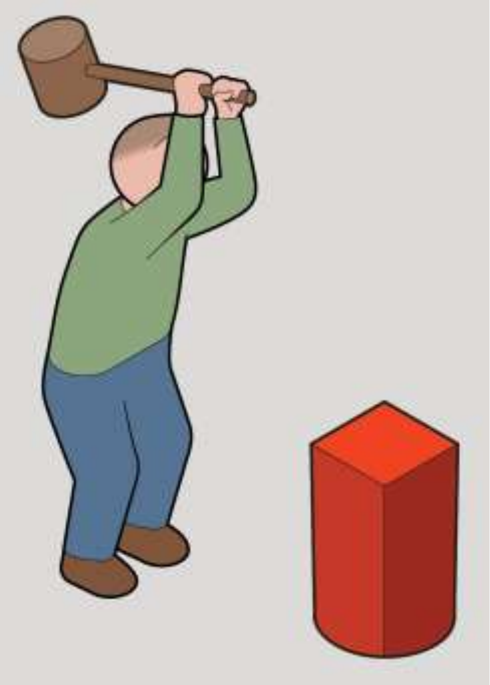
Build incrementally with fast, integrated learning cycles

Integration points are riddled with assumptions that need to be validated (uncertainties that need to be converted into knowledge).

Integration is often pushed off until late in the lifecycle (process guidelines, technical difficulty, siloed focus, “it should work”)

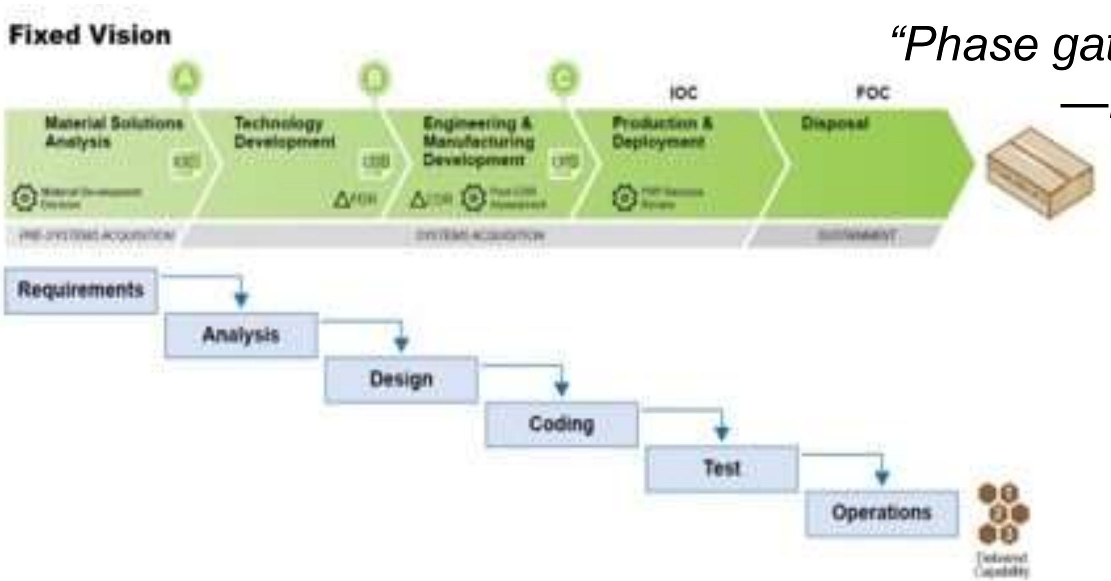
SAFe incorporates frequent integration points into the fabric of the framework.

Frequent integration uncovers and mitigates risk.



5 of 9

Base milestones on objective evaluation of working systems

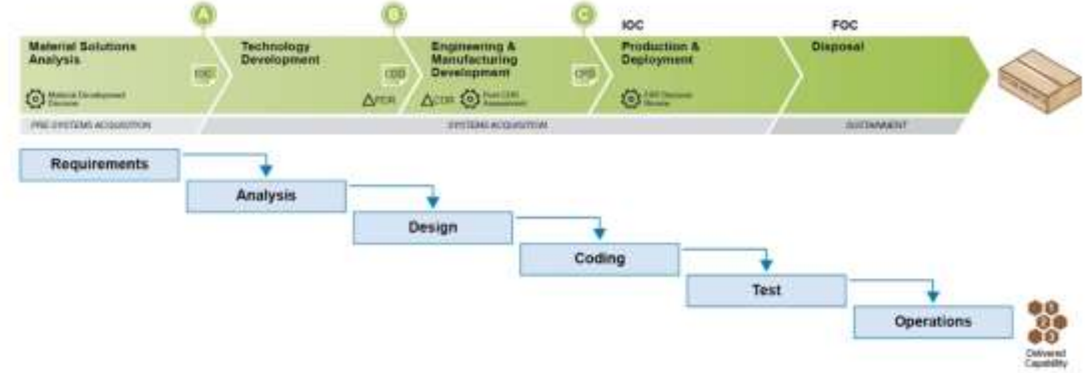


“Phase gates are evil .”
—Dr. Allen C. Ward

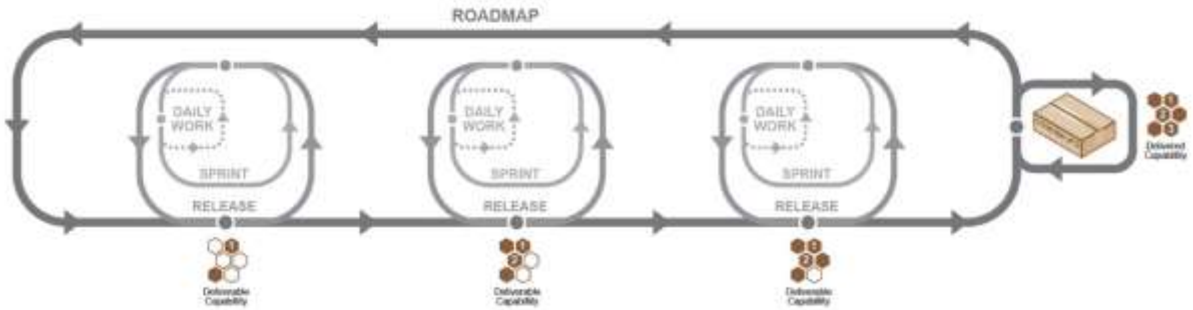
“There was in fact no correlation between exiting phase gates on time and project success… the data suggested the inverse might be true.”

The Lean Machine

Base milestones on objective evaluation of working systems



Assumes linear path to solution
Assumes learning can be compartmentalized & front-end loaded in advance of implementation
Contractually oriented (each phase is a mini-contract)



Assumes path to solution evolves over time
Assumes learning occurs throughout development and that the highest quality of knowledge is achieved by working code
Oriented towards ongoing discovery and negotiation

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

Why “Visualize and Limit WIP, Reduce Batch Sizes, and Manage Queue Lengths”?

To Optimize Flow and Reduce Time to Value

Batch Size and Flow: Hands-On Exercise

Break into groups of five people

- 4 coin flippers and 1 time-keeper
- 10 coins per group: 1 nickel and 9 pennies

Coin-flipper instructions:

- Flip the nickel first
- Flip each coin till you get heads
- One coin at a time
- Coin must be airborne
- Flip all 10 coins to get all heads
- Then pass all 10 coins to the next coin flipper

Time-keeper instructions:

- Measure the time it takes from the first flip of the first flipper to the last flip of the last flipper

Batch Size and Flow: Hands-On Exercise – Small Batch

6 of 9

Coin-flipper instructions:

- Flip the nickel first
- Flip each coin till you get heads
- One coin at a time
- Coin must be airborne
- Flip the 1st coin (i.e. – the nickel) till you get heads
- Once you get heads, pass this coin to the next coin flipper.
- Proceed to flip the next coin till you get heads, and so on.

Time-keeper instructions:

- Measure the time it takes the nickel to go from the first flip of the first flipper to the last flip of the last flipper
- Measure the time it takes all of the coins to go from the first flip of the first flipper to the last flip of the last flipper

Little’s Law – Average Longer Queues Lead to Average Longer Wait Time

6 of 9

Average wait time = average queue length / average processing rate

WQ = LQ / (#of items / time period) OR

WQ = LQ * (time period / # of items)

If the average time to create a latte is 2 minutes and the average queue of people waiting for a lattes is 8 deep, the average wait time for a latte is 16 minutes

Little’s Law and Batch Size

6 of 9

Little’s Law argues for reduced batch size as a way to speed up value delivery

BUT ..

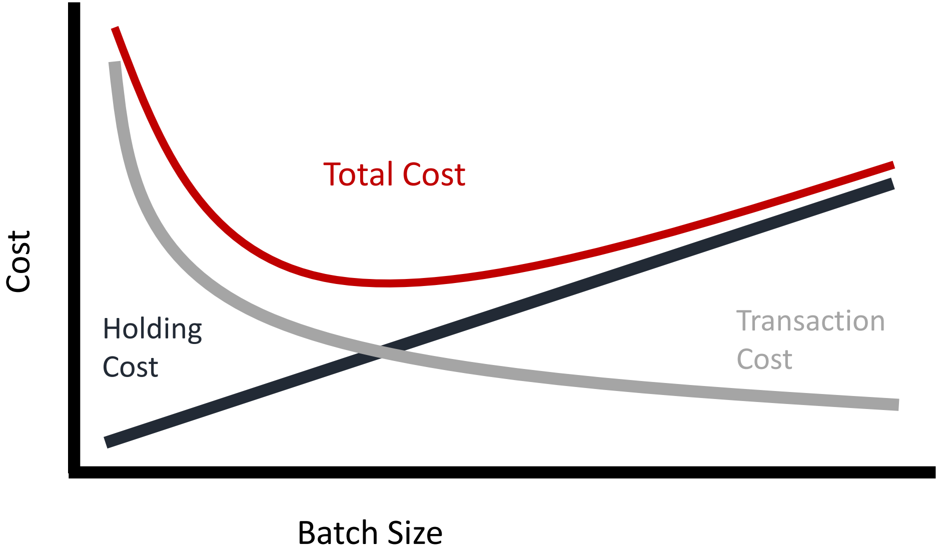
Holding cost and transaction cost strongly influence the economies of batch size

Transaction cost – The cost of sending a batch to the next process

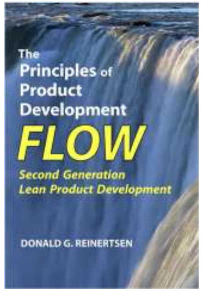
Holding cost – The cost of not sending it

Principles of Product Development Flow,
Don Reinertsen

Finding Optimal Batch Size



- Higher holding costs favor lower batch size
- Higher transaction costs favor higher batch size


The Principles of Product Development
FLOW
Second Generation Lean Product Development
DONALD G. REINERTSEN

Principle B12: “Reducing transaction cost per batch lowers overall costs.”

An Example Would Come in Handy Right About Now

6 of 9

Key Take-Away

Reduced transaction costs are a major enabler of smaller batch size and can dramatically alter the economies of how we work

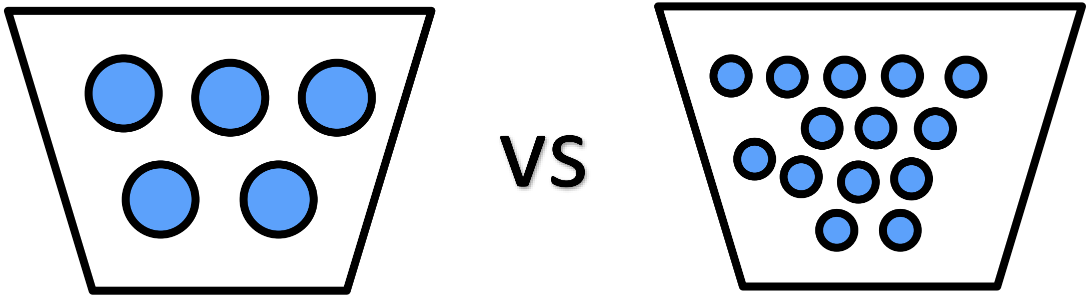
Reducing transaction costs example

https://youtu.be/RRy_73ivcms

2:09

Story Splitting is an Enabler of Smaller Batch Size Too

6 of 9



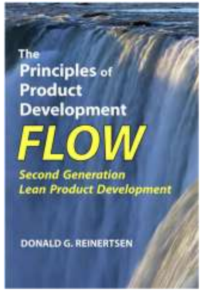
Splitting stories requires engineering judgment

Queues are the Root of Many Development Evils

The Principle of Queueing Waste: Queues are the root cause of the majority of economic waste in product development.

Queues create:

- Longer Cycle Time
- Increased Risk
- More Variability
- More Overhead
- Lower Quality
- Less Motivation


The Principles of Product Development
FLOW
Second Generation Lean Product Development
DONALD G. REINERTSEN

Principle Q2: "Queues are the root cause of the majority of economic waste in product development."

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

Apply Cadence, Synchronize with Cross-Domain Planning

7 of 9

“Cadence causes events to happen at regular time intervals

Synchronization causes multiple events to happen at the same time”

In SAFe we produce work on a cadence and we synchronize on a cadence

Principles of Product Development Flow,
Don Reinertsen

Apply Cadence, Synchronize with Cross-Domain Planning

7 of 9

Cadence Provides Established Pace and Timing of Work

- Predictability and lower overhead for key events
- Sets expectations around acceptance of new work
- Regularly scheduled integration points
- Protected time to be productive
- “Culture is realized in the habits of the organization”

Cadence is the basis of the Scrum lifecycle – for allocating work and producing increments

“Leading SAFe”, 4.5.0.1, Module 6

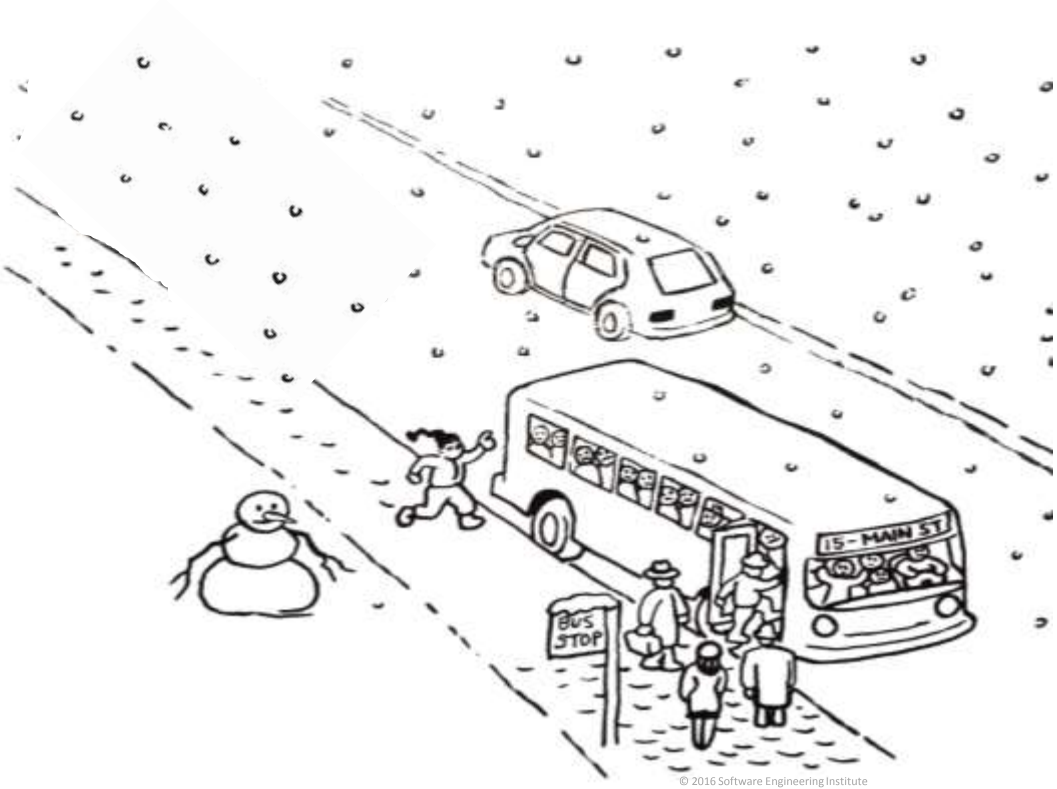
Apply Cadence, Synchronize with Cross-Domain Planning

Regular Synchronization Avoids Surprises

- Manage dependencies
- Align stakeholder perspectives
- Synchronize technical components

Synchronization is critical for scaling – to ensure that parallel activities with dependencies work together to provide systemic value

Cadence Enhances Predictability



© 2016 Software Engineering Institute

A Late Bus:

- Makes people scramble to get aboard
- They don’t know when the next one will get here

Then the next bus comes along empty

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

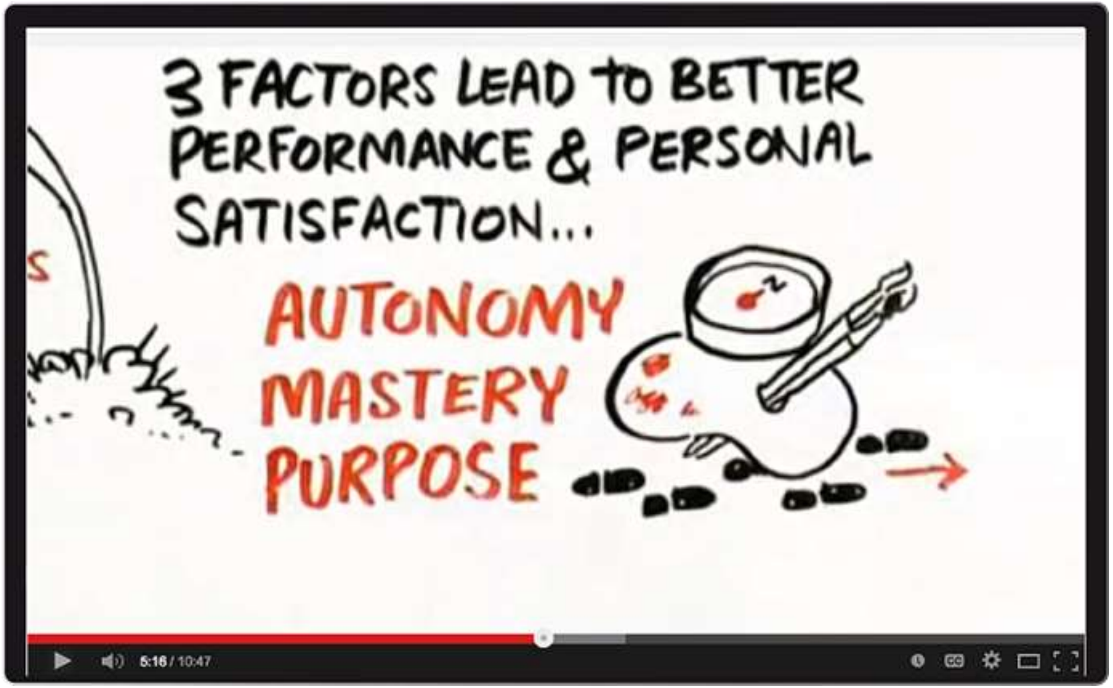
#7 - Apply cadence, synchronize with cross-domain planning

➡ #8 - Unlock the intrinsic motivation of knowledge workers

#9 - Decentralize decision-making

Unlock the Intrinsic Motivation of Knowledge Workers

8 of 9



3 FACTORS LEAD TO BETTER PERFORMANCE & PERSONAL SATISFACTION...

AUTONOMY

MASTERY

PURPOSE

RSA Animate

“Drive: The Surprising Truth About What Motivates Us”

Daniel H. Pink

https://youtu.be/u6XAPnuFjJc

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

The Principle of Small Decisions 9 of 9

“Many companies assume that the path to economic success lies on making a handful of big economic choices correctly.

This leads them to concentrate decision-making at high levels of authority and to design system that support high-level decision-making.

While big decisions are important, this bias means that most companies have weak systems to ensure that the many small economic decisions are made correctly.

Collectively these small decisions have enormous economic impact”

Principles of Product Development Flow,
Don Reinertsen

Making Decentralized Decision-Making Work 9 of 9

- Decision Rules
- Distributed Expertise
- Rapid Feedback
- Commander’s Intent

Commander’s Intent

9 of 9

“… succinctly describes what constitutes success for the operation. It includes the operation’s purpose, key tasks, and the conditions that define the end state. … A clear commander’s intent enables a shared understanding and focuses on the overall conditions that represent mission accomplishment. During execution, the commander’s intent spurs disciplined initiative.”

http://usacac.army.mil/CAC2/MilitaryReview/Archives/English/MilitaryReview_20131231_art011.pdf

Not All Decisions Should Be Decentralized

9 of 9

Define the economic logic behind a decision; empower others to actually make them.

Centralize	**De-centralize** everything else
Infrequent - Not made very often and usually not urgent *(ex: internationalization strategy)*	**Frequent and common** - Routine, every day decisions *(ex: team and program backlog)*
Long lasting - Once made, highly unlikely to change *(ex: common technology platform)*	**Time critical** - High cost of delay *(ex: point release to customer)*
Significant economies of scale - Provides large and broad economic benefit *(ex: compensation strategy)*	**Require local information** - Specific, and local technology or customer context is required *(ex: Feature criteria)*

Decentralizing decision making in nuclear submarine command
https://youtu.be/OqmdLcyES_Q
9:47

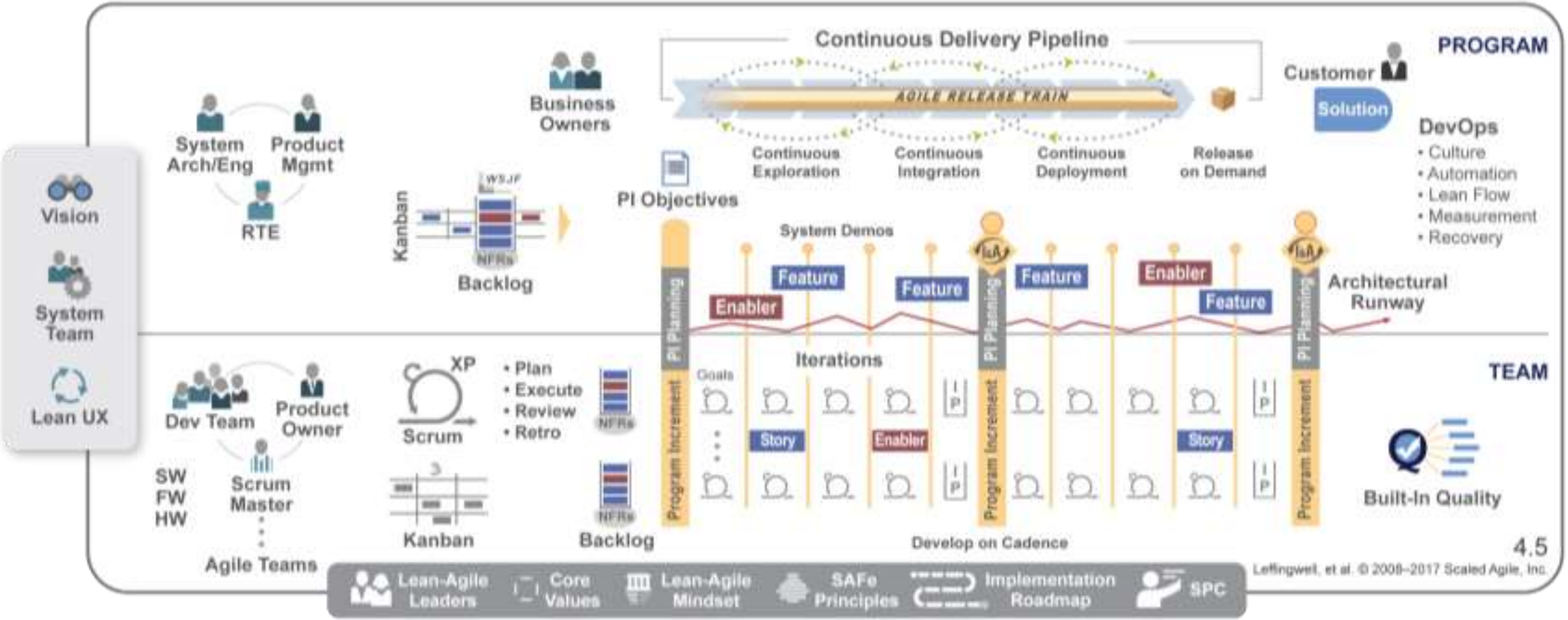
Implement Lean-Agile practices

Knowledge for people building the world's most important systems

SAFe® is a freely revealed knowledge base of integrated, proven patterns for enterprise Lean-Agile development.

scaledagileframework.com

Essential SAFe provides the basis for success



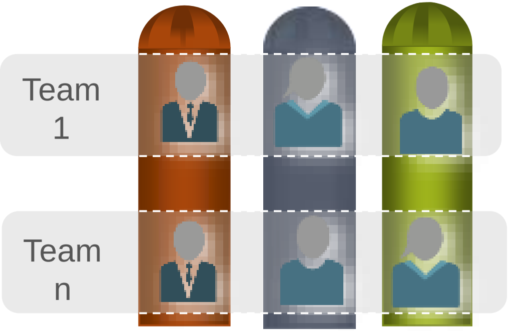
SCALED AGILE

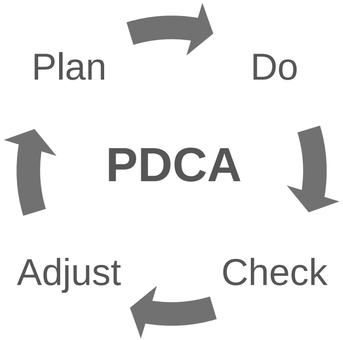
Nothing beats an Agile Team

- Cross-functional, self-organizing entities that can define, build and test a thing of value

- Applies basic scientific practice: Plan—Do—Check—Adjust

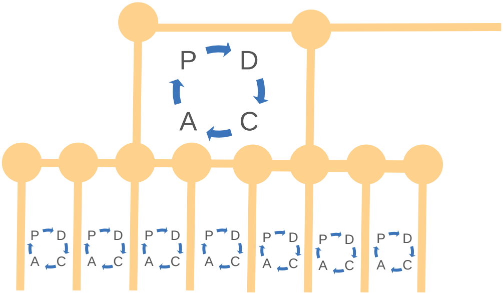
- Delivers value every two weeks

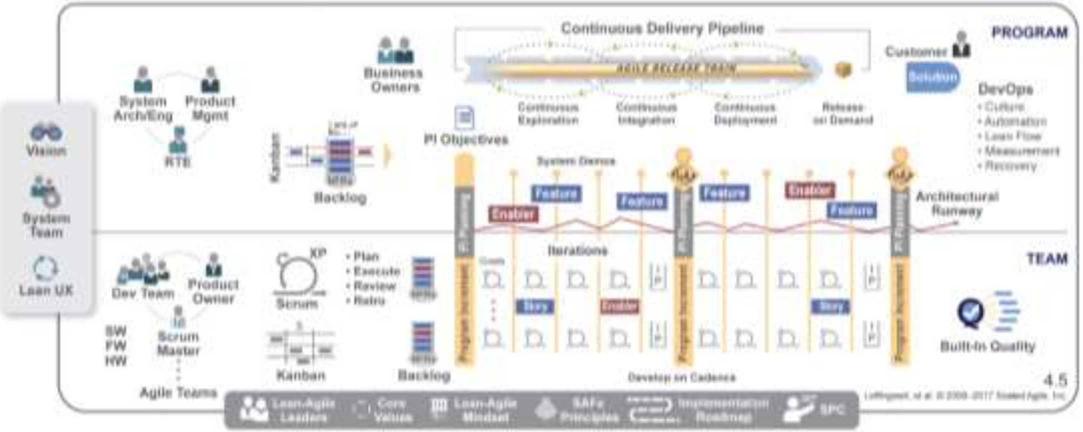




Except a team of Agile Teams

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



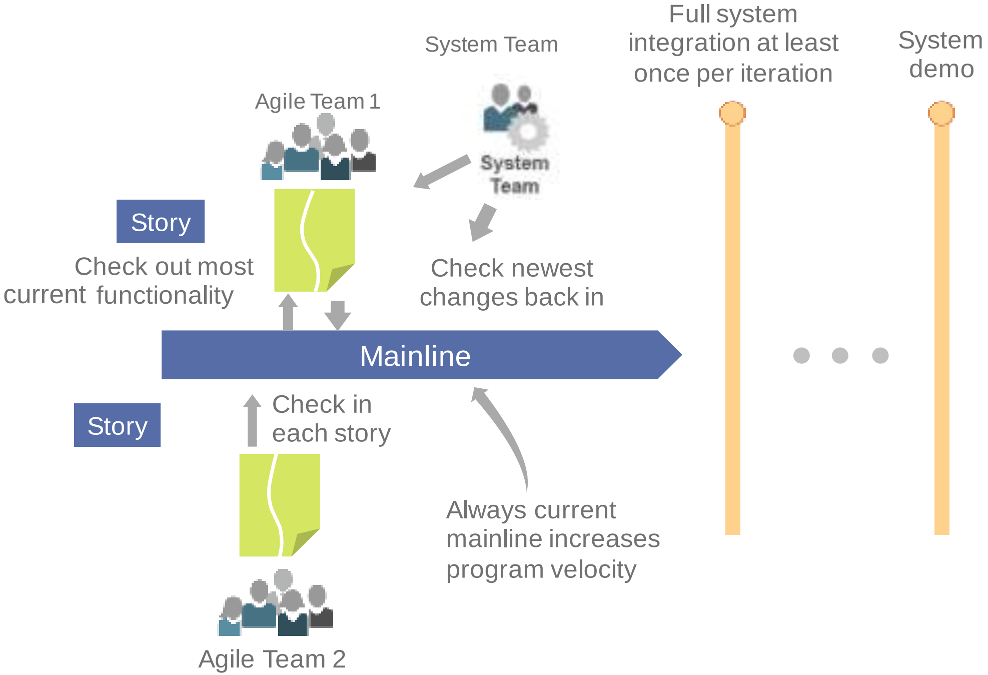


That integrates frequently

Integration points control product development.

— Dantar Oosterwal, The Lean Machine

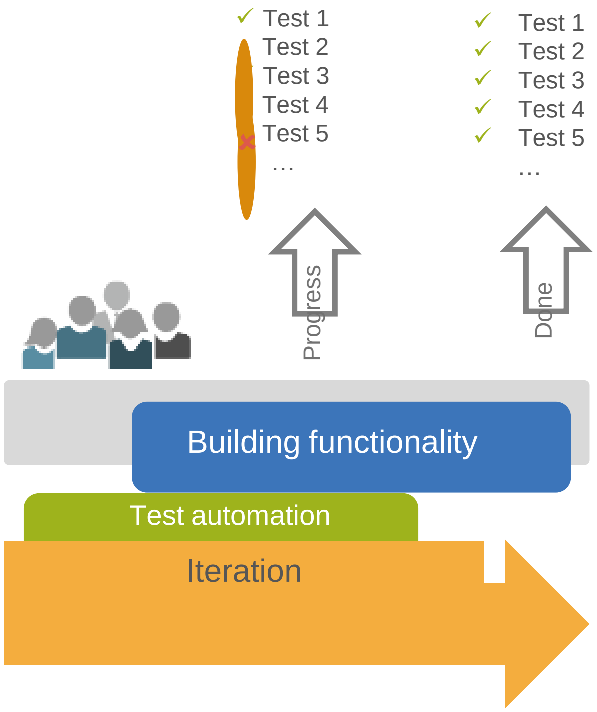
- Avoid physical branching for software
- Frequently integrate hardware branches
- Use development by intention in for inter-team dependencies


System Team
Full system integration at least once per iteration
System demo
Agile Team 1
System Team
Story
Check out most current functionality
Check newest changes back in
Mainline
Story
Check in each story
Always current mainline increases program velocity
Agile Team 2

Applies test automation

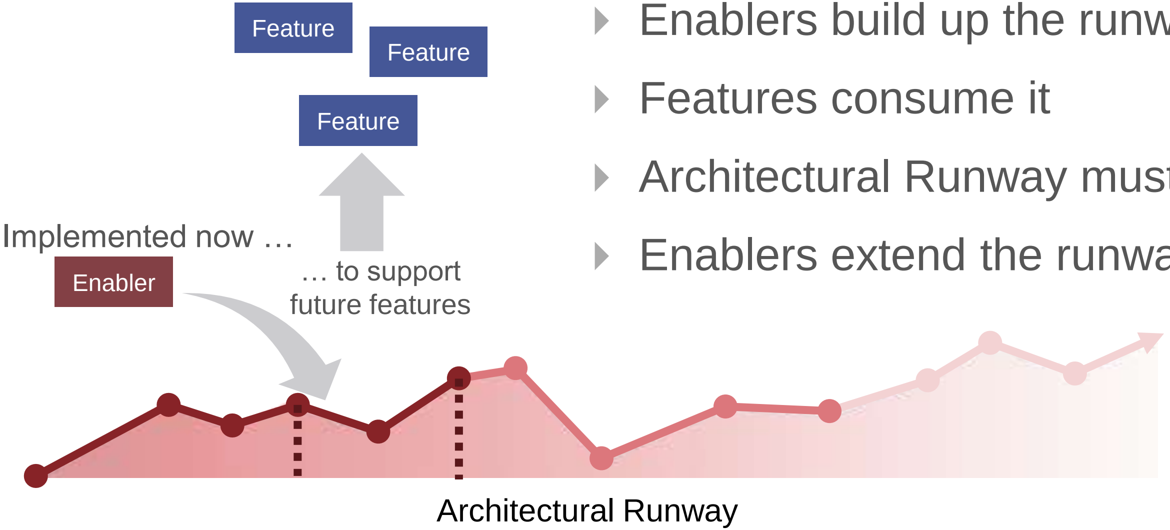
Test automation supports rapid regression testing

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



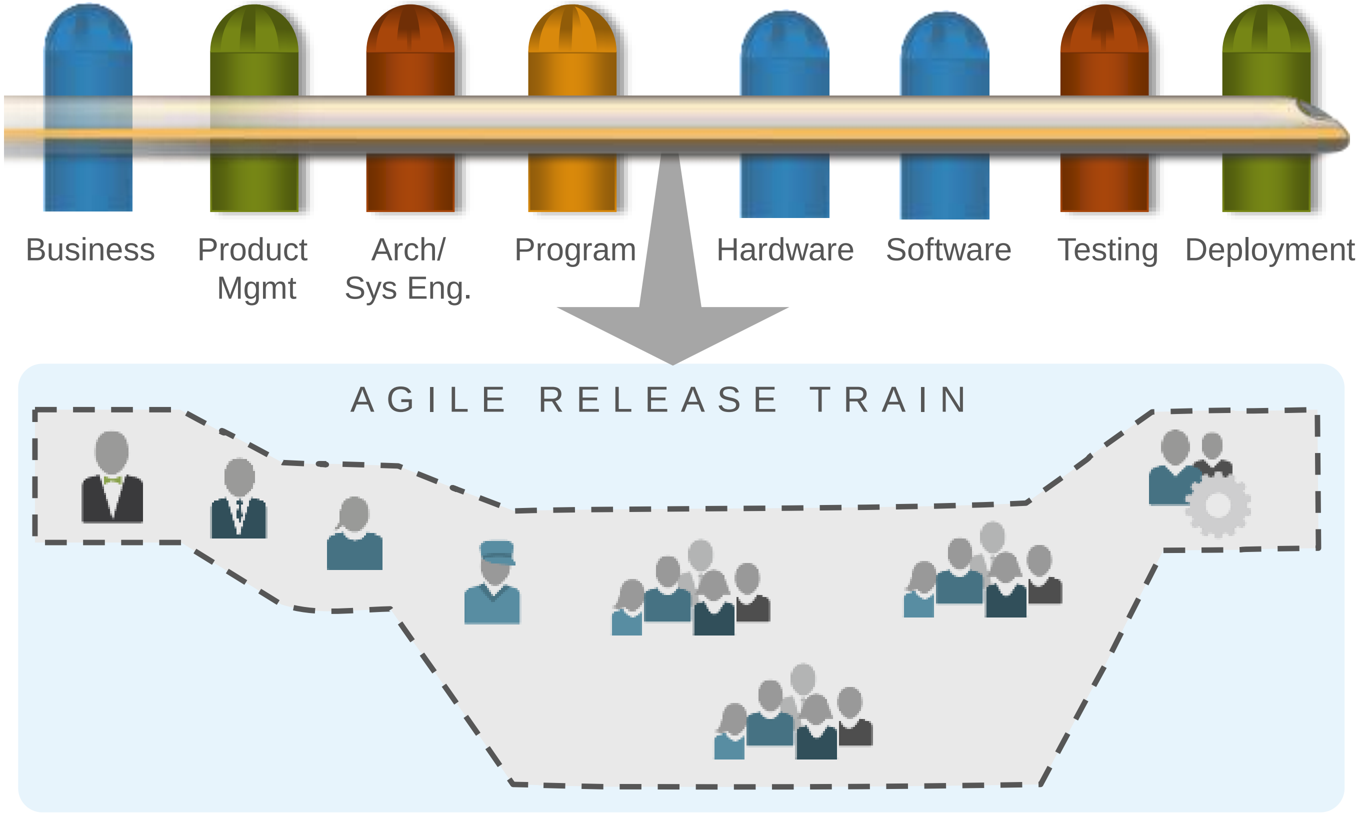
With some Architectural Runway

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



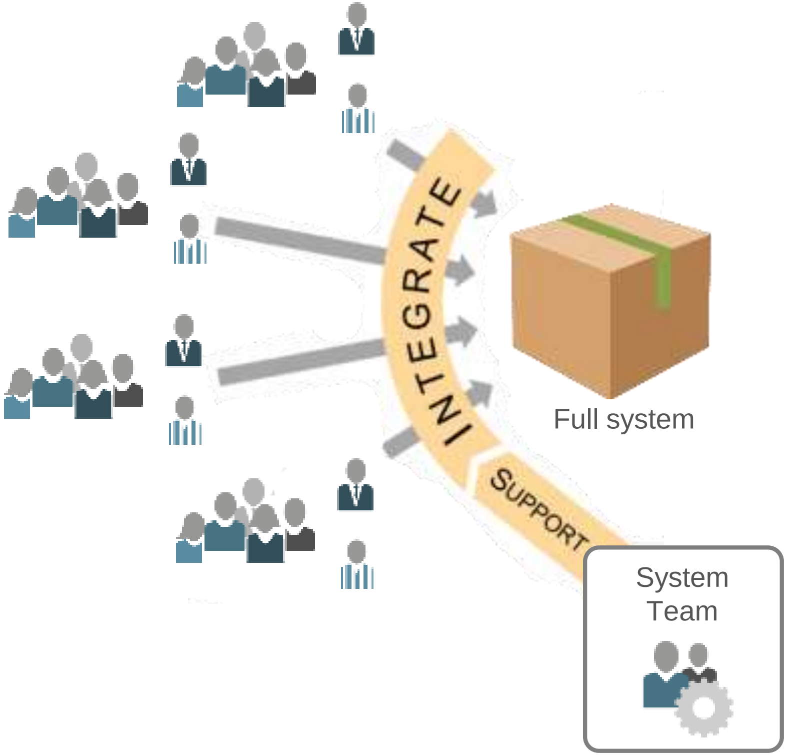
Synchronizes with PI Planning

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

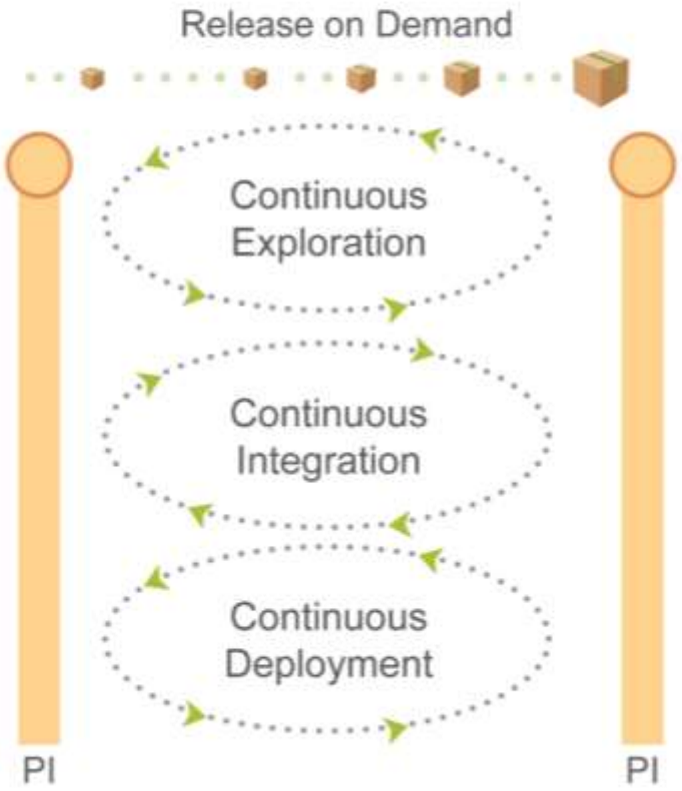
For a short video PI planning example, see: https://youtu.be/ZZAtl7nAB1M

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



Release on Demand

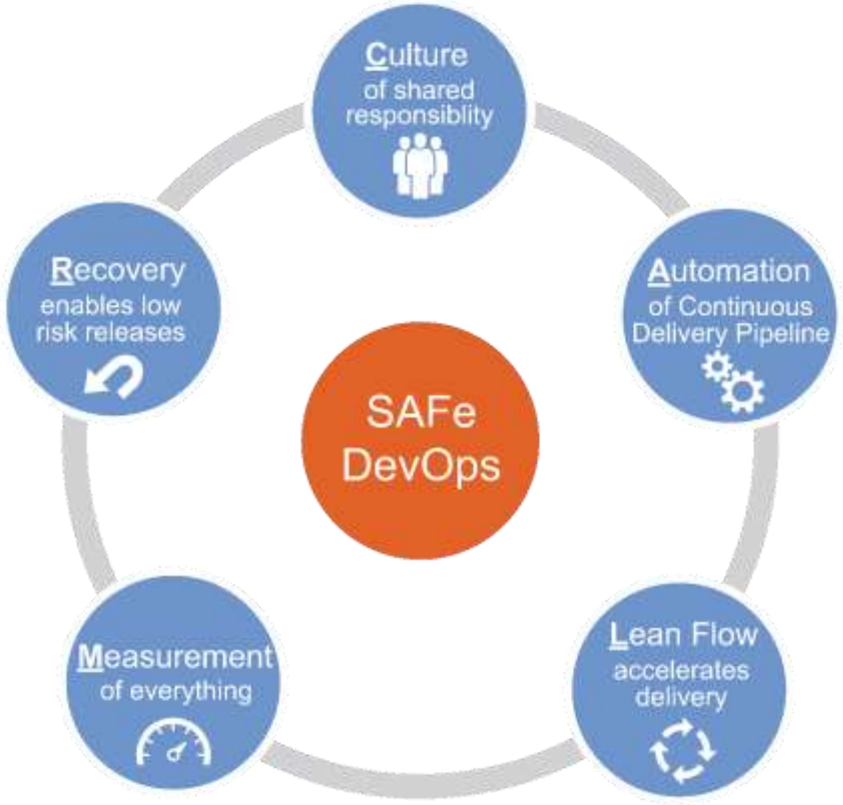
Continuous Exploration

Continuous Integration

Continuous Deployment

PI

PI



SAFe DevOps

Culture of shared responsiblity

Automation of Continuous Delivery Pipeline

Lean Flow accelerates delivery

Measurement of everything

Recovery enables low risk releases

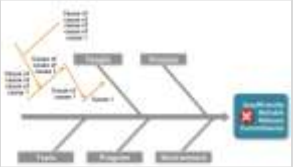
Inspects and Adapts every PI

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

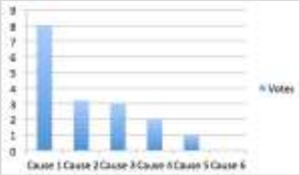
Agree on the problem to solve

Insufficiently reliable release commitments?

Apply root cause analysis (+ five whys)



Identify the biggest root cause using Pareto Analysis

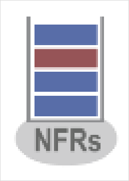


Restate the new problem for the biggest root cause

Insufficient architectural runway

Brainstorm solutions

Identify improvement Backlog items

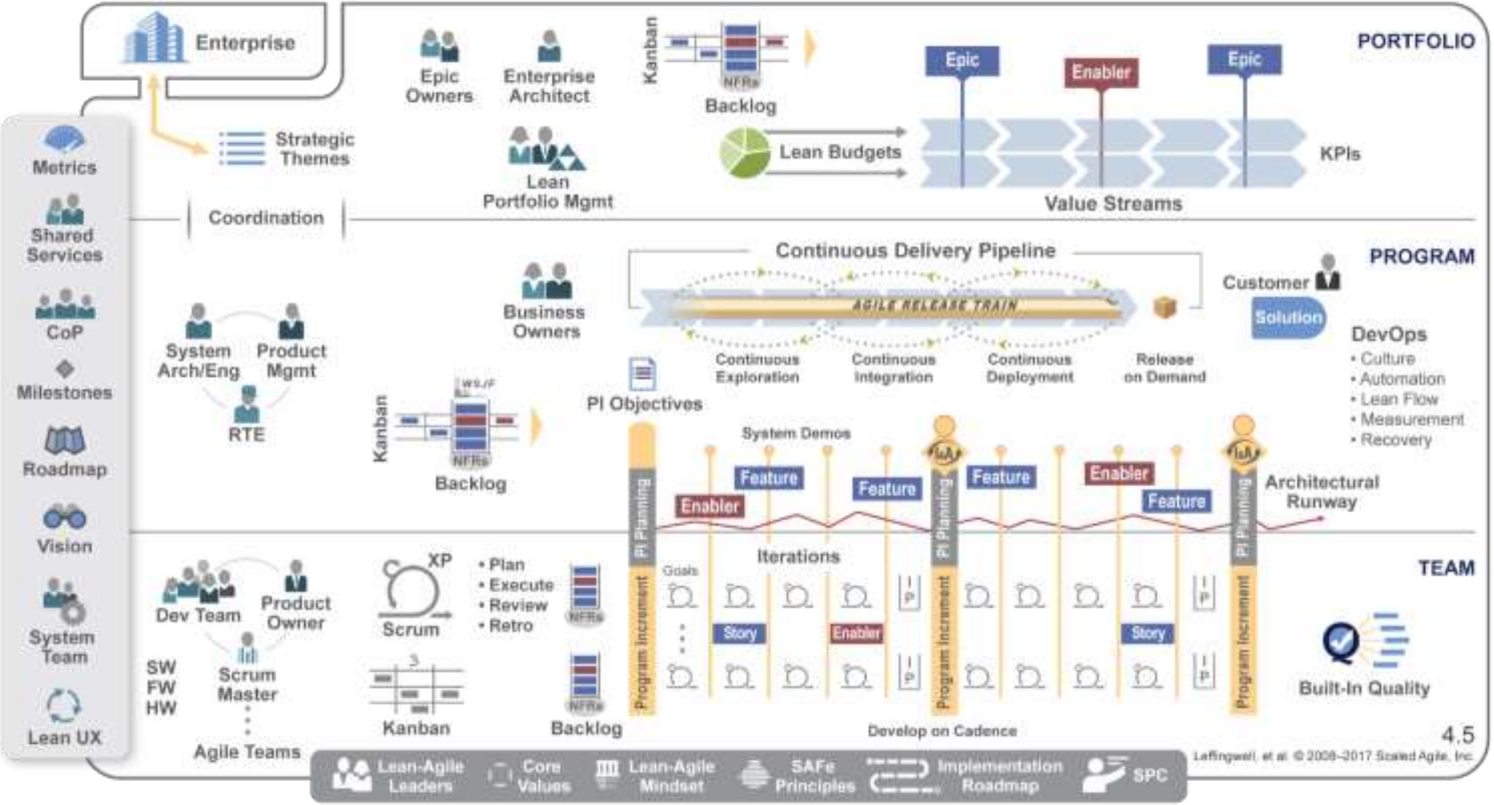


Ten Essential SAFe Elements

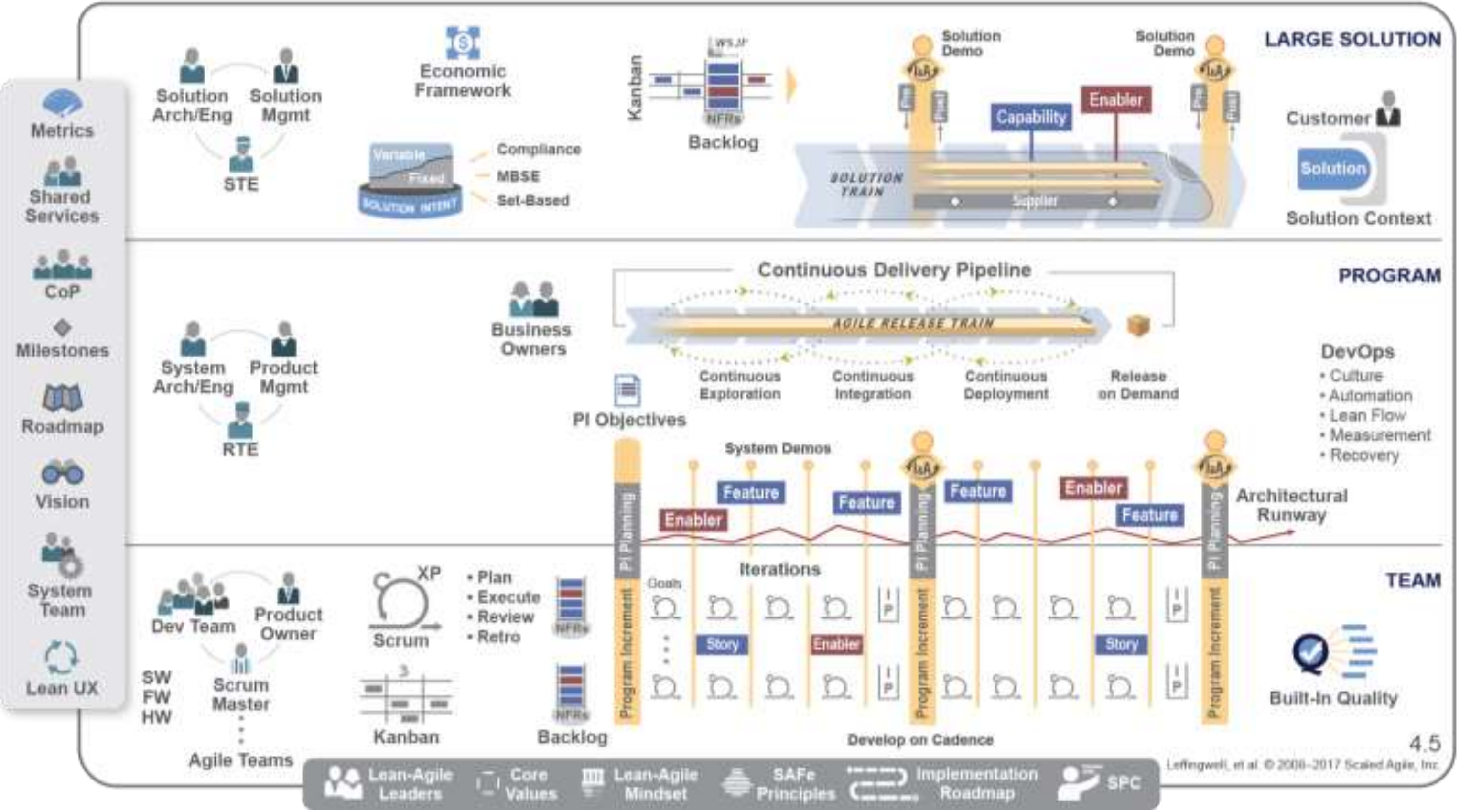
1. SAFe Lean-Agile Principles
2. Real Agile Teams and Trains
3. Cadence and Synchronization
4. PI Planning
5. DevOps and Releasability
6. System Demo
7. Inspect & Adapt
8. IP Iteration
9. Architectural Runway
10. Lean-Agile Leadership

Scaled Agile Inc: Essential SAFe Toolkit v4.5.0

Portfolio SAFe aligns strategy and execution

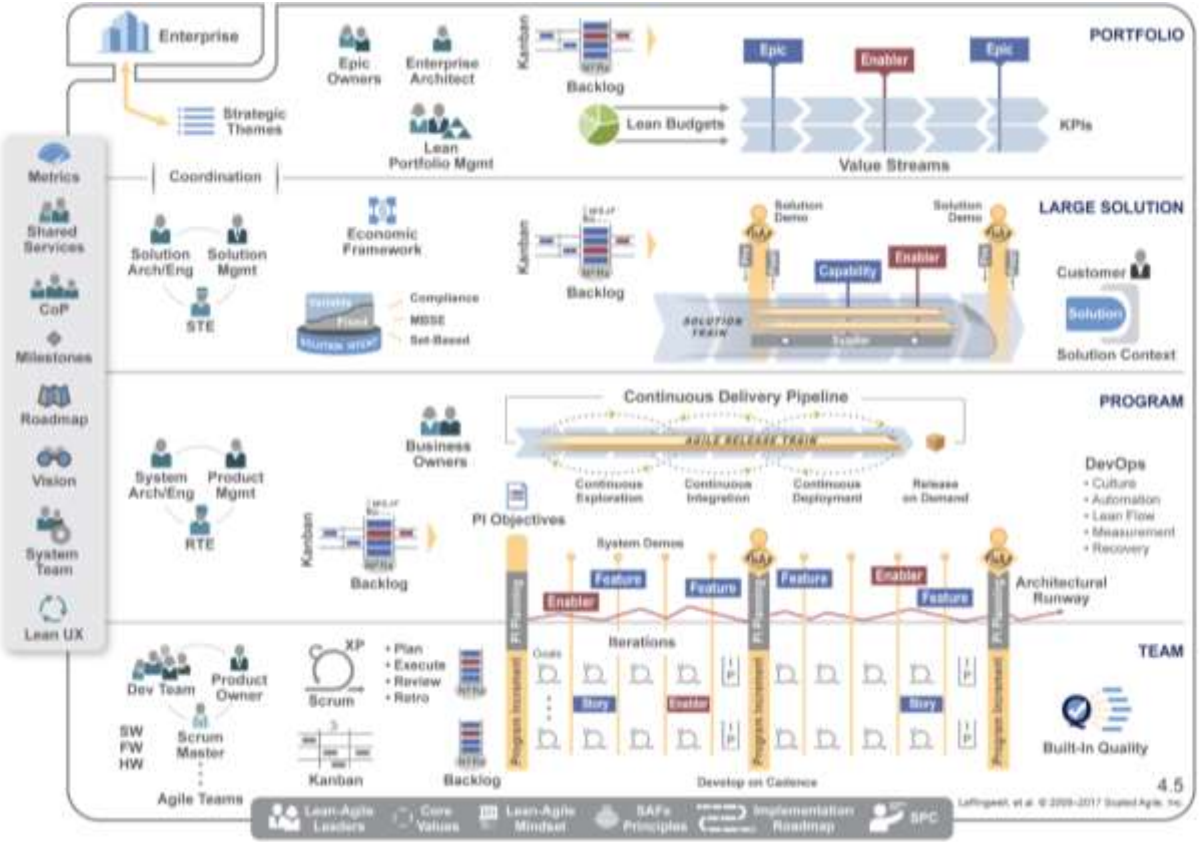


Large Solution SAFe coordinates ARTs with a Solution Train



SCALED AGILE

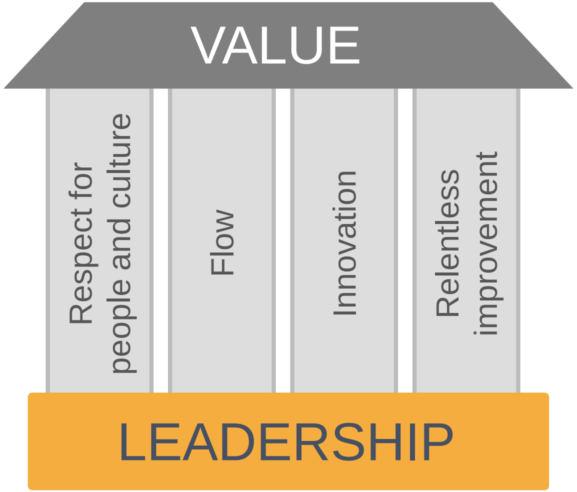
Full SAFe for large enterprises



SCALED AGILE

Lead the implementation

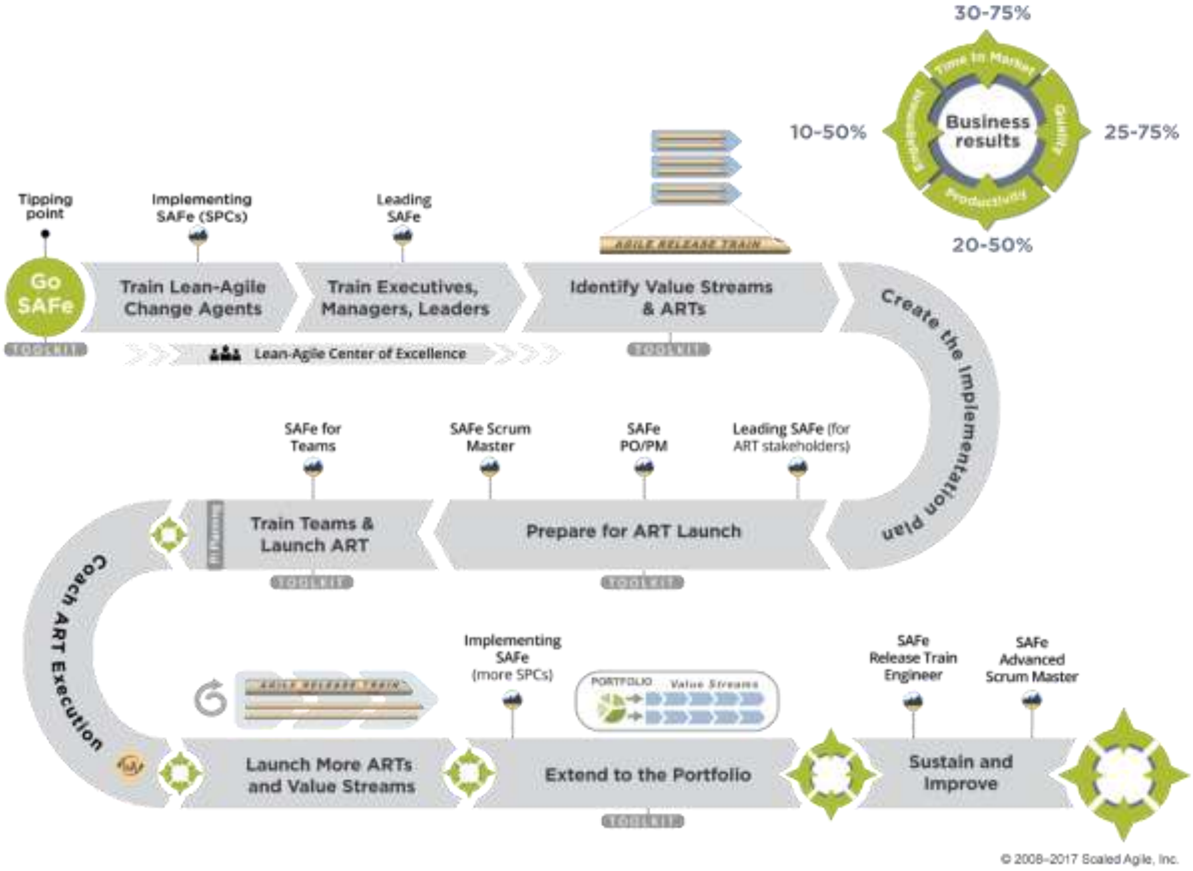
Leadership foundation



People are already doing their best; the problems are with the system. Only management can change the system.

—W. Edwards Deming

Implementation Roadmap

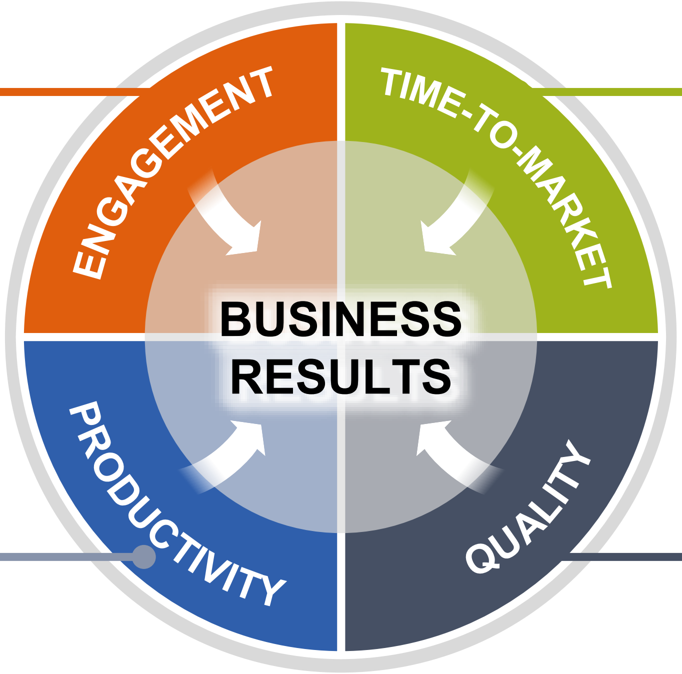


Get results

Business results

10 – 50% happier, more motivated employees

30 – 75% faster time-to-market

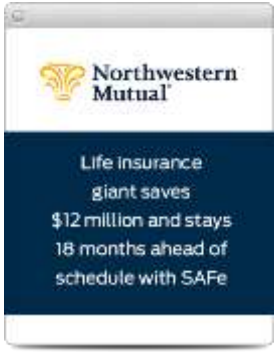


20 – 50% increase in productivity

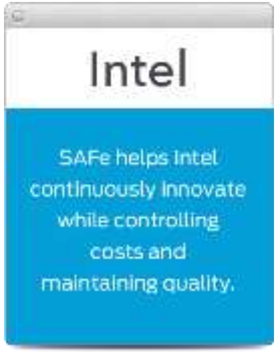
25 – 75% defect reduction

See ScaledAgileFramework.com/**case-studies**

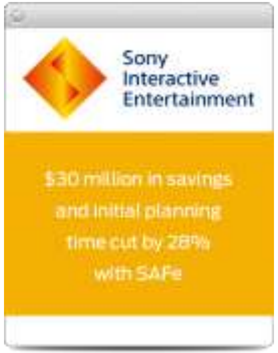
Financial Services / Electronics / Software / Telecom / Retail & Distribution / Government / Healthcare / Insurance / Medical Technology / Pharmaceutical / Media / Manufacturing / COTS Software / Customer Care & Billing / Outsourcing


Northwestern Mutual

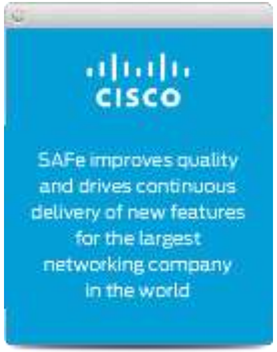
Life insurance giant saves $12 million and stays 18 months ahead of schedule with SAFe


Intel

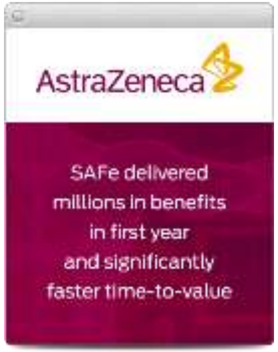
SAFe helps Intel continuously innovate while controlling costs and maintaining quality.


Sony Interactive Entertainment

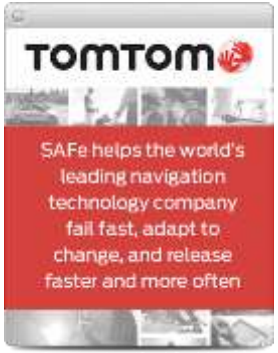
$30 million in savings and initial planning time cut by 28% with SAFe


cisco

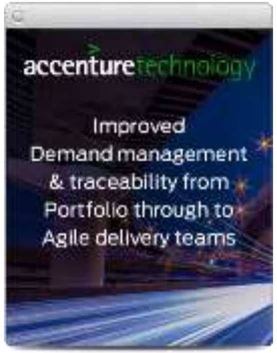
SAFe improves quality and drives continuous delivery of new features for the largest networking company in the world


AstraZeneca

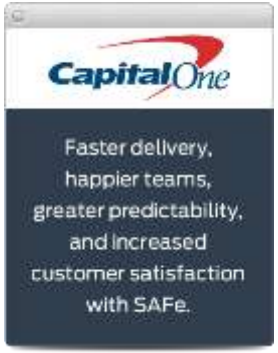
SAFe delivered millions in benefits in first year and significantly faster time-to-value


TomTom

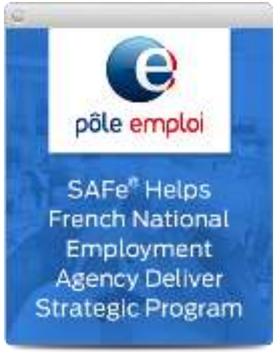
SAFe helps the world's leading navigation technology company fail fast, adapt to change, and release faster and more often


accenture technology

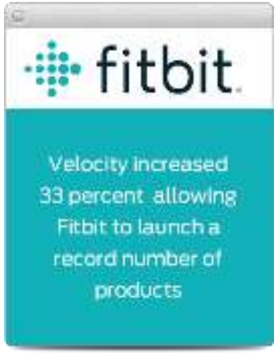
Improved Demand management & traceability from Portfolio through to Agile delivery teams


Capital One

Faster delivery, happier teams, greater predictability, and increased customer satisfaction with SAFe.


pôle emploi

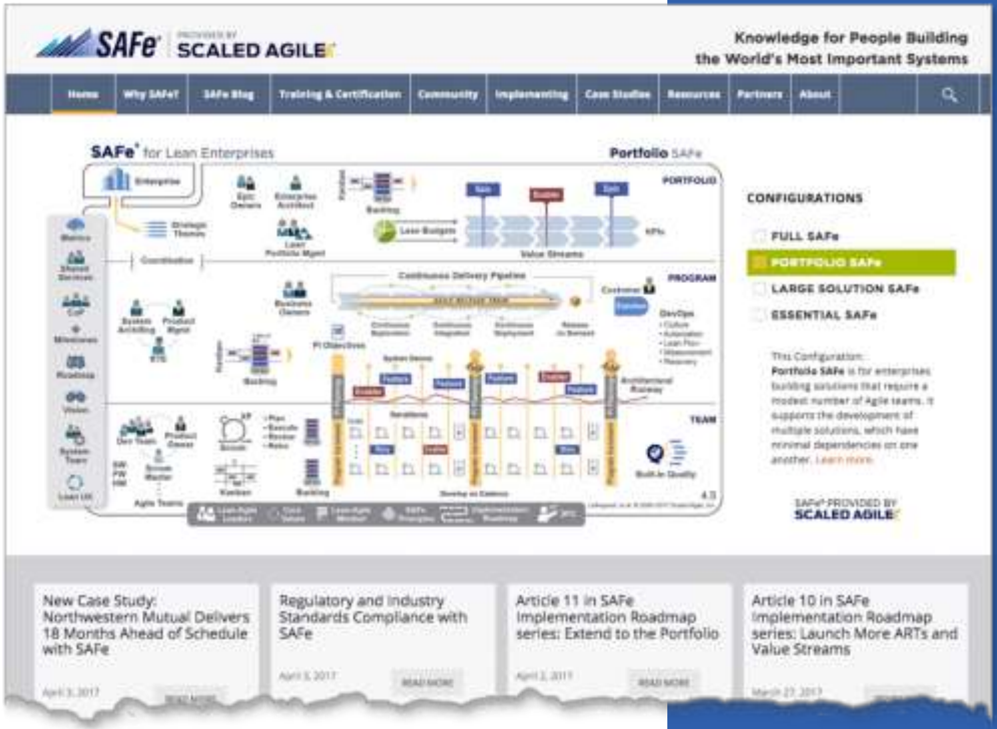
SAFe® Helps French National Employment Agency Deliver Strategic Program


fitbit

Velocity increased 33 percent allowing Fitbit to launch a record number of products

See ScaledAgileFramework.com/**case-studies**

Gain the Knowledge



Knowledge for People Building
the World's Most Important Systems

CONFIGURATIONS

□ FULL SAFe

■ PORTFOLIO SAFe

□ LARGE SOLUTION SAFe

□ ESSENTIAL SAFe

This Configuration:
Portfolio SAFe is for enterprises building solutions that require a modest number of Agile teams. It supports the development of multiple solutions, which have minimal dependencies on one another. Learn more.

SAFe® PROVIDED BY
SCALED AGILE

New Case Study: Northwestern Mutual Delivers 18 Months Ahead of Schedule with SAFe
April 3, 2017
READ MORE

Regulatory and Industry Standards Compliance with SAFe
April 3, 2017
READ MORE

Article 11 in SAFe Implementation Roadmap series: Extend to the Portfolio
April 2, 2017
READ MORE

Article 10 in SAFe Implementation Roadmap series: Launch More ARTs and Value Streams
March 27, 2017

ScaledAgileFramework.com

Explore the SAFe knowledge base and find free resources:

- Articles
- Guidance
- Presentations
- White papers
- Videos
- Case studies

ScaledAgile.com

Find SAFe

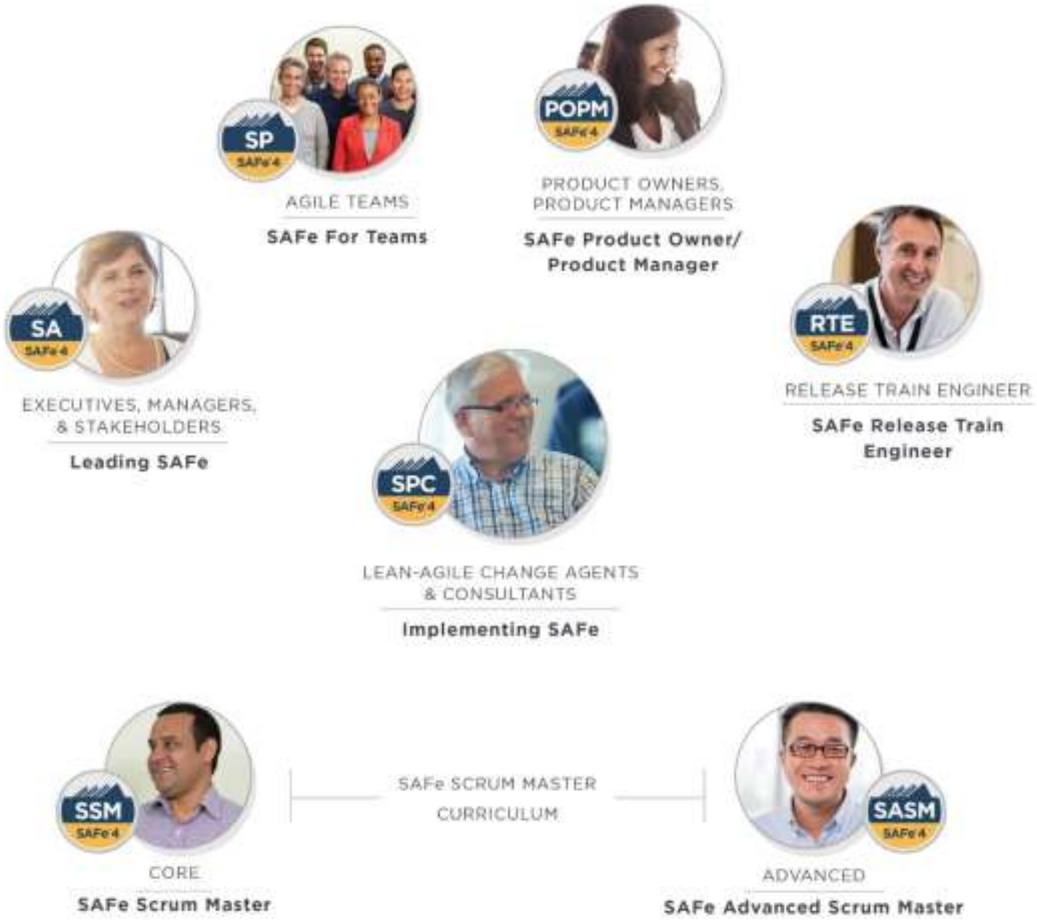
training worldwide

CORE

- Leading SAFe
- SAFe for Teams
- SAFe Scrum Master
- SAFe PO/PM

ADVANCED

- SAFe Advanced Scrum Master
- Implementing SAFe
- SAFe Release Train Engineer



AGILE TEAMS

SAFe For Teams

PRODUCT OWNERS, PRODUCT MANAGERS

SAFe Product Owner/ Product Manager

EXECUTIVES, MANAGERS, & STAKEHOLDERS

Leading SAFe

RELEASE TRAIN ENGINEER

SAFe Release Train Engineer

LEAN-AGILE CHANGE AGENTS & CONSULTANTS

Implementing SAFe

SAFe SCRUM MASTER CURRICULUM

CORE

SAFe Scrum Master

ADVANCED

SAFe Advanced Scrum Master

SAFe® for Lean Enterprises

180,000

SAFe certified professionals in 100+ countries

150

Scaled Agile Partners in 50 countries

70% US *Fortune* 100 enterprises have SAFe certified professionals

2 million

Annual visitors to SAFe and Scaled Agile websites

PLEDGE 1%

Pledged 1%

Scaled Agile stock equity & employee time to Pledge 1% campaign

SAFe:

Freely available knowledge base, downloads, and resources for people building the world's most important software and systems

SAFe SUMMIT

Freely Available

SAFe's knowledge base is freely available at **scaledagileframework.com**

Configurable

SAFe is able to accommodate enterprises of all sizes and industries

Fastest Growing Method

- **11th Annual State of Agile Report by VersionOne**
- **2017 Scaling Agile Report by cPrime**

SAFe cited as preferred solution for scaling Agile, making SAFe the most popular scaling method above Scrum, Scrum of Scrums, and all other frameworks

SAFe® PROVIDED BY SCALED AGILE®

Summary- Agile and Lean Methods and Practices

The family of Agile methods has grown since 2000 to incorporate methods that address team, project, and enterprise levels of scaling

Scrum focuses on the team management aspects of Agile software development and is the most commonly practiced Agile methodology.

SAFe-Scaled Agile Framework, is the most frequently adopted scaling framework for DoD contractors, so far.

Many of the scaling approaches, including SAFe, leverage Scrum practices as the team component of their methodology.

Hybrids of multiple methods are common practice in both industry and government

Summary- Agile and Lean Values and Principles

Embracing an Agile / Lean mindset and perspective is the key to successfully implementing Agile at the small team, program and enterprise levels

- This is why Agile, Lean and SAFe place so much emphasis on underlying values and principles

Multiple methods and practices can be used to implement the values and principles

- But methods and practices alone will not lead to success.

Agile Awareness Workshop for NC3

Module 3: Enabling an Agile/Lean Culture in Regulated Settings

July 2019
SEI Continuous Lifecycle Solutions Initiative

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.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM19-0696

Topics the SEI will address in this course include:

- **Introduction**
- **Basic Agile/Lean Concepts**
  - ➢ Today’s Landscape
  - ➢ Agile Tenets, Principles and Concepts
  - ➢ Scaling Agile
  - ➢ SAFe Foundations
- ***Enabling an Agile/Lean Culture in Regulated Settings***
- **Summary**

Bottom Line Up Front

Culture Trumps Technical Issues

Culture change due to Agile is an issue throughout industry and government

Small batches, small teams, decentralized decision making….

Adoption Strategies for Other Practices Also Work for Agile

Understand your adoption population and how quickly they can adopt

Support your adopters with useful transition mechanisms to help them move forward

Go In With Your Eyes Open

Adopting Agile is a multi-dimensional challenge

Use Agile RFA or a similar multi-dimensional self-assessment to understand YOUR adoption challenges

Culture Trumps All

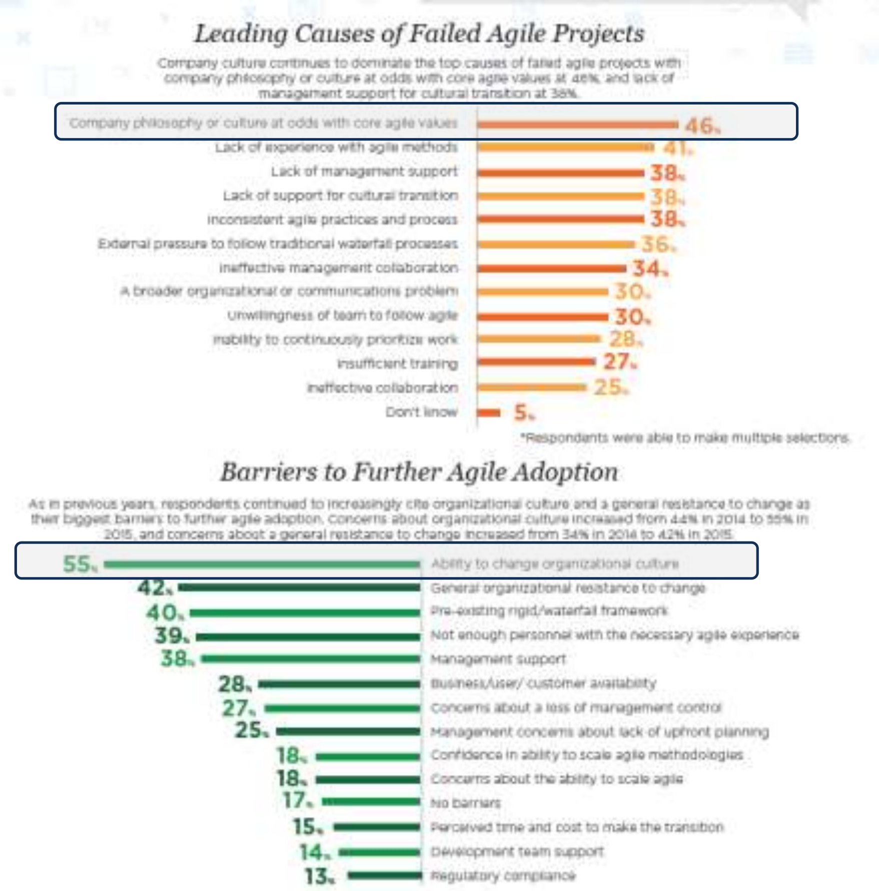
Culture Trumps Technical Issues

- “Culture eats strategy for breakfast” - Peter Drucker

Organizational culture eats strategy for breakfast, lunch and dinner
Culture
Strategy

Source: SAFe 4.5 and Torben Rick https://www.torbenrick.eu/blog/culture/organisational-culture-eats-strategy-for-breakfast-lunch-and-dinner/

Culture Change is an Issue Throughout Industry



Leading Causes of Failed Agile Projects

Company culture continues to dominate the top causes of failed agile projects with company philosophy or culture at odds with core agile values at 46%, and lack of management support for cultural transition at 38%.

*Respondents were able to make multiple selections.

Barriers to Further Agile Adoption

As in previous years, respondents continued to increasingly cite organizational culture and a general resistance to change as their biggest barriers to further agile adoption. Concerns about organizational culture increased from 44% in 2014 to 55% in 2015, and concerns about a general resistance to change increased from 34% in 2014 to 42% in 2015.

55% Ability to change organizational culture
42% General organizational resistance to change
40% Pre-existing rigid/waterfall framework
39% Not enough personnel with the necessary agile experience
38% Management support
28% Business/user/ customer availability
27% Concerns about a loss of management control
25% Management concerns about lack of upfront planning
18% Confidence in ability to scale agile methodologies
18% Concerns about the ability to scale agile
17% No barriers
15% Perceived time and cost to make the transition
14% Development team support
13% Regulatory compliance

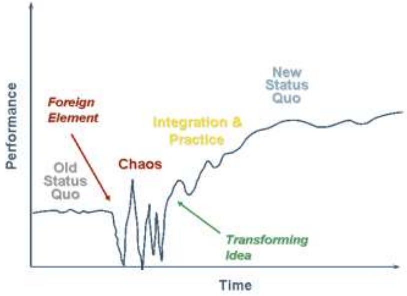
Source: 10th Annual State of Agile Report, Version One.

Adoption Strategies for Other Practices Also Work for Agile

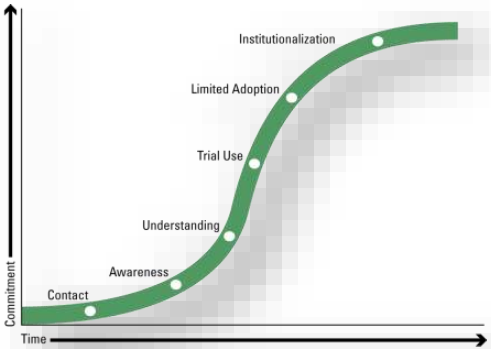
Well-Known Organizational Change Models

Satir Change Model

Understand the Change Cycle and Your Adoption Population



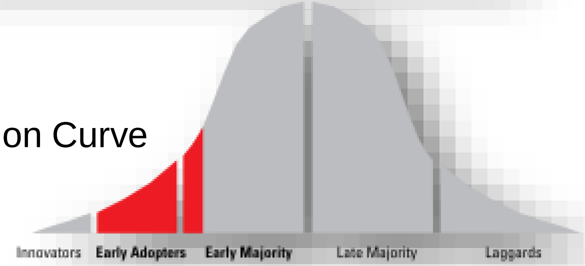
Prepare for Both Communication and Implementation Support Mechanisms that are Needed



Patterson-Connor Commitment Curve

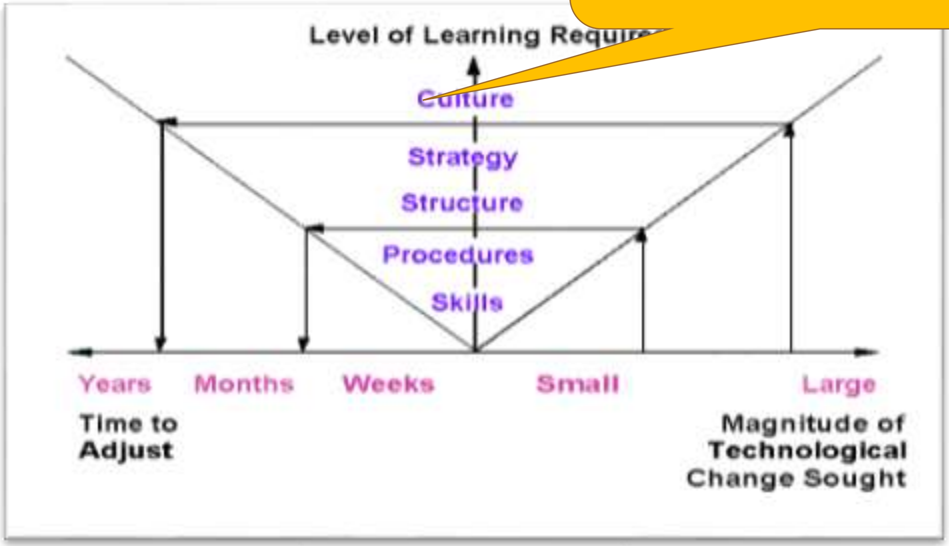
*Adapted from Daryl R. Conner and Robert W. Patterson, “Building Commitment to Organizational Change,” *Training and Development Journal* (April 1983): 18-30.

Technology Adoption Curve



Adler model for Practice/Technology Change Magnitude

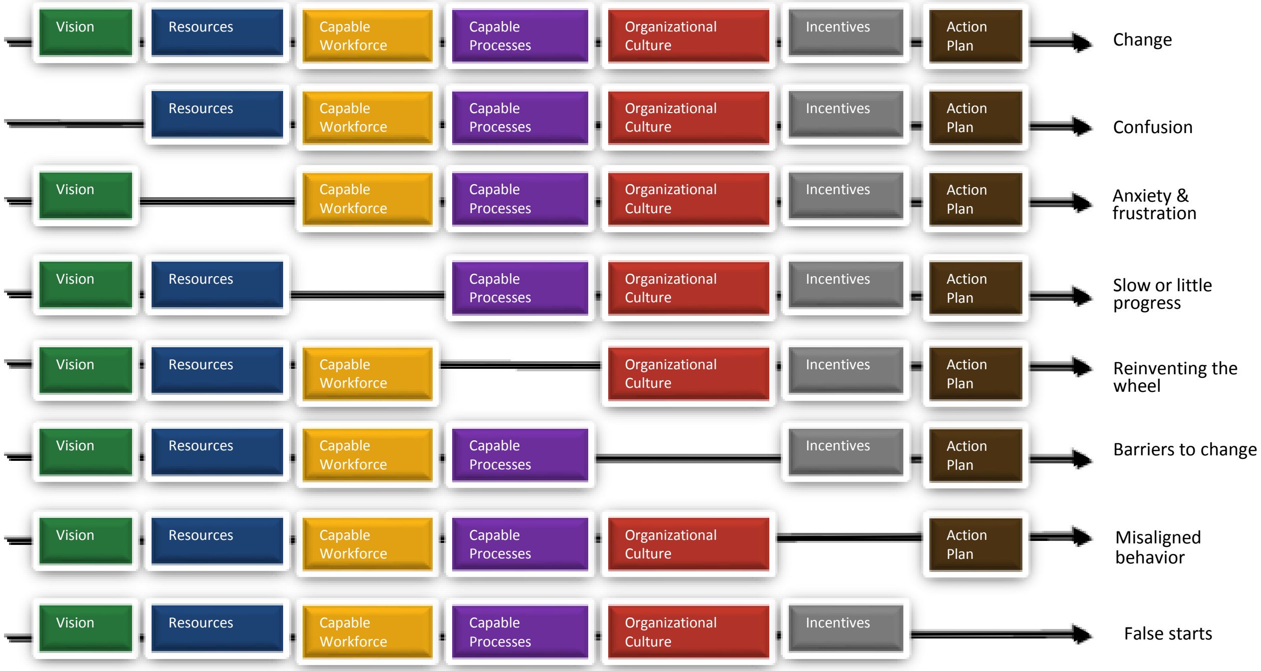
Agile and SAFe **adoption (especially at scale)** reflect **a cultural shift. That means all the levels below culture will have changes to make.**



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.

All the Dimensions of Change Need to Align



Vision	Resources	Capable Workforce	Capable Processes	Organizational Culture	Incentives	Action Plan	Change
	Resources	Capable Workforce	Capable Processes	Organizational Culture	Incentives	Action Plan	Confusion
Vision		Capable Workforce	Capable Processes	Organizational Culture	Incentives	Action Plan	Anxiety & frustration
Vision	Resources		Capable Processes	Organizational Culture	Incentives	Action Plan	Slow or little progress
Vision	Resources	Capable Workforce		Organizational Culture	Incentives	Action Plan	Reinventing the wheel
Vision	Resources	Capable Workforce	Capable Processes		Incentives	Action Plan	Barriers to change
Vision	Resources	Capable Workforce	Capable Processes	Organizational Culture		Action Plan	Misaligned behavior
Vision	Resources	Capable Workforce	Capable Processes	Organizational Culture	Incentives		False starts

Adapted by Buttles (2010) from: Delorise Ambrose, 1987

Go in With Your Eyes Open

“The Backwards Brain Bicycle”

https://youtu.be/MFzDaBzBlL0

7:57

Observations from Technology Transitions SEI Has Witnessed

Multi-dimensional

- Multiple dimensions have to be addressed simultaneously to achieve success, not just the technology content

Challenging to Duplicate Success

- Most organizations are very poor at transferring what they’ve learned from one technology transition effort to another

Multi-Audience

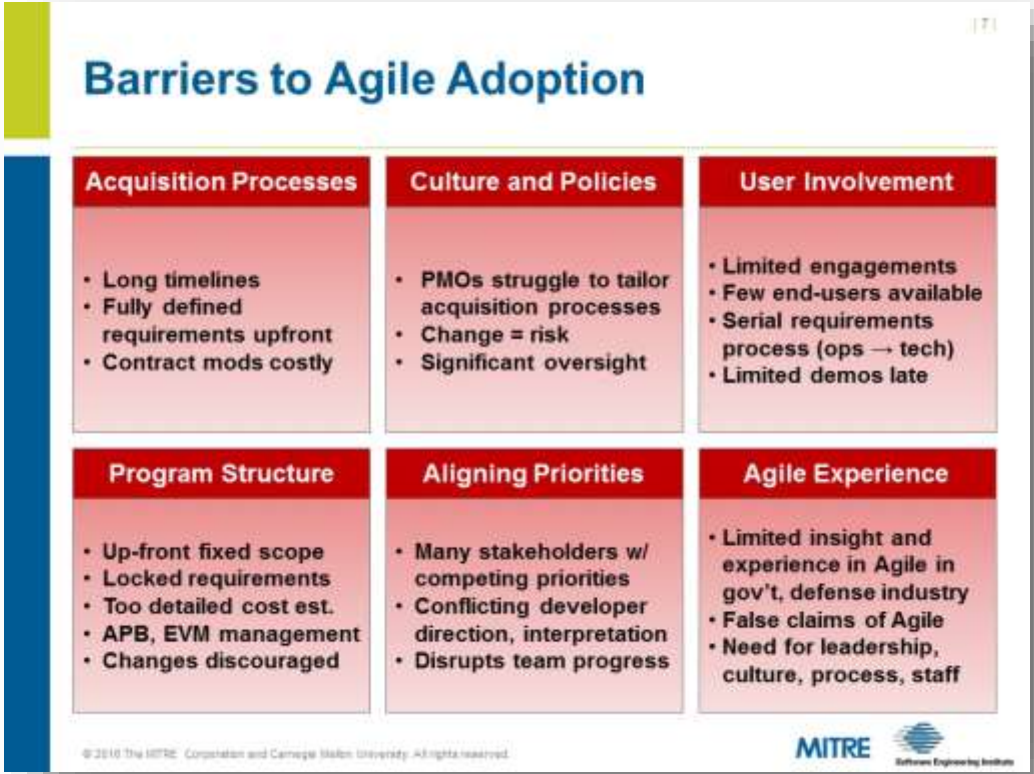
- Different audiences respond differently as they are introduced to the technology

Non-linear

- Acceptance of a new technology does not happen in a linear, predictable fashion

SEI/MITRE Perspective on Government Agile Adoption Barriers

Which of these do your programs face?



| 7 |

Barriers to Agile Adoption

Acquisition Processes

- Long timelines
- Fully defined requirements upfront
- Contract mods costly

Culture and Policies

- PMOs struggle to tailor acquisition processes
- Change = risk
- Significant oversight

User Involvement

- Limited engagements
- Few end-users available
- Serial requirements process (ops → tech)
- Limited demos late

Program Structure

- Up-front fixed scope
- Locked requirements
- Too detailed cost est.
- APB, EVM management
- Changes discouraged

Aligning Priorities

- Many stakeholders w/ competing priorities
- Conflicting developer direction, interpretation
- Disrupts team progress

Agile Experience

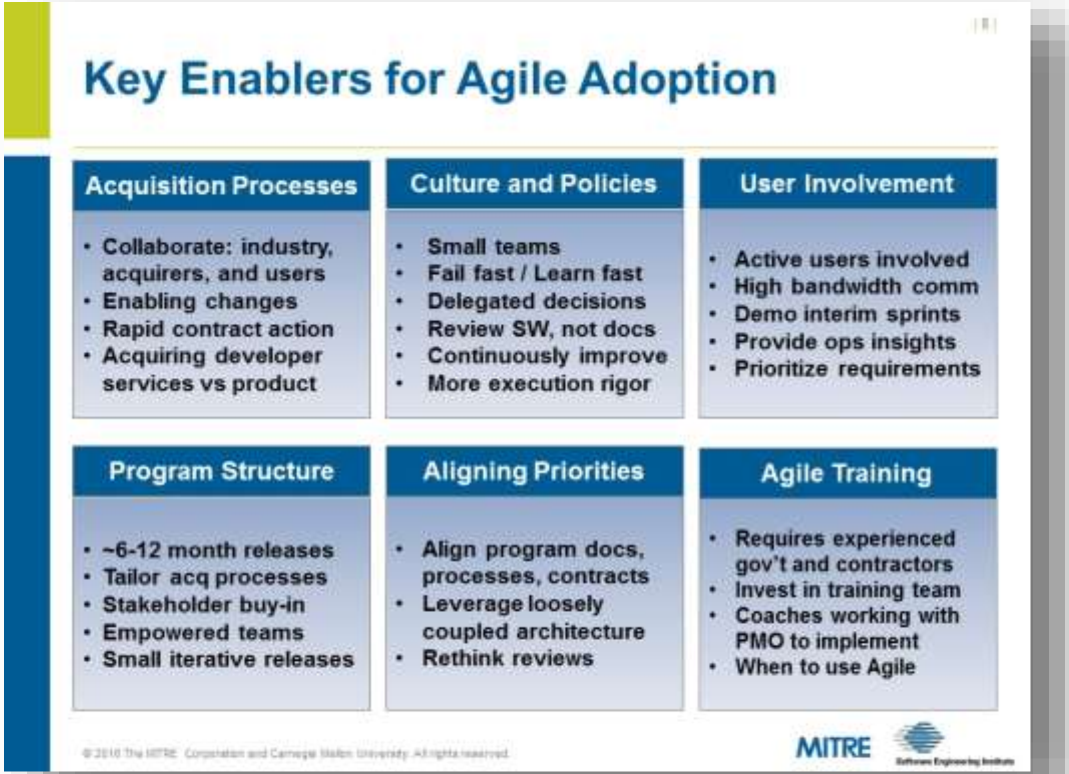
- Limited insight and experience in Agile in gov't, defense industry
- False claims of Agile
- Need for leadership, culture, process, staff

MITRE Software Engineering Institute

From SEI and Mitre briefing to General E. Pawlikowski April 8, 2016

SEI/MITRE Perspective on Government Enablers to Agile Adoption

Which of these do your programs exhibit?



Key Enablers for Agile Adoption

Acquisition Processes

- Collaborate: industry, acquirers, and users
- Enabling changes
- Rapid contract action
- Acquiring developer services vs product

Culture and Policies

- Small teams
- Fail fast / Learn fast
- Delegated decisions
- Review SW, not docs
- Continuously improve
- More execution rigor

User Involvement

- Active users involved
- High bandwidth comm
- Demo interim sprints
- Provide ops insights
- Prioritize requirements

Program Structure

- ~6-12 month releases
- Tailor acq processes
- Stakeholder buy-in
- Empowered teams
- Small iterative releases

Aligning Priorities

- Align program docs, processes, contracts
- Leverage loosely coupled architecture
- Rethink reviews

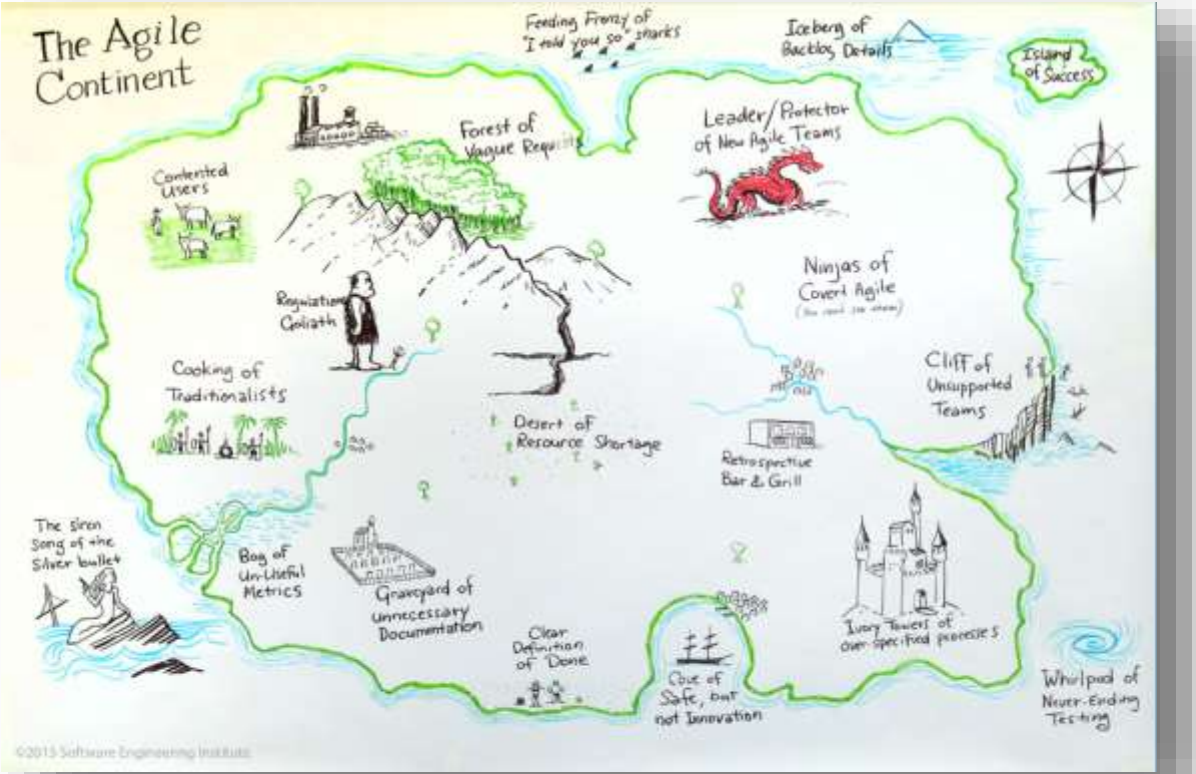
Agile Training

- Requires experienced gov't and contractors
- Invest in training team
- Coaches working with PMO to implement
- When to use Agile

MITRE Software Engineering Institute

From SEI and Mitre briefing to
General E. Pawlikowski
April 8, 2016

Go in with Your Eyes Open …
Understand your Organization's Agile Adoption Landscape


The Agile Continent
Feeding Frenzy of "I told you so" sharks
Iceberg of Backlog Details
Island of Success
Forest of Vague Requi…
Leader/Protector of New Agile Teams
Contented Users
Regulation Goliath
Ninjas of Covert Agile
Cooking of Traditionalists
Cliff of Unsupported Teams
Desert of Resource Shortage
Retrospective Bar & Grill
The Siren Song of the Silver bullet
Bog of Un-Useful Metrics
Graveyard of Unnecessary Documentation
Clear Definition of Done
Cove of Safe, but not Innovation
Ivory Towers of over-specified processes
Whirlpool of Never-Ending Testing
©2015 Software Engineering Institute

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

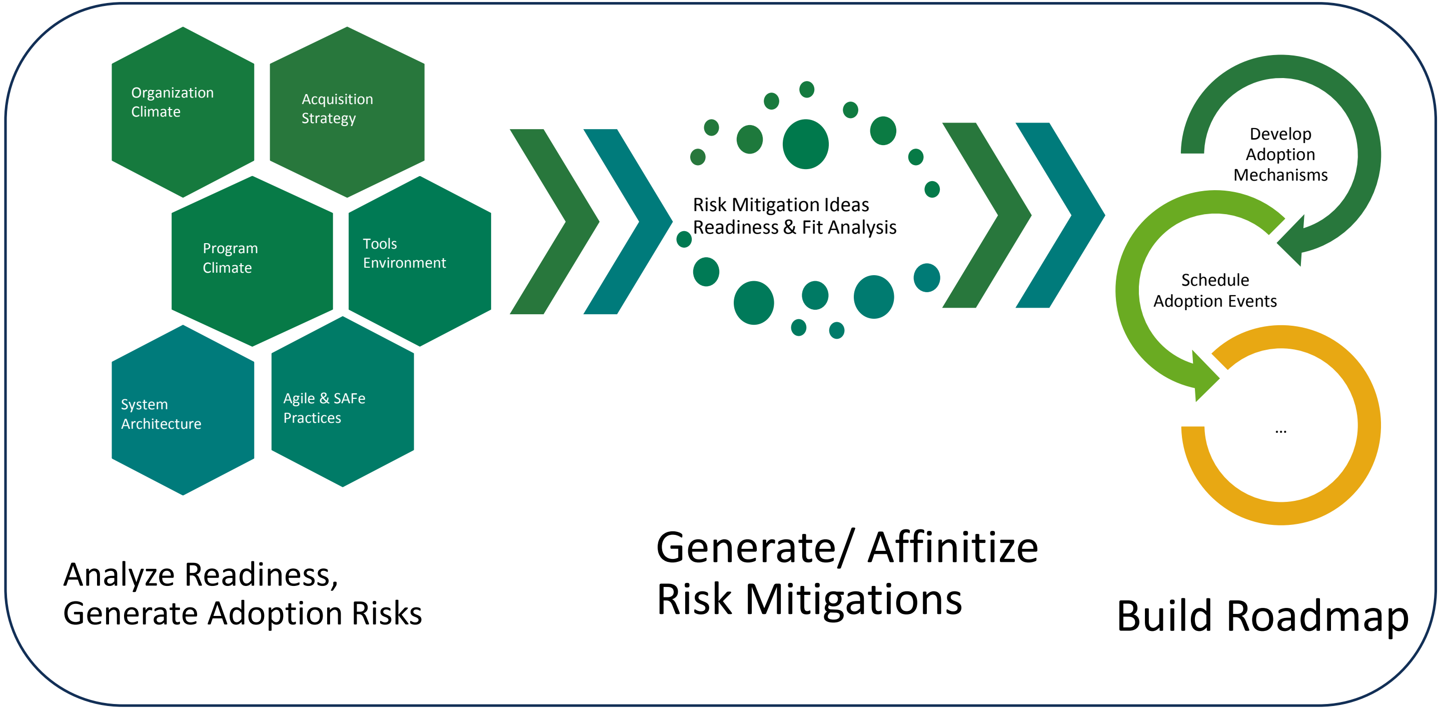
It helps by supporting identification of adoption risks and assisting in developing mitigation strategies for them

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.

Overview of SEI Readiness & Fit Analysis Workshop



Organization Climate

Acquisition Strategy

Program Climate

Tools Environment

System Architecture

Agile & SAFe Practices

Risk Mitigation Ideas
Readiness & Fit Analysis

Develop Adoption Mechanisms

Schedule Adoption Events

...

Analyze Readiness,
Generate Adoption Risks

Generate/ Affinitize
Risk Mitigations

Build Roadmap

Example Agile RFA Outcomes

Program that applied RFA prior to implementation:

- Was able to continue in a very volatile management environment (3 directors in 4 months—no connection to Agile implementation!)
- Agile RFA was a primary communication mechanism to new leadership about risks and opportunities with their culture
- One director “didn’t like the answers” (as to their culture fit at the time) but appreciated “going in with eyes open”

Program that applied RFA during implementation:

- Was able to identify the root cause of some of their adoption issues
- Changed how they communicated some of their progress to management
  - Improved management’s ability to understand both the opportunities and risks that Agile presented to the program

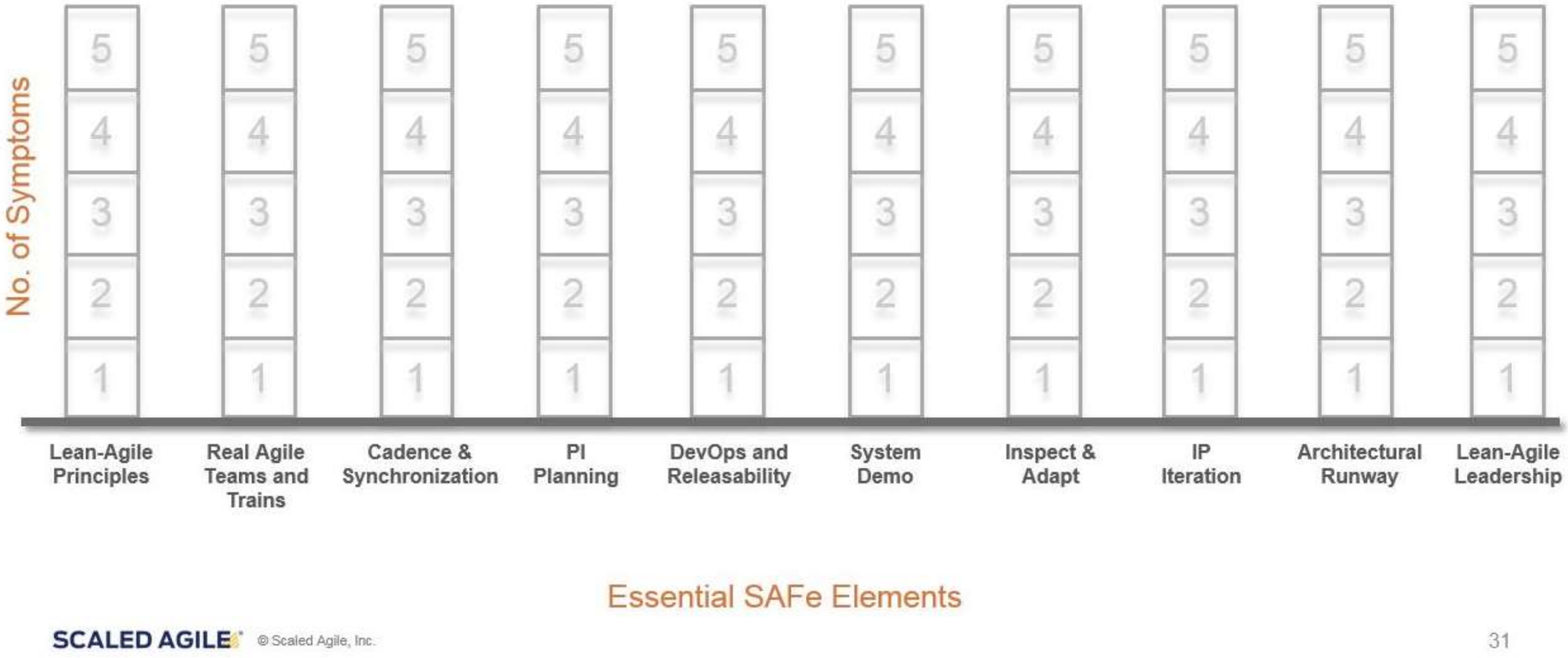
RFA Workshop Activities

- For each RFA dimension, identify (with the help of SEI-provided “thought 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
- “Thought Joggers” are statements expressed as though the Program is successfully executing in an Agile eco-system

Thought Joggers 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.

Essential SAFe Practices Self-Assessment



No. of Symptoms

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
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

Essential SAFe Elements

SCALED AGILE

Slide from Scaled Agile Inc: Essential SAFe Toolkit v4.5.0

Summary

Summary

Adopting Agile methods involves (sometimes significant) cultural shifts as well as practice changes

- Transition and adoption approaches for other major organizational changes will work for Agile adoption as well
- Many adoption support mechanisms exist out in the commercial world that can be adapted to regulated settings
  - The SEI technical notes and other resources (blogs, podcasts, etc.) on Agile adoption are meant to support acquisition practitioners in becoming knowledgeable about different issues they may encounter when adopting Agile or Lean methods

Going in "with eyes open", whether via RFA or other means, is a smart way to get started with Agile adoption

Contact Information

Nanette Brown

Senior Member of the Technical Staff

Telephone: +1 203.231.3999

Email: nb@sei.cmu.edu

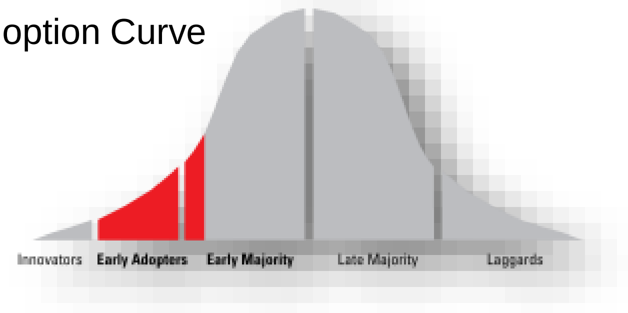
Backup Slides

Additional Information on Commonly Used Technology Adoption Models

Understand Your Adoption Population

People can
be threatened
by a new method...
RETRAINING
RESISTANCE
OR
FEAR of FIRING
EARLY
ADOPTERS
THE
CHASM!
EARLY
MAJORITY

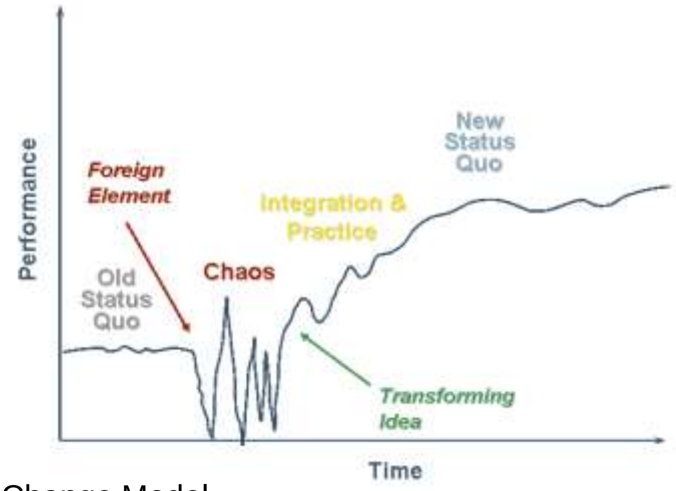
Technology Adoption Curve



Not all changes affect an individual or group the same way

- Some people who are early adopters for one type of technology are laggards for another
- Innovators and Early Adopters are good candidate for “technical feasibility” pilots – answering “Will the technology work at all?”
- Early Majority and Late Majority are better for “adoption feasibility pilots”– answering “What does it take to make the technology work HERE?”

Understand the Cycle of Change for Individuals and Groups

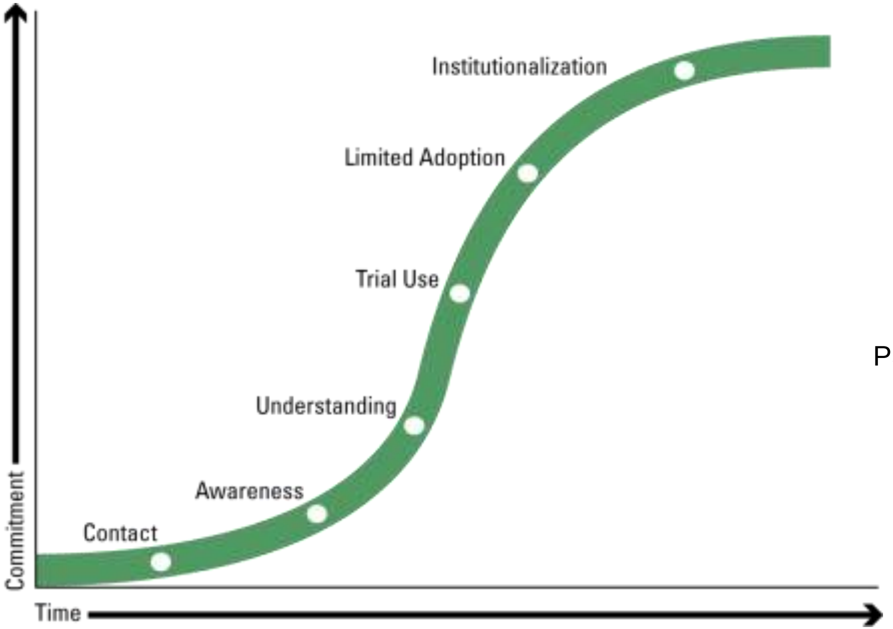


Satir Change Model

New Agile practices are your “Foreign Element”

Even after you have figured out your Transforming Idea, you still need to leave room for Integration & Practice before the desired new performance level is achieved

Prepare for Both Communication & Implementation Support Transition Mechanisms



Patterson-Connor Commitment Curve

Communication mechanisms support Contact, Awareness, & Understanding

Implementation mechanisms support Trial Use, Adoption, and eventually, Institutionalization

*Adapted from Daryl R. Conner and Robert W. Patterson, “Building Commitment to Organizational Change,” *Training and Development Journal* (April 1983): 18-30.