SAFe® DISTILLED SUPPLEMENT

WHAT'S NEW IN SAFe® 4.6

Richard Knaster
Dean Leffingwell

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Visit us on the Web: informit.com/aw

Main ISBN-13: 978-0-13-517049-6
ISBN-10: 0-13-517049-4

1 18

Contents

Preface

Today, major analysts and indicators point to SAFe as the world's most widely used framework for enterprise agility. We are gratified by SAFe's popularity, but we don't take it lightly. In fact, we interpret this as our personal responsibility to make sure that SAFe is always one step ahead when it comes to helping organizations navigate the disruptive forces that define today's—and tomorrow's—marketplace.

As soon as SAFe 4.5 was released in early 2017, we got busy thinking about what we could do to better help organizations transform into truly Lean enterprises. We looked closely at our most successful customers. What do they have in common? From the outside looking in, we saw the obvious: They are amazing innovators, they are always adapting, they are obsessed with being able to anticipate customer needs, and they respond to those needs in what seems like nanoseconds.

We zoomed in and took a much closer look, searching for patterns. What came into focus are five primary attributes—'core competencies'—that are clear differentiators between those who are getting the full benefits of Lean-Agile development, and those who aren't. The competencies—it turns out—are key. They set the stage for a fluid convergence of Agile, Lean, DevOps, and Lean Portfolio Management, which then enables the organization to operate in a truly Lean fashion. Companies that score high on the competencies are proven to respond more effectively to volatile market conditions, changing customer needs, and emerging technologies.

Identifying the competencies was one of those 'ah-ha' moments you get when something comes into focus that you haven't quite seen that way all along. Once they came into view, we knew that better explaining and enhancing these competencies was the next, most important step in evolving the Framework. This new understanding sets the stage for SAFe 4.6, which—through the Five Core Competencies of the Lean Enterprise—helps organizations build the right muscle memory to operate in a Lean fashion.

We have no doubt that this new way of working with SAFe—through the lens of the core competencies—will make a big difference in success rates, especially for organizations struggling with their transformations. Plus, SAFe 4.6 integrates all our new learnings in Agile technical practices, DevOps and continuous delivery, Lean portfolio management, and SAFe in government.

We're excited to be able to share our 'ah-ha' moment with you through SAFe 4.6. Our hope is that this latest version of the Framework will keep your delivery pipeline humming nicely, accelerate your journey to enterprise agility, and keep your competitors wondering what you'll do next.

About This Supplement

This supplement provides an overview of what's new in SAFe 4.6 and reflects our commitment to preserve the investment you made in purchasing and reading *SAFe® 4.5 Distilled*.

The good news is, everything you've learned from *SAFe® 4.5 Distilled* stills holds true. SAFe 4.6 is fully backward compatible with SAFe 4.5 and provides incremental additions to the guidance that enables organizations to transform into Lean Enterprises.

The content is organized in a way that allows you to quickly leverage the new concepts:

- The first chapter introduces the concept of the Lean Enterprise, the Five Core Competencies, and the ways in which they help organizations achieve the goal of the 'shortest sustainable lead time.'
- Chapters 2–6 cover the core competencies in depth: Lean-Agile Leadership, Team and Technical Agility, DevOps and Release on Demand, Business Solutions and Lean Systems Engineering, and Lean Portfolio Management.
- Chapter 7 summarizes SAFe for Government, which is new guidance for applying SAFe in the public sector.

The last section of this supplement contains the SAFe 4.6 glossary.

Since we will continue updating SAFe, please register your copy of *SAFe® 4.5 Distilled* at informit.com. This will give you convenient access to downloads, updates, and corrections as they become available.

To start the registration process, go to informit.com/register and log in or create an account. Enter the product ISBN (9780135170496) and click Submit. Once the process is complete, you will find any available bonus content under 'Registered Products.'

—Richard Knaster and Dean Leffingwell

Acknowledgments

The authors are deeply indebted to all those who have contributed to the development of the Framework. More than 125 sources—books and authors—contributed to the bodies of knowledge that underlie SAFe. In addition, another 100 or so contributors, reviewers, commenters, editors, graphic designers, and others make SAFe what it is today. But if we were to take time to thank all those who contributed, we wouldn't be able to call this supplement "Distilled." Fortunately, the SAFe contributors page (scaledagileframework.com/contributors) acknowledges those contributions.

However, it is appropriate to thank those who contributed most directly to this particular work:

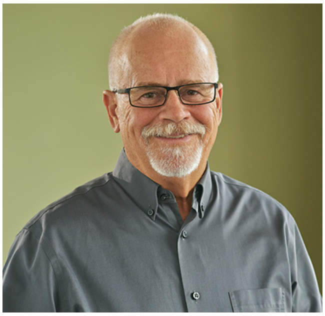
> Inbar Oren, SAFe Fellow and Methodologist
>
> Harry Koehnemann, SAFe Fellow
>
> Dr. Steve Mayner, SAFe Fellow
>
> SAFe contributors Luke Hohmann, Zach Nies, Ken Pugh, Marc Rix, and Eric Willeke

We would also like to thank our Addison-Wesley acquisition editor Greg Doench, production editor Julie Nahil, and copy editor, Jill Hobbs. We are also indebted to Scaled Agile, Inc., production designer Regina Cleveland and graphic designer Jerod Barker.

About the Authors

Richard Knaster, SAFe Fellow, Principal Consultant, Scaled Agile, Inc.

Richard has more than 30 years' experience in software and systems development, in roles ranging from developer to executive, and has been leading large-scale Agile transformations for well over 15 years. Richard actively works on advancing SAFe's Lean-Agile methods as a SAFe Fellow and methodologist. As a principal consultant, he is passionate about helping organizations create a better environment to deliver value, improve quality, flow and being more engaging and fun. Richard is also co-author of the *SAFe® 4.5 Reference Guide*.



Dean Leffingwell, creator of SAFe, Chief Methodologist, Scaled Agile, Inc.

Widely recognized as the one of the world's foremost authorities on Lean-Agile best practices, Dean Leffingwell is an author, serial entrepreneur, and software and systems development methodologist. His earlier books, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise* and *Scaling Software Agility: Best Practices for Large Enterprises*, form much of the basis of modern thinking on Lean-Agile practices and principles. He currently serves as CEO and Chief Methodologist to Scaled Agile, Inc., which he co-founded in 2011.

1

What's New in SAFe® 4.6 for Lean Enterprises?

Every business is a software business now. Agility isn't an option, or a thing just for teams; it is a business imperative.

—Dean Leffingwell, Creator of SAFe

Introducing SAFe 4.6 for Lean Enterprises

The *Scaled Agile Framework (SAFe) 4.6 for Lean Enterprises* is a knowledge base of proven, integrated principles, practices, and competencies for Lean, Agile, and DevOps.

More than the sum of its parts, SAFe is a scalable and configurable framework that helps organizations deliver new products, services, and solutions with the shortest sustainable lead time. It's a system that guides the roles, responsibilities, and activities necessary to achieve a sustained, competitive technological advantage.

Now in its fifth revision, SAFe is improving business outcomes for companies of all sizes across the world. It has helped enterprises and governments build complex digital solutions with dramatic improvements in time-to-market, employee engagement, quality, customer satisfaction, and improved economic outcomes. It also helps create cultures that are more productive, rewarding, and fun.

We define the *Lean Enterprise*[1] as a thriving digital-age business that delivers competitive systems and solutions to its customers in the shortest sustainable lead time. To address this larger vision, version 4.6 of SAFe introduces the *Five Core Competencies of the Lean Enterprise*, which are now the primary lens through which to understand and implement SAFe.

[1] https://scaledagileframework.com/safe-for-lean-enterprises

The Five Core Competencies of the Lean Enterprise

Each core competency is a set of related knowledge, skills, and behaviors, which together enable enterprises to deliver high-quality and high-value digital solutions. Mastery of the five core competencies enables enterprises to successfully navigate digital disruptions of their businesses and to effectively respond to volatile market conditions, changing customer needs, and emerging technologies.

Figure 1-1 highlights where these competencies appear on the new 4.6 Big Picture. Each of these competencies is briefly described next. Additionally, each has its own chapter to further elaborate the 'why' and the details of the competency.

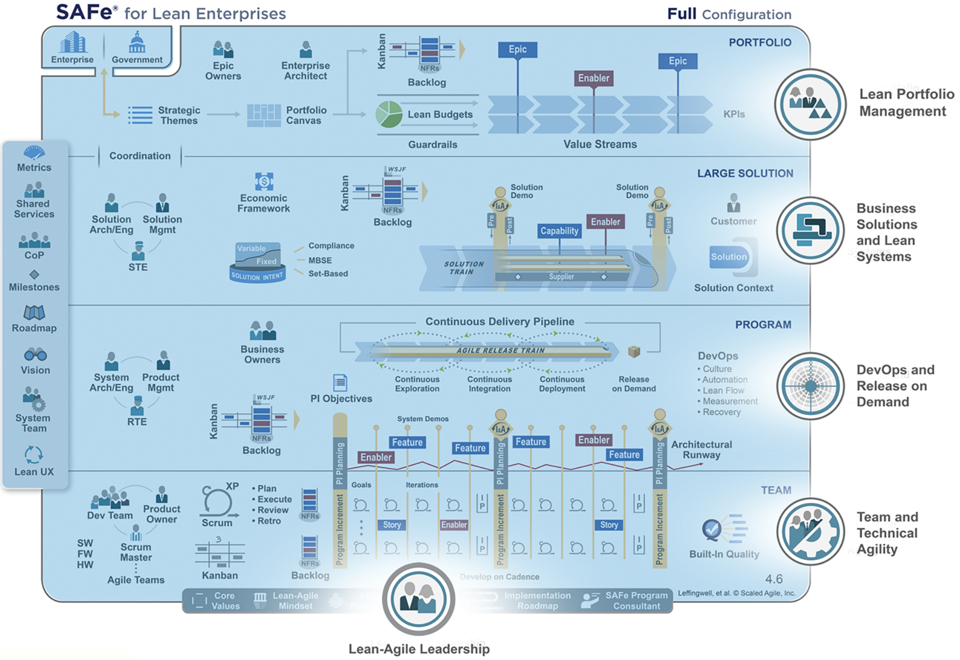


Figure 1-1. New SAFe 4.6 Big Picture, highlighting the five competencies of the Lean Enterprise

Lean-Agile Leadership

Lean-Agile Leadership is a foundational competency, and it appears as such on the Big Picture (Figure 1-2). This competency describes how Lean-Agile leaders drive and sustain organizational change and operational excellence by empowering individuals and teams to reach their highest potential. They do this by learning, exhibiting, teaching, and coaching SAFe's Lean-Agile mindset, values, principles, and practices.

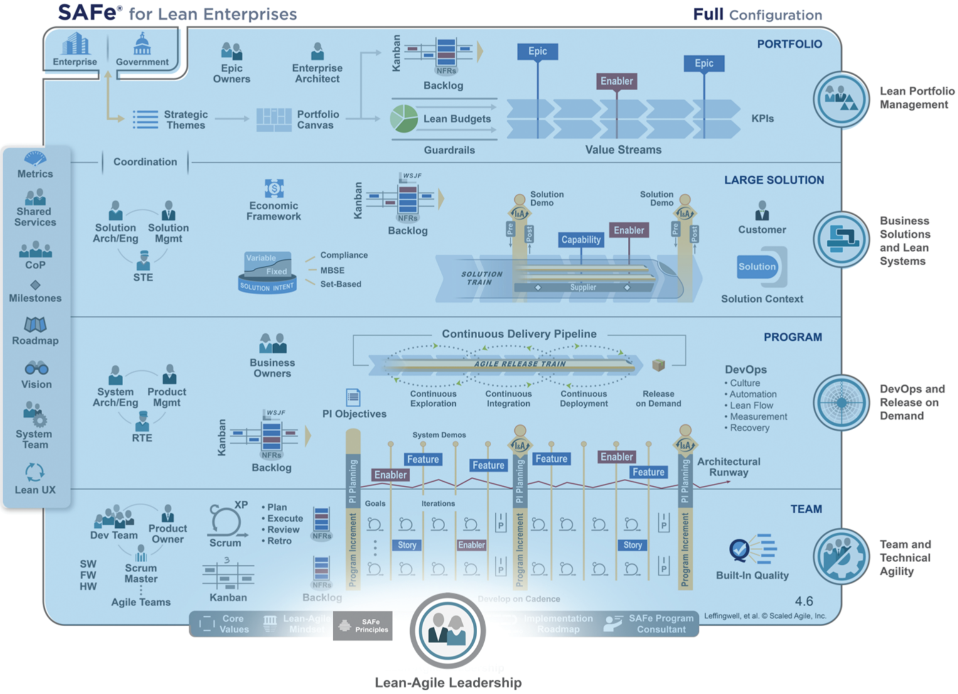


Figure 1-2. Lean-Agile Leadership competency, highlighting related framework changes

LEAN-AGILE LEADERSHIP RELATED FRAMEWORK CHANGES

- The prior 'Lean-Agile leaders' article has been combined with the new 'Lean-Agile Leadership competency' article and the content has been repurposed to this more advanced leadership paradigm.
- The SAFe principles[2] have been updated. Principles #3 and #7 have been largely rewritten.
- A new advanced topic article, on the 'Evolving Role of Managers in Lean-Agile Development,'[3] describes the changes and ongoing responsibilities of line management in the new way of working.

[2] https://www.scaledagileframework.com/safe-lean-agile-principles/

[3] https://www.scaledagileframework.com/the-evolving-role-of-managers/

Team and Technical Agility

The *Team and Technical Agility* competency (Figure 1-3) describes the critical skills and Lean-Agile principles and practices that are needed to create high-performing Agile teams who create high-quality, well-designed technical solutions. As a result it requires mastery of two essential skills:

- *Team agility* fosters and enables high-performing Agile teams that are cross-functional, organized around the flow of value, and operate using basic and effective Agile principles and practices.
- *Technical agility* describes the Lean-Agile technical practices that are necessary to create high-quality, well-designed technical solutions, which are resilient and fit for purpose.

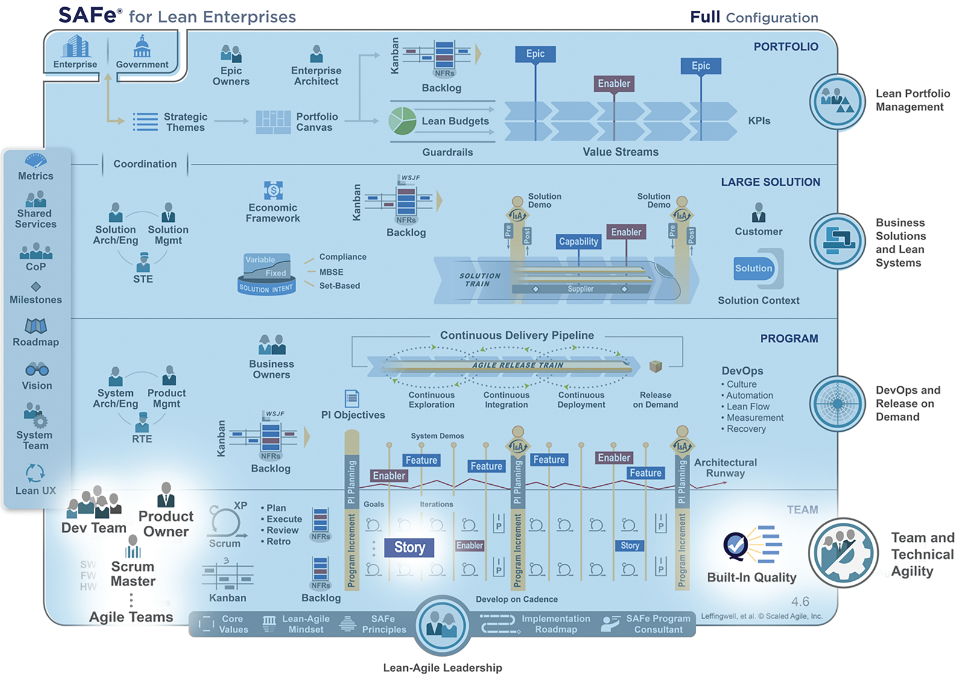


Figure 1-3. Team and Technical Agility competency, highlighting related framework changes

TEAM AND TECHNICAL AGILITY RELATED FRAMEWORK CHANGES

- The entirely rewritten Built-in Quality[4] article describes the five dimensions that enable building in quality—*flow*, *architecture and design quality*, *code quality*, *system quality*, and *release quality*.
- The Product Owner,[5] Scrum Master,[6] Dev Team,[7] Agile Teams,[8] and Story[9] articles were changed to reflect the new guidance and thinking from the Team and Technical Agility competency and their responsibilities in Behavior-Driven Development (BDD).
- New *advanced topic* articles cover the following topics:
  - Test-Driven Development (TDD)[10] is a test-first philosophy and practice that recommends building and executing tests before implementing the code or a component of a system.
  - Behavior-Driven Development (BDD)[11] is a test-first, Agile testing practice that provides built-in quality by defining—and wherever possible automating—tests before implementation of the code. This becomes an integral part of specifying and understanding both intended and actual system behavior.
  - Agile Testing[12] provides a comprehensive overview of Agile testing strategies using updated Agile testing quadrants.

DevOps and Release on Demand

The *DevOps and Release on Demand* competency (Figure 1-4) describes how implementing DevOps and a continuous delivery pipeline provides the enterprise with the capability to release value, in whole or in part, at any time necessary to meet market and customer demand.

[4] https://www.scaledagileframework.com/built-in-quality/
[5] https://www.scaledagileframework.com/product-owner/
[6] https://www.scaledagileframework.com/scrum-master/
[7] https://www.scaledagileframework.com/dev-team/
[8] https://www.scaledagileframework.com/agile-teams/
[9] https://www.scaledagileframework.com/story
[10] https://www.scaledagileframework.com/test-driven-development/
[11] https://www.scaledagileframework.com/behavior-driven-development/
[12] https://v46.scaledagileframework.com/agile-testing/

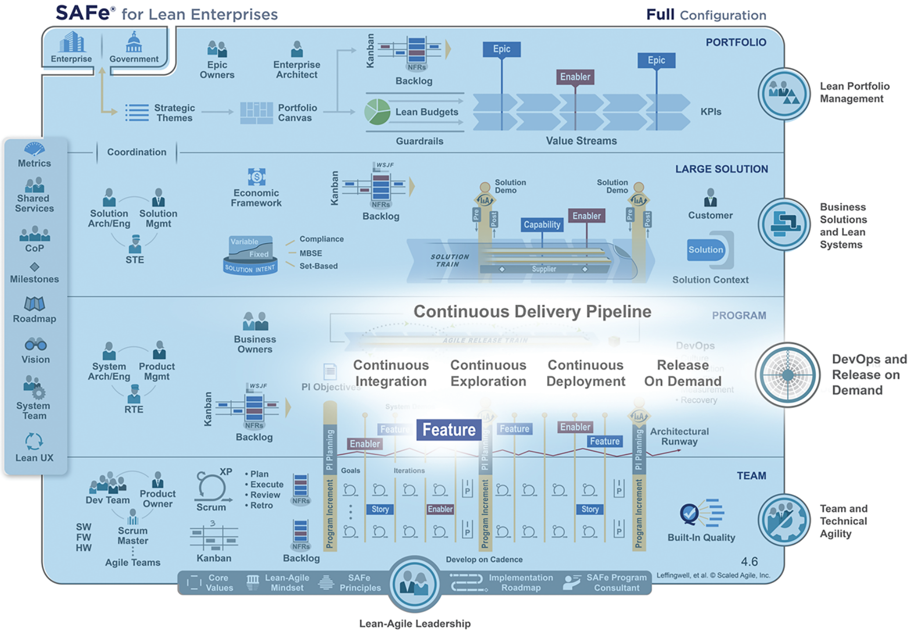


Figure 1-4. DevOps and Release on Demand competency, highlighting related framework changes

DEVOPS AND RELEASE ON DEMAND RELATED FRAMEWORK CHANGES

- Deeper and more advanced Continuous Delivery Pipeline[13] guidance includes mapping the current delivery pipeline and assessing and improving flow with the DevOps and Release on Demand[14] health radar.
- All-new Continuous Exploration,[15] Continuous Integration,[16] Continuous Deployment,[17] and Release on Demand[18] articles reflect the 16 dimensions of the DevOps health radar.
- Updates to the DevOps[19] article reflect changes resulting from the DevOps and Release on Demand competency.
- Minor updates to the Feature[20] article describe the central role of features in the continuous delivery pipeline and in BDD.

[13] https://www.scaledagileframework.com/continuous-delivery-pipeline
[14] https://www.scaledagileframework.com/metrics/#P12
[15] https://www.scaledagileframework.com/continuous-exploration/
[16] https://www.scaledagileframework.com/continuous-integration/
[17] https://www.scaledagileframework.com/continuous-deployment
[18] https://www.scaledagileframework.com/release-on-demand/
[19] https://www.scaledagileframework.com/devops/
[20] https://www.scaledagileframework.com/features-and-capabilities

Business Solutions and Lean Systems Engineering

This *Business Solutions and Lean Systems Engineering* competency (Figure 1-5) describes how to apply Lean-Agile principles and practices to the specification, development, deployment, and evolution of large, complex software applications and cyber-physical systems.

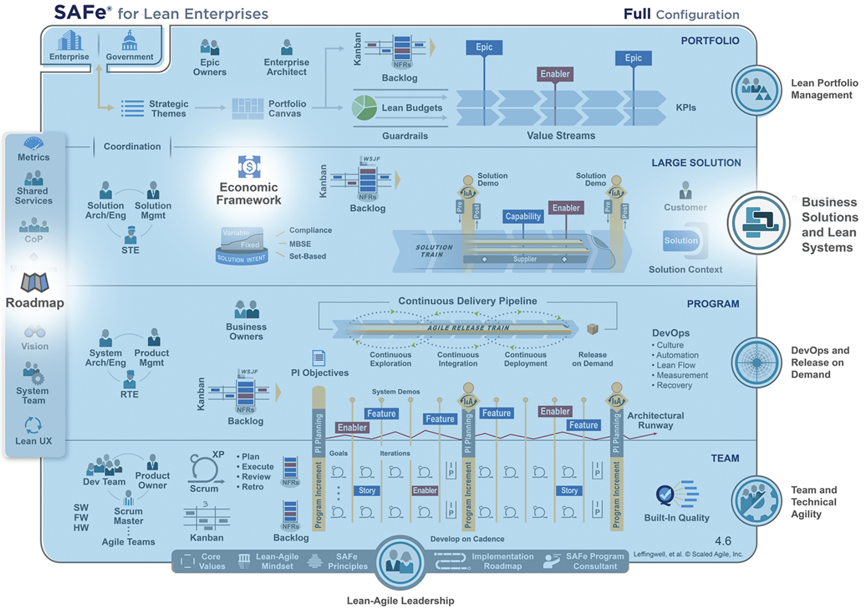


Figure 1-5. Business Solutions and Lean Systems Engineering competency, highlighting related framework changes

BUSINESS SOLUTIONS AND LEAN SYSTEMS ENGINEERING RELATED FRAMEWORK CHANGES

- The new Economic Framework[21] includes *four* primary elements:
  1. Operating within Lean budgets and guardrails
  2. Understanding solution economic trade-offs
  3. Leveraging suppliers
  4. Sequencing jobs for the maximum benefit (using WSJF)
- An all-new Roadmap[22] article introduces multiple planning horizons and the solution roadmap that provides a longer-term—often multiyear view, showing the key milestones and deliverables needed to achieve the solution vision over time. The roadmap also contains new guidance on how to understand and apply *market rhythms* and *events* to release planning.

[21] https://www.scaledagileframework.com/economic-framework/

[22] https://www.scaledagileframework.com/roadmap/

Lean Portfolio Management

The *Lean Portfolio Management* competency (Figure 1-6) aligns strategy and execution by applying Lean approaches and systems thinking to strategy and investment funding, Agile portfolio operations, and Lean governance.

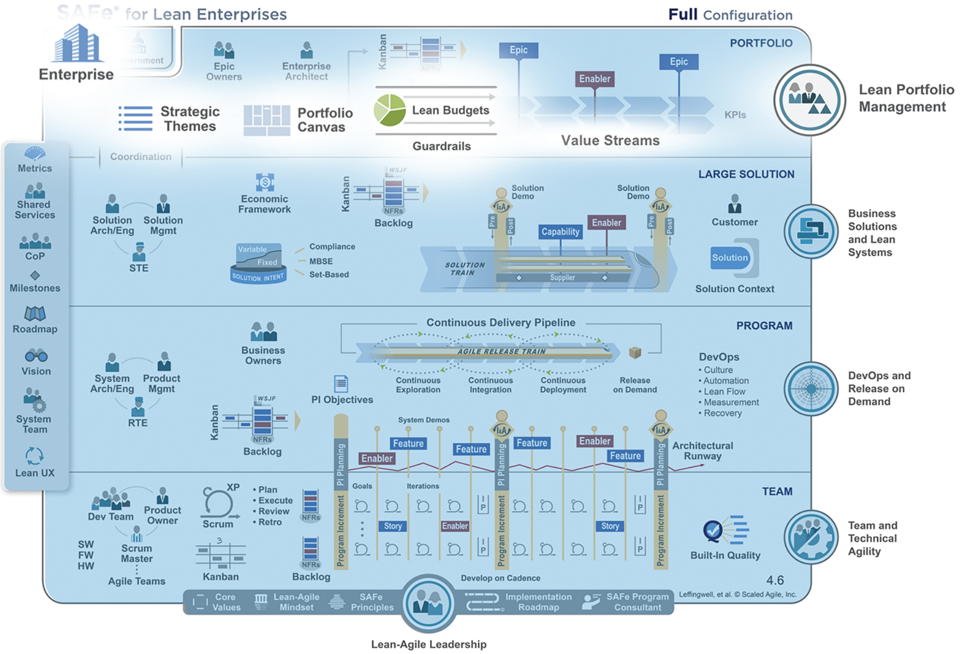


Figure 1-6. Lean Portfolio Management competency, highlighting related framework changes

LEAN PORTFOLIO MANAGEMENT RELATED FRAMEWORK CHANGES

- An updated strategy formulation is provided in the Enterprise[23] article, along with a better definition of the SAFe portfolio.
- Updated Strategic Themes[24] resulting from the new Lean Portfolio Management competency are available.
- A new Portfolio Canvas[25] article describes how a portfolio of solutions creates, delivers, and captures value for an organization and helps the organization evolve the portfolio to a new future state.

[23] https://www.scaledagileframework.com/enterprise/

[24] https://www.scaledagileframework.com/strategic-themes/

[25] https://www.scaledagileframework.com/portfolio-canvas/

- A new Lean Budget Guardrails[26] article provides guidance on how to ensure that the right investments are being made within the portfolio's budget.
- An updated Lean Budgets[27] article provides new guidance for moving from traditional budgets to Lean budgets, guiding investments by horizon, and applying participatory budgeting.
- The updated Value Streams[28] article includes a section on defining value streams with a revised development value stream canvas that aligns with the new portfolio canvas.

SAFe for Government

In addition to the core competencies, the SAFe 4.6 release includes new government guidance (Figure 1-7). This guidance describes a set of success patterns that can help public-sector organizations implement Lean-Agile practices with SAFe.

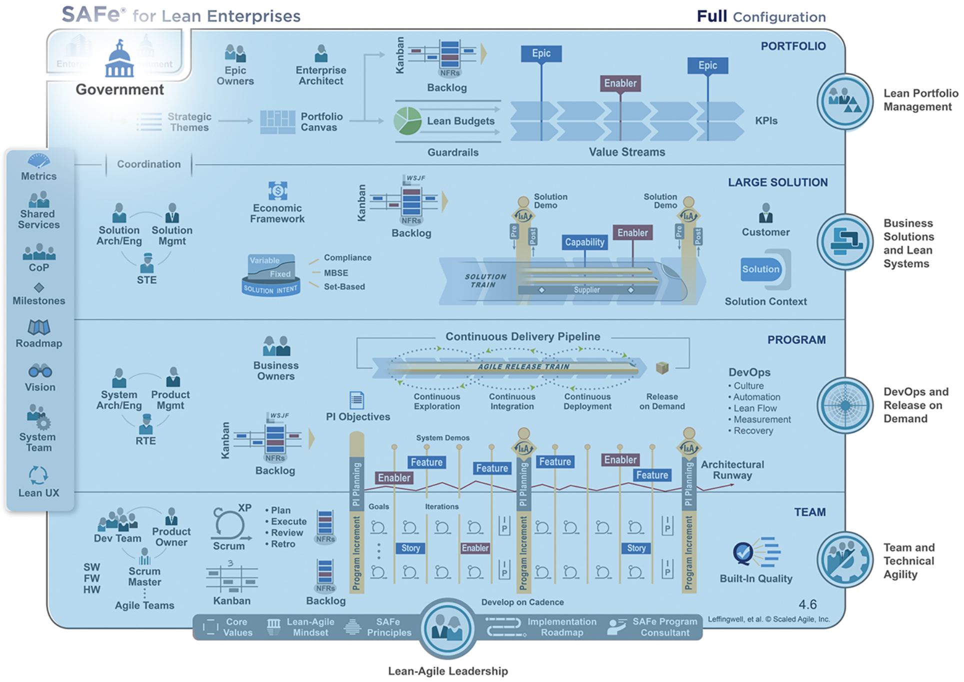


Figure 1-7. SAFe for Government

[26] https://www.scaledagileframework.com/guardrails/
[27] https://www.scaledagileframework.com/lean-budgets/
[28] https://www.scaledagileframework.com/value-streams/

The new Government[29] article on the SAFe website is a work in process that encompasses a number of new subarticles, and it will also serve as a landing page for a more comprehensive treatment of applying SAFe in a national, international, regional, or local government context.

The recommendations for Lean-Agile adoption in government do not modify SAFe's terms and practices to fit existing government practices. In fact, experienced practitioners in government services have reported that they achieve the best results when they use the SAFe model and terminology without modification.

SAFe Configurations

SAFe 4.6 continues to support the full range of development environments with the same four out-of-the-box configurations: Essential, Large Solution, Portfolio, and Full SAFe. But it does so with a simplified configuration selector with a tab metaphor (Figure 1-8).

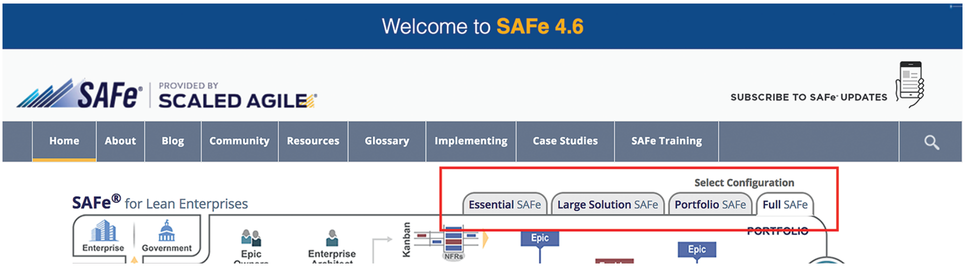


Figure 1-8. New Configurable SAFe selector

Summary

The digital transformation of industry is affecting companies worldwide. Today, virtually every business is dependent on software and solution development capabilities. SAFe helps organizations become Lean Enterprises, by enabling the entire development value stream—from idea to market release—to become more agile, leaner, and more responsive to change. Now in its fifth major revision, SAFe 4.6 is improving business outcomes for companies of all sizes around the world.

[29] https://scaledagileframework.com/government/

2

Lean-Agile Leadership Competency

"It's 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

Lean-Agile Leadership is one of the five core competencies of the Lean Enterprise. It describes how Lean-Agile Leaders drive and sustain organizational change and operational excellence by empowering individuals and teams to reach their highest potential. They do this by learning, exhibiting, teaching, and coaching SAFe's Lean-Agile mindset, values, principles, and practices.

Why Lean-Agile Leadership?

Among the SAFe competencies, Lean-Agile Leadership is foundational. Only an enterprise's managers, leaders, and executives can change and continuously improve the systems that govern how work is performed. Only these leaders can create the environment that encourages high-performing Agile teams to flourish and produce value. This results in happier, more engaged employees and increased productivity and innovation.

Figure 2-1 shows how Lean-Agile leadership anchors the foundation of SAFe with two primary responsibilities:

1. *Lean-thinking manager-teachers.* Leaders must have the knowledge and ability to think and act as 'Lean-thinking manager-teachers.'

2. *Leading the transformation*. Leaders drive the continuous journey of becoming a Lean enterprise and sustain the new way of working by fostering and inspiring the organizational changes needed.

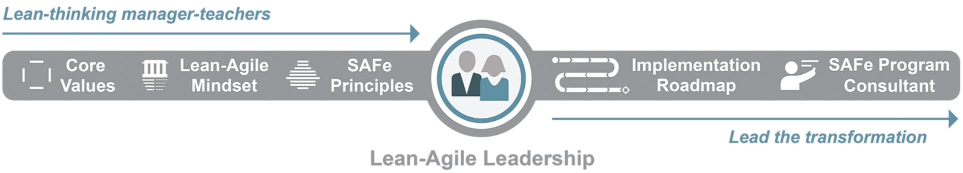


Figure 2-1. Lean-Agile leadership is the foundation of SAFe

Each of the responsibilities depicted in Figure 2-1 is described next.

Lean-Thinking Manager-Teachers

The journey to become a Lean enterprise is neither simple nor easy. Leaders must learn how to teach and coach, instead of directing and managing. Toyota calls this model of leadership Lean-thinking manager-teachers, who understand Lean at a deep level and teach it as part of their daily activities.[1] Moreover, as leaders evolve into Lean-thinking manager-teachers, they exemplify the core values, embrace a Lean-Agile mindset, apply SAFe principles, and lead the transformation. Each of these duties is described in the following subsections.

Exemplify the Core Values

At every opportunity, a leader's behavior plays a critical role in communicating, exhibiting, and emphasizing the core values. Here are some suggestions on how leaders can support and reinforce these values:

1. *Alignment*. Communicate the mission by establishing and expressing the strategy and vision. Provide relevant briefings and participate in Program Increment (PI) planning and backlog review and preparation.
2. *Built-in quality*. Demonstrate commitment to quality by refusing to accept or ship low-quality work. Support investments in capacity planning for maintenance and reduction of technical debt.

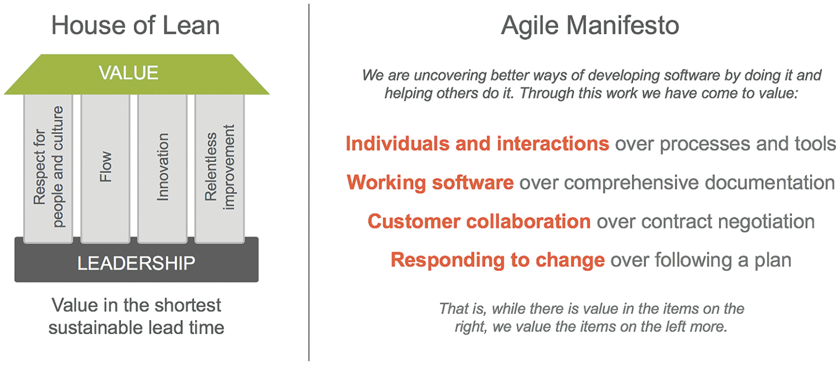
[1] Stoller, Jacob. *The Lean CEO: Leading the Way to World-Class Excellence*. McGraw-Hill Education. Kindle Edition.

3. *Transparency*. Visualize all relevant work. Take ownership and responsibility for errors and mistakes. Admit missteps and support others who acknowledge and learn from theirs. Never punish the messenger; instead celebrate learning.
4. *Program execution*. Many leaders participate as Business Owners in prioritization, PI execution, and reflection. All leaders help adjust scope to assure demand matches capacity. They aggressively remove impediments and demotivators.

At every opportunity, a leader's behavior and actions play a critical role in enabling these core values.

Embrace a Lean-Agile Mindset

The SAFe Lean-Agile mindset (Figure 2-2) is the combination of beliefs, assumptions, and actions of leaders and practitioners who embrace the Agile Manifesto and the SAFe House of Lean.



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.

Figure 2-2. SAFe House of Lean and Agile Manifesto

1. *Thinking Lean*. The SAFe House of Lean illustrates the various aspects of Lean thinking. The roof represents the goal of delivering value, while its pillars support respect for people and culture, flow, innovation, and relentless improvement to support the goal. Lean-Agile leaders are the foundation on which everything else stands.

2. *Embracing agility.* The Agile Manifesto[2] provides a value system and a set of 12 principles essential to successful Agile development. SAFe is built on the Agile values, principles, and methods as embodied by cross-functional Agile teams. Every leader must fully support and reinforce their intent and application.

Apply the SAFe Lean-Agile Principles

In addition to the Lean-Agile mindset, SAFe is based on nine immutable, underlying Lean-Agile principles. These tenets and economic concepts inspire and inform the roles and practices of SAFe, influencing leadership behaviors and decision-making (Figure 2-3):

#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

Figure 2-3. SAFe Lean-Agile principles

While all of SAFe's principles are relevant to leaders, the first two establish the economic and systems thinking required for solution and organizational development. Since management has the responsibility and authority to establish the culture of the organization, the last two principles are particularly applicable and essential to leadership and are described further here.

8. Unlock the intrinsic motivation of knowledge workers. Leaders support employees by creating an environment in which they can do their best work, and by seeking to eliminate any demotivating practices and policies. The Agile HR[3] article on the SAFe

[2] http://agilemanifesto.org/

[3] https://scaledagileframework.com/agile-hr/

website describes six significant themes for effective management of this next generation of knowledge workers.

9. Decentralize decision-making. This principle guides leaders to delegate decision-making, which reduces delays, improves product development flow and throughput, and facilitates faster feedback and more innovative solutions. Higher levels of employee empowerment are an additional, tangible benefit that reinforces principle 8.

Lead the SAFe Transformation

A Lean-thinking manager-teacher develops the thought processes and practical tools needed to start building the Lean enterprise. For this part of the journey, leaders also need to leverage the knowledge of organizational change management. Kotter describes eight steps for implementing successful change[4]:

1. Establish a sense of urgency.
2. Create the guiding coalition.
3. Develop the vision and strategy for change.
4. Communicate the change vision.
5. Empower employees for broad-based action.
6. Generate short-term wins.
7. Consolidate gains and produce more change.
8. Anchor new approaches in the culture.

These steps require the *active participation* of leaders who demonstrate commitment to the change. In fact, by "scripting the critical moves,"[5] as authors Dan and Chip Heath point out, these leaders inspire and motivate others by their own actions and create a clear path to success. The SAFe implementation roadmap, described in the next section, is designed for such a purpose.

The SAFe Implementation Roadmap

SAFe's implementation roadmap (Figure 2-4) is aligned and based in part on Kotter's eight change-management steps. For example, "establishing a sense of urgency" is

[4] Kotter, John P. *Leading Change*. Harvard Business Review Press. Kindle Edition.

[5] Heath, Dan and Chip. *How to Change Things When Change Is Hard*. Crown Publishing Group. Kindle Edition.

often created by engaging in many conversations that lead an organization to "reach the tipping point" and decide to 'Go SAFe.' The next move is to identify and train a core group of Lean-Agile change agents and leaders who will form the "powerful guiding coalition." This pattern continues throughout the roadmap and helps leaders 'know the way' as they help drive successful change.

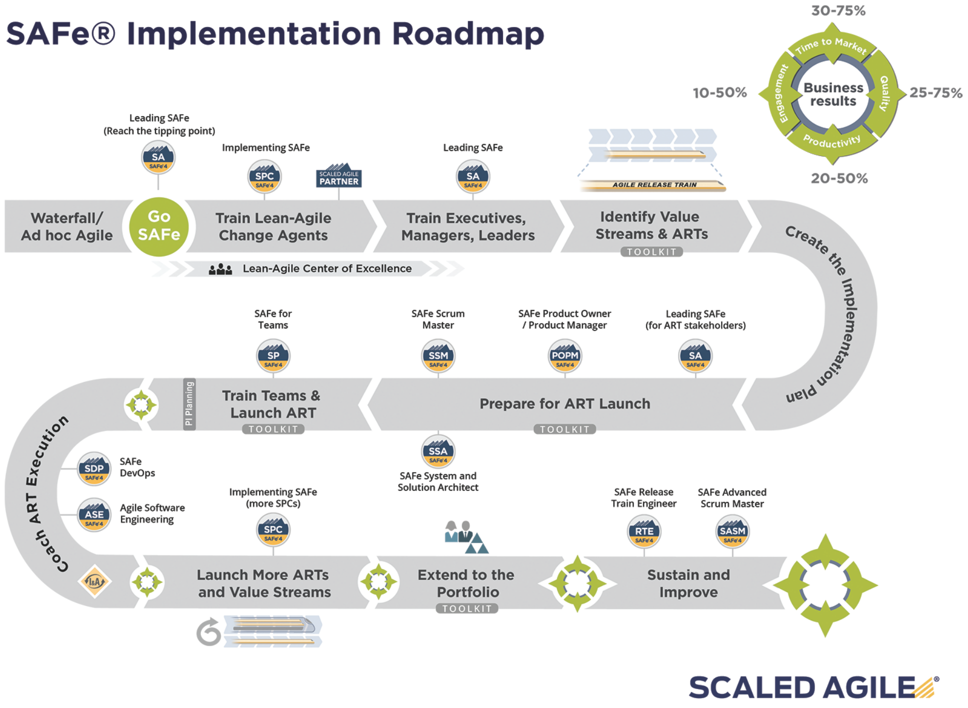


Figure 2-4. The SAFe Implementation Roadmap

SAFe IMPLEMENTATION ROADMAP CHANGES

The SAFe Implementation Roadmap has been updated with three new courses: SAFe System and Solution Architect, SAFe DevOps, and SAFe Agile Software Engineering. Moreover, a new step was added to the roadmap, 'Waterfall/Ad hoc,' to represent the state where many organizations begin their SAFe journey.

Role of the SAFe Program Consultant

SAFe Program Consultants (SPCs) are trained specialists who help lead the transformation. Many are line managers themselves. Others are natural internal change agents from other specific roles. Their intrinsic motivation to create a better future—combined with the training, tools, and courseware they receive—is critical for the successful implementation and sustainability of SAFe.

Role of the Traditional Manager in the SAFe Lean Enterprise

SAFe emphasizes the value of self-organizing, cross-functional teams; it's the 'DNA' of Agile development. As a result, employees no longer need daily management instruction and direction. While Lean-Agile development does not eliminate the need for sound management, the approach and focus of the Lean-thinking manager-teacher is different. This important topic is the subject of The Evolving Role of Managers in Lean-Agile Development[6] advanced topic article.

Summary

Effective leadership is critical for achieving a shift to a persistent and relentlessly improving Lean enterprise. For this, we need leaders who know *what* they are trying to do, and *how* they are going to go about it. In other words, we need *Lean-thinking manager-teachers* who understand how to *lead and sustain* the change.

[6] https://scaledagileframework.com/the-evolving-role-of-managers/

3

Team and Technical Agility Competency

"Continuous attention to technical excellence and good design enhances agility."

—Agile Manifesto

Team and Technical Agility is one of the five core competencies of the Lean Enterprise. This competency describes the critical skills and Lean-Agile principles and practices that are needed to create high-performing Agile teams who create high-quality, well-designed technical solutions.

Why Team and Technical Agility?

In Lean enterprises, Agile teams perform most of the work that delivers high-quality technical solutions that reliably meet a customer's needs. As a result, it requires mastery of two essential skill sets:

- *Team agility*. Small, cross-functional Agile teams are formed and aligned to a shared mission and purpose. They work as part of a long-lived, team of Agile teams, known as an Agile Release Train (ART). The ART is organized to maximize value delivery and provides the larger vision, direction, and solution outcomes.
- *Technical agility*. Agile technical principles and practices provide team members with the skills needed to create the best possible solutions. They help ensure systems are architecturally sound, of high quality, and easily modified to support future requirements.

Team Agility

Team agility is the first half of the *Team and Technical Agility* competency. Agile teams are cross-functional and have all skills necessary to define, build, test, and—whenever applicable—deploy value in short iterations. Agile teams operate in collaboration with the Business Owners and other stakeholders to create alignment, common understanding, and quick, predictable delivery of value.

Avoiding the pitfalls of working in functional silos reduces the risk of creating dependencies, which often result in significant delays. Agile teams have the authority and accountability to manage their own work, which in turn increases their productivity and reduces the overall time-to-market.

Applying Agile Methods

SAFe Agile teams often use a blend of Agile methods. They typically use Kanban to facilitate flow and apply the following Scrum practices:

- Work in short (usually two-week) iterations
- Break work into small user stories in the team backlog
- Jointly plan the work for the upcoming iteration
- Meet daily to assess progress toward the iteration goals
- Formally demonstrate a working system at the end of the iteration
- Discuss how to improve the process before starting the next iteration

The Kanban system helps visualize and manage work in process (WIP), thereby preventing teams from starting new user stories before they have finished the work in the queue. Agile teams commit to small batches of work, reducing feedback cycles and adjusting to changing needs as necessary.

Operating as Part of an Agile Release Train

Delivering large and complex systems requires multiple teams to work together as one team, with one mission, as part of an ART. The ART enables teams to collaboratively plan, integrate, demo, deploy, learn, and release together. It does this by organizing all the people needed to define, build, test, and deploy a solution for the customer. Each team understands and commits to achieving not only their objectives, but also the larger ART goals.

While team agility provides the way for teams to work and deliver value effectively in short timeboxes, *technical agility* is essential for building quality in and maintaining a fast, sustainable pace.

Technical Agility

The second half of the *Team and Technical Agility* competency, technical agility defines the Agile software engineering principles and practices that teams use to deliver solutions quickly and reliably. In SAFe, agile software engineering applies Lean-Agile values and principles, practices derived from Extreme Programming (XP), a test-first approach, emergent design, Agile modeling, proven approaches for software design, and more.

Establishing Flow

Agile teams operate in a fast, flow-based system to quickly develop and release high-quality business capabilities, as illustrated in Figure 3-1.

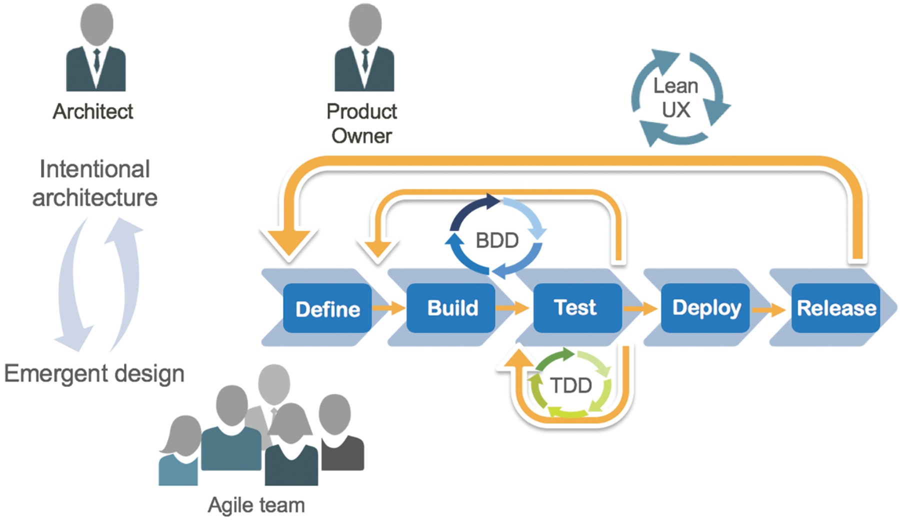


Figure 3-1. Agile Software Engineering in context

Agile teams define and execute many tests early, often, and at multiple levels. Tests are defined for code changes (Test-Driven Development [TDD]) and Story acceptance criteria (Behavior-Driven Development [BDD]). Building in quality supported by test automation ensures that Agile teams' frequent changes do not introduce new errors. Moreover, Agile teams create designs that continually evolve to meet current and future business needs.

Agile teams apply Lean UX practices to quickly and continually release small, minimal marketable features (MMFs) to facilitate learning and feedback. Each new feature is treated as a hypothesis to be validated.

If not properly designed, the solution's architecture can impact flow and inhibit the team's ability to independently release in small batches. To guard against this risk, Agile teams use component- or service-based architectures to support independently developing and releasing solution elements in small batches. System architects and teams collaborate on intentional and emergent architecture to build an architectural runway with a shared technical vision.

Thinking Test-First

Fast flow depends on building quality into the development process by conducting testing at multiple levels. Accordingly, teams write tests first, before coding and developing stories and features. Test-first applies to both functional requirements and non-functional requirements (NFRs) that define performance, reliability, and more. Figure 3-2 illustrates how a test-first approach shifts testing left in the traditional 'V Model' through earlier testing and automation.

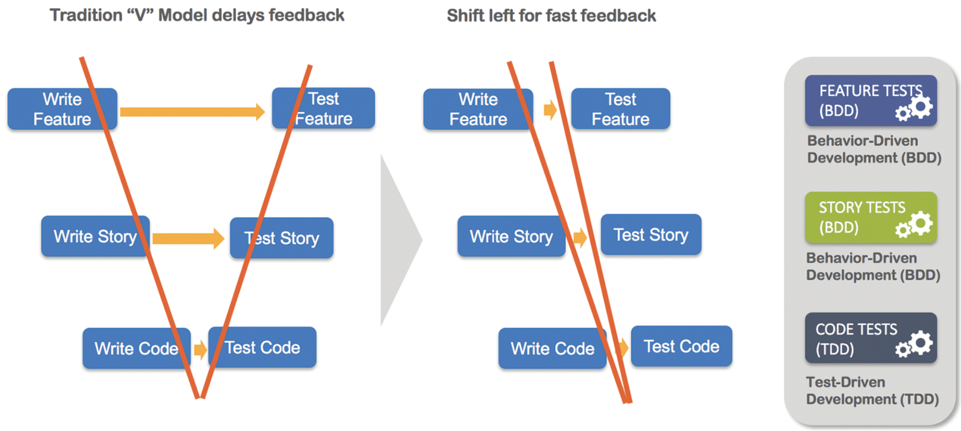


Figure 3-2. BDD and TDD shift testing left

Test are designed and automated to run quickly to support fast value flow. By building many story-level tests, the reliance on slower, end-to-end, expensive tests can often be reduced.

Creating Stories with Behavior-Driven Development

Agile teams apply user stories as small, implementable, and testable statements of system intent. BDD tests are written in nontechnical language that both the business and developers can understand (e.g., a domain-specific language such as Gherkin). BDD provides an approach for building a shared understanding of the software to be developed by discussing examples of the problem to be solved. The domain language allows tests to be quickly automated using the examples. These tests run continuously to maintain built-in quality and ensure new functionality does not break previously existing work.

Modeling Stories

Teams use lightweight models to communicate system design and behavior. Even simple diagrams can illustrate how system entities are related (such as a static model) or how they collaborate to provide system behavior (a dynamic model). Together, structural and behavioral models allow teams to collaborate on the system's current design and determine how to best evolve it into a new design.

Designing for Quality

A solution's code and design quality impact a team's ability to reliably and quickly deliver new functionality. Along with other methods, several XP[1] practices encourage good design, such as coding standards, paired work, collective ownership, and more.

As a system's requirements change, its design must evolve to support them. '*SOLID'* class design principles[2] make systems flexible so they can support new requirements easier. This acronym represents the following code design basics:

- *Single responsibility principle.* Entities should do one thing well. Gather together things that change for the same reason and separate things that change for different reasons.

[1] Beck, Kent. *Extreme Programming Explained.* Addison-Wesley. 1999.

[2] Martin, Robert. Design Principles and Patterns. https://web.archive.org/web/20150906155800/http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf 2000.

- *Open/closed principle*. Entities should be open for extension but closed for modification. Keep existing entities closed for change while extending them to support new behavior.
- *Liskov substitution principle*. Clients of an interface should not depend on a particular implementation and should be able to use any implementation without knowing the details of the implementation.
- *Interface separation principle*. Designers favor multiple client-specific interfaces over one, large, general-purpose interface.
- *Dependency inversion principle*. High-level modules should not depend on low-level modules, and both should depend on abstractions/interfaces.

Other design quality techniques include the following:

- Apply design patterns[3] that provide a general, repeatable solution to a commonly occurring problem.
- Use set-based design that explores multiple solution alternatives to arrive at the best design choice.

Implementing with Quality

Agile teams continually implement quality solutions by using practices such as TDD, refactoring, and emergent design. TDD is a development mindset in which a developer continually creates the smallest possible test and then implements code to pass that test in a rapid cycle.

Test doubles (or mocks, akin to a 'stunt double' in a movie) are used to stand in for system entities that are slow or not readily available. Of course, because a system may change frequently, its design must be able to evolve to better support future changes. Along with TDD, refactoring enables modification of the internal code structure without changing the solution's behavior, allowing the design to evolve over time.

Continuously Integrating and Deploying with Quality

Scaling often requires many engineers to make a lot of small changes, which must be checked for conflicts and errors. Continuous integration keeps all engineers working on the latest version of artifacts by frequently integrating changes. Continuous deployment

[3] Gama, Erich, et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. 1994.

provides fast feedback by automating the process that takes validated features from a staging environment and deploys them into the production environment, where they are readied for release.

Figure 3-3 illustrates a process in which every code commit results in launching an automated build, test, and deployment through the continuous delivery pipeline.[4] The *blue-colored masks* in this figure represent test doubles that reduce testing time and costs.

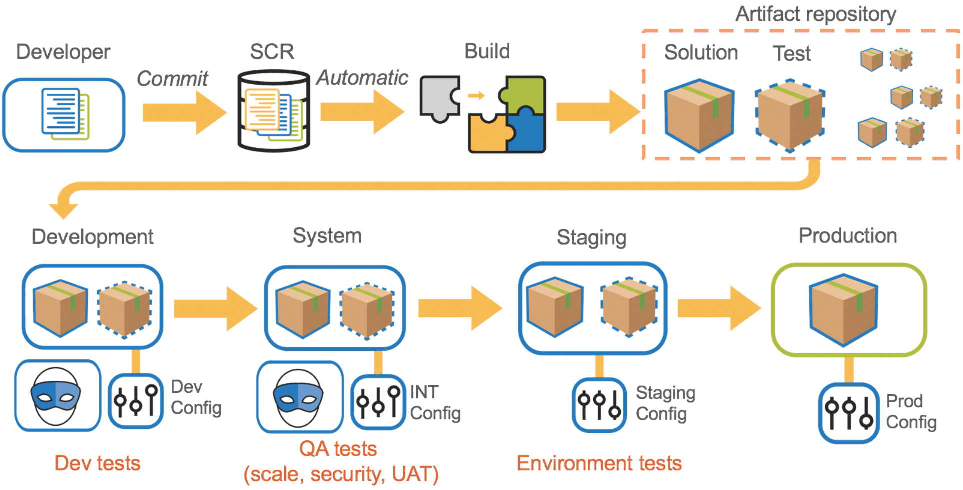


Figure 3-3. Continuous integration and deployment

Summary

Agile teams perform the majority of the work to build solutions that deliver customer value. The Lean enterprise's ability to execute and deliver depends on creating an environment in which these high-functioning Agile teams can flourish. These teams have the Agile knowledge and technical skills necessary to create high-quality solutions to meet current and future business needs.

[4] http://scaledagileframework.com/continuous-delivery-pipeline/

4 DevOps and Release on Demand Competency

"Work is not done when Development completes the implementation of a feature—rather, it is only done when our application is running successfully in production, delivering value to the customer."

—DevOps Handbook

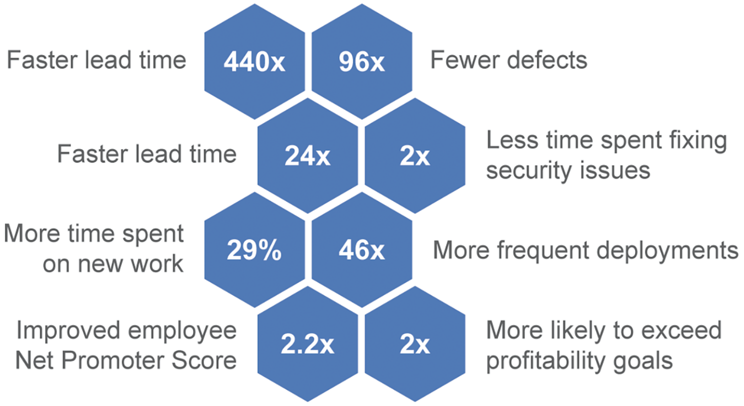
DevOps and Release on Demand is one of the five core competencies of the Lean Enterprise. This competency describes how implementing DevOps and a continuous delivery pipeline provides the enterprise with the capability to release value, in whole or in part, at any time necessary to meet market and customer demand.

The ability to release value to customers when they need it is crucial to success for today's businesses. That is why every Agile team, Agile Release Train (ART), and Solution Train must develop a DevOps mindset and build a 'continuous delivery pipeline.' The pipeline represents the flow of value from concept to cash, through the dimensions of continuous exploration, continuous integration, continuous deployment, and release on demand. DevOps represents the mindsets and practices that make this flow possible.

Why DevOps and Release on Demand?

As digital disruption continues to challenge today's enterprises—and as software becomes a bigger part of every company's ability to deliver and support its products and services—every enterprise faces the need to react to customer demand and needs faster with new digital solutions. In the past, a common impediment to fast delivery has been the gap between development and operations. Developers optimize for frequent releases and change; operations staff optimize for stability and availability. These different motivations create a gap that, if not addressed, can create a barrier to success.

As popularized by books including *The Phoenix Project*[1] and *The DevOps Handbook*,[2] the DevOps movement works to align development, operations, the business, 'Infosec,' and other areas to work together better by sharing the responsibility for improving business results. The reason is simple: High-performing organizations that apply DevOps capabilities dramatically outperform others in terms of both technical aspects and business outcomes, as Figure 4-1 illustrates.


Faster lead time
440x
96x
Fewer defects
Faster lead time
24x
2x
Less time spent fixing security issues
More time spent on new work
29%
46x
More frequent deployments
Improved employee Net Promoter Score
2.2x
2x
More likely to exceed profitability goals

Source: https://puppet.com/resources/whitepaper/state-of-devops-report

Figure 4-1. Puppet Labs state of DevOps survey[3]

Thanks to these extraordinary results from DevOps, *continuous delivery* has become the mantra for this movement. However, although it may reflect a desired end state for many businesses, not all organizations truly need to deliver quite so frequently. All businesses, however, share the need to *release on demand* so that solutions are always ready and available for release any time the business or market demands. This allows Agile teams, ARTs, and Solution Trains to react more quickly to their customers.

[1] Kim, Gene. *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win IT*. Revolution Press. Kindle Edition.

[2] Kim, Gene; Humble, Jez; Debois, Patrick; Willis, John. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press. Kindle Edition.

[3] Puppet Labs state of DevOps survey. https://puppet.com/resources/whitepaper/state-of-devops-report

Organizing Around Value Delivery

All the people who are needed to release on demand must be available to work together continuously to optimize the flow of value. As Figure 4-2 illustrates, the ART is a long-lived, cross-functional team of Agile teams that is intended to accomplish this objective. It includes critical program roles such as Product Management, System Architect/Engineering, and Release Train Engineer.

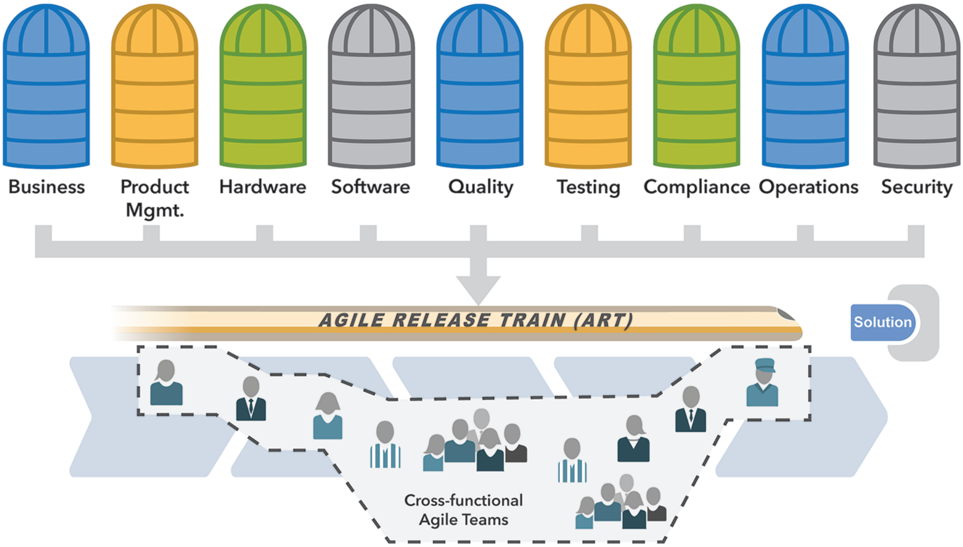


Figure 4-2. ARTs and Agile teams are cross-functional

While we call it 'DevOps,' other industry terms—such as 'DevSecOps' and 'BizDevOps'—emphasize that more than development and operations are needed for release on demand to be achievable. Therefore, all the skills and people necessary—business, compliance, security, data, operations, quality and others—are part of the ART.

Operationally, each ART is a long-lived virtual organization whose members (50–125 people) plan, commit, and execute together. ARTs are organized around the enterprise's significant value streams and exist solely to deliver value to those solutions for which they are responsible.

Figure 4-3 illustrates some key ART events and activities:

- Teams work in short, two-week iterations. Each iteration cycle includes planning, execution, backlog refinement, review, and retrospective.
- ARTs work on an 8- to 12-week program increment cadence.
- The entire ART plans together during PI planning.
- ART representatives meet regularly to assess progress toward the PI objectives in ART syncs, Scrum of Scrums, and PO syncs.
- The ART demonstrates the current working integrated system at the end of each iteration, in system demos.
- In the Inspect and Adapt event, the entire ART discusses how to improve the process.

IMPORTANT NOTE

While ARTs and Agile teams develop on a set cadence, they can release on demand according to market and governance conditions. Simply put, the planning and synchronization cadence does not determine the release cycle; they are two entirely separate concerns.

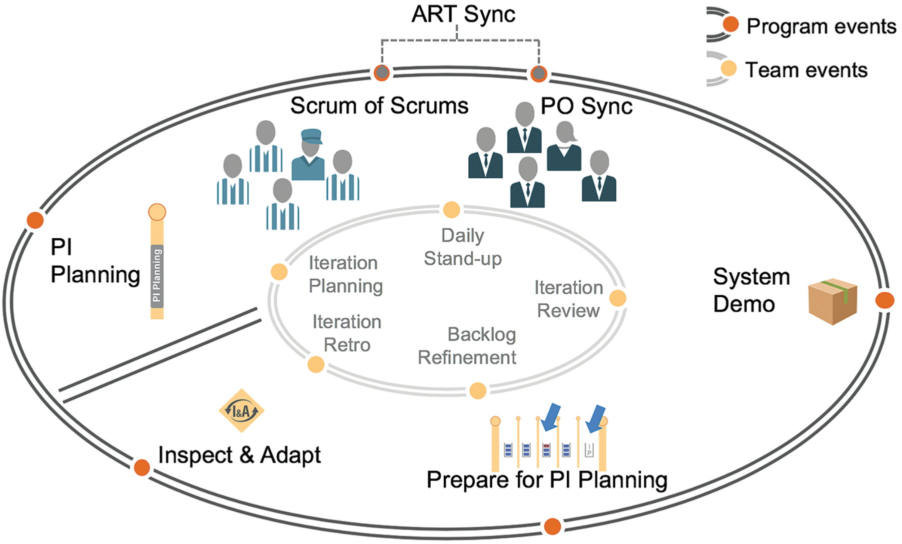


Figure 4-3. ART program and team events

Building the Continuous Delivery Pipeline

To improve the flow of value and enable release on demand, each ART establishes a continuous delivery pipeline. Figure 4-4 illustrates the continuous delivery pipeline that shows the flow of value through four dimensions: continuous exploration, continuous integration, continuous deployment, and release on demand. This flow represents a triple feedback loop, with value flowing to customers, while feedback and learning flow back to development to inform the decisions of what to build next. Each dimension is described in the paragraphs that follow.

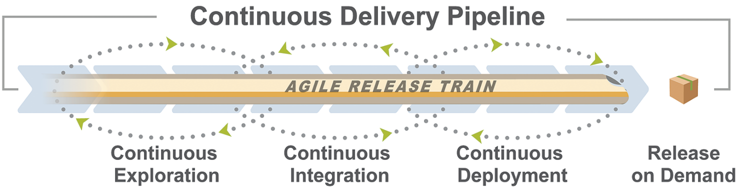


Figure 4-4. The continuous delivery pipeline

Continuous exploration fosters innovation and creates alignment on what needs to be built. It starts with a hypothesis about something that will provide value to customers. This hypothesis is analyzed and researched, and it leads to the development of a Minimal Viable Product (MVP). Once the hypothesis is defined, the solution's architecture is articulated, and features are defined and prioritized in the program backlog. Finally, the ART collaborates during Program Increment (PI) planning on what it will build during the next PI. There are four subdimensions of *continuous exploration,* as illustrated in Figure 4-5.

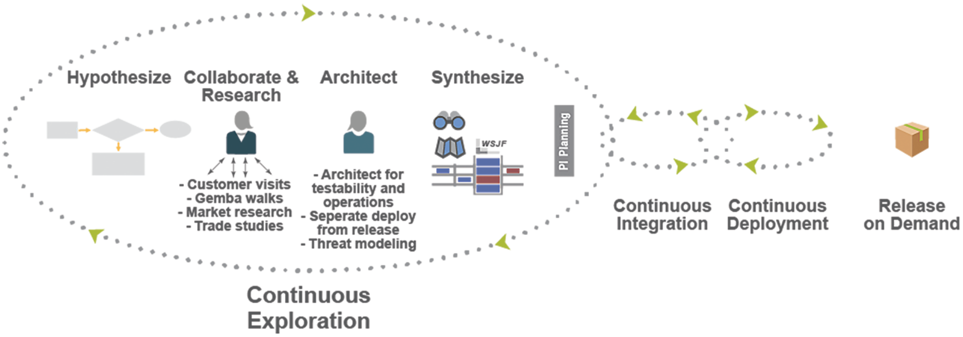


Figure 4-5. The four subdimensions of continuous exploration

Continuous integration builds quality into the development process by integrating the ongoing work of the Agile teams. All work is version controlled, and new functionality is built and continuously integrated into a full system or solution and tested end-to-end before being validated on a staging environment. There are four subdimensions of *continuous integration,* as shown in Figure 4-6.

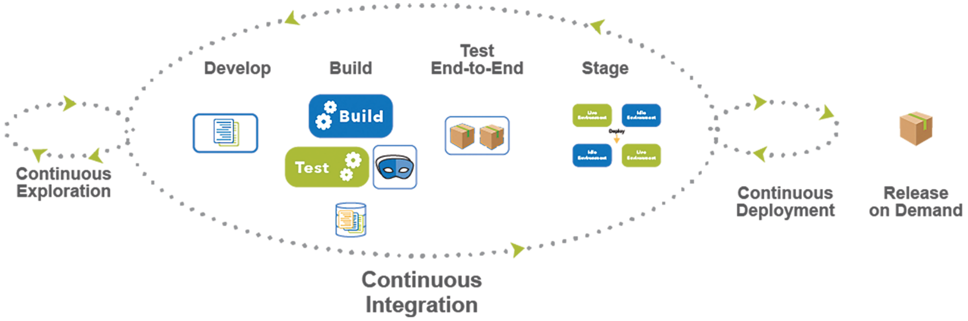


Figure 4-6. The four subdimensions of continuous integration

Continuous deployment takes changes from the staging environment and deploys them to production. At that point, the changes are verified and monitored to make sure they are working properly. This step makes the features available in production, where the business will determine the appropriate time to release them to customers. In addition, this dimension allows the organization to respond, roll back changes, or fix forward when necessary. There are four subdimensions of *continuous deployment,* as illustrated in Figure 4-7.

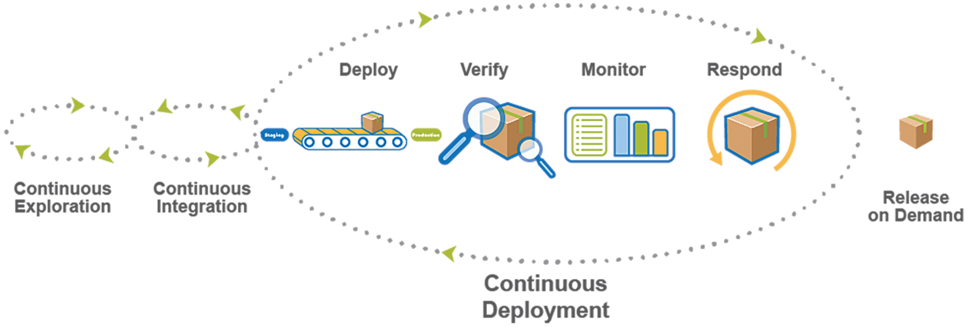


Figure 4-7. The four subdimensions of continuous deployment

Release on demand is the ability to make value available to customers all at once—or in partial increments—based on the needs of the market and the business. Moreover, the release on demand dimension focuses on enabling the business to measure the outcome of the hypothesis and learn what it needs to do next based on objective customer feedback. There are four subdimensions of *release on demand,* as shown in Figure 4-8.

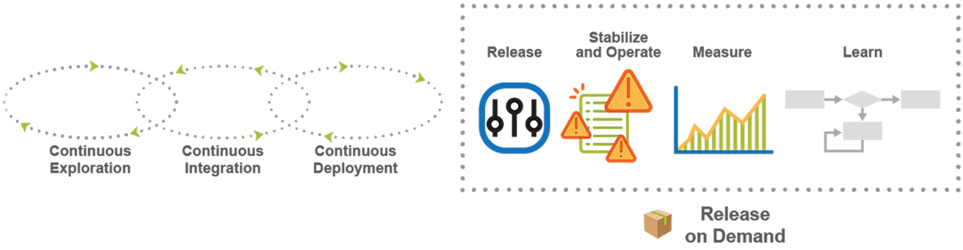


Figure 4-8. The four subdimensions of release on demand

The continuous delivery pipeline also enables mapping (Figure 4-9) and measuring the flow of value from concept to cash, or from hypothesis to validation. Continuous deployment makes it possible for features to be in production and available to be released when the business needs them.

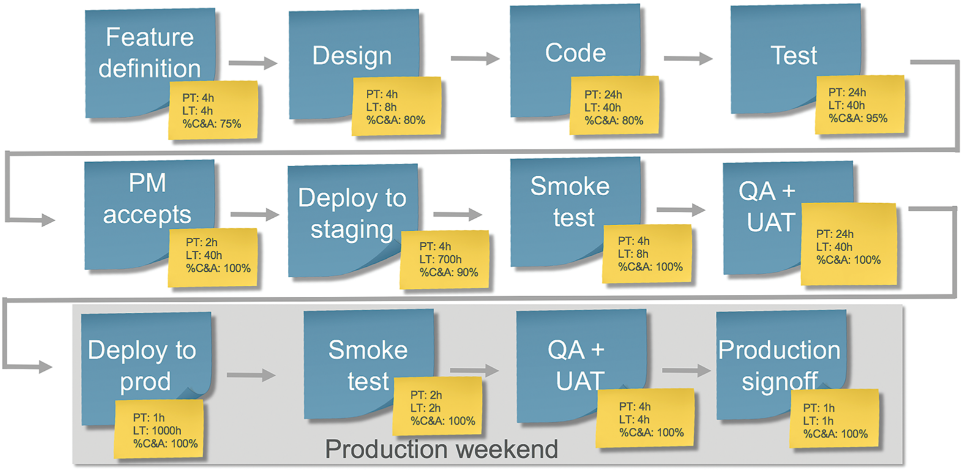


Figure 4-9. Mapping the continuous delivery pipeline

Three primary metrics are used to measure the flow of value in Figure 4-9:

- *Process time (PT):* The time it takes work to get done in one step.
- *Lead time (LT):* The time it takes work to go from the last step until it's done in the current step.
- *Percent complete and accurate (%PC&A):* The percentage of work that the next step can process without it being returned for rework.

Everything Is Continuous

Although a single feature flows through the value stream in a sequential manner, the teams work on all dimensions in parallel (Figure 4-10). That means that ARTs and Solution Trains, throughout every iteration and PI, will continuously engage in the following activities:

- Explore end-user value
- Build, integrate, and demo value
- Deploy to production
- Release value, whenever the business requires

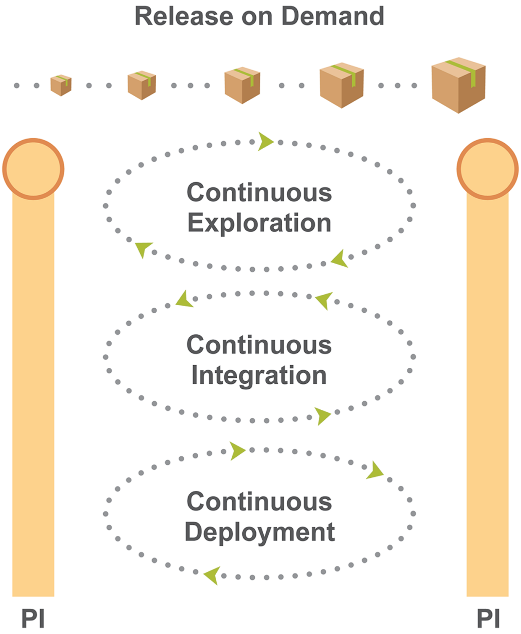


Figure 4-10. Exploration, integration, and deployment happen continuously

DevOps Enables Continuous Delivery and Release on Demand

Each train's DevOps capabilities enable release on demand. As shown in Figure 4-11, SAFe's 'CALMR' (**C**ulture, **A**utomation, **L**ean flow, **M**easurement, and **R**ecovery) approach to DevOps is grounded in five concepts:

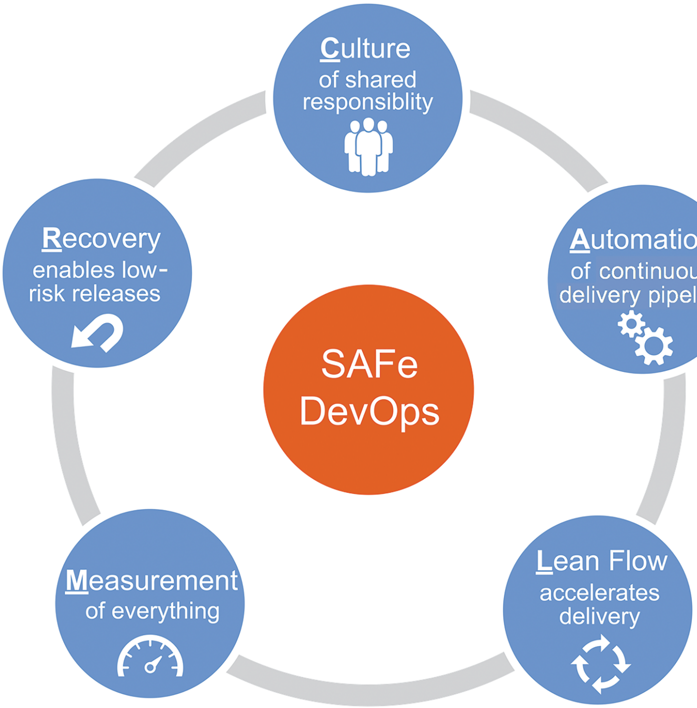


Figure 4-11. A CALMR approach to DevOps

- *Culture* represents the philosophy of shared responsibility for fast value delivery across the entire value stream. This includes everyone who contributes to value creation—from concept to cash—including Product Management, development, testing, security, compliance, operations, and more.
- *Automation* represents the need to remove human intervention from as much of the continuous delivery pipeline as possible to reduce errors, shorten time to market, and improve quality.

- *Lean flow* covers the concepts of limiting work in process (WIP), reducing batch size, and managing queue lengths, which enables the faster flow of value to the customer and faster feedback.
- *Measurement* is about understanding and measuring the flow of value through the pipeline, thereby fostering learning and continuous improvement.
- *Recovery* focuses on building systems that will allow fast fixes of production issues through automatic rollback and ‘fix forward’ (fix in production) capabilities.

Measuring and Advancing DevOps and Release on Demand

As a core competency of the Lean Enterprise, the organization’s ability to release on demand based on DevOps is essential for practitioners to understand. Figure 4-12 shows the *SAFe DevOps and Release on Demand Health Radar*, which helps trains assess their maturity in the 16 subdimensions of the continuous delivery pipeline. This health radar allows ARTs to determine their strengths and weaknesses and to identify the subdimensions that require attention and improvement. The maturity in each subdimension can be scored as ‘sit,’ ‘crawl,’ ‘walk,’ ‘run,’ or ‘fly.’

After the ART or Solution train identifies its problem areas based on its self-assessment, it needs to implement the improvements. Since SAFe enterprises take a systems view, it’s better to move from a ‘crawl’ across all the subdimensions of the radar to a ‘walk,’ rather than improving to ‘fly’ in one subdimension, while leaving the rest at ‘crawl.’ After all, the goal is to improve the flow of value across the entire continuous delivery pipeline, instead of in just one subdimension. The SAFe DevOps health radar is available in the Metrics[4] article on the SAFe website.

[4] https://scaledagileframework.com/metrics/

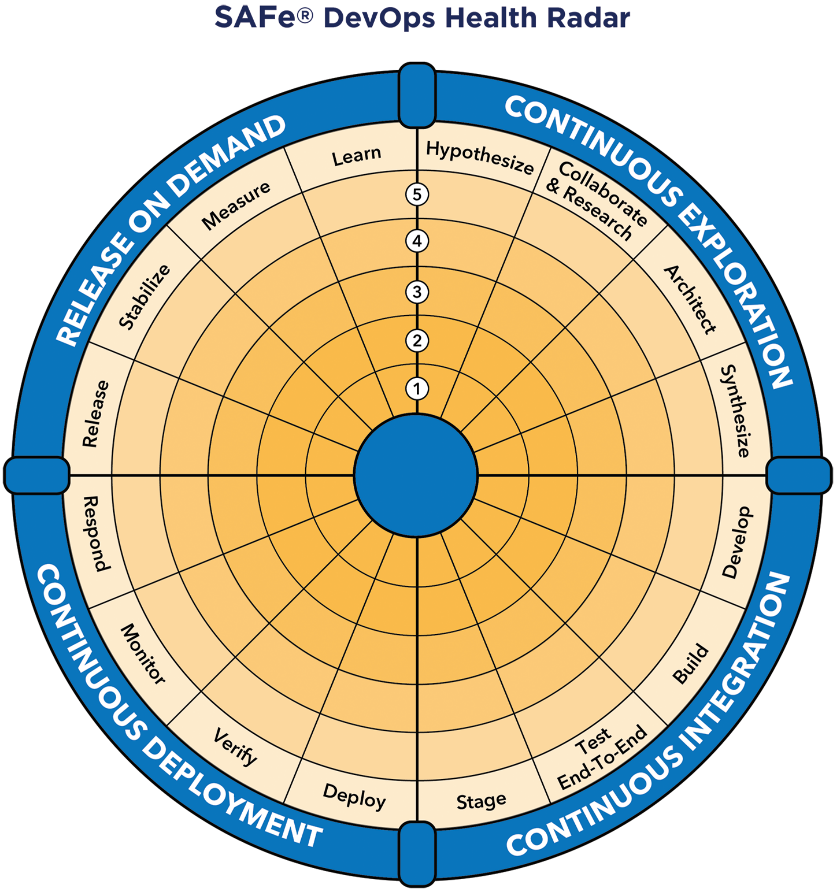


Figure 4-12. The SAFe DevOps health radar

Summary

In today's digital world, the ability to release new solutions on demand is essential. Agile teams enable this ability by responding to change and building quality in, as defined in the Team and Technical Agility competency. Enabling this competency enhances the ability of the Lean enterprise to build solutions and to deliver the strategy defined by Lean Portfolio Management (LPM). Sitting in the middle of the Big Picture, DevOps and Release on Demand serve as critical connectors tying all the competencies together, from Lean-Agile Leadership to LPM. ARTs provide the organizational design metaphor, cadence, and synchronization, while DevOps provides the concepts and practices that make this possible. Like the other competencies, DevOps builds on the foundation of Lean-Agile Leadership, without which neither the cultural shift nor the necessary investment in infrastructure can happen.

5

Business Systems and Lean Systems Engineering Competency

"I am an Engineer. I serve mankind, by making dreams come true."

—Anonymous

Business Solutions and Lean Systems Engineering is one of the five core competencies of the Lean enterprise. This competency describes how to apply Lean-Agile principles and practices to the specification, development, deployment, and evolution of large, complex software applications and cyber-physical systems.

Introduction

Mankind has always dreamed big. Scientists, engineers, and developers turn these big ideas into reality through innovation, experimentation, and collaboration with people from highly diverse fields. As our world becomes more complex and increasingly connected, systems become more unpredictable as they sometimes exhibit unanticipated behaviors.

For example, Figure 5-1 shows the business and technical architecture for an 'autonomous vehicle delivery' solution. It includes both large-scale, distributed multi-platform software applications and a complex cyber-physical system, which must all work together as one large, integrated solution.

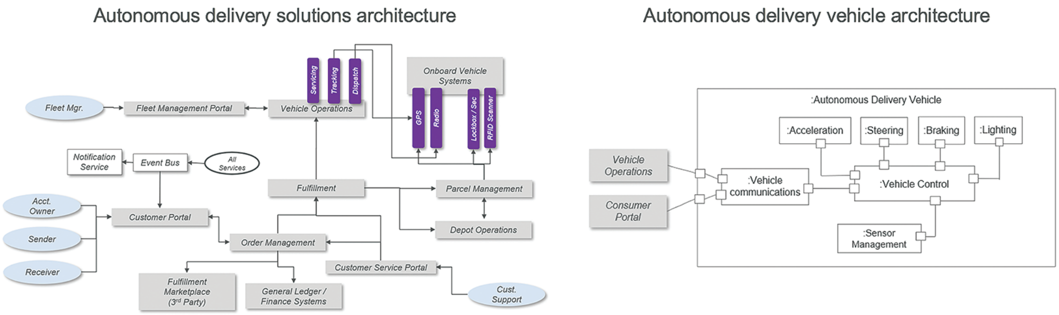


Figure 5-1. Autonomous vehicle systems with a large-scale business solution and Lean systems engineering aspects

Why Business Solutions and Lean Systems Engineering?

Historically, developing large solutions relied on sequential, phase-gated development methods with a 'Big Design Up-Front' (BDUF) phase, enabling the creation of a fixed schedule and budget. Large teams would work in silos on 'their part of the system,' having little or no knowledge of how other components were evolving. Progress was checked periodically through phase-gate milestones. End-to-end integration of the solution's components happened near the end of the project, leaving little time—or budget—to adjust the solution.

The result of this approach was usually higher than expected maintenance and operations expenses, lower profits, and other business problems. As an alternative, Lean Enterprises employ a Lean-Agile flow-based, value delivery model.

Eight Practices for Building Large Solutions with SAFe

While SAFe's principles are designed for scale, an additional set of practices is needed in a large solution context. Figure 5-2 summarizes these practices, which are described in the next sections.

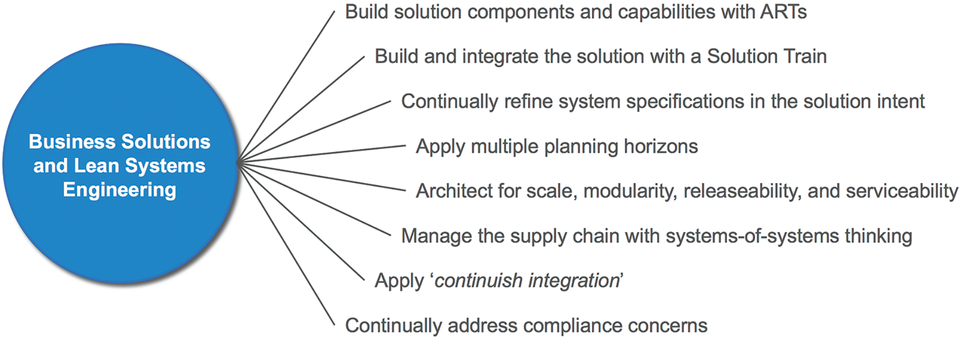

Business Solutions and Lean Systems Engineering
Build solution components and capabilities with ARTs
Build and integrate the solution with a Solution Train
Continually refine system specifications in the solution intent
Apply multiple planning horizons
Architect for scale, modularity, releaseability, and serviceability
Manage the supply chain with systems-of-systems thinking
Apply '*continuish integration*'
Continually address compliance concerns

Figure 5-2. Eight practices for building large solutions with SAFe

1. Build Solution Components and Capabilities with Agile Release Trains

Agile Release Trains (ARTs) are optimized to align and coordinate large groups of people as a team-of-Agile-teams. Large solutions are built by multiple ARTs that build capabilities, components, or a combination of each. ARTs are cross-functional, having all the skills needed to deliver a solution. The Solution Train aligns ARTs to a regular cadence to plan, demo, improve, and learn. The Program Increment (PI) cadence facilitates integration and demonstration of the entire system, allowing adjustment based on new knowledge, which is then planned in the next increment.

2. Build and Integrate the Solution with a Solution Train

Solution Trains coordinate ARTs through a shared business and technology mission. For example, Figure 5-3 shows a Solution Train that coordinates an autonomous vehicle delivery system for a large solution.

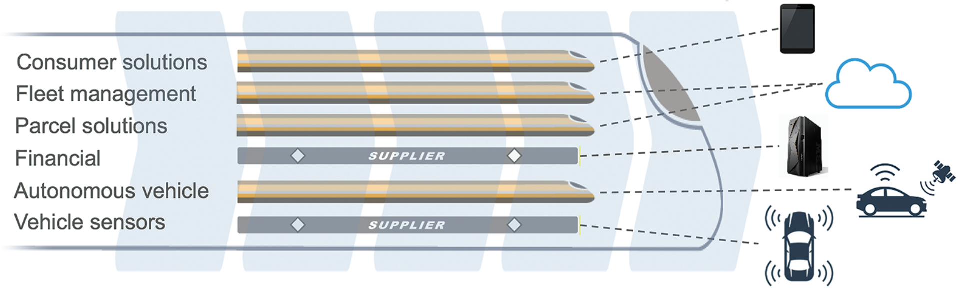


Figure 5-3. Solution Trains coordinate large solution delivery

Solution Trains integrate solutions at least once per PI to validate the current business and technical assumptions. ARTs with long lead times (e.g., those developing hardware) should deliver incremental solutions to support early learning, integration, and validation.

3. Capture and Refine Systems Specifications in the Solution Intent

Solution Trains must understand the intended requirements and anticipated design to meet the business objectives. The solution intent defines the solution's 'as-is' and 'to-be' specifications, serving two primary purposes:

1. Documents the existing requirements and design for validation and compliance
2. Aligns teams on the 'what' and the 'how' of the future functionality

Figure 5-4 illustrates how the solution intent aligns the value stream to a common view of the solution for a particular customer solution context, including logical and physical constraints.

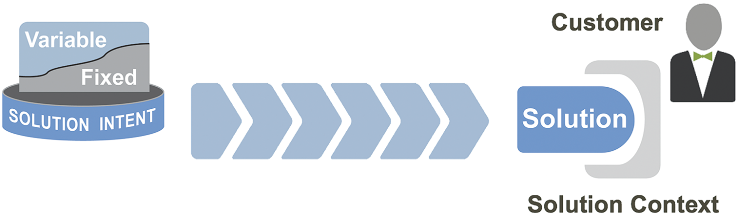


Figure 5-4. Fixed and variable solution intent and solution context

While some requirements and design decisions should be made early (fixed requirements), many can and should be delayed (variable requirements). Economic considerations will determine when exploring alternatives should stop and when decisions should become fixed. In the spirit of Principle #3, '*Assume variability and preserve options*,' SAFe enables the solution intent and context to vary during development so as to achieve the best business and technical results.

The solution intent provides a single source of truth for all requirements and design decisions, which, at scale, reduces the risk of misaligned work. Moreover, modeling, simulation, and prototypes allow teams to test multiple potential solutions and various assumptions quickly and efficiently.

4. Apply Multiple Planning Horizons

ARTs use backlogs and roadmaps to manage and prioritize work and to forecast the schedule. As new knowledge becomes available, work items can be easily changed and reprioritized to ensure delivery of the most valuable things.

SAFe's multiple planning horizons (Figure 5-5) enable better-decentralized decisions, moving short-term, detailed planning to ARTs. For example, the solution roadmap provides a multi-year product vision (2–5 years), while the PI roadmap (3–4 PIs) estimates nearer-term capabilities and milestones. The vision provides the broadest context—the aspirational purpose that creates the boundaries and framework for planning.

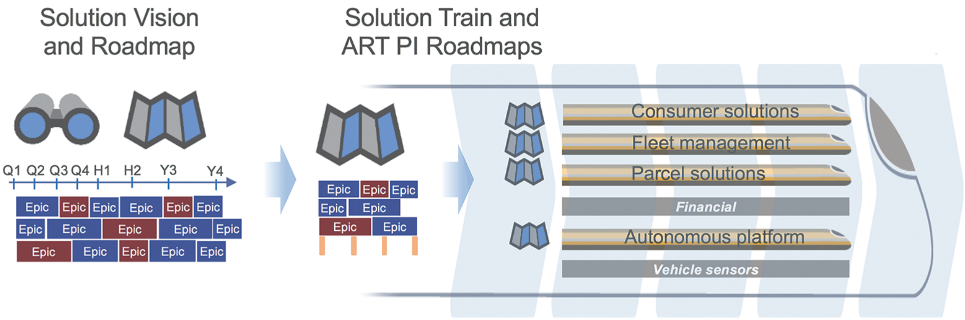


Figure 5-5. Multiple planning horizons facilitate realistic planning

Different levels of backlogs (e.g., solution, program and team) and roadmaps replace the traditional large, program management schedule, which was typically detailed far too early, and often by people who were not doing the actual work. Consequently, these plans were not realistic and received little commitment from teams.

5. Architect for Scale, Modularity, Releasability, and Serviceability

Deciding on an architectural approach too early limits innovation and exploration of perhaps even better technical and economic alternatives. In contrast, *intentional architecture* and *emergent design* informs decisions through a collaboration between architects and teams that occurs throughout the entire development life cycle.

Modular, service-based architectures that communicate through well-defined interfaces reduce dependencies between components, allowing ARTs to independently develop, test, deploy, and even release parts of large solutions. For example, Figure 5-6 shows how individual components of the autonomous delivery solution can be released independently, as long as the interfaces remain stable. Web-based services and mobile apps release frequently, as needed, through wireless updates. In contrast, updating sensors requires taking vehicles out of service and may trigger regulatory compliance requirements that must be met, which changes the economics toward infrequent releases.

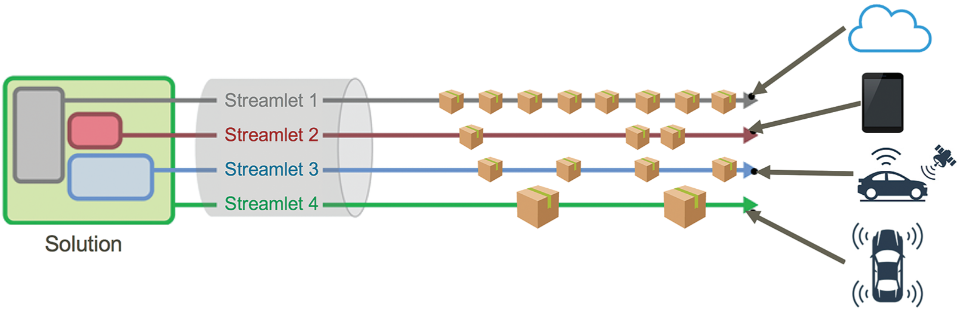


Figure 5-6. Architecture facilitates the ability to release individual system elements

6. Manage the Supply Chain with Systems-of-Systems Thinking

Large solutions must scale to the system-of-systems level and to supply chains. As an example, Figure 5-7 illustrates a supplier hierarchy where the autonomous vehicle platform is composed of a sensor management system, which is composed of a LIDAR system. The solution intent helps align requirements and designs, providing traceability for compliance.

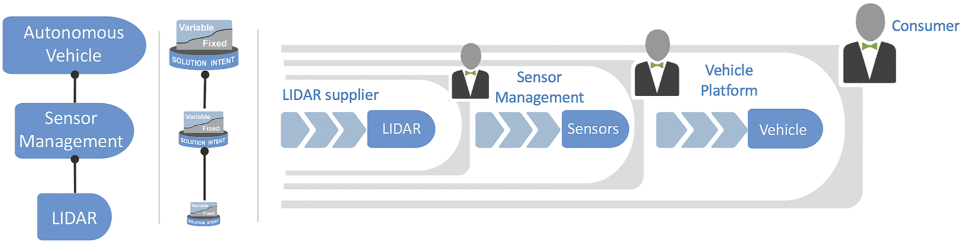


Figure 5-7. Supply chains with system-of-systems

7. Apply 'Continuish Integration'

Integration of large solutions is typically a slow, labor-intensive, expensive effort. Yet, validating assumptions through learning cycles requires integrating the entire system continually, at least once per PI, to demo progress, gain new knowledge, and adapt. Figure 5-8 illustrates how economic trade-offs should determine the optimal investment.

The chart on the left shows the optimal integration frequency based on the cost of integrating the system versus the cost of delayed feedback. The chart on the right shows how investing in build–deploy automation lowers the cost of integrating and allows more frequent integration, reducing the overall cost of delayed knowledge and feedback.

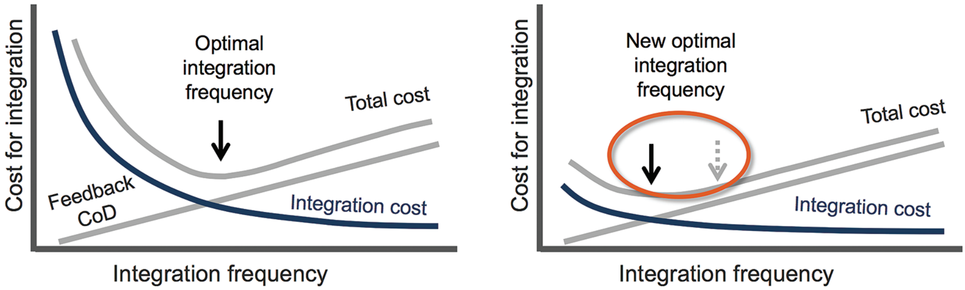
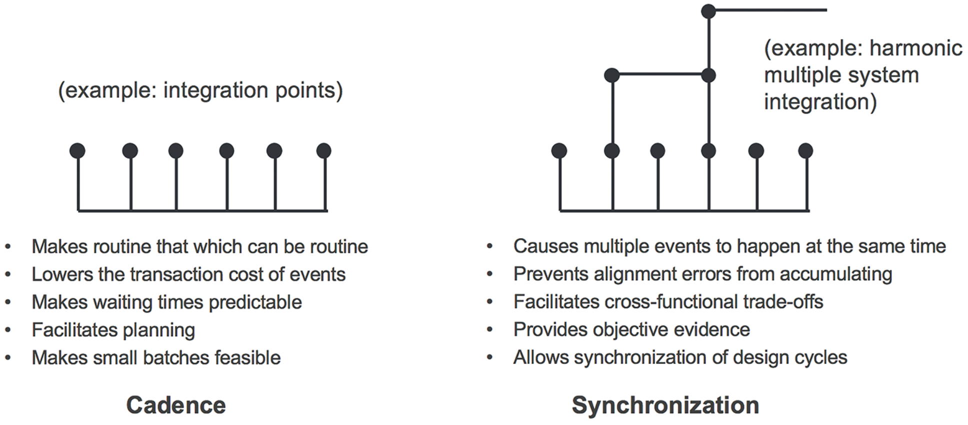


Figure 5-8. Automation changes the economics of frequent integration

‘Continuish integration’ concedes that integration cycles across disciplines will vary due to differences in lead times. Applying cadence and synchronization helps manage inherent variability in solution development. Synchronizing the varying cadences across disciplines enables Lean flow with frequent integration, as shown in Figure 5-9.



- Makes routine that which can be routine
- Lowers the transaction cost of events
- Makes waiting times predictable
- Facilitates planning
- Makes small batches feasible

Cadence

- Causes multiple events to happen at the same time
- Prevents alignment errors from accumulating
- Facilitates cross-functional trade-offs
- Provides objective evidence
- Allows synchronization of design cycles

Synchronization

Figure 5-9. Cadence and synchronization enable frequent integration

Many disciplines use models to learn faster, possibly at a lower fidelity, before building the final product. Models can also support end-to-end integration to demo partial or complete system-level functionality.

Figure 5-10 shows the desired result of continuous integration across the entire Solution Train with frequent contributions from all ARTs. The same economic trade-offs shown in Figure 5-8 apply, but contrast the cost of updating models (versus investing in integration) and the cost of delayed learning.

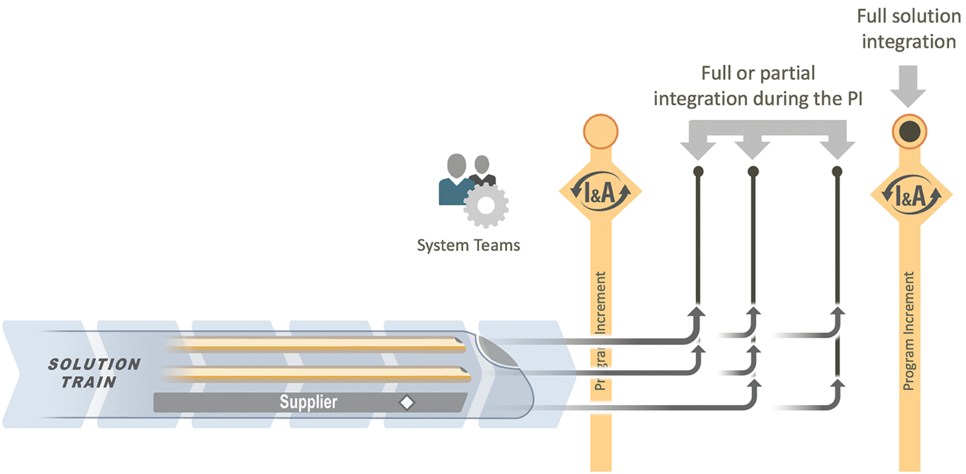


Figure 5-10. Frequently integrate an end-to-end solution

8. Continually Address Compliance Concerns

Many large solutions are often subject to regulatory oversight and compliance requirements. Organizations operating under such regulations may rely on a quality management system (QMS) to help ensure quality and reduce risk.

A Lean QMS ensures compliance with several types of non-functional requirements (NFRs), including regulatory, statutory, industry standards, business constraints, and enterprise architectural goals. Collectively, these NFRs are validated through compliance activities, which appear in a variety of automated and manual processes. They provide continuous evidence of compliance and the associated quality results. For more information, see the article, Achieving Regulatory and Industry Standard Compliance with SAFe,[1] on the SAFe website.

[1] http://scaledagileframework.com/achieving-regulatory-and-industry-standards-compliance-with-safe/

Summary

Organizations must continue to innovate and invest in big ideas to remain competitive. Lean Enterprises are adept at building large solutions that are optimized for fast innovation, learning, and adaptation. This competency offers a new set of practices, based on the fundamental Lean-Agile principles needed to drive development of large solutions. These principles show how to scale and produce large solutions faster, more predictably, and with better quality and fit for purpose.

6

Lean Portfolio Management Competency

"Most strategy dialogues end up with executives talking at cross-purposes because … nobody knows exactly what is meant by vision and strategy, and no two people ever quite agree on which topics belong where. That is why, when you ask members of an executive team to describe and explain the corporate strategy, you frequently get wildly different answers. We just don't have a good business discipline for converging on issues this abstract."

—Geoffrey Moore, author and management consultant

Lean Portfolio Management (LPM) is one of the five core competencies of the Lean Enterprise. It aligns strategy and execution by applying Lean and systems-thinking approaches to strategy and investment funding, Agile portfolio operations, and governance.

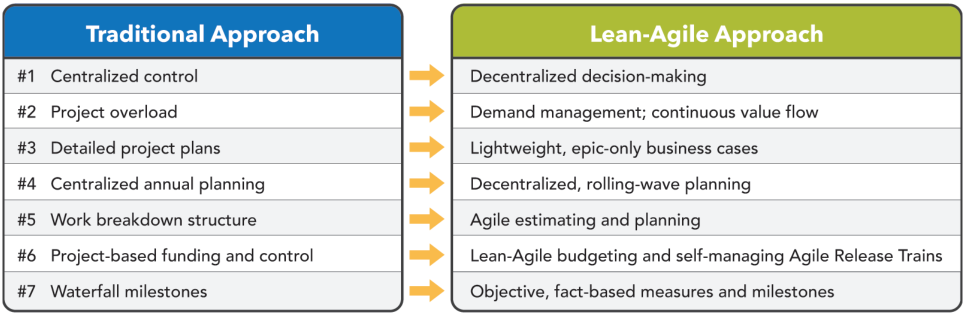
A SAFe portfolio manages a set of value streams for a specific business domain. Each delivers a set of software and system solutions that help the enterprise meet its business strategy, either by providing solutions directly to the customer or by deploying solutions that support internal business processes. In the small-to-midsize enterprise, one SAFe portfolio can typically govern the entire solution set. In larger enterprises, multiple portfolios may be needed, typically structured around the organization and funding model or lines of business.

Why Lean Portfolio Management?

Lean Portfolio Management approaches are needed to keep pace with the impact of digital disruption and the global economy—factors causing enterprises to work with more uncertainty, yet deliver more innovative solutions, much faster. Traditional portfolio approaches were not designed to handle the current, digital, business environment. Even worse, mixing legacy portfolio methods with Lean-Agile teams and trains can result in friction, delays, and significant loss of business agility.

Despite this reality, many legacy portfolio practices persist in today's business world, such as rigid and fixed annual planning and budgeting, document-driven and indirect progress measures of progress, constant overload of demand versus capacity, project-based funding and cost accounting, phase-gate approval processes, and more.

Clearly, portfolio management approaches must be modernized to support the new way of working. Experience shows that the payoff for embracing these changes can be enormous—the difference between riding the waves of digital disruption or drowning under them Fortunately, many enterprises have already gone down this path, and the change patterns are clear (Figure 6-1); many will occur naturally as part of a SAFe transformation.



Traditional Approach		Lean-Agile Approach
#1 Centralized control	→	Decentralized decision-making
#2 Project overload	→	Demand management; continuous value flow
#3 Detailed project plans	→	Lightweight, epic-only business cases
#4 Centralized annual planning	→	Decentralized, rolling-wave planning
#5 Work breakdown structure	→	Agile estimating and planning
#6 Project-based funding and control	→	Lean-Agile budgeting and self-managing Agile Release Trains
#7 Waterfall milestones	→	Objective, fact-based measures and milestones

Figure 6-1. Evolving traditional mindset to Lean-Agile thinking

Implementing Lean Portfolio Management

The LPM function has the highest level of decision-making and financial accountability within a portfolio. While the people who fulfill the LPM function have various titles and roles and are often distributed throughout the organization, they are the individuals who are ultimately responsible for portfolio strategy, operations, and governance.

Figure 6-2 illustrates the three essential collaborations needed to realize the LPM competency. Each of these collaborations is described in the next sections.

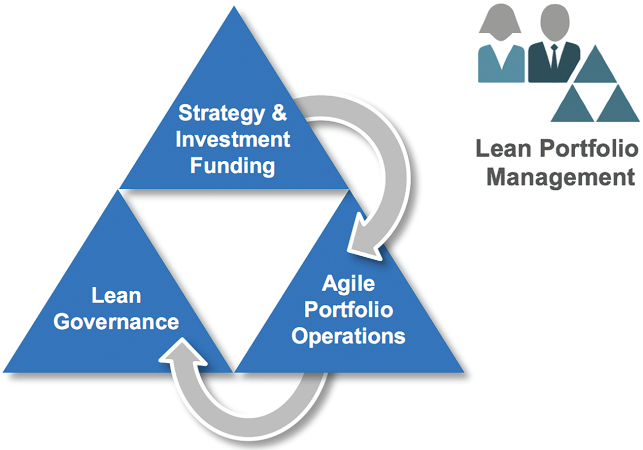

Strategy & Investment Funding
Lean Governance
Agile Portfolio Operations
Lean Portfolio Management

Figure 6-2. Three LPM collaborations

Strategy and Investment Funding

The strategy and investment funding collaboration engages enterprise executives, Business Owners, portfolio stakeholders, and Enterprise Architects, among others. Figure 6-3 illustrates how LPM fulfills four responsibilities for this collaboration. Each is described next.

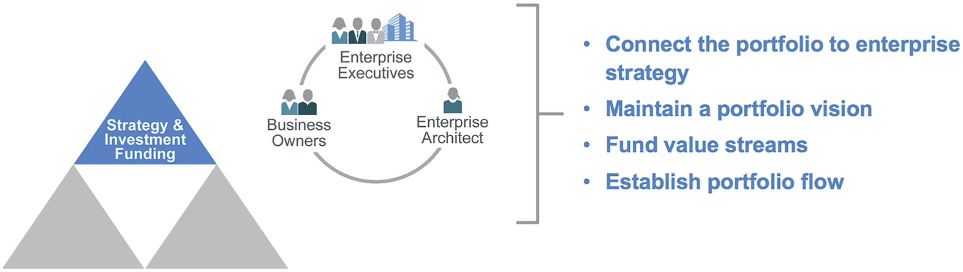

Strategy & Investment Funding
Enterprise Executives
Business Owners
Enterprise Architect
• Connect the portfolio to enterprise strategy
• Maintain a portfolio vision
• Fund value streams
• Establish portfolio flow

Figure 6-3. The strategy and investment funding collaboration

Connect the Portfolio Strategy to the Enterprise Strategy

The portfolio is connected to the business strategy by strategic themes and the budget, and it provides feedback to the organization via the portfolio context (key performance indicators, qualitative data, and Lean budget guardrails). Strategic themes provide the differentiation needed to achieve the desired future state. They help align and communicate the strategy throughout the portfolio.

Maintain a Portfolio Vision

The portfolio canvas has two main purposes: to capture the current state and to define and elaborate the strategy needed to achieve a highly desirable future state. The portfolio strategy and canvas provide critical inputs to the portfolio backlog and Lean budgets. As new information is learned, LPM periodically reviews and updates the canvas.

Fund Value Streams

Lean budgets provide value stream funding aligned with the strategic themes and business strategy. Guardrails, when applied to budgets, help ensure that the right investments are being made within that plan. Specifically, guardrails help steer a course in which LPM approves large initiatives, investment horizons guide spending, and regular investment in technology, infrastructure, and maintenance. This new funding model allows the enterprise to eliminate or reduce the need for traditional project-based funding and cost accounting, which reduces friction, delays, and overhead.

Establish Portfolio Flow

Implementing business objectives requires balancing the amount of work originating from the portfolio with the work that arises as each train responds to customer needs. Portfolio business and enabler epics are used to capture, analyze, and approve new, large business and technology initiatives that require the collaboration of multiple value streams. The portfolio Kanban system is designed to visualize and limit work in process (WIP), reduce batch sizes of work, and control the length of queues for the largest investment initiatives.

Agile Portfolio Operations

SAFe's Lean-Agile mindset fosters the *decentralization* of strategy execution to empowered trains. Even then, however, systems thinking must be applied to ensure that those trains are aligned and operate within the broader enterprise context. Typically, some form of *Agile portfolio operations* is required to accomplish these goals in larger enterprises. The responsibilities of Agile portfolio operations are typically carried out by some combination of an Agile Program Management Office (APMO), Communities of Practice (CoPs), a Lean-Agile Center of Excellence (LACE), and ART and Solution Train Engineers.

Figure 6-4 illustrates the collaboration and responsibilities of this function. Each is described in the next sections.

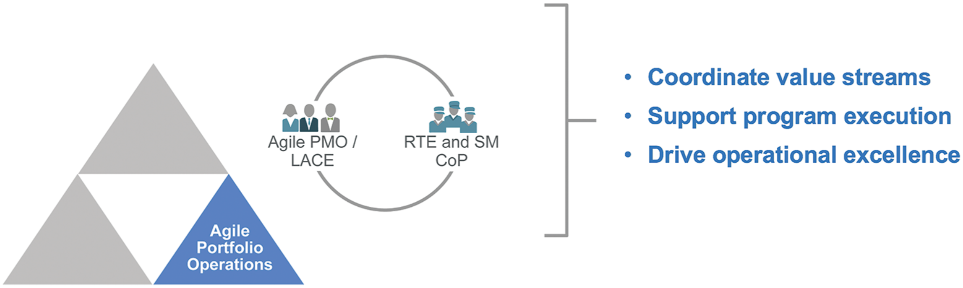

Agile Portfolio Operations
Agile PMO / LACE
RTE and SM CoP
• Coordinate value streams
• Support program execution
• Drive operational excellence

Figure 6-4. Agile portfolio operations collaboration

Coordinate Value Streams

Although many value streams operate independently, cooperation among a set of solutions can provide some portfolio-level capabilities and benefits that competitors can't match. Indeed, in some cases, this is the ultimate goal: to offer a set of differentiated solutions in which new integrated capabilities may emerge to respond to expanding end-user patterns.

Support Program Execution

Many enterprises have discovered that holding on to centralized decision-making and traditional mindsets can delay the move to Lean-Agile practices. In turn, some enterprises have abandoned the PMO approach, moving the majority of their responsibilities to trains. Other organizations are better served by redesigning this function to become an APMO.

The PMO has specialized skills, knowledge, and strong relationships with executives and other key stakeholders, which are extremely useful for SAFe adoption. The APMO often takes on additional responsibilities:

- Sponsor and communicate the change vision
- Actively participate in the rollout and foster the new ways of working

- Lead the move to Lean-Agile budgeting and objective, evidence-based progress reporting
- Foster more Agile contracts and Leaner supplier and customer partnerships

The LPM function—working in collaboration with the APMO and LACE—can help develop, harvest, and apply successful program execution patterns across the portfolio.

Drive Operation Excellence

LPM—or, by proxy, the APMO—also has a leadership role in helping the organization relentlessly improve. This leadership is often supported by a persistent LACE, which can be a part of the APMO. The APMO may also sponsor and support CoPs for Release Train Engineers (RTEs) (and Solution Train Engineers [STE]s), as well as Scrum Masters. These role-based CoPs provide a forum for sharing effective Agile program execution practices and other institutional knowledge. The APMO can also establish and maintain the systems and reporting capabilities that ensure the smooth deployment and operation of the value stream investment.

Lean Governance

In the final collaboration, Lean governance influences spending, guides future expense forecasts and milestones, and provides high-level oversight of development. These stakeholders and their responsibilities are shown in Figure 6-5 and are described in the following sections.

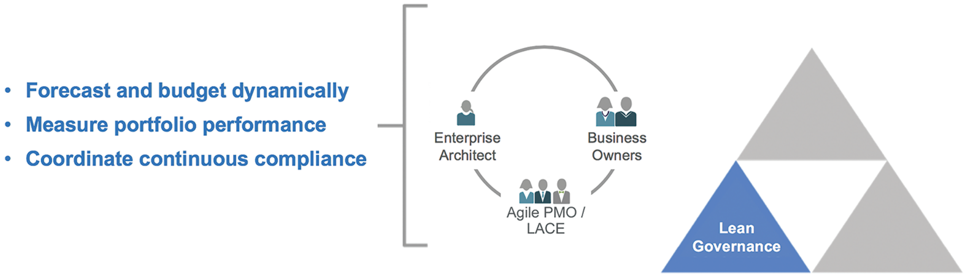

• Forecast and budget dynamically
• Measure portfolio performance
• Coordinate continuous compliance
Enterprise Architect
Business Owners
Agile PMO / LACE
Lean Governance

Figure 6-5. Lean governance collaboration and responsibilities

Forecast and Budget Dynamically

SAFe provides a Lean budgeting process that includes Agile approaches for estimating, forecasting, and longer-term roadmapping. While value streams are primarily self-organizing and managing, they require management's approval for funding. As a result, LPM leads the tuning of the value stream budgets over time. Funding will evolve based on changing business needs but should be adjusted only at PI boundaries, as Figure 6-6 illustrates.

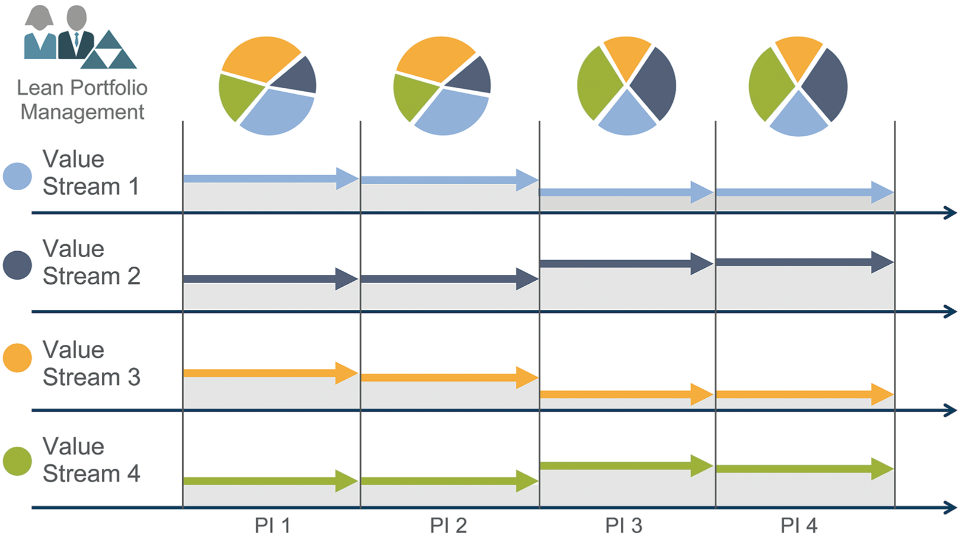


Figure 6-6. Value stream budgets are adjusted dynamically over time

Typically, these budgets are adjusted twice annually. When adjustments occur less frequently than that, spending may remain fixed for too long, limiting agility. Conversely, more frequent changes to the budgets may seem to support higher levels of agility but can create too much uncertainty and an inability to commit to any near-term course of action.

Measure Lean Portfolio Performance

Each portfolio must also establish the minimum metrics needed to assure:

- The implementation of the strategy
- Alignment of spending with the agreed-upon boundaries
- Continual improvements in business and technical results

Lean measures for assessing the internal and external progress for a portfolio can be found in the Metrics[1] article on the SAFe website.

Applying Innovation Accounting

Many seemingly desirable portfolio measures are *lagging* economic indicators. For example, metrics for return on investment (ROI) and new markets penetrated can take a long time to obtain. Instead, the organization needs fast feedback from *leading* indicators, many of which are not financial metrics. "To address uncertainty and speed time to market, a different kind of accounting framework is needed to quickly validate product assumptions and increase learning."[2] *Innovation accounting* serves this purpose by empirically measuring and communicating the true progress of innovation. At the portfolio level, applying innovation accounting is directly encouraged by the epic hypothesis statement and Lean business case artifacts and guidance.

Portfolio Sync

In addition to the ART Sync, some enterprises hold a 'Portfolio Sync' meeting, which typically occurs biweekly, or more or less frequently as needed. The LPM function, APMO personnel, or other appropriate portfolio stakeholders may facilitate the Portfolio Sync. Its purpose is to gain visibility into how well the portfolio is progressing toward meeting its strategic objectives, including reviewing value streams and program execution and other investments governed by the portfolio.

Coordinate Continuous Compliance

Large solutions are typically subject to audit and compliance[3] requirements. These may include internal or external financial auditing constraints and industry, legal, or regulatory guidelines. Compliance requirements may impose significant limits on solution development and operations. Traditional approaches to compliance tend to defer these activities to the end of development—a practice that subjects the enterprise to the risk of late discovery and subsequent rework, and may even lead to compromising legal and regulatory exposure. A more continuous approach, coordinating ongoing compliance with relevant standards, is recommended.

[1] http://scaledagileframework.com/metrics/

[2] Ries, Eric. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses.* Crown Publishing Group.

[3] http://scaledagileframework.com/compliance/

Summary

As Moore's quote informs us, the supposedly easy task of 'agreeing on strategy' is not such a simple thing in the enterprise. Only by allocating the *right investments* toward building the *right things* can an enterprise achieve its ultimate business objectives. The LPM competency is designed to address this challenge, which is implemented by the LPM function. In turn, the LPM function provides a set of three collaborations—strategy and investment funding, Agile portfolio operations, and Lean governance—intended to bring leadership together. Each has a set of stakeholders and responsibilities.

7

SAFe for Government

"If there's one thing government needs desperately, it's the ability to quickly try something, pivot when necessary, and build complex systems by starting with simple systems that work and evolve from there, not the other way around."[1]

—Jennifer Pahlka, Founder, Code for America,
Former U.S. Deputy CTO 2012

SAFe for Government

SAFe for Government is a new domain in SAFe. Here, new articles describe a set of success patterns that help public-sector organizations implement Lean-Agile practices in a government context.

Why SAFe for Government?

Although the transformation challenges in government look similar to those faced by commercial companies, the organizational context, culture, and governance authorities in the public-sector environment are truly unique.

Government acquisition processes and laws are intended for a different purpose than are commercial constructs. They seek to create a fair playing field among potential providers so that the public gets the best value for its money. But these same regulations and practices can also create bureaucracy, delays, and waste greater than that seen in the private sector. Moreover, government agencies do not typically have competitive market pressure or the profit motives that drive the cost-efficiency and rapid innovation in a

[1] Pahlka, Jennifer. Coding a better government. http://bit.ly/2GwFhO1

commercial environment. Funding is different, too, and far less incremental. Funding is typically provided by legislative bodies in an annual appropriations process that moves slowly. Even the concept of ‘value’ in a government technology program is often difficult to understand and quantify.

Even with these differences, just like for-profit organizations, the government is increasingly dependent upon technology. And just like private-sector organizations, government’s traditional approaches for developing and supporting new technologies have proved insufficient for addressing 21st-century concerns. Agile practices have shown promise in the public sector. However, the size and complexity of government systems—ranging from an unemployment benefits website for French citizens to the U.S. F-22 fighter jet—require far more than team-level Agile practices can provide.

Brief History of SAFe Adoption Across Government

U.S. Government

Since 2012, the increased interest in Lean, Agile, and DevOps accelerated when two new U.S. government agencies, 18F and U.S. Digital Service, were created to help bring modern development practices to U.S. federal IT programs. The U.S. government’s “Digital Services Playbook”[2] and “TechFAR Handbook”[3] were two early resources used by government programs to modernize development practices and to change the acquisition process to support Agile contracts.

Since then, the number of additional government-authored resources on Agile adoption has grown significantly, as has the number of published success stories of federal programs getting better results after their transformation to Lean-Agile practices. For example, the U.S. Department of Homeland Security has made Agile its formal standard for software development. In support of this trend, the Government Accountability Office (GAO) and the Office of Management and Budget (OMB) are publishing audit and budgeting regulations for Agile programs. Congress has also directed increased training in Agile as well as authorized the modernization of acquisition practices to support Agile.

[2] https://playbook.cio.gov/

[3] https://playbook.cio.gov/techfar/

Global Governments

Similar trends are being experienced in the development of systems for state and local governments, as well as governments across the globe. For example, the U.K. government has engaged in an Agile transformation effort that spans hundreds of development teams across many departments and agencies. Moreover, the French employment agency (Pôle Emploi), the Dutch Tax Authority, and the Australian Postal Service have used SAFe to guide their transformations to Lean-Agile practices at scale. In the case of Pôle Emploi, the transformation has resulted in better on-time delivery of employment benefits and increased satisfaction with the agency's services for both hiring businesses and job seekers.

SAFe Adoption in Government Is Growing

"In 10 months', time, we have turned a failing effort into a success story for the warfighter, for our organizational culture, and for the taxpayer. We could not have executed a turnaround this fast without SAFe."

— Scott Keenan, JLVC Program Manager, U.S. Department of Defense

Many government technology programs are large and complex, involving hundreds (sometimes thousands) of practitioners. Solutions are now often built by multiple Agile teams of teams that perform the following activities:

- Plan and work together
- Manage cross-team dependencies
- Integrate frequently
- Demonstrate working software and systems iteratively
- Share learning for relentless improvement

Often, these large solutions include small groups of government employees working closely with large numbers of contractor personnel. Multiple suppliers on different contracts may work on the same program across dispersed geographies. High assurance and compliance requirements, governance regulations, and abnormally long acquisition lead times further complicate government projects.

In the last five years, an increasing number of government agencies have adopted SAFe as the process model for technology development for all the same reasons that attracted their commercial counterparts to SAFe. As shown in Figure 7-1, SAFe is now being used in hundreds of programs across a large number of government agencies.

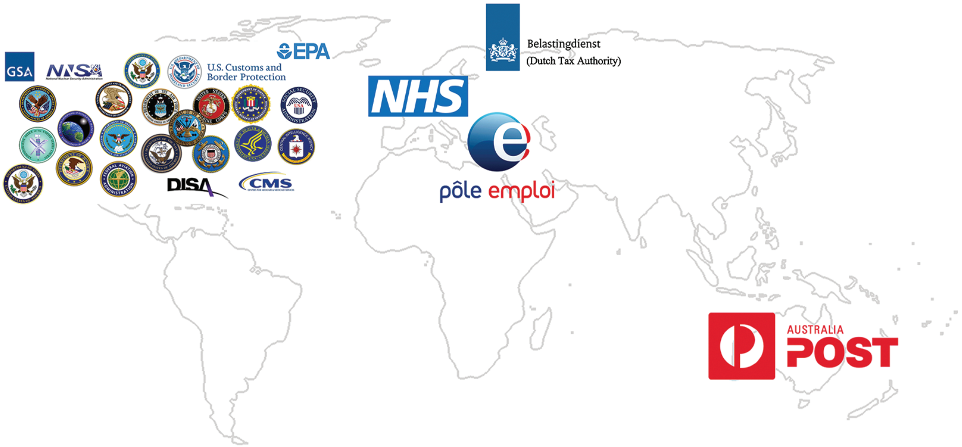


Figure 7-1. Global government agencies with programs using SAFe

Lean-Agile Adoption Challenges in Government

Even with the increased momentum toward Lean-Agile practices and use of SAFe, several barriers have delayed widespread government adoption of this approach. The most frequently cited challenges include the following:

- Poor implementations of Agile in the past, creating a reluctance to try it again
- Waterfall-centric governance and life-cycle policies that are not easily changed
- An acquisition workforce that lacks experience with Agile contracts and Lean contracting practices
- A project-oriented (rather than Agile), continuous-flow-of-value mindset, that is deeply ingrained in government culture
- Long acquisition life cycles that create waste and delays in delivering value
- Lack of a common enterprise Lean-Agile framework, which leads to limits synergy between programs

The Solution: SAFe for Government

Government as a Lean Enterprise

Increasingly, government agencies are being challenged by similar forces of change that are driving their commercial counterparts to accelerate their own Lean-Agile transformations. Digital disruption, globalization, ever-increasing cyber-threats, aging legacy systems, and increasing dependency on technology for business and mission success are just a few of those factors.

The hallmark of a Lean Enterprise is the ability to deliver *the best quality and value in the sustainably shortest lead time*. SAFe provides guidance by describing the success patterns that help organizations achieve these competencies. The evidence provided by the many case studies of SAFe implementations in government agencies suggests that these competencies are as applicable to public-sector organizations as they are for commercial enterprises.

SAFe's Specific Guidance for Government

Practitioners who have used SAFe in both contexts report that there are far more similarities in development between private industry and government than there are differences. That contention makes sense because the system that is being built doesn't know who it is being built *for*. In fact, experienced practitioners report that they achieve the best results when the SAFe model, principles, practices, and terminology are used without modification.

Thus, Lean-Agile adoption in government does not require a different version of SAFe or require modifying SAFe terms and practices to fit government protocols. As a consequence, program personnel can take advantage of the many different resources—classes, articles, consultants, books, forums, and other information—that are necessary to successfully implement SAFe practices.

However, because of specific factors in government technology development, additional guidance is required to help change agents as they lead public-sector transformations. The practices that follow address the main challenges of adopting SAFe in government programs.

1. *Building on a solid foundation of Lean-Agile values, principles, and practices.*[4] Waterfall ways of thinking and processes are deeply ingrained in government technology programs. Government leaders and practitioners alike, along with their industry partners, must understand how and why the Lean-Agile mindset is fundamentally different from past approaches to technology development.
2. *Creating high-performing teams of teams of government and contractor personnel.*[5] Lean-Agile development is a team sport. Teams working on government programs often include a combination of government and contractor personnel. Due to competing interests, this relationship is often strained, inhibiting the development of high-performing teams and ultimately the rapid delivery of valuable, quality solutions.
3. *Aligning technology investments with agency strategy.*[6] Government technology programs can be proposed, approved, and funded for a variety of reasons. Technology investments are often made on the basis of assumed funding of past initiatives, rather than using limited development funds to ensure that investments are aligned with the current strategic imperatives of the agency.
4. *Transitioning from projects to a Lean flow of epics.* The nature of government projects often leads to committing to point solutions too early in the process. Moreover, committing to delivering *all planned features* can be wasteful since the highest-priority features deliver most of the value. Instead, the Lean approach implements large initiatives (Epics) using a flow-based model that delivers features incrementally.
5. *Adopting Lean budgeting aligned to value streams.* Along with the shift from projects to a lean flow of Epics, there is a change in how budgeting is performed. Instead of funding individual pieces of work, budgets fund "the factory," which can build whatever the agency needs based on evolving priorities.
6. *Applying Lean estimating and forecasting in cadence.* Traditional funding practices often lock programs into the 'one right answer' created up front and used to define contractual terms with multiple vendors. Lean estimating and forecasting techniques are lightweight and provide needed adaptability to constantly changing conditions, while still providing critical reporting and accountability.

[4] http://scaledagileframework.com/building-a-solid-lean-agile-foundation/

[5] http://editor.scaledagileframework.com/creating-high-performing-teams-of-teams/

[6] http://scaledagileframework.com/aligning-technology-investment-with-agency-strategy/

7. *Modifying acquisition practices to enable Lean-Agile development and operations.* The acquisition process is the major barrier to adoption of SAFe in government agencies. Contracting officers still depend on approved boilerplate language for executing traditional technology projects and programs. That results in new acquisitions that are firmly grounded in terms and conditions that inhibit agility.
8. *Building in quality and compliance.* A continuous flow of value is not possible if the compliance processes, and governance processes such as verification and validation, remain big-batch activities performed only at the end. Every part of the technology value stream—including compliance and governance—must also work in smaller batches using a continuous delivery pipeline enabled by DevOps.
9. *Adapting governance practices to support agility and lean flow of value.* Traditional governance requires programs to plan everything up front, provide detailed project plans for unknown work, pass (and document_heavy deliverables through arbitrary phase gates, and more. Lean-Agile methods provide sufficient oversight to ensure the delivery of mission-enabling capabilities but within a far more reasonable timeline and cost.

Summary

Government adoption of Lean-Agile is significantly accelerating. However, Agile practices are often limited to development teams and do not address the program and portfolio challenges of strategy alignment, budgeting, project-centric planning, acquisitions, governance, compliance, and more. Agencies also lack a common language and set of enterprise-wide practices to create synergies between Agile programs and practitioners.

The SAFe for Government guidance, supported by the SAFe for Government course,[7] equips agency leaders with the tools needed to overcome many common barriers to adoption of Lean-Agile practices, enabling better mission results.

[7] https://www.scaledagile.com/safe-for-government/

Glossary

The Scaled Agile Framework (SAFe) 4.6 glossary defines all of the terms on the Big Picture.

Agile Architecture

Agile Architecture is a set of values and practices that support the active evolution of the design and architecture of a system while implementing new system capabilities.

Agile Release Train (ART)

The Agile Release Train is a long-lived team of Agile teams and stakeholders that incrementally delivers (and, where applicable, operates) one or more Solutions in a value stream.

Agile Team

The SAFe Agile Team is a cross-functional group people who have the responsibility to define, build, test, and where applicable deploy and release, some element of Solution value—all in a short Iteration timebox.

Architectural Runway

The Architectural Runway consists of the existing code, components, and technical infrastructure needed to implement near-term features without excessive redesign and delay.

Built-In Quality

Built-In Quality practices ensure that each Solution element, at every increment, meets appropriate quality standards throughout development.

Business Owners

Business Owners are a small group of stakeholders who have the primary business and technical responsibility for governance, compliance, and return on investment (ROI) for a Solution.

Business Solutions and Lean Systems Engineering

The Business Solutions and Lean Systems Engineering competency describes how to apply Lean-Agile principles and practices to the specification, development, deployment, and evolution of large, complex software and cyber-physical systems.

Capability

A Capability is a higher-level Solution behavior that typically spans multiple ARTs. It is sized and split into multiple features to facilitate its implementation in a single PI.

Communities of Practice (CoPs)

Communities of Practice are people who share a common interest in a specific domain and who collaborate regularly to share information, improve their skills, and actively work on advancing their knowledge.

Compliance

Compliance refers to a Lean-Agile development strategy that allows teams to build systems that have the highest possible quality and meet the relevant regulatory or industry standards.

Continuous Delivery Pipeline

The Continuous Delivery Pipeline represents the workflows, activities, and automation needed to move a new piece of functionality all the way from ideation to release.

Continuous Deployment (CD)

Continuous Deployment is the process that takes validated Features from a staging environment and deploys them into the production environment, where they are readied for release.

Continuous Exploration (CE)

Continuous Exploration is the process of continually exploring Customer and market needs, which fosters innovation and builds alignment on the Vision, Roadmap, and Feature set for a Solution.

Continuous Integration (CI)

Continuous Integration is the process of developing, testing, integrating, and validating Features in a staging environment where they are readied for deployment and release.

Core Values

The four Core Values of alignment, built-in quality, transparency, and program execution represent the fundamental beliefs that are key to SAFe's effectiveness.

Customers

Customers are the ultimate buyers of every Solution. They are an integral part of the Lean-Agile development process and the Value Stream and have specific responsibilities in SAFe.

Dev Team

The Dev Team is a subset of the Agile team, which consists of dedicated professionals who can develop, test, and deploy a Story, Feature, or component.

Develop on Cadence

Develop on Cadence is a method for managing the inherent variability of systems development by making sure important events and activities occur on a regular, predictable schedule.

DevOps

DevOps is a mindset, a culture, and a set of technical practices, which provide communication, integration, automation, and close cooperation for planning, developing, testing, deploying, releasing, and operating a Solution.

DevOps and Release on Demand

The DevOps and Release on Demand competency describes how implementing DevOps and a continuous delivery pipeline provides the enterprise with the capability to release value, in whole or in part, at any time necessary to meet market and customer demand.

Economic Framework

The Economic Framework is a set of decision guidelines that align everyone with the financial objectives of the Solution and inform the economic decision-making process.

Enablers

Enablers support the activities needed to extend the Architectural Runway to provide future business functionality. These include exploration, infrastructure, compliance, and architecture development.

Enterprise

The Enterprise represents the business entity to which each SAFe portfolio belongs.

Enterprise Architect

The Enterprise Architect promotes adaptive design and engineering practices, and drives architectural initiatives for the portfolio.

Epic

An Epic is a container for a Solution development initiative large enough to require analysis, the definition of a Minimum Viable Product (MVP), and financial approval before implementation.

Epic Owners

Epic Owners are responsible for coordinating portfolio Epics through the Portfolio Kanban system.

Essential SAFe Configuration

The Essential SAFe configuration is the basic building block for all SAFe configurations and is the simplest starting point for implementation. It brings the core competencies of *Lean-Agile Leadership*, *Team and Technical Agility*, and *DevOps and Release on Demand* to the enterprise.

Feature

A Feature is a service that fulfills a stakeholder need and is sized to be delivered by a single Agile Release Train (ART) in a Program Increment (PI).

Foundation

The Foundation contains the supporting principles, values, mindset, implementation guidance, and leadership roles needed to deliver value successfully at scale.

Full SAFe Configuration

The Full SAFe configuration is the most comprehensive version of the framework that applies all five core competencies to support enterprises that build and maintain a portfolio of large, integrated Solutions.

Innovation and Planning (IP) Iteration

The Innovation and Planning Iteration provides dedicated time for innovation and learning, PI Planning, and Inspect and Adapt (I&A) events; it also serves as an estimating buffer for meeting PI Objectives.

Inspect & Adapt (I&A)

The Inspect and Adapt is a significant event, held at the end of each Program Increment (PI), during which the current state of the Solution is demonstrated and evaluated by the train.

Iteration

Each iteration is a standard, fixed-length timebox, during which Agile teams deliver incremental value in the form of working, tested software and systems.

Iteration Execution

Iteration Execution is how Agile teams manage their work throughout the Iteration timebox, resulting in a high-quality, working, tested system increment.

Iteration Goals

Iteration Goals are a high-level summary of the business and technical goals that the Agile Team agrees to accomplish in an Iteration.

Iteration Planning

Iteration Planning is an event during which all team members determine how much of the Team Backlog they can commit to delivering during an upcoming Iteration.

Iteration Retrospective

The Iteration Retrospective is a regular meeting during which Agile Team members discuss the results of the Iteration, review their practices, and identify ways to improve.

Iteration Review

The Iteration Review is a cadence-based event, during which each team inspects the increment at the end of every Iteration to assess progress, and then adjusts its backlog for the next iteration.

Large Solution Level

The Large Solution Level contains the roles, artifacts, and processes needed to build large and complex Solutions.

Large Solution SAFe Configuration

The Large Solution SAFe configuration brings the *Business Solutions and Lean Systems Engineering* competency to those building the largest and most complex Solutions. This configuration supports multiple Agile Release Trains (ARTs) and suppliers.

Lean Budget Guardrails

Lean Budget Guardrails describe budgetary, governance, and spending policies and practices for the Lean budgets allocated to a specific portfolio.

Lean Budgets

Lean Budgets comprise a set of funding and governance practices that increase development throughput by decreasing funding overhead and friction.

Lean Enterprise

The Lean Enterprise is a thriving digital-age business that delivers competitive systems and Solutions to its customers in the shortest sustainable lead time.

Lean Portfolio Management (LPM)

The Lean Portfolio Management competency aligns strategy and execution by applying Lean and systems-thinking approaches to strategy and investment funding, Agile portfolio operations, and governance.

Lean User Experience (Lean UX)

Lean User Experience design is a mindset, culture, and a process that embraces Lean-Agile methods.

Lean-Agile Leadership

The Lean-Agile Leadership competency describes how Lean-Agile leaders drive and sustain organizational change and operational excellence by empowering individuals and teams to reach their highest potential. They do this by learning, exhibiting, teaching, and coaching SAFe's Lean-Agile mindset, values, principles, and practices.

Lean-Agile Mindset

The Lean-Agile Mindset is the combination of beliefs, assumptions, and actions of SAFe leaders and practitioners who embrace the concepts of the Agile Manifesto and Lean thinking.

Lean-Agile Principles

SAFe is based on nine immutable, underlying Lean and Agile Principles. These tenets and economic concepts inspire and inform the roles and practices of SAFe.

Metrics

Metrics are agreed-upon measures used to evaluate how well the organization is progressing toward the Portfolio, Large Solution, program, and team's business and technical objectives.

Milestones

Milestones are used to track progress toward a specific goal or event. There are three types of SAFe milestones: Program Increment (PI), fixed-date, and learning milestones.

Model-Based Systems Engineering (MBSE)

Model-Based Systems Engineering is the practice of developing a set of related system models that help define, design, and document a system under development.

Non-functional Requirements (NFRs)

Non-functional Requirements define system attributes (e.g., reliability, performance) that serve as constraints or restrictions on the design of the system across the different backlogs.

Portfolio Backlog

The Portfolio Backlog is the highest-level backlog in SAFe, providing a holding area for upcoming business and enabler Epics.

Portfolio Canvas

The Portfolio Canvas is a type of Business Model Canvas that has been adapted to charter and describe the structure and purpose of a SAFe portfolio.

Portfolio Kanban

The Portfolio Kanban is a mechanism used to visualize, manage, and analyze the prioritization and flow of portfolio Epics from ideation to implementation and completion.

Portfolio Level

The Portfolio Level contains the principles, practices, and roles needed to initiate and govern a set of development Value Streams.

Portfolio SAFe Configuration

The Portfolio SAFe configuration applies the *Lean Portfolio Management* competency to align portfolio execution to the enterprise strategy, and organizes development around the flow of value through one or more value streams.

Pre- and Post-PI Planning

Pre- and Post-Program Increment (PI) Planning events are used to prepare for, and follow up after, PI Planning for Agile Release Trains (ARTs) and Suppliers in a Solution Train.

Product Management

Product Management is responsible for identifying Customer needs, prioritizing Features, guiding the work through the Program Kanban, and developing the program Vision and Roadmap.

Product Owner (PO)

The Product Owner is a member of the Agile team who is responsible for defining user Stories and prioritizing the Team Backlog to streamline program execution.

Program Backlog

The Program Backlog is the holding area for upcoming Features and Enablers, which are intended to address user needs and deliver business benefits and build its Architectural Runway.

Program Increment (PI)

A Program Increment is a timebox during which an Agile Release Train (ART) delivers incremental value in the form of working, tested software and systems. PIs are typically 8–12 weeks long. The most common pattern for a PI is four development Iterations, followed by one Innovation and Planning (IP) Iteration.

Program Increment (PI) Objectives

Program Increment Objectives are a summary of the business and technical goals that an Agile Team or train intends to achieve in the upcoming Program Increment (PI).

Program Increment (PI) Planning

Program Increment Planning is a cadence-based, face-to-face event that serves as the heartbeat of the Agile Release Train (ART), aligning all the teams on the ART to a shared mission and Vision.

Program Kanban

The Program and Solution Kanban systems are a method to visualize and manage the flow of Features and Capabilities from ideation to analysis, implementation, and release through the Continuous Delivery Pipeline.

Program Level

The Program Level contains the roles and activities needed to continuously deliver Solutions via an Agile Release Train (ART).

Release Train Engineer (RTE)

The Release Train Engineer is a servant leader and coach for the Agile Release Train (ART). The RTE's major responsibilities are to facilitate the ART events and processes and assist the teams in delivering value.

Release on Demand

Release on Demand is the process by which new functionality is deployed into production and released immediately or incrementally to Customers based on demand.

Roadmap

The Roadmap is a schedule of events and Milestones that communicate planned Solution deliverables over a planning horizon.

SAFe Implementation Roadmap

The SAFe Implementation Roadmap describes a strategy and an ordered set of activities that have proved effective in successfully implementing SAFe.

SAFe Program Consultants (SPCs)

Certified SAFe® Program Consultants are change agents who play a critical role in successfully implementing SAFe.

SAFe for Government

SAFe for Government is a set of success patterns that help public-sector organizations implement Lean-Agile practices in a government context.

SAFe for Lean Enterprises

SAFe® for Lean Enterprises is a knowledge base of proven, integrated principles, practices, and competencies for Lean, Agile, and DevOps.

Scrum Master

The Scrum Master is a servant leader and coach who helps an Agile team follow the Agile process, removes impediments, and fosters an environment for high-performing teams, continuous flow, and relentless improvement.

ScrumXP

ScrumXP is a lightweight process to deliver value for cross-functional, self-organized teams within SAFe. It combines the power of Scrum project management practices with Extreme Programming (XP) practices.

Set-Based Design (SBD)

Set-Based Design is a practice that keeps requirements and design options flexible for as long as possible during the development process.

Shared Services

Shared Services represents the specialty roles, people, and services required for the success of an Agile Release Train (ART) or Solution Train, but that cannot be dedicated full-time.

Solution

Each Value Stream produces one or more Solutions, which are products, services, or systems delivered to the Customer, whether internal or external to the Enterprise.

Solution Architect/Engineer

The Solution Architect/Engineering role is filled by an individual or small team that defines a shared technical and architectural vision for the Solution under development.

Solution Backlog

The Solution Backlog is the holding area for upcoming Capabilities and Enablers, each of which can span multiple ARTs and is intended to advance the Solution and build its architectural runway.

Solution Context

The Solution Context identifies the operational environment for a Solution, providing an understanding of requirements, usage, installation, operation, and support of the Solution.

Solution Demo

The Solution Demo is where the results of development efforts from the Solution Train are integrated, evaluated, and made visible to Customers and other stakeholders.

Solution Intent

The Solution Intent is the repository for storing, managing, and communicating the knowledge of current and intended Solution behavior.

Solution Management

Solution Management is responsible for identifying Customer needs, prioritizing Capabilities, guiding the work through the Solution Kanban, and developing the Solution Vision and Roadmap.

Solution Train

The Solution Train is the organizational construct used to build large and complex Solutions that require the coordination of multiple Agile Release Trains (ARTs), as well as the contributions of Suppliers. It aligns ARTs with a shared business and technology mission using the Solution Vision, Backlog, and Roadmap, and an aligned Program Increment (PI).

Spanning Palette

The Spanning Palette contains various roles and artifacts that may apply to a specific team, program, Large Solution, or Portfolio context.

Stories

Stories are short descriptions of a small piece of desired functionality, written in the user's language. They are sized to be implemented in small, vertical slices within a single Iteration.

Strategic Themes

Strategic Themes are differentiating business objectives that connect a portfolio to the strategy of the Enterprise, which influence the portfolio strategy and provide business context for decision-making.

Supplier

A Supplier is an internal or external organization that develops and delivers components, subsystems, or services that help Solution Trains and Agile Release Trains provide Solutions to their Customers.

System Demo

The System Demo is an event where the work from all teams on the Agile Release Train is integrated, evaluated, and made visible to Customers and other stakeholders.

System Team

The System Team is a specialized Agile team that assists in building and supporting the Agile development environment, typically including development and maintenance of the toolchain that supports the Continuous Delivery Pipeline.

Team and Technical Agility

The Team and Technical Agility competency describes the critical skills and Lean-Agile principles and practices that are needed to create high-performing Agile teams who create high-quality, well-designed technical solutions.

Team Backlog

The Team Backlog contains user and enabler Stories that originate from the Program Backlog, as well as Stories that arise locally from the team's local context.

Team Kanban

Team Kanban is a method that helps teams facilitate the flow of value by visualizing workflow, establishing Work In Process (WIP) limits, measuring throughput, and continuously improving their process.

Team Level

The Team Level contains the roles, activities, events, and processes that Agile Teams build and deliver value in the context of the Agile Release Train (ART).

Value Stream Coordination

Value Stream Coordination provides guidance to manage dependencies and exploit the opportunities in a portfolio.

Value Streams

Value Streams represent the series of steps that an organization uses to build Solutions that provide a continuous flow of value to a customer.

Vision

The Vision is a description of the future state of the Solution under development. It reflects Customer and stakeholder needs, as well as the Features and Capabilities proposed to meet those needs.

Weighted Shortest Job First (WSJF)

Weighted Shortest Job First (WSJF) is a prioritization model used to sequence jobs (e.g., Features, Capabilities, and Epics) to help realize the maximum economic benefit.