

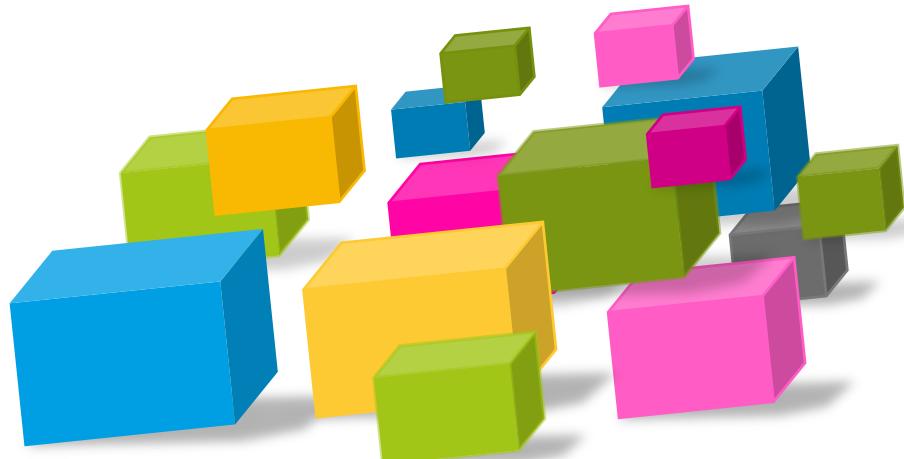
Scaling Agile – The SAFe Framework

“Everybody, all together, from early on”

Kim Horn

Version 1.0

28 Nov 2022





There are many ‘Ways of Working’ today.

“Everyone of us knows how to develop our own software, but as a software community we have no widely accepted common ground.”

Ivar Jacobson



SAFe - A Balanced Common Ground

- Use the best of Agile and Lean
 - Doing Vs Thinking.
- Provides an Agile ‘top down’ formality and structure to align an organisation.
 - With decentralised decision making, autonomous teams.
- **Goals:**
 - Improve productivity, have better motivated people, less defects, and faster time to market.
 - Make our business as a whole more Agile and Lean.

What is SAFe ?



“SAFe® 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 in the shortest sustainable lead time. It’s a system that guides the roles, responsibilities, and activities necessary to achieve a sustained, competitive technological advantage.

Combining the power of Agile with Lean product development, DevOps, and systems thinking, SAFe synchronizes alignment, collaboration, and delivery for multiple Agile teams. SAFe dramatically improves business agility by accelerating productivity, time-to-market, quality, employee engagement, and more.”

<https://www.scaledagileframework.com>

<https://www.scaledagile.com>

Why Use SAFe – 11 Reasons ?



SAFe was originally developed to provide a big picture view of how work flowed from product management (or other stakeholders), through governance, program, and development teams, out to customers.

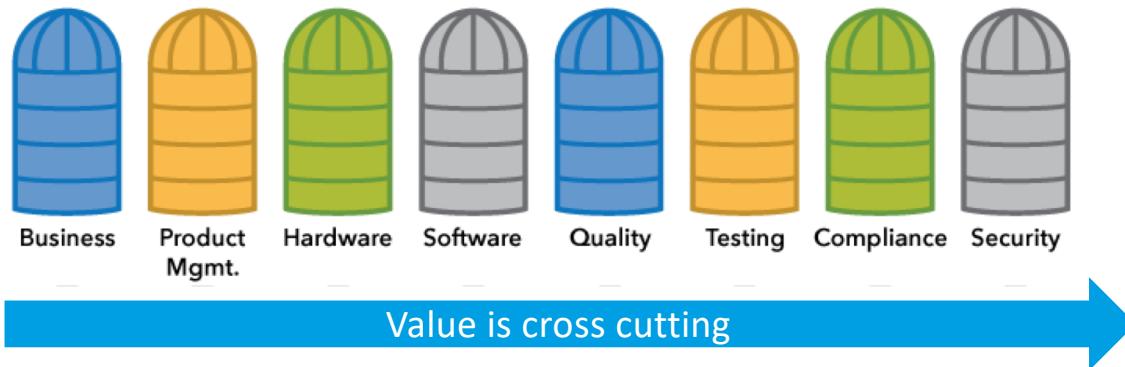
1. It is publicly available and free to use.
2. It is available in a highly approachable, usable form: <https://www.scaledagileframework.com>
3. It is lightweight, in the best sense of the term.
4. **It is practical but based on well defined core values, principles and practices**
5. It is specific.
6. It conveniently codifies some of the most common agile practices in use today.
7. **It offers useful extensions to common agile practices.**
8. **It grounds agile practices in an enterprise context.**
9. **It offers a complete picture of software development.**
10. It is regularly maintained.
11. It is supported by 250+ Scaled Agile Partners for training, consulting, implementation services.



Core Concept - Agile Release Train (ART)

SAFe outlines a consistent approach to planning, execution and delivery of value, called an ART.

- It is a lightweight “program container”, a long lived virtualised team.
- It brings multiple **Agile Teams** together on a consistent cadence every 8-12 weeks in a **Program Increment (PI)**.
- ARTs are cross-functional and have all the people they need to deliver value. Breaks down silos.

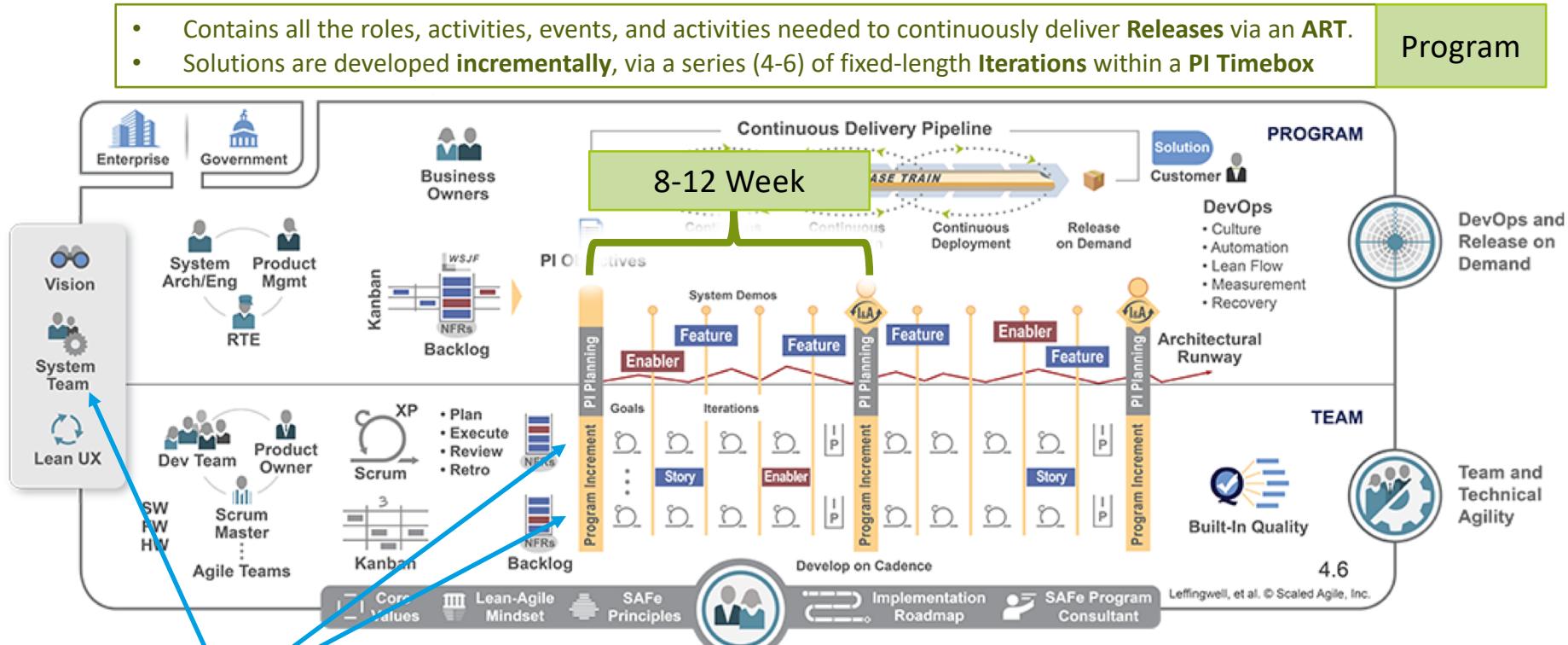


- **Lean Upfront Planning:** At the start of each PI, the ART comes together to plan what they will deliver in that PI.
- This opportunity to work together as a team of teams helps organizations uncover, plan for, and address cross-team dependencies, risks, and impediments.



Essential Configuration – 2 Levels

- Contains all the roles, activities, events, and activities needed to continuously deliver **Releases** via an ART.
- Solutions are developed **incrementally**, via a series (4-6) of fixed-length **Iterations** within a PI Timebox

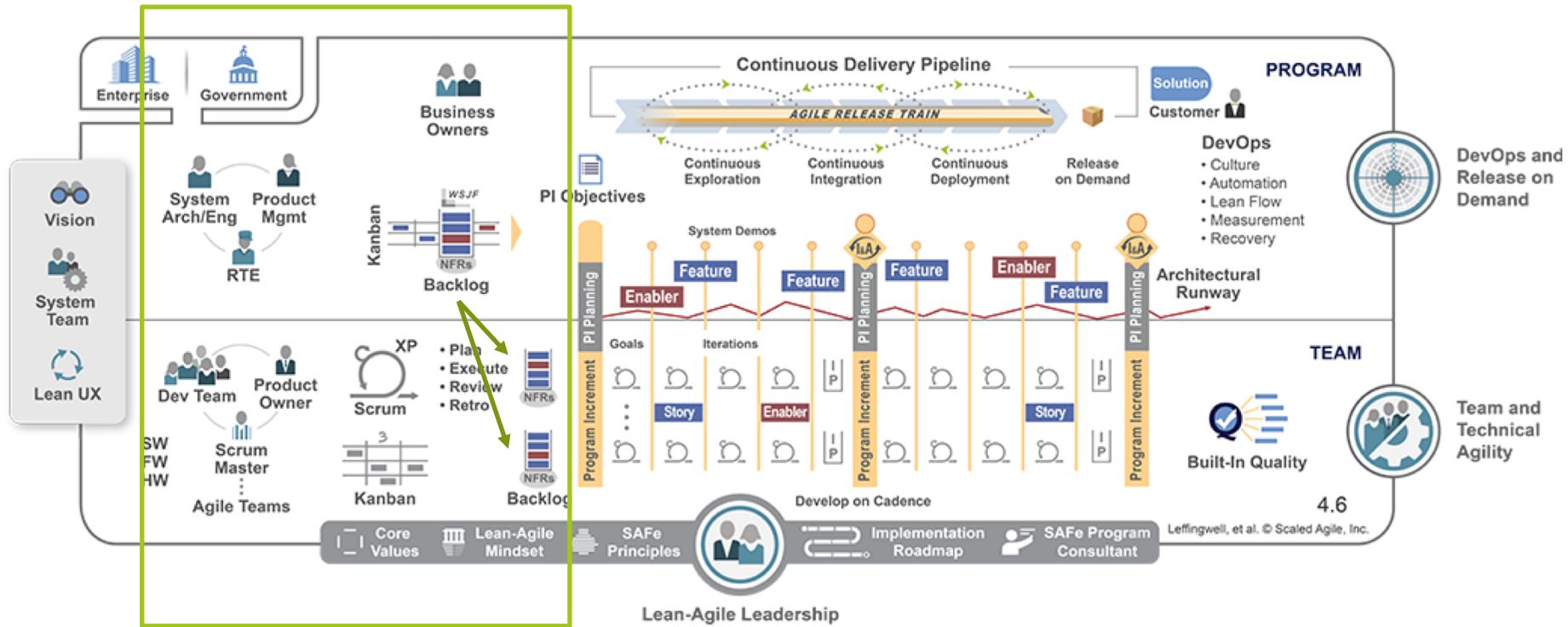


Teams

- Contains all the roles, activities, events, and processes to build and deliver value.
- Define, build, and test **Stories** from their Team Backlog Iteratively (2 weeks);



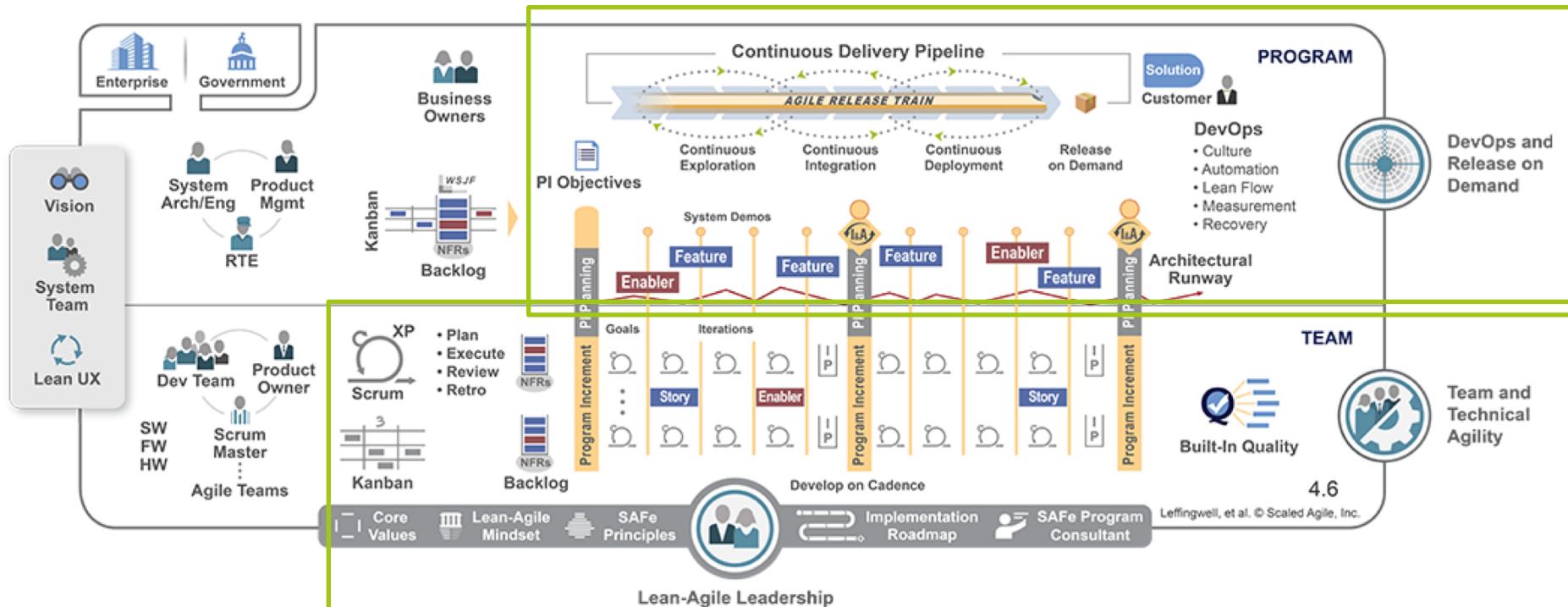
Backlog, Kanban and Roles





Competencies and Processes

Continuous Delivery Pipeline – the workflows, activities, and automation needed to provide a constant release of value to the end user. Small batches with **Release on Demand. DevOps**

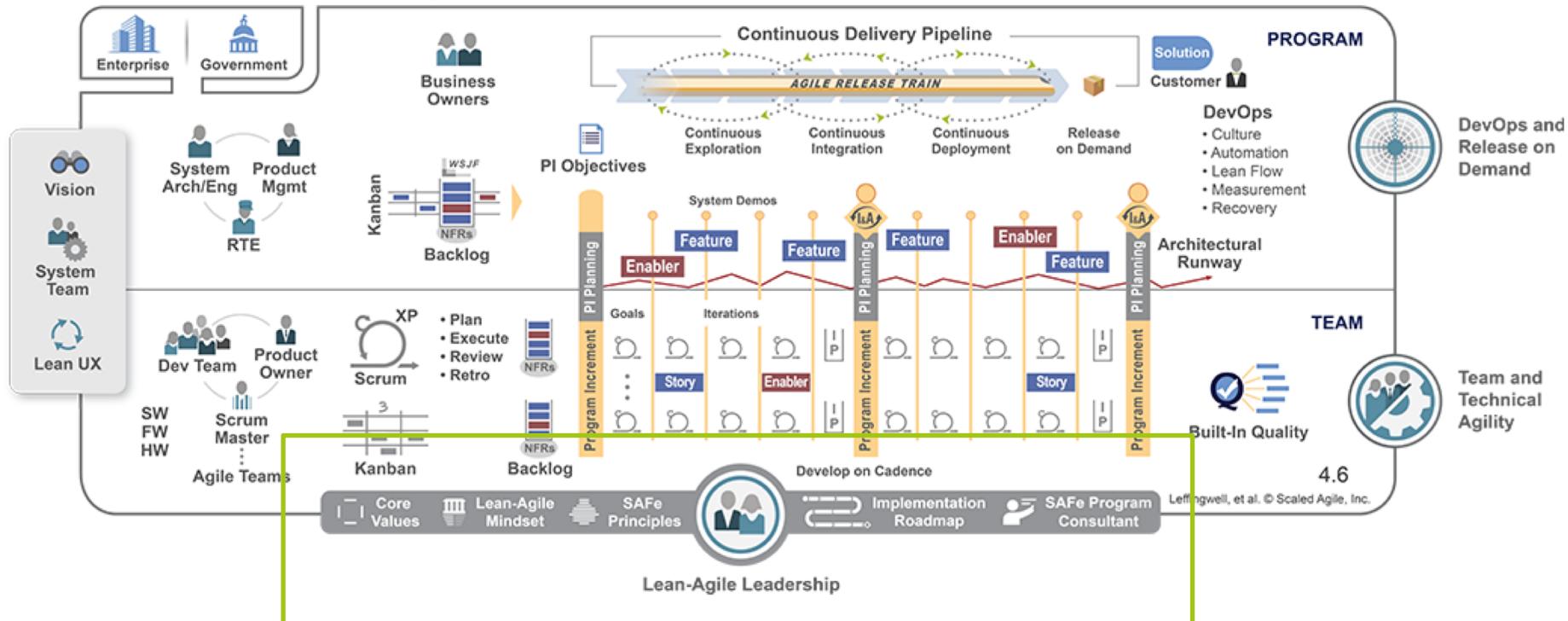


Multiple Agile Teams apply XP, Scrum or Kanban



Lean-Agile Leadership Competency

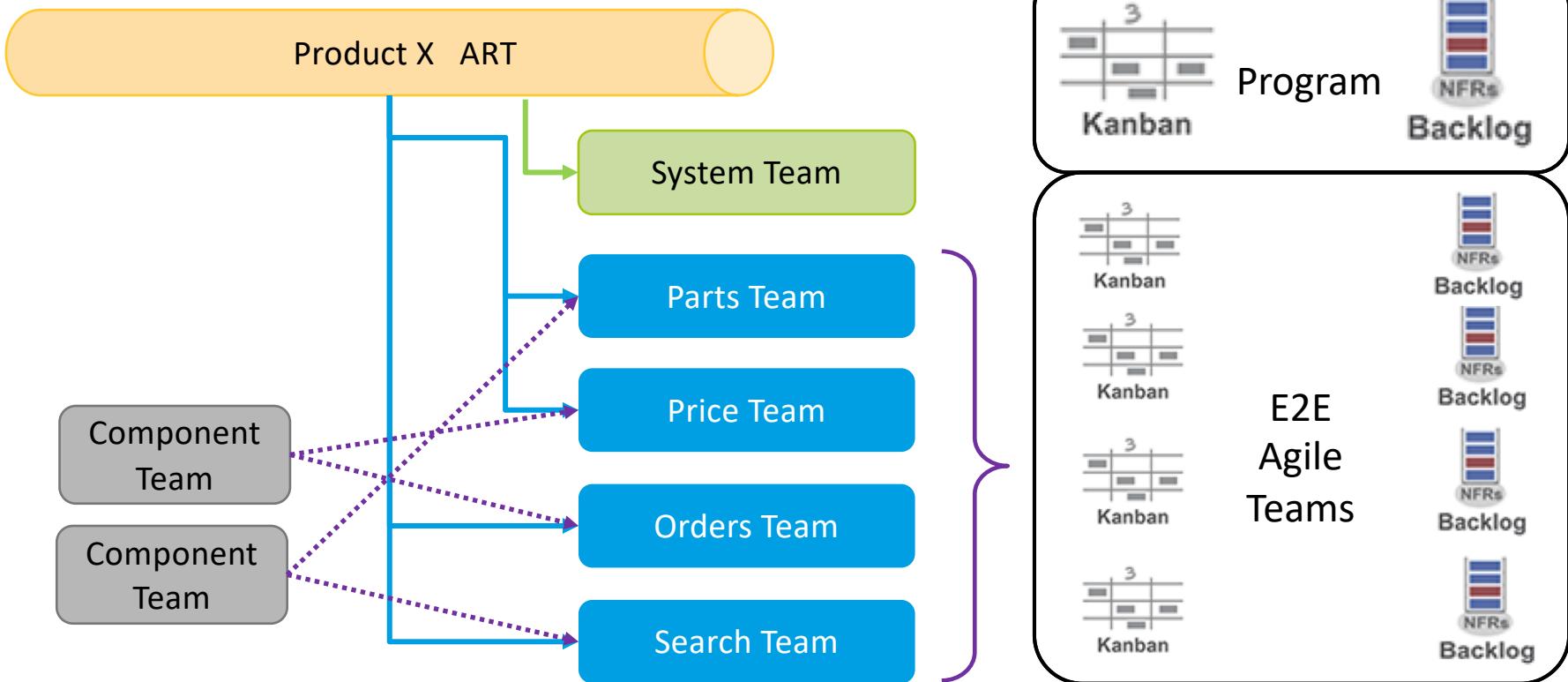
“sufficiently powerful guiding coalition” of stakeholders is needed.



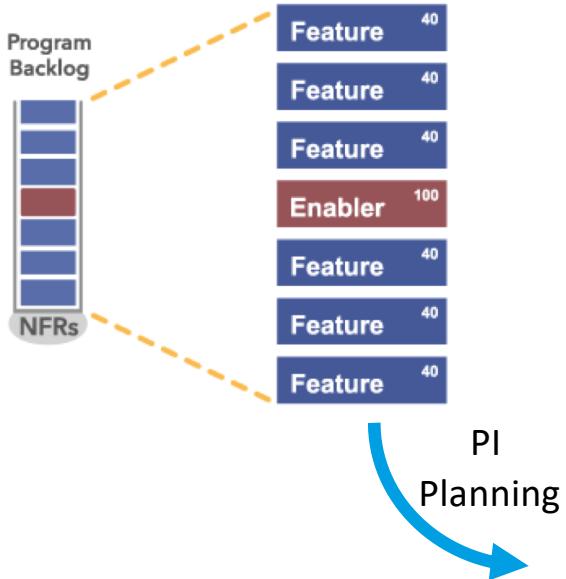


Example – We Need Capability Aligned Teams

Potential Small Teams based around microservice capabilities, e.g.



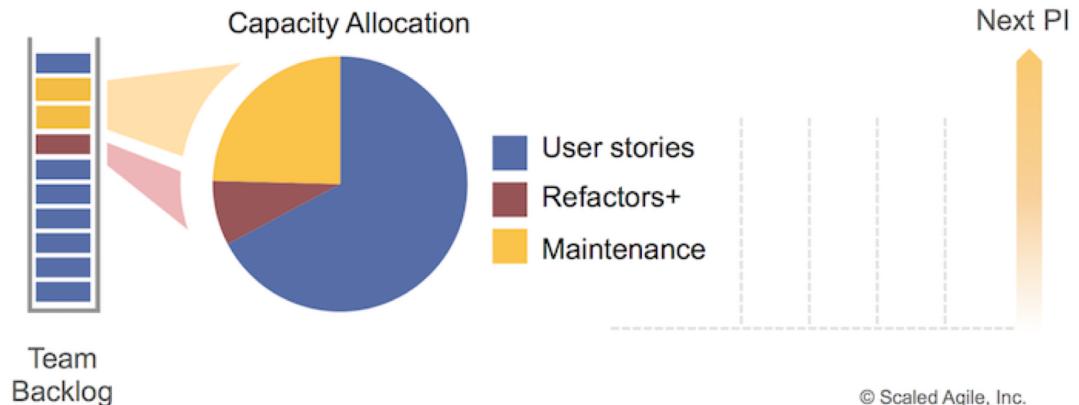
2 Backlogs: Program to Team – A Consistent Planning Mechanism



Capacity Allocation determines the balance of work on which backlog items. For example how much Enabler vs Feature work

The 2 Backlogs (Program and Team) include:

- Enablers
- Program Epics & Features
- User Stories
- Refactoring Items
- Maintenance Items
- NFRs - Non-functional requirements constrain the Backlog, and occur at all levels. They need to be eventually tested.



© Scaled Agile, Inc.

© 2015 Kim Horn



Architecture Links Enablers and Features

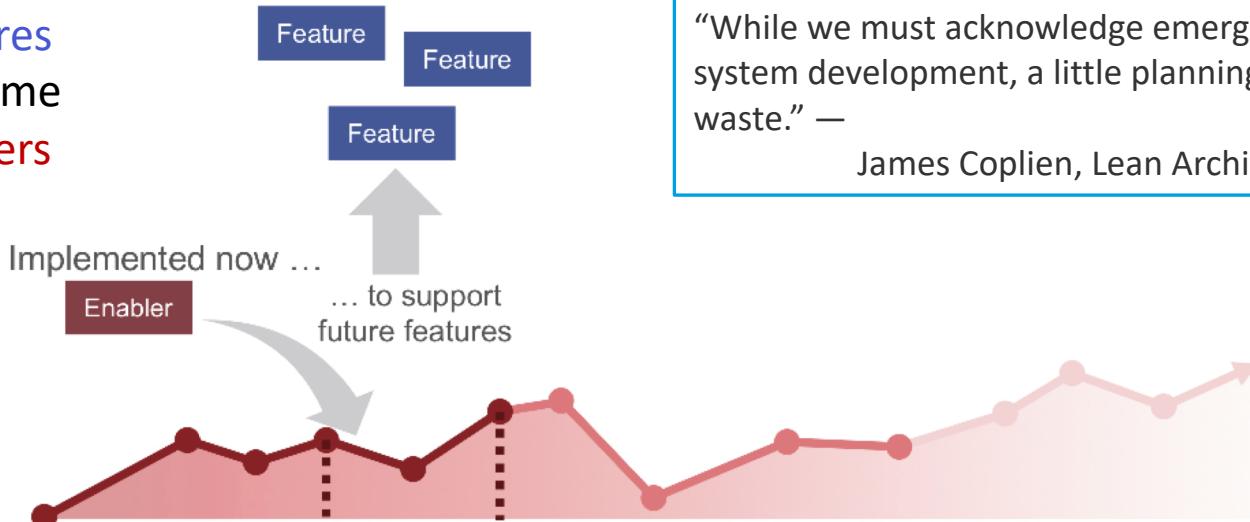
Architecture Runway provides ‘just enough’ technical enablement to keep velocity high and avoid excessive redesign and delays. Bigger systems often require more runway. Continuously maintained.

- Intentional Up Front Architecture merges Emergent Design – Architecture Evolves.

Agile Architecture builds out the runway; it 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.

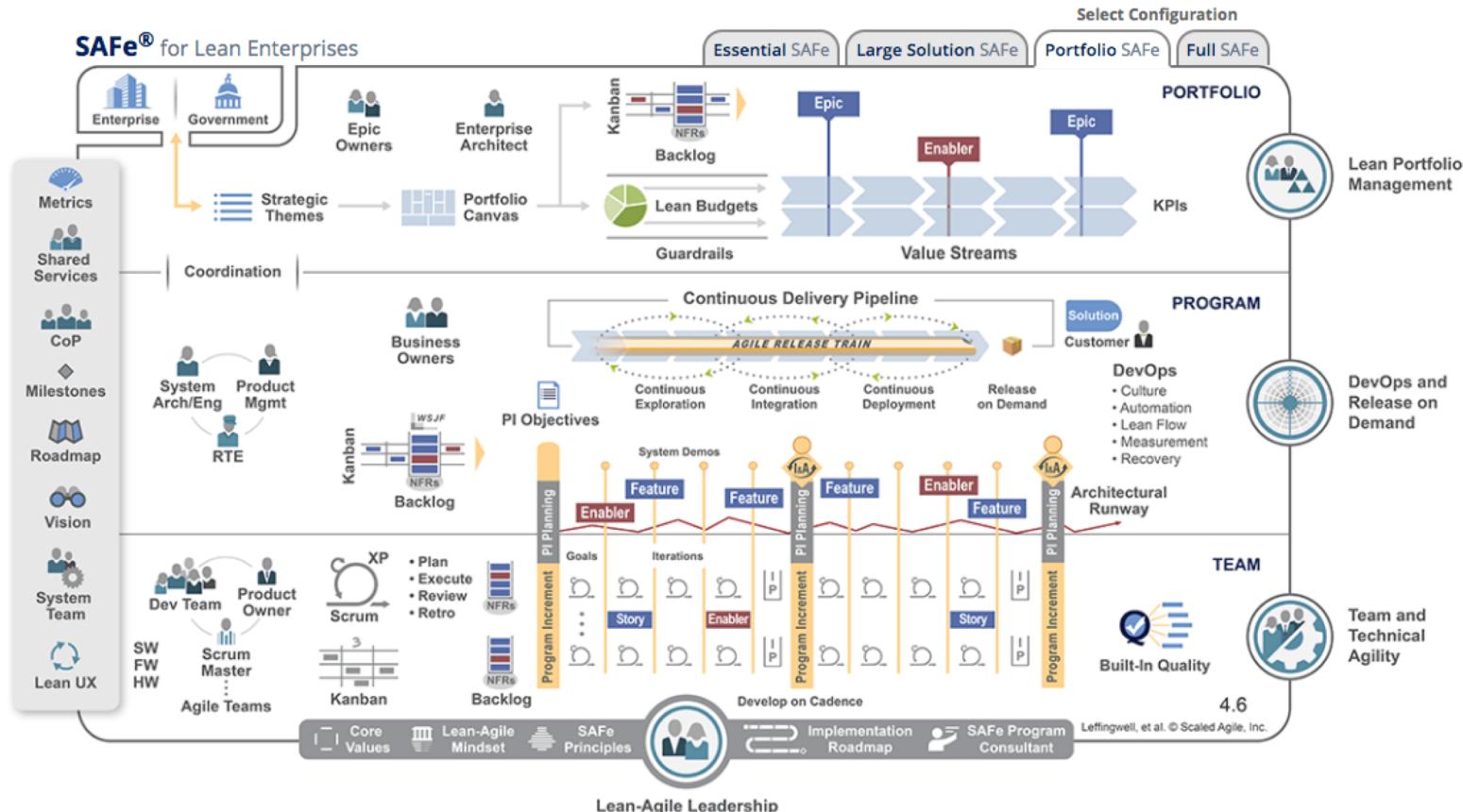
- Architecture works as intermediaries for: Governance, Interfaces, Dependencies

Features
consume
Enablers





Agile Portfolio across Value Streams aligns ARTs.

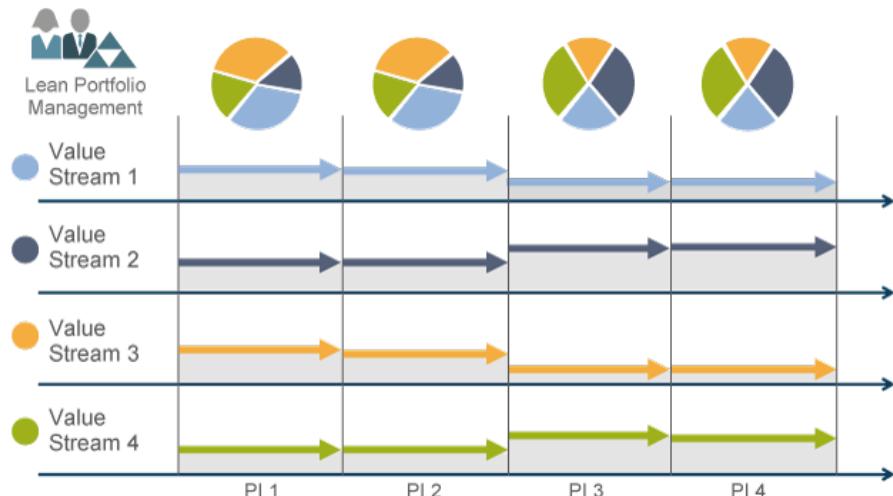




Lean Portfolio Management (LPM) Competency

To address the challenge of defining, communicating, and aligning strategy, the LPM function has the highest level of decision-making and financial accountability for the value streams and solutions in a portfolio.

Three essential collaborations needed to realize the Lean Portfolio Management competency:

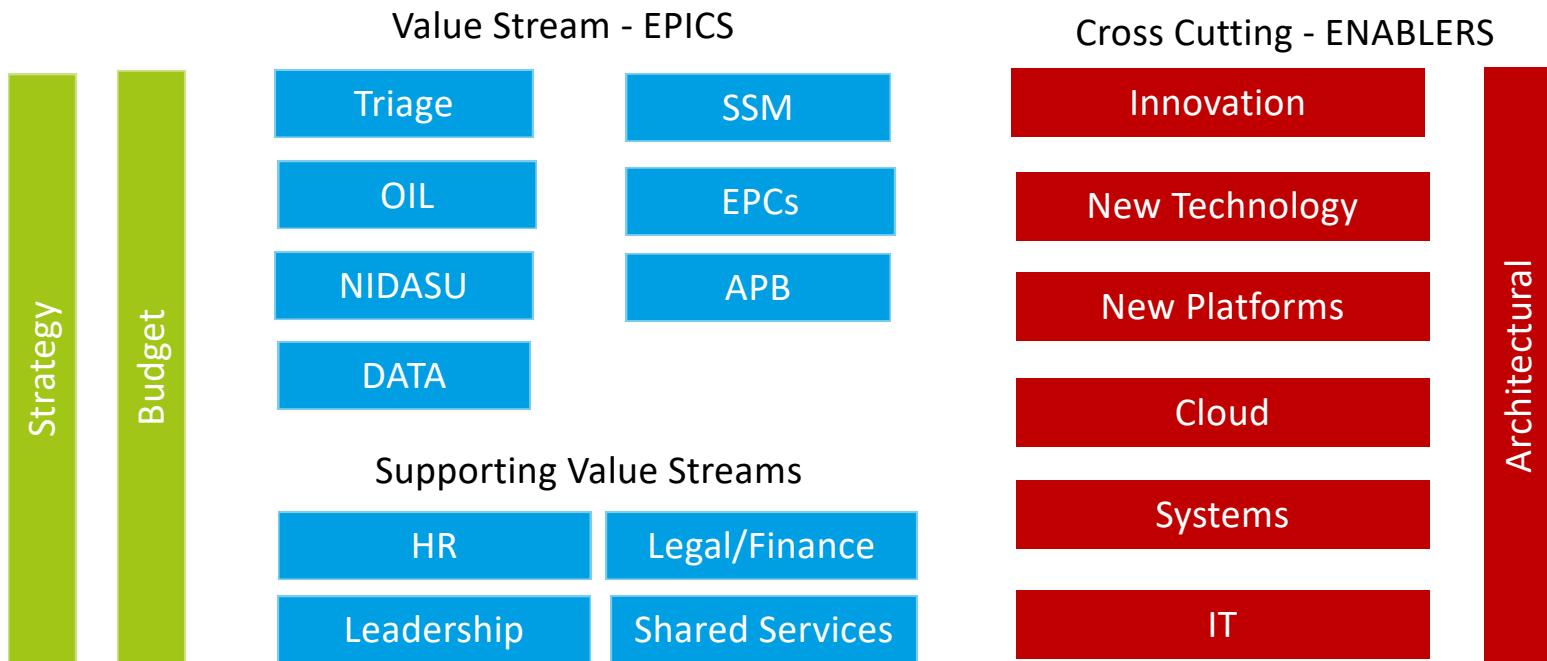


© Scaled Agile, Inc.

Infomedia Portfolio Example

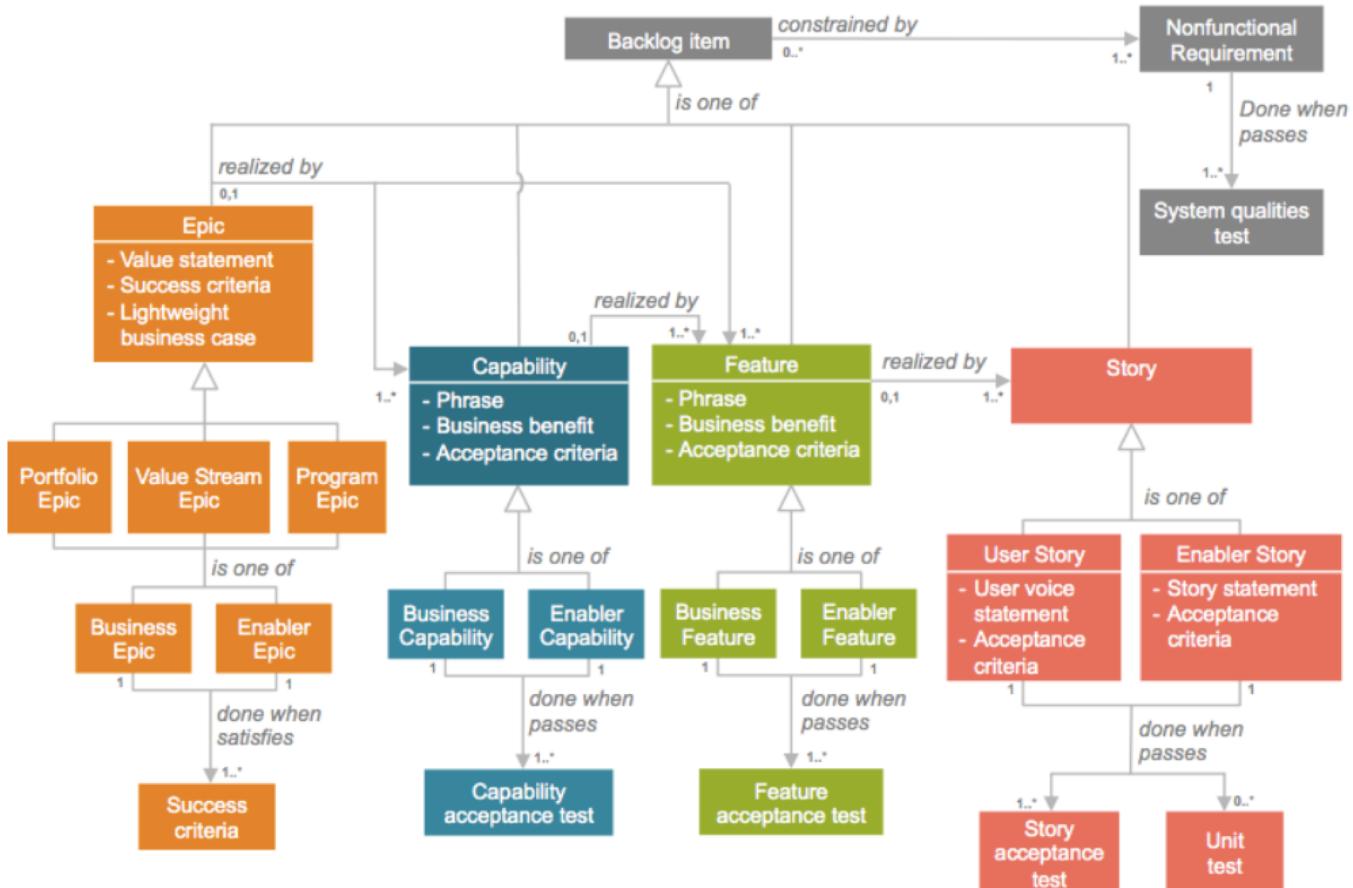


The portfolio canvas describes how a portfolio of solutions creates, delivers and captures value for an organization. It also helps define and align the portfolio's Value Streams and Solutions to the goals of the enterprise, the strategy and vision, in an agile way.





Scalable Backlog and Requirements Model



Unites: Non-Functional, Traditional and Agile Requirements, at various levels of granularity and organisation levels, into a scalable backlog model for execution and planning.



The Goal: Value

Shortest sustainable lead time. Best quality and value to people and society.
High morale, safety, customer delight.

Respect for people and culture

- ▶ People do all the work
- ▶ Your customer is whoever consumes your work
 - Don't overload them
 - Don't make them wait
 - Don't force them to do wasteful work
 - Don't impose wishful thinking
- ▶ Build long-term partnerships based on trust
- ▶ Culture change comes last, not first
- ▶ To change culture, change the organization

Flow

- ▶ Optimize continuous and sustainable throughput of value
- ▶ Build in quality; flow depends on it
- ▶ Understand, exploit, and manage variability
- ▶ Avoid start-stop-start project delays
- ▶ Use informed-decision making via fast feedback

Innovation

- ▶ Producers innovate; customers validate
- ▶ Get out of the office
- ▶ Provide time and space for creativity
- ▶ Apply innovation accounting
- ▶ Pivot without mercy or guilt

Relentless improvement

- ▶ A constant sense of danger
- ▶ Optimize the whole
- ▶ Consider facts carefully, then act quickly
- ▶ Apply Lean tools to identify and address root causes
- ▶ Reflect at key milestones; identify and address shortcomings

Foundation: Lean-Agile Leadership

Management applies and teaches Lean thinking, bases decisions on this long-term philosophy.
Principles of Lean-Agile Leadership.

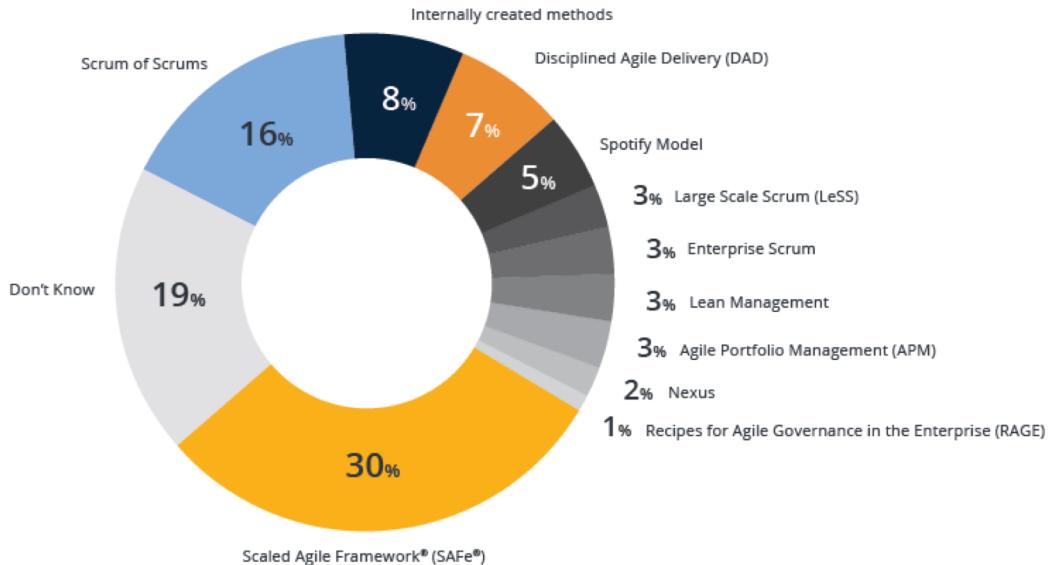
© Scaled Agile, Inc.



Key Themes:

- Reduced project costs are a primary driver for Agile adoption:
 - 71% reduction in costs
- DevOps is a higher priority than 2018:
 - Rated as Important by 90%
 - 73% have initiative in play
- End-to-end traceability is key to improving DevOps practices:
 - 38% said its most valuable capability
- Value Stream Management is new, but important:
 - Rated as Important by 67%
- SAFe dominates scaling methods:
 - Continues to grow in adoption
 - Chosen by 30% of respondents

"More than 70 percent of Fortune 500 businesses having certified SAFe practitioners on-site."



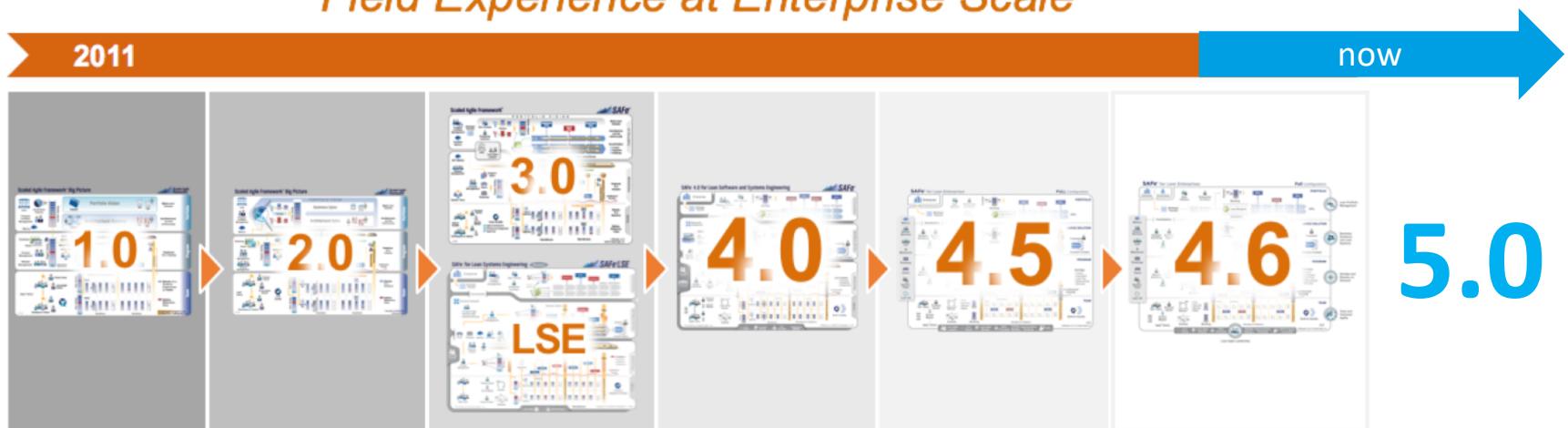
SAFe is Continually Improving



“SAFe dominates scaling methods— The Scaled Agile Framework® (SAFe®) continues to grow in adoption as the scaling method of choice and was identified as the scaling method of choice by 30% of respondents.”

May 7 2019, ‘State of Agile Report’

Field Experience at Enterprise Scale

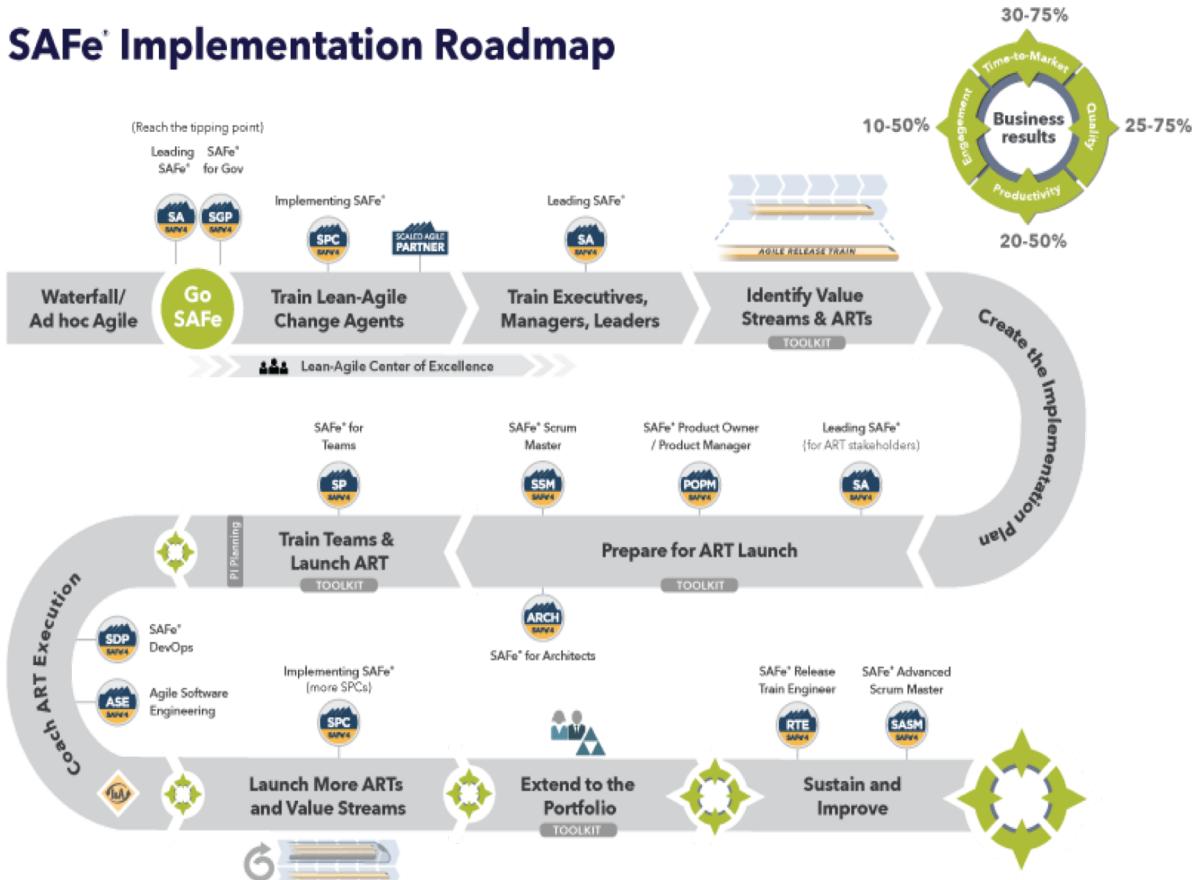


Lean Product Development | Agile Development | DevOps | Systems Thinking

Implementation Roadmap



SAFe® Implementation Roadmap





The End



Some questions about your/our current process ?

1. Is it understood by all, is it approachable, well documented ?
2. Is it complete and consistent ?
 - Does it integrate: development, dev-ops, architecture, program, portfolios, HR, QA, strategy ?
 - Does it include all the roles required; do people really know what they have to do, when ?
3. Does it scale from small to large, and upward:
 - At all levels of the organization, and across team structures ?
 - Is quality managed at all levels ?
4. Is it configurable; to match different goals ?
5. Do we understand how Governance happens, does it work, is autonomy working ?
6. Do we produce useful requirements, are they well understood, do they at work at multiple levels ?
7. Do we utilise the best from Agile and Lean ?
8. Are all the dependencies understood to deliver on time, without delays?
9. Do we have the capability to design and manage the process, the collateral and change:
 - is it our core business, do we do this well ?
- 10. Maturity - Do we want to try and improve productivity further, have better motivated people, less defects, and faster time to market ? Do we have the right measures and process in place to do this improvement ?**
- 11. Do we want our business as a whole to be more 'agile', at all levels ?**



There are nine underlying Lean and Agile principles that drive the roles and practices in SAFe:

- Take an economic view
- Apply systems thinking
- Assume variability; preserve options
- Build incrementally with fast, integrated learning cycles
- Base milestones on objective evaluation of working systems
- Visualize and limit work-in-progress, reduce batch sizes, and manage queue lengths
- Apply cadence (timing), synchronize with cross-domain planning
- Unlock the intrinsic motivation of knowledge workers
- Decentralize decision-making



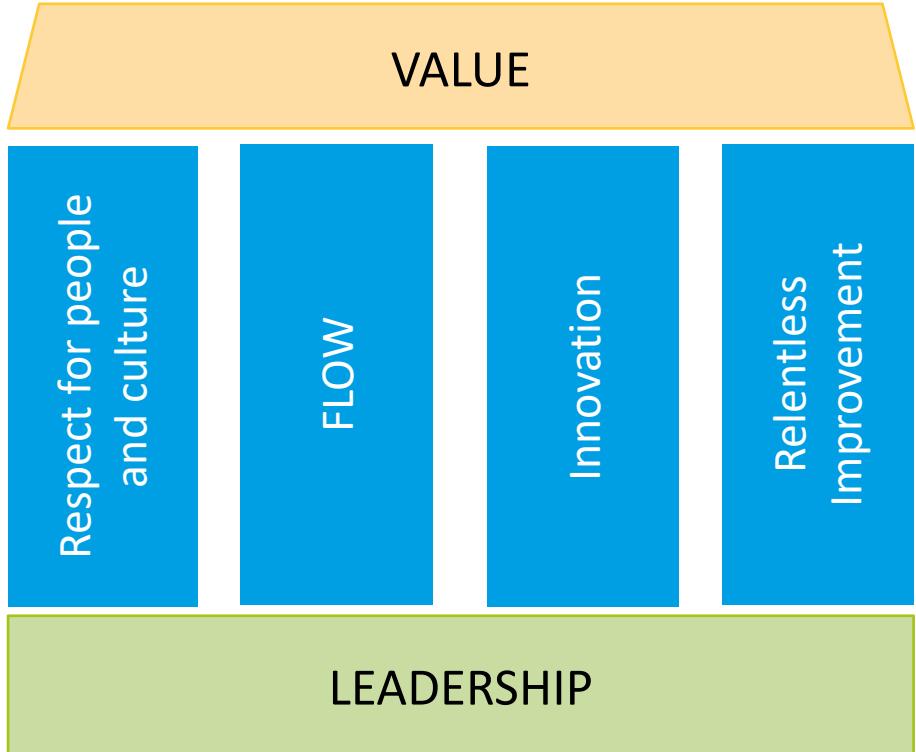
A prime goal of SAFe is to reduce the common Lean Wastes, and Improve Productivity. These wastes were defined by Toyota in Lean Manufacturing and now are applied to Agile Software Product Development:

| Lean Manufacturing | Agile Development |
|--------------------|---------------------|
| Inventory | Partially done work |
| Processing | Extra processing |
| Overproduction | Overproduction |
| Transportation | Task switching |
| Waiting | Waiting / delays |
| Motion | Hand-offs |
| Defects | Defects (rework) |



Other Scaled Agile Frameworks

- Scrum of Scrums
- LeSS: Large Scale Scrum
- Nexus (Scrum.org's scaled agile)
- Scrum at Scale (Jeff Sutherland's version)
- DAD: Disciplined Agile Delivery (IBM's version)



Purpose:

Achieve the sustainably shortest lead time with:

- Best quality and value to people and society
- High morale, safety, customer delight.
- Goal of LEAN is to reduce waste (muda)



The Values of the Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

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

agilemanifesto.org

Agile Manifesto 12 Principles



1. **Customer satisfaction through early and continuous software delivery** – Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.
2. **Accommodate changing requirements throughout the development process** – The ability to avoid delays when a requirement or feature request changes.
3. **Frequent delivery of working software** – Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.
4. **Collaboration between the business stakeholders and developers throughout the project** – Better decisions are made when the business and technical team are aligned.
5. **Support, trust, and motivate the people involved** – Motivated teams are more likely to deliver their best work than unhappy teams.
6. **Enable face-to-face interactions** – Communication is more successful when development teams are co-located.
7. **Working software is the primary measure of progress** – Delivering functional software to the customer is the ultimate factor that measures progress.
8. **Agile processes to support a consistent development pace** – Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.
9. **Attention to technical detail and design enhances agility** – The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.
10. **Simplicity** – Develop just enough to get the job done for right now.
11. **Self-organizing teams encourage great architectures, requirements, and designs** – Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.
12. **Regular reflections on how to become more effective** – Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

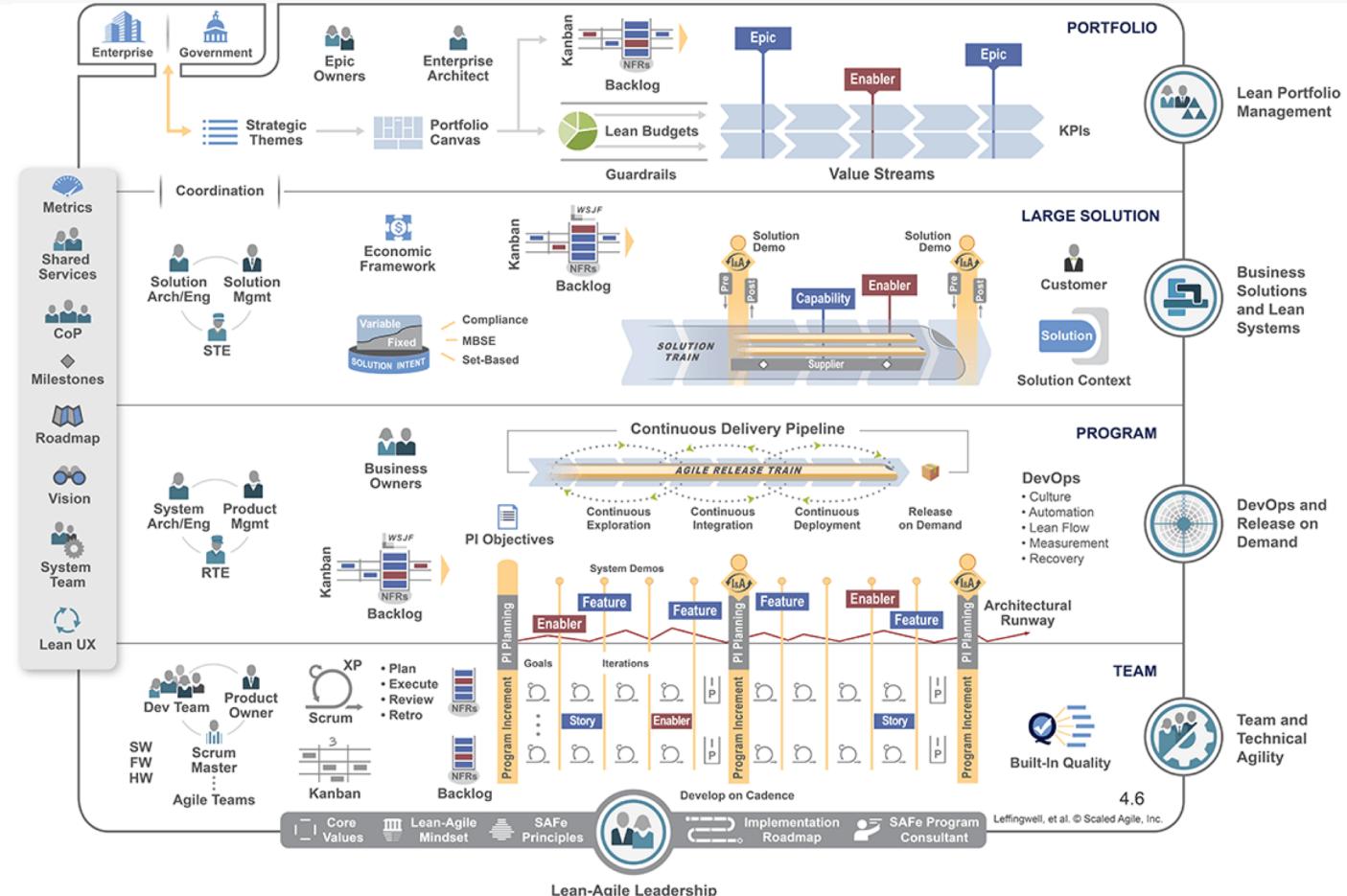
4 Types of SAFe Configurations - Introduce 5 Competencies



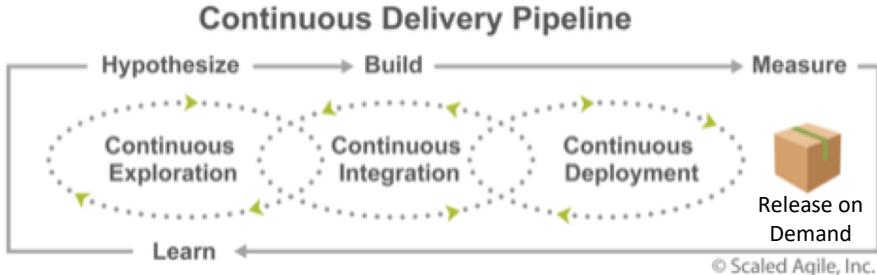
1. **Essential SAFe** - the most basic configuration. It provides a starting point for implementing SAFe and describes the most critical elements needed to realise the majority of the framework's benefits. Provides the **Team and Technical Agility, DevOps and Release on Demand, and Lean-Agile Leadership** competencies.
2. **Portfolio SAFe** - provides the **Lean Portfolio Management competency** which aligns portfolio execution to enterprise strategy. It organizes development around the flow of value through one or more value streams
3. **Large Solution SAFe** - introduces the **Business Solutions and Lean Systems Engineering competency**, which supports those building the largest and most complex solutions that require multiple Agile Release Trains and Suppliers, but do not require portfolio-level considerations.
4. **Full SAFe** - includes all Five Core Competencies of the Lean Enterprise. It is the most comprehensive version of the Framework and supports enterprises that build and maintain a portfolio of large and complex solutions.



The Full Model



The Two Levels



Program:

- Contains the roles and activities needed to continuously deliver **Solutions** and **Releases** via an **ART**.
- Solutions are developed and delivered **incrementally**, via a series (4-6) of fixed-length **Iterations** within a **PI timebox**.
- ARTs are guided by architecture, and product management (Internal VOC).
- **Continuous Delivery Pipeline** – the workflows, activities, and automation needed to provide a constant release of value to the end user. Small batches with **Release on Demand**.
- **DevOps** – A mindset, culture, and a set of technical practices. It provides communication, integration, automation, and close cooperation among all the people needed to plan, develop, test, deploy, release, and maintain a solution.
- The ART aligns Agile Teams to a common business and technology mission with agreed **PI objectives**.

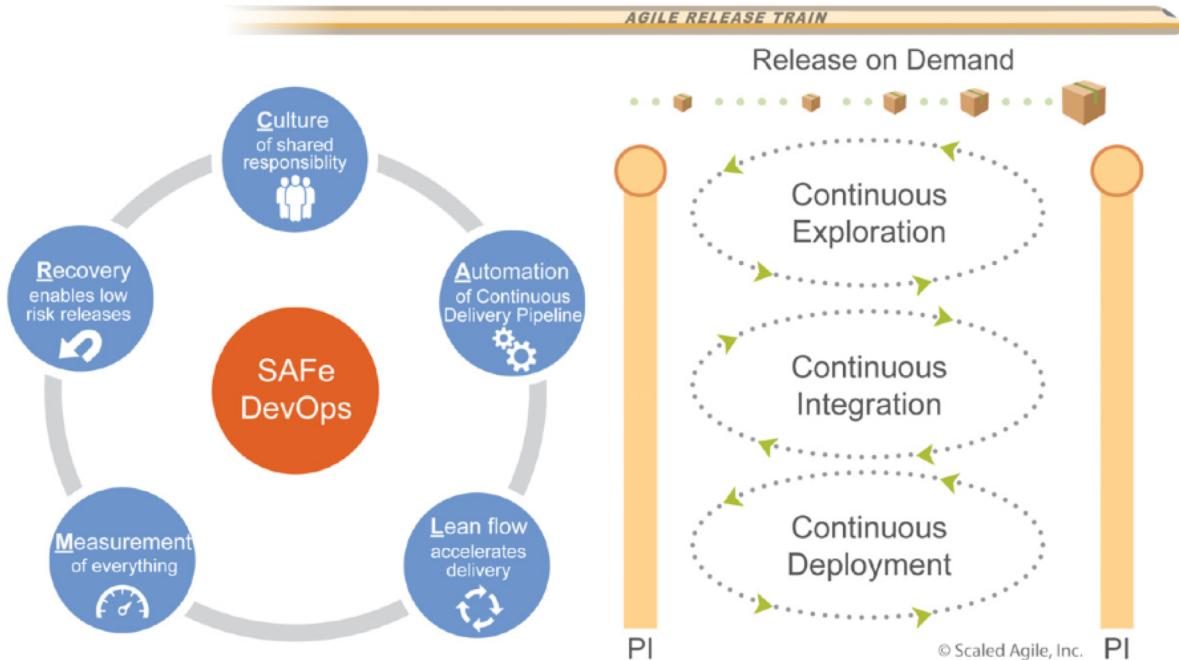
Team:

- Contains all the roles, activities, events, and processes to build and deliver value, as **Team Increments**;
- Responsible for defining, building, and testing Stories from their Team Backlog Iteratively;
- May be multiple Agile Teams within a program that apply Scrum or XP or Kanban.



ARTs apply DevOps (CI/CD); Releasability

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



The System Team is a specialised Agile Team that assists in building and using the Agile development environment, including Continuous Integration, test automation, and Continuous Deployment.

The System Team supports the integration of assets from Agile teams, performs end-to-end Solution testing where necessary, and assists with deployment and release.

Architectural Runway – Implements Architectural Strategy



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

- It provides the necessary technical foundation for developing business initiatives and implementing new Features and/or Capabilities.
- The development of new features and capabilities consumes the architectural runway, continual investment must be made to extend it by implementing **Enablers**.
- **Enablers** support the activities needed to extend the Architectural Runway to provide future business functionality. These include exploration, infrastructure, compliance, and architecture development. Enablers go on the backlogs.
- Some enablers fix existing problems with the Solution, such as improving the performance or User Experience

Without the Architecture Runway, **emergent design** fails to handle the complexity of large-scale system development, and the following problems start to occur:

- Excessive redesign and delays reduce velocity Systems become difficult to integrate, validate, and maintain;
- Decline of system qualities, known as Nonfunctional Requirements (NFRs);
- Reduced collaboration and synchronization among teams;
- Low reuse of common components and redundancy of solution elements.

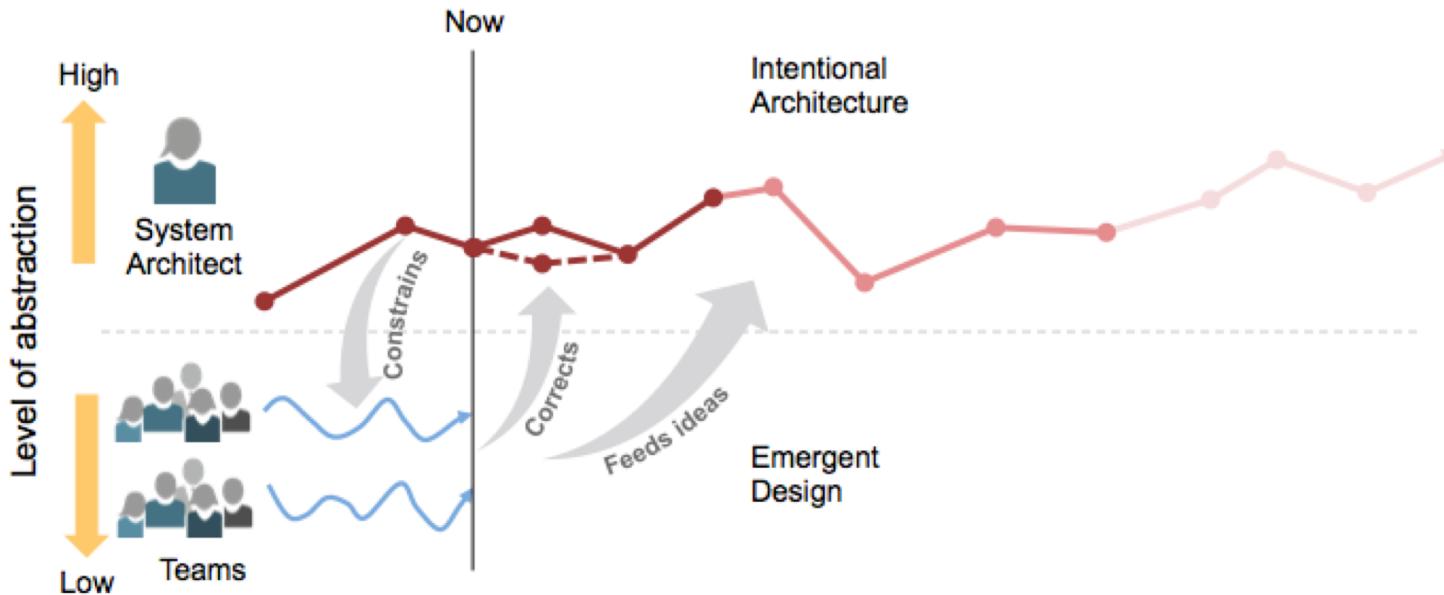


Intentional Vs Emergent Architecture

“While we must acknowledge emergence in design and system development, a little planning can avoid much waste.” —

James Coplien, Lean Architecture

- Architecture can just emerge, Emergent Design, and it can be Intentional, with just enough up front work to provide direction.
- Intentional Architecture corrects emergent design via collaboration.





The **Program** level uses three main activities to help coordinate the ART:

- **PI Planning** – A cadence-based, face-to-face planning event that serves as the heartbeat of the ART, aligning all the teams on the ART to the mission and vision.
- **System Demo** – Provides an integrated view of new features for the most recent iteration delivered by all the teams in the ART. Each demo provides ART stakeholders with an objective measure of progress during a PI.
- **Inspect & Adapt** – Is a significant event where the current state of the solution is demoed and evaluated. Teams then reflect and identify improvement backlog items via a structured problem-solving workshop.

Artefacts:

- **The Vision** is a description of the future state of the Solution under development. It reflects Customer and stakeholder needs, as well as the Feature and Capabilities, proposed to meet those needs.
- **Features and Epics;**
- **Program Backlog** – Holds the upcoming **Features**, and **Enablers** that build out the Architectural Runway.
- **Program Kanban** – It manages the flow of features and enablers through the Continuous Delivery Pipeline.
- **PI Objectives** – Specific business and technical goals that an ART intends to achieve in the next PI.
- **Architectural Runway** – Architecture work defined by **Enablers**.



- **Iteration Planning** – Is an event in which an Agile team determines the Iteration Goals and how much of the team backlog they can commit to during an upcoming iteration. Team capacity determines the number of stories and enablers that are selected.
- **Iteration Review** – Is a cadence-based event in which the team inspects the increment at the end of the iteration and adjusts the team backlog based on feedback. All work done during the iteration is demoed during the iteration review.
- **Iteration Execution** – Is how the Agile team develops an increment of an effective, high-quality, working, tested system within the timebox. Agile team holds a 15-minute timeboxed Daily Stand-up meeting to synchronize team members, review progress, and identify issues.
- **Iteration Retrospective** – Is an event held at the end of the iteration for the Agile team to review its practices and identify ways to improve. The retrospective is based on the qualitative and quantitative information presented during the iteration review.
- **Backlog refinement** – Is an event held once or twice during the iteration to refine, review, and estimate stories and enablers in the team backlog.
- **Innovation and Planning (IP) Iteration** – Provides the teams with an opportunity for exploration and innovation, dedicated time for planning, and learning through informal and formal channels. Where a release is on the PI boundary, teams perform final system verification, validation, and documentation. It also provided a planning buffer.

Role of PMO in SAFe In Program and Portfolios



The RTE and VSE roles have responsibility for steering delivery and facilitating Agile processes, including escalating impediments, managing risk and driving continuous improvement – all of which are core capabilities of a PMO team.

As the program and portfolios increase in scale, so do the activities and they cannot be discharged by one person. In these situations, the RTE/VSE may require a small, highly skilled team. This team can also act as a centre of excellence, participating in Lean-Agile transformation and coaching leaders, teams and Scrum Masters in the adoption of new processes.

- Team Scrum Masters.
- Agile Centre of Excellence
- Coaching people in the adoption of new practices
- Subject matter expertise in Agile delivery or change management
- The ability to prioritise a portfolio and align projects to strategy
- Demand and resource planning
- Financial and data analysis
- Program and portfolio planning
- Creating and maintaining Agile reporting and governance
- Risk, issue and dependency management