Achieving Regulatory and Industry Standards Compliance with the Scaled Agile Framework® (SAFe®)

A Scaled Agile, Inc. White Paper
August 2017

SAFe® PROVIDED BY SCALED AGILE®

Abstract

"It is not necessary to change. Survival is not mandatory."

–W. Edwards Deming

Many enterprises build high-assurance systems that have an unacceptable social or economic cost of failure. These include medical devices, automobiles, aircraft, banking and financial services, defense systems, and more. In order to protect the public, these systems are typically subject to extensive regulatory oversight and rigorous compliance standards. To reduce risk and ensure compliance, the organizations responsible for building these solutions have historically relied on comprehensive quality management systems that, in turn, incorporate stage-gated waterfall life-cycle models.

Unfortunately, given the dynamics of rapid advances in technology, market disruption, and a global economy, these current practices are proving inadequate to the challenge. These legacy approaches simply do not scale to the needs of large systems, even when development teams follow Agile practices. They also do not keep pace with the accelerating time-to-market demands of increased competition. Even when the higher Cost of Delay (CoD) is accepted as the price of doing business in regulated industries, a greater concern is that these traditional models do not always eliminate risk or increase quality. Massive automotive recalls are a common occurrence. Entire fleets of airlines have been grounded due to technical failures (e.g., Southwest and Delta within a month of each other in the summer 2016). The global failure and recall of the Samsung Galaxy Note 7 was both a financial and public relations nightmare. Leaders in these companies are looking for a better way.

The Scaled Agile Framework® (SAFe®) offers specific Lean-Agile success patterns to address these challenges. This white paper highlights those practices and addresses how SAFe allows companies building regulated, high-assurance systems to decrease risk while increasing quality, compliance, and transparency.

Table of Contents

Introduction

Traditional development models have historically obstructed organizational efforts to meet regulatory requirements. Practices from a waterfall legacy[1] create an environment with large batches of work, long cycles between system integration (builds), and delayed feedback on progress. Such an environment defers compliance activities until the end of the project, and provides little insight into progress throughout the lifecycle. This often results in missed deadlines, business outcomes that fall short of expectations, and lower quality. By contrast, Lean-Agile principles and practices strive to build in quality incrementally, early, and throughout the development lifecycle. This includes elements and activities that enable meeting regulatory mandates.

Regulatory Requirements meet Agile Development

At first glance, the practices associated with Lean-Agile and those associated with traditional compliance processes appear to be diametrically opposed, with conflicting goals and disparate communities. Through rigorous, stage-gated activities, the compliance world emphasizes quality, safety, and security to ensure that systems perform their intended purpose without causing harm. Those systems demonstrate adherence to specifications through verification and validation (V&V) activities, and often must provide evidence of adherence to standards through reviews, audits, and sign-offs. To this community, change and variability equal added risk and uncertainty.

By contrast, Lean-Agile development strives to discover the ultimate and optimal system iteratively, by creating an environment for learning. Building a working system in frequent, small batches confirms or rejects design hypotheses. Continuous customer/stakeholder collaboration provides fast feedback on decisions and the ability to adapt to new knowledge. Validated learning explores alternatives and helps ensure development creates products that meet the needs of customers. To this community, change and variability provide the ability to create products that excite customers and generate better economic results for the business. Figure 1 illustrates this conundrum.

[1] While pure waterfall model development is rare, few systems build and integrate the end-to-end solution with frequency and intention prescribed by Lean-Agile practices. System plans may include incremental "builds," but the build timeframes are often several months or even years instead of weeks. And the focus is not feedback for validated learning and adapting. We use the term "waterfall"here to imply both the mindset and the linear approach to product development.

Regulatory and compliance	Agile Manifesto
Quality, safety, security, efficacy Specifications Verification and validation Objective evidence Inspections, audits, sign-off Quality management systems Metrics – defects, requirements coverage, code coverage, traceability	**Working software** over comprehensive documentation **Customer collaboration** over contract negotiation **Responding to change** over following a plan Welcome changing requirements….

Figure 1. Contrasting traditional regulatory and compliance concerns with Agile values

Of course, businesses in high-assurance industries need and expect to achieve both goals. This whitepaper shows how to balance the needs of *both* communities by using Lean-Agile principles and the Scaled Agile Framework (SAFe).

The Role of the Quality Management System (QMS)

To satisfy compliance standards, organizations must demonstrate that their systems meet their intended purpose without causing harm. They must also have the objective evidence required to prove conformance to those standards. An organization's Quality Management System (QMS) defines policies, processes, and procedures that ensure development activities and outcomes comply with all relevant regulations, and provide the artifacts required to prove it. Compliance requirements originate from a range of statutory, regulatory, and industry standards. Compliance experts define their organization's QMS to aggregate all concerns, as shown in Figure 2.

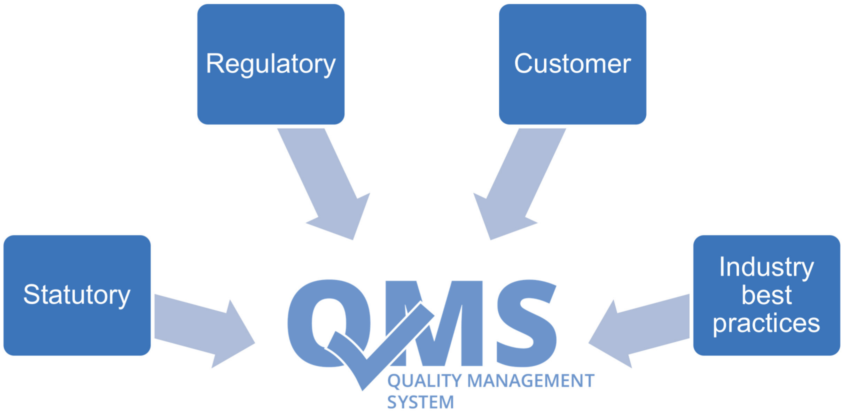
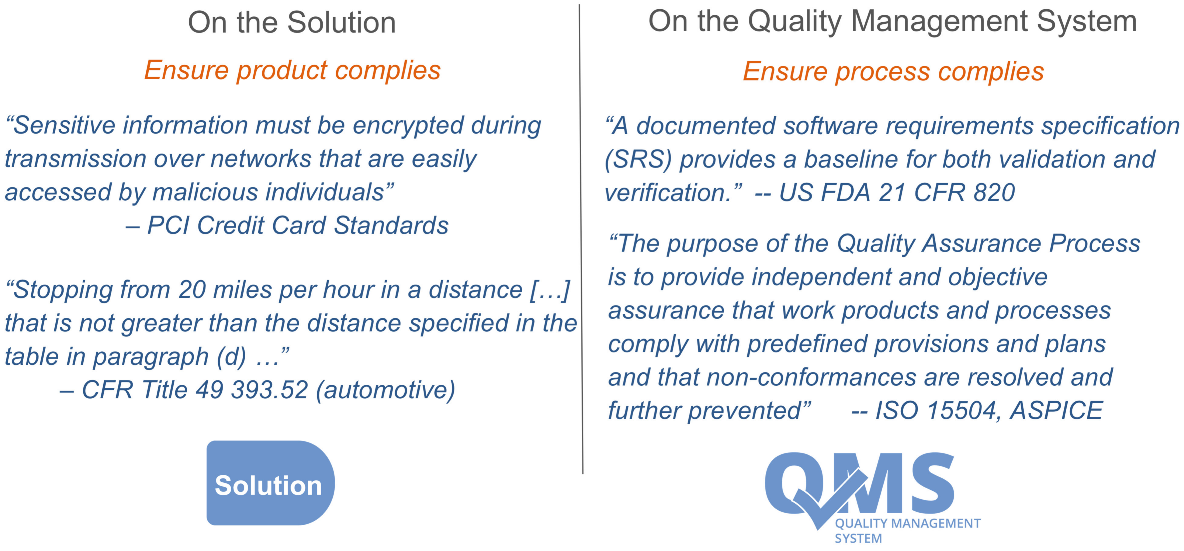


Figure 2. A Quality Management System integrates multiple compliance concerns

Regulations can govern both the *product* and the *process*. Product regulations typically govern the solution's function and performance, and are most often defined by statutory language. These rules manifest themselves as functional and non-functional system requirements. Process regulations also govern the development rigor (analysis, steps, documentation, reviews, etc.) required to demonstrate that the product meets its intended function and was built following industry standards and best practices. These guidelines are frequently reflected in the organization's QMS policies and procedures. Figure 3 provides examples of product regulation on the solution and process regulations on the QMS.



On the Solution

Ensure product complies

"Sensitive information must be encrypted during transmission over networks that are easily accessed by malicious individuals"
– PCI Credit Card Standards

"Stopping from 20 miles per hour in a distance […] that is not greater than the distance specified in the table in paragraph (d) …"
– CFR Title 49 393.52 (automotive)

On the Quality Management System

Ensure process complies

"A documented software requirements specification (SRS) provides a baseline for both validation and verification." -- US FDA 21 CFR 820

"The purpose of the Quality Assurance Process is to provide independent and objective assurance that work products and processes comply with predefined provisions and plans and that non-conformances are resolved and further prevented" -- ISO 15504, ASPICE

Figure 3. Compliance must address both the product and the process

While some regulations are not subject to interpretation (e.g., specific vehicle stopping distance), many product regulations and most process regulations require interpretation by the organization. Regulations define the "what." The organization defines "how." Sometimes the "how" constrains or bottlenecks teams adopting Lean-Agile practices who are striving to deliver quickly and continuously. Organizational interpretations of regulations embedded in the QMS can be more lore than fact, treated as gospel not to be questioned. As we will discuss later in this whitepaper, Lean-Agile continuous improvement practices often touch parts of the entire organization, including compliance. To be successful, everyone in the organization must be aligned on the changes presented here, even those who have historically resisted change.

As an example, *US FDA 21 CFR 820* regulations state that a "documented software requirements specification (SRS) provides a baseline for both validation and verification." A traditional interpretation would define and detail the SRS up-front in order to address the requirement. The organization's processes may even define an early phase gate milestone review to "ensure" the SRS quality. By contrast, a Lean-Agile interpretation would use a vison document, a backlog, and/or a much lighter SRS used to align and focus everyone on a common direction. These would contain detail sufficient only to gain agreement on what is being built. Any added detail is simply speculation at that time. In this model, the SRS evolves over time, and teams continually keep the solution, its specifications, and its tests synchronized.

The previous example demonstrates how a traditional, legacy waterfall process and mindset can embed itself in our compliance culture, along with its many problems and issues. Examples include early commitment to an incomplete or inaccurate specification, limited opportunity for feedback or validating assumptions, late discovery of issues, and no systematic way to improve. But it doesn't have to be this way, as Dean Leffingwell notes:

> *"Any notion that we are mandated to apply a single-pass, waterfall model to software development is an industry myth, one which has likely been perpetuated by our own waterfall past ("we have always done it this way") and our existing quality management system, and not because 'the regulations make us do it.'"*
>
> ***–Dean Leffingwell***

Implementing a Lean QMS

Unfortunately, the waterfall-centric QMS system seriously inhibits, and can even prevent, the adoption of newer methods, as the older methods are hard coded into the only approved way of working. As Figure 4 illustrates, SAFe describes an incremental approach to both development and compliance. This means those who want the benefits of Lean-Agile development (faster time to market and higher quality to name a few) will typically have to evolve a Lean QMS.

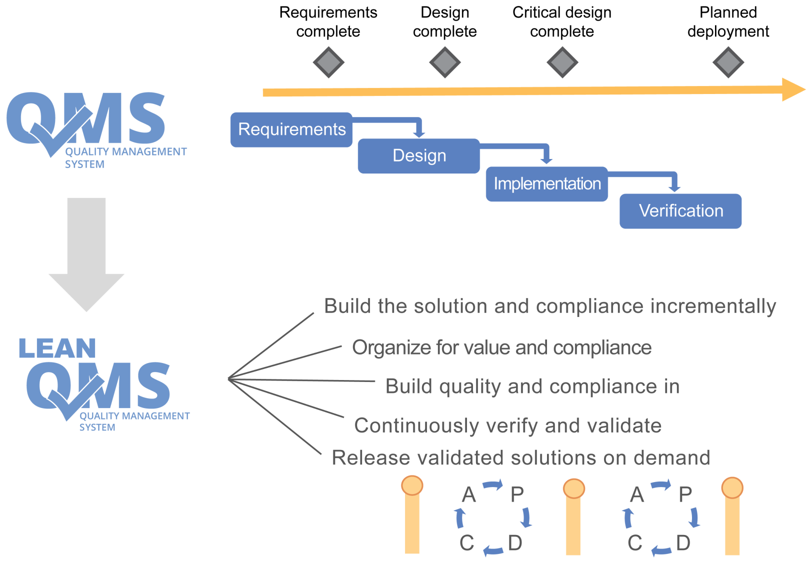

Requirements complete
Design complete
Critical design complete
Planned deployment
QMS
QUALITY MANAGEMENT SYSTEM
Requirements
Design
Implementation
Verification
LEAN
QMS
QUALITY MANAGEMENT SYSTEM
Build the solution and compliance incrementally
Organize for value and compliance
Build quality and compliance in
Continuously verify and validate
Release validated solutions on demand
A
P
C
D
A
P
C
D

Figure 4. A Lean-Agile QMS improves quality and makes compliance more predictable

This whitepaper shows how Lean-Agile and compliance goals coexist using SAFe to make progress more visible, and to incorporate compliance activities into the regular flow of work. This includes:

- Taking an incremental approach to creating and assessing compliance information
- Including compliance teams and their concerns in the product development ecosystem to collaborate on planning, executing, assessing, and adapting
- Incorporating compliance in Agile quality practices – automating, adapting, and continuously improving
- Integrating V&V and compliance activities into the iterative development flow

Each of these topics is discussed in the sections below.

Build the Solution and Compliance Incrementally

A large misconception in traditional systems engineering models based on a waterfall approach is that the ideal end solution can be known up-front, *before development begins*. The belief is that with enough documentation and check points along the way, the only remaining challenge is to build exactly what the specifications require. But picking a 'point solution' too early can create poor outcomes, as Figure 5 illustrates.

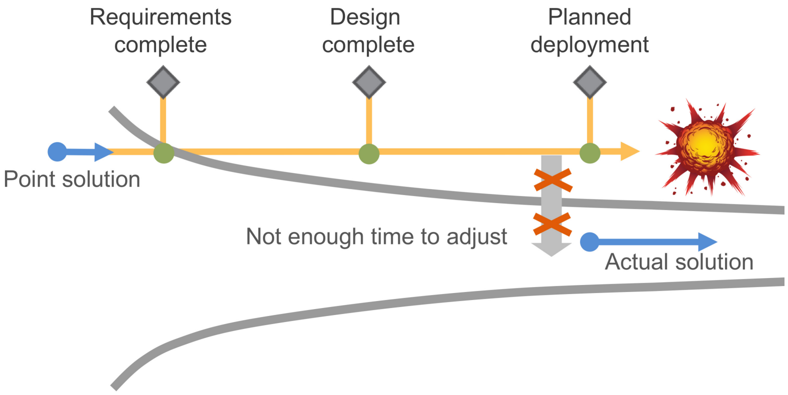


Figure 5. Point solutions increase the risk of failure

More realistically, when development begins, engineering teams do not have all the answers. Instead, they have a set of hypotheses that must be tested through a series of short, iterative experiments that either prove or reject them. In Figure 4, the hypothesis was that the point solution identified in the beginning was going to meet the needs of the organization. However, the optimal decision was actually quite different, and that insight was not gained by using traditional stage gates. When we consider Deming's learning cycle of Plan-Do-Check-Adjust (PDCA), the problem becomes obvious. Waterfall models do not provide frequent, continuous PDCA learning cycles, offering no objective measure of progress (working hardware and software) to create validated learning. Figure 6 contrasts this approach with the continuous learning loops that occur in an incremental development model.

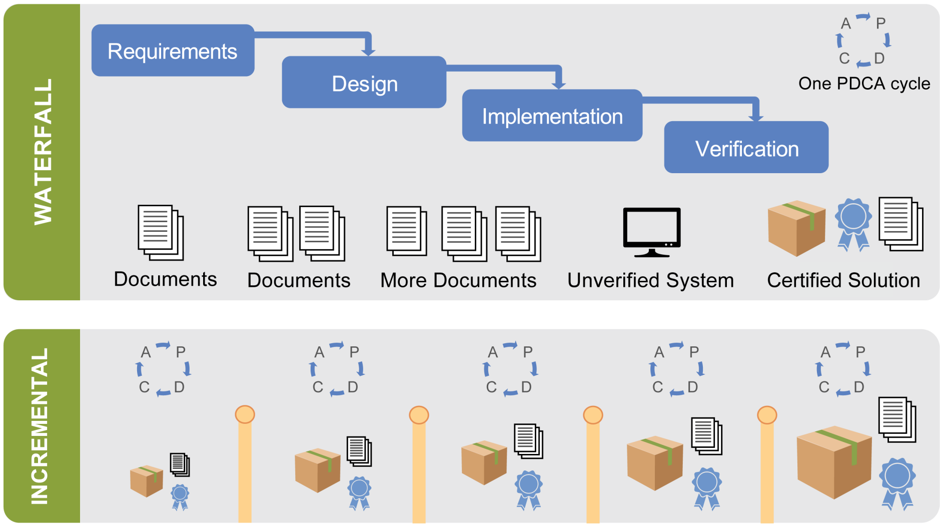


Figure 6. Rapid learning cycles increase quality and reduce risk

There are two important sentiments here. First, building smaller, working parts of the solution early allows V&V and compliance activities to begin sooner, preventing the large bow wave of testing and compliance activities at the end. As Figure 5 shows, each increment assesses the current solution, as well as progress towards compliance, providing early feedback on the system's ultimate release-ability. Second, specifications are created over time, in small batches, with faster feedback on decisions, and the opportunity for continuous review and assessment. In short, specifications evolve with the system.

SAFe addresses this by focusing on small, predictable batches of new functionality and frequent PDCA cycles through cadence and synchronization of Iterations and Program Increments (PIs) of multiple Iterations. Validated learning from each PDCA cycle is immediately incorporated into the planning of the next Iteration or increment. Compliance concerns are addressed and improvements applied in small batches as the system evolves, instead of waiting until the end and addressing compliance as one giant batch.

Organize for Value and Compliance

Large enterprises have historically structured their organizations around technical domains (systems, software, mechanical, testing, quality, security, operations, etc.). This has the advantage of grouping like skillsets and simplifying people management. However, this organizational construct creates silos, institutionalizing the risks of increased handoffs between groups, which delays value delivery and decreases quality. A better approach is to organize horizontally, aligning cross-functionally to support 'Value Streams,' a foundational construct of Lean.

Value Streams are long-lived sequences of activities required to create a continuous flow of value to customers. Organizing around Value Streams establishes a virtual organization of all the people, information, resources, and materials required to produce end-to-end value, as shown in Figure 7.

Members of the value stream are aligned, committed (primarily full-time, not shared resources), and integrated to quickly collaborate on defining, building, and assuring the system. Compliance is a critical part of the value stream. Its concerns and work must be included in value stream activities, which are discussed in depth later in this paper.

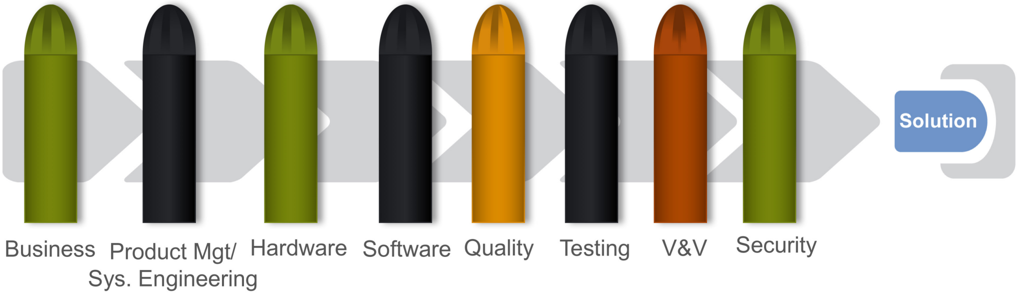


Figure 7. Value Streams cut across organizational silos to deliver value

Figure 8 shows how SAFe realizes value streams via the organizational construct of the 'Agile Release Train' (ART). The ART is a team-of-teams who work together, plan together, demonstrate results together, and continuously improve together. SAFe provides deep guidance and practices to align ARTs on a common goal, removing the us-vs-them, toss-over-the-wall mentality. ART teams and individuals are committed to each other's success. Dependencies are resolved quickly without the need to traverse organizational barriers. Rapid learning cycles occur not just within individual teams, but also across the entire team of teams.

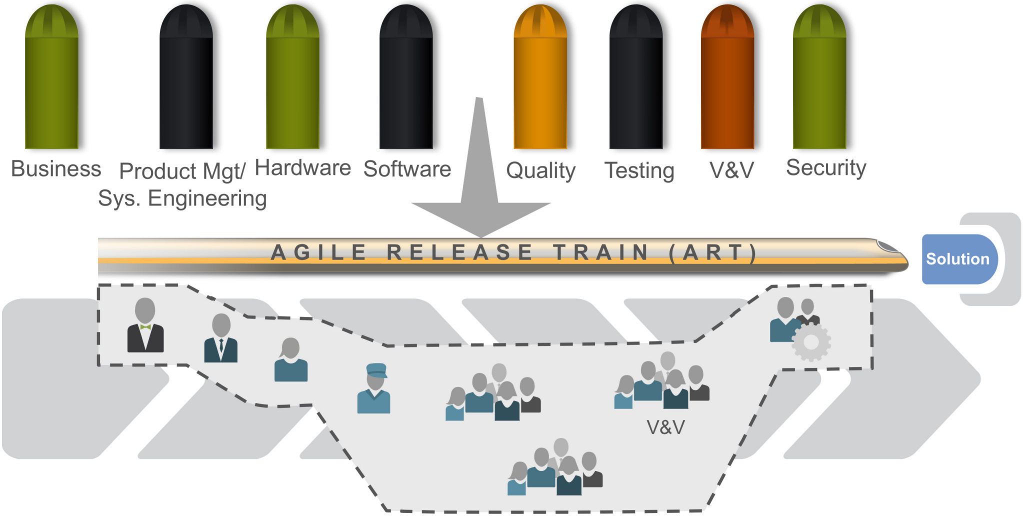


Figure 8. Agile Release Trains bring together all disciplines, including quality, testing, and IV&V

To achieve success, ARTs require all the skills necessary to build and release their solution, including those responsible for quality assurance (QA), security, testing, V&V, and even independent validation and verification (IV&V). While some regulations require independence for compliance representatives (typically through reporting and incentives), their continuous participation as members of the ART is critical for overall solution success. Some specialized skills, e.g., QA, may be shared across teams, or even across ARTs and appear as Shared Services in SAFe.

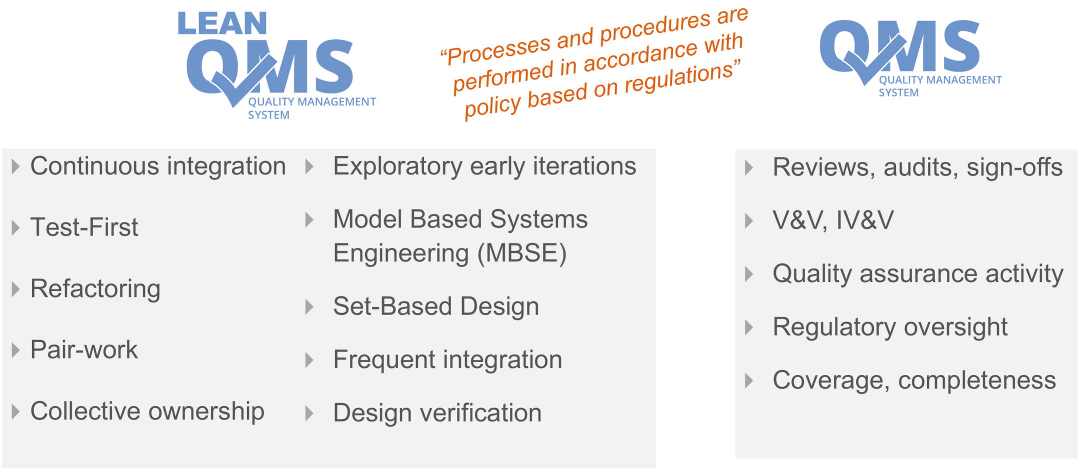
The ART aligns teams to a common mission via a single Vision, Roadmap, and Program Backlog. Alignment includes all concerns including compliance. Consequently, we need compliance personnel participation to maintain all artifacts and we need them participate in ART events. Program Managers ensure the Program Backlog properly reflects compliance priorities. During PI Planning, teams ensure that their work includes appropriate compliance activities. And during Inspect and Adapt (I&A), the entire ART looks for better ways to build quality and compliance concerns into the regular flow of work. Participation in these events is discussed later in this paper.

Build in Quality and Compliance

Built-in Quality is one of SAFe's four Core Values, as well as a core principle of the Lean-Agile Mindset. It helps avoid the cost of delay associated with recall, rework, and defect fixing. Lean builds quality in by automatically detecting abnormalities and, when detected, stopping the entire system to focus everyone on resolving the problem. This philosophy 'takes a systems view' to optimize the whole, ensuring fast flow across the entire value stream, and making quality everyone's job. In Lean, quality is a culture, not a role or job title.

In the same way, compliance concerns are built into the development process. In Figure 9 below, the box on the left lists Lean-Agile practices for ensuring quality. Agile strives to build these practices into the regular flow of work, performing them consistently and continually. The box on the right lists common compliance activities that should also be built into the regular flow. Reviews and audits are performed as work in the backlog (Stories) and completed in small batches. Ideally, V&V and IV&V teams receive solutions at every Iteration, and certainly every Program Increment (PI) for testing and quality assessments, versus at the end of a 12-month "build" cycle.

Additionally, compliance metrics such as code coverage and traceability completeness are assessed continuously at each iteration to ensure they are trending towards a successful delivery.


LEAN QMS QUALITY MANAGEMENT SYSTEM
"Processes and procedures are performed in accordance with policy based on regulations"
QMS QUALITY MANAGEMENT SYSTEM

- Continuous integration
- Test-First
- Refactoring
- Pair-work
- Collective ownership

- Exploratory early iterations
- Model Based Systems Engineering (MBSE)
- Set-Based Design
- Frequent integration
- Design verification

- Reviews, audits, sign-offs
- V&V, IV&V
- Quality assurance activity
- Regulatory oversight
- Coverage, completeness

Figure 9. Build-in quality and compliance practices

SAFe also builds quality in through frequent integration and automated testing of system components, as illustrated in Figure 10. Developers automate the build-deploy-test process to support continuous, frequent integration and fast learning cycles. Figure 10 also shows how compliance roles help ARTs automate regulatory concerns as part of the automated process.

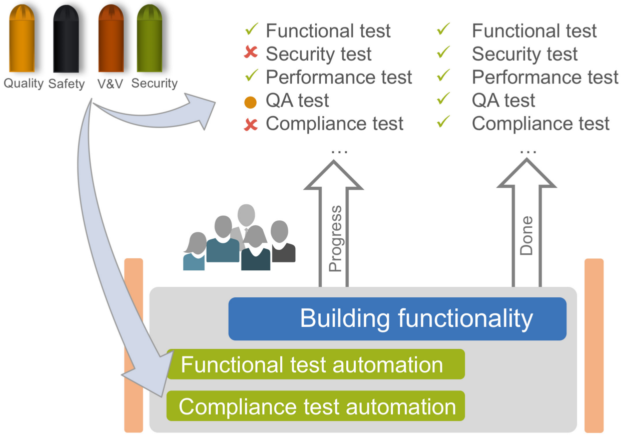


Figure 10. Build compliance concerns into the design-build-test automation

Unfortunately, not all activities can be automated, as some regulatory requirements mandate manual reviews and audits. But like all Lean-Agile activities, they are decomposed into work on the teams' backlog and performed within the Iteration and/or PI. The goal is to reduce the last sign-off activity from a large, extended event to a quick, boring, non-event through small batches performed as part of flow. Strategies for making compliance activities part of flow are discussed in the next section.

Once part of flow, the program receives fast feedback on the degree to which the teams' compliance activities are meeting compliance objectives and, conversely, how those compliance activities may be impacting team performance. Figure 11 shows the feedback cycle between team activities and the practices defined by the Lean QMS. Teams and compliance experts collaborate to understand the best solution for gathering the proper data and evidence to meet compliance goals. They then build those activities into the regular Agile flow of work.

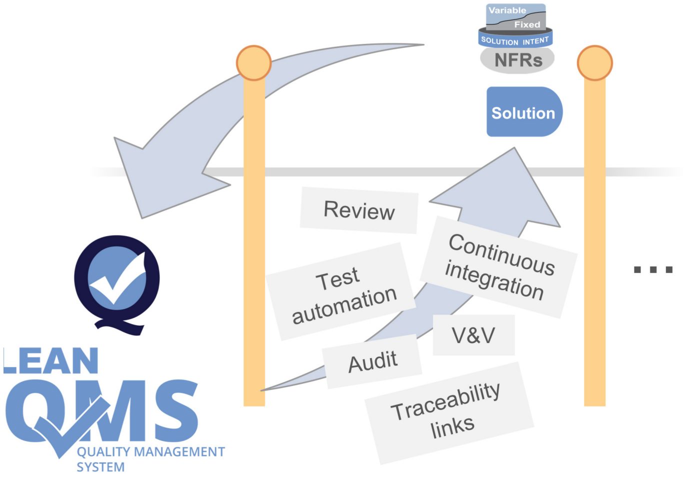


Figure 11. Program Increments provide a feedback loop for compliance activities and practices

Continuously Verify and Validate

SAFe takes a Lean-Agile approach to requirements and their use for verification and validation. Traditional practices decompose system requirements hierarchically, using a system-to-subsystem, system-to-hardware/software, or other nested relationships. These requirement hierarchies are defined early by lead systems engineers and handed off to teams for implementation over the full development lifecycle. Normally, they would be considered fixed and often require significant effort to change.

Figure 11 shows the contrasting Lean-Agile approach, where the system requirements are decomposed into hierarchical backlog items. Backlogs are managed by teams who detail them just in time to be implemented. This Lean-Agile process for requirements management better facilitates fast learning cycles and adaptation based on new knowledge. It also helps give system builders—the people who know best how to implement the system—flexibility, providing better economic decisions and faster delivery. Note that both traditional and Lean-Agile approaches provide a set of system requirements to be used as the basis for V&V and compliance activities.

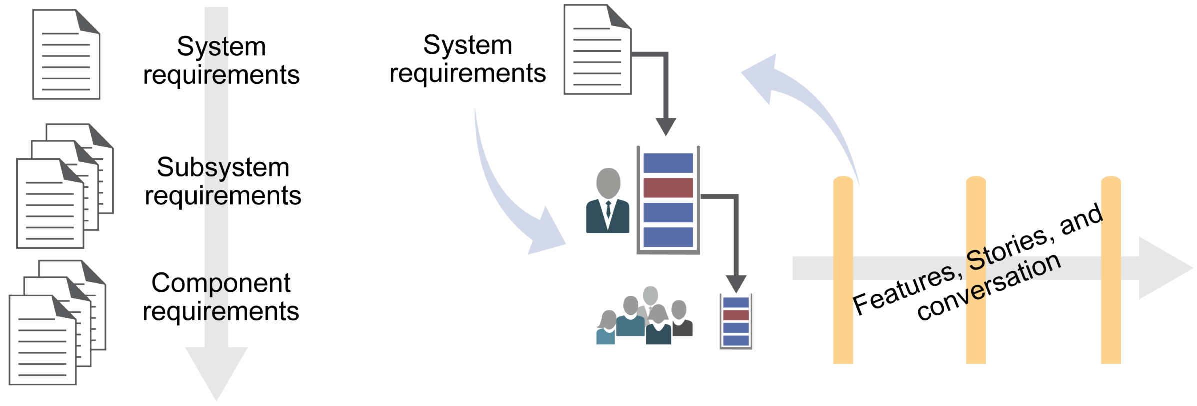


Figure 12. Contrasting traditional and Lean-Agile requirements decomposition

SAFe captures requirements and designs used for V&V in the Solution Intent. System requirements, as described above, are managed in a requirements model that may be further decomposed into other system models (see Figure 13). These models, and the traceability to support compliance, evolve in tandem with the system under development. Verification determines that the current system increment was built according to the specifications captured in the backlog, and in alignment with the Solution Intent (i.e., we built the solution right). Validation determines if the increment's backlog items meet the system's fitness for use (i.e., we built the right solution). Traceability within the Solution Intent ensures the artifacts produced each increment (software, hardware components, etc.) address regulatory and compliance specifications, providing end-to-end traceability (evidence) that V&V requirements have been met.

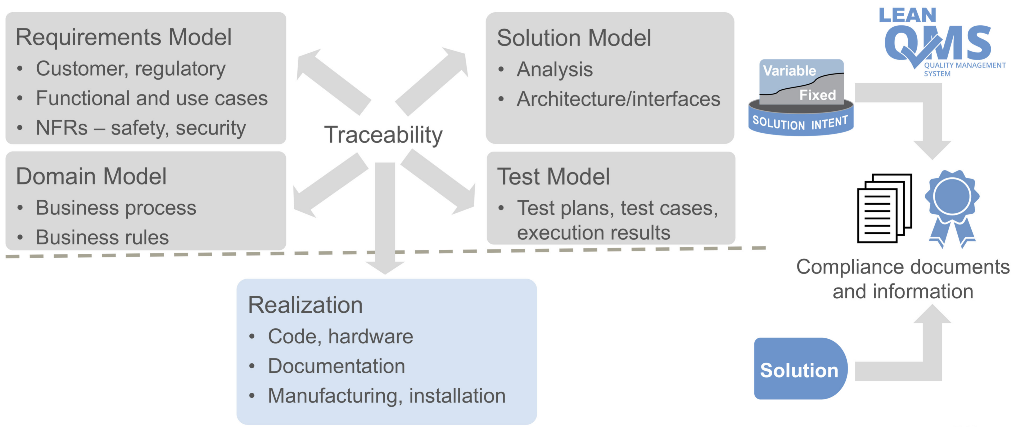


Figure 13. SAFe's Solution Intent provides the traceability needed for verification

The SAFe Requirements Meta-Model

As shown in the previous section, system-level requirements in SAFe are managed in the Solution Intent, while the requirements details are managed by the teams in backlog items. In SAFe's meta-model, all forms of requirements have test cases (see Figure 14), created over time by the teams doing the work. Each Increment yields new functionality and also adds new tests to the solution's test space. As the number of tests grows, emphasizing test automation and providing sufficient test resources are vital to ensuring that testing activities do not become a bottleneck.

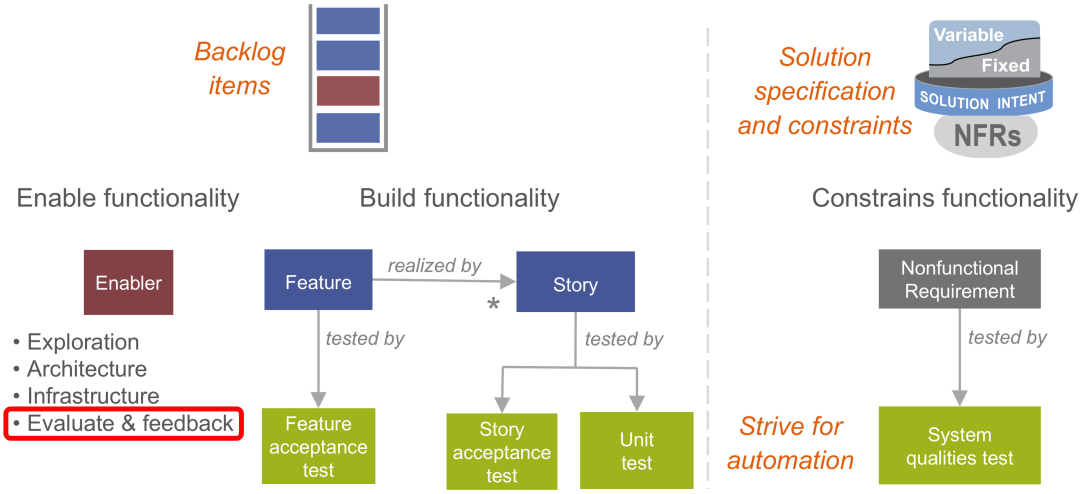


Figure 14. SAFe's requirements meta-model supports verification and validation

Compliance work appears in two places in the meta-model. First, it can be part of the Definition of Done (DoD) for any backlog item. Some DoD examples include ensuring each Story is peer reviewed, or that impacted requirements are revisited for each Feature. The second place compliance work appears is in the backlog as planned work for the team, including formal reviews and other scheduled items required for compliance. SAFe provides an 'Enabler' backlog item type for this purpose, allowing teams to identify and estimate compliance work and to ensure it is prioritized, estimated, and has visibility in the backlog.

Make V&V and Compliance Activities part of Regular Flow

SAFe supports continuous verification by incrementally building the necessary artifacts in the Solution Intent over a series of PIs. Verification activities are implemented as part of flow (as backlog items or DoD as described above). While the artifacts will satisfy the complete objective evidence needs at the end of development, they are created iteratively throughout the life of the system. SAFe also supports continuous validation as Product Owners (POs), customers, and end users participate in ART planning and demos to help validate decisions.

Figure 15 illustrates continuous V&V and compliance. In this example, regulations require design reviews and that all actions be recorded and resolved. The "Design review" Enabler backlog item provides the objective evidence of the review and its DoD ensures actions are recorded and resolved. If needed, the actions themselves could be tracked as Enabler Stories. Regulations may also require that all changes be reviewed, which is addressed by a required "Peer review" DoD for all Stories.

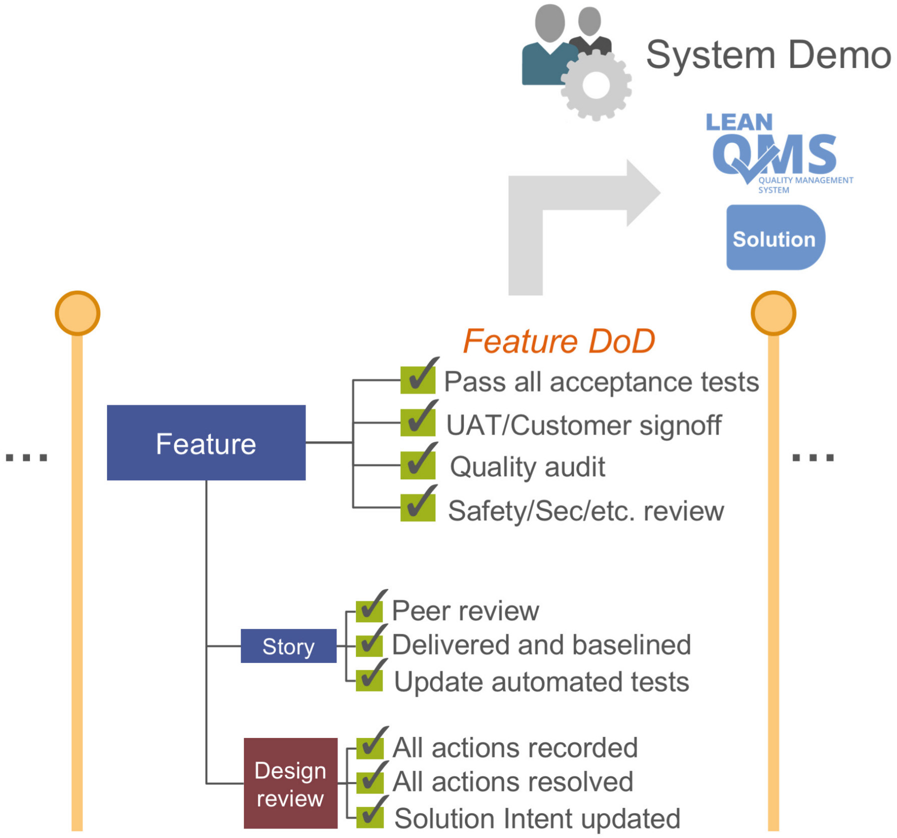


Figure 15. Verification and validation concerns are built into the system as part of flow

SAFe builds and demonstrates the integrated solution frequently, at least as frequently as the Iteration System Demo. Building and integrating frequently allows for continuous validation from User Acceptance Test (UAT), customers, and end users. Each Iteration, the System Demo provides objective evidence that the integrations are performing as intended, and that the entire system has advanced forward, maintaining quality and compliance requirements.

Program Increment (PI) boundaries provide additional opportunities to conduct V&V activities incrementally and assess progress towards compliance, as shown in Figure 16. In SAFe, each PI ends with an Innovation and Planning (IP) Iteration that provides, among other things, time to integrate and assess the results of the last PI and collect metrics to support the PI's Inspect & Adapt (I&A) workshop. I&A is a regular time to reflect, collect data, solve problems, and take action. It consists of three parts: the PI System Demo, Quantitative measures to review metrics and trends, and the Retrospective and Problem Solving Workshop.

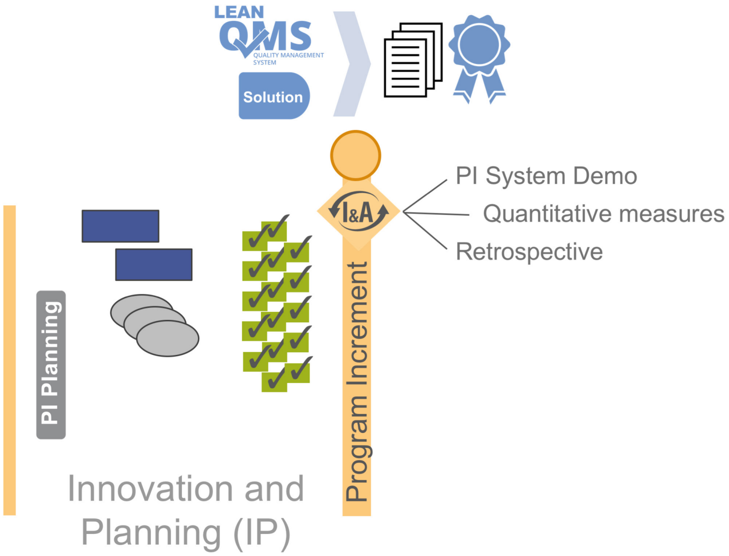


Figure 16. Program Increments help ARTs assess V&V and compliance progress

The PI System Demo includes results of compliance work. This might include infrastructure building to better support automating tests and compliance, results and feedback from significant reviews or audits, and results of any milestone compliance events (e.g., flight test, clinical trial).

Qualitative measures show, among other metrics, the data and trends towards certifying and releasing the product. Compliance measures might include current status on requirements coverage, testing code coverage, and peer review results. SAFe also offers several relevant quality metrics - defects, total test, percentage of tests automated, etc. Assessing metrics each increment requires a Lean-Agile mindset. After all, the goal is not simply to meet compliance requirements each increment. The goal is to understand how the program is progressing towards achieving compliance and identify areas of concern that need to be addressed. From this perspective, trend data for these metrics is typically more interesting than the data points at any PI boundary.

Finally, the Retrospective and Problem Solving Workshop includes concerns from compliance activities. Is the program sufficiently addressing compliance goals? Do the metric trends indicate that the solution will meet any upcoming milestones required for certification? Are policies or procedures to ensure compliance inhibiting development or restricting flow? From these types of questions come potential compliance improvements to explore in an upcoming PI.

Release Validated Products on Demand

Releasing is the last step in SAFe's Continuous Delivery and Release on Demand pipeline. Only via the release can the business monetize its development investment, assess benefit hypothesizes, and receive market feedback. However, the decision to release must balance the value these benefits offer with the transaction cost of performing the release. As compliance becomes part of flow, release transaction costs decrease such that benefits of more frequent releases start to outweigh the costs. The continuous compliance approach described here offers a substantial reduction in the risks and costs associated with releasing high-assurance systems.

That said, the final release process in high-assurance environments may have additional steps:

- Generating the final release solution with its required documentation and objective evidence
- Baselining and archiving the final configuration
- Final testing that may be performed in the true operational environment and/or witnessed by assessors
- Final approvals by internal and/or external assessors (quality, safety, security, etc.)

And since both quality and compliance have been built-in throughout the development process, these last activities can move from large, extended events to become routine and integrated activities.

Final Thoughts

Organizations operating in high-assurance, compliant environments have historically struggled to balance the principles and practices of Lean-Agile development with the hard reality of having to create systems that simply cannot fail. Some Agile methods and even practitioners share in this blame. Team-level Agile practices strive to optimize teams, not value streams, and may leave the (incorrect) impression that making teams go faster with Agile equates to no documentation, no requirements, no specifications, and even no oversight. Consequently, some organizations have drawn the conclusion that Agile practices cannot work in their industry.

As this white paper has shown, SAFe is rooted in Lean and Agile principles, and focuses on optimizing the whole system through continuous delivery, organizing around value, and building quality practices into the system. SAFe has the necessary features to address the scalability, quality, and compliance needs for these organizations. SAFe is currently being used on many product development efforts, including medical equipment, weapons systems, commercial aircraft, space vehicles, banking and finance services. Lean-Agile practices at scale through SAFe are not only possible in high-assurance product development. SAFe actually helps these programs improve their regulatory compliance while delivering value with higher quality, greater predictability, and faster time to market.

Learn More

If you would like to learn more about SAFe, visit these websites:

- Learn about real world implementations at **scaledagileframework.com/case-studies**
- Browse the Framework at **scaledagileframework.com**
- Find role-based SAFe training and certification at **scaledagile.com**
- View SAFe presentations and videos at **scaledagileframework.com/videos-and-presentations**
- Read SAFe Distilled: Applying the Scaled Agile Framework® for Lean Software and Systems Engineering – **scaledagile.com/safe-distilled**
- Read Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise – **bit.ly/AgileSWReq**

SCALED AGILE®

About Scaled Agile, Inc.

Based in Boulder, Colorado, Scaled Agile's mission is to help system and software-dependent enterprises achieve better outcomes, increase employee engagement, and improve business economics through adoption of Lean-Agile principles and practices based on the Scaled Agile Framework® (SAFe®). Scaled Agile supports over 130,000 practitioners of the Framework through training, certification, consulting services, and a global partner network that reaches over 35 countries and 350 cities.

Contact

Scaled Agile, Inc.

5480 Valmont Rd, Suite 100

Boulder CO 80301 USA

+1.303.554.4367

support@scaledagile.com

scaledagile.com