**White Paper: Agile Framework**

Scaling Agile and the Evolving Role of Architecture in Enterprise Delivery (VA Context)

White Paper: Agile Framework – Scaling Agile and the Evolving Role of Architecture in Enterprise Delivery (VA Context)

## Executive Summary

Agile methods have redefined software development by encouraging iterative delivery, customer-centric design, and responsive planning. Within the VA, scaling Agile practices demands careful framework selection (SAFe, LeSS, Scrum@Scale) and alignment with governance structures like the Veteran-focused Integration Process (VIP) and Technical Reference Model (TRM). This paper details how architecture plays a critical role in large-scale Agile environments and provides a roadmap for integrating architectural leadership within Agile teams.

1. Scaling Agile: Frameworks for the Enterprise

## 1.1 SAFe – Scaled Agile Framework

- Overview: Combines Lean thinking, Agile practices, and DevOps into a comprehensive approach suited for large programs.

- Core Layers:

- Team: Basic unit using Scrum/Kanban.

- Program (ART): Group of 5–12 Agile teams delivering in sync.

- Solution: For large solutions spanning multiple ARTs.

- Portfolio: Aligns initiatives with strategic goals and funding.

- Cadences:

- Program Increment (PI) planning every 8–10 weeks.

- Synchronization ceremonies across teams to align work.

- Artifacts:

- Epics: Large initiatives.

- Features and Stories: Iterative breakdown.

- Roles:

- Release Train Engineer (RTE): Facilitates ART.

- Solution Architect: Ensures solution-level coherence.

- System Team: Handles CI/CD and DevOps support.

## 1.2 LeSS – Large-Scale Scrum

- Overview: Extends Scrum for multi-team setups without adding hierarchy.

- Key Principles:

- Shared product backlog.

- Single Product Owner.

- Common Definition of Done across teams.

- Minimal Roles:

- No extra hierarchy; maintains Scrum simplicity.

- Coordination:

- Shared sprint planning.

- Joint sprint reviews and retrospectives.

## 1.3 Scrum@Scale

- Overview: Modular scaling model co-created by Jeff Sutherland.

- Two Meta-Cycles:

- Scrum Master Cycle: Improves delivery by removing impediments.

- Product Owner Cycle: Aligns delivery with strategy.

- Executive MetaScrum: Links executive priorities to execution.

- Benefits:

- Lightweight, customizable.

- Suitable for federated orgs.

2. The Role of Architecture in Agile

Agile doesn’t replace architecture—it shifts the function from upfront design to continuous collaboration.

## 2.1 Epic Owner

- Definition: A leadership role in SAFe, stewarding the delivery of cross-cutting initiatives.

- Responsibilities:

- Draft Epic Hypothesis Statements and Lean Business Cases.

- Align work with Lean Portfolio Management.

- Collaborate with solution and enterprise architects.

- Prioritization Tool: WSJF (Weighted Shortest Job First).

## 2.2 Architecture Runway

- Definition: Technical groundwork to support near-term development without delays.

- Includes:

- Foundational APIs, services, and infrastructure.

- Ready-to-use integration points.

- Managed by:

- Architects in SAFe.

- Technical leads in LeSS or Scrum@Scale.

- Maintained through:

- Enabler stories and architectural spikes.

## 2.3 Incremental Design

- Core Idea: Just Enough Design, Just In Time (JEDI).

- Practice:

- Use spikes to explore unknowns.

- Begin with a minimal architecture and adapt based on feedback.

- Use fitness functions to maintain design integrity.

- Team Collaboration:

- Architects embed with teams.

- Architecture Decision Records (ADRs) document evolving decisions.

3. Example: Scaling Agile for a National Benefits Platform

Scenario: A benefits modernization initiative uses 15 Scrum teams.

## - Framework: SAFe with full portfolio configuration.

- Highlights:

- Epic Owner collaborates with Program Office on legislative alignment (e.g., PACT Act).

- Agile Release Train includes teams for backend, frontend, notifications, and claims.

- Architecture Runway includes:

- Shared API Gateway

- Event streaming platform (Kafka)

- Zero Trust foundations

- Incremental Design:

- Evaluate GraphQL vs. REST in early PIs using spikes.

- Build API adapters iteratively.

4. VA Integration: Agile and the VIP Process

- VIP Overview: Veteran-focused Integration Process ensures compliance with cybersecurity, accessibility (Section 508), and governance.

## - Agile-VIP Alignment:

- PI Planning aligns Epic delivery to VIP Milestones 2 and 3.

- SSP and POA&M documentation built incrementally within sprints.

- Architecture reviews mapped to TRM requirements.

## - Benefits:

- Enables early-stage ATO planning.

- Synchronizes technical velocity with compliance milestones.

## 5. Key Resources

| Area | Resource |

|---------------------|-----------------------------------|

| SAFe Reference | SAFe Architect Guide |

| LeSS Knowledge Base| LeSS.works |

| Scrum@Scale Toolkit| Scrum@Scale |

| Certifications | Scrum.org |

| Architecture | ThoughtWorks Tech Radar, Clean Architecture |

| Agile + Architecture| "Agile Architecture in SAFe" White Paper |

## 6. Comparison Summary

| Feature | SAFe | LeSS | Scrum@Scale |

|---------------------|----------------|--------------|-----------------------|

| Structure | Highly structured | Minimalist | Modular, federated |

| Team Count | 100+ | ~8 | Flexible |

| Architecture Role | Explicit roles | Shared | Decentralized |

| Portfolio Management| Formal Epics | Light | Executive MetaScrum |

| Best Fit | Federal, regulated | Startups | Distributed orgs |

## Conclusion

VA’s path to scalable Agile hinges on deliberate framework selection, integration with VIP governance, and evolving the architect’s role. Agile frameworks are not one-size-fits-all, and the architecture function must scale to support decentralized teams while aligning enterprise goals.

## Next Steps:

- Align roadmaps with TRM and VIP checkpoints.

- Equip architects for Agile roles through training and CoPs.

- Standardize Epic and Enabler templates to support compliance.

- Build architecture playbooks for Agile delivery environments.