Developing Complex Solutions with SAFe

Alex Yakyma
SAFe Fellow and Principle Consultant

© 2016 Scaled Agile, Inc. All Rights Reserved.
Start simple and lightweight…

SAFe® 4.0 for Lean Software and Systems Engineering

Coordination

Expand one level

AGILE RELEASE TRAIN

Release Any Time

Program Increment

Program Runway

Built-In Quality

Provided by Scaled Agile, Inc.
And expand to scale large Value Streams
The problem of product development at large scale

Late delivery

Phase gate SDLC isn't helping reduce risk

Problems discovered too late

Massive growth in complexity

System #1

Too little visibility

Under-estimated dependencies

Poor morale

Hard to manage distributed teams

System #2

Common cause: both the organization and the solution under development are not designed for flow

© 2016 Scaled Agile, Inc. All Rights Reserved.
System #1: The Enterprise
Organize around value

*Align the organization around projects and product lines.*
—Allen C. Ward

1. Fewer handoffs, faster value delivery
2. Easier to build in quality
3. Built-in alignment between the business and software development
4. Optimizing the system as a whole

Result: Higher customer satisfaction
Example 1: IT

Operational Value Stream (Example: Order processing)

Trigger: Customer order

People who operate the systems

Value: Order delivered

Systems:
- CRM
- Shipping/Tracking
- Billing

Development Value Stream

People who build the systems
Example 2: Solution builder

For solution builders, operational and development Value Streams are the same.
Build Agile Release Trains

- A long-lived, self-organizing Team-of-Agile Teams

- A virtual organization of 5–12 teams (50–125+ individuals) that plans, commits, and executes together

- Includes all functions critical to value delivery

- Operates on a cadence

- Aligned to a common mission via a single Program Backlog

- Frequently produces valuable and evaluable system-level solutions
Value Streams are realized via ARTs

Multiple, smaller Value Streams can be realized by a single ART

Product 1
Product 2

Some Value Streams fit well within the limit, and can be realized by a single ART

System

Larger Value Streams require multiple ARTs

Large solution
What to organize an ART around?

ARTs can be organized around Subsystems or Capabilities

For example: A train can be organized around “Customer Enrollment” end-to-end functionality (Capability area) or alternatively, can work on mobile and tablet clients (Subsystem)
The implications of org structure

Orthogonal to value

Aligned with value

Capability

Capability

Capability

Capability
Operate on aligned cadence

Establish objective governance and coordination across trains
SAFe structure, 10,000 ft view
System #2: The Solution
Enable Flow and Agility with Architecture

Otherwise you will likely encounter:

- Unmanageable dependencies across teams
- Unsustainable enterprise velocity
- Inability to quickly respond to change
- Poor quality, deployability and maintainability of solutions
Emergent design and intentional architecture

Every team deserves to see the bigger picture. Every team is empowered to design their part.

- Emergent design – teams grow the system design as user stories require
- Intentional architecture – fosters team alignment and defines Architectural Runway

A balance between emergent design and intentional architecture is required for speed of development and maintainability of the Solution
Architectural Runway

Architectural Runway—existing code, hardware components, etc. that technically enable near-term business features

Example: A new, fuzzy search algorithm will enable a variety of future features that can accept potentially erroneous user input

- Enablers build up the runway
- Features consume it
- Architectural Runway must be continuously maintained
- Architectural capacity allocation supports building runway
Build and Validate Incrementally

- Test architectural assumptions early and often to prevent waste
- Validate the runway by building thin slices of value on top of it
- Design for Testability

![Diagram showing the process of implementing a part of an architectural feature, validating it through a small user story, and ensuring testability.]

Validate by implementing a “steel thread”: a small user story
Implement a part of Arch Feature
Architectural Feature
Facilitate Flow with an Architectural Kanban

- Kanban systems enhance the visibility of workflow
- Limits excess WIP, enhances throughput
- Supports realistic forecasting

**WIP Limits**

**FUNNEL**
All ideas are welcome

**REVIEW**
Roughly prioritized epics

**ANALYSIS**
Deeper analysis; Go/no-go decision

**BACKLOG**
Prioritized and ready for implementation

**IMPLEMENTING**
Agile Release Trains collaboratively implement epics
SAFe realizes a set of connected Kanban systems.
Evolve Architecture to Facilitate Flow

Reduce the gap with contemporary design paradigms

- Domain-Driven Design
- Micro-services architecture
- SOA
- Good-old, use-case oriented OO: SOLID, CCP, SDP, CRP

Features can then be contained …

**Orthogonal…**

**Aligned…**
Learn more…

- SAFe website: www.scaledagileframework.com
  - Agile Architecture
  - Architectural Runway
  - Architecture Kanban System
  - System/Enterprise Architect

- Preview of the next version:
  www.preview.scaledagileframework.com

- Enabling Flow and Agility with Architecture in SAFe:
  www.youtube.com/watch?v=yz3x4i8LYFQ

- Contact: alex.yakyma@scaledagile.com
Questions?