Opportunities for Project Managers in the Lean-Agile Enterprise with SAFe®

Richard Knaster
SAFe Fellow, Principal Consultant

Dr. Steve Mayner
SPCT, Senior Consultant
Agenda for today’s discussion

1. Why SAFe?
2. Brief overview of SAFe
3. Traditional vs. Agile mindset
4. How SAFe handles PMI core processes
5. Opportunities for Project Managers
6. Learn more
Online quick poll

Rate your knowledge of SAFe on a scale of 1-5.
1. Why SAFe?
The challenges we face today... apply to both traditional and Agile delivery

- Late delivery
- Lack of transparency
- Poor quality and late discovery of problems
- Phase gates don’t reduce risk
- Dissatisfied customers
- Too many dependencies
- Too early selection of designs
- Too much complexity
- No systematic improvement
- Poor morale
Enterprises can’t keep pace

Our methods must keep pace with an increasingly complex world.

- Business innovation is increasingly being delivered via software and systems, and companies must respond quickly to challenges and opportunities.
- Agile has produced better business outcomes, but it was developed for small, collocated teams.
- Agile methods alone don’t address the top-down concerns of business strategy, as Agile teams work bottom-up.
In an “Adapt or Die” business climate

“Since 2000, 52 percent of the names on the Fortune 500 list are gone, either as a result of mergers, acquisitions or bankruptcies.”

— Teresa Novellino | New York Business Journal

From the article “Don't get cozy, Fortune 500: it's do-or-die time for digital disruption” Click to view the article
2. Brief overview of SAFe
What is SAFe?

SAFe® is a freely revealed knowledge base of integrated, proven patterns for enterprise Lean-Agile development.

Core Values

1. Built-In Quality
2. Program execution
3. Alignment
4. Transparency

scaledagileframework.com
Embraces Lean-Agile values

House of Lean

VALUE

Respect for people and culture
Flow
Innovation
Relentless improvement

LEADERSHIP

Value in the shortest sustainable lead time

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.*
Has nine principles that make it effective

#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
## Is it “SAFe”?

<table>
<thead>
<tr>
<th><strong>US Fortune 100 Enterprises</strong></th>
<th><strong>Global Fortune 500 Enterprises</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>70% have SAFe-trained practitioners</td>
<td>10% have SAFe-trained practitioners</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SAFe-trained practitioners</strong></th>
<th><strong>SPCs in countries</strong></th>
<th><strong>Increase in productivity</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>120,000 in 105 countries</td>
<td>5,500 in 56 countries</td>
<td>30 - 75% Faster time to market: SAFe Case Studies</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Scaled Agile Partners</strong></th>
<th><strong>Annual page views</strong></th>
<th><strong>Social Media Followers</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>in 35 countries</td>
<td>12 million across SAFe &amp; Scaled Agile websites</td>
<td>25,000</td>
</tr>
<tr>
<td></td>
<td>1.3 million websites users</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>PLEDGED</strong></th>
<th><strong>ANNUAL</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1% Scaled Agile Stock Equity &amp; Employee Time to Pledge 1% Campaign</td>
<td><strong>SAFe SUMMIT</strong></td>
</tr>
</tbody>
</table>

**PLEDGED** 1%
Scaled Agile Stock Equity & Employee Time to Pledge 1% Campaign

**ANNUAL**

- 25,000 Social Media Followers
- 12 million annual page views across SAFe & Scaled Agile websites
- 1.3 million websites users
3. Traditional vs. Agile mindset
# From traditional to Lean-Agile governance

<table>
<thead>
<tr>
<th>Leadership Model</th>
<th><strong>Traditional governance</strong></th>
<th><strong>Lean-Agile governance</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Centralized command and control, transactional leadership</td>
<td>Self-organizing, decentralized, empowered; servant leadership</td>
</tr>
<tr>
<td>Work Planning</td>
<td>Detailed project plans</td>
<td>Backlog of prioritized Epics, Roadmap for 3 – 4 PIs, continuous flow of value</td>
</tr>
<tr>
<td>Funding</td>
<td>Project based funding</td>
<td>Value Stream funding of Epics delivered via ARTs</td>
</tr>
<tr>
<td>Resource Allocation</td>
<td>Multitasked, people brought to the work</td>
<td>Dedicated teams, work brought to the teams</td>
</tr>
<tr>
<td>Progress Tracking</td>
<td>Document deliverables at waterfall (variable) milestones</td>
<td>Working Features/Capabilities predictably delivered every PI</td>
</tr>
<tr>
<td>Architecture</td>
<td>Committed up front in design docs; locked in long-term early in the cone of uncertainty</td>
<td>Just enough runway for near-term Features; preserves options to pivot based on validated learning</td>
</tr>
<tr>
<td>Quality</td>
<td>Inspected at the end when final product is delivered</td>
<td>Built in incrementally, with rapid feedback cycles and auto testing</td>
</tr>
</tbody>
</table>
Fix the date, float the scope

The plan creates cost/schedule estimates

The Vision drives feature intent and estimates

- Agile Teams show that dates matter and they meet their commitments
- Business Owners understand how priorities matter
- Fix time and quality, not scope
From traditional Project Manager to Servant Leader

Move away from ...  
- Coordinating individual contributions  
- Acting as a subject matter expert  
- Driving toward specific outcomes  
- Knowing the answer  
- Directing  
- Talking about deadlines and technical options  
- Driving the “right” (your) decisions  
- Fixing problems rather than helping others fix them

Move toward ...  
- Coaching the whole team to collaborate  
- Being a facilitator  
- Being invested in the team’s overall performance  
- Asking the team for the answer  
- Letting the team find their own way  
- Guiding  
- Focusing on business value delivery  
- Doing the right thing for the business right now  
- Facilitating team problem-solving

Lyssa Adkins, *Coaching Agile Teams*
4. How SAFe handles Five PMI core knowledge areas
Scope management

“Future product development tasks can’t be predetermined. Distribute planning and control to those who can understand and react to the end results.”

—Michael Kennedy, Product Development for the Lean Enterprise
## Scope management—from traditional to Lean-Agile mindsets

<table>
<thead>
<tr>
<th>From traditional approach</th>
<th>To SAFe’s Lean-Agile approach</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scope Definition</strong></td>
<td>Vision and Roadmap</td>
</tr>
<tr>
<td><strong>Detailed project plans and WBS</strong></td>
<td>Agile estimating and planning</td>
</tr>
<tr>
<td><strong>Scope Change Control</strong></td>
<td>Frequent backlog refinement and appropriate decentralized content authority</td>
</tr>
<tr>
<td><strong>Centralized annual planning</strong></td>
<td>Decentralized, rolling-wave planning</td>
</tr>
<tr>
<td><strong>Project Charter</strong></td>
<td>Agile Release Train Canvas</td>
</tr>
<tr>
<td><strong>Scope Verification</strong></td>
<td>Acceptance Criteria and TDD and ATDD</td>
</tr>
<tr>
<td><strong>Scope Monitoring</strong></td>
<td>Objective, fact-based measures and milestones (e.g. working software)</td>
</tr>
</tbody>
</table>
Set the direction with the Vision and Roadmap

Vision

A long view:

- How will our future solution solve the larger customer problems?
- How will it differentiate us?
- What is the future context within which our solutions will operate?
- What is our current business context, and how must we evolve to meet this future state?

A three-PI rolling Roadmap

Program Increment = a time box (8-12 weeks)

PI 3
- Road Rage ported (part I)
- Brickyard port started
- Distributed platform demo
- ALL GUIs for both games demonstrable
- Multiuser architecture
- New Road Rage features (see objectives for details)
- New Brickyard features (see objectives for details)

PI 4
- E3 Expo Tradeshow!
- Road Rage completed (single user)
- Brickyard Ported (single user)
- Road Rage multiuser demonstrable
- First multiuser game feature for Road Rage

PI 5
- Road Rage (multiuser) first release
- Brickyard ported multiuser demo
- New features for both games (see backlog)

Committed
Forecast

© 2017 Scaled Agile, Inc. All Rights Reserved.
Plan face-to-face with the entire program (ART)

- All stakeholders face-to-face (but typically include multiple locations)
- Management sets the mission, with minimum possible constraints
- Requirements and design emerge
- Important stakeholder decisions are accelerated
- Teams create—and take responsibility for—plans

For a short video PI planning example, see: https://youtu.be/ZZAtl7nAB1M
Manage scope with decentralized content authority

- Epics are large initiatives that largely replace projects
- Capabilities describe the higher level behaviors of a Solution
- Features are services that fulfill user needs
- User stories are statements of intent
Provide continuous objective evidence

Base milestones on objective evaluation of working systems.

Solution Demos

Team and integrated System Demos
Time management

“A predictable organization does not guess about the future and call it a plan; it develops the capacity to rapidly respond to the future as it unfolds.”

—Mary and Tom Poppendieck, Implementing Lean Software Development
# Time management—from traditional to Lean-Agile mindsets

<table>
<thead>
<tr>
<th>From traditional approach</th>
<th>To SAFe’s Lean-Agile approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>R&amp;D determines the schedule based on a fixed scope</td>
<td>The business determines the schedule based on market needs. R&amp;D determines how much work can be done within a fixed timebox.</td>
</tr>
<tr>
<td>Milestones mainly driven by critical path and phase gates</td>
<td>Fixed length iterations and Program Increments</td>
</tr>
<tr>
<td>Dependencies are used to determine the critical path</td>
<td>Program Board is used to manage dependencies between teams. Cross-functional teams and trains help avoid unnecessary dependencies and delays.</td>
</tr>
</tbody>
</table>
Manage time with cadence and synchronization

**Cadence**
- Transforms unpredictable events into predictable events
- Makes waiting times predictable
- Facilitates planning; provides more efficient use of resources

**Synchronization**
- Synchronization causes multiple events to happen at the same time
- Sync events facilitate cross-functional tradeoffs of people and scope

Program Increment (8-12 weeks)

Program Increment

Do

Check

Plan

Adjust

ART PDCA

Team PDCA
Cost management

“Agile software development and traditional cost accounting don’t match”

—Rami Sirkia and Maarit Laanti
## Cost management—from traditional to Lean-Agile mindsets

<table>
<thead>
<tr>
<th>From traditional approach</th>
<th>To SAFe’s Lean-Agile approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>The plan drives resource needs and costs</td>
<td>Lean-Agile Budgeting. Fund Value Streams not projects.</td>
</tr>
<tr>
<td>Focus on near 100% resource utilization to optimize people cost</td>
<td>Focus on throughput of value and reducing costly delays</td>
</tr>
<tr>
<td>Develop a cost baseline upfront for the entire project</td>
<td>Fixed cost baseline for a Program Increment (PI). No variances analysis needed.</td>
</tr>
<tr>
<td>Focus on cost compliance to upfront plan</td>
<td>Focus on adapting to customer needs and developing minimal viable product</td>
</tr>
<tr>
<td>Earned Value Management</td>
<td>Burn-up charts</td>
</tr>
<tr>
<td>Manage cost through preventing change or change control</td>
<td>Welcome changes. Manage cost by working on most valuable work first.</td>
</tr>
</tbody>
</table>
Empowerment and governance with Lean-Agile budgeting

**TRADITIONAL**
- Projects are the basic unit of work
- People are “brought to the work”
- Measure compliance to inherently uncertain work

**RESULT**
- High overhead
- Us vs. them
- Low throughput, productivity
- Low morale

**LEAN**

**Governance**
- Exercise fiscal governance with dynamic budgeting
- Approve Epic-level initiatives

**Empowerment**
- Fund Value Streams, not projects
- Empower Value Stream content authority
- Work brought to the teams

**RESULT**
- Full control of total spend
- No budgetary surprises
- Higher throughput
- Higher morale

Beyond project cost accounting with SAFe
Provide objective evidence of fitness for purpose

© 2017 Scaled Agile, Inc. All Rights Reserved.
Control costs with increased flexibility

ART budgets and resources are unaffected by feature cost overruns or changing priorities.

Planned: Feature 1 | Feature 2

Actual: Feature 1 | Feature 2

Keep people working on the right feature for the right reasons

Delay this feature as necessary

Fixed cost per PI
SAFe provides fiscal governance with dynamic budgeting

Financial governance is still in place. Adjust budgets dynamically to meet changing business needs.
“Inspection does not improve the quality, nor guarantee quality. Inspection is too late. The quality, good or bad, is already in the product.”

“You cannot inspect quality into a product.”

—Harold F. Dodge

Source: https://blog.deming.org/2012/11/inspection-is-too-late-the-quality-good-or-bad-is-already-in-the-product/
## Quality management—from traditional to Lean-Agile mindsets

<table>
<thead>
<tr>
<th>From traditional approach</th>
<th>To SAFe’s Lean-Agile approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality is inspected at the end when final product is delivered</td>
<td>Quality is built in incrementally, with rapid feedback cycles and automated testing</td>
</tr>
<tr>
<td>Testers and developers work in separate silos</td>
<td>Testers and developers are part of cross-functional agile teams</td>
</tr>
<tr>
<td>Audit quality for compliance</td>
<td>Conduct team and system demos. Continuous automated testing.</td>
</tr>
<tr>
<td>Focus on compliance to original requirements</td>
<td>Focus on adapting solution to customer needs. Monitor using automated tests.</td>
</tr>
<tr>
<td>Lessons learned at end of project</td>
<td>Retrospectives every 2 weeks. Improvements made continuously</td>
</tr>
</tbody>
</table>
Built-in Quality

“*You can’t scale crappy code*” (or hardware, or anything else)

- Ensures that every increment of the solution reflects quality standards
- Quality is built-in, not inspected in, after the fact
- Software quality practices include continuous integration, test-first, refactoring, pair-work, collective ownership and more
- Hardware quality is supported by exploratory, early iterations, frequent system level integration, design verification, modeling and set-based design
Risk management

“The biggest risk is not taking any risk. In a world that's changing really quickly, the only strategy that is guaranteed to fail is not taking risks.”

—Mark Zuckerberg, Founder, Facebook

Source: BrainyQuote
## Risk management—from Traditional to Lean-Agile mindsets

<table>
<thead>
<tr>
<th>From traditional approach</th>
<th>To SAFe’s Lean-Agile approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upfront risk identification, quantitative analysis</td>
<td>Identify risks and dependencies during PI planning</td>
</tr>
<tr>
<td>Risk response planning</td>
<td>ROAM risks</td>
</tr>
<tr>
<td>Risk monitoring and controlling</td>
<td>Scrum of Scrums, daily stand-ups, big visible information radiators, Program Board to manage dependencies</td>
</tr>
<tr>
<td>Project Manager own risks.</td>
<td>Team members own the risks. Risks are reduced with each incremental delivery of working solutions.</td>
</tr>
</tbody>
</table>
ROAMing risks during PI Planning

After all plans have been presented, remaining program risks and impediments are discussed and categorized.

ROAMing risks:

- **Resolved** – Has been addressed; no longer a concern
- **Owned** – Someone has taken responsibility
- **Accepted** – Nothing more can be done. If risk occurs, release may be compromised.
- **Mitigated** – Team has plan to adjust as necessary
Building incrementally accelerates value delivery
Organize for faster learning and reduced risk

Agile Release Trains are cross functional and have all the people they need to define and deliver value every two weeks. This reduces hand-offs and delays and improves the reliability of commitments.

Agile Teams power the train:
Cross-functional teams apply Scrum, XP and Kanban and built-in quality practices to produce quality working system increments every iteration.
5. Opportunities for Project Managers

“Learning is not compulsory; neither is survival.”

—W. Edwards Deming
Leverage your existing project management skills

- Leadership
- Humility
- Effective Listening
- Team Building
- Motivation

- Communication
- Collaboration and knowledge sharing
- Influencing
- Managing Conflict
- Decision making
- Political and cultural awareness
- Negotiation

Source: Software Extension to the PMBOK 5th edition
Apply them to critical team roles

**Scrum Master** facilitates team events, drives Agile behavior and coaches the team.

**Product Owner** acts as the customer for team and prioritizes their work. Defines and accepts requirements.

**Development Team** is everyone needed to define, build, and test an increment of value. Developer, Tester, Business Analyst, etc.
Release Train Engineer acts as the chief Scrum Master for the train

Product Management is responsible for customer needs. Owns the vision and product backlog, prioritizes features for optimum economics of the system.

System Architect/Engineering align ARTs to a common technological and architectural vision

Solution Management responsible customer needs across ARTS, owns the value stream backlog, prioritizes capabilities for the optimal economics of the solution

Value Stream Engineer is a servant leader and coach for a value stream. Facilitates value stream events and activities.
SAFe is proven by dozens of real-world case studies
Including the public sector

Agencies / programs *known* to be using SAFe:

- FAA
- US Customs
- US Courts
- DOJ
- VA
- NGA
- CIA
- FBI
- SSA
- NHS
- NNSA
- USPTO
- US Post Office
- US Immigration
- Dutch Tax Authority
- French National Employment Agency
- And more!
Why? SAFe gets better business results

- Happier, more motivated employees
- 30 – 75% faster time to market
- 20 – 50% increase in productivity
- 50%+ defect reduction

See ScaledAgileFramework.com/case-studies
Online quick poll

Which statement best describes why you attended the webinar today?

- [ ]
- [ ]
- [ ]
- [ ]
- [ ]
Online quick poll

Which topic interests you the most for a future webinar?
6. Learn more
Gain the knowledge

Find SAFe training worldwide at:
ScaledAgile.com

Leading SAFe® 4.0
with SA Certification

Implementing SAFe® 4.0
with SPC4 Certification

Explore the SAFe knowledge base and find free resources at:
ScaledAgileFramework.com
Gain the knowledge

http://www.scaledagile.com/safe-whitepaper/
SAFe Release Train Engineer course and certification

- Effectively perform the role of the Release Train Engineer
- Apply Lean-Agile knowledge and tools to execute and release value through Agile Release Trains and value streams
- Support value delivery in large and complex value streams
- Facilitate activities that foster relentless improvement in the Agile Release Train
- Build a high-performing Agile Release Train by becoming a servant leader and coach
- Continue your learning journey and develop an ART action plan
Product Owner attributes

Focused on delivering stories and enablers to the train, tasked with helping the team *build the right things at the right time*.

- Ability to communicate
- Good business sense
- Technical foundation
- Trust
- Courage
- Content authority
Product Manager attributes

Focused on the business aspects and the market at large, tasked with building the right features at the right time.

- Sense of balance
- Forward thinking
- Solid understanding of current solution
- Trust
Release Train Engineer attributes

The Release Train Engineer is a servant leader who facilitates and guides the work of the ART:

- Empathetic and good listener
- Gravitas to create an environment of mutual influence and trust
- Interested in developing people
- Uses influence rather than authority to persuade
- Systems thinker
- Supports the commitments made by the teams without providing task direction