Scaling Lean-Agile to Large, Complex Systems

Harry Koehnemann
Director of Technology
321Gang
harry@321gang.com
Systems Engineering Challenges

- Increasing complexity
- Rapid reduction in cycle times
- Risk meeting customer/market needs
- Products in a continuous release cycle
- Solutions cross organizational boundaries (Systems of Systems)
- System-wide collaboration demands (BOF vs. BOM)
- Increased product variation
- Compliance - contractual, regulatory
SAFe Lean-Agile principles

#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
Development occurs in an uncertain world

Aggressively evaluate alternatives. Converge specifications and solution set.
—Allen Ward

- You cannot possibly know everything at the start
- Requirements must be flexible to make economic design choices
- Designs must be flexible to support changing requirements
- Preservation of options improves economic results
Apply fast learning cycles

Product development is the process of converting uncertainty to knowledge
—Dantar P. Oosterwal

Integration points control product development

- Integration points accelerate learning
- Development can proceed no faster than the slowest learning loop
- Improvement comes through synchronization of design loops and faster learning cycles

The Lean Machine: How Harley Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development
—Dantar P. Oosterwal
Align everyone on a common cadence

Waterfall, driven by early decisions and fixed schedule

Iterative, driven by learning and adaptive roadmap

Aligned on a common, regular cadence
Base milestones on objective evaluation of working systems

Regularly assess progress of product, compliance, and process towards delivery
1) Organizing around value

1. More focused teams
2. Fewer handoffs, delays, waiting
3. Easier to build in quality
4. Optimizing the system as a whole

Result: Faster delivery, higher quality, higher customer satisfaction

50-125 practitioners of multiple disciplines.
Value Streams may deliver to other VVSs

- Aligned on a common cadence
- Get comfortable with *Collective Ownership*
- Requires *Continuous Integration*
- Leverage *Community of Practices / Scrum of Scrums*
Large value streams require multiple ARTs

- Organize around capabilities or subsystems
- Each train may include multiple engineering disciplines
Ensure EVERYONE is on the value stream

- Reduced waste – waiting, delays, hand offs, batch sizes, WIP
- Ensure commitment to increment goals

Cross-functional Teams
Product Manager (Chief Eng)
Technical Authorities (Sys Eng)
Customer

Quality  Safety  Suppliers  Regulatory
Take a Systems View

- SAFe takes an end-to-end view of our development system
- Intent drives a solution for a customer within a context
- Built by a Value Stream aligned on a common cadence
- Intent, Solution, and Solution Context all have variability
Solution May Require a Context

- Supports Suppliers and Systems-of-Systems
- Customer continuously collaborates on multiple dimensions
  - Content (backlog), technical, I&T, program/budget
2) Manage change

- Change managed is connected Kanban systems
- Leans traditional “boards”/CCBs with flow, WIP limits, capacity matching
- Centralizes strategy; decentralizes decisions
- Increases visibility into the flow
- Provides hierarchical content governance system
3) Build the solution incrementally

- Cadence supports natural Work-In-Process limits that foster learning cycles across the Value Stream
Embrace variability

- Allow specification to emerge based on knowledge
- Building what we already know, explore what we don’t know

Solution Context

Variable

Fixed

Solution Intent

Modeling  Trade study  Modeling
Simulation  Capability  Capability  Capability

©2012-2016 321 Gang
MBSE facilitates emergent specifications

- Generate specifications from models in Solution Intent
- Everyone contributes; everyone takes a systems view
- Ensures single source of truth, both as-is and to-be
Organize Solution Intent with Models (MBSE)

Communication
Impact analysis
Strategic reuse
Generating compliance documents
4) Build quality in

Assure every increment of the solution reflects quality standards

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

Hardware
- Exploratory early iterations
- Model Based Systems Engineering (MBSE)
- Set-Based Design
- Frequent, system-level integration
- Design verification
Support frequent integration and testing

- The goal of frequent integration and testing is to provide frequent feedback on newly built functionality.

- Trade-offs are inevitable in terms of:
  - Frequency of integration
  - Depth of integration
  - Fidelity of feedback
Solution Integration is a set of trade-offs

- Solution integration is realized as a combination of real subsystems and proxies of various complexity
- Interfaces supports the ability to grow level of fidelity (SIL → HIL)
- Maturity of infrastructure and practices lowers cost of integration over time
- Bring in manufacturing early; use true production facility
Trade-off example: Fidelity of feedback

The cost of integrating an actual subsystem can be too high while substituting it with a stub may provide no useful information. Every subsystem dictates a unique trade-off.
Include quality concerns in definition of done (DoD)

- Strive for lean process – attack delay, waste, WIP

- EVERYONE commits
  - Peer review
  - Deploy to end-to-end testing env
  - Quality review, approval
  - Customer signoff

- Baseline everything
- Submit compliance docs
- Regulatory sign off

- Generate compliance docs for status
- Assess progress (coverage, sign-off, etc.)
- Deploy to IV&V
5) Make progress visible

- Assess progress at each increment
- Evaluate progress of solution *and* progress towards compliance
- Otherwise, quality and certification are uncertain
Measure progress at each System Demo

**Epic Progress**

**Epic Burnup**

**PI Predictability**

**Epic DoD**

**Cumulative Flow**

**Program Board**
Adopt SAFe

- Map Value Streams …
- …Train Everyone…

<table>
<thead>
<tr>
<th>Courses</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leading SAFe (SA Certificate)</td>
<td>2</td>
</tr>
<tr>
<td>SAFe for Teams (SP Certificate)</td>
<td>2</td>
</tr>
<tr>
<td>SAFe SM (Certification/Orientation)</td>
<td>2 / ½</td>
</tr>
<tr>
<td>SAFe PO (Certification/Orientation)</td>
<td>2 / ½</td>
</tr>
<tr>
<td>SAFe PPM</td>
<td>2</td>
</tr>
<tr>
<td>Implementing SAFe (SPC Certification)</td>
<td>4</td>
</tr>
</tbody>
</table>

- … And Launch ARTs
Your homework assignment
THANK YOU

Harry Koehnemann
Director of Technology
harry@321gang.com