Applying the Scaled Agile Framework (SAFe) to Lean Systems Engineering

Harry Koehnemann
Director of Technology
321Gang
harry@321gang.com
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
Our Current System Cannot Address Challenges

- System requirements, design, and schedule “defined” up front
- Difficult to defer decisions; reluctance to provide detail
- Project success defined by executing plan vs. delivering value
- Slow value deliver - delays, WIP, handoffs
- Not aligned to deliver value
Where Does Agile Excel?

- Answer: Alignment and Collaboration
- But, only solves **team-level** alignment and collaboration

Principle of Alignment: There is more value created with overall alignment than local excellence -- Don Reinertsen

**Diagram:**
- Product Owner
- Program
- Design/Dev/Test
- Requirements
- Sprint Planning
- Sprint Backlog (Stories)
- 30-day Sprint
- Potentially Shippable Software
- Sprint Review (Demo)
- Sprint Retrospective
### Apply 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
Assume Variability; Preserve Options

**Aggressively evaluate alternatives.**
**Converge specifications and solution sets. — Allen Ward**

- You cannot possibly know everything at the start
- Requirements must be flexible to economic design choices
- Designs must be flexible to changing requirements
- Preservation of options improves economic results
Integrate and Test Frequently

“Integration points control product development”

- Integration points are pull events that accelerate learning
  - Routine communication
  - Reduce variation
  - Objective evaluation
- 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
Coordinate and Align Within and Across Layers

Content
- Solution Manager
- Product Manager
- Product Owner

Process/Coordination
- Value Stream Coordinator
- RTE
- Scrum Master

Technical
- Solution Arch/Eng
- System Arch/Eng
- System DevOps

Integration/Test
- Systems DevOps
- Team
Organize Around Value

- Value doesn’t recognize organizational or geographic boundaries
- Organize your people around your Value Streams

Value Streams deliver solutions
Programs Are NOT (necessarily) Value Streams

- Value Streams do not (necessarily) equate to contracts or programs
- Too much “people motion” leads to:
  - Unpredictable team performance
  - Lost productivity - *form-storm-norm-perform*
  - Limited reuse
  - Localized optimizations
  - No economies of scale

Don’t bring people to the work… …Bring work to the people

Program 1
Program 2
Program 3

Program 1
Program 2
Program 3
Use Value Streams to Deliver Product Variants

- Eliminate handoffs
- Organize around common product lines vs separate V.Ss
- Variants may be by feature or time (model year ‘17, ‘18)
Value Streams Deliver Solutions

- Solution may be a component or feature
- Fund ARTs and cross-stream initiatives, not projects
Solution May Require a Context

- Represents Suppliers and Systems or Systems
- Customer continuously collaborates on multiple dimensions
  - Content (backlog), technical, integration, program/budget
ARTs Build Value Stream’s Solution

- ARTs deliver each increment (and ideally sprint)
- Continually integrate solution at least each increment
- Value Stream roles coordinate – content, technical, I&T
Replace Gates with Cadence-Based Learning Cycles

Base milestones on objective evaluation of working system

- “Pull” event to integrate entire system
- Work may include simulations, models, experiments, etc.
Learning Moves Variable to Fixed

- Simultaneously learn what we know, discover what we don’t know
- Enablers create knowledge, decisions, and runway to build Features
- Decisions made with sufficient time to support feature building
- Accelerated by MBSE and Set-Based Design
Record Knowledge in System Intent

- Repository of collective system knowledge
- Single source of truth to communicate decisions
- Populated by results of Enabler work
- Facilitates impact analysis
- Supports regularity and contractual compliance
Use Models (MBSE) to Organize System Intent

- Communication
- Impact analysis
- Strategic reuse
- Source for generating compliance documents

Requirements Model
(functionality, constraints)

System Model
(structure, behavior, simulation, parametrics, allocations)

Domain Models
(designs, analysis, etc.)

System Model (structure, behavior, simulation, parametrics, allocations)

Domain Models
(designs, analysis, etc.)

MCAD
Thermal, ...
Software

©2012-2015 321 Gang
Set-Based Design preserves options to make the best economic decisions based on objective evidence.

**Set-Based**
- Multiple Design Options

**Point-Based**
- Single Option

*Realizes the Cone of Uncertainty*
- Too much to adjust

*No predictable stabilization*
Manage Change with Solution Kanban

- Organizes “boards”/CCBs
- Prioritized by Solution Management
- Reviewed by System Engineering (utilize System Intent)
- Analyzed by affected stakeholders
Connected Kanbans

- Backlogs contain centralized initiatives and local context
  - *Centralized strategy; decentralized decisions*
- Increased visibility into the flow
- Hierarchical content governance system
Change Requires Leadership

People are already doing their best; the problems are with the system.

Only management can change the system.

—W. Edwards Deming
### Acquire the Knowledge

<table>
<thead>
<tr>
<th>Implementing SAFe</th>
<th>Leading SAFe</th>
</tr>
</thead>
<tbody>
<tr>
<td>► Ingrain deep SAFe knowledge (SPCs)</td>
<td>► Develop Lean-Agile leaders</td>
</tr>
<tr>
<td>► Identify value streams; structure ARTs</td>
<td>► Organize and support ARTs</td>
</tr>
<tr>
<td>► Train others</td>
<td>► Implement Agile Portfolio</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supporting role-based curriculum</th>
<th>SAFe for Agile Teams &amp; ART Quick start</th>
</tr>
</thead>
<tbody>
<tr>
<td>► Product Manager/ Product Owner</td>
<td>► Organize and train Agile Teams</td>
</tr>
<tr>
<td>► Portfolio Management</td>
<td>► Start the Train</td>
</tr>
<tr>
<td>► Scrum Master *</td>
<td>► Plan and execute the first Program Increment</td>
</tr>
<tr>
<td>► Release Train Engineer *</td>
<td></td>
</tr>
</tbody>
</table>
THANK YOU

Harry Koehnemann
Director of Technology
harry@321gang.com
Rational Support For SAFe LSE