An Executive’s Guide to the Scaled Agile Framework

Al Shalloway
CEO, Net Objectives

Al Shalloway
CEO, Founder

alshall@NetObjectives.com
@AlShalloway

copyright 2014. Net Objectives, Inc. All Rights Reserved
Common Organizational Structure

Inspired by Dan North, BSC/ADP 2012
**Hierarchical**

**What they can manage**

Their people
- How busy they are
- Their “productivity”

The quality of work of their people

---

**The Nature of Our Work**
We Manage This Way

even though value flows this way

Hierarchical vs. Lean Management

<table>
<thead>
<tr>
<th>What they can manage</th>
<th>What they need to manage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Their people</td>
<td>Time-to-market</td>
</tr>
<tr>
<td>• How busy they are</td>
<td>Effects of upstream groups on their teams</td>
</tr>
<tr>
<td>• Their “productivity”</td>
<td>Effects of downstream groups on their teams</td>
</tr>
<tr>
<td>The quality of work of their people</td>
<td></td>
</tr>
</tbody>
</table>
Who is managing the value?

Time-to-Market
How often does work wait?

What percent of the time is work moving forward?
How much of the time is it waiting for something else to be done?
How would you know?
No one is managing this in most companies.

What happens when adding value is delayed?

Between getting requirements and using them?
Between writing a bug and it being detected?
Between two groups getting out of sync?
1. Identify the actions taken in the value stream

- Request
- Approve
- Reqs
- Sign Off
- Analysis
- Design
- Review
- Code
- Test
- Deploy
1. Identify the actions taken in the value stream
2. What was the real time from start to finish of the action?

1. Identify the actions taken in the value stream
2. What was the real time from start to finish of the action?
3. What was the average time working on this vs working on other things?
1. Identify the actions taken in the value stream
2. What was the real time from start to finish of the action?
3. What was the average time working on this vs working on other things?

4. Identify time between actions

5. Identify any loop backs required
1. Identify the actions taken in the value stream
2. What was the real time from start to finish of the action?
3. What was the average time working on this vs working on other things?
4. Identify time between actions
5. Identify any loop backs required

**Calculate Process Cycle Efficiency:**

\[
PCE = \frac{\text{Avg Time Worked}}{\text{Total Cycle Time}}
\]

- Request: 0.5 / 0.0 hrs
- Approve: 120 hrs
- Reqs: 60 / 100 hrs
- Sign Off: 320 hrs
- Analysis: 40 / 60 hrs
- Design: 40 / 80 hrs
- Review: 2 / 0 hrs
- Code: 80 / 200 hrs
- Test: 40 / 200 hrs
- Deploy: 3 / 5 hrs

**Calculations:**

- **Total Worked:** 509 hrs
- **Total Cycle Time:** 3433 hrs
- **PCE:** 14.9%

**Delays:**
- Finding
- Redoing
- Reworking
- Waiting
- Bottlenecks
- Information delay
- Untested code
- Unread requirements
- Transaction related
- Coordination related

© Copyright Net Objectives, Inc. All Rights Reserved
Which gives a better return?

Getting better at what you do

Eliminating delays between what you do

high utilization

throughput

Here’s a spot!

...and another!

© Copyright Net Objectives, Inc. All Rights Reserved

Eliminating delays between what you do
Time Available for New Features

- Maximum capacity of the team
- Time to add new functionality
- Time lost to unused planning and estimation
- Time lost to interruptions due to changing requirements
- Time lost to thrashing on work between systems
- Time spent fixing bugs within and across systems

Years in future

Communication overhead as add more people

legacy organization:
matrix resources to projects

Project 1
Project 2
Project 3
Project 4
Project N
let’s create a pilot project

Experience has shown that if you create a cross-functional co-located team you will improve 3x without changing your process.

First Order Solution
Leaves Several Problems

How do we feed the teams?

How do we handle dependencies?

How do we handle product portfolio management?

What do we do with those that don’t quite fit the model?

How do we best integrate?

Lean Changes Our View

Tells us to look at the workflow, not the people

We can’t manage it if we don’t see it

Managing indirectly is not as powerful as managing it directly
The Ideal Scrum Solution

Why is this so hard to achieve?

Ignoring that it may be difficult to have one team build the entire product, there is still the problem of sharing particular people.
What We Need to Do

• Identify, size, sequence work
• Allocate our people to most important work
• Don’t start what you can’t finish
• Organize into teams and manage flow
• Avoid interruptions coming from outside of business drivers
• Avoid delays by:
  • managing work in process
  • automating testing
  • Keeping teams in cadence
  • Continuously integrating
  • Drive from business value

Scaled Agile Framework

A proven, publicly-facing framework for applying Lean and Agile practices at enterprise scale.
lean thinking provides the tools we need

The Goal: Value

- Respect for People
- Product Development Flow
- Kaizen

Foundation: Leadership

Goal: Speed, Value, Quality

Minimize delays, handoffs and non-value added activities

All we are doing is looking at the timeline, from the where the customer gives us an order to where we collect the cash. And we are reducing the time line by reducing the non-value added wastes.

– Taiichi Ohno

Agile is about delivering incremental business value, not team iterations

– Alan Chedalawada

Most software problems will exhibit themselves as a delay.

– Al Shalloway

THE GOAL

- Sustainably shortest lead time
- Best quality and value to people and society
- Most customer delight, lowest cost, high morale, safety
Provide quality systems within which people can work
**Optimize the Whole**

Optimize to realize business value... not just to improve development work.

Implement lean across
- an entire value stream
- the complete product

---

**Eliminate Waste**

Only work on things of value and that you know how to achieve.

Only start work that you know you can complete.

Not working on the most important things

Poor collaboration

Delays in workflow

Delays in feedback

Duplicate efforts

Re-learning

Technical Debt
product development flow

1. Take an economic view
2. Actively manage queues
3. Understand and exploit variability
4. Reduce batch sizes
5. Apply WIP constraints
6. Control flow under uncertainty: cadence and synchronization
7. Get feedback as fast as possible
8. Decentralize control

kaizen

BECOME RELENTLESS IN:
- Reflection
- Continuous improvement as an enterprise value

We can do better
- A constant sense of danger
- Small steady, improvements
- Consider data carefully, implement change rapidly
- Reflect at milestones to identify and improve shortcomings
- Use tools like retrospectives, root cause analysis, and value stream mapping
- Protect the knowledge base by developing stable personnel and careful succession systems
Teams work in cadence

Develop on cadence, deliver on demand
Agility

is about
Business Value Increments
not development cycles

A system is a network of interdependent components that work together to try to accomplish the aim of the system.
A system must have an aim. Without an aim, there is no system.

W. Edwards Deming
drive from the portfolio

Portfolio Vision gives the system an **aim**
Centralized strategy, decentralized execution
Investment themes provide operating budgets for trains
Kanban systems provide portfolio visibility and WIP limits
Objective metrics support governance and kaizen
Value description via **Business** and **Architectural** epics

alignment

more value is created with **overall alignment**
than with local excellence. – Don Reinertsen

Clear content authority
Face-to-face planning
Aligned Team, Program and Business Owner objectives
Cross-team and cross-program coordination
Architecture and UX guidance
Match demand to throughput
systems must be managed

A system must be managed. It will not manage itself. Left to themselves, components become selfish, competitive, independent profit centers, and thus destroy the system... The secret is cooperation between components toward the aim of the organization.

W. Edwards Deming

the program level drives the teams

Self-organizing, self-managing team-of-agile-teams
Continuous value delivery
Aligned to a common mission via a single backlog
Common sprint lengths and estimating
Face-to-face planning cadence for collaboration, alignment, synchronization, and assessment
Value description via Features and Benefits
alignment, synchronization, and cadence

Today's development processes typically deliver information asynchronously in large batches. Flow based processes deliver information in a regular cadence of small batches.

Don Reinertsen

Develop on Cadence, Deliver on Value

Development occurs on a fixed cadence. The business decides when value is released.
program execution

**Agile Release Trains** – self-organizing teams of agile teams – reliably and frequently deliver enterprise value

- Driven by vision and roadmap
- Lean, economic prioritization
- Frequent, quality deliveries
- Fast customer feedback
- Fixed, reliable cadence
- Regular inspect and adapt drives continuous improvement

nothing beats an agile team

- Empowered, self-organizing, self-managing cross-functional teams
- Valuable, fully-tested software increments every two weeks
- Scrum project management practices and XP-inspired technical practices
- Teams operate under program vision, system, architecture and user experience guidance
- Value description via **User Stories**
transparency

transparency builds confidence, alignment and trust

All backlogs and progress visible to all stakeholders

Objective reporting based on working, tested, evaluated code

Everyone understands backlog, capacity, velocity, WIP

Management leads and fosters open environment
leading knowledge workers

Workers are knowledge workers if they are more knowledgeable about the work they perform than their bosses.

Peter Drucker

Workers themselves are best placed to make decisions about how to perform their work and how to modify their processes. To effectively lead, the workers must be heard and respected.

Knowledge workers have to manage themselves. They have to have autonomy.

Continuing innovation has to be part of their work, and their responsibility

foundation: leadership

Lean Thinking Manager-Teachers

- Take responsibility for Lean-Agile success
- Understand and teach Lean-Agile behaviors
- Are trained in practices and tools of continuous improvement
- Teach problem solving and corrective action
- See with their own eyes. “No useful improvement was ever invented at a desk.”
- Managers develop people. People develop solutions.
Conclusion

The foundation of Lean is **leadership**
The foundation of SAFe is **YOU**

Next Steps

- **Become a SAFe Lean Thinking Manager-Teacher**
- **Launch Agile Release Trains**
- **Leverage the Community**
- **Leverage the Community**

Browse the framework
www.scaledagileframe.com

Read the book
Agile Software Requirements

Check out www.netobjectives.com/safe

Take a course
Leading SAFe in **Atlanta, Charlotte, Dallas, Milwaukee, Raleigh, San Jose, Seattle, Washington DC**

Accelerate value delivery with your first Agile Release Train

Contact Net Objectives (see next slide)
Join the community at community.ScaledAgile.com

Join the community at community.ScaledAgile.com
Net Objectives

Agile at Scale consulting and implementation for a decade.
The premier provider of SAFe-related consulting and training.
The primary contributors to the materials in the SAFe code quality section.

**Technical Training**
- Design Patterns
- ATDD / TDD
- Emergent Design
- Scrum/XP

**SAFe-Related Services**
- SPC Training
- Leading SAFe with Net Objectives Extensions
- Portfolio Management
- Product Manager and Product Owner Training
- SAFe Kanban
- SAFe consulting

---

Thank You

Al Shalloway
email: alshall@netobjectives.com
Twitter tag @alshalloway

Register at [www.netobjectives.com/register](http://www.netobjectives.com/register) to receive notices of monthly webinars
See [www.netobjectives.com/resources](http://www.netobjectives.com/resources)

See upcoming Leading SAFe courses at [www.netobjectives.com/events](http://www.netobjectives.com/events)