SCALING SOFTWARE AGILITY

Agile Portfolio and Program Management in the Scaled Agile Framework

Eight Transformational Patterns

About Dean Leffingwell

Author

- Agile Requirements
- Agile Software Requirements: Best Practices for Large Enterprises
- Scaling Software Agility

Coach

- Agile Enterprise Coach
- Agile Executive Mentor
- Chief Methodologist
- Cofounder/Advisor

Executive

- Founder and CEO, ProQuo, Inc.
- Senior VP, Rational Software
- Founder/CEO, REQA, Inc.
- Founder/CEO, RELA, Inc.
- Founder/CEO, Colorado MEDtech
Agenda

- Portfolio and Program Management Responsibilities
- Legacy Mindsets in Portfolio and Program Management
- The Scaled Agile Framework
- Eight Transformational Patterns
  - Rethinking Strategy and Investment Funding
  - Rethinking Program Management
  - Rethinking Governance
- Summary: The AGILE PPMO

PPMO Responsibilities

The PPMO has a central role in strategy and investment funding, program management and governance
Legacy Mindsets Handicaps Agility

- Historically, PMO practices were based on the waterfall model of development
  
  (We shouldn’t blame these folks for governing us the way we told them to back when!)

- But now, we are transforming the entire business to a leaner and more agile enterprise, with ever-faster delivery of valuable, high quality software

- These legacy mindsets have to change........(just like everything else did)

---

Legacy Mindsets

Strategy and Investment Funding

<table>
<thead>
<tr>
<th>Mindset</th>
<th>Manifestation</th>
<th>Problems</th>
</tr>
</thead>
</table>
| “we can plan out a full year of projects” | - Long term program commitments  
   - Teams must justify personnel a year out | - Plans are obsolete, but not treated that way  
   - Projects impossible to kill once started  
   - Everyone lives a lie |
| “Maximize utilization”        | - Resources committed long range  
   - 100% allocation before “what if”  
   - Key resources assigned to multiple projects | - No time to think or innovate  
   - Dedicate resources to task or lose resources  
   - Thrashing – productivity lost of most valuable resources  
   - No flex to changing priorities  
   - Exhaustion, burnout |

Source: Establishing an Agile Portfolio to Align IT Investments with Business Needs -- Thomas and Baker, DTE Energy, by DTE Energy - Implementing and extending agile practices since 1998
## Legacy Mindsets

### Program Management

<table>
<thead>
<tr>
<th>Mindset</th>
<th>Manifestation</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>“widget engineering”</td>
<td>▶ Fixed schedule, functionality planning&lt;br&gt;▶ Big Up Front Design/Analysis (BUFD)&lt;br&gt;▶ Detailed requirements specifications</td>
<td>▶ Long range detailed commitment&lt;br&gt;▶ Resources committed year in advance&lt;br&gt;▶ Analysis paralysis&lt;br&gt;▶ Specs are wrong, hard to change</td>
</tr>
<tr>
<td>“Get it done”</td>
<td>▶ Belief that best case plans must succeed</td>
<td>▶ Deferred recognition of plan vs. actual&lt;br&gt;▶ Late discovery and re-negotiation&lt;br&gt;▶ Extended risk profile&lt;br&gt;▶ Loss of credibility, mistrust</td>
</tr>
<tr>
<td>“order taker mentality”</td>
<td>▶ Do what you are told&lt;br&gt;▶ We are the boss of you</td>
<td>▶ False agreements. No buy in.&lt;br&gt;▶ Misses IT innovation contribution&lt;br&gt;▶ Failure to meet expectations –mistrust&lt;br&gt;▶ No empowerment, lower motivation</td>
</tr>
</tbody>
</table>

### Governance and Oversight

<table>
<thead>
<tr>
<th>Mindset</th>
<th>Manifestation</th>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Control through data”</td>
<td>▶ Fine grain reporting and overhead&lt;br&gt;▶ Detailed wbs, earned value metrics, loaded Gantt charts</td>
<td>▶ Reporting overhead slows value delivery&lt;br&gt;▶ Metrics don’t reflect actual progress</td>
</tr>
<tr>
<td>“Control through milestones”</td>
<td>▶ Milestone reporting on intermediate artifacts</td>
<td>▶ Milestones do not reflect actual progress&lt;br&gt;▶ Annoys the team “they just don’t get it”</td>
</tr>
</tbody>
</table>
Eight Transformational Patterns

What we need is a transformation “roadmap”, one that builds an Agile PPMO on Lean-Agile Principles

<table>
<thead>
<tr>
<th>Legacy Mindset</th>
<th>Lean-Agile Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 Too Many Projects</td>
<td>Limiting Work in Process</td>
</tr>
<tr>
<td>#2 Detailed Project Plans</td>
<td>Lightweight Business Cases</td>
</tr>
<tr>
<td>#3 Annual Funding</td>
<td>Incremental Funding</td>
</tr>
<tr>
<td>#4 Centralized Annual Planning</td>
<td>Decentralized Rolling-Wave Planning</td>
</tr>
<tr>
<td>#5 Work Breakdown Structure</td>
<td>Agile Estimating and Planning</td>
</tr>
<tr>
<td>#6 Projects</td>
<td>Agile Release Trains</td>
</tr>
<tr>
<td>#7 PMBOK</td>
<td>Agile Project Management</td>
</tr>
<tr>
<td>#8 Waterfall Milestones</td>
<td>Fact-Based Governance</td>
</tr>
</tbody>
</table>

Legacy PPMO  Agile PPMO

The Scaled Agile Framework
The Scaled Agile Framework

The Scaled Agile Framework is a proven, framework for applying Lean and Agile practices at enterprise scale

More on SAF:
- Synchronizes the vision, planning, interdependencies, and delivery of many teams
- Works well for teams of 50-100
- Has been scaled to over hundreds of teams and thousands of people
- Web version available to public in February 2012
- For more info, see [ScaledAgileFramework.com](http://ScaledAgileFramework.com)


© 2008-2011, Leffingwell, LLC. All rights reserved.
PPMO in the Framework

SAF provides lean operating models for Strategy and Investment Funding, Program Execution, and Governance

Governance
- Clear content authority
- Drives incremental delivery
- Fact-based (code, test, quality, customer feedback) lifecycle governance

Strategy & Investment Funding
- Limit Work in Process
- Economic Prioritization

Program Management
- Continuous value delivery
- Self-managing programs

Rethinking Strategy and Investment Funding
1. Too Many Projects
2. Detailed Project Plans
3. Annual Funding
4. Centralized Annual Planning
**#1. From Too Many Projects**

To Limiting Work in Process

**Too Many Projects**
- It is far easier to start projects than to complete them
- All dev teams have accepted (or been given) **more work than they can deliver**
- Result: excessive work in process, team members thrashing
- Lowers productivity (20% per switch) … causes delays in outcome…: Result: Even **more work is loaded** into the system
- Like a freeway, congestion results and throughput comes to a **halt**

**Controlling WIP**
- Development wip is invisible and has no natural predators
- Provide visual indicators of work in process
- Establish WIP limits for # projects in flight
- Build a Portfolio Management Kanban System

---

**Portfolio Kanban System**

Kanban systems manage the flow of value via visibility and Work in Process (WIP) limits

**What Kanban Systems do**
- Visually manages units of value through the whole system, from when they enter, until they leave
- Limiting WIP, creates an optimum, sustainable pipeline of value flow
- Limiting WIP provides visibility as to when there is capacity for new work
- WIP Limits are adjusted and their effects measured as the system is continuously improved

**Why we need one for the Portfolio**
- Adopt and drive agile thinking, language and behavior
- Drive incremental-ism in implementation of business epics
- Establish priorities based on economics
- Make work in process (WIP) visible
- Establish WIP limits to control queue sizes, limit global WIP and help assure continuous flow
- Drive effective collaboration with development teams
**Portfolio Kanban System**

**Solution Needs Identification**

1. **Funnel**
   - Product roadmap
   - New business opportunity
   - Cost savings
   - Solution problem

**Evaluation**

2. **Backlog**
   - Refine understanding
   - Est. cost of delay
   - Refine effort est.
   - Relative ranking

3. **Analysis**
   - Solution alternatives
   - Collaboration
     - Solution management
     - Architects
     - Market/sales/business
     - Development
   - Weighted rank
   - Business case

4. **Implementation**
   - Ownership transitions
   - Teams begin implementing at release planning boundaries
   - Teams break epics into features
   - Analyst support on “pull” basis

**Innovation feedback**

- Release planning boundary
- Agile Release Trains

**Activity:****
- Effort size estimate
- Value size estimate
- Investment theme alignment

**Authority approves epic**
- Meets threshold criteria

**Business analyst pulls Epic**
- Lead analyst assigned

**Portfolio Management Team/Product Council Approval**

**Agile Portfolio Management Tools**

- UK Division
- Team Blue

**Portfolio Status**

**Portfolio Items Board**

- Backlog
  - P1222: Auto-provisioning
  - P1222: Network security restrictions
  - P1234: Chinese localization

- Analysis
  - P12267: Real-time accounting reports

- Approval
  - P1232: CFR Part 11 compliance

- In Dev
  - P2828: Auto-provisioning

- Done
  - P1231: Japanese localization
  - P2828: Credit card payments
#2. From Detailed Project Plans

## To Lightweight Business Cases

<table>
<thead>
<tr>
<th>Detailed Project Plans</th>
<th>Lightweight Business Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detailed business case justifications become project plans</td>
<td>1-2 page business case form</td>
</tr>
<tr>
<td>False precision – detailed requirements over-constrain solution implementation</td>
<td>Not much detail</td>
</tr>
<tr>
<td>May contain redundant schedule, budget, ROI information</td>
<td>Business cases that make the cut get exploratory iterations funding</td>
</tr>
<tr>
<td>Investment in the business case causes resistant to changing the case and plan</td>
<td>Easily cancelled if progress not acceptable</td>
</tr>
<tr>
<td>Too much overhead for quarterly update</td>
<td>Fast ROI if it is</td>
</tr>
</tbody>
</table>

### Lightweight Business Case

<table>
<thead>
<tr>
<th>Epic Name</th>
<th>Go or NO Go Recommendation</th>
<th>Date entered</th>
<th>Analyst</th>
<th>Epic Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Changes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description of the Epic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stakeholders (Identifies key business sponsors who will be supporting the initiative)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Users and markets affected (Describe the user community of the solution and any markets affected)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Products, programs, in house or outsource development (Describe recommendations for where the epic is to be developed)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impact on sales, distribution, Estimated development timeline</td>
<td>Start Date:</td>
<td>Completion date: (Estimated calendar date or number of PSIs)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated investment</td>
<td>Incremental Implementation Strategy</td>
<td>(Breaks initiative down into preliminary epics or sub-epics that fit the companies PSI cadence)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weighted rating</td>
<td>Reevaluation check points (If the epic is large, identifies potential milestones or checkpoints for reevaluation)</td>
<td>(Brief summary of the analysis that has been formed to create the business case. Pointers to other data, feasibility studies, models, market analysis, etc. that was used on the creation of the business case)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis summary</td>
<td>Attachments</td>
<td>Project Stakeholder Needs Assessment (see Chapter 7)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>System Stakeholder Needs Assessment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#3. From Annual Funding

To Incremental Funding

### Annual All or None Funding
- Legacy model drives "all or none" funding.
- Once funds committed, they are virtually impossible to de-allocate.
- Commitments are included in the annual budget planning process.
- Departments must justify the use of resources **through** the fiscal year.
- So they create more detailed plans that justify the annual budget.
- Resulting projects are almost impossible to kill.
- This vicious cycle drives the opposite of the behavior we are trying to achieve.

### Agile – Incremental Funding
- Bad paths can be truncated more quickly.
- Resources can be moved to the best opportunities throughout the fiscal year.
- Base incremental funding on objective demonstrations of working software (instead of milestones based on proxy documentation).
- Continuous opportunities to assess and adjust.
- Teams have real motivation to deliver immediate value.

See "Beyond Budgeting". [www.bbrt.org](http://www.bbrt.org)

---

#4. From Centralized, Annual Planning

To Decentralized, Rolling-Wave Planning

### From: Gantt

**Issues:**
- Irrelevant to agile teams.
- Hard to maintain.
- Always obsolete.
- If a team isn’t on the plan, is it a “bad” team or “bad” plan?
- Measure **paper, not software**.

### To: Independent, asynchronous, Sprints

**Issues:**
- Most agile companies start here.
- Little coordination amongst teams.
- Non-harmonized schedules.
- No visibility beyond the next sprint.
- Little or no system level visibility.

### To: Agile Release Train

**Issues:**
- Coordinates sprints.
- Multi-sprint visibility and commitment.
- Teams work out interdependencies on the fly.
- Full system visibility.
- Requires set-based development.
Decentralized, Rolling-wave Planning

- Teams plan Face-to-Face on a fixed cadence
- Result: Agreed-to objectives
- Commitment from the team
- Updated roadmap: high confidence “Release Next”

Rethinking Program Management

5. WBS
6. Projects
7. PMBOK
#5. From WBS

## WBS Estimating and Planning
- Traditional project estimates tasks at the lowest leaf
- Requiring all leaves be identified before estimate is given
- Forces Big Up Front Analysis and estimates based on false precision
- Force fits later activities into a flawed WBS

## Agile Estimating and Planning
- Agile teams develop and monitor “velocity” based on story points at iteration level
- Story points can be normalized across teams
- Standardized estimating by analogous model can also be applied at the level of Epics and Features
- Epic estimating can be used for gross, portfolio-level planning
- Feature estimates can be used for release (product) level planning
- Story points used for iteration planning

### Velocity Chart
- Epic: 4,000 pts
- Feature: 500 pts
- Story: 8 pts

---

## Enterprise Backlog Estimation

- Portfolio estimates
- Program estimates
- Team estimates (all in story points)
#6. From Projects

## Everything is a Project
- Getting anything done (new feature or epic) requires creation of a new project
- Projects have significant overhead, planning, resourcing, execution, closure
- Once started, projects take on a life of their own. They develop antibodies to change and closure.
- Many small projects cause people to multiplex
  - Each project switch takes 20% overhead

## Continuous Content Delivery
- Dedicated teams stop multiplexing. No one works on more than one team. No team on more than one program.
- Project scheduling replaced by standard cadence
- New initiatives appear as new content: prioritized at each iteration/release boundary
- Work in a PSI is fixed
- Team resources are adjusted only at cadence boundaries

---

### Agile Release Trains

Investment Themes are realized by one or more “Agile Release Trains”

- Release trains are long lived value streams that align teams to a common mission
- Each provides continuous product development flow (individual project chartering not required)
Release Train Tracking

- Release burndown shows overall progress of release against release plan
- A standard agile report
- Automatically compiled from “time remaining” story updates in agile project management tooling
- Most valuable to program, release and executive managements

- Feature completeness report shows status of each release feature over time
- Automatically compiled from stories completed/stories remaining in agile project management tooling
- Most valuable to program and product managers

#7. From PMBOK

Traditional, performed by the Project Manager
- Work Breakdown Structure estimating
- Gantt Charts scheduling
- Critical Path analysis
- Reporting

To Agile Project Management
- Performed by the team
  - High priority feature
    - Story 1: 2 pts
    - Story 2: 4 pts
    - Story 3: 2 pts
  - Actual velocity based estimating
  - Iteration planning
  - Release planning and roadmap
  - Release/iteration review/retrospective
#8. From Waterfall Milestone

### Waterfall Milestones
- Teams report milestones with document-based reviews
- Subjective, milestone reports do not correlate to actual project status
- Teams “report to” project office (leader as conductor/boss)
- Teams cannot proceed until and unless they pass milestones (start-wait-start-wait waste cycle)
- Scheduling delays and overhead
- Process changes dictated by “those who know best”

### Fact-based
- Milestones are iterations and incremental releases of working code
- Status and quality are quantitative, not subjective
- Project office “comes to teams” (enabling leadership model)
- Teams default model is to proceed unless stopped by business case (no process driven delays/waste)
- No scheduling delays and overhead
- Process changes applied and harvested from “those who do”
Governance – Content Authority

Scaled product owner role

Governance – Agile Milestones

- All development occurs in PSI increments
- Agile milestones drive and support incremental delivery
- Quality and release governance standards apply to all releases

Program charter/Release train approval

Incremental Releases

Quality and release governance criteria

Potentially shippable Increments
The Agile PPMO

Mission: Enable, foster, and empower enterprise agility for business results

- Limiting Work in Process
- Lightweight business cases
- Incremental funding
- Decentralized rolling wave planning
- Agile estimating and planning
- Agile Release Trains
- Agile Project Management

The Agile PPMO enables, fosters, and promotes lean and agile practices across the enterprise
Sources and More About

- Sliger and Broderick. 2008. *The Software Project Manager’s Bridge to Agility*
- Krebs, Jochen. 2008. *Agile Portfolio Management*
- Agile Project Management Leadership Network
  [http://apln.org](http://apln.org)
  DTE Energy, Proceedings of Agile 2008