Agile Transformation at Scale
Gail Ferreira, PhD, CSP, SPC, LSSBB, PMP
Lean Agile Coach & Practice Leader
drgail.Ferreira@gmail.com
Agile Project Management
Agenda

- Background
- The Culture of Change
- Building Innovative Products
- Transformation at Scale
“...the ability to both create and respond to change in order to profit in a turbulent business environment.”

- Jim Highsmith
The Culture of Change

1. **Organizational structure** is about how you create teams and organize them.

2. **Agile practice** is about the methods and tools you choose to introduce.

3. **People and culture** is about changing the hearts and minds of the organization.

- All three aspects are essential to sustain agility of any kind within the organization.
The Culture of Transformational Change

Agile Adoption is about the ‘Agile Doing’ side of the equation.

Transformation is about changing the ‘Agile Being’ side of the equation.

*Long term results* require both adoption and transformation to be successful.

*Culture is the #1 Challenge with Agile Transformation.*
Transformation at Scale

Culture + Infrastructure

= Agile Transformation at Scale
Three main challenges in scaling teams:

**Coordinating** work across teams

**Integrating** work across teams

**Maintaining** technical integrity of the system
Scaling Approaches

Each of the popular scaling approaches offers a certain value proposition, focus, options for implementation, cost implications and other attributes

• Scrum of Scrums (SoS)
• Large Scale Scrum (LeSS) - Larman/Vodde
• Scaled Agile Framework (SAFe) - Leffingwell
• Spotify “model” (Tribes, Squads, Chapters & Guilds) – Kniberg
• Scrum at Scale – Sutherland/Brown
• Disciplined Agile Delivery (DAD) + Agility at Scale Ambler/Lines
**Scrum of Scrums**

*Scrum of scrums* is a technique used to scale Scrum up to large groups (over a dozen people), consisting of dividing the groups into Agile teams of 5-10. Each daily scrum within a sub-team ends by designating one member as "ambassador" to participate in a daily meeting with ambassadors from other teams, called the Scrum of Scrums.
Insights – LeSS (Large Scale Scrum)

*Craig Larman characterizes LeSS as:*

an organizational design based on ten LeSS Principles:

1. Large Scale Scrum is Scrum
2. Transparency
3. More with Less
4. Whole Product Focus
5. Customer Centric
6. Continuous Improvement
7. Lean Thinking
8. Systems Thinking
9. Empirical Process Control
10. Queueing Theory
Example

Requirement Area A

Requirement Area B

Requirement Area C

Requirement Area D

Overall Product Owner

large-scale Scrum framework 1

up to 5 or 10 teams
Insights – SAFE

... Scaled Agile Framework

- Is applicable whenever at least a few hundred software practitioners are working cooperatively on related products and solutions
- Has generated great amounts of interest from enterprises
- Is becoming supported by a large number of Agile Tools

- The SAFe “big picture” has 3 levels: portfolio, program, team
- Relatively more prescriptive, pragmatic
- More controversial
- Evolving, growing, doing more types of training
- Release Trains
Spotify Model
Insights – Spotify

... Spotify offers a culture-centric approach to Scaling Agile

- Squads have end-to-end autonomy over their products
- Loosely coupled, tightly aligned
- Infrastructure
- Client Applications
- Features

- Squads = Scrum Teams
- Chapters = Competency Areas
- Tribes = Lightweight matrix of squads and chapters
- Guilds = Communities of Practice
- Self-service, Open source model
- Focus on enabling each other
- Release Train
Insights – Scrum at Scale

Reproducible Patterns

1. Modularity allows versatility.
2. Scrum is modular.
3. Deploying incrementally is modular.
4. Modularity supports a pattern library.
Insights – DAD

DAD seeks to extend Scrum for enterprise scale challenges

- People-first, learning-oriented hybrid agile approach
- Risk-value delivery lifecycle and goal driven
- Agility at scale is about explicitly addressing the challenges teams face in the real world
- Promotes Enterprise Awareness
- Key Differentiator – explicitly recognizes that Agile teams are governed
DAD supports a robust set of roles

- Team Lead
- Product Owner
- Architecture Owner
- Team Member
- Stakeholder
Concept: the Agile 3C rhythm

The coordinate-collaborate-conclude rhythm occurs at several levels on a disciplined agile delivery (DAD) project:

<table>
<thead>
<tr>
<th>Release rhythm</th>
<th>Inception</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration rhythm</td>
<td>Inception</td>
<td>Construction</td>
<td>Transition</td>
</tr>
<tr>
<td>Daily rhythm</td>
<td>Inception</td>
<td>Construction</td>
<td>Transition</td>
</tr>
</tbody>
</table>
DAD Teams Are Enterprise Aware

Disciplined Agilists: Work closely with enterprise groups

Follow existing roadmaps where appropriate

Leverage existing assets

Enhance existing assets

Enterprise Awareness
“How can I help my organization?”

Departmental Awareness
“How can I help my department?”

Team Awareness
“How can I help the team?”

Individual Awareness
“How can I be the best me?”
Transformation at Scale

A Step by Step approach towards Enterprise Agile Adoption

- Define Basic Agile Model for the organization.
- Agree on a high level adoption roadmap
- Identify Agile practices and prioritize
- Identify Pilots and assess for risks

- Initial pilots Kick-off
- Introduce Agile best practices to the teams
- Train and Mentor for several sprints
- Refine Agile model based on the learning/feedback
- Identify Additional pilots
- Create roadmap

Projects should be evaluated to determine their suitability for Agile.
Proposed steps for Rollout – Principles and Practices

• Ensure Executive buy-in

• Implement Scrum as a program / product management framework

• Introduce Continuous Engineering practices

• Small, incremental rollout is proposed

• Identify pilot product to implement the agile principles and practices
Pilot Rollout & Trainings

- Agile and Scrum trainings for the entire team

- Developers

- Testers

- Product Owner / Product Manager

- Scrum Master(s)

- Requirements writing (Stories, Use Cases)

- Estimation Techniques
  - Relative Estimation
  - Story points
  - Planning Poker
Alignment

• Enable Customer Participation
• Provide Vision
• Resolve Organizational Impediments & Risks
• Provide Resources
• Resolve Priority disputes
• Product Owner able to represent decisions made with Stakeholders
• Stakeholders voice is heard, differences resolved, vision clarified, priorities and value established
• Represent the Customer and Business needs
• Use Change Management
• Deliver Value
• Partner w/Business

Executive Steering

Chief Product Owner

Product Definition Office

Chief BA
Chief UX
Deploy Lead
PMO Rep
POs

Support Functions
- Architect
- Business Analyst
- Development
- QA
- Deployment

Value Stream

Enterprise Backlog

Executive Backlog

Product Backlog

Product Owner

Product Owner

Product Owner

Business Stakeholder

Business Stakeholder

Business Stakeholder

Business Stakeholder

Business Stakeholder

Business Stakeholder

Business Stakeholder

Business Stakeholder

Agile Teams

Agile Teams

Agile Teams

Agile Teams
Establish Communication & Collaboration Mechanisms

Keys to Effective Distributed Delivery
- Onshore & Offshore Tech Leads
- Offshore Customer Representatives
- Periodic travel rotation for offshore resources
- Good Agile PM tools
- Automated Build/Continuous Integration
- Overlapping hours and daily standups
- Leveraging technology maximally
- Leverage Coaching & Common Training
- Whole teams offshore
- Common Planning
Leverage Continuous Integration & Automation

**Continuous Integration**
- Developer
- Source Code
- Source Code Control System
- Continuous Integration Service
- Source Code Build & Test

**Test First Development**
1. Write Test
2. Run Test
3. See Success
4. Run Test
5. See Failure

**Automation**
- Automated Builds
- Regression Tests
- Functional Tests
- System/Performance Tests
Measuring Success
How do we measure success of Agile teams?

<table>
<thead>
<tr>
<th>Process Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Maturity Index</td>
<td>Agile Maturity Index at the end of every Sprint</td>
</tr>
<tr>
<td>Customer Satisfaction</td>
<td>Customer rating at the end of every Sprint</td>
</tr>
<tr>
<td>Team Satisfaction</td>
<td>Team satisfaction rating at the end of every Sprint</td>
</tr>
<tr>
<td>Retrospective Action Items completed</td>
<td>% of Retrospective items completed</td>
</tr>
<tr>
<td>Automation capability</td>
<td>% of Automation scripts / test scripts</td>
</tr>
<tr>
<td>Requirement Change on Sprint – Discipline</td>
<td>% of Requirement change within the Sprints</td>
</tr>
</tbody>
</table>
# Key Agile Metrics – Using IT Balanced Scorecard

<table>
<thead>
<tr>
<th>Balanced Scorecard</th>
<th>Strategic Objectives</th>
<th>Performance Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial Compliance</td>
<td>Improve Budget Performance</td>
<td>1. Time to Value&lt;br&gt;2. Cost of Value&lt;br&gt;3. Earned value / release</td>
</tr>
<tr>
<td>Processes</td>
<td>Project Delivery</td>
<td>1. Velocity&lt;br&gt;2. Avg project cycle time&lt;br&gt;3. Schedule performance index</td>
</tr>
<tr>
<td></td>
<td>Decrease Defects</td>
<td>1. Defects&lt;br&gt;2. Unit Test Coverage&lt;br&gt;3. System Test Coverage&lt;br&gt;4. UAT Defects Found&lt;br&gt;5. Running tested features</td>
</tr>
<tr>
<td>Learning and Growth</td>
<td>Enable Agile Transformation</td>
<td>1. Increase Agile Utilization.&lt;br&gt;2. Build community of practice.</td>
</tr>
</tbody>
</table>
### Sample Metrics

<table>
<thead>
<tr>
<th>Process Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity</td>
<td>Actual Number of story points achieved in an iteration</td>
</tr>
<tr>
<td>Drag factor</td>
<td>Actual Effort (unplanned) / Capacity</td>
</tr>
<tr>
<td>Iteration Defects</td>
<td>No. of defects at the Iteration end / LOC</td>
</tr>
<tr>
<td>Story de-scoping index</td>
<td>Stories descoped / Stories planned</td>
</tr>
<tr>
<td>Effort deviation</td>
<td>(Actual effort – Planned effort) *100 / Planned effort</td>
</tr>
<tr>
<td>Build success rate</td>
<td>Builds passed / Total # of builds</td>
</tr>
<tr>
<td>Automation &amp; Test Coverage</td>
<td>Unit test coverage, test case execution, etc</td>
</tr>
</tbody>
</table>

### Balanced Scorecard

- **Customer Value**
- **Customer Satisfaction**
- **Automated Test Results**
- **Code Quality**
- **Learning & Growth**
- **Financial Compliance**

### Burn Down & Burn Up Charts
**Agile PMO Distinctions**

1. Define progress in terms of business value. Focus on results instead of effort.
2. Burn up charts providing actual progress, cost incurred, and value achieved.
3. Meaningful measurement of the code assets
4. Automation for reduced burden and increased accuracy
5. Conduit for clearing impediments, ensuring collaboration, getting resources
6. Ensure alignment to strategic goals and value.

**Balanced Scorecard**
- Financial Compliance
- Customer Satisfaction
- Automated Test Results
- User Scenarios
- Code Quality

**Code Quality**
- Extent of duplication
- Cyclomatic complexity
- Presence of large methods
- Code encapsulation

**Qualification via Automation**
- Unit Testing
- Integration Testing
- Functional Testing
- Automated regression
- Maximum Coverage
- Non-auto Exploratory

**Outcome based status as slices of user functionality.**

**Continuous Integration**

**Automation**

**Integrated Measurement**

**Actual Progress (Value Delivered)**

= Ship it!

**User Stories Delivered**

**Time (Iterations)**

**Projected**

**Actual**
Case Studies
Single Team / Single Product

Sub 25 person product company and a start-up

• Started with team level practices
• Lots of attention early to team culture
• Began engaging leaders on strategy and portfolio management
• Currently integrating marketing, sales, and support

Methods:
Scrum of Scrums, Spotify
Multi-Team / Single Product

Sub 100 person product company. 10 years old and privately owned.

- Program level first ...established a PO team
- Three tightly integrated Scrum teams
- Defined the portfolio governance layer
- Established the relationship between strategy and support
- Modeled the overall value stream and wrapped up the Scrum process in a two-tiered Kanban

Methods:
LeSS, Scrum of Scrums, Spotify
Multi-Team / Multi-Product

Large multi-national organization. Scope is a 500 person development organization with 55 Scrum teams.

- Started with a basic view of the portfolio layer
- Portfolio level value stream mapping, RACI
- Built out the program management layer with PO teams to develop a requirements management capability
- Program level value stream mapping, RACI, introduced agile tooling
- Introduced Scrum at the team level

Methods:
SAFe, Scrum at Scale, LeSS, DAD
Product of Products


- Scrum teams by product / component.
- Product owner teams established.
- Portfolio level governance model.
- Lean/TOC planning model.
- Integration with a traditional PMO for metrics and monitoring.

Methods:
SAFe, Scrum at Scale, LeSS, DAD
Questions
A Few Good Resources ...
Links

Disciplined Agile Delivery (DAD) + Agility at Scale – Ambler/Lines
http://disciplinedagiledelivery.com

Large Scale Scrum (LeSS) – Larman / Vodde

Scaled Agile Framework (SAFe) – Leffingwell
http://scaledagileframework.com

Scrum at Scale
http://www.scruminc.com

Spotify Model
https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/
HyperGrowth Done Right - Lessons from the Man who Scaled Dropbox and Facebook


Wisdom from Hypergrowth Companies

http://www.startuplessonslearned.com/2013/10/wisdom-from-hyper-growth-companies.html
<table>
<thead>
<tr>
<th>Strategy Map</th>
<th>Strategic Objectives</th>
<th>Performance Measures</th>
<th>Targets</th>
<th>Initiatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial Compliance</td>
<td>Improve Project Budget Performance</td>
<td>1. Margin Contribution (or Revenue, or Cost, etc) 2. Earned Value/Release 3. Avg CPI – Cost Performance Index 4. Time to Value</td>
<td>1. TBD 2. TBD 3. 1 Month</td>
<td>• Measure IT contribution to Revenue  • Enable operations to release value every month to production  • Reduce cycle time  • Use business measures</td>
</tr>
<tr>
<td>Customer Value</td>
<td>Increase External CSAT</td>
<td>1. Customer Satisfaction –CSAT</td>
<td>&gt;6/qtr</td>
<td>• Implement CSAT for all branches and end users of software</td>
</tr>
<tr>
<td>Internal Process</td>
<td>Increase accuracy of Estimates</td>
<td>1. E0/E1 Estimate Variance Delta</td>
<td>&lt;20%/spr</td>
<td>• Measure and monitor estimates every sprint, take active role in reducing variance</td>
</tr>
<tr>
<td>On Time Project</td>
<td>On Time Project Delivery</td>
<td>1. Velocity &amp; Std. Deviation 2. Drag factor 3. Avg Project Request Cycle Time – Requested 4. Avg Project Request Cycle Time - Scheduled 5. Sprint &amp; Release Burndown performance trend deviation 6. Customer Satisfaction –CSAT 7. Avg SPI – Schedule Performance Index</td>
<td>1. Varies/spr 2. &lt;5%/spr 3. #prr/days 4. #prs/spr 5. &lt;10% 6. &gt;6/qtr 7. TBD</td>
<td>• Ensure operational measures are collected and collated every sprint, by every sprint team.  • Ensure sprint and release burndowns are posted daily (hours for teams, story points for everyone)  • Ensure CSAT surveys are created and part of goals  • PM to help teams track SPI during the release</td>
</tr>
<tr>
<td>Decrease Defect RFCs</td>
<td>Decrease Defect RFCs in Production</td>
<td>1. Defects/Story Point 2. Unit Test Coverage 3. System Test Coverage 4. UAT Defects Found 5. Running Tested Features</td>
<td>1. 0 2. &gt;90% 3. &gt;90% 4. 0 5. &gt;90%</td>
<td>• Ensure consistent defect management across CRP and INC test teams.  • Measure coverage every sprint  • Measure running tested features</td>
</tr>
<tr>
<td>Increase Audit</td>
<td>Increase Audit Compliance %</td>
<td>1. PQA Compliance</td>
<td>&gt;90%</td>
<td>• Educate teams about compliance, SM to help ensure compliance</td>
</tr>
<tr>
<td>% of agreed scope</td>
<td>Increase % of agreed scope</td>
<td>1. Story de-scoping index 2. User Story Volatility</td>
<td>1. &lt;10% 2. &lt;5%</td>
<td>• IT and Business measure needed. Indication of partnership. Ensure collection and review.</td>
</tr>
<tr>
<td>Learning and Growth</td>
<td>Enable Agile Transformation</td>
<td>1. Increase Agile Utilization</td>
<td>&gt;25%/qtr</td>
<td>• Transformation team proactively working the roadmap to work toward 100% (of goal) participation.</td>
</tr>
<tr>
<td>Enable People &amp;</td>
<td>Enable People &amp; Culture</td>
<td>1. Increase Ace Participation ((people/events)/month)</td>
<td>&gt;33%</td>
<td>• Define and formalize the ACE program and participation opportunities and guidelines.</td>
</tr>
<tr>
<td>Measurement</td>
<td>Frequency</td>
<td>Target</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-----------</td>
<td>--------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>E0/E1 EV Delta</td>
<td>Sprint</td>
<td>&lt;20%</td>
<td>(E1 – E0/E1)*100</td>
<td></td>
</tr>
<tr>
<td>Velocity &amp; Std. Deviation</td>
<td>Sprint, Release</td>
<td>varies</td>
<td>varies by team</td>
<td></td>
</tr>
<tr>
<td>Drag factor</td>
<td>Sprint, Release</td>
<td>&lt;5%</td>
<td>(Actual Effort in story pts unplanned / velocity ) days, measure end to end time from the time the request was submitted by the business to the time it goes live in production. Alternatively, can measure the &quot;request to scheduled&quot; cycle time and that can be added to the &quot;scheduled to production&quot; cycle time (below). Value range will vary by request type (maint, s/m/l project), but also want average of all.</td>
<td></td>
</tr>
<tr>
<td>Avg Project Request Cycle Time - Requested</td>
<td>Sprint, Release</td>
<td>varies</td>
<td>days, measure end to end time from the time the request was scheduled by the business to the time it goes live in production. Value range will vary by request type (maint, s/m/l project), but also want average of all.</td>
<td></td>
</tr>
<tr>
<td>Avg Project Request Cycle Time - Scheduled</td>
<td>Sprint, Release</td>
<td>varies</td>
<td>days, measure end to end time from the time the request was scheduled by the business to the time it goes live in production. Value range will vary by request type (maint, s/m/l project), but also want average of all.</td>
<td></td>
</tr>
<tr>
<td>Sprint &amp; Release Burndown</td>
<td>Daily</td>
<td>varies</td>
<td>varies by team</td>
<td></td>
</tr>
<tr>
<td>Customer Satisfaction - CSAT</td>
<td>Sprint</td>
<td>&gt;6</td>
<td>Implement for all groups (dev, support, biz functions, etc). Low CSAT indicate less collaboration and potential increase in cycle time and/or duration. Also apps to Scope</td>
<td></td>
</tr>
<tr>
<td>Defects/Story Point</td>
<td>Sprint</td>
<td>varies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit Test Coverage</td>
<td>Sprint</td>
<td>&gt;90%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Test Coverage</td>
<td>Sprint</td>
<td>&gt;90%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UAT Defects Found</td>
<td>Sprint</td>
<td>0</td>
<td>Harder target due to Agile structure and PO involvement. New code going through UAT should have 0 defects.</td>
<td></td>
</tr>
<tr>
<td>Running Tested Features</td>
<td>Sprint</td>
<td>100%</td>
<td>Running Tested Features. Cumulative number of tests that are running for tested (non-broken) features. Drives performance to ensure running code and tested code. Should increase steadily on from the first sprint w/tests.</td>
<td></td>
</tr>
<tr>
<td>Story de-scoping index</td>
<td>Sprint</td>
<td>&gt;10%</td>
<td>((Story pts Descoped + Story pts Added + Story pts Modified)/ Story pts Originally Planned) * 100. This is important for helping track business activities in relation to changing scope. Scope can slip due to development issues, but it also can slip do to a lack of focus and prioritization and changing mindsets on the part of the business. Creates accountability on both sides of the house.</td>
<td></td>
</tr>
<tr>
<td>User Story Volatility</td>
<td></td>
<td>&lt;5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Earned Value</td>
<td>Sprint</td>
<td>varies</td>
<td>Sum Total of earned value of stories completed. The pre-requisite of this approach is to have story values assigned as part of the demand and prioritization process. Should be tracked by the business.</td>
<td></td>
</tr>
<tr>
<td>Agile Utilization</td>
<td>Month</td>
<td>as per roadmap</td>
<td>Deliver competitive business value, cost optimization, improved compliance, and strategic business alignment. (% Agile hrs/% total hrs)</td>
<td></td>
</tr>
<tr>
<td>Get Involved - ACE Participation</td>
<td>Sprint</td>
<td>&gt;30%</td>
<td>(% of people involved in some initiative)</td>
<td></td>
</tr>
</tbody>
</table>