Agile & Lean Program Management: Scaling Collaboration Across the Organization

Johanna Rothman
Agile and Lean Program Management: Scaling Collaboration Across the Organization
@johannarothman
www.jrothman.com
781-641-4046
What Worked?

• Clear program vision

• Clear deliverables

• Feature teams who delivered at least every day

• Unit tests and system tests as support

• Respect and trust across the organization
Program Management

- Organizing and coordinating several projects’ results into one deliverable.
- That deliverable has the value to the organization.
Programs Are Riskier Than Projects

- Projects don’t scale linearly
- The larger and the longer the program, the more risky it is
- The more pieces the program has, the more risky it is
Programs are Complex or Complicated

- **Complex**
  - Unknown unknowns
  - probe-sense-respond
  - Emergent Practice

- **Complicated**
  - Known unknowns
  - sense-analyse-respond
  - Good Practice

- **Chaotic**
  - Unknowable unknowns
  - act-sense-respond
  - Novel Practice

- **Obvious**
  - Known knowns
  - sense-categorise-respond
  - Good Practice
Agile Team Collaboration

The team produces shippable product on a regular basis
Scale from One Team to Many Teams?

• What do you need to scale collaboration from one team to multiple teams?

• Hint: you do not need a framework
“Scaling” Agile requires:
Autonomy
Collaboration
Exploration
Use Sense-Making to Assist the Program

- Where do you have unknowns?
- What can you do to move from unknown to known?
- What principles work?
- Think and experiment:
  - Probe, sense, respond
Organize the Teams

• Any form of agile or lean works for the project teams
  
  • Small batch size and continuous integration

• As long as the team delivers, you don’t care how they organize themselves

• Evolving architecture

• Plan to replan
Collaborate Across the Organization

• Transparency of progress
• Transparent and pervasive communication
• Problem-solving autonomy
• Hierarchies slow everything down
• Information has to flow freely around the organization
What’s the Most Effective Way to Move Information In Your Organization?
Rumor Mill
Small World Networks

• Small world networks are more-and-less connected agile teams

• How connected are you to everyone else?
  • Some of you are highly connected
  • Some less so

• Programs take advantage of the network so they don’t need hierarchies
Use Small-World Networks

• Feature teams take responsibility

• Small-world networks provide autonomy and collaboration

• Communities of practice help with exploration
Roadmaps & Backlogs Provide Team Autonomy

- Roadmap is the wish list
- Backlog is what teams will do
- Rank everything by value
- Sometimes learning about risk is valuable
# Agile Roadmap

**Agile Roadmap for a Product: Several Quarters Out**

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Release Tulip</td>
<td>External Release Daisy</td>
<td>External Release Rose</td>
<td>External Release Carnation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Int. Release 1</th>
<th>Int. Release 2</th>
<th>Int. Release 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
</tbody>
</table>
# One Quarter Perspective

## Agile Roadmap for a Product: This Quarter

<table>
<thead>
<tr>
<th>Internal Release 1</th>
<th>Internal Release 2</th>
<th>Internal Release 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
<tr>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
<td>Feature Sets/Themes</td>
</tr>
</tbody>
</table>

![Diagram](image.png)

- **MVP for release**
- **Story**
- **...**

Rothman
Change is Key

- The more often you release, the more successful the product will be
- Fast feedback all over the program
- Inspect and adapt
- Value-based approach to product development
Lean is Essential

• See the flow of work all over the organization
• Manage WIP
• Create knowledge
• See the whole
Collaborate with Tests & Integration

• Create a program environment where people can collaborate
  • Tests and CI help teams collaborate
• Safe places to explore
• Autonomous teams
  • Deliver features often
Agile Changes Program Management

- Teams (not managers) manage:
  - Commitment
  - How they build features
  - Evolve architecture
  - Program management
  - Removes obstacles
  - Collects and explains program status
Core Program Team Manages Risks Across the Organization
Software Program Team Manages Obstacles for Feature Teams
Seeing Program Progress

• Program level measures!

• Working product is the best measure
  • Do not even think about using team velocities and merging them

• Consider product backlog burnup
Product Backlog Burnup

Product Backlog Burnup Chart

Cumulative Features

Iteration or interim date
Release Frequency, Costs & Decisions

Potential for Release Frequency

<table>
<thead>
<tr>
<th>Software as a Service</th>
<th>Boxed Software</th>
<th>Product with Firmware</th>
<th>Software with Hardware or Mechanical components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous Deployment:</td>
<td>Often:</td>
<td>Less Often:</td>
<td></td>
</tr>
<tr>
<td>As often as several</td>
<td>But the cost</td>
<td>The cost of release</td>
<td></td>
</tr>
<tr>
<td>times a day</td>
<td>of release</td>
<td>is high</td>
<td></td>
</tr>
<tr>
<td></td>
<td>is still high</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Infrequently:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Every release might be a major release</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architectural decisions and releasing costs less</td>
<td>Architectural decisions and releasing costs more</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last responsible moment is later</td>
<td>Last responsible moment is earlier</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Agile and Lean Program Management Scales

- Trust teams to do their work (autonomy)
- Release often (to see progress and invite collaboration)
- Frequent releases encourage exploration
- It’s agile and lean all the way down
Stay in Touch?

- Many articles on jrothman.com and Pragmatic Manager email newsletter
- Look for Agile and Lean Program Management: Scaling Collaboration Across the Organization (in beta)