Scaling Kanban

Using Kanban to create and manage product development flow at the program/portfolio level

Yuval Yeret
Kanban Lead - AgileSparks
Lean/Kanban @ Israel
A: Scaling+Kanban
What is the Question?
Geoffrey Moore’s - Crossing the Chasm

Technology Adoption Lifecycle

Innovators
Early Adopters
Early Majority
Late Majority
Laggards

"The Chasm"

Area under the curve represents number of customers
We Are HERE

Geoffrey Moore’s - Crossing the Chasm

@yuvalyeret
## Early Majority Pragmatic Enterprises

<table>
<thead>
<tr>
<th>Risk Averse</th>
<th>Allergic To Dogma</th>
<th>Expect Whole Solutions</th>
<th>Complex Environments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lots of Legacy/Debt</td>
</tr>
</tbody>
</table>

@yuvalyeret
So Lean/Agile approaches as of 2011 need to...

<table>
<thead>
<tr>
<th>Help manage Risk</th>
<th>Be context-Specific</th>
<th>Develop a complete picture</th>
<th>Answer issues of Scale and Legacy Without requiring day 0 Overhauls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not be huge Risks themselves</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Some enterprise challenges we encounter

“Doing agile at the team level is fine. But how do we deal with the things we cannot get into the team?”
Q: How do we benefit from Lean/Agile principles At Scale?
Step #1: identify and work with minimally valuable features
Limit the size of work - at all levels
Visibility – Monitor and control the size of work
Step #2: Visualize Flow
Tool: The Cumulative Flow Diagram
How to do a CFD
How to do a CFD
What can teams learn from Cumulative Flow?

![Diagram showing Cumulative Flow metrics such as Total Scope, Dev Burnup, Done Burnup, Real Done Burnup, Work Process (WIP), and Average Cycle Time.]
Which is BETTER?
Team A

Team B

Team C

Now What?
Sum \((A+B+C)\)
Program-level Aggregate CFD
Visualize Overall Progress
Nice reports are not enough!
Can you tell what these represent?
Some “smells” we see out there

<table>
<thead>
<tr>
<th>Longer Sprints</th>
<th>Sprint Synchronized Handoffs</th>
<th>Agile Teams Waterfall Release</th>
</tr>
</thead>
</table>

@yuvalyeret
<table>
<thead>
<tr>
<th>Product Level</th>
<th>Comp A</th>
<th>Comp B</th>
<th>Comp C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backlog Impact on Components</td>
<td>Features in A</td>
<td>Features in B</td>
<td>Features in C</td>
</tr>
<tr>
<td>Features in A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features in B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features in C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrative Features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Level</td>
<td>Comp A</td>
<td>Comp B</td>
<td>Comp C</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>Backlog Impact on Components</td>
<td>Team A</td>
<td>Team B</td>
<td>Team C</td>
</tr>
<tr>
<td>Features in A</td>
<td>Team A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features in B</td>
<td></td>
<td>Team B</td>
<td></td>
</tr>
<tr>
<td>Features in C</td>
<td></td>
<td></td>
<td>Team C</td>
</tr>
</tbody>
</table>
| Integrative Features | | | ????????????
Learn about Feature Teams

Scaling Lean & Agile Development
Thinking and Organizational Tools for Large-Scale Scrum
Craig Larman
Bas Vodde

Practices for Scaling Lean & Agile Development
Large, Multisite, and Offshore Product Development with Large-Scale Scrum
Craig Larman
Bas Vodde

@yuvalyeret
Now – Let's Scale!
Use a Program-level Feature Kanban Board
Swimming Lanes per Product
Features and Stories hierarchy
Projects Portfolio for a large IT shop
<table>
<thead>
<tr>
<th>Backlog</th>
<th>Discovery/Analysis</th>
<th>Doing</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y4</td>
<td>Z3</td>
<td>Y3</td>
<td>Y1</td>
</tr>
<tr>
<td>Z4</td>
<td>X3</td>
<td>X4</td>
<td>Z2</td>
</tr>
<tr>
<td>X1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Or... Use Colors

@yuvalyeret
Learn more about Kanban
That was EASY
What about...
<table>
<thead>
<tr>
<th>Backlog</th>
<th>Discovery/Analysis</th>
<th>Doing</th>
<th>Implementation</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td></td>
<td></td>
<td>X1</td>
<td></td>
</tr>
<tr>
<td>X3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y4</td>
<td></td>
<td></td>
<td>X2</td>
<td></td>
</tr>
<tr>
<td>Z3</td>
<td></td>
<td>Y2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X+Y+Z</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Z1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Card with color #FF6C00 example**
- **Card with color #F3A4E5 example**
- **Card with color #FF00FF example**
- **Card with color #F2F2F2 example**
# Portfolio/Program Level

## Backlog Impact on Products

<table>
<thead>
<tr>
<th>Features in Z</th>
<th>Comp A</th>
<th>Comp B</th>
<th>Comp C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features in Z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features in B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features in C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrative Features</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Features in Y</th>
<th>Comp A</th>
<th>Comp B</th>
<th>Comp C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features in A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features in B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features in C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrative Features</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Features in X</th>
<th>Comp A</th>
<th>Comp B</th>
<th>Comp C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features in A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features in B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Features in C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrative Features</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Cross-Product Features

[Diagram showing cross-product features]
“We don’t yet have the political power or resolve to create real Feature Teams end to end. So what do we do?”
WIP / Cycle Time / Management Overhead – Less is Better…

- Product Component Teams w/ Iteration Handoffs between them
- Product Component Teams w/ Flow
- Product Feature Teams w/ Iteration Handoffs between them
- Product Teams w/ Flow
- Full Cross-Product Feature Team
Best – Cross-Product Feature Team / Task Force
Let them organize around the work
Limit Task Forces in Progress...
What does Versatility mean at Scale?

Try... Reducing the number of different skill sets, and generalizing some specializations
60 seconds on Other Possible System Designs
“The Pipeline”
No dependencies – Expand/Collapse
Dependency “Networks”?
What about Coordination at Scale?
Anyone using Scrum of Scrum?

<table>
<thead>
<tr>
<th>15 minutes</th>
<th>As needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Each participant answers four questions:</td>
<td>Resolve problems and discuss issues on the team backlog.</td>
</tr>
<tr>
<td>1. What has your team done since we last met?</td>
<td>No personal names</td>
</tr>
<tr>
<td>2. What will your team do before we meet again?</td>
<td></td>
</tr>
<tr>
<td>3. Is anything slowing your team down on in their way?</td>
<td></td>
</tr>
<tr>
<td>4. Are you about to put anything in another team’s way?</td>
<td></td>
</tr>
</tbody>
</table>

@yuvalyeret
Questions about Scrum of Scrum

• Are we doing it because we have Scrum+Scale? What problem are we trying to solve?

• Is Scrum of Scrum used for coordinating 100% of the work, or just the exceptions?
Coordination at scale
Each participant answers four questions:
1. What has your team done since we last met?
2. What will your team do before we meet again?
3. Is anything slowing your team down on in their way?
4. Are you about to put anything in another team’s way?

As needed
Resolve problems and discuss issues on the team backlog.
How about Improving at Scale?

- How many of you can name a process improvement they are trying this month?
Early Process Feedback and Adjustment

• Lean/Kanban - Improvement happens as part of flow

• Retrospectives/Kaizen Events are still important
Try... Operational Reviews

Action Items from last review

<table>
<thead>
<tr>
<th>Action Item</th>
<th>Start Date</th>
<th>Status</th>
<th>Comments</th>
</tr>
</thead>
</table>

WIP CFD
- Size Based
- No of Cards

CFD

Control Chart (calendar time, no size)

MMF Cycle Time Control Chart (based on size)

Analysis – What we learn from this
- Some examples:
  - MPPs are growing larger
  - What to do? Try to break MPPs
- Some very high cycle times on specific features
  - Why? due to lack of collective ownership and waiting for one part of the team
  - Try pairing. Get others taking stories with the guidance, etc.
  - Fill talent matrix and take action accordingly with the team.

Experiments/Interventions

@yuvalyeret
Try... an Improvement Service

Scaling Lean & Agile Development
Thinking and Organizational Tools for Large-Scale Scrum
Craig Larman
Bas Vodde

Practices for Scaling Lean & Agile Development
Large, Multisite, and Offshore Product Development with Large-Scale Scrum
Craig Larman
Bas Vodde
<table>
<thead>
<tr>
<th>Idea</th>
<th>Votes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improvement</td>
<td>1</td>
</tr>
<tr>
<td>Improvement</td>
<td>Yuval</td>
</tr>
<tr>
<td>Improvement</td>
<td>Erez</td>
</tr>
<tr>
<td>Improvement</td>
<td>Elad</td>
</tr>
<tr>
<td>Improvement</td>
<td>Elad</td>
</tr>
<tr>
<td>Improvement</td>
<td>1</td>
</tr>
</tbody>
</table>

Pull Improvement Needs from Teams
Will visualizing and managing flow in each of those roads be enough?
Need to visualize and manage the **global** end to end **flow** across **shared** resources.
Sharing

http://www.flickr.com/photos/kwl/4809326028/
http://www.flickr.com/photos/justin_case/54087334/in/photostream/
Manage flow at the level at which resources are shared
### Example Policy

**Classes of Service for Downstream Involvement**

#### Risk Profiling ➔ Involvement mode of the shared resource

---

**@yuvalyeret**

---

#### Classes of Treatment

<table>
<thead>
<tr>
<th>Next</th>
<th>Sprint Backlog</th>
<th>Dev + Integ.</th>
<th>Test</th>
<th>Done</th>
<th>Integration/System/Load Testing Team</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Next</td>
<td>WIP</td>
<td>WIP</td>
<td>WIP</td>
<td>WIP</td>
</tr>
<tr>
<td>S1</td>
<td>Intercept calls by DB bad guys list</td>
<td>Intercept all calls (simple copy)</td>
<td>Send intercepted calls to server (HTTP)</td>
<td>Intercept calls for xml bad guy list</td>
<td>Intercept calls for hardcoded bad guys list</td>
</tr>
<tr>
<td></td>
<td>Monitor if interception server is alive</td>
<td>Change interception server by xml</td>
<td></td>
<td></td>
<td>Send intercepted calls to local folder</td>
</tr>
</tbody>
</table>

---

@yuvalyeret
What about Due Dates and Commitment?

Does Flow mean no commitment?

What do you mean tell the customers to go Agile? Come on! Be REAL!
REAL reason to for Due Dates?
Use Fixed Date class of service

@yuvalyeret
No REAL external Due Date?

try Internal Cycle Time Soft Goals
based on learned capabilities

Avoid local due dates!

Try... Monitoring Aging + Stage
What does it mean to Manage WIP at Scale?

Projects/Features in certain high attention Class of Service Fixed Date / Cost / etc.

Projects

Cross-Product Features/Task Forces

Shared Resources
Scaling Kanban

Guidance for real enterprises

Pragmatic approach to change

Early Feedback, Learning, Managed Delivery – At Scale

@yuvalyeret
yuval@agilesparks.com
@yuvalyeret
twitter

Get the slides at 🌐
http://www.slideshare.net/yyeret

Blogging at
http://yuvalyeret.com