Supplemental Materials for

*Agile and Lean Program Management: Scaling Collaboration Across the Organization*

All contents copyright Johanna Rothman. Provided for your reading pleasure. If you want to use an image or wording, please contact me, jr at jrothman dot com.

Figure 2.1: Cynefin Framework

Figure 2.2: Large Program, One Coherent Product

Figure 2.3: Inter-Related Product Program
Figure 2.4: Integrated System Product Program

Figure 2.5: What Your Core Team Might Look Like
Figure 2.6: What Your Software Program Team Might Look Like

Figure 3.1: Different Teams

Figure 4.1: Agile Program Charter Template

Agile Program Charter Template: Start here and make it yours
1. Product vision: Why does the organization want the product?
2. Release criteria or acceptance criteria: how you know the product is done.
3. Major dates with program implications, such as demo or target release date.
4. Product roadmap or pointer to it
5. Pointer to other plans: Deployment plan, Sales plan, Training plan, etc.
6. You may need a pointer to the program risk list.

Remember: You can always add more, but people do not read long documents.
Figure 4.2: Program Risk Template

<table>
<thead>
<tr>
<th>Number</th>
<th>Risk Description</th>
<th>Probability</th>
<th>Severity</th>
<th>Exposure</th>
<th>Trigger Date</th>
<th>Mitigation Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The potential problem</td>
<td>High, medium or low</td>
<td>High, medium or low</td>
<td>3A, 3B</td>
<td>Date by which you need to act</td>
<td>Plan to deal with the risk</td>
</tr>
</tbody>
</table>

Figure 4.3: A Potential 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>

Figure 4.4: Example of an Agile Roadmap for One Quarter

Product Example: One Quarter Agile Roadmap

<table>
<thead>
<tr>
<th>Internal Release 1</th>
<th>Internal Release 2</th>
<th>Internal Release 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Logic, Part 1</td>
<td>Source Logic, New IP</td>
<td>Source Logic, Part 3</td>
</tr>
<tr>
<td>Admin, Part 1</td>
<td>Admin, Part 2</td>
<td>Admin, Part 2</td>
</tr>
</tbody>
</table>

All contents copyright Johanna Rothman, http://www.jrothman.com. Provided for your reading pleasure. If you want to use an image or wording, please contact me, jr at jrothman dot com.
Figure 5.1: Agile Roadmap for One Quarter

<table>
<thead>
<tr>
<th>Feature</th>
<th>Business Value Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature 1</td>
<td>2500</td>
</tr>
<tr>
<td>Feature 2</td>
<td>2000</td>
</tr>
<tr>
<td>Feature 3</td>
<td>1950</td>
</tr>
<tr>
<td>Feature 4</td>
<td>500</td>
</tr>
<tr>
<td>Feature 5</td>
<td>250</td>
</tr>
<tr>
<td>Feature 6</td>
<td>249</td>
</tr>
<tr>
<td>Feature 7</td>
<td>200</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>7649</strong></td>
</tr>
</tbody>
</table>

Figure 5.2: Ranking with Business Value Points
Figure 6.1: Possible Kanban for a Core Team

Figure 6.2: Potential for Release Frequency

Figure 7.1: Possible Picture of Nine Team Small-World Network
Figure 7.2: Community of Practice

![Community of Practice Diagram](image1)

Figure 8.1: Possible Kanban Board for a Core Team

![Kanban Board Diagram](image2)

Figure 9.1: Program Team Estimates by Team

<table>
<thead>
<tr>
<th>Team</th>
<th>Relative Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>12 weeks duration, 50% confidence</td>
</tr>
<tr>
<td>Team 2</td>
<td>16 weeks duration, 25% confidence</td>
</tr>
<tr>
<td>Team 3</td>
<td>8 weeks duration, 90% confidence</td>
</tr>
<tr>
<td>Team 4</td>
<td>10 weeks duration, 10% confidence</td>
</tr>
<tr>
<td>Team 5</td>
<td>7 weeks duration, 90% confidence</td>
</tr>
</tbody>
</table>
Figure 10.1: Program Feature Chart

Figure 10.2: Product Backlog Burnup
Figure 10.3: Voicemail Product Backlog Burnup, After Several Interim Releases

Figure 10.4: Program Obstacle Report

Figure 11.1: Comparison of Fixed and Growth Mindset
Figure 12.1: Release Frequency and the Cost of Architectural Decisions

<table>
<thead>
<tr>
<th>Software as a Service</th>
<th>Sprint Software</th>
<th>Product with Features</th>
<th>Definitive Repeatability or Mechanical Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous</td>
<td>Infrequently</td>
<td>Architected several times a day</td>
<td>Architected once or twice a year</td>
</tr>
<tr>
<td>Infrequently</td>
<td>Infrequently</td>
<td>Less often</td>
<td>Low cost</td>
</tr>
<tr>
<td>Last responsible moment is later</td>
<td>Last responsible moment is earlier</td>
<td>Architectural decisions and releasing costs low.</td>
<td>Architectural decisions and releasing costs more.</td>
</tr>
</tbody>
</table>

Figure 14.1: One Quarter Agile Roadmap for a Robot

Possible One Quarter Agile Roadmap for a Robot

<table>
<thead>
<tr>
<th>Internal Release 1: Component Only</th>
<th>Internal Release 2: Component Only</th>
<th>Internal Release 3: Joint Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mechanical Joint exploration/EAO</td>
<td>Mechanical Joint exploration/EAO</td>
<td>Joint Design, features 1-4 in emulator</td>
</tr>
<tr>
<td>Silicon: First phase layout to check for heat and performance</td>
<td>Silicon: First phase layout to check for heat and performance</td>
<td>Joint Design, features 5-6 in simulator</td>
</tr>
<tr>
<td>OS: Interact and driver interactions, features 1-4</td>
<td>OS: Interact and driver interactions, features 1-8</td>
<td>Create physical prototypes in lab</td>
</tr>
<tr>
<td>FPAA: Boot functions, 1-9</td>
<td>FPAA: Boot functions, 1-2</td>
<td>Simulate OS input to current joint design</td>
</tr>
<tr>
<td>Program Concern: Develop landing zone</td>
<td>Program Concern: Develop landing zone</td>
<td>Simulate OS input to current joint design</td>
</tr>
</tbody>
</table>

Figure 14.2: Possible Mechanical Engineering Kanban

<table>
<thead>
<tr>
<th>Ready</th>
<th>Analysis</th>
<th>Initial (proof)</th>
<th>Simulation</th>
<th>Simulation Test</th>
<th>Prototypes</th>
<th>Test</th>
<th>Final Test</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Possible to iterate up until here. From here, it's all serial.

All contents copyright Johanna Rothman, http://www.jrothman.com. Provided for your reading pleasure. If you want to use an image or wording, please contact me, jr at jrothman dot com.
### Figure 14.3: Possible Silicon Kanban

<table>
<thead>
<tr>
<th>Ready</th>
<th>Analysis</th>
<th>Landing Zone Specification</th>
<th>Simulation</th>
<th>Place &amp; Route</th>
<th>Prototype</th>
<th>Spec-out Review</th>
<th>Fabrication</th>
<th>Pilot</th>
<th>Final Test</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Analysis might include initial schematic design and review
Possible to iterate up until here. From here, it's all serial

### Figure 14.4: Possible FPGA Kanban

<table>
<thead>
<tr>
<th>Ready</th>
<th>Analysis</th>
<th>Develop &amp; Simulate</th>
<th>Simulation Test</th>
<th>Prototype</th>
<th>Prototype Test</th>
<th>Pilot</th>
<th>Final Test</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 15.1: Staggered Development and Testing

Two weeks of development

Two weeks of testing

This duration is the entire time box that counts

Figure 15.2: Implement by Feature

Figure 15.3: Curlicue Features
Figure 17.1: Staged Delivery Life Cycle

Figure 17.2: Design-to-Schedule Life Cycle

Feature teams develop their feature in each Design, Code, Integrate & Test box. The features are not all the same size. The teams manage scope, not timeboxes.

Feature teams develop their feature in each Design, Code, Integrate & Test box. The project stops when the project team runs out of time for new work.
Figure 17.3: Release Train: Each train releases on the same relative day each quarter
Glossary

If you are not familiar with the terms I've used, here are the definitions.

**Adaptive**: Any approach that allows you to adjust your practices or behavior to the current reality.

**Agile**: You work in small chunks, finishing work that is valuable to the customer in the order the customer specifies. The value of working in an agile way is that you have the ability to change quickly, because you complete work.

**Backlog**: Ranked list of items that need to be completed for the product.

**Cost of Delay**: The revenue impact you incur when you delay a project. Aside from "missing" a desired release date, you can incur Cost of Delay with multitasking, or waiting for experts, or from one team waiting for another in the program. All of these problems—and more—lead to delay of your product release.

**Community of Practice**: A way to share knowledge among people who belong to different teams, and share the same interests or function. For example, in a program, you might have an architecture community of practice that helps any developer learn how to evolve the design of the product. A test community of practice would provide a forum for testers to discuss what and how to test.

**Flow**: The team takes a limited number of items to complete, and uses the WIP limit instead of a timebox as a way to control how much work the team takes.

**Generalizing Specialist**: Someone who has one skill in depth, and is flexible enough to be able to work across the team to help move a feature to done.

**Hardening Sprint**: If a team does not complete all the work they need for a release, they may need a hardening sprint to complete all the testing for a release. This is an indication the teams are not really getting to done each iteration. They have work in progress past the end of the iteration.

**Inch-pebble**: Inch-pebbles are one-to-two day tasks that are either done or not done.

**Iteration**: A specific timebox. For agile projects, that time is normally one to four weeks. In programs, I like even smaller iterations because you want feedback more often and want to build momentum.

**Kanban**: Literally the Japanese word for "signboard." A scheduling system for limiting the amount of work in progress at any one time.

**Lean**: A pull approach to managing work that looks for waste in the system.

**MVP**: Minimum viable product. What is the minimum you can do, to create an acceptable product? This is not barely good enough quality. This is shippable product. However, this is minimal in terms of features.

**Pairing**: When two people work together on one task.

**Parking Lot**: This is a place to put issues you don't want to lose but don't necessarily want to address at this time.

All contents copyright Johanna Rothman, http://www.jrothman.com. Provided for your reading pleasure. If you want to use an image or wording, please contact me, jr at jrothman dot com.
Spike: If you cannot estimate a story, timebox some amount of work (preferably with the entire team) to learn about it. Then you will be able to know what to do after the day or two timebox.

Servant Leadership: An approach to managing and leading where the leader creates an environment in which people can do their best work. The leader doesn't control the work; the team does. The leader trusts the team to provide the desired results.

Sprint: An iteration in Scrum.

Swarming: When the team works together to move a feature to done, all together.

Technical Debt: Shortcuts a team takes to meet a deliverable. Teams might incur technical debt on purpose, as a tactical decision. Technical teams can have architectural, design, coding, and/or testing debt. Program teams might have risk or decision debt---the insufficiency of work for managing risks or making decisions.

Timebox: A specific amount of time in which the person will attempt to accomplish a specific task.

WIP or Work in Progress: Any work that is not complete. When you think in lean terms, it is waste in the system. Note that you do not get credit for partially completed work in agile.
Supplemental Materials for
Agile and Lean Program Management: Scaling Collaboration Across the Organization

Annotated Bibliography


[DWE07] Dweck, Carol. Mindset: The New Psychology of Success. Ballantine Books, New York, 2007. This book discusses the fixed mindset and the growth mindset. If you have the fixed mindset, you believe you can only do what you were born with. If you have the growth mindset, you believe you can acquire new skills and learn. The growth mindset allows you to improve, a little at a time.


All contents copyright Johanna Rothman, http://www.jrothman.com. Provided for your reading pleasure. If you want to use an image or wording, please contact me, jr at jrothman dot com.
Supplemental Materials for
*Agile and Lean Program Management: Scaling Collaboration Across the Organization*

2002. The original and definitive text on servant leadership. The forewords and afterwords provide significant value to understanding how servant leaders work.


[PAT14] Patton, Jeff. *User Story Mapping: Discover the Whole Story, Build the Right Product.* O'Reilly, Sebastopol, CA, 2014. A terrific way to explain your stories to yourself. This book will help you move from epics and themes to stories your feature teams can build.


[BCD05] Rothman, Johanna and Esther Derby. *Behind Closed Doors: Secrets of Great Management.* Pragmatic Bookshelf, Dallas, TX and Raleigh, NC, 2005. We describe the Rule of Three and many other management approaches and techniques in here.

[ROT07] Rothman, Johanna. *Manage It! Your Guide to Modern, Pragmatic Project Management.* Pragmatic Bookshelf, Dallas, TX and Raleigh, NC, 2007. If you want to know more about how to estimate task size, establish a project rhythm, or see a project dashboard, this is the book for you. I have references about why multitasking is crazy in here.

[ROT09] Rothman, Johanna. *Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects.* Pragmatic Bookshelf, Dallas, TX and Raleigh, NC, 2009. Sometimes, program managers encounter project portfolio decisions with the feature set, or the request for people to multitask. This book helps you manage all the work in your project portfolio. I also have more references about why multitasking is crazy in here.


All contents copyright Johanna Rothman, [http://www.jrothman.com](http://www.jrothman.com). Provided for your reading pleasure. If you want to use an image or wording, please contact me, jr at jrothman dot com.
Supplemental Materials for
Agile and Lean Program Management: Scaling Collaboration Across the Organization


Agile principles are at http://www.agilemanifesto.org/principles.html
More from Johanna

I consult, speak, and train about all aspects of managing product development. I provide frank advice for your tough problems. I'm more interested in helping you become more effective than I am in sticking with some specific approach. There's a reason my newsletter is called the "Pragmatic Manager"--that's because I am!

If you liked this book, you might also like the other books I've written:

*Diving for Hidden Treasures: Uncovering the Cost of Delay in Your Project Portfolio*, http://www.jrothman.com/books/diving-for-hidden-treasures/

*Predicting the Unpredictable: Pragmatic Approaches to Estimating Project Schedule or Cost*, http://www.jrothman.com/books/predicting-the-unpredictable-pragmatic-approaches-to-estimating-cost-or-schedule/

*Project Portfolio Tips: Twelve Ideas for Focusing on the Work You Need to Start & Finish*

*Manage Your Job Search*, http://www.jrothman.com/books/manage-your-job-search/

*Hiring Geeks That Fit*, http://www.jrothman.com/books/hiring-geeks-that-fit/


In addition, I have essays in:

*Readings for Problem-Solving Leadership*, https://leanpub.com/pslreader

*Center Enter Turn Sustain: Essays on Change Artistry*, https://leanpub.com/changeartistry

I'd like to stay in touch with you. If you don't already subscribe, please sign up for my email newsletter, the Pragmatic Manager, http://www.jrothman.com/pragmaticmanager, on my website. Please do invite me to connect with you on LinkedIn, http://www.linkedin.com/in/johannarothman, or follow me on Twitter, @johannarothman.

I would love to know what you think of this book. If you write a review of it somewhere, please let me know. Thanks!

Johanna