Agile: Not just for software

Eileen Wrubel
Copyright 2018 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM18-0407
Why do we care?

**Deliver performance at the speed of relevance**

**Streamline rapid, iterative approaches from development to fielding**

National Defense Strategy Summary
Jan 2018

"Simply delivering what was initially required on cost and schedule can lead to failure in achieving our evolving national security mission — the reason defense acquisition exists in the first place."

Honorable Frank Kendall
Under Secretary of Defense (AT&L)
2015 Performance of The Defense Acquisition System
Agile Manifesto

Through this work we have come to value:

That is, while there is value in the items on the right, we value the items on the left more.

http://www.agilemanifesto.org/
Twelve Agile Principles Support the Manifesto

1. Highest priority is satisfy the customer through early and continuous delivery of software

2. Welcome changing requirements, even late in development

3. Deliver working software frequently, from a couple of weeks to a couple of months

4. Business people and developers must work together daily throughout the project

5. Build projects around motivated individuals. Provide environment and support they need

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation

7. Working software is the primary measure of progress

8. Agile processes promote sustainable development…a constant pace indefinitely

9. Continuous attention to technical excellence and good design enhances agility

10. Simplicity—the art of maximizing the amount of work not done—is essential

11. The best architectures, requirements, and designs emerge from self-organizing teams

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Key Elements of Scrum

- Product Backlog
- Sprint Backlog
- Daily Scrum Meeting
- 24 hrs.
- 2-4 weeks
- Potentially Shippable Product
### Key Elements of Kanban

<table>
<thead>
<tr>
<th>TASKS</th>
<th>IN PROGRESS</th>
<th>STOPPED</th>
<th>COMPLETED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defined States the Work</td>
<td>Rules about Limiting Work in Process for Each State</td>
<td>Explicit Acceptance Criteria</td>
<td></td>
</tr>
<tr>
<td>Progresses Through</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Useful for Tasks That Don’t Easily Conform to a Time Box**
SAFe Lean-Agile principles

#1 - Take an economic view

#2 - Apply systems thinking

#3 - Assume variability; preserve options

#4 - Build incrementally with fast, integrated learning cycles

#5 - Base milestones on objective evaluation of working systems

#6 - Visualize and limit WIP, reduce batch sizes, and manage queue lengths

#7 - Apply cadence, synchronize with cross-domain planning

#8 - Unlock the intrinsic motivation of knowledge workers

#9 - Decentralize decision-making
Economic View: What’s valuable to you?

Spend time doing only valuable work

Sequence work according to value

Use decision rules about value to communicate the economic view
There’s No One True Way

Assume there are multiple ways to get from here to there
Agile Software Teams + Systems Engineering Teams

- Agile software teams interacting with traditional systems engineering
- Systems engineers acting as Agile team members
- Systems engineers applying Agile methods to their own work
Agile software teams interacting with stakeholders

Stakeholders acting as Agile team members

Stakeholders applying Agile methods to their own work
Consider the Full Value Stream

- All stages from initial concept to final realization of value
- Optimize the flow across those stages, not the productivity of one stage over all others
- The *product* or *service* is a system too
Related Business Processes
How do I take a 2-week slice of this?
Iterate on Both Process and Product
What’s next?

How Agile can you get?