

Software Solutions Symposium 2017

March 20–23, 2017

A Tale of Two (Agile) Programs

Suzanne Miller
Will Hayes

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



Copyright 2017 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

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.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0004526

Agenda



Setting the Stage

The Eco-System of Agile in Government

Two Programs Contexts

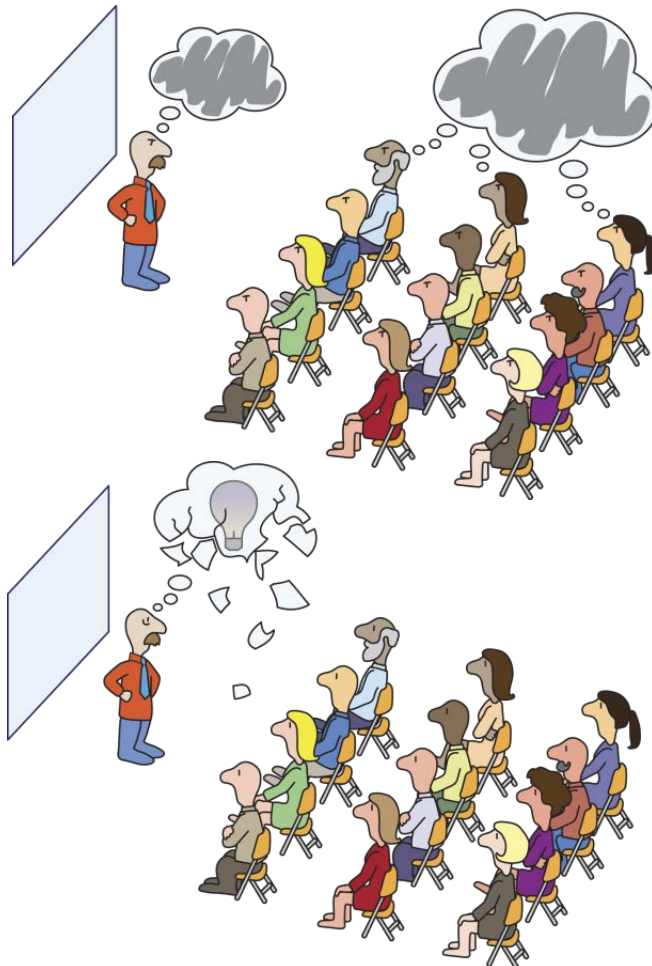
Gimmees and Gotchas for Our Two Programs-Role Play

Getting Off the Stage

Setting the Stage

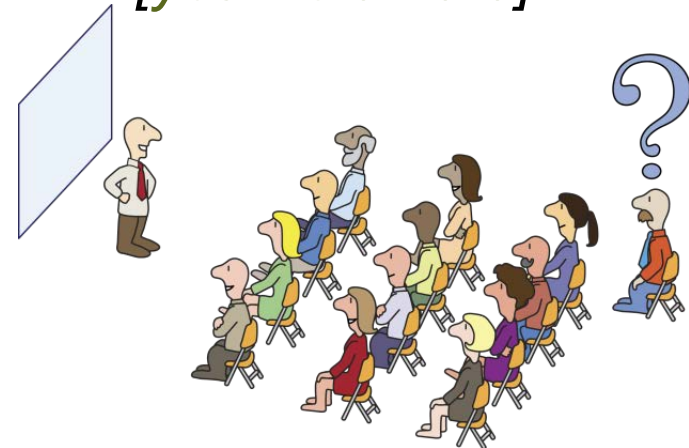


Audience—Who are You?



*What role do you have in
(Agile) software
acquisition?*

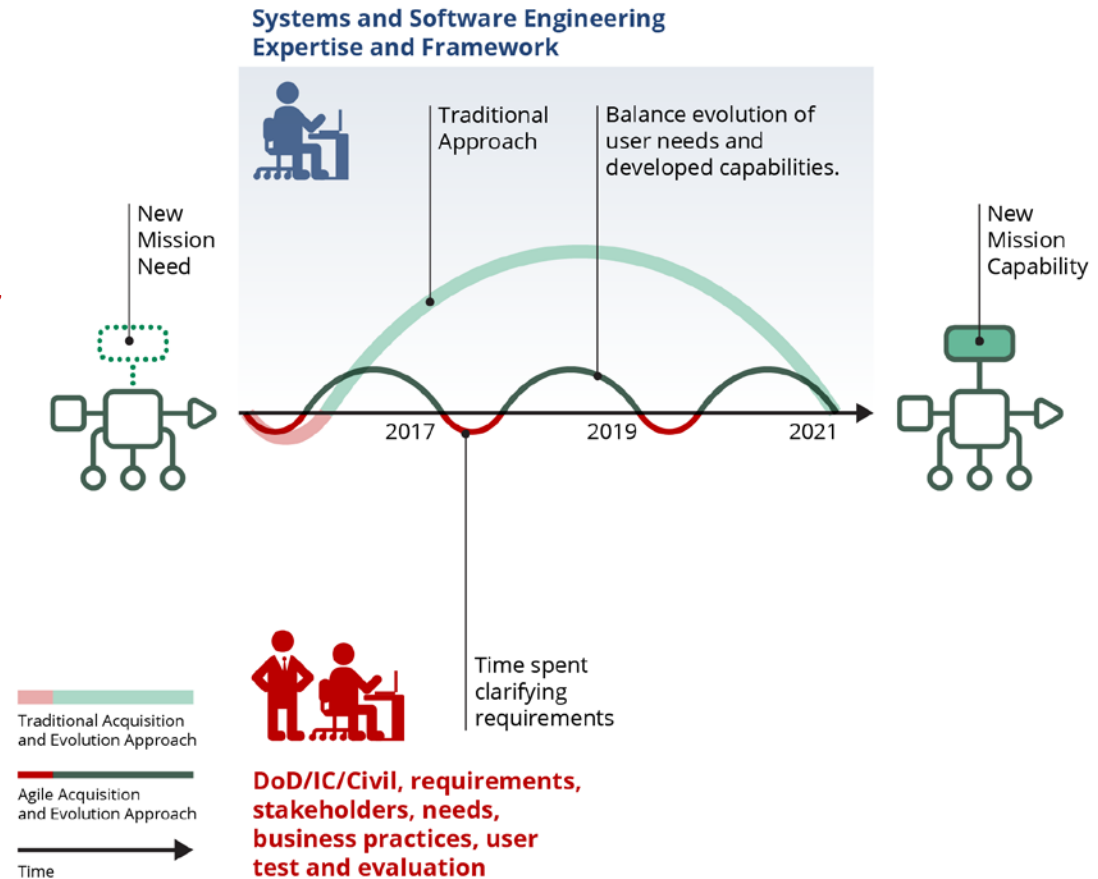
- Agile Sponsor*
- Agile Champion*
- Engineer*
- Program Management*
- Budget Staff*
- Contracting Staff*
- [your role here]...*



Motivation for Agile: Gov't Acquisition and Innovation

Many regulated environments, like the DoD, NEED innovation and NEED incremental improvements to their systems.

Many of them are now willing to consider changing their approach *if they can do it without getting in trouble with their governing statutes and regulations.*



Assumption: You Already Understand the Agile Manifesto and Principles

Is the above a good assumption for this audience?

What are the tenets and principles trying to enable?

1. Increase speed of development of high business value software
 - Speed=the time it takes to move ideas from one person to another
 - A reason that all Agile approaches emphasize face-to-face contact as much as possible
2. Take advantage of the benefits of small batch sizes
 - Fast feedback, fast learning, etc
 - What is the next smallest thing I can do that increases the business value delivered?
3. Reduce non-value-added work and rework
 - Reduce “time to fielding”
 - BUT understanding what is truly non-value added
 - Non-value for designer may be HIGH value for sustainer

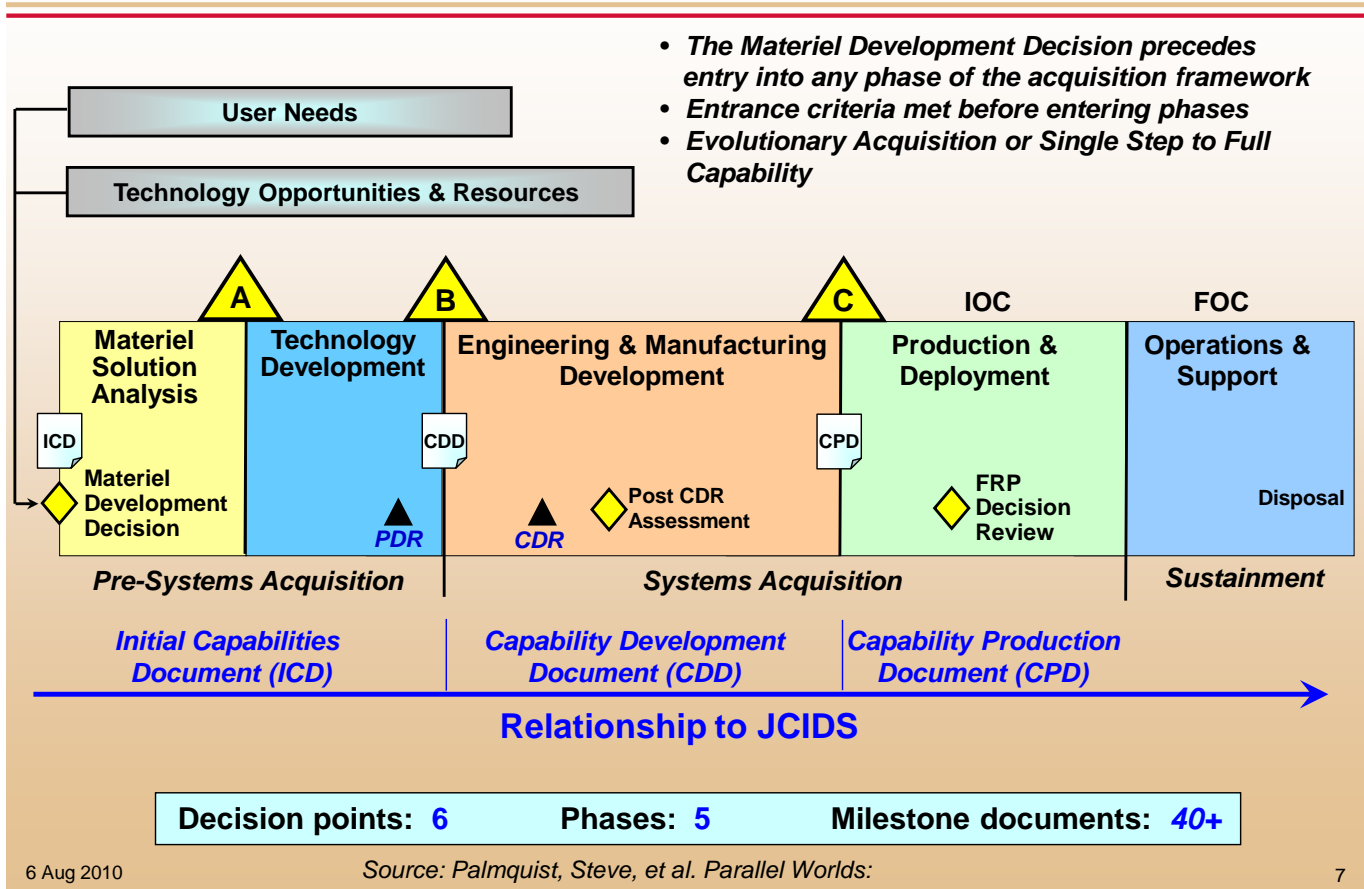
The Eco-System of Agile in Government



Agile Methods are Used in Context of a Larger Program (More Often Than Not)



The Defense Acquisition Management System

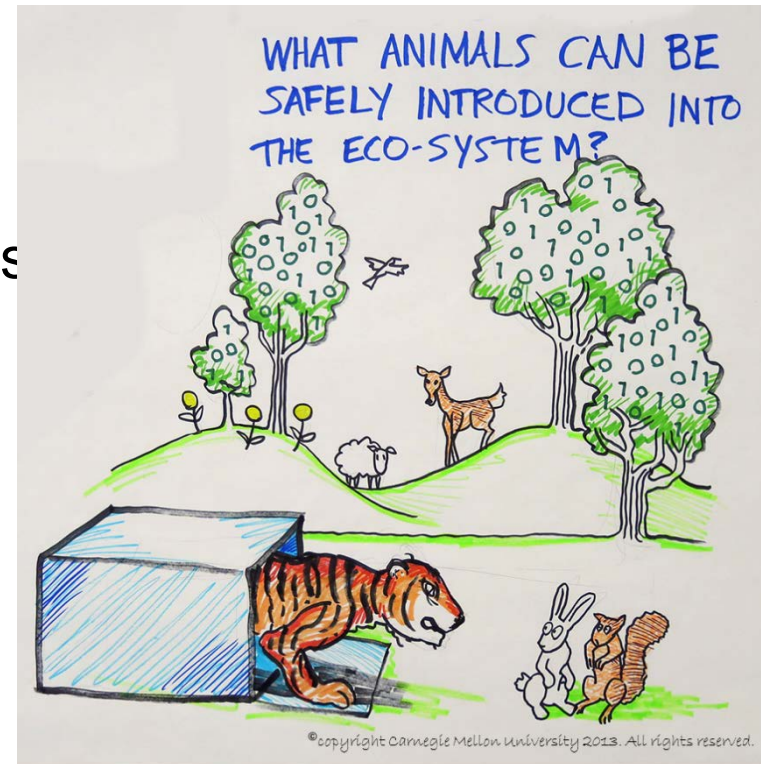


- *The Materiel Development Decision precedes entry into any phase of the acquisition framework*
- *Entrance criteria met before entering phases*
- *Evolutionary Acquisition or Single Step to Full Capability*

Components of the Eco-System

Components include but not limited too:

- Program Management Office
- Contracts
- Finance
- System Engineering
- User (end, interfacing system(s) users
- Stakeholders
- Information Assurance
- Supply Chain
- Developers
- Testers
-



Two Program Eco-Systems



Two Programs (SKULLY and FOX), Somewhat Common Contexts

Attributes SKULLY and FOX have in common:

- deployed on a global scale
- highly complex data fusion requirements
- operate in a System Of Systems context
- multiple contracted development & service providers engaged
- elaborate supply chain considerations
- full range of ACAT program levels operating under formal DoD acquisition process, in parallel, mapped out over a long time horizon
- managed by combination of military & government civilian personnel

Mechanics of Next Section

SuZ and her partner will each play the sponsor role for one of the programs

SuZ will review each “Gimmee” or “Gotcha” section and then the two sponsors will have a discussion about similarities and differences in their experience.

Gimmees and Gotchas



Gimmees & Gotchas

Gimmees are a list of behaviors that give you confidence that your program office and/or contractor is embracing an agile process.



Gotchas are a list of behaviors that may indicate problems currently exist or on the horizon in your agile program.

These Gimmees & Gotchas **are not** intended to be all inclusive nor are they a checklist. The goal of these is to help identify areas to investigate further and focus your energy toward a successful program.

First Steps Gimmees



- Your motivation, trade-offs, benefits, and expectations for using an Agile approach are clearly understood and communicated
- There is an explicit understanding that the requirements are expected to evolve
- Automated testing is planned for and budgeted
- Contract and CDRLs allow flexibility and incremental delivery
- Entire program team is aware that the Jan 2015 DoD 5000.02 has new lifecycle descriptions that support more "agile" approaches
- Hindrances and enablers for agile implementation are acknowledged
 - paths to success are identified, preferably through a method like SEI Readiness/Fit Analysis

First Steps Gotchas



- Senior managers and stakeholders reluctant to use agile or are unaware that Agile is in use
- Software development is constrained to a hardware-based Work Breakdown Structure
- Mindset that document completion equals progress is prevalent
- Program exhibits a risk averse culture
- Integration testing isn't planned to occur until just before final delivery
- Testing isn't budgeted until much later in the program
- Agile is being treated like a silver bullet

Readiness Gimmees



- Your Agile approach has been tailored to best meet your program's needs
- Program office staff including systems engineers understand their role in the agile process you're using
- The agile manifesto and principles are understood throughout the organization
- Appropriate training has been provided for the entire organization
- Expectations and artifacts necessary for milestone decisions have been agreed to and documented
- Agile roles and responsibilities have been clearly assigned
- The definition of done has been established and includes what documentation is required for iteration and increment deliveries
- The program office is open to changing roles
- Leadership and staff are educated on differences from the way they are used to doing business
- Program develops and utilizes adoption support (communication and implementation) mechanisms

Readiness Gotchas



- Your testing function/organization has not been integrated into the day-to-day activities
- Requirements stability, operating environment, and the evolution of the technology base has not been fully assessed
- Constraints are imposed for the sake of tradition
- Contract progress payments are based on "earned value" for the accounting period vs the Agile working cadence
- Regulations are cited as a reason not to embrace agile approaches

Implementation Gimmees



- Users and other stakeholders can accommodate incremental deliveries
- Necessary and beneficial documentation has been identified
- Requirements can be prioritized without pushback
- Agile requirements constructs –stories, features, epics, etc. -- have clear completion criteria
- (Incremental)technical reviews are structured to understand technical issues and mitigate technical risks
- Iteration and release reviews are used to build a case to demonstrate readiness to pass milestone reviews
- Agile measurements are integrated into overall management metrics
- Measurements are focused on "are we producing sufficient value fast enough?"
- User requirements are validated during the creation of user stories and features
- The program office is changing how they perform their responsibilities (e.g. incremental technical reviews, CDRL deliveries...)

Implementation Gotchas



- Your program or contractor is proposing agile as a quick fix for existing failures on the program
- Team metrics are used for comparisons
- Users and stakeholders who are not actively engaged in the Agile processes can't adapt to small batch deliveries
- Oversight activities are abandoned
- Multiple organizational change initiatives compete with Agile for the attention of leadership
- Cadence of multiple interacting teams hasn't been synchronized
- Focus is on compliance rather than mission success
- Derived requirements can't be reprioritized based on what is learned in early development iterations
- Contractor leadership and engineering implementation staff are not aligned on process changes needed to support Agile delivery
- Incentives to contractor don't reflect Agile/lean principles

Getting off the Stage -- Summary



A Few Take Aways

Even though training isn't the only transition mechanism that is useful in adopting Agile, it's an important one

- The program that didn't have the bulk of their government or contractor staff trained in the methods they were using suffered more startup difficulties than the one that invested in early and extensive training
- Contractor vs government pushing for Agile does make a difference in how interactions can occur
 - But even when the contractor is the one pushing, there may be areas where transparency is still difficult to achieve
- Both settings ran into challenges in implementing automated testing
 - It's never too early to start on that path!
- Benefit seen in both settings:
 - *Program risks that would not have been identified until much later have already been dealt with*

Overall Messages

Agile is not a silver bullet for software acquisition, but it can be used in a government setting when appropriate

Early training in Agile/Lean principles and concepts changes the nature of the conversation among roles and between government and contractor

- Lack of early training hampers implementation decision making

How the program office tasks for oversight are performed in a program using Agile is different from a traditional acquisition, but the responsibility of the PMO remains

- Being clear about roles and responsibilities in your Agile process is key
- Mapping your Agile process to the traditional, expected DoD acquisition process supports communication

Even if huge schedule/cost gains aren't immediately visible, identification of program/technical risks much earlier in the life cycle than typical is seen as a win in both programs

Even if a program suffers initial mis-steps in Agile adoption, perseverance and appropriate coaching/training can make a large positive impact

Contact Information

Suzanne Miller

Principal Researcher

SSD/CTSD

Telephone: +1 412-268-9143

Email: smg@sei.cmu.edu

Will Hayes

Principal Engineer

SSD/CTSD

Telephone: +1 412-268-6398

Email: wh@sei.cmu.edu

Web

www.sei.cmu.edu

www.sei.cmu.edu/contact.cfm

www.sei.cmu.edu/acquisition/research

U.S. Mail

Software Engineering Institute

Customer Relations

4500 Fifth Avenue

Pittsburgh, PA 15213-2612

USA

Customer Relations

Email: info@sei.cmu.edu

Telephone: +1 412-268-5800

SEI Phone: +1 412-268-5800

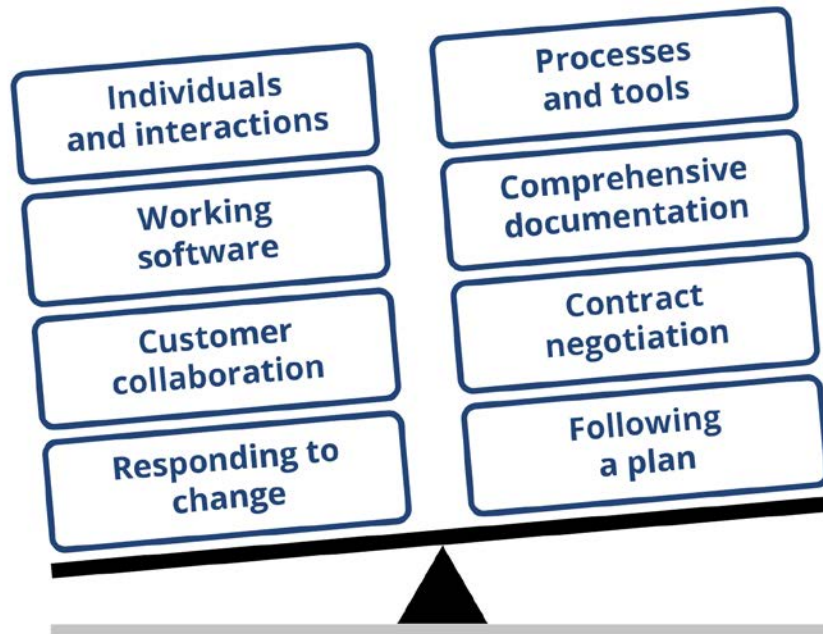
SEI Fax: +1 412-268-6257

Software Solutions Symposium 2017

Backup Slides

Agile Manifesto—STILL the Basis for Agile Thinking in Industry

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.

Common myth:

The manifesto is often misinterpreted to mean:

no documentation, no process, and no plan!

<http://www.agilemanifesto.org/>

Agile Principles Accompanying the Manifesto₁ – All are important aspects of building an Agile culture

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.

Agile Principles Accompanying the Manifesto₂ – All are important aspects of building an Agile culture

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.

Adapted from <http://agilemanifesto.org/principles.html>