

Agile Across Large DoD HW/SW Programs

MARCH 22, 2023

Ms. Brigid O'Hearn
Senior Software Acquisition Innovation Specialist



Document Markings

Copyright 2023 Carnegie Mellon University.

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.

DM23-0257

Agenda

- Aircraft Platforms with Common Software
- Agile Deep Dive

Agile Across Large HW/SW Programs

Aircraft Platforms with Common Software

Aircraft Platforms with Common Software Challenges



F-35s and F-15s Take-off for training [Source: DVIDS
<https://www.dvidshub.net/about/copyright>]

- Multiple models or variants
 - Different aerodynamics
 - Different versions of avionics
- Multiple Users
 - International partners or Foreign Military Sales
 - Several US Military Services
 - Unique missions
- Multiple Vendors
 - Interface & integration challenges

Aircraft Platforms with Common Software

Best Practices



C-130J in Alaska [Source: DVIDS <https://www.dvidshub.net/about/copyright>]

- Product line engineering approach:
 - Modern software approaches
 - Hardware abstraction
 - Digital Engineering
- Multiple Users:
 - Transparent governance
 - Robust processes (requirements, funding, prioritization, etc)
- Multiple Vendors:
 - Modular Open Systems Approach
 - Automated integration; hardware in the loop labs; modeling

Agile Across Large HW/SW Programs

Agile Deep Dive

Agile Transformation of Legacy Systems

Challenges



F-35s and F-15s Take-off for training [Source: DVIDS
<https://www.dvidshub.net/about/copyright>]

- Existing waterfall processes often take 3-10 years to deliver a capability
- Program offices often operate on oversight vs collaboration
- Stakeholder involvement is less frequent
- Personnel have little experience or training in modern software and/or digital transformation methods
- Organizations are not resourced or organized for Agile

Agile Transformation of Legacy Systems

Best Practices



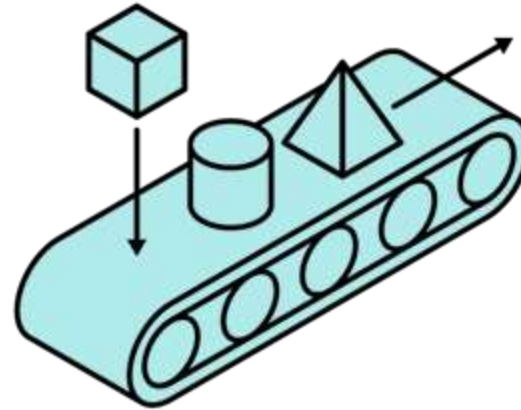
C-130J in Alaska [Source: DVIDS <https://www.dvidshub.net/about/copyright>]

- Identify Minimum Viable Processes
 - Shift from oversight to insight
 - User engagement and feedback
 - Define full requirements but decompose/prioritize into deployable partial capabilities
- Resourcing, organization, training
 - Partner to resource the team
 - Team alignment vs organizational assignment
 - Training with hands-on experience
 - Embed personnel where needed

Agile Represents a Major Requirements Transition – Different Acquisition Objectives

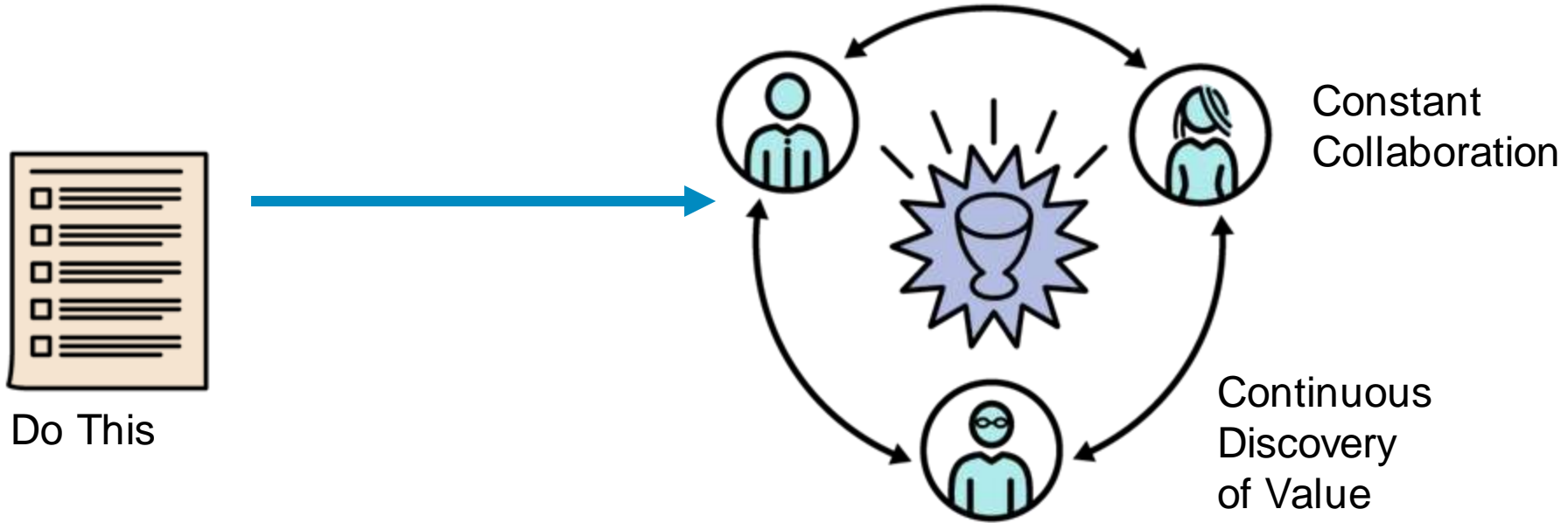


Buying a
Box

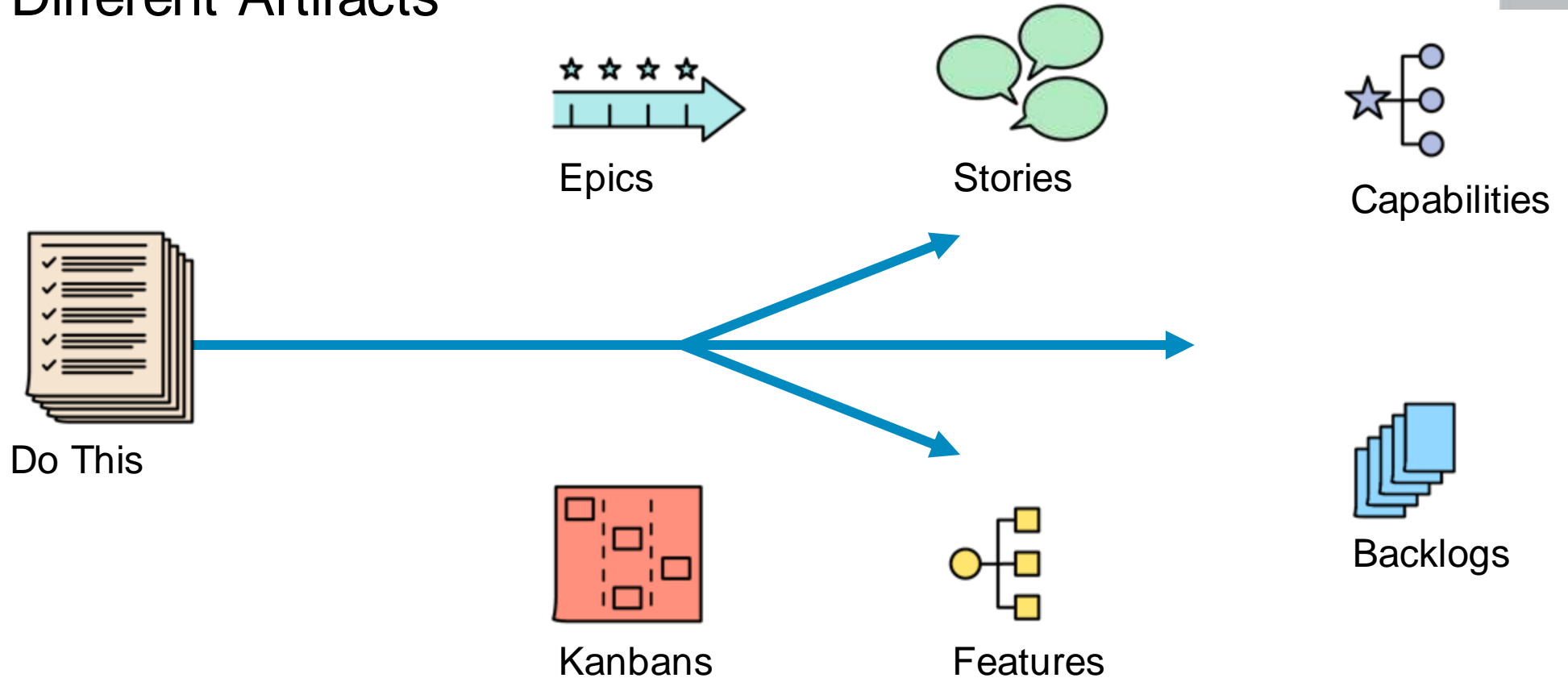


Buying an ongoing
delivery stream

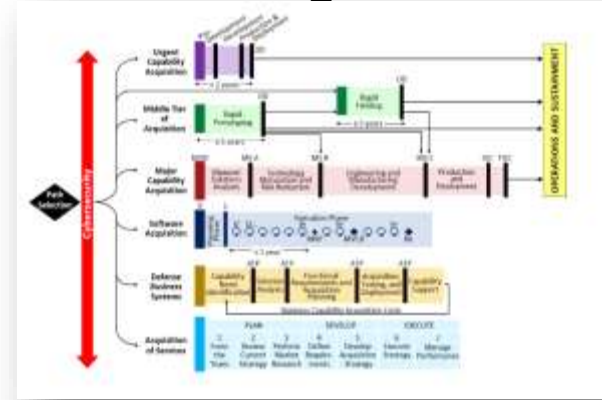
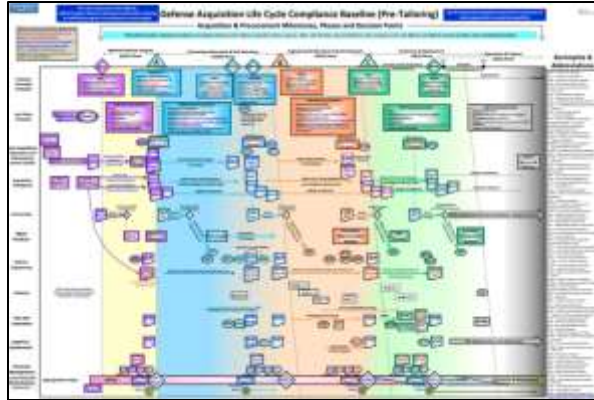
Agile Represents a Major Requirements Transition – Different Acquisition Objectives



Agile Represents a Major Requirements Transition – Different Artifacts



Agile Represents a Major Requirements Transition – Different Practices for Requirements Management

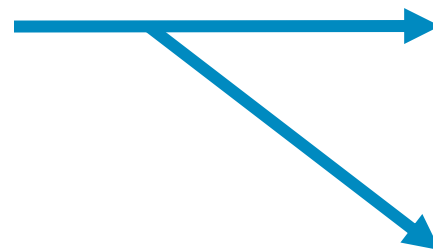


New ways of acquiring systems.

Note the Software Acquisition Pathway (SWP) for software-dominant products.

<https://aaf.dau.edu/>

Date-based Milestones



Increment-Based Demos

Definition of Done



Team Increment



System Increment



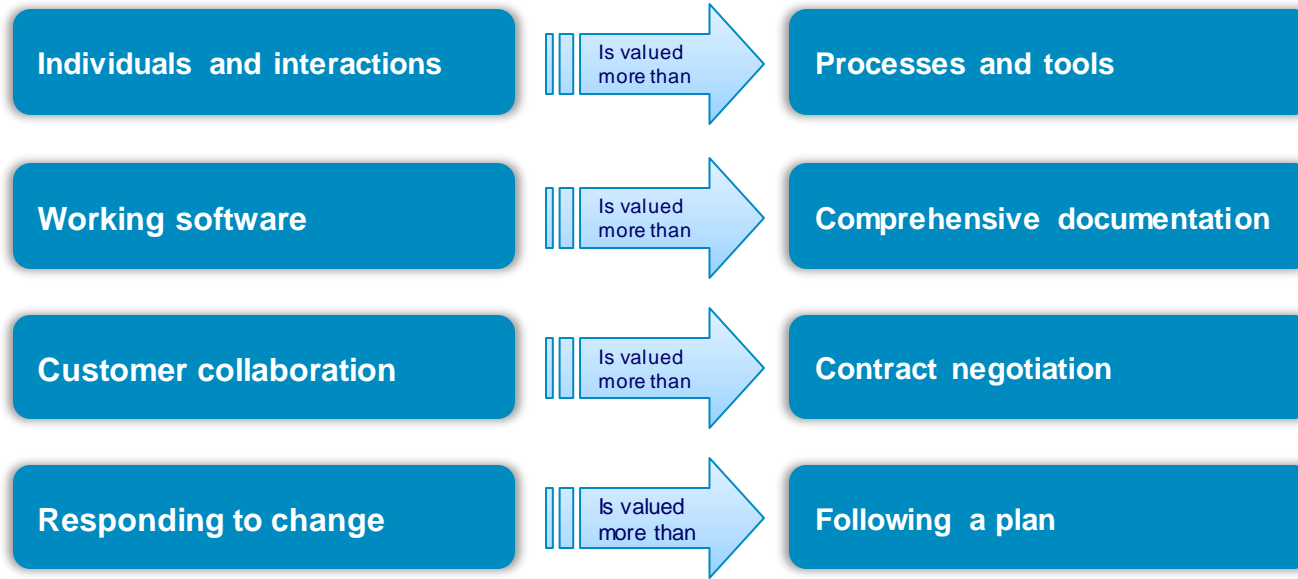
Solution Increment



Release

Agile Values from Manifesto for Agile Software Development

While there is value in the items on the right, **we value the items on the left more.**

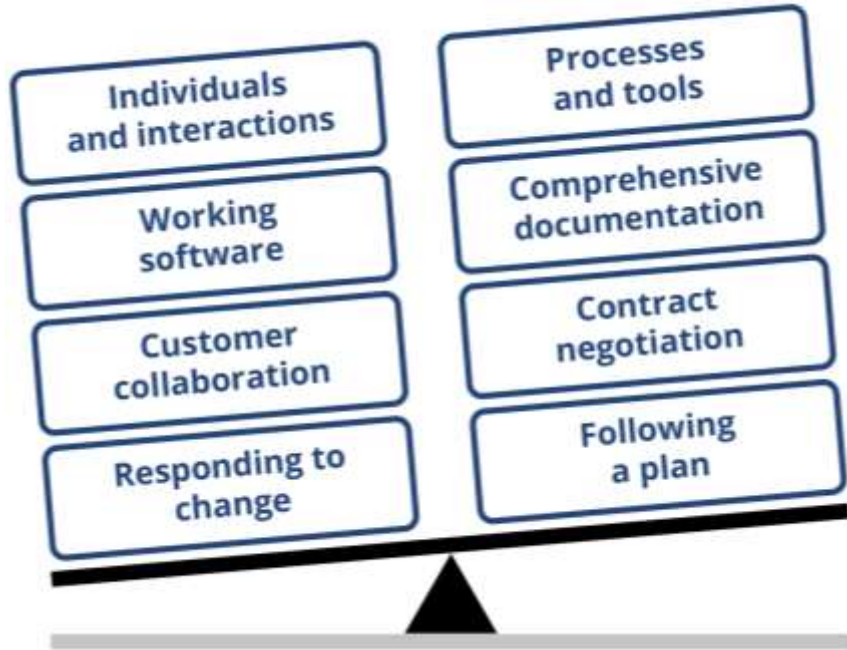


Which side do you think will benefit your users more?

<http://agilemanifesto.org>

Reorienting the Manifesto for Agile Software Development Toward *Multi-Program System Integration*

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.

<https://agilemanifesto.org/history.html>

Agile

Traditional

Many small batch interactions

Few large batch interactions

Demos/User feedback that explicitly exercise interfaces early and often

Primarily documentation review until late life cycle

Continuous refinement of interface-prioritized backlog across programs

Requirements & interface documents coordinated at infrequent designated acquisition points

Decisions/reqmts constantly validated with data from implementation

Decisions/reqmts periodically validated with analysis until data from implementation available (late)

Bottom Line for Programmatics of Complex Systems

INSIGHT is a necessary *enabler* to effective oversight

- Well-established CDRLs and DIDs may not always be the best source of insight
 - Aversion to all off-nominal conditions
 - Conformance to plan becomes the goal
- Agile development settings promote transparency and ongoing insight
 - *Available mechanisms, however, require proactive participation from the acquirer to be effective*

OVERSIGHT is an element of what makes the DoD Acquisition Ecosystem Work

- Oversight mechanisms established by program management determine:
 - Nature of information made available
 - Frequency of communication
 - Urgency/importance
- Well-established procedures & templates convey oversight requirements
 - *Recent developments like Adaptive Acquisition Framework change some of those requirements*



Compliance



Collaboration

Batch Size

Aspirations for Small Batch:

- We can learn from even small pieces being implemented/done
- “Stop starting, start finishing”
- Work in Progress is limited to enhance flow through the system
- 100% utilization of resources is recognized as limiting flow, flexibility, and work accomplishment
- Short time between when a defect is found and when it was created
 - Root cause analysis easier with current work, rather than work done in the past
- LOTS of integration happening across entire system, building confidence
- Tendency to “build quality in”

Typical Large Batch Realities:

- “Nothing is done until everything is done”
- More Work in Progress is good
- 100% utilization of resources is a goal
- Optimistic reporting of progress in order to “keep the program sold”
- Large scope integration events identify defect levels that strain resources
 - Increases number of potential defects that affect multiple areas of the system
 - Reduces confidence in system robustness
 - Harder for engineers to find sources of defects
- Tendency toward “test quality in”

Feedback

Aspirations for a Broader Aperture:

- Recognition that demo doesn't EQUAL test, but INFORMS it
- Stakeholder participation in demos of small pieces of functionality
- Open, continuous feedback about both the fact of and the meaning of progress or lack thereof
- Info from demos is fed forward to testing and certification staff to ensure alignment
- Definition of Done that includes certification needs (cyber, DT/OT, ATC, ATO, etc.)
- Participation on continuous integration team by govt staff seen as a high priority

Typical of "Primarily Document" Focus:

- Prefer larger, less frequent demos
- Requirements documents seen as "ground truth" for user needs, even when known to be superseded
- Constraints on opportunities for feedback
- Rushed feedback on documents
- More investment in documenting "to be" state than in documenting "as built"
 - Using documents to "lock down" design,
 - Then struggling to keep them current?

Requirements

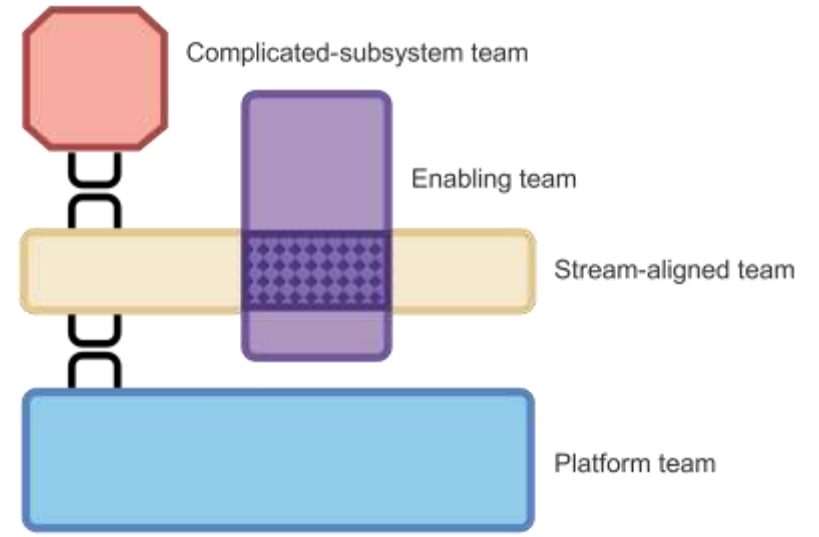
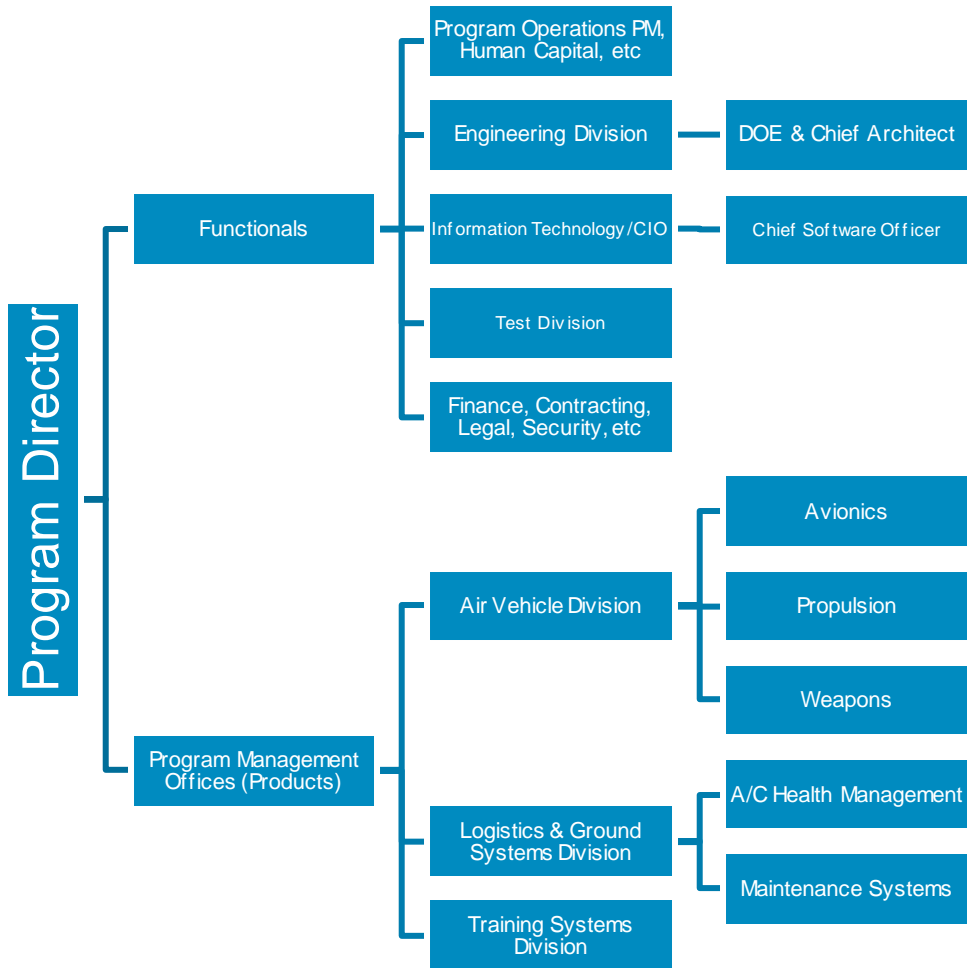
Aspirations for Iterative Approach:

- Mix of “push” and “pull” communication across govt/contractor interface as requirements are elaborated/refined
 - Facilitated by workflow and collaboration tools
- Frequent high bandwidth meetings keep the relationship going, not just technical work
- Transparency among stakeholders is an essential ingredient to build trust
- Frequent small batch prioritizations build a solid base of understanding of current state and progress

Typical of “Single Delivery” Focus:

- Work must commence early to limit risk
- Narrow time window to set the baseline
- Increasing resistance to requirements change over time, though knowledge of real user need continues to evolve
- Favoring breadth over depth in reviews
 - Hard to take in the large requirements set
 - Time for “digging in” on critical issues is rarely available during the review
- Sometimes we get as far as we can, declare success, and track action items until the next event

Organizational Alignment vs Team Assignment



© Matthew Skelton and Manuel Pais from *Team Topologies*

Contact



Brigid O'Hearn
Senior Software Acquisition
Innovation Specialist

Telephone: +1 478.952.0534

Email: bpohearn@sei.cmu.edu



Questions?