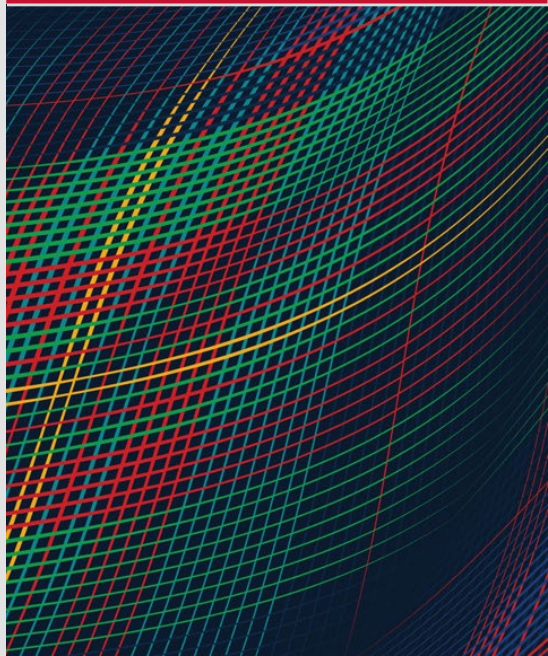
Carnegie Mellon University
Software Engineering Institute

Agile Software Development – Tech 101



MAY 2023

Forrest Shull
Lead for Defense Software Acquisition Policy Research
Software Engineering Institute, Carnegie Mellon University
2021 President of the IEEE Computer Society

© 2023 Carnegie Mellon University

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

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.

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.

DM23-0474

Agenda

- Why does DoD care about Agile software development?
- Intro to Agile software development
- Beyond buzzwords: What does Agile software look like in DoD?
- Myths and other things to be aware of

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

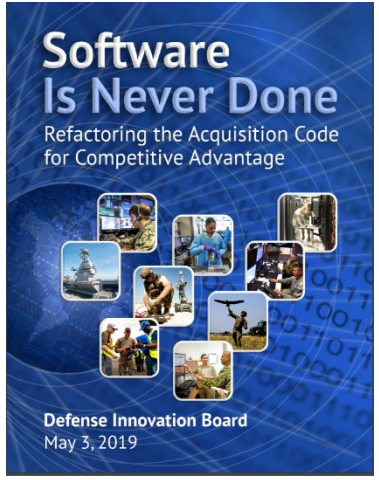
Agile Software Development – Tech 101

Why does DoD care?

Carnegie Mellon University
Software Engineering Institute

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Why Does DoD Care about Agile Software Development?



High-visibility performance problems with the traditional approach:
For example, Senator John McCain in 2017: "…it is unfortunate that the Air Force had already spent more than **half a billion dollars over the last ten years** on the AOC 10.2 upgrade, and yet **the program has not delivered any meaningful capability**… **Even more unfortunately, this program is only one example of the Department's troubling record on software-intensive systems**."

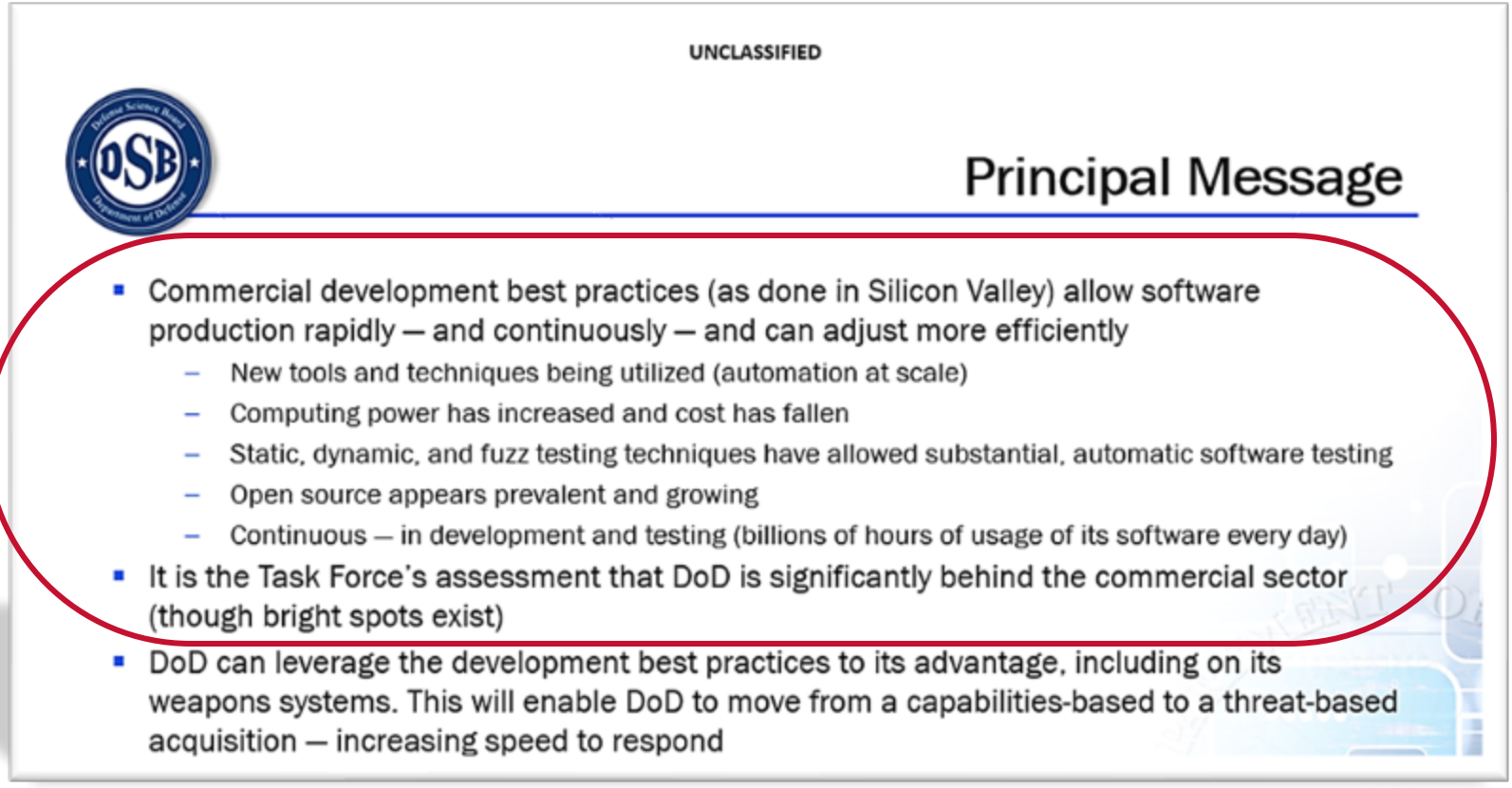
2019: DoD's traditional approach to software development "is broken and a leading source of risk to DoD: it takes too long, is too expensive, and exposes warfighters to unacceptable risk by delaying their access to tools they need to ensure mission success."

– Defense Innovation Board, "Software is Never Done"

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

2018 Defense Science Board Report: *Design and Acquisition of Software for Defense Systems*



UNCLASSIFIED

Principal Message

- Commercial development best practices (as done in Silicon Valley) allow software production rapidly — and continuously — and can adjust more efficiently
  - New tools and techniques being utilized (automation at scale)
  - Computing power has increased and cost has fallen
  - Static, dynamic, and fuzz testing techniques have allowed substantial, automatic software testing
  - Open source appears prevalent and growing
  - Continuous — in development and testing (billions of hours of usage of its software every day)
- It is the Task Force's assessment that DoD is significantly behind the commercial sector (though bright spots exist)
- DoD can leverage the development best practices to its advantage, including on its weapons systems. This will enable DoD to move from a capabilities-based to a threat-based acquisition — increasing speed to respond

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

2022: DSD Memo & Software Modernization Strategy

Carnegie Mellon University Software Engineering Institute

> "The Department's **adaptability increasingly relies on software** and **the ability to securely and rapidly deliver resilient software capability is a competitive advantage that will define future conflicts**. Transforming software delivery times from years to minutes will require significant change to our processes, policies, workforce, and technology.
> …Given this requires the **combined focus of DoD senior leadership**, I expect all offices and personnel to provide the support necessary to make software modernization a reality."

– DSD Kathleen Hicks, Department of Defense Software Modernization

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Agile Software Development – Tech 101

Intro to Agile Software Development

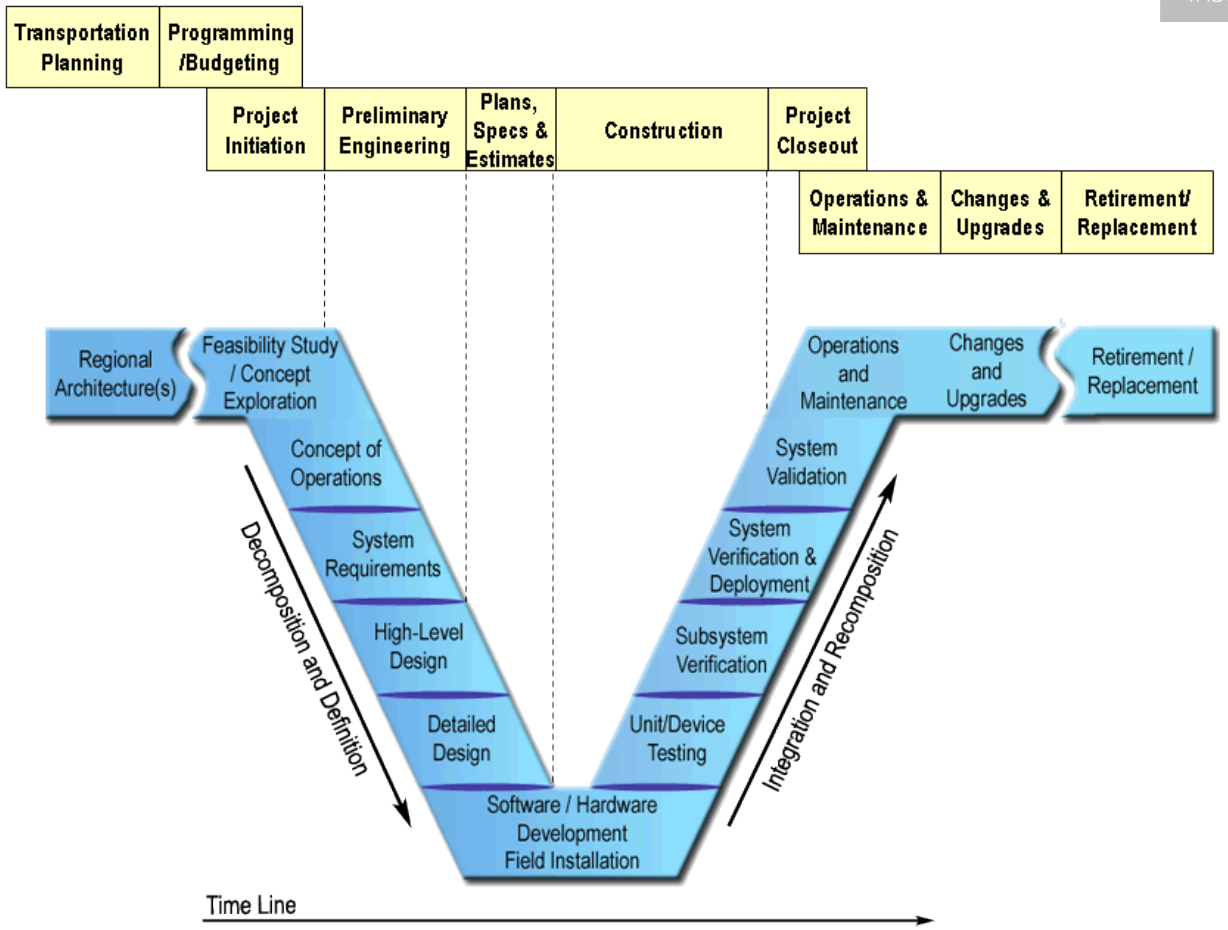
Carnegie Mellon University
Software Engineering Institute

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Traditional Approach: The Classic Engineering "V Model"

Software development is a creative, engineering activity – Any change requires:

- Requirements analysis
- Architecting / designing
- Implementing in code
- Testing
- Deployment
- Configuration management
- etc.



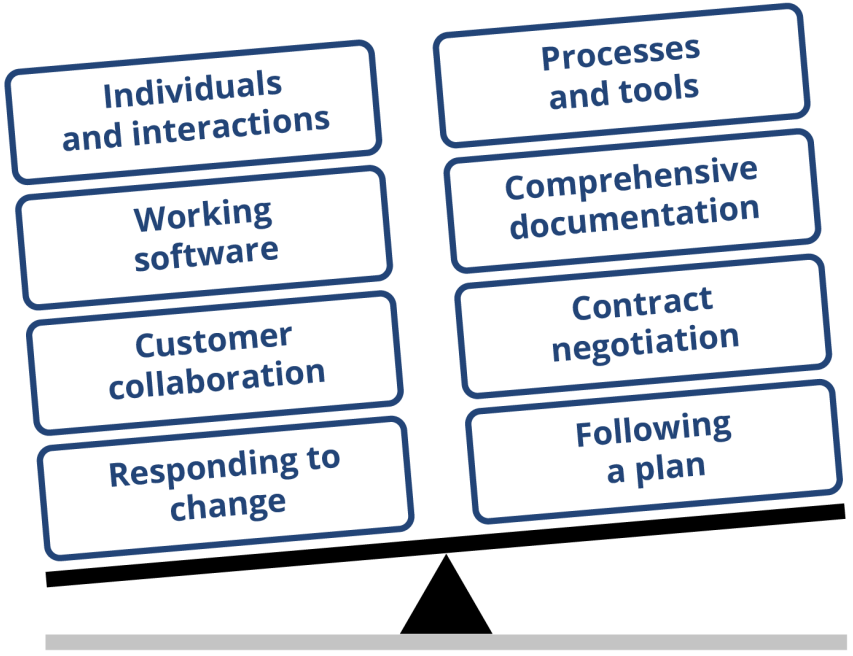
Source: Palmquist, Steve, et al. Parallel Worlds:

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

2001: 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.

Common myth

The manifesto is often misinterpreted to mean:

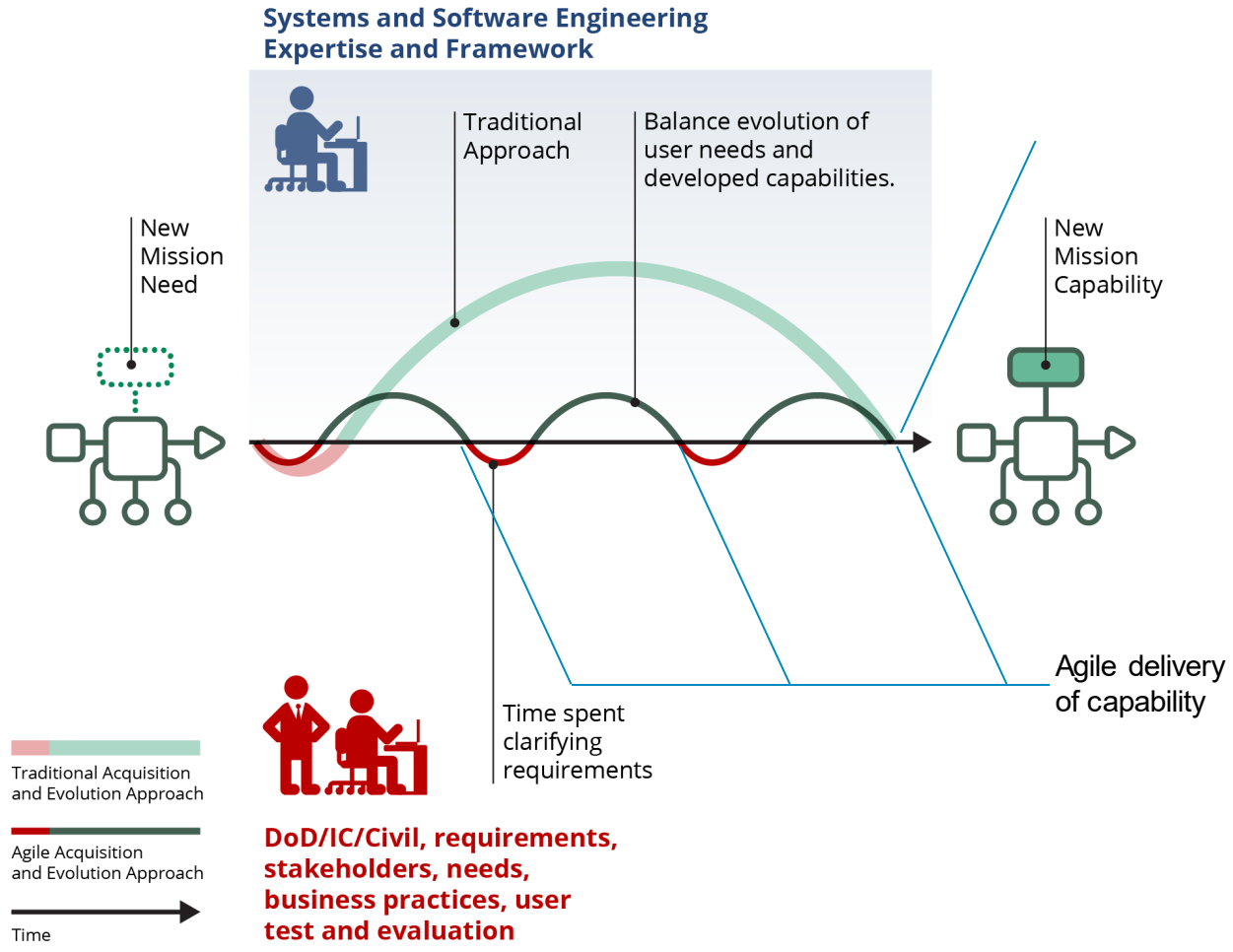
"no documentation, no process, and no plan"!

© 2023 Carnegie Mellon University

https://agilemanifesto.org/

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Agile vs. Traditional Software Development



© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Twelve Agile Principles Support the Manifesto

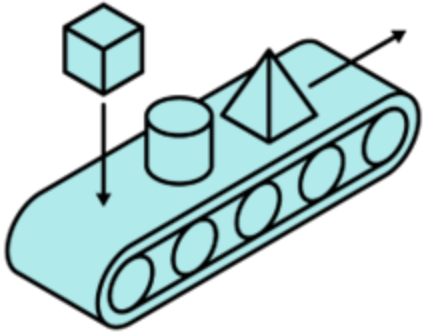
1. Highest priority is to satisfy the customer through early and continuous delivery of software
2. Welcome changing requirements, even late in development
3. Deliver working software frequently, from a couple of weeks to a couple of months
4. Business people and developers must work together daily throughout 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.

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

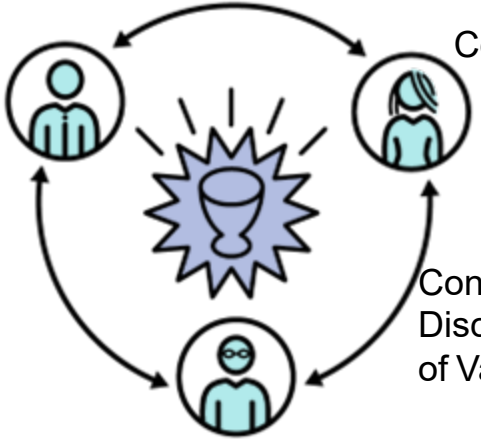
Agile Represents a Major Requirements Transition – Different Acquisition Objectives

Buying a Box



Buying an ongoing delivery stream

"Do This"



Constant Collaboration

Continuous Discovery of Value

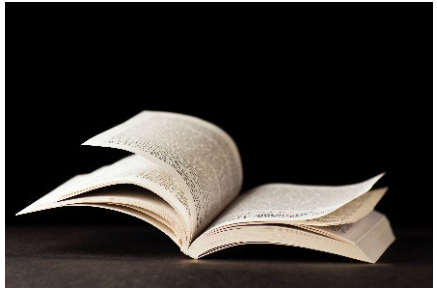
© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

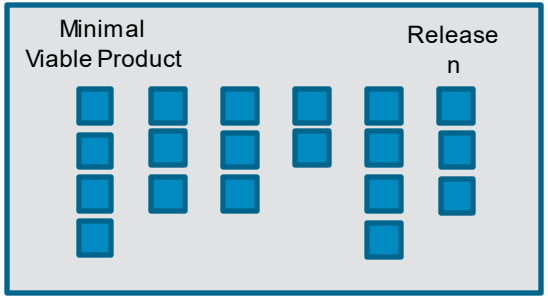
A Different Approach to Software Requirements…

Traditional: Thorough Reqts Analysis and Decomposition – Relatively static reqts document.

Agile: Expect changes and re-prioritization as you go and learn more about the real needs and mission space.



High-level requirements
Periodic updates - based upon user feedback, evolving mission / threats, evolving technologies…



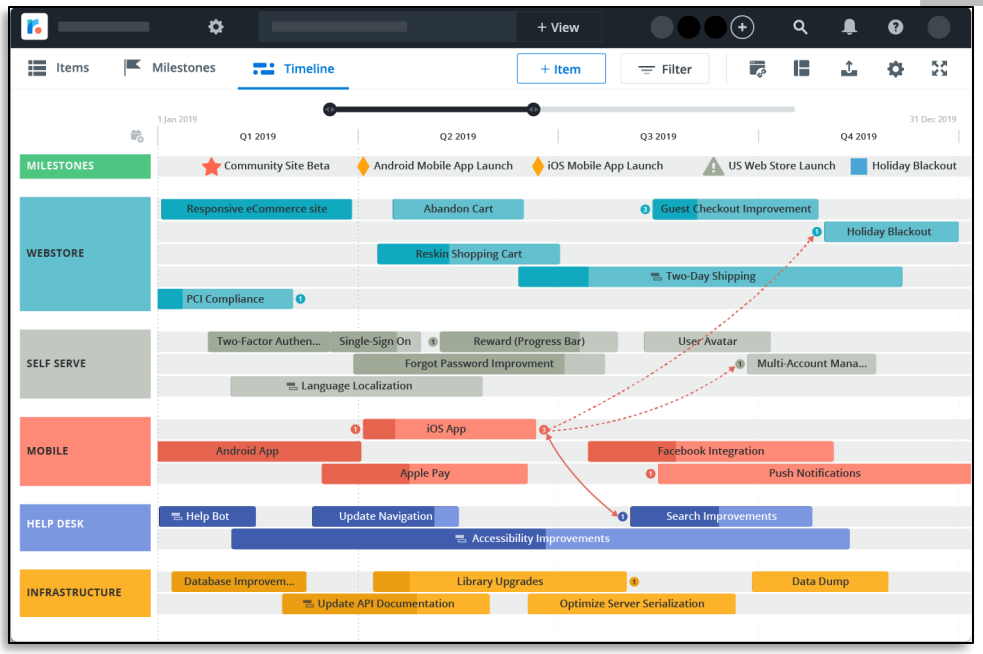
Detailed "requirements"
Backlog of prioritized work to be done

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Planning and Tracking: The Product Roadmap

- High-level visual summary that maps out the vision and direction of product offerings over time
- Describes the goals and features of each software iteration and increment
- Allowed to evolve – but retain traceability to major capability deliveries

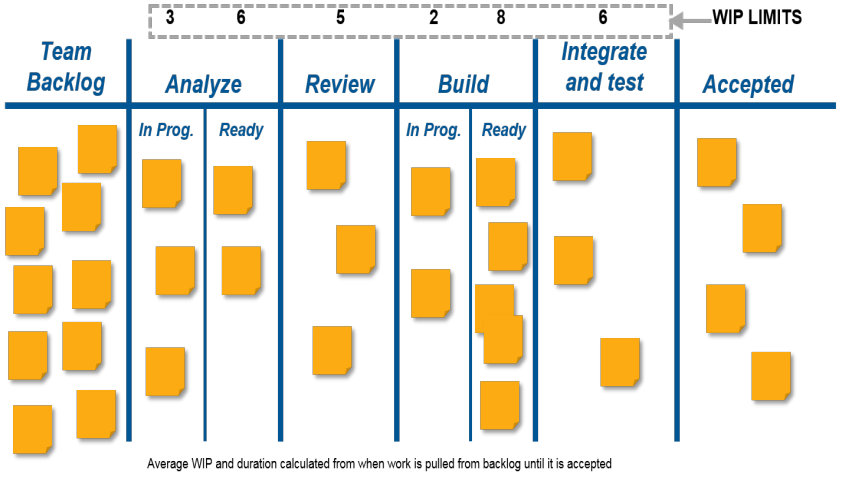


- **Tracking** needs to take into account progress against roadmap + delivered value – *not* conformance to plan.
  - Congress / GAO focused on frequency of capability delivery

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Illustrative Example - Kanban



Defined States the Work Progresses Through

Rules about Limiting Work in Process for Each State

Definitions of “Special” Classes of Tasks (Due Date, Expedited, or others as defined by the project)

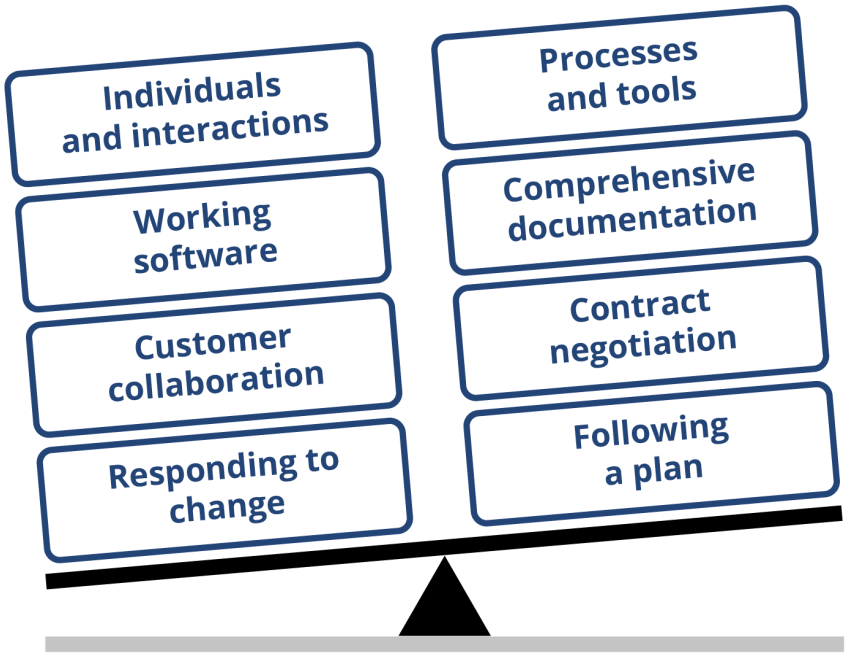
Explicit Acceptance Criteria

Useful for Tasks that Don’t Easily Conform to a Time Box

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

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



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)

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

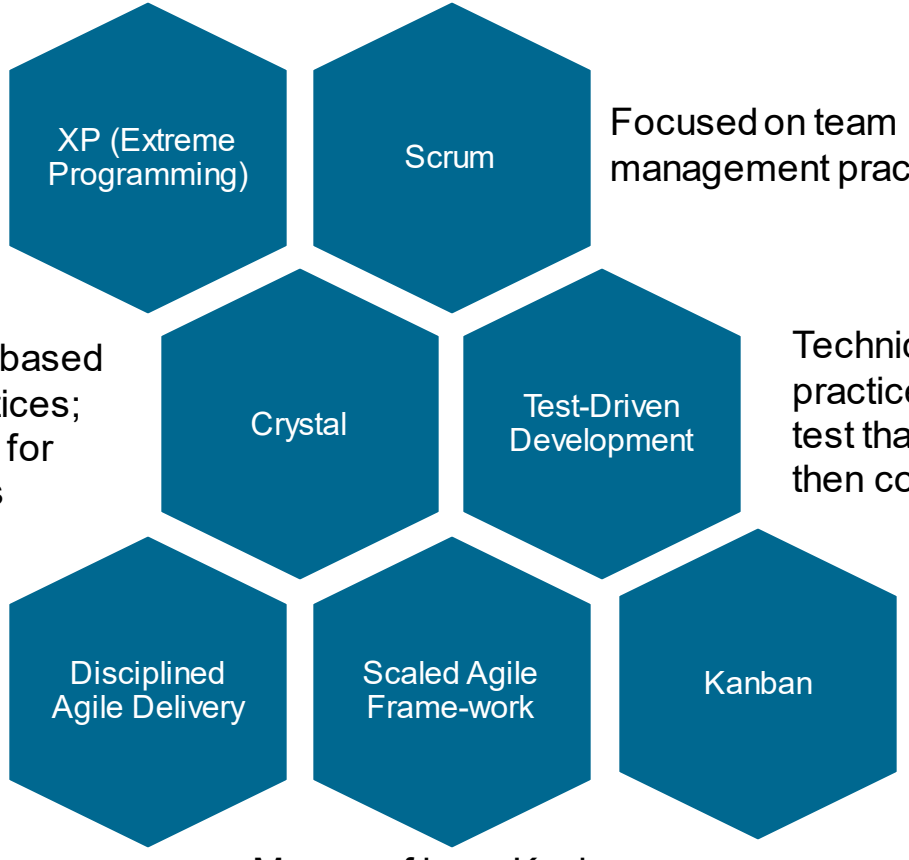
Agile Software Development – Tech 101

Beyond Buzzwords: What Does Agile Software Look Like in DoD?

Carnegie Mellon University
Software Engineering Institute

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Many Methods Generally Termed "Agile"



Focused on team technical practices

Focused on team management practices

Encourages risk-based selection of practices; different patterns for different contexts

Technical and management practices focused on writing the test that proves acceptance, then coding to that

Originally derived from Rational Unified Process, designed to scale

Pull-based approach particularly favored for services like security, systems engineering

Merger of lean, Kanban, and other Agile methods to support large scale projects

There are multiple ways to achieve agility – each team will need to select and tailor approach to its needs

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Defense Innovation Board "Agile BS Detector"

The DIB SWAP team identified some key "red flags" that a project is not really Agile, whatever buzzword they may be using.

- The software team doesn't have someone talking with / observing *actual users* in action.
- The software team doesn't have access to continuous ongoing feedback from users (e.g., bug reports, user assessments).
- The team prioritizes meeting requirements over getting something useful into use quickly.
- Different roles on the program act autonomously / stove-piped. ("It's not my job" is a bad sign.)
- Manual processes that could be automated are tolerated.

https://media.defense.gov/2019/May/02/2002127286/-1/-1/0/DIBGUIDEDETECTINGAGILEBS.PDF

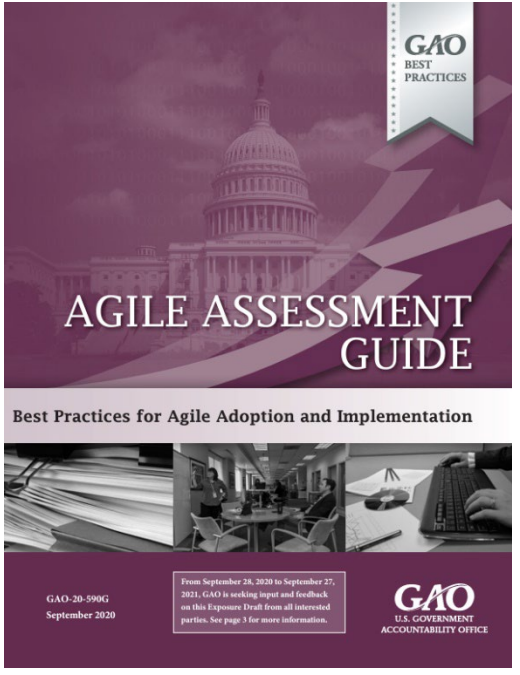
© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

GAO Agile Software

To aid federal agencies in assessing readiness to adopt Agile methods, GAO compiled a set of best practices (which tie to questions for the auditor).

- Team dynamics
  - Team composition supports Agile methods
  - Work is prioritized to maximize value for the customer
  - Repeatable processes are in place
- Program operations
  - Staff are appropriately trained in Agile methods
  - Technical environment enables Agile development
  - Program controls are compatible with Agile
- Organization environment
  - Organization activities support Agile methods
  - Organization culture supports Agile methods
  - Organization acquisition policies and procedures support Agile methods

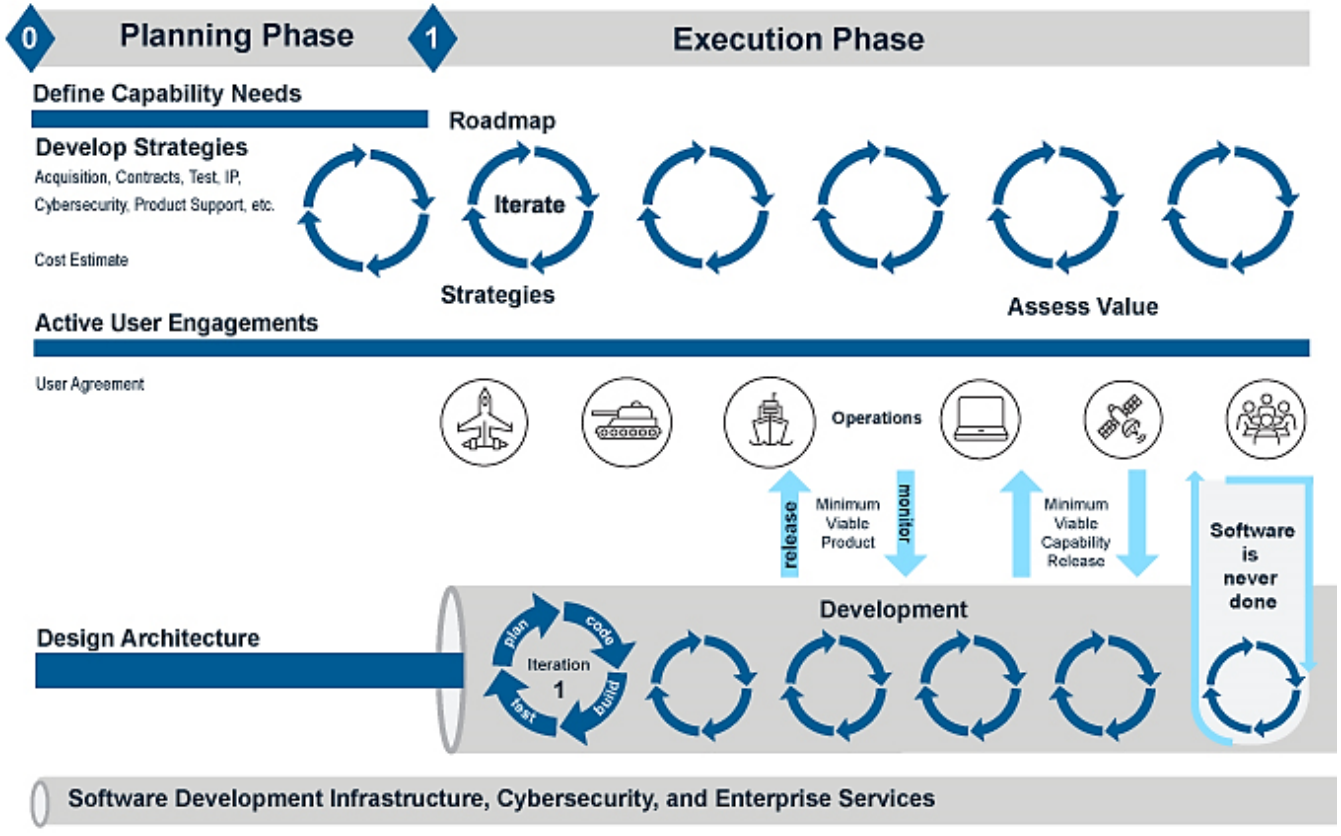


https://www.gao.gov/assets/gao-20-590g.pdf

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

DoD SW Acq. Pathway: Built on Agile Principles



- **Rapid and iterative delivery to the operational environment to meet the highest priority user needs**
- Use of modern software development practices (Agile, DevSecOps, Lean)
- Capitalizes on active user engagement and enterprise services
- Tightly coupled mission-focused government-industry software teams
- Automated tools for development, integration, testing, certification

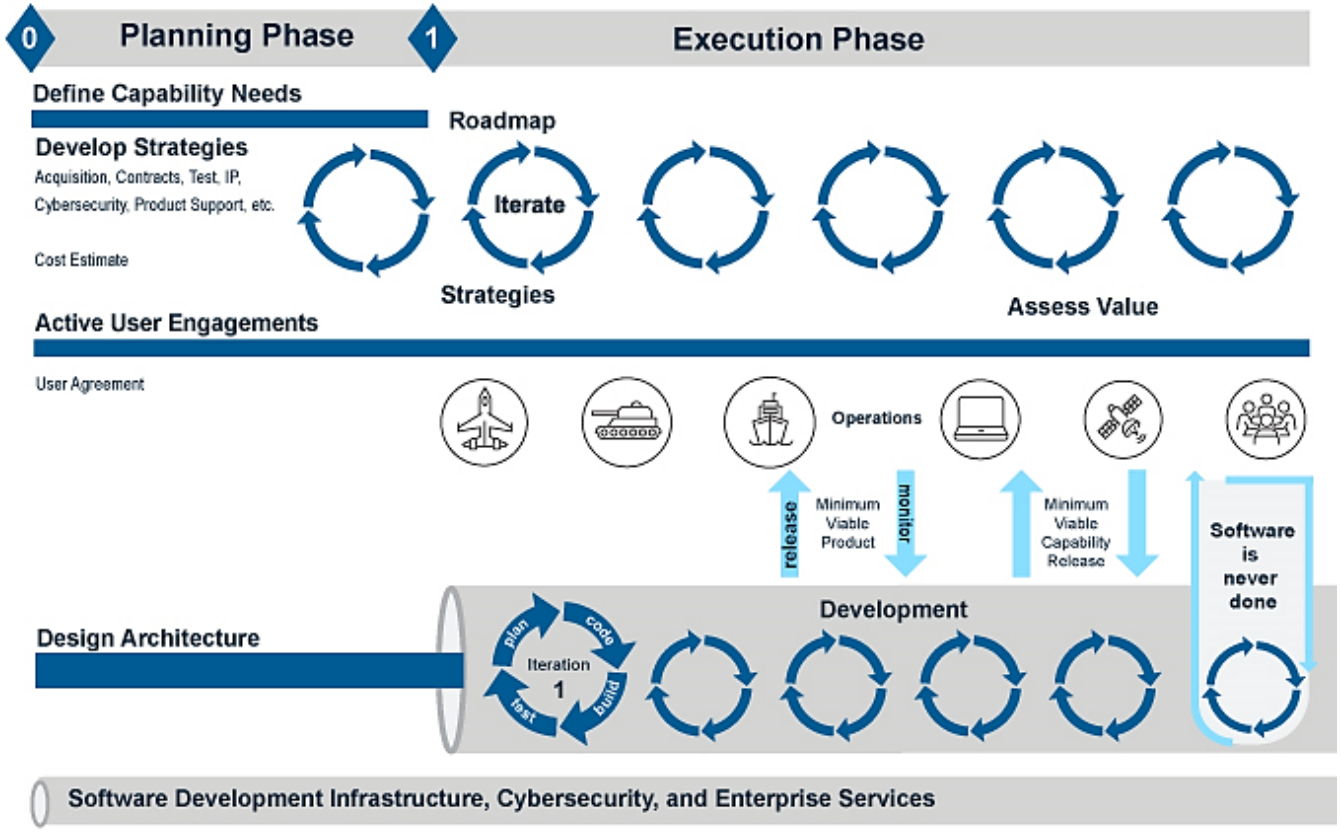
Source: DODI 5000.02 Section 4.2

https://aaf.dau.edu/aaf/software/

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

SW Acq. Pathway: A Substantial Departure for DoD



Agile Principle: "Working software is the primary measure of progress"

SWP removes focus on process, measures progress instead based on whether software is produced & deployed, and its impact.

"New capabilities will be **delivered to operations at least annually** to iteratively meet requirements, but more frequent updates and deliveries are encouraged where practical."

"**Value assessments** will be performed at least annually… to determine if the mission improvements or efficiencies realized from the delivered software are timely and worth the current and future investments from the end user perspective."

https://aaf.dau.edu/aaf/software/

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Where Things Are Today

- Evidence-based policy in place (DoDI 5000.87), based on:
  - Agile pilots (2018 NDAA Sec 873 & 874)
  - Department early adopters & thought leaders
- Increasing numbers of programs adopting the software pathway (SWP)
- DoD is expanding the amount of guidance and direction available
  - Specific sub-paths for embedded, application, and business software
- GAO has found much to like* and would like to see some aspects of the SWP applied more broadly
  - e.g., GAO recognizes SWP for addressing key principles of “employing right-sized teams,” “using Agile development to promote iteration,” “prioritizing schedule throughout development by off-ramping capabilities when necessary”

https://www.gao.gov/assets/gao-22-104513.pdf

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Some Key Challenges

- Culture change – from top-down to delegated decision-making
- Work to be done to ensure that other acquisition policies and expectations are updated / applied consistently with Agile approaches
- Resourcing – different patterns of when expertise is needed available
  - e.g., testers being available “early and often” versus for specific test events
  - e.g., expectations for frequent and consistent user involvement
- Knowledgeable government product owners
  - Shift to prioritizing requirements from “it’s all important”

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Agile Software Development – Tech 101

Agile Myths & What To Watch Out For

[[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Carnegie Mellon University
Software Engineering Institute

Watch Out For: Technical Debt

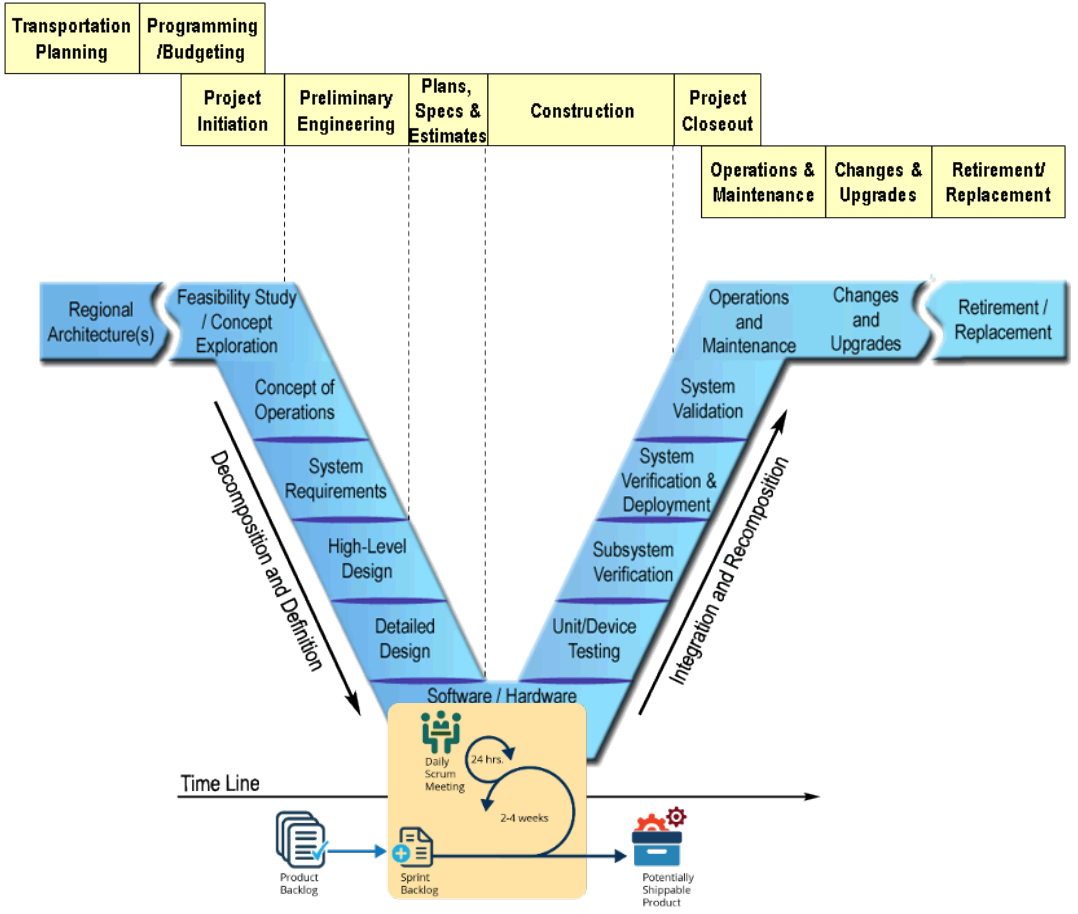
Technical Debt (TD): “an element of design or implementation that is expedient in the short term, but that would result in a technical context that can make a future change costlier or impossible”

- Strategically taking on TD allows helpful tradeoffs to be made, like getting working software in front of users and receiving feedback faster.
- Allowing TD to pile up on the backlog (or not keeping track) over time slows the rate at which future changes can be made.

Since addressing TD is often not visible to the end users, software teams can be incentivized to prioritize new capabilities at the expense of dealing with TD – but this is not sustainable.

© 2023 Carnegie Mellon University
[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Watch Out For: "WaterScrumFall"



Source: Palmquist, Steve, et al. Parallel Worlds:

This isn't enough.

Optimizing one part of the process:

- Doesn't optimize the whole process
- Simply exposes roadblocks by other parts of the process

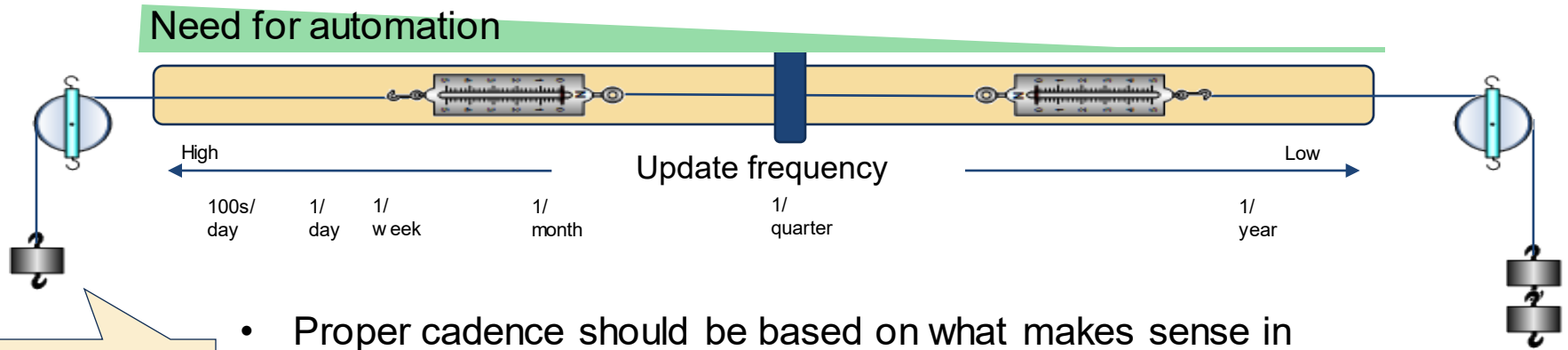
"Agile at the bottom of the V" loses benefits of agility:

- Too many decisions are made too early
- No learning opportunities

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Myth: The Right Cadence is Weekly [Monthly, Daily, …]



Forces such as:
- User demand
- Competitiveness
- Automation

- Proper cadence should be based on what makes sense in context.
- Considerations within DoD:
  - The need for OT, ATO, certifications, etc., add to cycle time – With effort, this can be reduced.
  - But in some cases tactics and training need to be updated… This depends on user needs / expectations in context.

Forces such as:
- System size
- Complexity
- Regulations

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Myth: Agile Means “No Documentation”

The Agile Manifesto values “working software” over “comprehensive documentation” – but that doesn’t mean documentation has ***no*** value.

- Consider where documentation can add value.
- Make the tasks visible parts of work planning.
- Manage the tradeoffs.

SWP carries this through to acquisition documentation as well.

- High-level strategy documents are preferred over detailed plans.
- Keep documents short and easy to update as the team learns more and improves planning.

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Myth: Agile Means “No Data”

Agile programs can generate more data than traditional approaches – but not all of it is easy to interpret out of context.

- Every iteration is an experiment – you should be able to determine how well it succeeded.
- Continuous monitoring can potentially collect many points of data about quality.
- There should be multiple opportunities to collect user feedback / user evaluations.
- Multiple deployments over shorter time windows provides many opportunities to collect data from use.

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Myth: Agile Means "No Architecture"

Some authors warn Agile teams away from "Big Design Up Front" (BDUF)

- Rationale: Teams can spend a lot of time in the beginning analyzing requirements and building an architecture that is meant to be robust and easily changeable – but are usually bad at predicting the types of changes that will likely need to be accommodated.

However, building an architecture that will be easily adaptable or changeable for a long system lifetime is not trivial.

- This is a task that requires experience and expertise.
- SWP includes a Planning Phase to give programs time to experiment with prototypes, initial capabilities, and architecture options.
- Some styles are more suited than others (pub-sub, MOSA, microservices…), but every style is not suited for every problem.

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Myth: It Doesn't Work for Safe / Secure / Critical /… Systems

Sample of reported outcomes on DoD/federal programs:

- Quantifiable cost savings and 6-month early delivery
- Significant cost avoidance
- Reduced rework & unplanned releases
- Dramatically increased productivity/capacity, with reduced cost of delivery
- Improved insight into contractor performance and progress
- Early discovery & resolution of Cat 1 defects (one year prior to integration test event)
- Early discovery & resolution of interface issues
- Improved flight test efficiency
- Early insight for end users into functionality of delivered system
- Better responsiveness to users with rapidly fluctuating requirements
- Heightened awareness & collaboration, improved realization of tradeoffs
- Improved workflow management

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Contact Information

Dr. Forrest Shull
Lead, Software Acquisition Process Research

Email: fjshull@sei.cmu.edu

Web
www.sei.cmu.edu
www.sei.cmu.edu/acquisition/research

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

Customer Relations
Email: info@sei.cmu.edu
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.

Selected Additional Resources

MITRE's Acquisition in the Digital Age (AiDA): http://aida.mitre.org

SEI Agile Adoption in Government Blog Series: https://insights.sei.cmu.edu/sei_blog/agile-adoption-in-government/

Webinar: Practical Considerations for Adopting Agile/Lean in Government: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=502861

Webinar: Five Keys to Effective Agile Test Automation for Government Programs: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=503507

Webinar: Three Secrets to Successful Agile Metrics: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=507850

Agile in the DOD Podcast Series: http://www.sei.cmu.edu/podcasts/agile-in-the-dod/index.cfm

Podcast: How Risk Management Fits in to Agile & DevOps In Government: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=514190

UPDATE

Scaling Agile Methods for Department of Defense Programs: http://resources.sei.cmu.edu/asset_files/technicalnote/2016_004_001_484647.pdf

Agile Metrics: Progress Monitoring of Agile Contractors: http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=77747

RFP Patterns and Techniques for Successful Agile Contracting: http://resources.sei.cmu.edu/asset_files/specialreport/2016_003_001_484063.pdf

Contracting for Agile Software Development in the Department of Defense: An Introduction http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=442499

Written Testimony of the Software Engineering Institute's Agile in Government Team to House Ways and Means SSA Subcommittee http://waysandmeans.house.gov/wp-content/uploads/2016/07/20160714SS-Testimony-Hayes.pdf

Agile Methods: Selected DoD Management and Acquisition Concerns: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9769

© 2023 Carnegie Mellon University

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution.