National Defense Industrial Association
Integrated Program Management Division

Agile Framework Overview

IPMD Agile Committee

Content & Learning Objectives

1. The Problem Space
2. Recent History of Agile
3. What is a Mind-set?
4. The Agile Mind-set
5. Why Agile?
6. Shift from Waterfall to Agile
7. Agile Planning Artifacts
8. Agile Roles
9. Agile Ceremonies
10. Agile Performance Artifacts
11. Iterative Systems Engineering
12. Agile Inputs Checklist
13. Conclusion

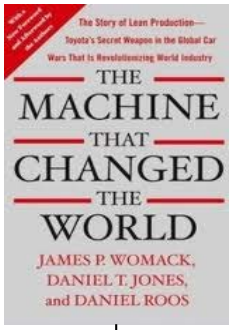
The Problem Space

- Threats that the United States faces are changing at an ever increasing pace, and the Department of Defense's (DoD's) ability to adapt and respond is now determined by its ability to develop and deploy software to the field rapidly.[1]
- To maintain advantage, DoD needs to procure, deploy, and update software that works for its users at the speed of mission need, executing more quickly than our adversaries. Statutes, regulations and cultural norms that get in the way of deploying software to the field quickly weaken our national security and expose our nation to risk. [1]
- This presents a challenge to PMs – what tools are available to deliver the right capabilities to the warfighter faster?
- The Agile Framework is a mechanism being utilized to meet the challenge
- This presentation provides an overview of the Agile Mind-set and essential implementation characteristics of the Agile framework

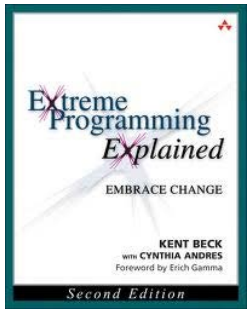
The War-fighter Requires Capabilities sooner
The Agile Framework addresses this Challenge

[1] Defense Innovation Board (DIB), SW Acquisition and Practices (SWAP) Study Report [Draft], 3/13/2019

Recent History Of Agile



Term “Lean” used to describe Toyota Production System



Kent Beck publishes *Extreme Programming Explained*

1990

1999

1995

OOPSLA '95

Schwaber & Sutherland present on Scrum

2001

Leading SW practitioners Coin term “Agile” and publish the “Agile Manifesto”

Understanding how Agile became formalized

What is a Mind-Set? [1]

- Your Mind-set is how you think about acting in a given situation – it's the thinking behind your actions.
- Mind-set can be viewed as composed of Values, Beliefs and Principles
  - Values: What you consider **most important** in a given situation
  - Beliefs: What you hold to be **true** in that type of situation
  - Principles: The **standards** that guide your choices, decisions and actions. Principles are based on your Values and Beliefs
- Example: Mind-set for writing a book on playing tennis
  - Values: it's important to me that my book be easily understood and help the reader, whether a novice or expert, improve their tennis game
  - Beliefs: I believe that many how-to books are overly technical. I believe my readers would prefer a book that describes tennis simply, is supportive and not condescending
  - Principles: Therefore my writing principles include writing simply and conversationally, progressing gradually from basic to more complex techniques, and including examples and pictures to enhance understanding.

Mind-set – The Thinking Behind Your Actions

[1] *The Agile Mind-Set*, Gil Broza, 2015

The Agile Mind-set: Values

Manifesto for Agile Software Development[1]

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to **value**:

Individuals and Interactions over processes and tools

Working Software over comprehensive documentation

Customer Collaboration over contract negotiation

Responding to Change over following a plan

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

Agile Values relate to the Team, the Customer and the Work and are based on experience with successful projects

[1] agilemanifesto.org

The Agile Mind-set: Beliefs[1]

- People:
  - Competent, motivated, trusted, and supported people do well
  - As human beings, people will get some or even many things wrong
  - The best model that manages the downside and elevates the upside is the self-organizing, collaborative team
- Customer:
  - Customers can't - and, being adaptive, shouldn't – pinpoint future needs and wants
  - Therefore, the sensible thing to do is focus intently on what the customer needs now
  - Knowing the top needs and fulfilling them is being effective, which matters more than being efficient
- The Work:
  - For complex work, emergence or evolution is an appropriate response to complexity
  - The best enabler of emergence is the short feedback loop
  - Since feedback, emergence and adaptation imply frequent change, the cost of change can remain low. When this isn't the case, Agile will probably will not be a good fit.

- Agile practitioners adhere to these beliefs, not because they have been formally proven, but because they see enough compelling evidence for the validity.

Agile beliefs are based on compelling evidence from successful projects

[1] *The Agile Mind-Set,* Gil Broza, 2015

The Agile Mind-set: Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
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. The sponsors, developers, and users should be able to maintain 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.

Agile Principles derive from Agile Values and Beliefs

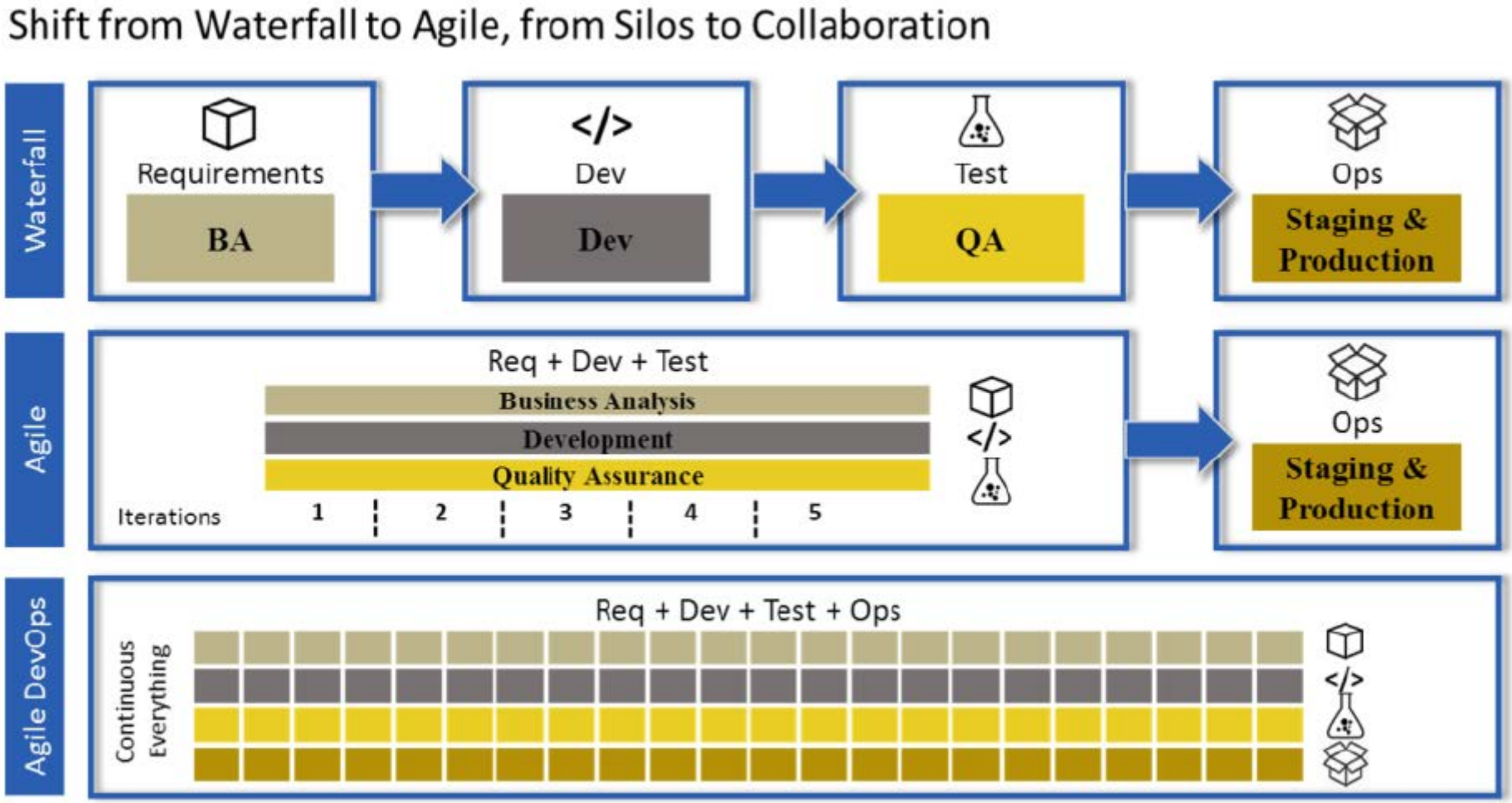
Why Agile?

- The main benefit of iterative development — the ability to catch errors quickly and continuously, integrate new code with ease, and obtain user feedback throughout the development of the application — will help the DoD to operate in today's dynamic security environment, where threats are changing faster than Waterfall development can handle.[2]
- All software procurement programs should start small, be iterative and build on success – or be terminated quickly. [3]

Iterative development is needed to operate in today's dynamic security environment

[2] Defense Science Board, Design and Acquisition of Software for Defense Systems, February 2018
[3] Defense Innovation Board, Ten Commandments of Software, April 2018

Shift from Waterfall to Agile



Defense Science Board, Design and Acquisition of Software for Defense Systems, February 2018, Figure 3

Shift in Emphasis from Phase Completion to Capability Delivery

Agile Planning Artifacts

- Product Backlog
  - The master list of all functionality at the Capability* and Feature level that is desired in the product; prioritized from most to least important
  - Definition of Done: complete as mutually agreed to by all parties and conforming to an organization's standards, conventions and guidelines
- Product Roadmap
  - Time-phased delivery plan for the functionality in the Product Backlog
- Agile Implementation Plan
  - Engineering Development Plan
  - Agile Cadence (sprint, increment or release, spin)
  - Agile Roles Defined
  - Training

Agile planning discipline based on comprehensive planning artifacts

* also referred to as Epic

Agile Roles

- Product Owner
  - The person responsible for developing and maintaining the Product Backlog by representing the interests and business value of the stakeholders.
- Scrum Master
  - The person responsible for the Scrum process, making sure it is used correctly and maximizes its benefits.
  - Helps team self-organize and removes team impediments
- Development Team
  - A cross-functional group of people that develops a potentially releasable increment of “Done” product each Sprint
  - structured and empowered by the organization to organize and manage their own work

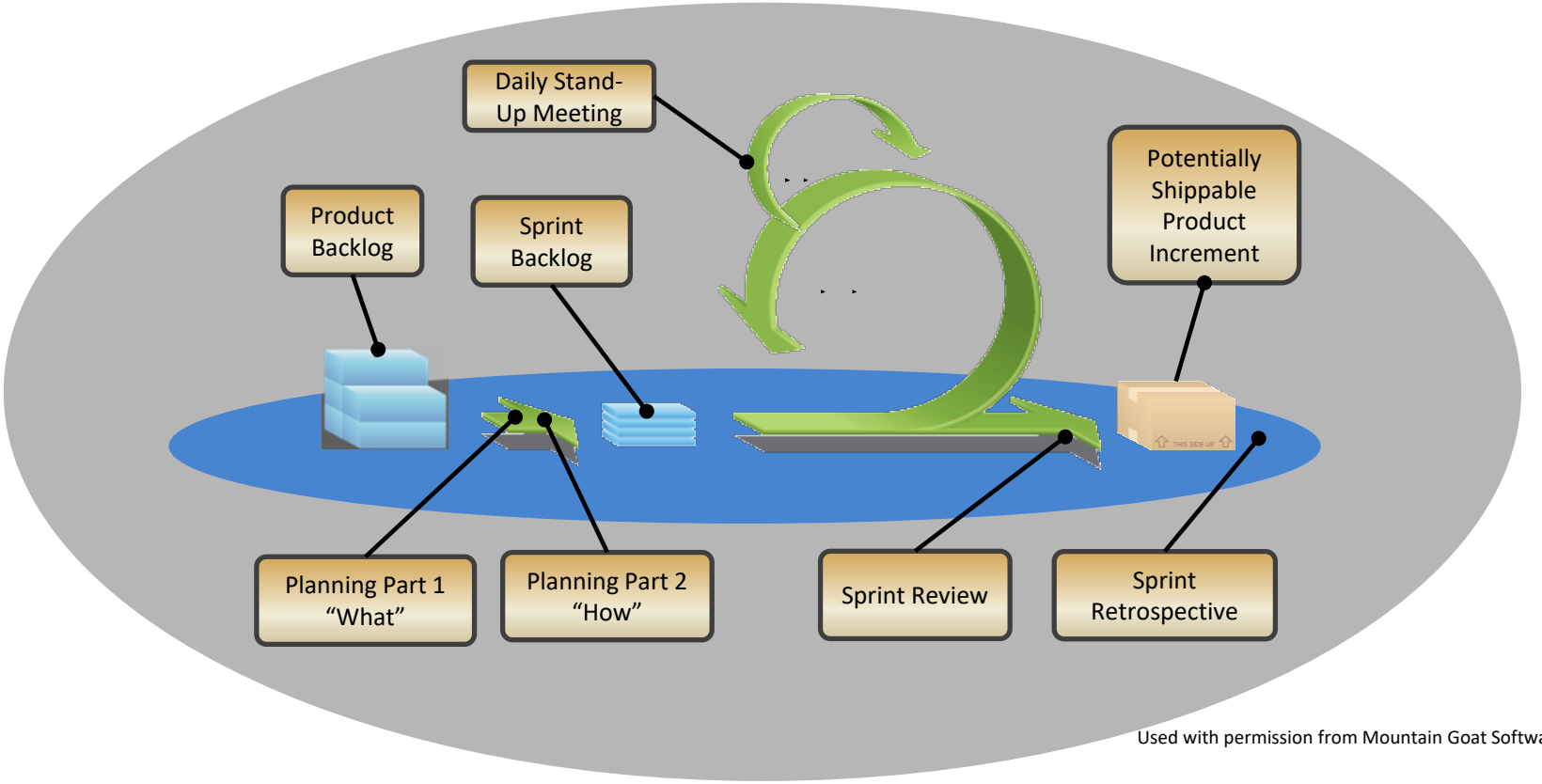
These are the typical Agile roles for program execution

Agile Ceremonies: Planning Events

Planning Level	Planning Frequency	Planning Horizon	Planning Precision	Planning Artifact
Product Planning	Project Startup Updates throughout the project	Project Duration	Capabilities Releases	Product Backlog Product Roadmap Minimum Viable Product
Release Planning	Each Cadence Release	Cadence Release	Features Stories	Product Backlog Updates Release Plan
Sprint Planning	Each Sprint	Weeks	Stories Tasks	Sprint Backlog
Daily Planning	Daily	Day	Tasks	Updated Sprint Backlog

Agile provides well-defined, periodic planning for program execution

Agile Ceremonies: Sprint Planning and Execution

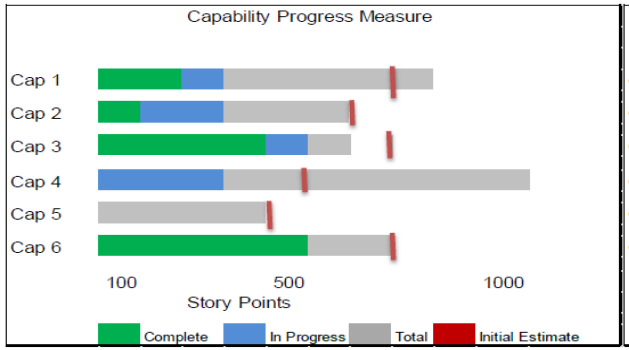


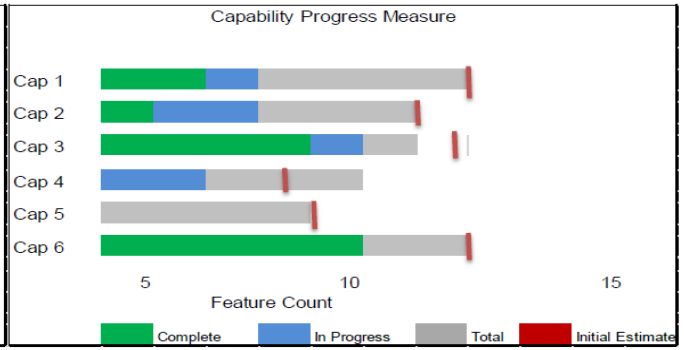
Used with permission from Mountain Goat Software

Sprints provide consistent, periodic review, delivery and feedback

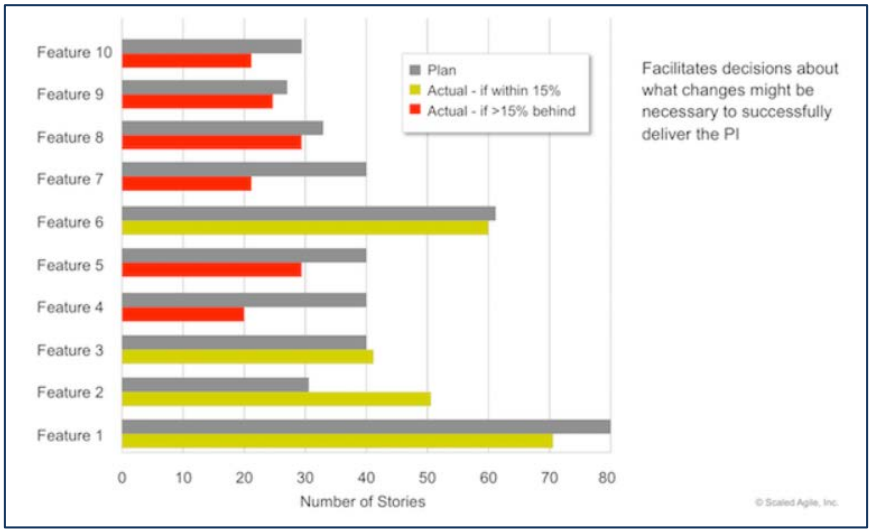
Agile Performance Artifacts

Capability Progress (Program Leadership)





1



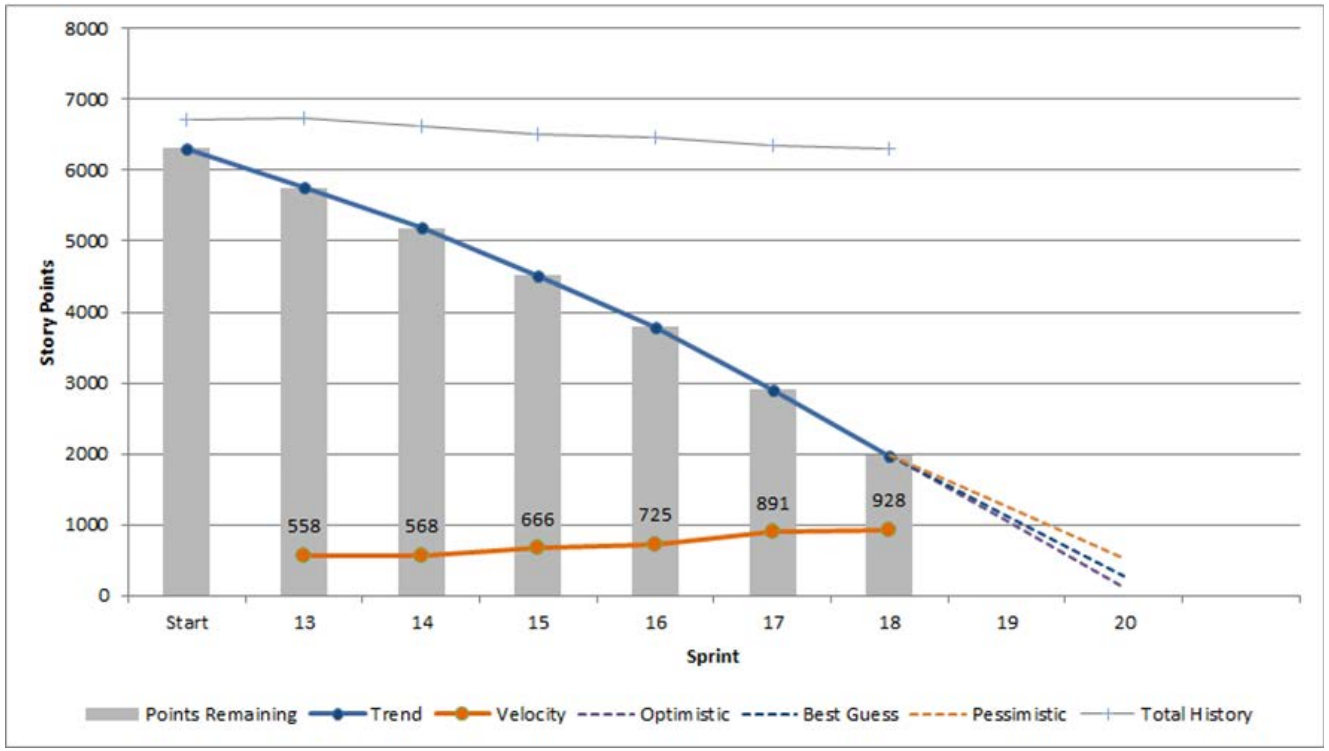
2

Use performance artifacts that make sense for your program

1 PARCA Agile and EVM PM Desk Guide 2018
2 https://www.scaledagileframework.com/metrics/#P0

Agile Performance Artifacts (cont.)

- Increment (Release) Burndown (Program Leadership)



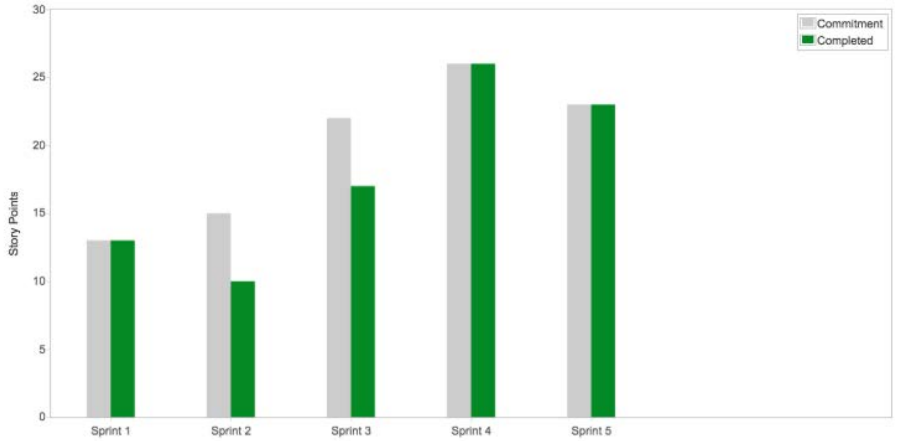
3

Use performance artifacts that make sense for your program

3 Figure B-4 http://www.ndia.org/-/media/sites/ndia/divisions/ipmd/division-guides-and-resources/ndia_ipmd_evm_agile_guide_version1_2_march262018.ashx?la=en

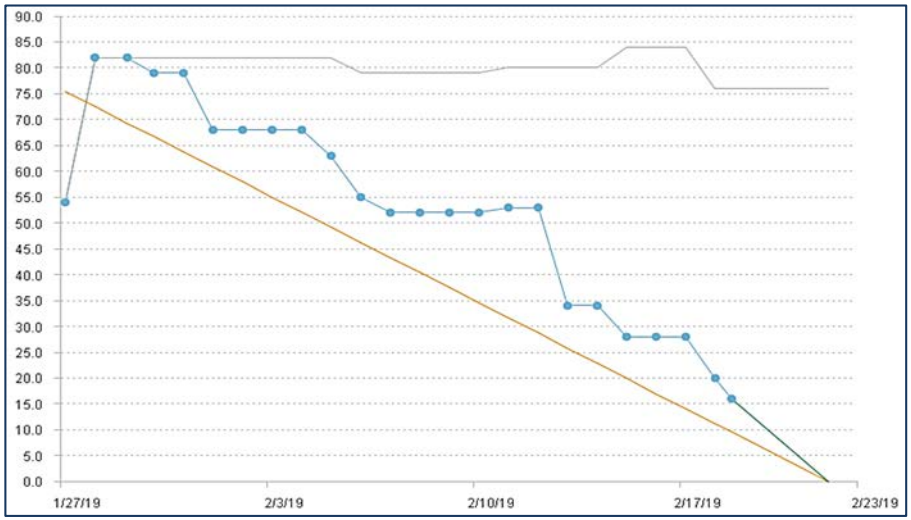
Agile Performance Artifacts (cont.)

Velocity (Team)



4

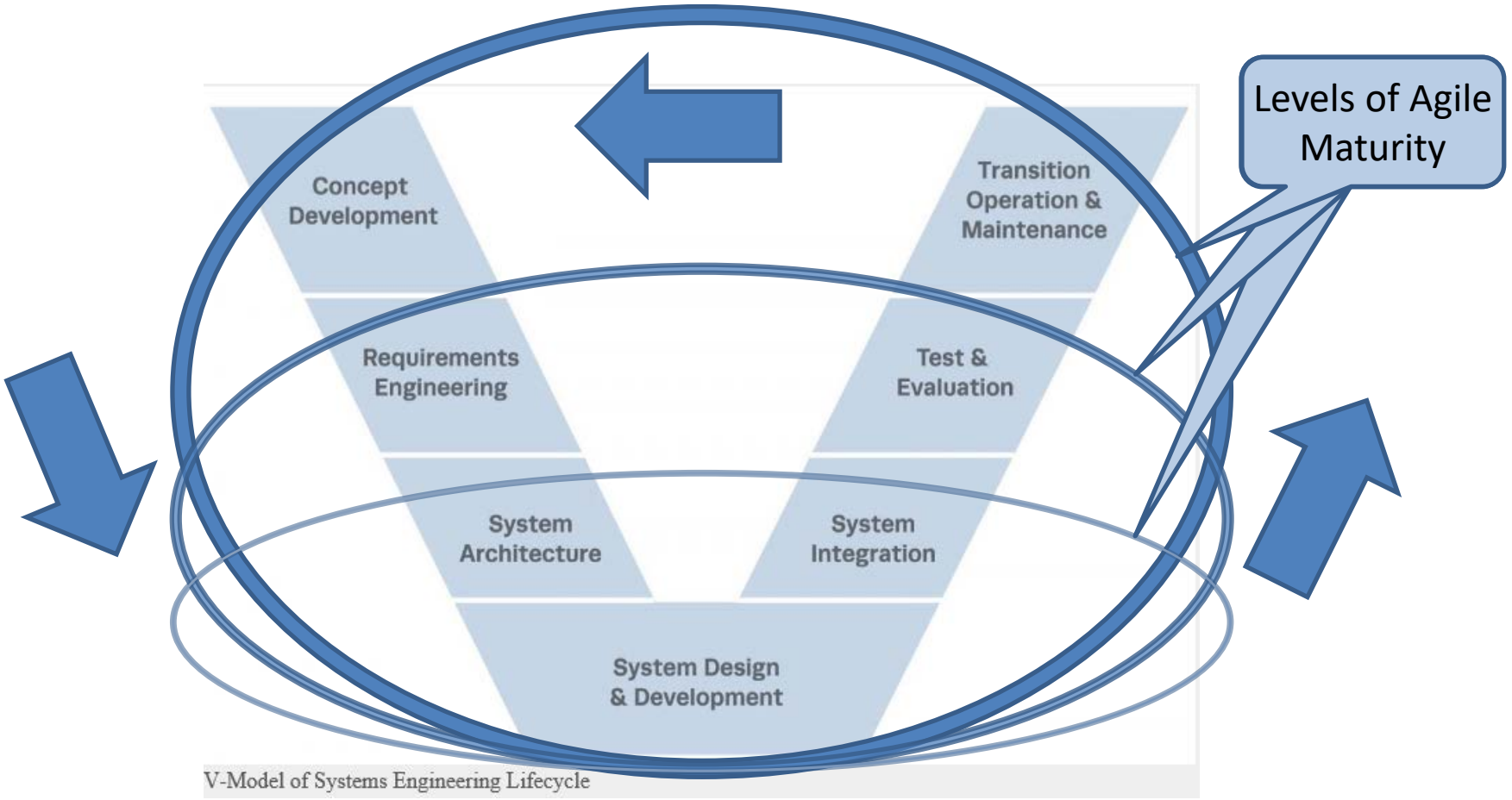
Sprint Burndown (Team)



Use performance artifacts that make sense for your program

4 https://confluence.atlassian.com/jirasoftwareserver/velocity-chart-938845700.html

Iterative Systems Engineering



V-Model of Systems Engineering Lifecycle

https://www.mitre.org/publications/systems-engineering-guide/systems-engineering-guide/the-evolution-of-systems

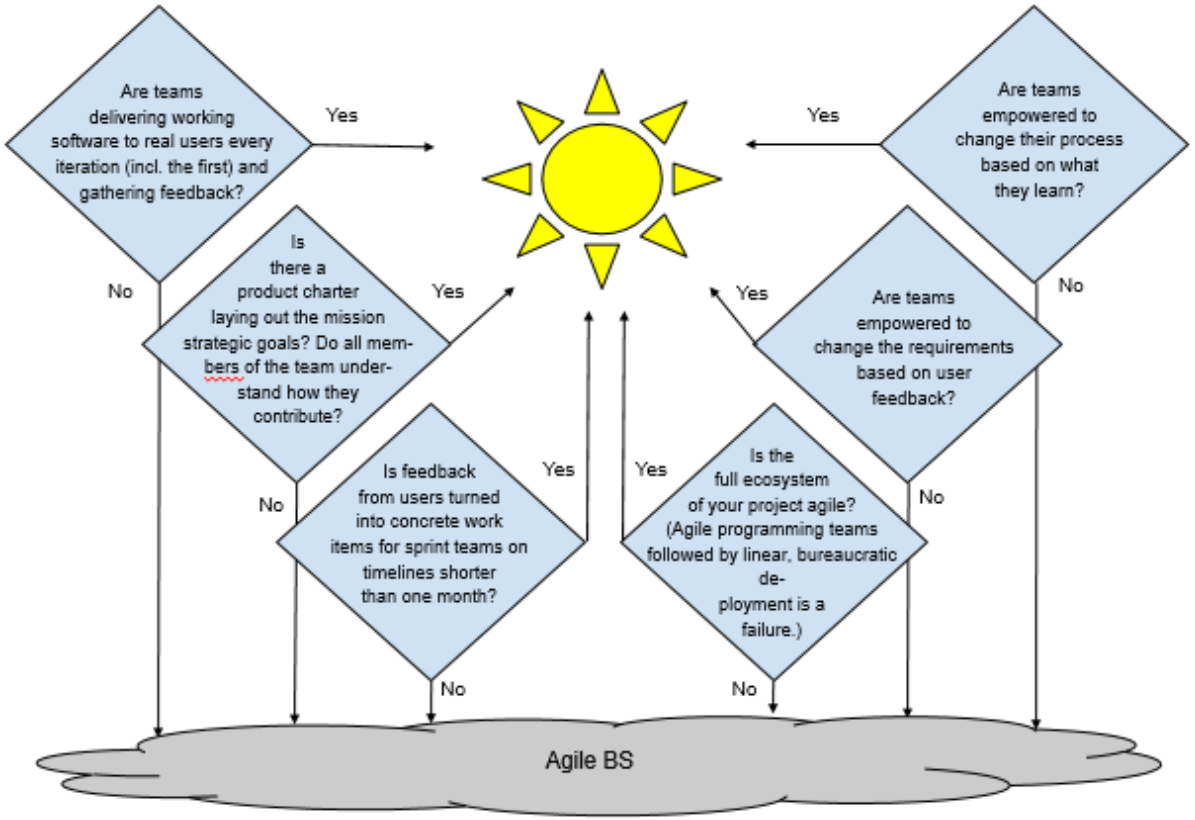
Iterative Systems Engineering Delivers Capabilities Incrementally

Agile Implementation Input Checklist

- ✓ Product Backlog (including Agile Hierarchy)
- ✓ Product Roadmap
- ✓ Agile Cadence
- ✓ Definition of Done
- ✓ Deployment or Promote to Production Plan
- ✓ Iterative Testing Approach
- ✓ Agile Metrics Identified

Planning Out Your Agile Framework and Implementation Prior to Program Startup is Essential

Conclusion



Defense Innovation Board Report: https://media.defense.gov/2018/Oct/09/2002049591/-1/-1/0/DIB_DETECTING_AGILE_BS_2018.10.05.PDF

The Agile Mind-set and Framework is People-centered Collaborative Development that Delivers Essential System Capabilities at Mission Speed