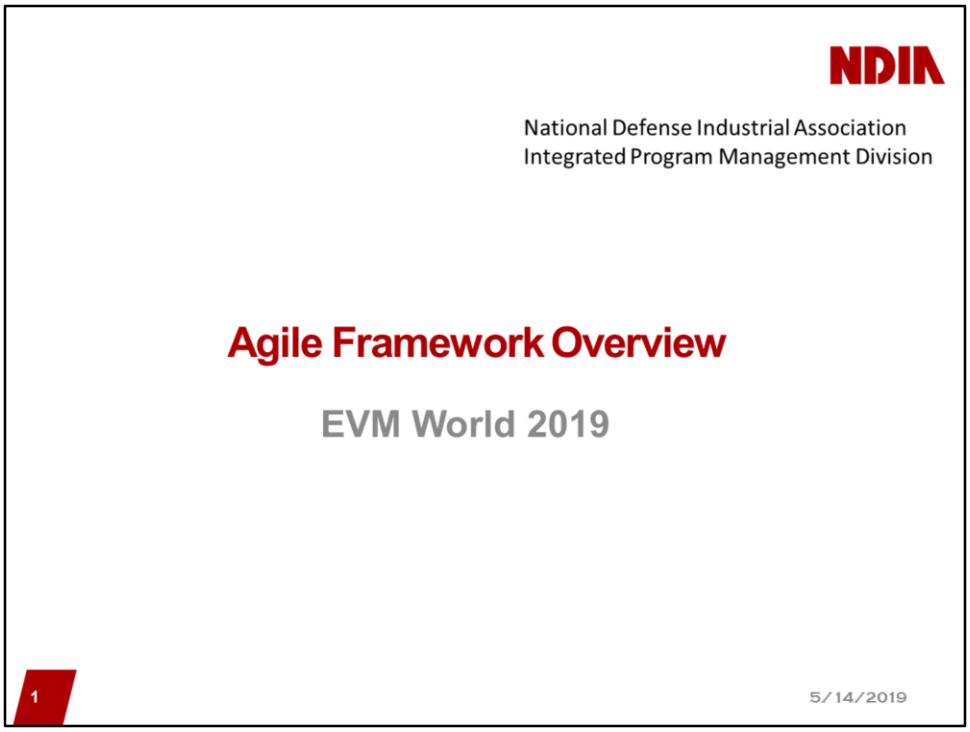
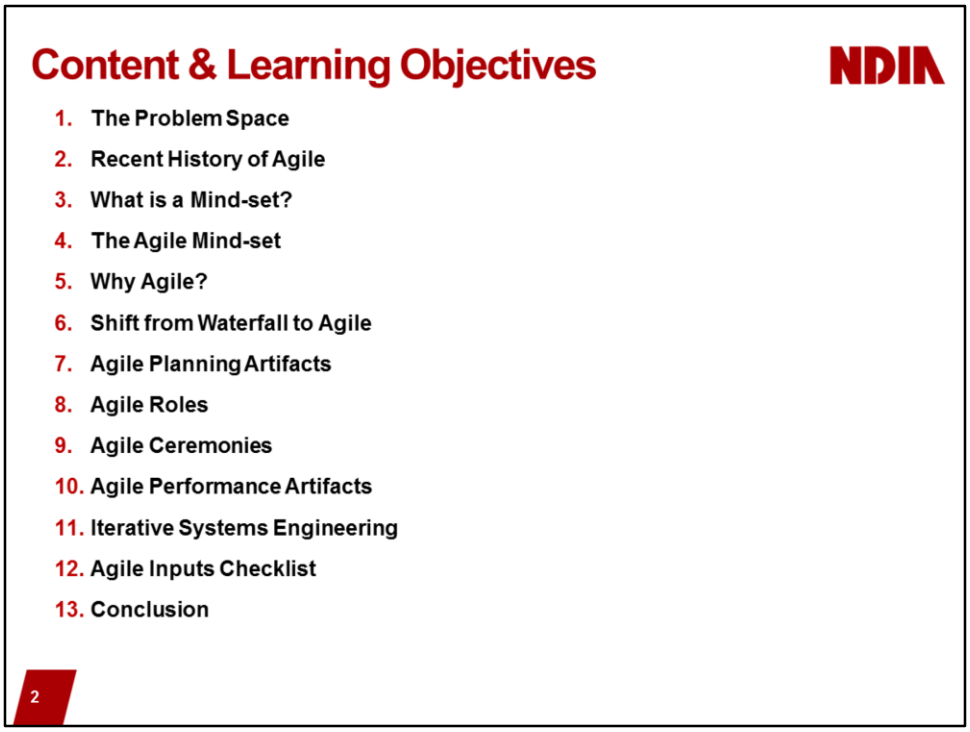



The purpose of this Overview is to provide those unfamiliar with the Agile Development Framework an overview of Agile's essential characteristics and why Agile is the right approach to meet current-day security threats faced by the warfighter.




This session will cover the listed topics. Implementing the right schedule and utilizing the right change management tools in an iterative environment has been creating challenges for our program managers. By the end of this session, we hope you have enough of an understanding of an iterative framework and a basic understand of how to implement it.

## The Problem Space

NDIA

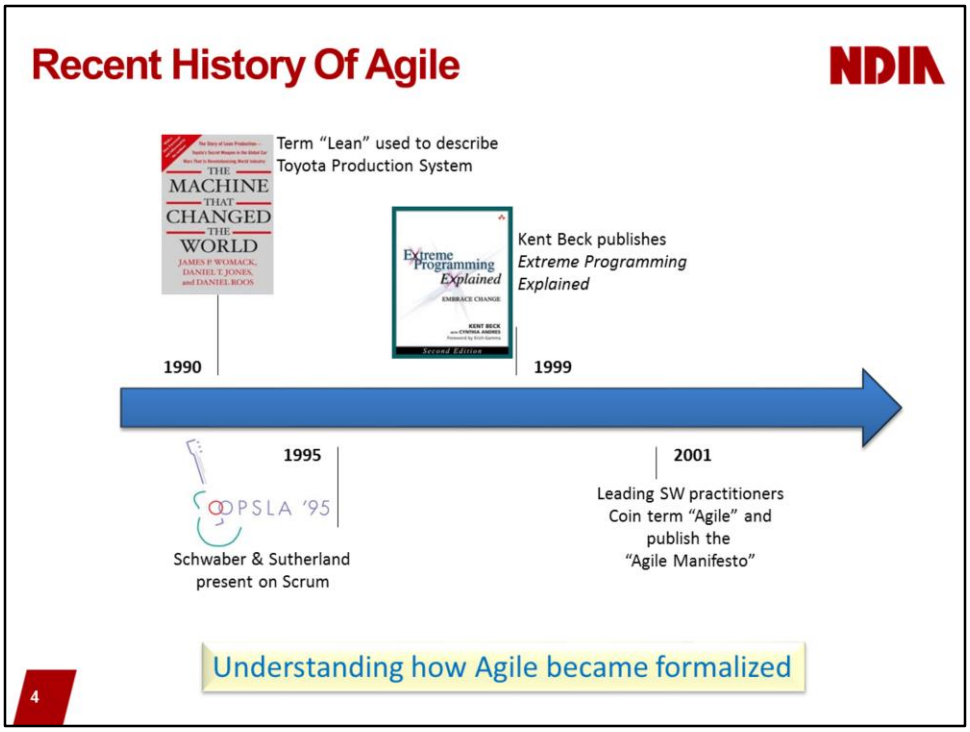
- Threats that the United States faces are changing at an ever increasing pace, and the Department of Defense's (DoD's) ability to adapt and respond is now determined by its ability to develop and deploy software to the field rapidly.[1]
- To maintain advantage, DoD needs to procure, deploy, and update 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

3

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

Beginning the discussion with an explanation of the problem space.




1990 – The machine that changed the world coins term "Lean" for TPS
1995 – Schwaber and Sutherland present on the Scrum methodology OOPSLA '95

Features such as:
- Short iterations (<=1 month) of working software
- Self managing teams

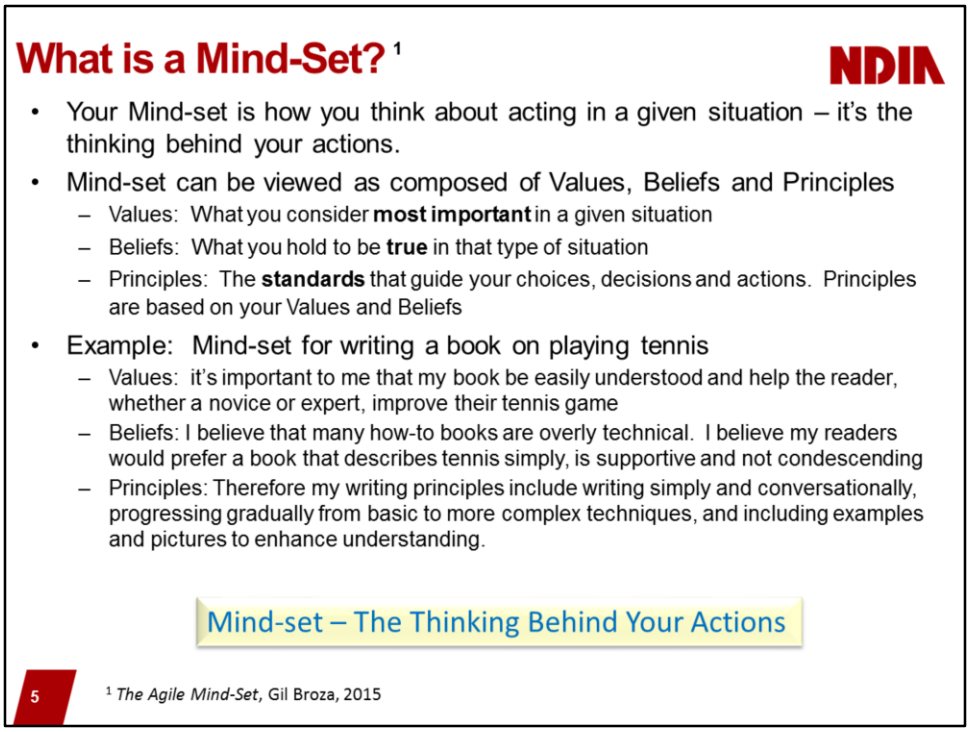
1999 – Extreme Programming Explained is published by Kent Beck (work from 1996-99)

Features such as:
- Pair Programming
- Test Driven Development
- 40 hour workweek (sustainable pace)
- Continuous Refactoring

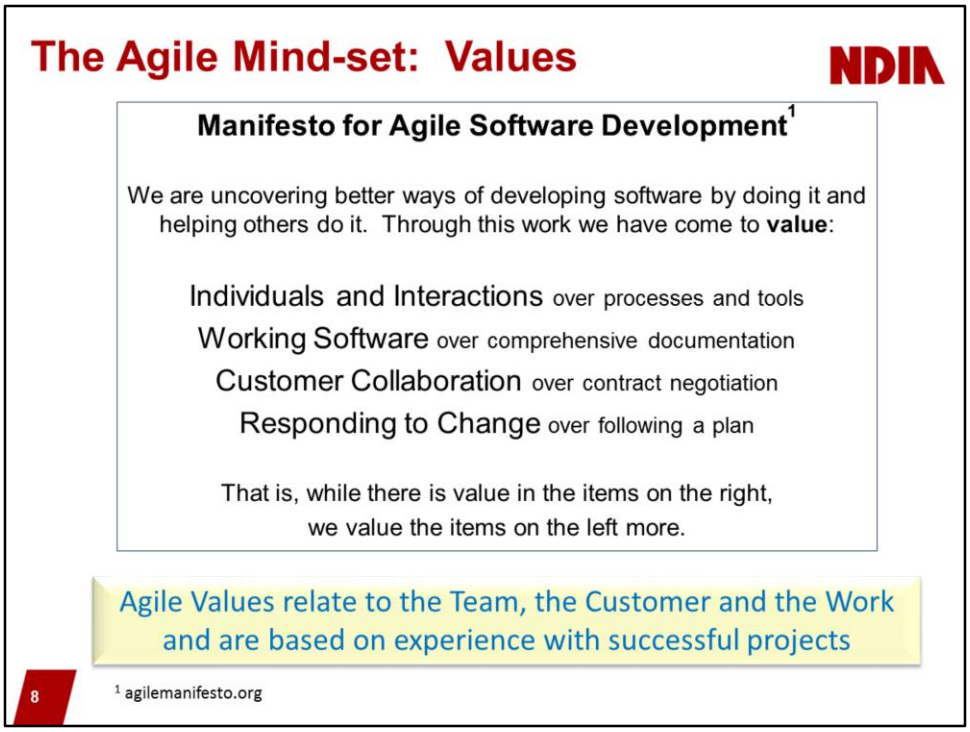
2001 - "Agile" coined at meeting of 17 Leading Practitioners in Utah (Agile Manifesto created)

These were software leaders comparing notes on what works – not pushing theories, products or services

Another point: Some consider "Agile" or "Lightweight" processes to be a return to the earlier days of the discipline before the processes began to bloat in an ineffective effort to pursue higher productivity

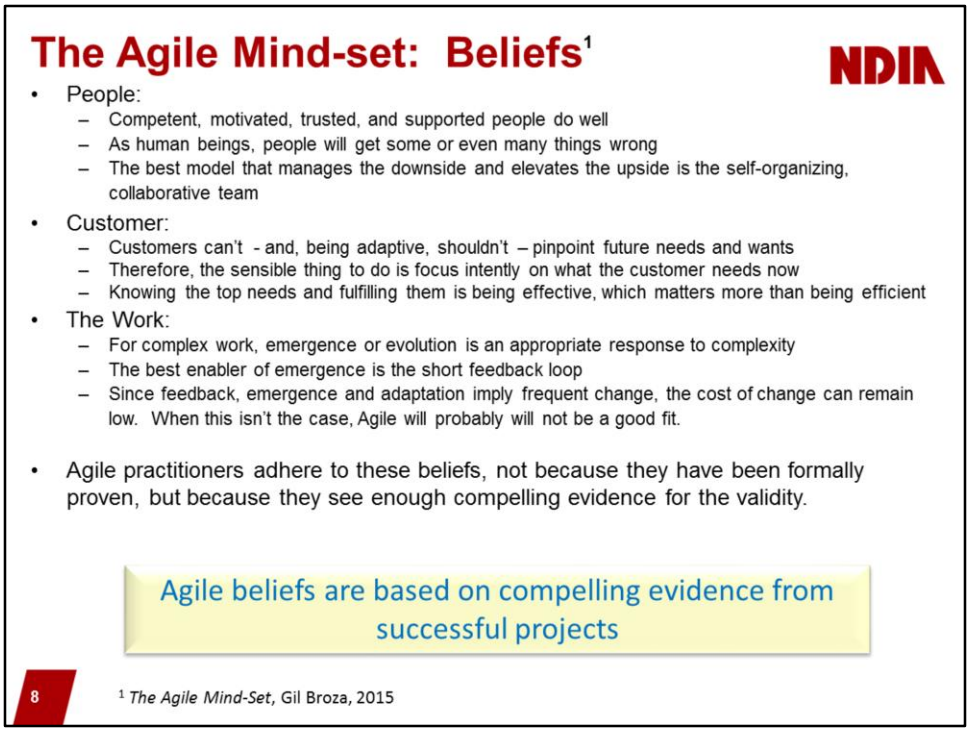



This slide lays out the basic elements of a Mind-set.
Emphasize the source of this information: Book by Gil Broza called the Agile Mind-set released in 2015.
Emphasize that it is important to understand your Mind-set because that drives the actions you take in any situation.



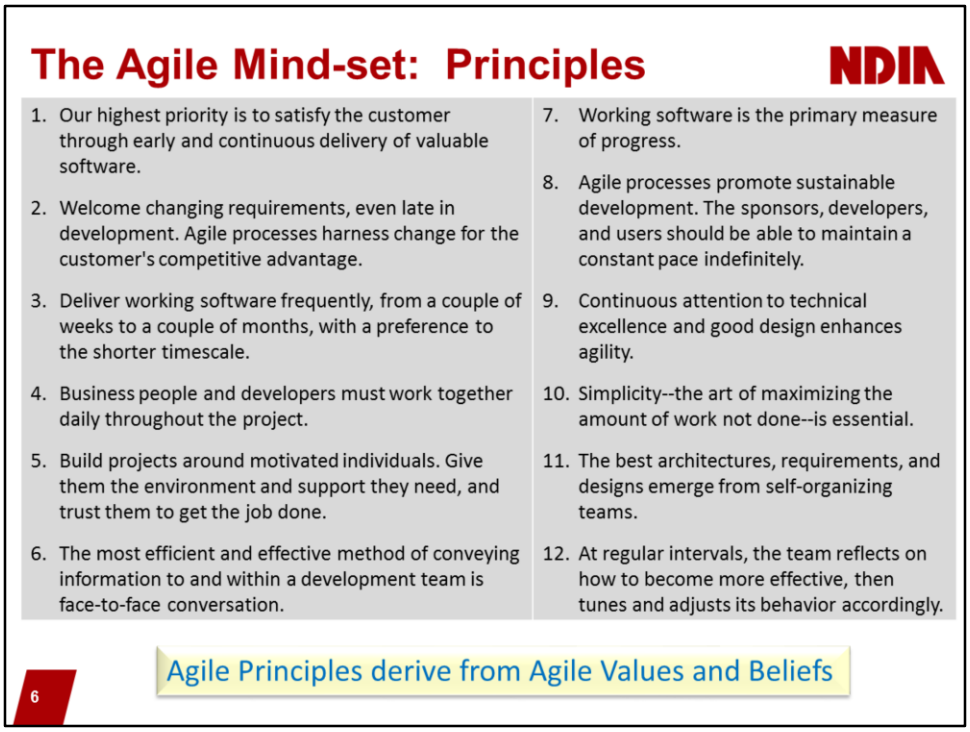

The Agile Mind-set, like other Mind-sets has Values, Beliefs and Principles
This slide reviews the Values using the Agile Manifesto published in 2000 by notable Agilists including: Jeff Sutherland, Jeff Beck, Ken Schwaber
Further elaborate the Values:

1. Individuals and Interactions – People come first before product and process
2. Working Software - Early and Frequent Value Delivery
3. Customer Collaboration
4. Responding to Change – Adaptation



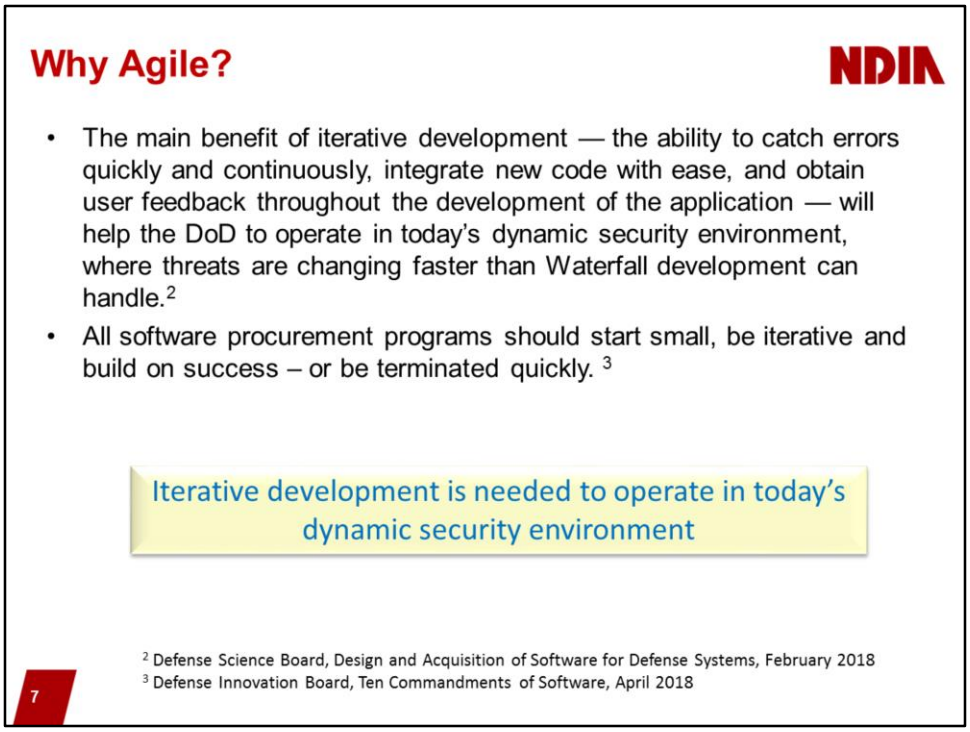

People:
First bullet corresponds to Theory Y of managing people vs Theory X developed by Douglas McGregor, *The Human Side of Enterprise*
Theory X: Team members dislike their work and have very little motivation; results in **authoritarian**, **micro-managing** style of management
Theory Y: People take pride in their work and see it as a challenge; results in **participative** management style based on **trust** that people will take ownership of their work and do it effectively by themselves.

Customer:
First bullet: pinpoint means **in great detail**; customers shouldn't try to figure out in great detail their future needs.




Associated Values:

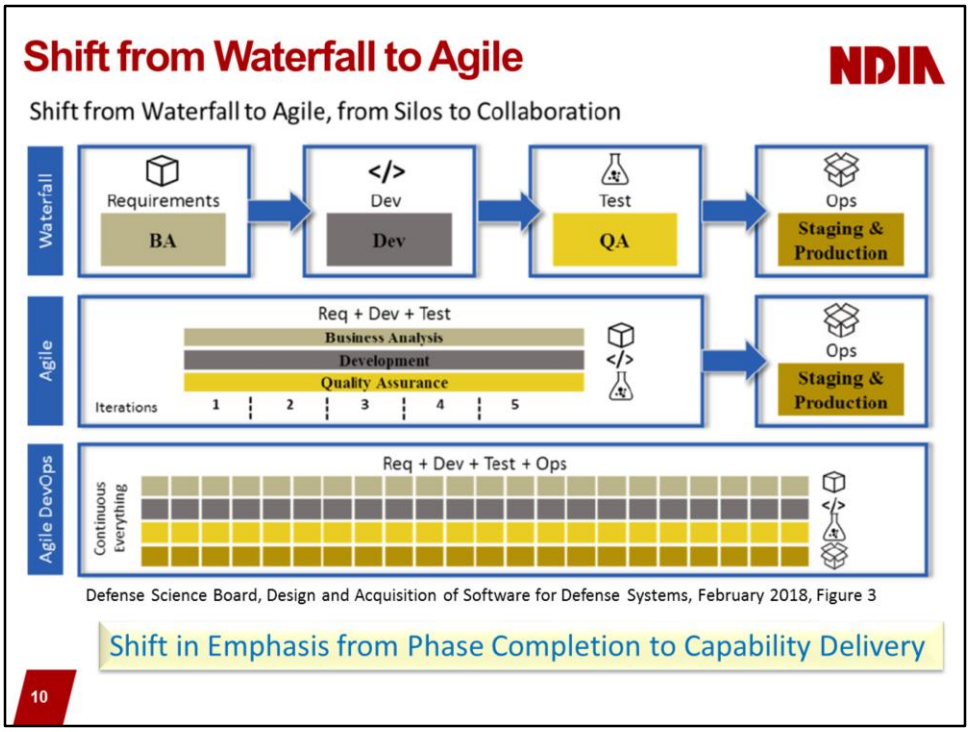
1. Working Software – Early and frequent value delivery
2. Responding to Change – Adaptation
3. Working Software – Early and frequent value delivery
4. Customer Collaboration
5. Individuals and Interactions – People come first before product and process
6. Customer Collaboration
7. Working Software – Early and frequent value delivery
8. Individuals and Interactions – People come first before product and process
9. Responding to Change – Adaptation
10. Working Software – Early and frequent value delivery
11. Individuals and Interactions – People come first before product and process
12. Individuals and Interactions – People come first before product and process




Agile is a **solution** to a problem

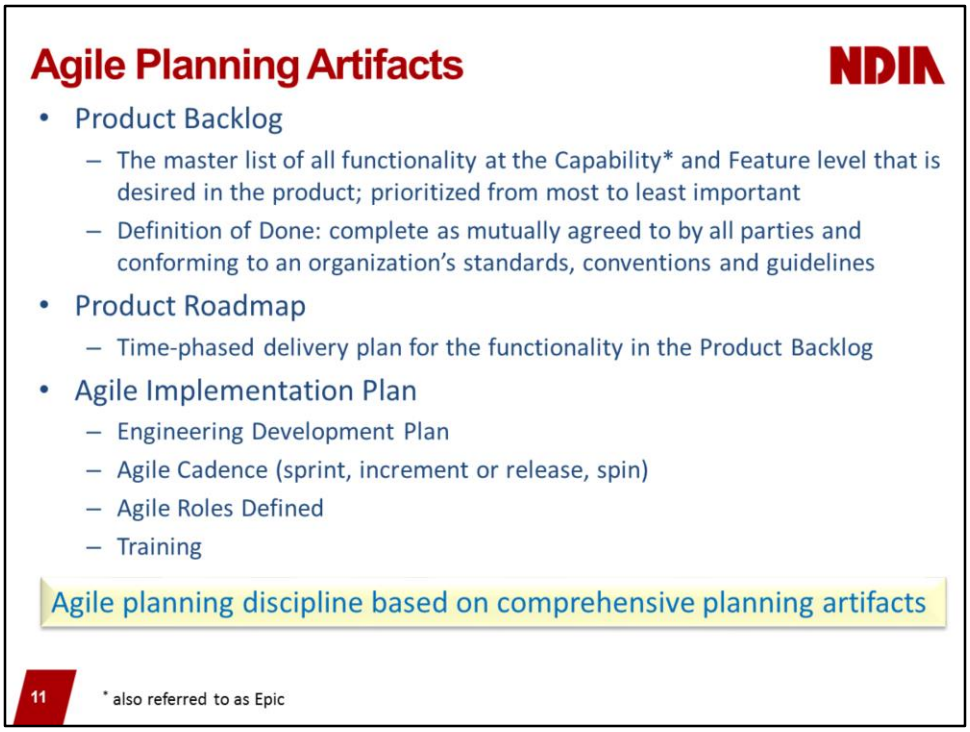
Agile Framework provides the processes and tools to enable rapid, incremental delivery of essential capabilities

The point is to use the iterative process to gain insight and awareness that if you are going to fail, it is better to do so more quickly before investing years of time and money.

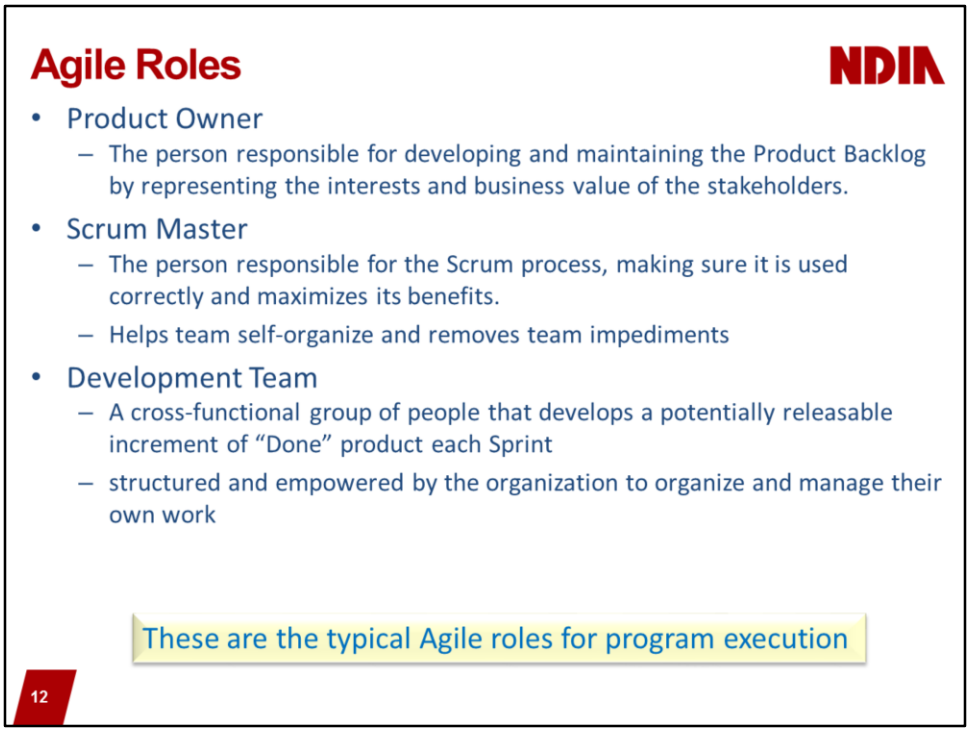



Waterfall is a sequential process which emphasizes completion of development phases and resists change, especially in later phases

Agile is an iterative process which emphasizes delivery of capability to the end-user and welcomes change based on user feedback to make sure the right system is delivered.

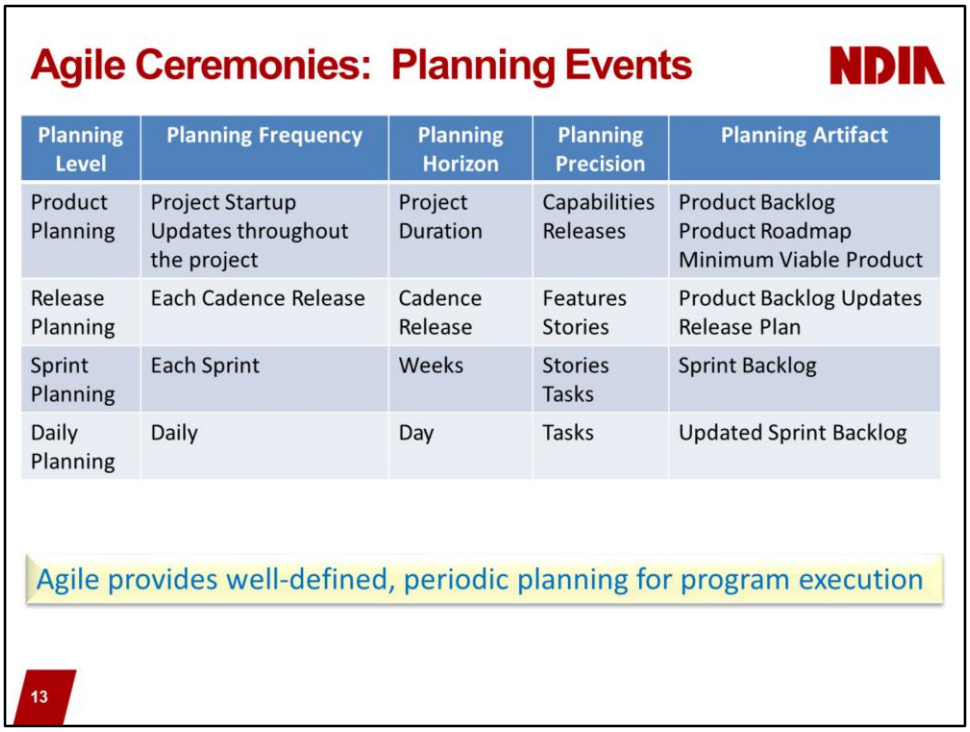



What do you need to be successful in planning a program and establishing a schedule with the PM?




Note: The roles described are from Scrum, a leading Agile development methodology

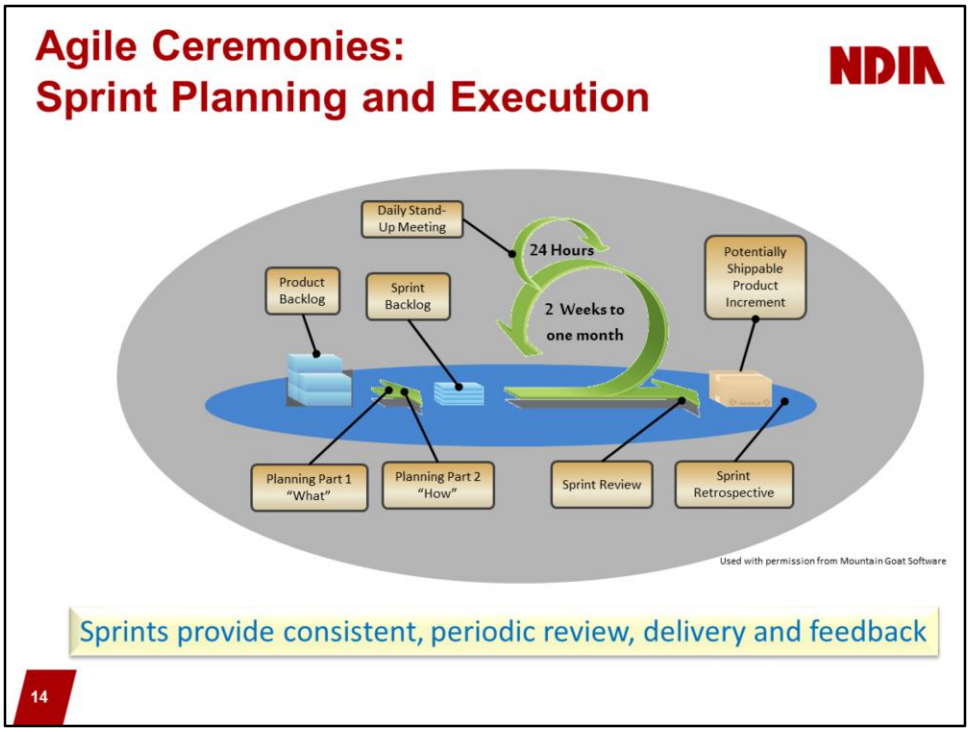
These roles are critical to understand how their responsibilities work together to deliver capabilities and validate they meet the business objectives of the contract.



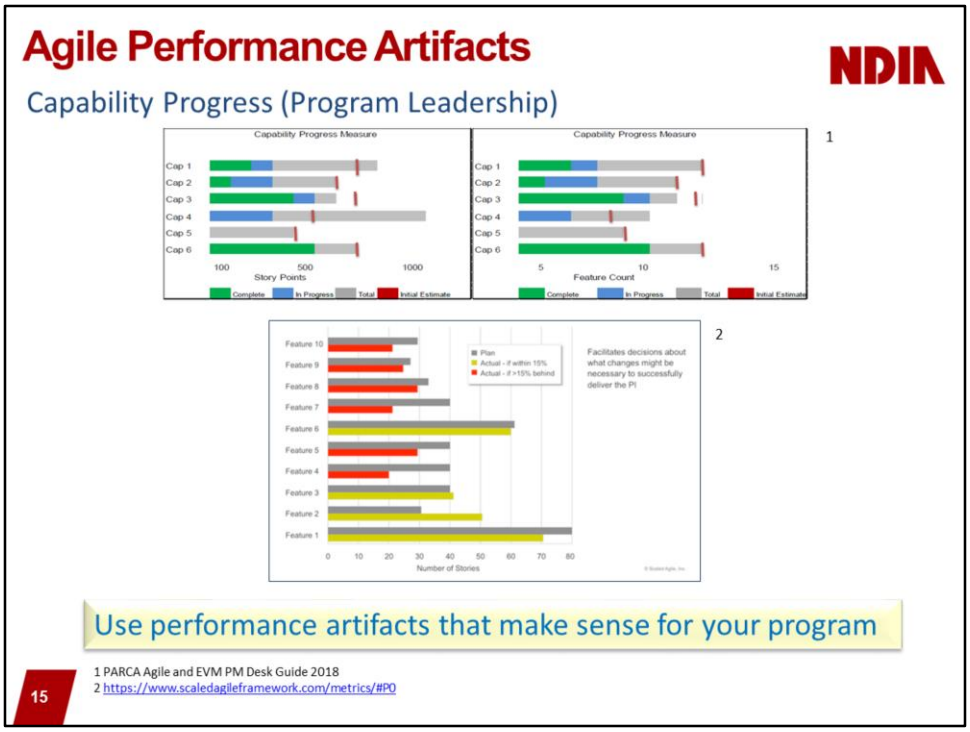

**Teaching Notes:**
Convey the iterative, recursive nature of Agile planning. There are multiple levels of planning that refine the understanding/definition of work to be done over the life of the project.
Focus is on the target (meeting business objectives), not the plan. Changing plans to adapt to changing customer needs or other program circumstances is normal and expected.
**Product Planning:**
performed at the beginning of the program;
defines all contract scope at the Capability level in the product backlog;
Creates the product roadmap by time-phasing the product-backlog in accordance with contract milestones and deliverables
Provides the technical scope definition of the initial performance measurement baseline
Repeated as needed throughout the life of the program based on program progress and customer direction
**Cadence Release Planning:**
Defines features for the upcoming rolling wave and maps those features to specific work packages to establish the updated earned value baseline
Features are decomposed from their parent Capabilities
May also be referred to as Increment Planning
There are two types of Releases:
1) Cadence Release – which is a time-based release and occurs on a regular schedule, typically quarterly, and is released either internally for baseline management or externally to the client/production environment. Cadence Releases most closely align to EVM Rolling Wave Planning and may result Schedule Variance if planned scope in the release are delayed to a future release.
2) Capability Release – is a scope based release and is not held to a regular delivery schedule – the release will be issued when it is ready, and therefore will not likely show a schedule variance, but would likely show a cost variance if it is late.

**Sprint and Daily Planning**
From an EVM perspective, Provides updated outlooks for in-progress features and work packages.
Sprint Planning includes breaking Features down into User Stories, assigning Story Points to the Stories, and Prioritizing the Stories in the Sprint Backlog

**Key Points:**
•Agile planning starts at the beginning of the project with a definition of the scope of the entire project at a high level.
•The scope definition is refined throughout the life of the project through a series of regular Release Planning events.
•The Feature level scope definition coming out of Release Planning is aligned to/reflected in the IMS as part of Rolling Wave planning.
•Release Planning is also referred to as Increment Planning in the Scaled Agile Framework (SAFe).




Slide had animation
Planning Part 1 – directed by PO
Planning Part 2 – directed by Team
Goal every sprint (iteration) is working product that is potentially shippable.
Potentially because downstream customer may not be prepared to receive it but the team has made sure the product is ready to go.
Retrospective provides feedback on the team's processes to encourage and enable continuous improvement




There are different artifacts and progress reports available from all agile tools. Some are useful for program leadership and management. Some are useful for team management.

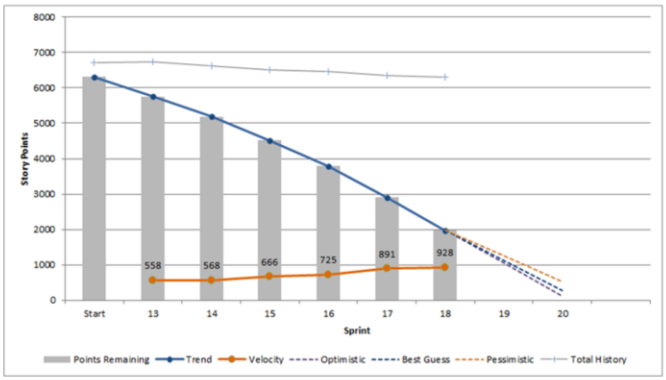
A report showing the completion of the items on a road map can be used to confirm if capabilities are being completed relative to the initial estimate.

Take note of the source of each graph. The links provided are reputable agile sources. They have many more types of reports for you to get familiar with for application to your program.

## Agile Performance Artifacts (cont.)

NDIA

- 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

16

This chart is from the NDIA Agile Guide.

The Start Bar shows the number of Story Points at the Start of the release and then shows the remaining story points at the end of each successive sprint (13-18) with projections to sprint 20.

This chart provides an estimate, of when the team is expected to complete the work remaining in the release backlog

The gray line across the top is the total cumulative story points in the backlog and is showing the change in total story points over time, adjusting for additions and deletions.

The gray vertical bars indicate the number of incomplete story points at the beginning of each sprint, with the Start bar always equaling the current total story points in the release backlog and the remaining bars equaling the height of the previous bar minus the sprint velocity.

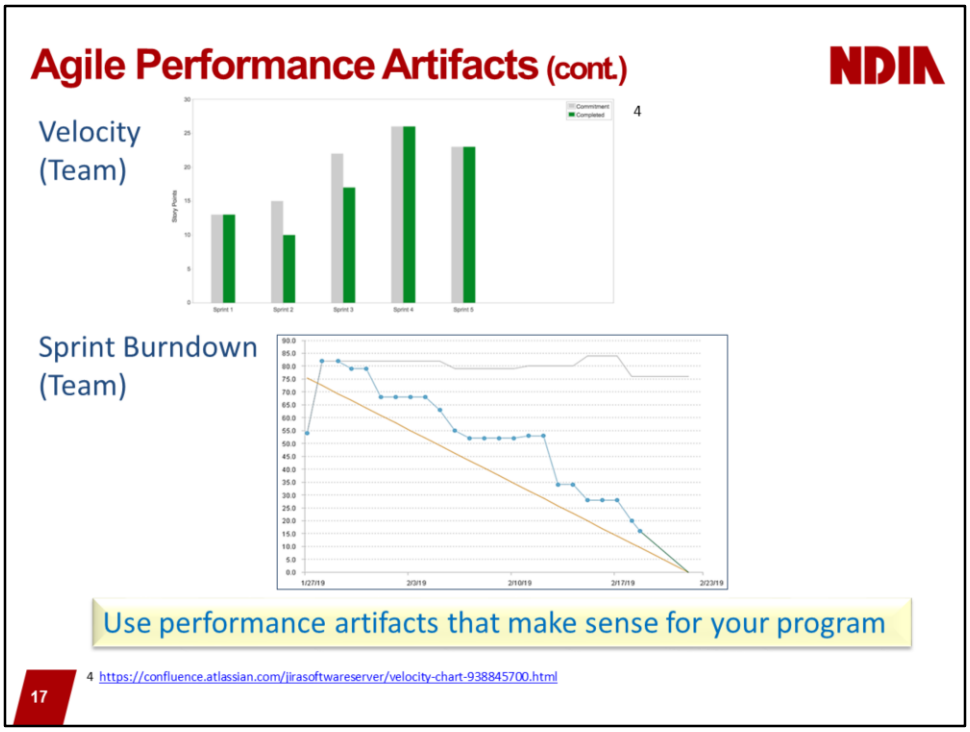
The solid orange line is Velocity (points completed each sprint) and represents the capacity of the team to complete work (as recent past performance could indicate future performance (similar to CPI or SPI).

The solid blue line shows the current completion trend.

To determine "when we are expected to complete the remaining backlog, the blue line is extrapolated to the point where it crosses the x-axis (remaining points = 0). In the example shown, there were

- Optimistic (purple dotted line) which looks like it assumes future work to complete at the same rate as presently executing (best of the last n iterations) and should complete in Sprint 20
- Most Likely (blue dotted line), average velocity of the last n iterations, still has a chance to complete in Sprint 20
- Pessimistic (orange dotted line), worst of the last n iterations, may actually push overall completion to a 21st sprint

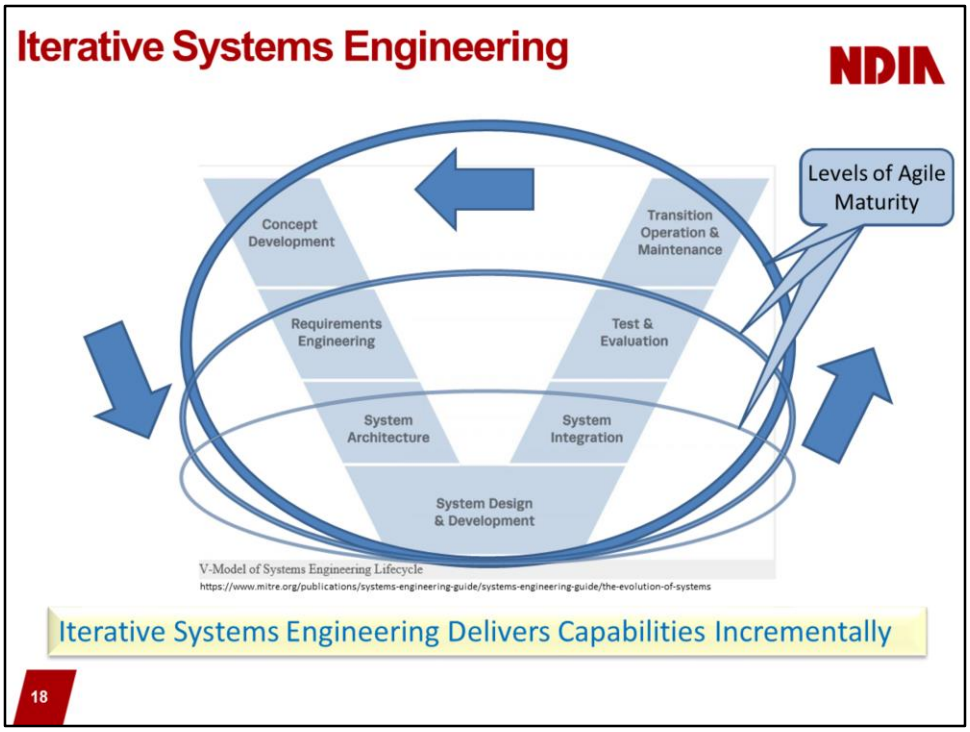
Work that is not completed as planned within a cadence release is re-prioritized in the backlog and moved to the next release




Velocity – story points completed each sprint; team capacity; used by team to help determine how much to plan in upcoming sprints;
Also used by team to see the impact of their retrospectives in identifying and resolving impediments that affect team velocity.

Sprint Burndown Chart
"The Sprint Burndown Chart makes the work of the Team visible. It is a graphic representation that shows the rate at which work is completed and how much work remains to be done. The chart slopes downward over Sprint duration and across Story Points completed. What makes the chart an effective reporting tool is that it shows Team progress towards the Sprint Goal, not in terms of *time* spent but in terms of *how much work remains*."
https://www.scruminc.com/sprint-burndown-chart/

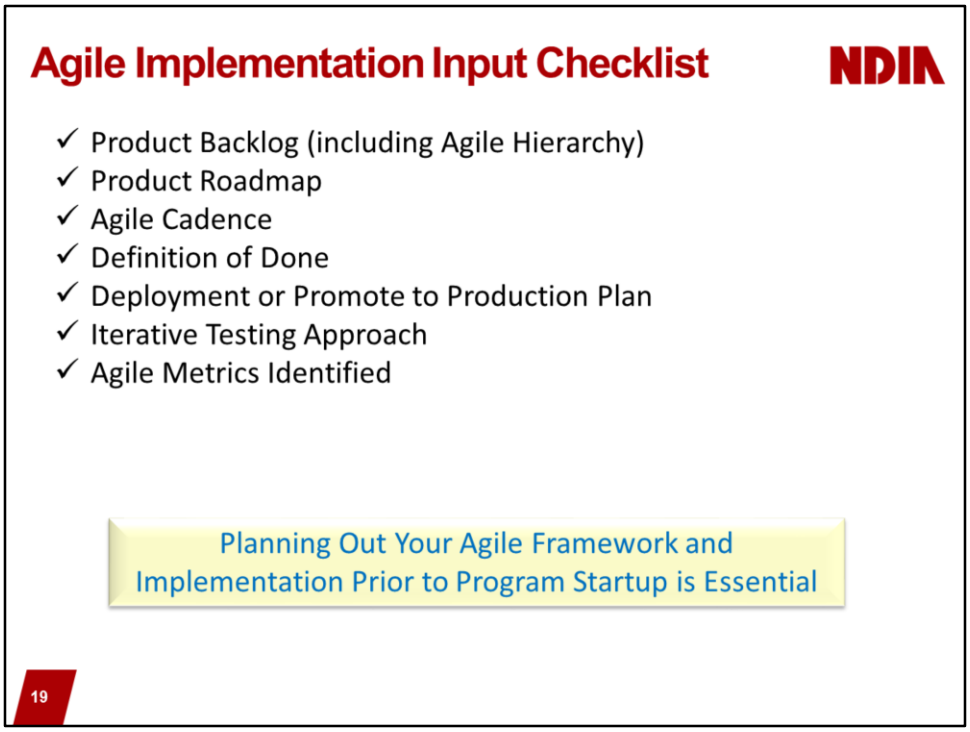
The grey line across the top is the total story point in the sprint backlog on each day of the sprint
The blue line is the total points remaining undone in the sprint backlog on each day of the sprint
The slope of the line after the last blue dot shows the rate at which work needs to be complete to finish the sprint backlog on time
The yellow line is the "ideal" burndown – the constant rate required to complete sprint backlog on time




Three rings:
Small: lowest level of agile maturity – iterations include only low-level requirements, development and integration in the developing organizations environment
Medium: higher level of agile maturity – iterations include test and evaluation on target hardware but not the actual system
High: highest level of maturity – iterations include testing on actual target system

Maximize the Agile maturity level, collaborating with and making releases to the end-customer as frequently as possible to obtain user feedback and ensure building the right system.

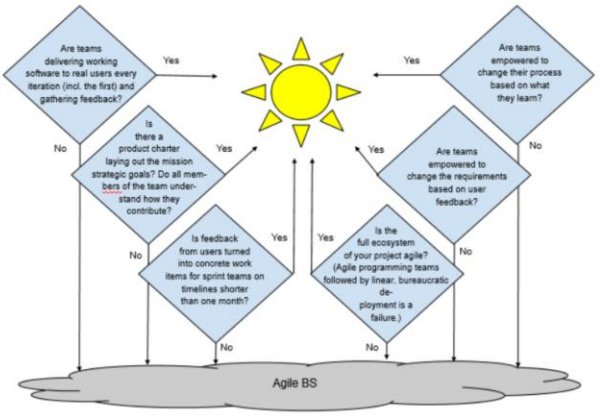
**Acquisition Authority** is a major factor in the level of Agile maturity possible (e.g. Test and Selloff requirements are typically large-batch when the system is "complete"




Print out this slide and use it in program set up to ensure you are at a good point to begin your schedule.

## Conclusion

NDIA




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

20

This is the DIBs version of knowing if you are really using agile and what does agile really mean.