

# AF Cyber Mission Platform (CMP)

## *Analyzing the Full Suite of Agile Metrics*



**Alex Smith, CTR**

(AFLCMC/HNC, Quantech Services)

**Brian Opaska, CIV**

(AFLCMC/HNC)

**William Lane, CTR**

(AFLCMC/HN, Quantech Services)



U.S. AIR FORCE

“Virtually every Air Force system is software controlled. To gain and maintain a competitive edge, we must shift towards agile software development and close integration with operators and developers.”

“Programs making the transition from waterfall to agile should have proven processes, metrics, contracting approaches, and user feedback loops rather than having to go invent them.”



**Dr. William Roper**  
**Assistant Secretary**  
**USAF Acquisition, Technology,**  
**and Logistics**



- **CMP: Program Overview and Agile Transformation**
  - Overview, History
  - Metrics and Tools (JIRA)
  - Team Sizing
- **OUSD Acquisition & Sustainment Office: Agile Metrics Guidance**
  - Recommendation of OUSD (A&S) Agile Metrics Guidance
  - Exploration of CMP tracked metrics with comparison to OUSD (A&S) Guidance
    - Agile Process Metrics
    - Agile Quality Metrics
    - Agile Product Metrics
    - Agile DevSecOps Metrics
- **Recommended Minimum Set of Metrics**
- **Way Ahead for CMP**

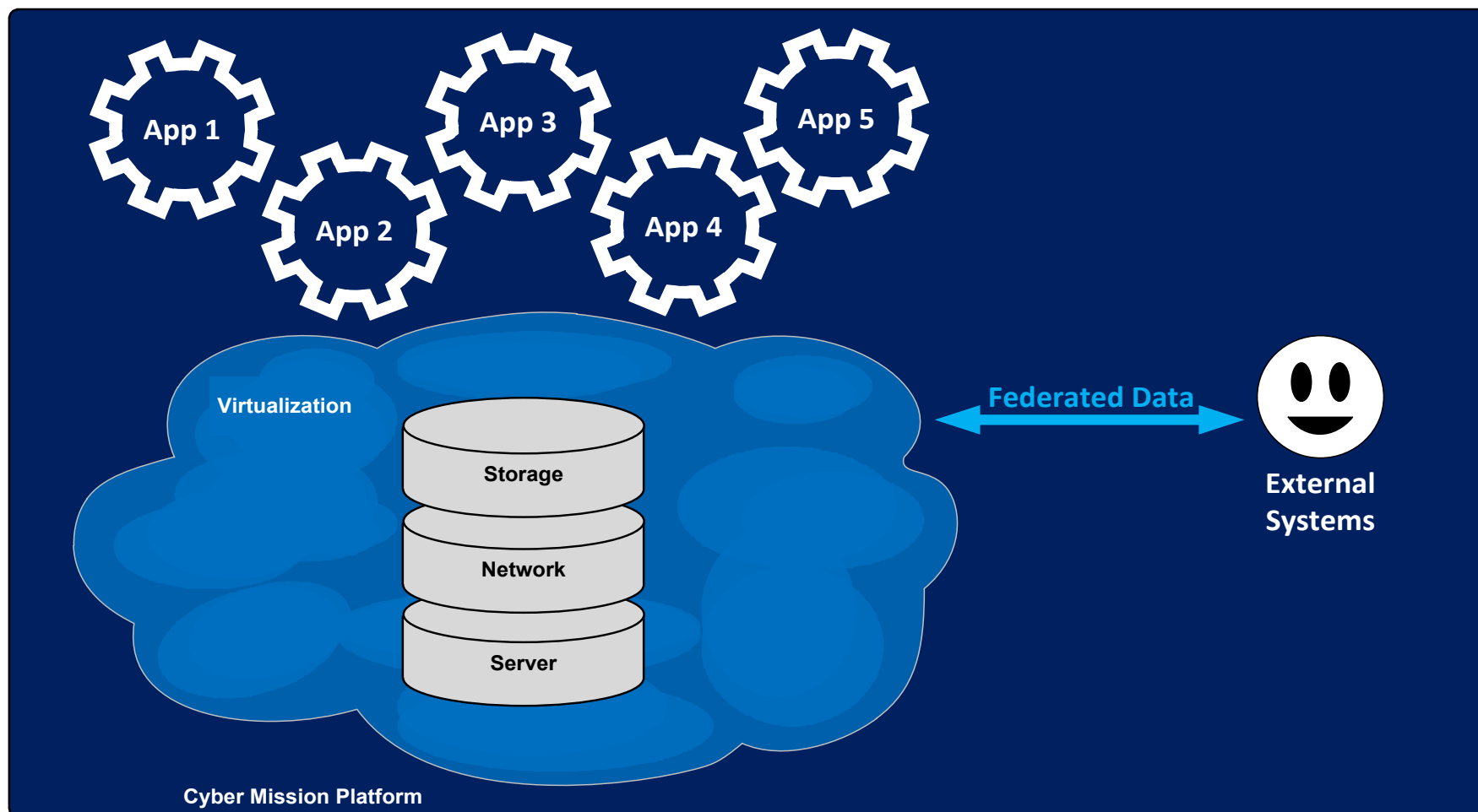




U.S. AIR FORCE

---

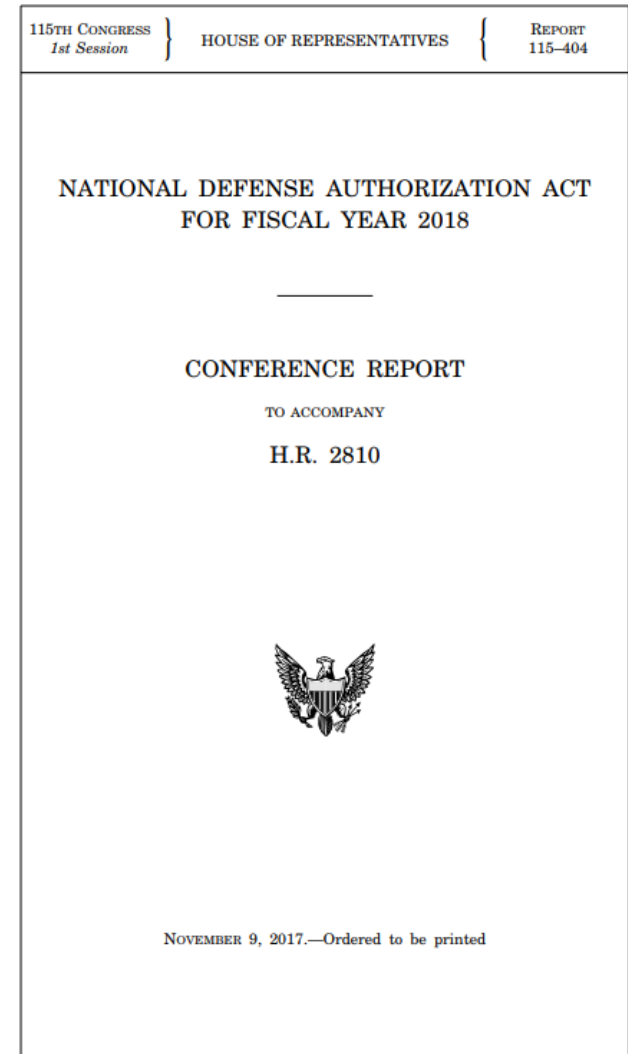
# **AF Cyber Mission Platform:** *Program Overview and Agile Transformation*



- **CMP is a common framework for Cyber Mission Forces**
- **Mission and support applications increase automation for complex missions**

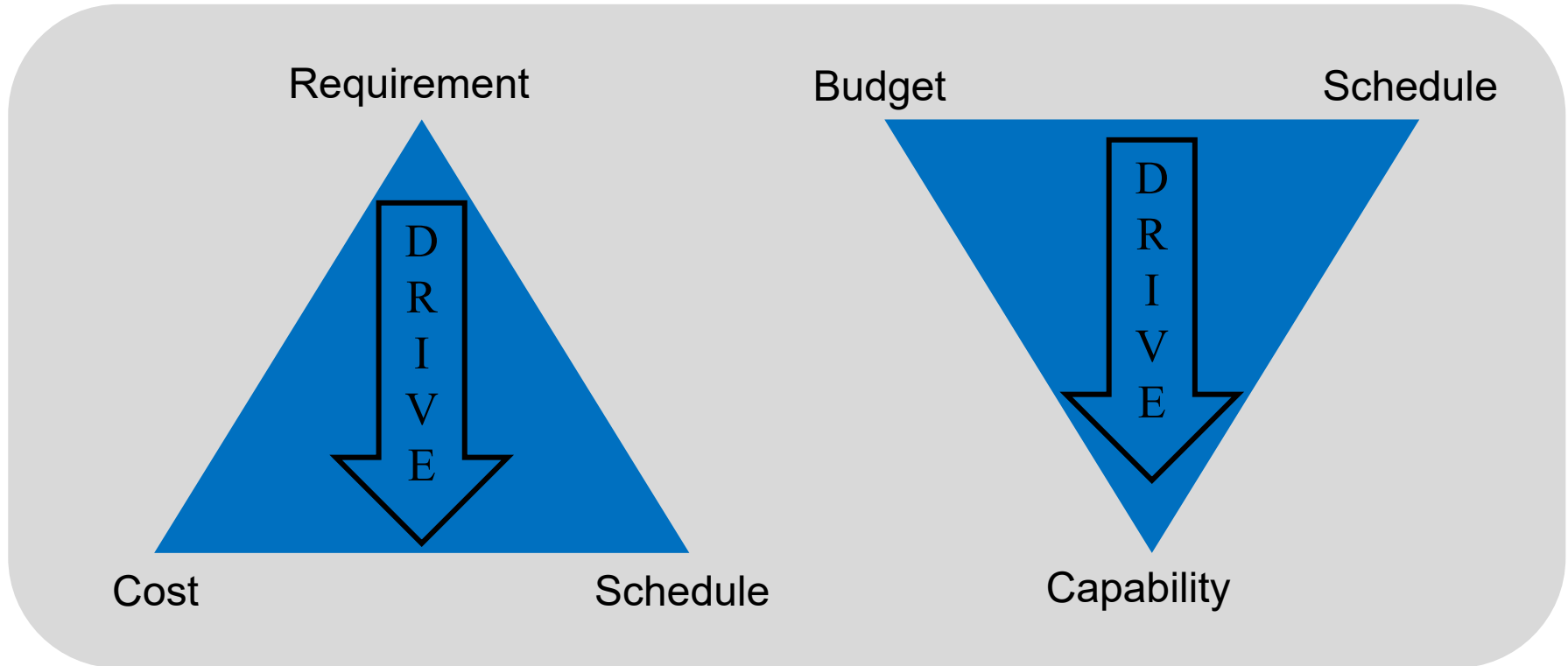


- **Originally established as ACAT III in FY14 with Statement of Requirements re-validation in FY17**
- **Completed Sec 874 Pilot Program as identified by FY18 National Defense Authorization Act (NDAA) Nov 2017**
- **Adopted SAFe Agile for Program Planning beginning Jan 2019**
  - MAJCOM transferred from AFSPC to ACC
  - Refine processes and interactions with partners and stakeholders
  - Integrate with Multi-Year Program Execution Plan (MYPEP) process and roles





# CMP Program Strategy Overview

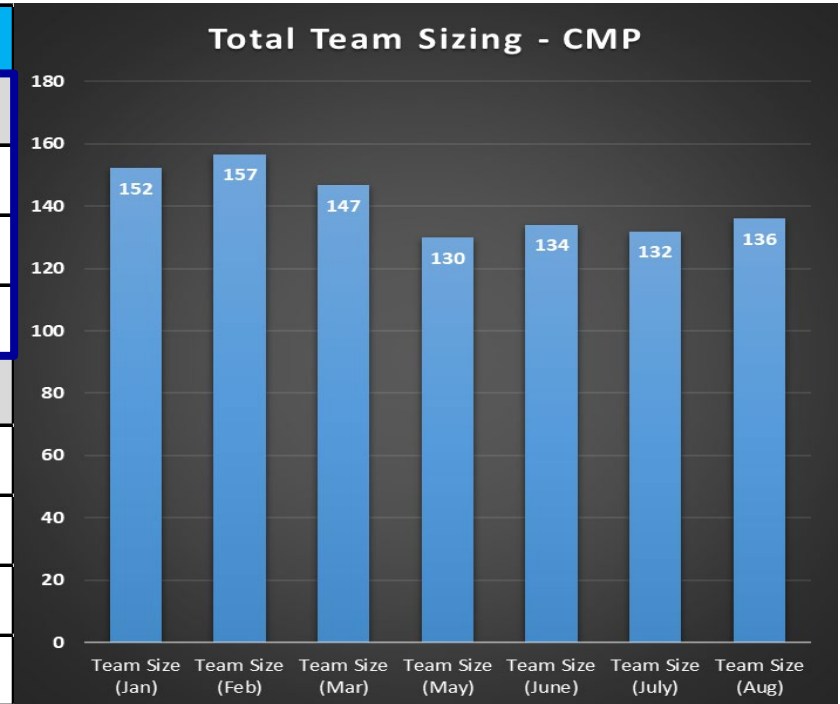


- Capacity-based Life Cycle Cost Estimates built without discrete scope in alignment with Sec. 874 direction (No EVMS, No SLOC)
- Contract Type is Time x Materials
- Program utilizes DI2E, Confluence, and JIRA for program planning and progress tracking

# CMP Program Structure and Management Tools



Team Pages		
Development Teams		
App 1	App 2	App 3
App 4	App 5	App 6
App 7	App 8	
Systems and Shared Services Teams		
Test	Infrastructure	Systems Team
Shared Services	Cybersecurity	LeanUX
Integrated Systems Support	Service Desk	PMO
Growth Board	Architecture	



- **CMP utilizes Confluence and JIRA**
- **Confluence – used for Program Structure tracking (Team sizing, strategy, artifact documentation, etc.)**
- **JIRA – used for Software Effort progress tracking**





***This should be easy, right?***

10 Teams x 8 Heads Per Team x Duration x Cost Per Head

***My work here is done?***



# OUSD Acquisition & Sustainment: Agile Metrics Guidance (2019)

“The increasing complexity of systems, decentralization of tools and technologies, and rapid pace of change pose **challenges in delivering** meaningful, secure, and modern **capabilities** that meet user expectations on time and within budget”

“Addressing these issues requires a **significant change** in approach to planning and delivery of capability to the warfighter: specifically, a **shift to Agile** project management and delivery”

“**Transitioning** from a waterfall to an Agile approach **represents a true paradigm shift** - impacting at all levels of the organization, including programs, projects, and enabling technologies”



# **OUSD Acquisition & Sustainment: Agile Metrics Guidance (2019)**

**Draft guide released Jun 2019**

**Post-draft collaboration between OUSD (A&S) and Cost Community; Software Special Assistant Dr. Jeff Boleng through AFCAA SRDR WG, SW IT CAST, PSM**

**Consistent with metric recommendations from Government Accountability Office (GAO), Defense Innovation Board (DIB), Defense Science Board (DSB)**

**Covers full spectrum of metrics – Process, Quality, Product, DevSecOps, Cost, Value**

**Version 1.0 (Sep 2019) expanded to include metrics more meaningful to cost community released. Honest Discussion of strengths and weaknesses of each metric.**

# OUSD Acquisition & Sustainment: Agile Metrics Guidance (2019)

## Process Metrics

- Story Points
- Velocity
- Velocity Variance
- Velocity Predictability
- Story Completion Rate
- Sprint Burndown
- Release Burn-Up
- Cumulative Flow Diagram

## Quality Metrics

- Recidivism
- First Time Pass Rate
- Defect Count
- Test Coverage
- Number of Blockers

## Product Metrics

- Delivered Features (or Capabilities)
- Delivered Value Points
- Level of User Satisfaction

## DevSecOps Metrics

- Mean Time to Restore (MTTR)
- Deployment Frequency
- Lead Time
- Change Fail Rate



# Agile Process Metrics

## Velocity Predictability

“Insight into **predictability** and provide data points for conversations on root causes of the fluctuations”

“Users want to know **how long** it will take to see their **requested change**; predictability will help the team provide better estimates”



U.S. AIR FORCE

# Cyber Mission Platform: Agile Process Metrics

## Velocity Predictability

Requirement



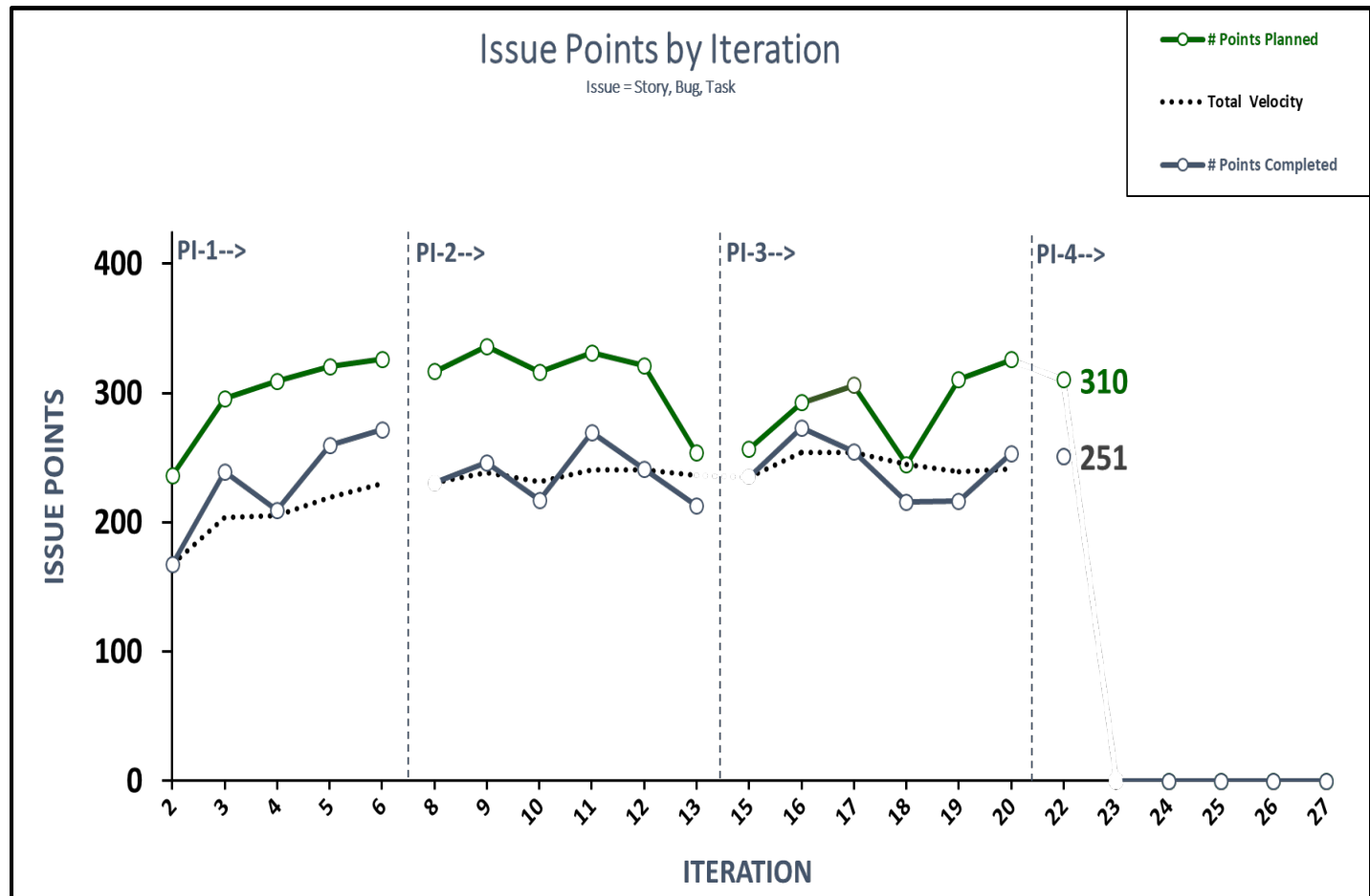
Feature



Issue (Story)



Issue Point(s)



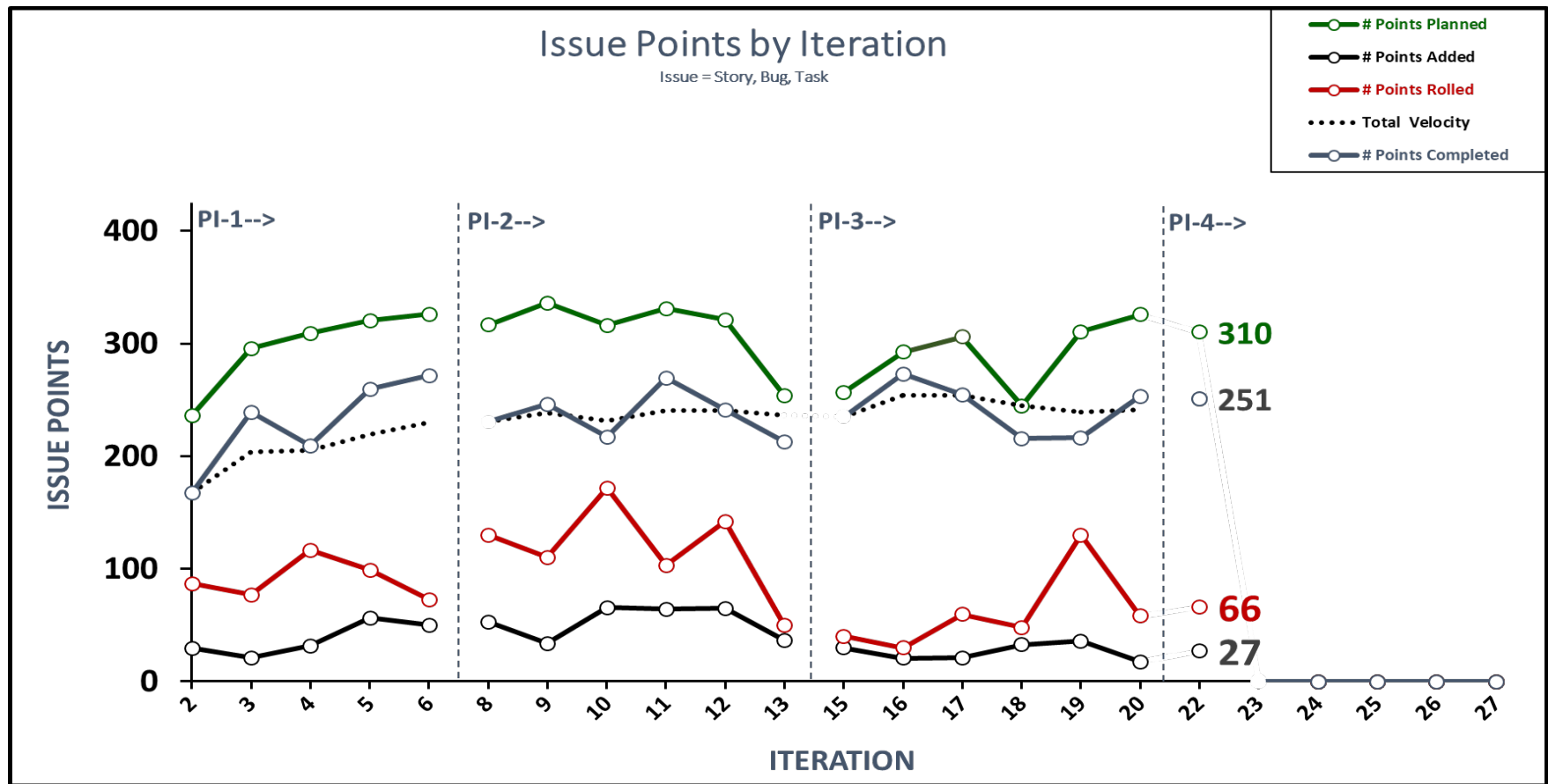
- How well are we planning and executing?



U.S. AIR FORCE

# Cyber Mission Platform: Agile Process Metrics

## Velocity Predictability



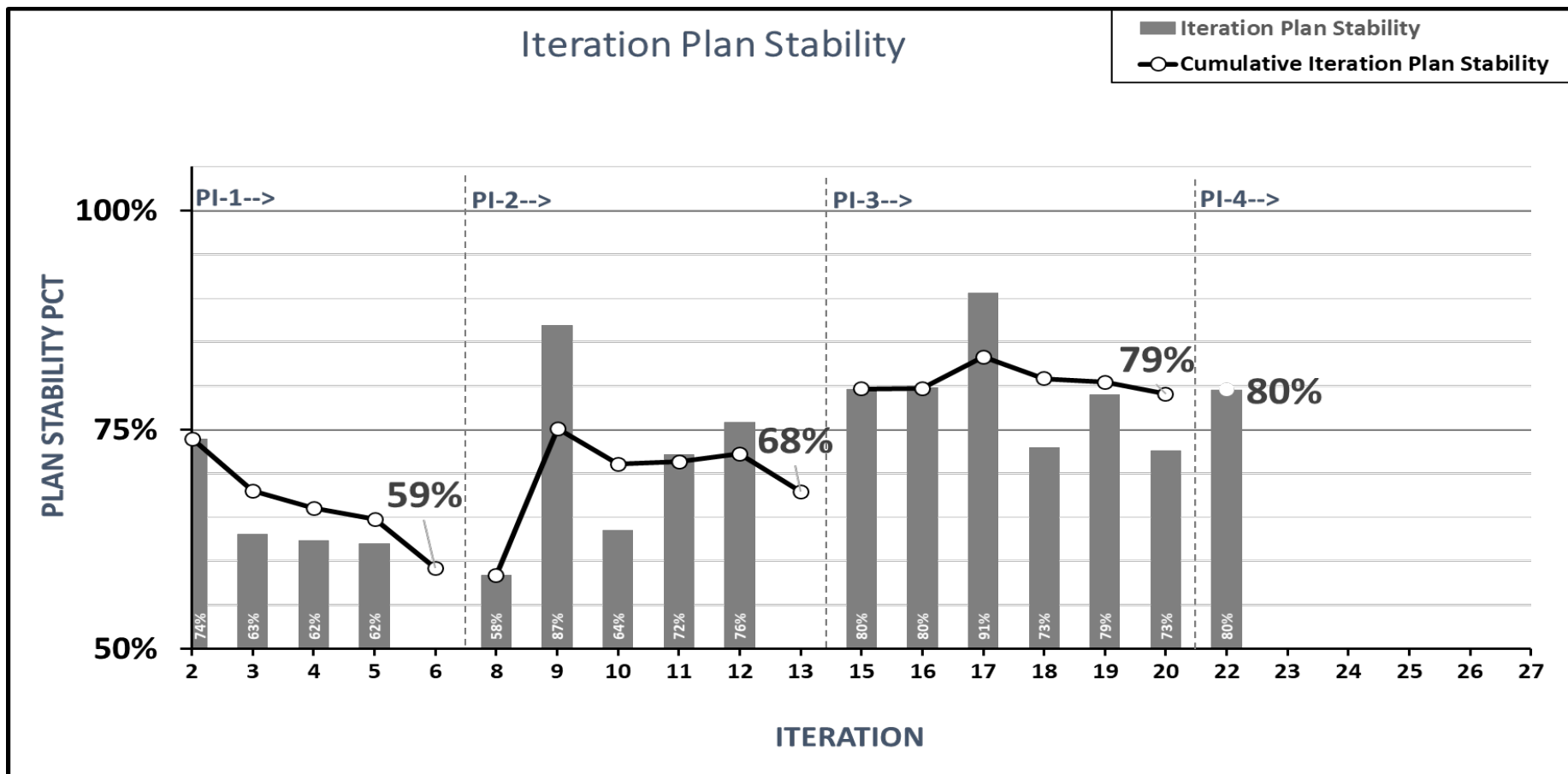
- Consistently over-committing, rolling a significant amount of scope (story points) to the next iteration/sprint, and adding additional scope (story points) within the iteration/sprint
- Programs often take many PIs for velocity to stabilize



U.S. AIR FORCE

# Cyber Mission Platform: Agile Process Metrics

## Iteration Plan Stability



- How stable is our plan? Is stability improving as the transformation advances through several PIs?
- Iteration Plan Stability =  $1 - (\text{Nbr of Issues Added} + \text{Nbr of Issues Removed}) / \text{Nbr of Issues Committed}$



# Agile Process Metrics

## Story Completion Rate

---

“Number of stories completed in a given sprint or release”

“Good way of communicating **progress on user requirements** to the Product Owner and users as opposed to story points, which can be vague or difficult for users to map to actual progress against requirements.”



U.S. AIR FORCE

# Cyber Mission Platform: Agile Process Metrics

## Story Completion Rate



- Issue completion (Story, Bug, Task) is more meaningful to the user than Story Points



U.S. AIR FORCE

# Cyber Mission Platform: Agile Process Metrics

## Predictability Goals Dashboard

	Metric	Scale
<b>Agile Team Measures</b>		
Are Agile Teams meeting their Iteration Plan (measured by Issues and by Issue Points)?	Iteration Plan Completion	Objective: 100% Threshold: 90+%
How much work "leaks" to a future Iteration? Is there a bow-wave?	Iteration Leakage	Objective: 5-% Threshold: 15-%
How stable/volatile are the Iteration Plans?	Iteration Plan Stability	Objective: 90+-% Threshold: 80+-%

- Collaboration between SPO and OEM to measure and assess process metrics



# Agile Product Metrics

## Delivered Features

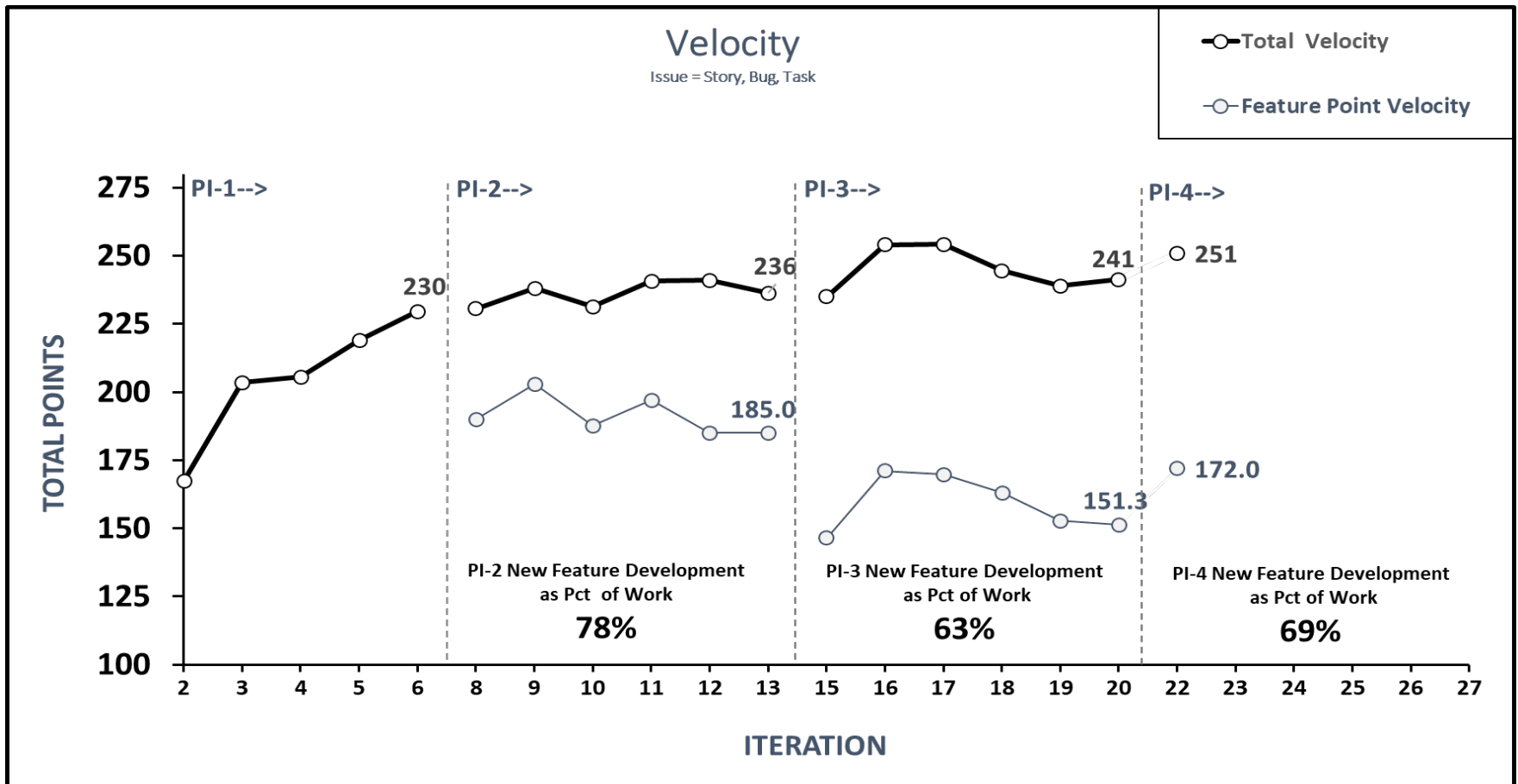
“Count of delivered features measures the business-defined **features accepted and delivered**”

“What is important is to find a functionally equivalent measure that provides insight into the **pace of value delivery** such as capabilities, user stories or requirements.”



# Cyber Mission Platform: Agile Product Metrics Delivered Features

U.S. AIR FORCE



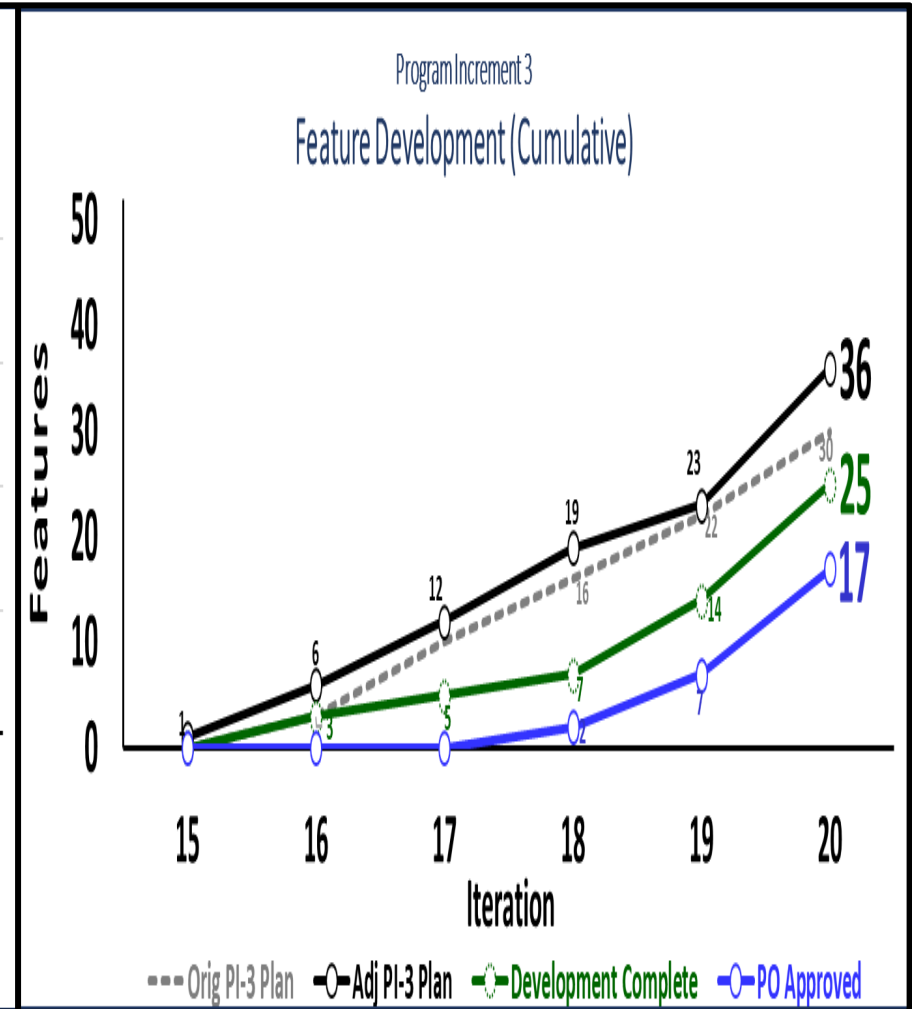
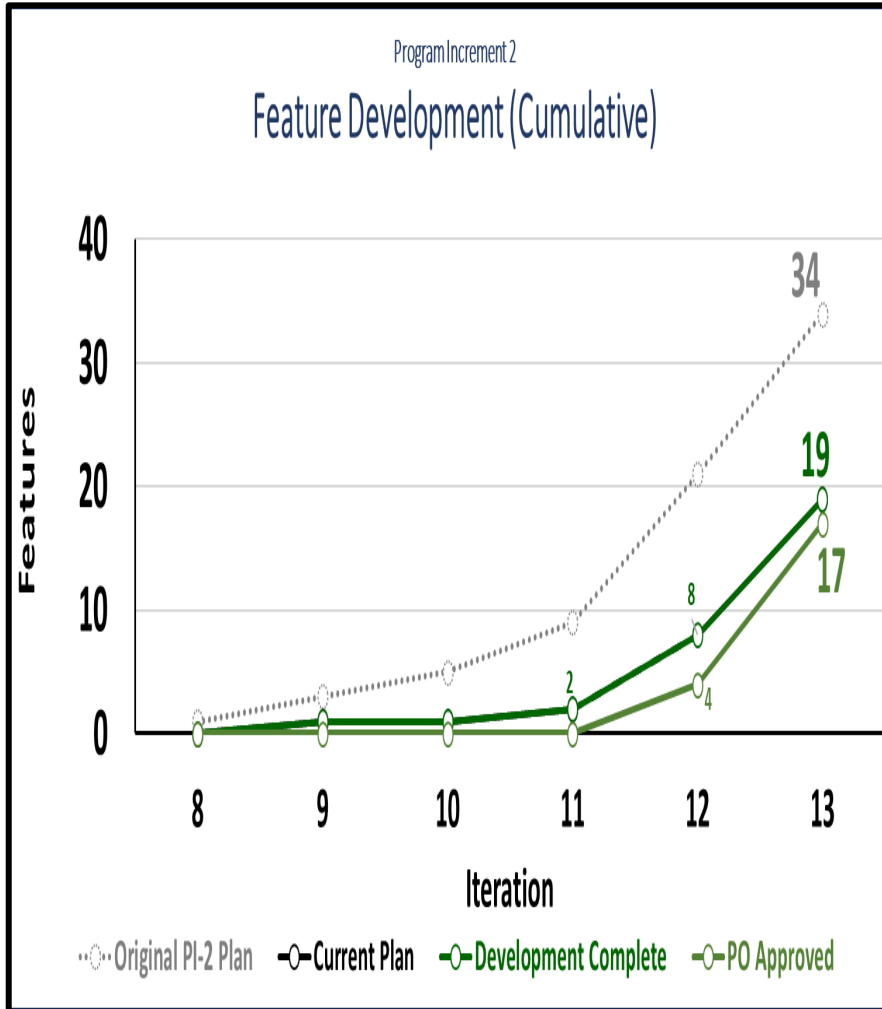
- **Some scope (story points) might not contribute directly to a Feature (or Capability)**  
Common Non-Feature work includes Enablers, Spikes, Improvements, etc.
- **Feature delivery projections for a capability/value roadmap should account for non-feature work over time**



# Cyber Mission Platform: Agile Product Metrics

## Delivered Features

U.S. AIR FORCE

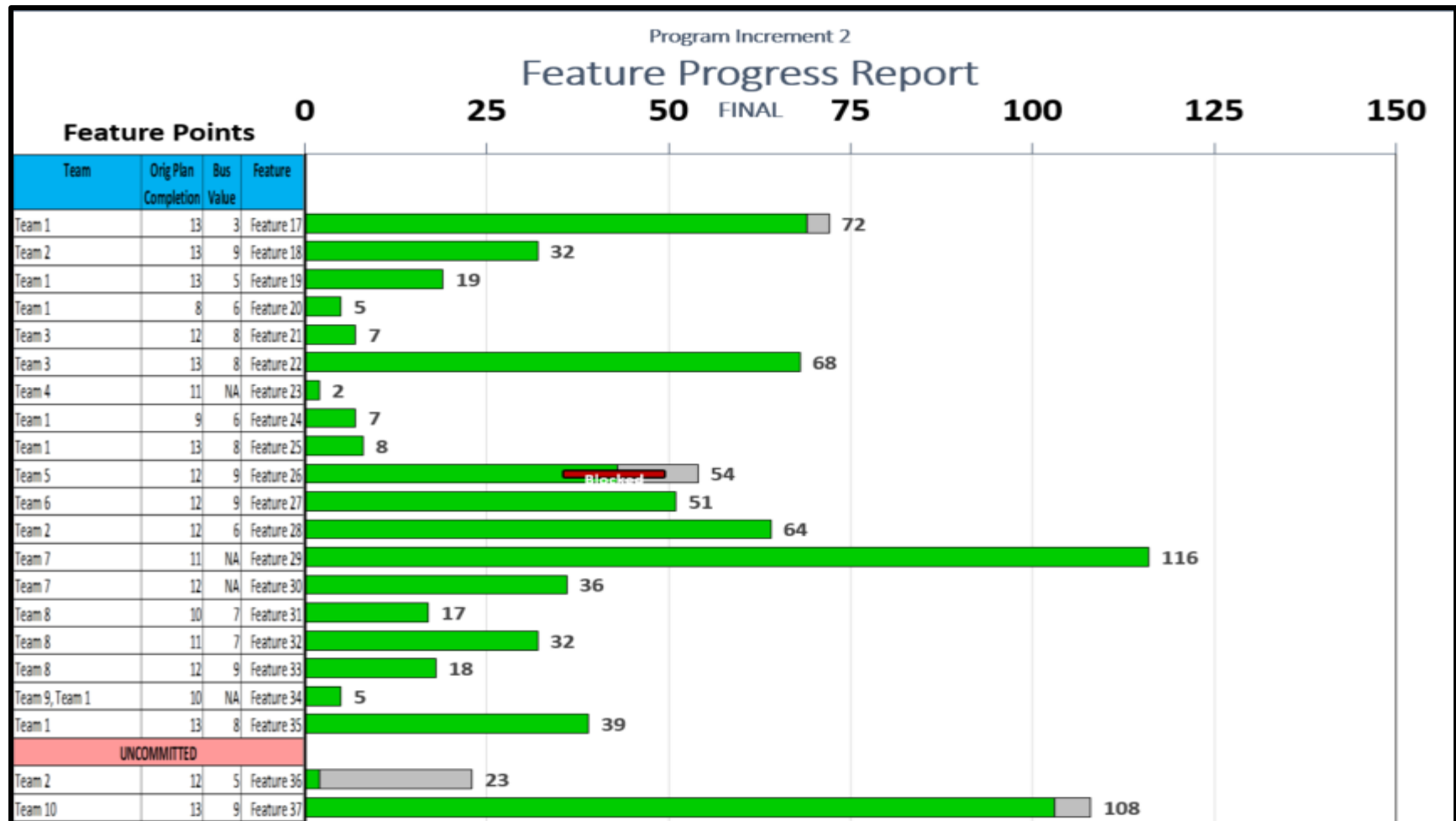


- Feature completion is more meaningful to Users and Leadership than Stories or Story Points



# Cyber Mission Platform: Agile Product Metrics Delivered Features

U.S. AIR FORCE



- PI 2 completed 19 Features with progress towards an additional 19 (17 Committed, 2 Uncommitted)



# Agile Quality Metrics

## Number of Blockers

“Number of events that **prohibit** the **completion** of an activity. The blockers cannot be resolved by the individual assigned to complete the activity and needs assistance to remove the blocker.”

“Understanding the number of blockers in a given sprint or release can inform the program management team and the organization of **potentially larger issues** such as issues related to governance or organizational structure”

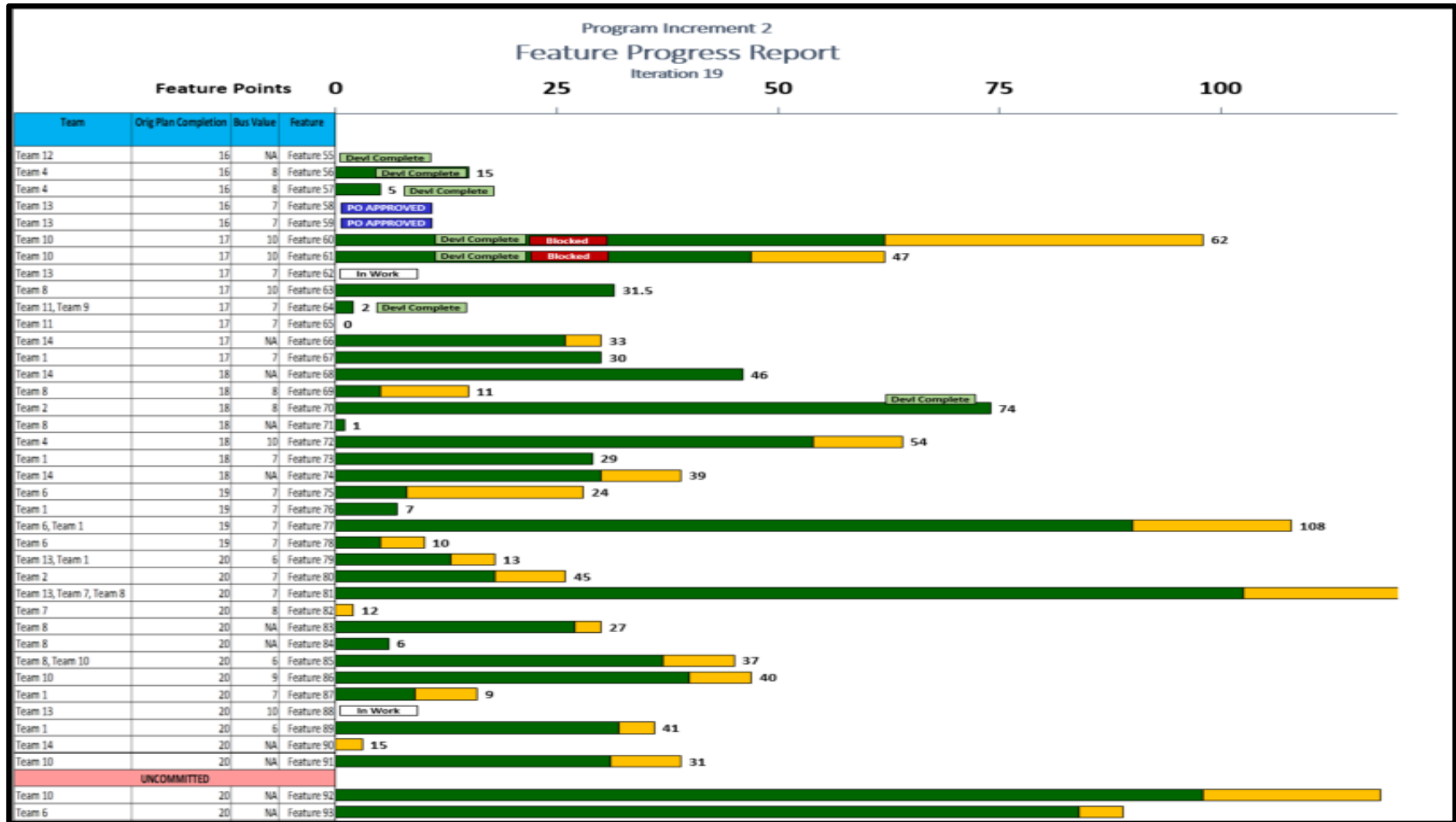




U.S. AIR FORCE

# Cyber Mission Platform: Agile Quality Metrics

## Number of Blockers



- PI 3 adds additional Features; Blockers prevent completion

## Agile Product Metrics

### Delivered Value Points (Business Value)

---

“The count of value points delivered to users for a given release. Value points are usually **defined by the users** to indicate an assigned business value to a given feature.”

“Capturing value points on an ongoing basis **requires** ongoing, **active involvement from the users** to assign value points prior to development activities, and reassess the value once delivered.”

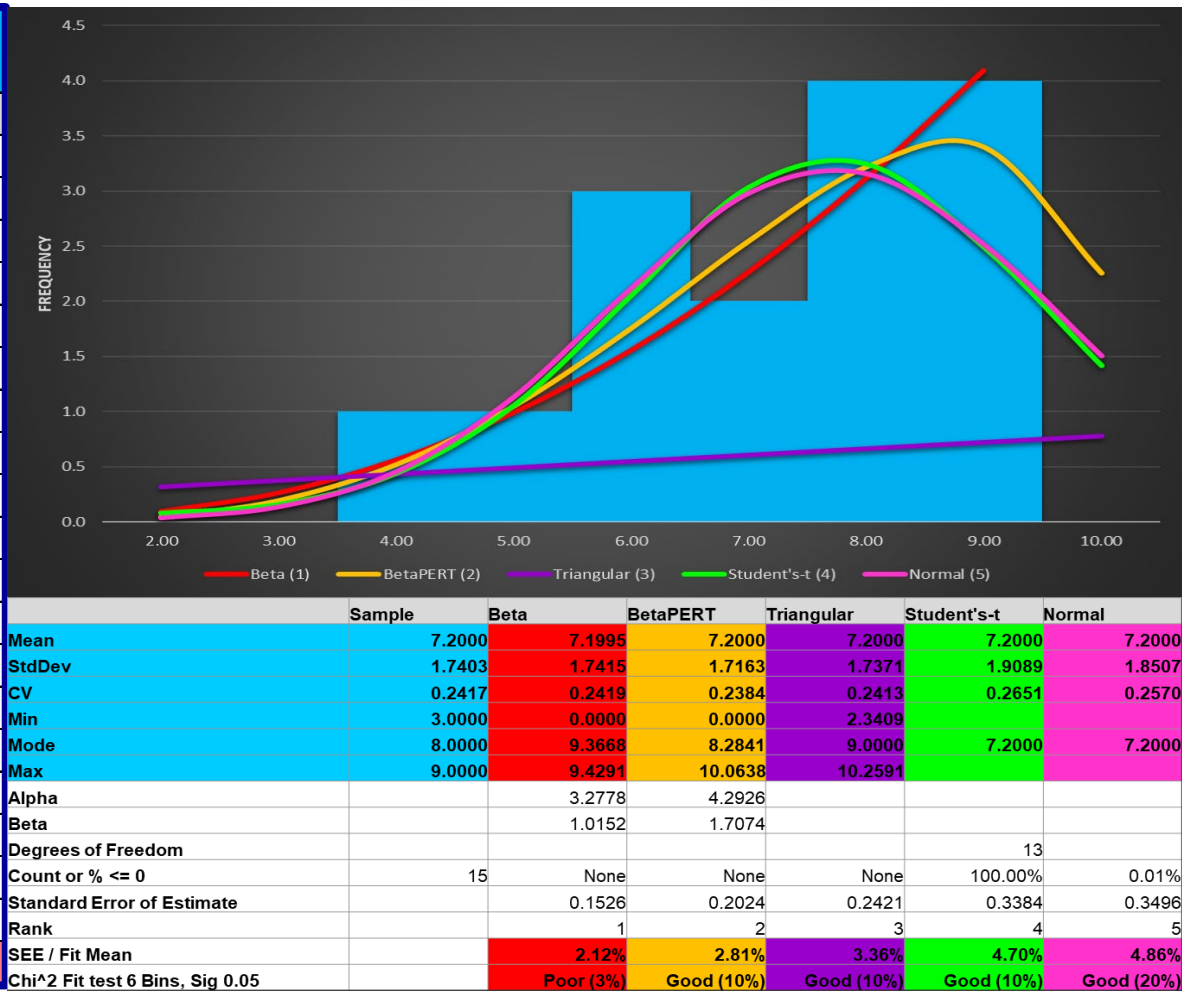


U.S. AIR FORCE

# Cyber Mission Platform: Agile Product Metrics

## Delivered Value Points (Business Value)

Program Increment 2 Features	Bus Value	Status	Delivered Value
<b>Total Committed</b>	<b>108</b>		<b>88</b>
Feature 17	3	Open	0
Feature 18	9	Closed	9
Feature 19	5	Closed	5
Feature 20	6	Closed	6
Feature 21	8	Closed	8
Feature 22	8	Closed	8
Feature 23	-	Closed	0
Feature 24	6	Closed	6
Feature 25	8	Closed	8
Feature 26	9	Blocked	0
Feature 27	9	Closed	9
Feature 28	6	Closed	6
Feature 29	-	Closed	0
Feature 30	-	Closed	0
Feature 31	7	Closed	7
Feature 32	7	Closed	7
Feature 33	9	Closed	9
Feature 34	-	Closed	0
Feature 35	8	Open	0
<b>Total Uncommitted</b>	<b>0</b>		<b>14</b>



- Pairing Feature Completion with Business Value provides insight into how much Value was delivered to the User within a PI

# Agile Product Metrics

## Level of User Satisfaction

---

“The measure of user satisfaction based on the **value being delivered** by the product or solution”

“User satisfaction is a good indicator of value delivery since user satisfaction will most likely increase as value is **consistently delivered** and **quality** is improved”

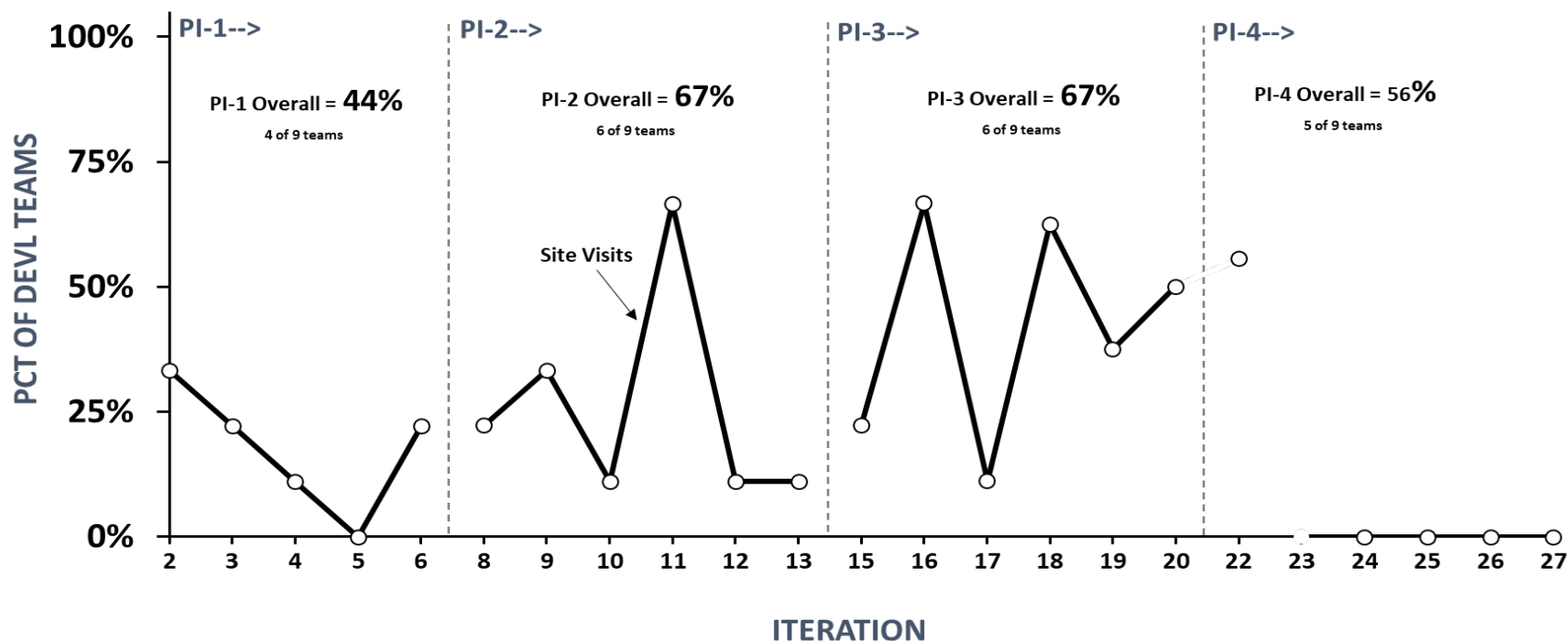


U.S. AIR FORCE

# Cyber Mission Platform: Agile Product Metrics

## Assessing User Engagement

Pct of Devl Teams Engaging the Operators



- User happiness may be difficult to measure but user interactions are a step in the right direction
- Percentage of Teams interacting with Users at least once within an Iteration is depicted
- How well are we engaging the users?



# Agile DevSecOps Metrics

## Lead Time (Cycle Time)

“The flow metric that represents **how long it takes for a requirement to be delivered**”

“Being able to achieve a **measurable and predictable lead time** for new work would be a significant achievement in improving accuracy and **predictability of estimates**”

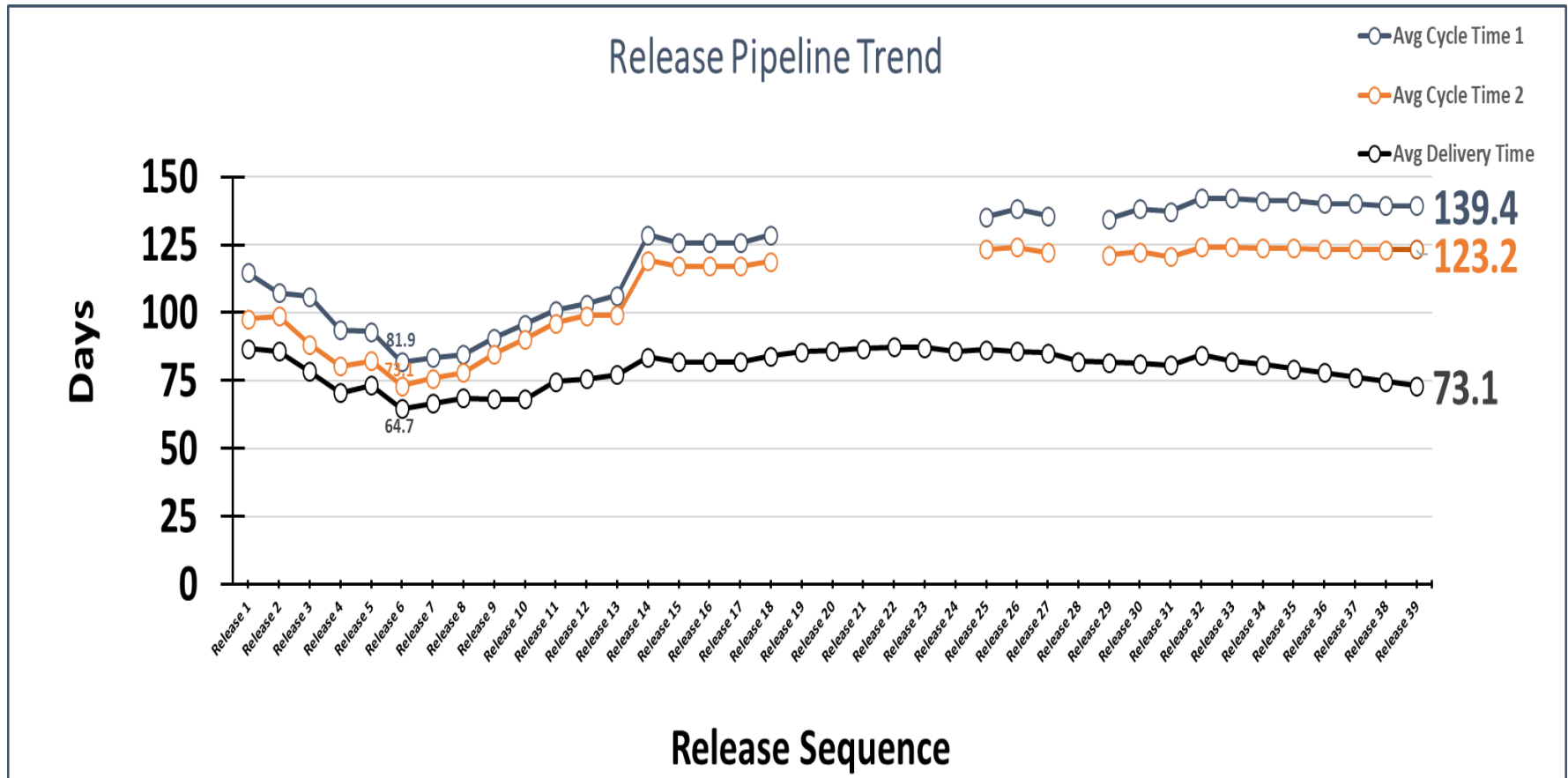
“Improving lead time requires addressing the flow related issues that lead to unpredictability, which requires a **commitment from everyone** involved including the user community and product stakeholders”



U.S. AIR FORCE

# Cyber Mission Platform: Agile DevSecOps Metrics

## Lead Time (Cycle Time)



- **Cycle Time 1** – Time from items created in JIRA to Fielded at Site
- **Cycle Time 2** – Time from development start to Fielded at Site
- **Delivery Time** – Time from development completion to Fielded at Site
- **Avg Time from development start to development completion** =  $(123.2 - 73.1) = 50.1$  Days



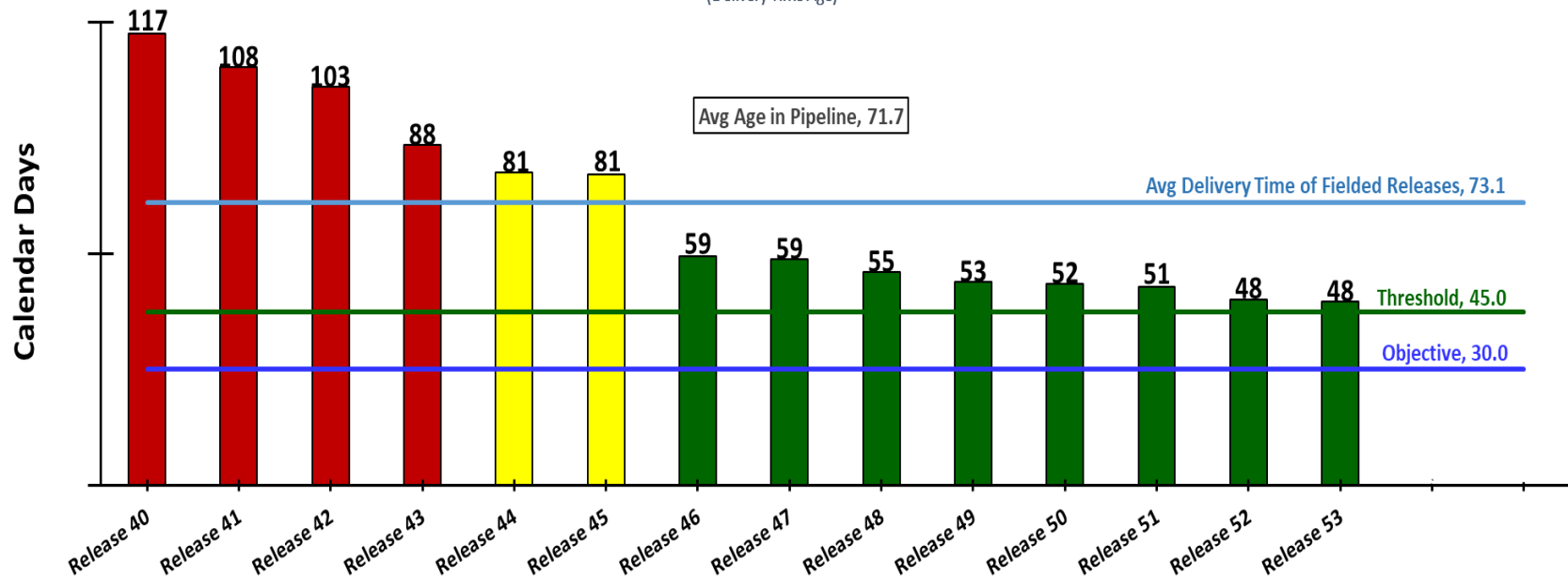
U.S. AIR FORCE

# Cyber Mission Platform: Agile Metrics

## Lead Time (Cycle Time)

Age of Releases in Pipeline

(Delivery Time Age)



Release	Feature(s) in Release	Business Value	Days in Pipeline
40	1	10	117
41	1	NA	108
50	3	21	52

- Staging and Test activities averaging an additional 73.1 days beyond development completion...why? Number of Release? Level of Automated Test?



# Agile DevSecOps Metrics

## Deployment Frequency

“Deployment frequency provides information on the **cadence of deployments** in terms of time period between deployments”

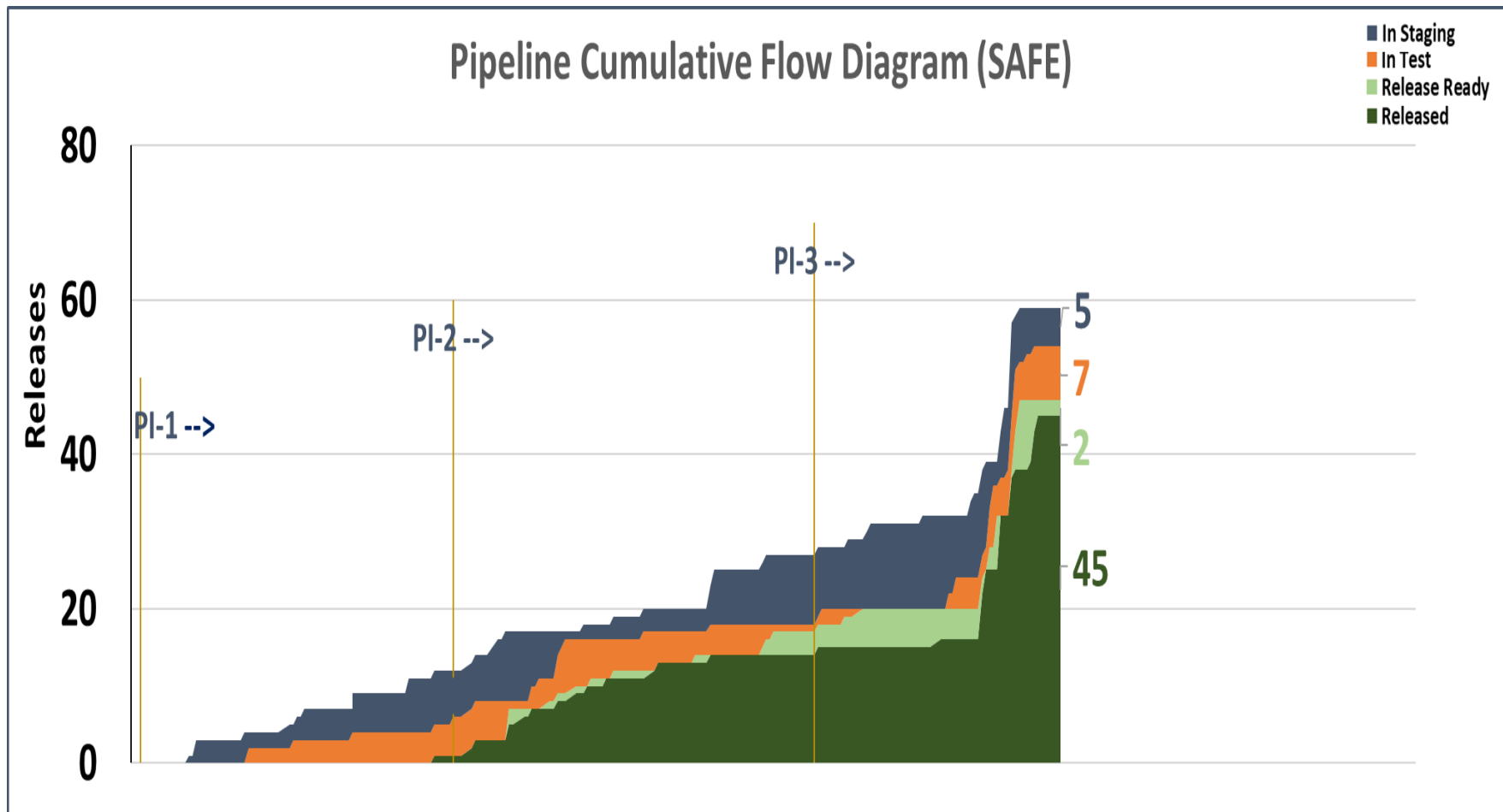
“Lean-Agile practices promote **fast feedback** and delivering value to the customer. In order to accomplish this, the team must develop a regular delivery cadence. ”



U.S. AIR FORCE

# Cyber Mission Platform: Agile Metrics

## Deployment Frequency



- Development teams producing more releases than pipeline can handle?
- Need for additional levels of Automated Test?



U.S. AIR FORCE

# Cyber Mission Platform: Agile Metrics Pipeline Dashboard

PIPELINE									
Status				Measures					
<div>Releases in the Pipeline14releases</div> <div>Avg Age in Pipeline72calendar days</div> <div>Oldest in Pipeline117calendar days</div> <div>Oldest 10 in Pipeline83calendar days</div> <div>Last release19 Novwas8cal days ago</div> <div>PI-4 Releases (so far)11releases</div>				Pipeline Times		AVG DELIVERY TIME			
				All Rel					
				*Cycle Time	123.2calendar days				
				**Delivery Time	73.1calendar days				
				Delivery Time Breakdown					
Integration		41.8calendar days	Releases Created in PI-1		82.7	19 of 19 releases created in PI-1 are fielded			
Install Rehearsal		21.8calendar days	Releases Created in PI-2		104.7	10 of 10 releases created in PI-2 are fielded			
Install Prod (S1)		11.5calendar days	Releases Created in PI-3		67.7	12 of 18 releases created in PI-3 are fielded			
			Releases Created in PI-4		35.1	3 of 11 releases created in PI-4 are fielded			
			Releases Fielded in PI-1		52.9	6 releases fielded during PI-1			
			Releases Fielded in PI-2		75.6	15 releases fielded during PI-2			
			Releases Fielded in PI-3		87.5	18 releases fielded during PI-3			
			Releases Fielded in PI-4		52.1	11 releases fielded during PI-4 (so far)			
All days above are calendar days									
*Cycle Time - Time from Development startup until fielded at Site 1				**Delivery Time - Time from Development completion until fielded at Site 1				MTBRel = Mean Time Between Releases	

- Collaboration between SPO and OEM to measure and assess DevSecOps Metrics



U.S. AIR FORCE

---

# **Recommended**

## ***Minimum Set of Metrics***



# Recommended “Minimum Set of Metrics”

## Process Metrics

- Story Points
- Velocity
- **Story Completion Rate**
- Sprint Burndown
- **Release Burn-Up**

## Quality Metrics

- Recidivism
- **Defect Count**
- Number of Blockers

## Product Metrics

- **Delivered Features**
- Delivered Value Points
- **Level of User Satisfaction**

## DevSecOps Metrics

- Mean Time to Restore (MTTR)
- Deployment Frequency
- Change Fail Rate

- Editorial – Most useful for cost analysis purposes in a large program



U.S. AIR FORCE

# Recommended “Minimum Set of Metrics” Cost and Value

## Cost Metrics

- Total Cost Estimate
- Burn Rate

## Value Metrics (Cost)

- Cost per Delivered Value Point
- Cost per Delivered Feature
- Cost per Delivered Story Point

## Value Metrics (Delivery)

- Delivered Value Points per Unit of Cost
- Delivered Features per Unit of Cost
- Delivered Story Points per Unit of Cost



U.S. AIR FORCE

---

# Cyber Mission Platform

## *Way Ahead*



U.S. AIR FORCE

# Cyber Mission Platform Way Ahead

**Collect &  
Analyze  
Data**

**Innovate &  
Develop  
Metrics**

**Effective  
Decision  
Analysis  
Support**







U.S. AIR FORCE

---

# Backup/Sandbox

- **The Milestone Decision Authority (MDA) should immediately require the Program Manager (PM) to build a program-appropriate framework for status estimation. Example metrics include:**
  - Sprint Burndown
  - Epic and Release Burndown
  - Velocity
  - Control Chart
  - Cumulative Flow Diagram

# Defense Innovation Board (DIB): Recommended Metrics for SW Dev (July 2018)

#	Metric	Target value (by software type) <sup>i</sup>				Typical DoD values for SW
		COTS <sup>ii</sup> apps	Custom -ized SW <sup>iii</sup>	COTS HW/OS <sup>iv</sup>	Real-time HW/SW <sup>v</sup>	
1	Time from program launch to deployment of simplest useful functionality	<1 mo	<3 mo	<6 mo	<1 yr	3-5 yrs
2	Time to field high priority fcn (spec → ops) or fix newly found security hole (find → ops) <sup>vi</sup>	N/A <1 wk	<1 mo <1 wk	<3 mo <1 wk	<3 mo <1 wk	1-5 yrs 1-18 m
3	Time from code committed to code in use	<1 wk	<1 hr	<1 da	<1 mo	1-18 m
4	Time req'd for full regression test (automat'd) and cybersecurity audit/penetration testing <sup>vii</sup>	N/A <1 mo	<1 da <1 mo	<1 da <1 mo	<1 wk <3 mo	2 yrs 2 yrs
5	Time required to restore service after outage	<1 hr	<6 hr	<1 day	N/A	?
6	Automated test coverage of specs / code	N/A	>90%	>90%	100%	?
7	Number of bugs caught in testing vs field use	N/A	>75%	>75%	>90%	?
8	Change failure rate (rollback deployed code)	<1%	<5%	<10%	<1%	?
9	% code available to DoD for inspection/rebuild	N/A	100%	100%	100%	0%
10	Complexity metrics	#/type of specs structure of code #/type of platforms		# programmers #/skill level of teams #/type deployments		Partial/ manual tracking
11	Development plan/environment metrics					
12	"Nunn-McCurdy" threshold (for any metric)	1.1X	1.25X	1.5X	1.5X each effort	1.25X Total \$

# Government Accountability Office (GAO): Agile Guide 2019 (Draft)

---

- **Metrics should be appropriate at individual levels of the organization and be quantifiable, meaningful, repeatable, and actionable**
  - Lead Time
  - Cycle Time
  - How frequently a Feature is delivered and its Value
- **Releases and items in the product backlog, such as Epics and User Stories, should be captured within the road map**
- **Agile teams should establish appropriate metrics early...and evaluate a development team from the start**
  - Velocity
  - Features or Stories delivered
  - Customer Satisfaction

# OUSD(A&S) Full Metrics Descriptions

---

- **Process Metrics**

- Story Points – relative sizing of stories
- Velocity – the amount of work (usually in story points) that the team completes in a given sprint (or release)
- Velocity Variance – standard deviation from average velocity or difference from the mean velocity
- Velocity Predictability – the difference between planned and completed velocity
- Story Completion Rate – the number of stories completed in a given sprint (or release)
- Sprint Burndown Chart – used to estimate a team's pace of work accomplished daily
- Release Burnup – used to measure the amount of work completed for a given release based on the total amount of work planned for the release
- Cumulative Flow Diagram – used to visualize the flow of work through a process

# OUSD(A&S) Full Metrics Descriptions

---

- **Quality Metrics**

- Recidivism – stories that are returned to the team for various reasons
- First-Time Pass Rate – the number of stories, features, or capabilities that pass the first time
- Defect Count – the number of defects per sprint (or release)
- Test Coverage – provides insight into the level of testing that is integrated within the end-to-end development value stream process
- Number of Blockers – the number of events that prohibit the completion of an activity or work item

- **Product Metrics**

- Delivered Features – the count of delivered features accepted and delivered (some teams may prefer to measure delivered capabilities)
- Delivered Value Points – the count of value points delivered to users for a given release
- Level of User Satisfaction – the measure of user satisfaction based on the value delivered by the product or solution

# OUSD(A&S) Full Metrics Descriptions

---

- **DevSecOps Metrics**

- Mean Time to Restore – how quickly a system or solution can be restored to functional use after a critical failure
- Deployment Frequency – cadence of deployments in terms of time elapsed between deployments
- Lead Time – flow metric for how long it takes to deliver a required solution
- Change Fail Rate – rate of number of changes that do not pass

# OUSD(A&S)

## Full Metrics Descriptions

---

- **Cost Metrics**

- Total Cost Estimate – total estimated cost for the product being developed and/or service being acquired
- Agile Team Cost – size and annual cost of the development teams to include the average size and makeup as well as number of teams
- Total Hardware, Software, Cloud, and Licensing Costs – cost of hardware, software, and licensing fees
- Total Program Management Costs – size and annual cost of the program management team (includes Government and contractor)
- Allocation of Development Costs – allocation of development costs related to defect resolution, new feature or capability implementation, and code refactoring (technical debt)
- Percentage of Resources by Function – percentage of programmers, designers, user interface engineers, system architects, and other key development categories within an agile team
- Software Licensing Fees – license costs related to software and cloud-based application services



# OUSD(A&S)

## Full Metrics Descriptions

---

- **Cost Metrics (Cont.)**

- Computing Costs – license costs related to cloud-based application and computing services
- Bandwidth Costs – bandwidth costs
- Storage Costs – fees for storage costs
- Other Costs – costs not associated with one of the above categories
- Burn Rate – incurred cost over a period of time (monthly burn rate)

- **Value Metrics**

- Cost per Delivered Value Point = Cost to Date / Total Delivered Value Points
- Costs per Delivered Feature = Total Cost to Date / Total Delivered Features
- Cost per Delivered Story Point = Cost to Date / Total Delivered Story Points
- Delivered Value Points per Unit of Cost = Total Delivered Value Points / Cost to Date
- Delivered Features per Unit of Cost = Total Delivered Features / Cost to Date
- Delivered Story Points per Unit of Cost = Total Delivered Story Points / Cost to Date