AF Cyber Mission Platform (CMP)
Analyzing the Full Suite of Agile Metrics
“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**
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
• 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**

**CMP utilizes Confluence and JIRA**

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

### Confluence

<table>
<thead>
<tr>
<th>Team Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Development Teams</strong></td>
</tr>
<tr>
<td>App 1</td>
</tr>
<tr>
<td>App 4</td>
</tr>
<tr>
<td>App 7</td>
</tr>
<tr>
<td><strong>Systems and Shared Services Teams</strong></td>
</tr>
<tr>
<td>Test</td>
</tr>
<tr>
<td>Shared Services</td>
</tr>
<tr>
<td>Integrated Systems Support</td>
</tr>
<tr>
<td>Growth Board</td>
</tr>
</tbody>
</table>

### Total Team Sizing - CMP

![Bar chart showing team size over time](chart.png)
This should be easy, right?

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

My work here is done?
“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”
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
“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”
Cyber Mission Platform: Agile Process Metrics
Velocity Predictability

How well are we planning and executing?

- Issue Points by Iteration
  - Issue = Story, Bug, Task

- Requirement → Feature → Issue (Story) → Issue Point(s)

- Present for the ICEAA 2021 Online Workshop - www.iceaaonline.com
• 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
• How stable is our plan? Is stability improving as the transformation advances through several PIs?
• Iteration Plan Stability = 1 – (Nbr of Issues Added + Nbr of Issues Removed) / Nbr of Issues Committed
“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.”
• Issue completion (Story, Bug, Task) is more meaningful to the user than Story Points
# Cyber Mission Platform: Agile Process Metrics

## Predictability Goals Dashboard

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

- **Collaboration between SPO and OEM to measure and assess process metrics**
“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.”
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

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

Presented for the ICEAA 2021 Online Workshop - www.iceaaonline.com
• PI 2 completed 19 Features with progress towards an additional 19 (17 Committed, 2 Uncommitted)
“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”
- PI 3 adds additional Features; Blockers prevent completion
“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.”
<table>
<thead>
<tr>
<th>Program Increment 2 Features</th>
<th>Bus Value</th>
<th>Status</th>
<th>Delivered Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Committed</strong></td>
<td>108</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>Feature 17</td>
<td>3</td>
<td>Open</td>
<td>0</td>
</tr>
<tr>
<td>Feature 18</td>
<td>9</td>
<td>Closed</td>
<td>9</td>
</tr>
<tr>
<td>Feature 19</td>
<td>5</td>
<td>Closed</td>
<td>5</td>
</tr>
<tr>
<td>Feature 20</td>
<td>6</td>
<td>Closed</td>
<td>6</td>
</tr>
<tr>
<td>Feature 21</td>
<td>8</td>
<td>Closed</td>
<td>8</td>
</tr>
<tr>
<td>Feature 22</td>
<td>8</td>
<td>Closed</td>
<td>8</td>
</tr>
<tr>
<td>Feature 23</td>
<td>-</td>
<td>Closed</td>
<td>0</td>
</tr>
<tr>
<td>Feature 24</td>
<td>6</td>
<td>Closed</td>
<td>6</td>
</tr>
<tr>
<td>Feature 25</td>
<td>8</td>
<td>Closed</td>
<td>8</td>
</tr>
<tr>
<td>Feature 26</td>
<td>9</td>
<td>Blocked</td>
<td>0</td>
</tr>
<tr>
<td>Feature 27</td>
<td>9</td>
<td>Closed</td>
<td>9</td>
</tr>
<tr>
<td>Feature 28</td>
<td>6</td>
<td>Closed</td>
<td>6</td>
</tr>
<tr>
<td>Feature 29</td>
<td>-</td>
<td>Closed</td>
<td>0</td>
</tr>
<tr>
<td>Feature 30</td>
<td>-</td>
<td>Closed</td>
<td>0</td>
</tr>
<tr>
<td>Feature 31</td>
<td>7</td>
<td>Closed</td>
<td>7</td>
</tr>
<tr>
<td>Feature 32</td>
<td>7</td>
<td>Closed</td>
<td>7</td>
</tr>
<tr>
<td>Feature 33</td>
<td>9</td>
<td>Closed</td>
<td>9</td>
</tr>
<tr>
<td>Feature 34</td>
<td>-</td>
<td>Closed</td>
<td>0</td>
</tr>
<tr>
<td>Feature 35</td>
<td>8</td>
<td>Open</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total Uncommitted</strong></td>
<td>0</td>
<td></td>
<td>14</td>
</tr>
</tbody>
</table>

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

Presented for the ICEAA 2021 Online Workshop - www.iceaaonline.com
“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”
• 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?
“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”
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

Presented for the ICEAA 2021 Online Workshop - www.iceaaonline.com
Cyber Mission Platform: Agile Metrics
Lead Time (Cycle Time)

- Staging and Test activities averaging an additional 73.1 days beyond development completion…why? Number of Release? Level of Automated Test?
“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.”
Development teams producing more releases than pipeline can handle?
Need for additional levels of Automated Test?
## Pipeline Times

<table>
<thead>
<tr>
<th>Status</th>
<th>Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Releases in the Pipeline</td>
<td>AVG DELIVERY TIME</td>
</tr>
<tr>
<td>Avg Age in Pipeline</td>
<td>Releases Created in PI-1</td>
</tr>
<tr>
<td>Oldest in Pipeline</td>
<td>Releases Created in PI-2</td>
</tr>
<tr>
<td>Oldest 10 in Pipeline</td>
<td>Releases Created in PI-3</td>
</tr>
<tr>
<td>Last release</td>
<td>Releases Created in PI-4</td>
</tr>
<tr>
<td></td>
<td>Releases Fielded in PI-1</td>
</tr>
<tr>
<td></td>
<td>Releases Fielded in PI-2</td>
</tr>
<tr>
<td></td>
<td>Releases Fielded in PI-3</td>
</tr>
<tr>
<td></td>
<td>Releases Fielded in PI-4</td>
</tr>
<tr>
<td></td>
<td>MTBRel = Mean Time Between Releases</td>
</tr>
</tbody>
</table>

### Pipeline Times

- **Release Fielded in PI-1**: 6 releases fielded during PI-1
- **Release Fielded in PI-2**: 15 releases fielded during PI-2
- **Release Fielded in PI-3**: 18 releases fielded during PI-3
- **Release Fielded in PI-4**: 11 releases fielded during PI-4 (so far)

### Delivery Time Breakdown

- **Integration**: 41.8 calendar days
- **Install Rehearsal**: 21.8 calendar days
- **Install Prod (S1)**: 11.5 calendar days
- **MTBRel**: 7.0 calendar days

### Average Delivery Time

- **PI-1**: 82.7 calendar days
- **PI-2**: 104.7 calendar days
- **PI-3**: 67.7 calendar days
- **PI-4**: 35.1 calendar days

### Notes

- **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**

---

Presented for the ICEAA 2021 Online Workshop - www.iceaaonline.com
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
Recommended “Minimum Set of Metrics”
Cost and Value

<table>
<thead>
<tr>
<th>Cost Metrics</th>
<th>Value Metrics (Cost)</th>
<th>Value Metrics (Delivery)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Total Cost Estimate</td>
<td>• Cost per Delivered Value Point</td>
<td>• Delivered Value Points per Unit of Cost</td>
</tr>
<tr>
<td>• Burn Rate</td>
<td>• Cost per Delivered Value Point</td>
<td>• Delivered Features per Unit of Cost</td>
</tr>
<tr>
<td></td>
<td>• Cost per Delivered Feature</td>
<td>• Delivered Story Points per Unit of Cost</td>
</tr>
<tr>
<td></td>
<td>• Cost per Delivered Story Point</td>
<td></td>
</tr>
</tbody>
</table>
Cyber Mission Platform

Way Ahead
Cyber Mission Platform
Way Ahead

Collect & Analyze Data

Innovate & Develop Metrics

Effective Decision Analysis Support

Presented for the ICEAA 2021 Online Workshop - www.iceaaonline.com
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)

<table>
<thead>
<tr>
<th>#</th>
<th>Metric</th>
<th>Target value (by software type)</th>
<th>Typical DoD values for SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Time from program launch to deployment of simplest useful functionality</td>
<td>COTS apps &lt;1 mo, Customized SW &lt;3 mo, COTS HW/OS &lt;6 mo, Real-time HW/SW &lt;1 yr</td>
<td>3-5 yrs</td>
</tr>
<tr>
<td>2</td>
<td>Time to field high priority fcn (spec → ops) or fix newly found security hole (find → ops)</td>
<td>N/A &lt;1 mo, &lt;1 wk</td>
<td>1-5 yrs, 1-18 m</td>
</tr>
<tr>
<td>3</td>
<td>Time from code committed to code in use</td>
<td>&lt;1 wk, &lt;1 hr</td>
<td>1-18 m</td>
</tr>
<tr>
<td>4</td>
<td>Time req’d for full regression test (automat’d) and cybersecurity audit/penetration testing</td>
<td>N/A &lt;1 da, &lt;1 mo</td>
<td>2 yrs, 2 yrs</td>
</tr>
<tr>
<td>5</td>
<td>Time required to restore service after outage</td>
<td>&lt;1 hr, &lt;6 hr, &lt;1 day</td>
<td>N/A</td>
</tr>
<tr>
<td>6</td>
<td>Automated test coverage of specs / code</td>
<td>N/A &gt;90%, &gt;90%, 100%</td>
<td>?</td>
</tr>
<tr>
<td>7</td>
<td>Number of bugs caught in testing vs field use</td>
<td>N/A &gt;75%, &gt;75%, &gt;90%</td>
<td>?</td>
</tr>
<tr>
<td>8</td>
<td>Change failure rate (rollback deployed code)</td>
<td>&lt;1%, &lt;5%, &lt;10%, &lt;1%</td>
<td>?</td>
</tr>
<tr>
<td>9</td>
<td>% code available to DoD for inspection/rebuild</td>
<td>N/A 100%, 100%, 100%</td>
<td>0%</td>
</tr>
<tr>
<td>10</td>
<td>Complexity metrics</td>
<td>#/type of specs structure of code</td>
<td>Partial/manual tracking</td>
</tr>
<tr>
<td>11</td>
<td>Development plan/environment metrics</td>
<td>#/type of platforms</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>“Nunn-McCurdy” threshold (for any metric)</td>
<td>1.1X 1.25X 1.5X</td>
<td>1.25X Total $</td>
</tr>
</tbody>
</table>
• 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
• **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
• **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
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
• **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
• **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