AE2A in C2ISR: Right Metrics for the Right Audience

Matt Hoffman (AFLCMC/HBGF)
Kyle Davis (AFLCMC/HBG, Quantech Services)
Elizabeth Ashwood (AFLCMC/HBG, Quantech Services)
Jennifer Lee (AFLCMC/HBG, Quantech Services)
William Lane (AFLCMC/HBF, Quantech Services)
As Software and Information Systems solutions and technologies adapt and modernize, so should the role of cost analysis within the technology domain.

Defense Science Board (DSB) calls for programs to “modernize cost and schedule estimates and measurements” to evolve from a SLOC approach.

National Defense Authorization Act (NDAA) further specifies to “use a modern tracking tool to execute requirements backlog tracking; and use agile development metrics”

The right metrics can provide valuable insight to the right audience to support critical decisions with relevant data.
AE2A in C2ISR: Right Metrics for the Right Audience

Outline

• Command Control Intelligence Surveillance Reconnaissance (C2ISR) – DCGS-AF Background
• Key Agile and Scaled Agile Framework (SAFe) Terminology
• Integrating with the Teams & Understanding Their System Usage
• Metrics & Analysis for Software Development Teams
• Metrics & Analysis for User Organizations & Operators
• Metrics & Analysis for Decision Makers / Senior Leadership
• Successes & Key Lessons Learned
• Way Forward
DCGS Transformation is a large Government led effort implementing Agile Software development practices across the Portfolio starting in 2015

Over 30 individual teams (60+ contracts) within the organization populate information within Confluence and JIRA that our analysts collect and analyze

This data and analysis allows us to maximize the value we can provide to our teams, users, and decision makers within our agile programs
AE2A in C2ISR: Right Metrics for the Right Audience
Key SAFe/Agile Terminology

• **DI2E**: Defense Intelligence Information Enterprise; provides software development tools and resources to DoD personnel, including JIRA

• **JIRA**: Agile program management tool designed to assign and track issue progress

• **Issue**: A work ticket (unit of scope) within the progress tracking tool JIRA; includes Stories, Bugs, Tasks, etc.

• **Agile Development Team**: A team of software developers working issues

• **ART**: Agile Release Train; a group of agile development teams focused on a larger, specific mission domain

• **Users**: User Organizations (ACC) and System Operators who establish requirements and objectives that decompose into issues

• **PI**: Program Increment; a time box (3 months) cadence for agile software development activities
  • Users and Developers establish and prioritize Issues for each ART at the beginning of every PI at Planning Events and evaluate outcomes at the end of every PI at Retrospectives

• **Senior Leadership**: Decision Makers allocating resources and making trade-off decisions

AE2A in C2ISR: Right Metrics for the Right Audience
Key SAFe/Agile Terminology

• **DI2E**: Defense Intelligence Information Enterprise; provides software development tools and resources to DoD personnel, including JIRA

• **JIRA**: Agile program management tool designed to assign and track issue progress

• **Issue**: A work ticket (unit of scope) within the progress tracking tool JIRA; includes Stories, Bugs, Tasks, etc.

• **Agile Development Team**: A team of software developers working issues

• **ART**: Agile Release Train; a group of agile development teams focused on a larger, specific mission domain

• **Users**: User Organizations (ACC) and System Operators who establish requirements and objectives that decompose into issues

• **PI**: Program Increment; a time box (3 months) cadence for agile software development activities
  • Users and Developers establish and prioritize Issues for each ART at the beginning of every PI at Planning Events and evaluate outcomes at the end of every PI at Retrospectives

• **Senior Leadership**: Decision Makers allocating resources and making trade-off decisions
Getting Integrated with Teams & Understanding Their System Usage
What tracking tools are we using to monitor program progress? How do I get access?

What are the correct search parameters to apply to get data for the right team(s)? Did I capture it all?

Should I check to ensure the system is being used effectively? Will the analysis be beneficial?
AE2A in C2ISR: Right Metrics for the Right Audience
Complexity of Working with JIRA

**Epic** – A higher-level solution that defines and manages the highest level initiatives in a portfolio. These should be directly translated from incoming 1067 requirements.

**Feature** – A service that fulfills a stakeholder need to be delivered by a single ART in a single PI.

**Enabler** – Activities needed to extend the Architectural Runway to provide future business functionality including exploration, infrastructure, compliance, and architecture development.

**Task** – An objective that must be achieved. Tasks are smaller work items (can be completed in a day or so) that build a story and by itself, is devoid of business benefit.

**Bug** – Problem the impairs product or service functionality.

**Improvement** – An enhancement to an existing feature.

**Capability** – A higher-level solution that defines an Epic’s Minimum Viable Product.

**Spikes** – A type of exploration Enabler Story that represent activities such as research, design, investigation, exploration, and prototyping.

**User Story** – Short descriptions of a small piece of desired functionality (written in the user’s perspective to convey value) sized so they can be completed in a single iteration.

- JIRA is customizable and can be unique to the individual team
- Need to understand how each team / area uses their progress tracking tool
ARTs within AF DCGS use many of the same main issue types, but low-level issue types can still vary from ART to ART.

For our analysis we have consolidated these definitions into overarching themes.
• Requirements decompose into smaller work packets – the expectation is that this is easily traceable; however, in implementation, it is not – Need to evaluate Requirements Traceability
• Higher level issues should have “outward links” whereas lower level issues should only have “inward links”
• Higher percentages of unlinked issues is not always negative
• For instance, we would expect higher percentages of unlinked Bugs than we would expect for Capabilities, Features, or Stories
Almost all features are linked to a higher capability; when linked % of features starts to decline, this might be an indicator of improper system usage.

Critical to monitor requirements traceability rigor.
Audience: The Software Development Teams
AE2A in C2ISR: Right Metrics for the Right Audience
Metrics for Agile Development Teams

- All development teams within an ART join together at PI Planning Events to work together towards the objectives identified and commit to scope they can accomplish over the next PI
  - Historical data and analysis can help the teams throughout their planning cycles

- Key Metrics for Development Teams:
  - Team Planning Health and Stability
  - Cumulative Issue Resolution by Team
Teams can use this data to ensure they are forecasting capacity appropriately.

Cumulative issue productivity shows the average number of issues a team is able to complete in a PI and can be used to inform planning health and stability.

For more experienced teams, we would expect the productivity to be stable over time.
Teams can use this data to ensure they are forecasting capacity appropriately.

Average Story Point Productivity shows the average number of story points a team is able to complete in a PI and can be used to inform planning health and stability.

For more experienced teams, we would expect the productivity to be stable over time.
User Story delivery is a good measure of planning health and ability to complete all Stories planned within a given PI.
Agile theory that a Feature should be sized to complete within a single PI
Understanding historic resolved Features helps assess appropriate sizing and resolution projections

Avg. Issues Per Feature: 7
5 PIs to resolve ~50%
~40% Remaining after 6 PIs

Avg. Issues Per Feature: 4
3 PIs to Resolve ~50%
~27% Remaining after 6 PIs

Avg. Issues Per Feature: 4
2 PIs to Resolve ~50%
~27% Remaining after 6 PIs

Avg. Issues Per Feature: 5
4 PIs to Resolve ~50%
~30% Remaining after 6 PIs
AE2A in C2ISR: Right Metrics for the Right Audience
Team Planning Health Trend Analysis: Capabilities

- Capabilities are larger and can span multiple PIs
- Assist teams by assessing planning velocity and identifying disconnects / accounting for Non-Requirements scope in planning capacity (Deficiencies, Enhancements, Non-Functional)

Avg. Features Per Capability:
- PI 4: 5 ~20% Remaining after 10 PIs
- PI 7: 2 60% Remaining after 7 PIs
Audience: User Organizations and System Operators
Users set the objectives for each PI and care about when they will have capabilities and features resolved.

Key Metrics for Users:
- Capability Sizing
- Capability Resolution / Productivity
- Feature Resolution / Productivity
AE2A in C2ISR: Right Metrics for the Right Audience
Metrics for Users: Capability Sizing

- Important to understand Size and Complexity for historic Capabilities when considering Capability Resolution

<table>
<thead>
<tr>
<th>PI Resolution</th>
<th>Summary</th>
<th>Size - Features Per Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI 6</td>
<td>Capability 2800 Ingests Sensor 1 Imagery</td>
<td>4</td>
</tr>
<tr>
<td>PI 7</td>
<td>Capability 2814 Store and Catalog Secondary Product</td>
<td>3</td>
</tr>
<tr>
<td>PI 8</td>
<td>Capability 2808 Geospatial Visualization and SA</td>
<td>24</td>
</tr>
</tbody>
</table>

PI 10 Major Test Event – all work focused primarily on TPR resolution
Resolution of Capability is the most important metric.

Understanding workload prioritization and effort towards Non-Capability scope (Deficiencies, Enhancements, Non-Functional) is critical to assessing Capability Resolution timelines.
• Feature productivity is also critical
• Feature productivity is also dependent on prioritization of Non-Feature scope (Deficiencies, Enhancements, Non-Functional)
AE2A in C2ISR: Right Metrics for the Right Audience
“Definition of Done” JIRA Resolution Types

<table>
<thead>
<tr>
<th>Resolution Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Done</td>
<td>The issue has met all criteria defined in a program’s definition of done; in most cases this means that tests have been completed, the solution has been demonstrated, and solution was accepted.</td>
</tr>
<tr>
<td>Duplicate</td>
<td>The issue and its purpose are duplicated by another issue in JIRA, common when multiple teams are working towards a common issue.</td>
</tr>
<tr>
<td>Fixed</td>
<td>Any issue that has been marked as “ready for integration test” and has been resolved; most common in Bugs and Features.</td>
</tr>
<tr>
<td>OBE</td>
<td>Overcome By Events, the particular issue will no longer have effort expended to resolve and deliver a result.</td>
</tr>
</tbody>
</table>

- Issues in JIRA that have been resolved can be placed in multiple categories of Resolution Status.
- Important to not misinterpret “resolved” as “delivered” or “operationally fielded.”
Although there has been an increase in issues resolved as Duplicate, OBE, and Fixed, current cumulative percentage is minimal at ~6%

The key is to keep Users informed and aware that small percentages of issues are being resolved but not delivered
Audience: Senior Leadership
AE2A in C2ISR: Right Metrics for the Right Audience
Metrics for Senior Leadership

- Senior Leaders need to understand allocation of work and how user requirements are being balanced with technical debt; this is especially important in situations where Gov’t is the Lead Integrator and manage risks such as
  - Maintaining Requirements Traceability Rigor / System Usage
  - Shifting work allocation without Senior Leader weigh-in

- Key Metrics for Senior Leadership
  - Work Allocation – Requirements/Functions vs Technical Debt
  - Cost Effort Associated with Delivered Capability/Feature
AE2A in C2ISR: Right Metrics for the Right Audience
Data Analysis For Senior Leadership – Work Allocation

- Work Allocation Analysis – providing Leadership insight into WIP towards Requirements vs. WIP towards other necessities
- Accounting for productivity towards Non-Requirement scope is critical when establishing Capability / Feature timelines and allocating resources
AE2A in C2ISR: Right Metrics for the Right Audience
Data Analysis For Senior Leadership – Work Allocation

- Accounting for productivity towards Non-Requirement scope is critical when establishing Capability / Feature timelines and allocating resources
• Traditional cost metrics still apply! Understanding Capability Sizing and Effort Months required for a Capability allows Decision Makers to make trade-offs based on Value of a Capability
• CAUTION: This view can be misleading as all effort months for a Capability are attributed to only one single PI rather than capturing progress made in prior PIs
As additional data is collected each PI the Cumulative Effort Months Per Resolved Capability will become steady and reflect the amount of work necessary from inception to completion.

Traditional cost metrics still apply! Understanding Capability Sizing and Effort Months required for a Capability allows Decision Makers to make trade-offs based on Value of a Capability.
• Don’t forget about all of the other costs; System Level Support is inclusive of Program Management, Systems Engineering, Test Teams, etc.
• More budget allocated here will reduce budget available for capabilities...
AE2A in C2ISR: Right Metrics for the Right Audience
Data Analysis For Senior Leadership

- This process can be applied to assess capability resolution after accounting for work allocation towards Non-Requirements scope.
Successes & Key Lessons Learned
Successful integration is crucial to ensuring effective system usage and providing critical information to teams, users, and leadership and can lead to:

- Recurring time slots at PI Planning Events to brief latest metrics
- System usage rigor through Rules of Engagement implementation

### Rules of Engagement

<table>
<thead>
<tr>
<th>Rule</th>
<th>Required or Best Practice</th>
<th>Product Owner/SM Responsible</th>
<th>QC before opening a sprint</th>
<th>QC before closing a sprint</th>
<th>QC at creation of issue</th>
<th>Cost Analysis Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>User the &quot;Story&quot; ticket with the book as an icon ≠ not the &quot;User Story&quot; ticket with a green plus ≠ for user stories. The Green Plus ≠ is also the icon for &quot;Features&quot; and this cannot be changed. Stories should be linked to corresponding features as &quot;Child Tasks&quot;. NOTE: This would be a good area for ART standardization.</td>
<td>Required</td>
<td>Product Owner</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Make sure all the user Stories are in priority order in the backlog and in the sprint. The top ticket should be the most important. Last ticket should be least important.</td>
<td>Best Practice</td>
<td>Product Owner</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Make sure the team decides on the number of user story points for each user story or spike. Use 1, 2, 3, 5 or 8 points per ticket. If a ticket is larger than 8 points it should be broken into smaller user stories or spikes.</td>
<td>Required</td>
<td>Scrum Master</td>
<td>Yes</td>
<td></td>
<td>Yes (Critical)</td>
<td></td>
</tr>
<tr>
<td>Always do a good scrub of the backlog and the sprint prior to starting the sprint.</td>
<td>Required</td>
<td>Product Owner</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Once a sprint has started try to limit scope change (adding new tickets) to the sprint as much as possible.</td>
<td>Best Practice</td>
<td>Scrum Master</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>When adding new user stories always fill in:</td>
<td>Required</td>
<td>Product Owner</td>
<td>Yes</td>
<td>Yes (Critical)</td>
<td>All 4 elements are critical</td>
<td></td>
</tr>
<tr>
<td>• Value Stream(s): GEOINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Release Train(s): High Altitude</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Epic: Pick from list</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Tasked Team: Pick from list</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## AE2A in C2ISR: Right Metrics for the Right Audience

### Way Forward

<table>
<thead>
<tr>
<th>Data Analysis</th>
<th>Reporting Findings</th>
<th>Capability Roadmap</th>
<th>Continue Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Collect and analyze new data every 3 months in step with DCGS ART cadence</td>
<td>• Continue to provide data and visualization to each key focus area (Teams, Users, Leadership) to inform decision making</td>
<td>• Use applicable data to forecast Capability Delivery timelines • Account for strategic focus changes over time</td>
<td>• Integrate additional information identified as beneficial and continue integration into additional ARTs</td>
</tr>
</tbody>
</table>
Fail fast, fail often – but learn, adapt and forecast accurate capability delivery timelines based on all available data
Back Up
AE2A in C2ISR: Right Metrics for the Right Audience
Getting Integrated with the Team

- **PI Planning Events** – critical starting point; allows for understanding of User-defined requirements / objectives and how the Teams decompose work into smaller action items
- **Prepare for PI Planning** – collect and analyze data from JIRA to share with teams as necessary
- **Inspect & Adapt** – hear and understand how progress went during the PI from the teams
**JIRA** is used for progress tracking in C2ISR (and throughout PEO Digital)

- **Issues Navigator**, depicted, is where programmatic tickets for Epics, Capabilities, Features, and Stories are able to be exported for detailed analysis
If the tool is not utilized correctly and links are not properly created...then insight into traceability of work decomposition (Capability -> Feature -> Story -> Task) may be lost.
Initial research in one DCGS ART led us to believe that our teams had stopped utilizing JIRA.

What Happened?
- Team began complete re-structure of epics in PI 6-8
- Proper Epics were not communicated and work done was not being captured
- Integration with the team was crucial for understanding what happened and what new project boards and keys were in use

Need to know what information you’re looking for and how to find it...not as easy as one simple dashboard
AE2A in C2ISR: Right Metrics for the Right Audience
Metrics for Users: Capability Sizing

- Important to understand Size and Complexity for historic Capabilities when considering Capability Resolution

Features Per Resolved Capability Histogram

10% of Resolved Capabilities were >8 Features in Size
• We are in the process of collecting metrics to provide leadership with a database of resolved Capabilities, with which analogies can be drawn for future Capabilities
• Shown above are all Capabilities within ART 2; inclusive of those remaining Unresolved
Accounting for productivity towards Non-Requirement scope is critical when establishing Capability / Feature timelines and allocating resources.
As additional data is collected each PI the Cumulative Effort Months Per Resolved Capability will become steady and reflect the amount of work necessary from inception to completion.

An understanding of work allocation also allows for understanding of EM Scaled to encompass only Requirement satisfying efforts.
• Traditional cost metrics still apply! Understanding Feature Sizing and Effort Months required for a Feature allows Decision Makers to make trade-offs based on Value of a Feature
• CAUTION: This view can be misleading as all effort months for a Capability are attributed to only one single PI rather than capturing progress made in prior PIs
As additional data is collected each PI the Cumulative Effort Months Per Resolved Feature will become steady and reflect the amount of work necessary from inception to completion.

Traditional cost metrics still apply! Understanding Feature Sizing and Effort Months required for a Feature allows Decision Makers to make trade-offs based on Value of a Feature.
As additional data is collected each PI the Cumulative Effort Months Per Resolved Feature will become steady and reflect the amount of work necessary from inception to completion.

An understanding of work allocation also allows for understanding of EM Scaled to encompass only Requirement satisfying efforts.