Cost Estimating Issues for MAIS Programs Using an Agile Approach for SW Development

22 September 2015

Presented by Richard Mabe
Tech Advisor, IT and Electronics Division
AF Cost Analysis Agency

(MAIS Working Group Chair, supporting the CADE Senior Steering Group, OSD CAPE)
Foreword

• This presentation includes ideas and view points gathered from several sources:
  – SW and Info Systems Cost IPT
  – MAIS Working Group
  – SRDR Working Group
  – GAO Expert Meetings
  – SME Experience (Service Cost Agencies; PEO and PMO Staffs)

• None of the information is proprietary
• It has been peer reviewed and adjusted to reflect a set of “working” concepts for discussion
Scope and Definitions

- **What is a MAIS Program?**

**From DODI 5000.02, Table 1:**
- ACAT 1AM or ACAT 1AC
- >$40M* (CY14) in any FY
- >$165M* through deployment
- >$520M* for Life Cycle
- SW Intensive; Incremental^`
- Information Technology Systems
- Defense Business Systems

(*Note 1: Includes all appropriations, regardless of fund source.)
(^Note 2: Program Increments may be treated as separate acquisition programs (Ref. Fig 5)).

- **Not all Software is developed for a MAIS Program, but all MAIS Programs include significant Software development/ configuration/ COTS integration/ new object enhancements/ etc ………..**
“Cost” Compared to “Earned Value”

<table>
<thead>
<tr>
<th>Cost</th>
<th>Common</th>
<th>Earned Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unconstrained</td>
<td>Budget Constrained</td>
<td></td>
</tr>
<tr>
<td>Life Cycle Estimate</td>
<td>Schedule (IMS) Program Plan</td>
<td>Schedule Centric Plan</td>
</tr>
<tr>
<td>Based on history, analogies or engineering buildup, factors, CERs</td>
<td>SW Labor</td>
<td>Based on known Work Packages and future Planned Effort</td>
</tr>
<tr>
<td></td>
<td>SEPM Labor</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IAT Labor</td>
<td></td>
</tr>
<tr>
<td>Reported in dollars, but estimate = $f(\text{Cost, Schedule and Technical Data})$</td>
<td>Reported in dollars, even for Schedule variance; No Technical Measures (e.g., SW progress related to SW Size)</td>
<td></td>
</tr>
<tr>
<td>Used for budget estimating, cost trade-off studies and other support</td>
<td>Used to evaluate and track budget and schedule performance</td>
<td></td>
</tr>
<tr>
<td>Includes uncertainty related to the tech data and cost models</td>
<td>SW Size/Scope</td>
<td>Incorporates the Vendor’s perceived project risk/reserve</td>
</tr>
<tr>
<td></td>
<td>SW Quality</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Productivity</td>
<td></td>
</tr>
<tr>
<td>Based on a Program WBS</td>
<td>Mil-Std-881C WBS Items</td>
<td>Based on a Contract WBS</td>
</tr>
</tbody>
</table>

Issues related to Agile affect the Common Ground between Cost and EVM
Discussion of the “Issues”
(Not all Issues are Problems)
Agile Process

- Focus is on what features can be delivered per iteration
- Not fully defined what functionality will be delivered at the end of each iteration
- Full Working System functionality likely requires multiple iterations
"Working System" = Sum of the Iterations, following System Level Integration and Testing

- Iterations do not align with Delivered SW functional "End Items" specified by User Requirement
- Iterations do not align with EVM Work Packages – which may or may not match SW End Item
- Backlog is inherently stacked by priority – or by need-at-the-moment – or by what?
- Lines between the SW Team, Systems Engineers, Contractor PM and Gov’t PM are squishy

Who manages the requirement, sets the backlog priority, and ensures the user gets what they asked for?

Where does “Agile” fit – what is the “Agile” portion and metrics?

Who manages the Sys Intg and continued improvement?

Who manages the requirement, sets the backlog priority, and ensures the user gets what they asked for?
ISSUE 1: WHERE DOES AGILE ACTUALLY “FIT” IN THE COST WBS / MODEL?

ERP Based DBM >>>>
(SAP, Oracle, etc …)
- Business Functions:
  - Finance
  - HR
  - Logistics

Non-ERP IT/Info/DBM System >>>>
(COTS + Custom)
- Planning System
- Databases
- Simulations

C4I Distributed Msn System >>>>
- Command Center
- DCGS (Intel)
- Infrastructure

Define Requirements
- Perceived Problem
- Problem Statement
- Measureable Business Outcome

Blue Print
Configure (OTB)
Develop RICE-FW Obj

Procure/ Configure COTS SW
Develop Custom SW
Integrate the System (HW + SW)
Data Migration
Deploy Sustain

Procure/ Configure COTS SW
Build Mil Function SW
Integrate the System (HW + SW)
Data Migration
Deploy Sustain

DOES AGILE EXTEND BEYOND THE SW PORTION OF THE PRIME SYSTEM? HOW FAR?
## Issues

### ISSUE 2: HOW IS THE AGILE PORTION SCALED, MEASURED AND ESTIMATED (LIFE CYCLE COST; ESTIMATE AT COMPLETE)? – BEST PRACTICES:

<table>
<thead>
<tr>
<th>Measuring the SW Effort</th>
<th>Character of the Delivered SW Product</th>
<th>Sustain the Delivered SW Product</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td><strong>Productivity</strong></td>
<td><strong>Road Map</strong></td>
</tr>
<tr>
<td>COUNT “SHALLS”</td>
<td>HOURS/SLOC</td>
<td>(End Item Oriented)</td>
</tr>
<tr>
<td>SLOC/ESLOC</td>
<td>HOURS/ITEM</td>
<td>SW END ITEM (CPEI)</td>
</tr>
<tr>
<td>CONFIGURATIONS</td>
<td>HOURS/OBJECT</td>
<td>CONFIGURED ERP OBJECTS</td>
</tr>
<tr>
<td>RICE-FW OBJECTS</td>
<td>HOURS/ITEM</td>
<td>DEVELOPED OBJECTS</td>
</tr>
<tr>
<td>FUNCTION POINTS</td>
<td>HOURS/ITEM</td>
<td></td>
</tr>
<tr>
<td>USER STORIES</td>
<td><strong>Velocity</strong></td>
<td>MAINTENANCE RELEASE</td>
</tr>
<tr>
<td>STORY POINTS</td>
<td>HOURS/SP</td>
<td>- ADAPTIVE/CORRECTIVE/</td>
</tr>
<tr>
<td></td>
<td><strong>Burn Down Rate</strong></td>
<td>PREVENTIVE</td>
</tr>
<tr>
<td></td>
<td>SP/SPRINT</td>
<td>- NEW CAPABILITY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>UPGRADE COTS/ERP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAINTENANCE RELEASE</td>
</tr>
</tbody>
</table>
## Issues

### ISSUE 3: COLLECTING DATA AND MANAGING DATA TO INFORM FUTURE ESTIMATES.

**UPDATES IN-WORK FOR KEY REPORTING FORMATS; DATA TO BE STORED IN COST ANALYSIS DATA ENTERPRISE CADE**

- **CSDR PLAN (FORM 2794):** Commodity specific WBS (Mil-Std-881C, Appendix K); Specified reporting levels/frequency
- **FINANCIAL (FORM 1921):** Apply commodity specific WBS; Includes O&S; Includes Metadata
- **SOFTWARE (SRDR):** Adds sizing and configuration for ERP applications
- **TECH DATA (FORM 1921-T):** Data Center HW; Infrastructure
- **PROGRAM OFFICE (CARD):** Tabular data; Formatted templates
ISSUE 4: WHO MANAGES THE REQUIREMENT, SETS PRIORITIES AND ENSURES CAPABILITY IS DELIVERED AS REQUIRED AND WHEN REQUIRED TO COMPLETE THE SYSTEM?

**General Approach >>>>**
- Define Req’s
- Roadmap
- Rec/NR

**Build SW (Code & Unit Test)**
(Release Plan, Velocity, Backlog)

**Integrate the System**
(HW + SW)

**Deploy**
**Sustain**

**Gov’t*:**
- Define
- Prioritize
- Influence Scrum?

**Contractor:**
- Schedule Backlog
- Execute Sprint
- Manage Velocity
- Meet Gov’t needs (e.g., how much is enough per Increment?)

**Shared:**
- Contractor Support
- Government Lead
- 3rd Party Hosting/Cloud

*The government owns the requirement.
BACK-UPS
# Agile Terms Table from GAO*

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
<th>Also called</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Roadmap</td>
<td>High level view of the features the project will set out to accomplish along with the expected business value.</td>
<td>Project Vision</td>
</tr>
<tr>
<td>Release Plan</td>
<td>Schedule for developing working software that identifies the expected number of sprints and features that will be included in a release.</td>
<td>High level Requirements</td>
</tr>
<tr>
<td>Epics</td>
<td>High level capabilities.</td>
<td>Capabilities</td>
</tr>
<tr>
<td>Features</td>
<td>Next level below an epic which represents items of specific business value. Some features may need several stories to be complete.</td>
<td></td>
</tr>
<tr>
<td>User Story</td>
<td>Small chunk of software that identifies business value and success criteria that can be completed within a sprint timebox. A user story defines the work to be done to satisfy a feature.</td>
<td></td>
</tr>
<tr>
<td>Story Points</td>
<td>Assessed value of effort for an epic, feature, or user story based on team consensus.</td>
<td></td>
</tr>
<tr>
<td>Sprints</td>
<td>Short-term, timeboxed effort for delivering an agreed upon number of story points.</td>
<td>Iterations, increments</td>
</tr>
<tr>
<td>Product Backlog</td>
<td>List of prioritized user stories identified as Must Haves, Should Haves, Could Haves and Nice to Haves</td>
<td>Requirements backlog, feature list</td>
</tr>
<tr>
<td>Burn down Chart</td>
<td>Burn down charts represent completed user stories and reflect the rate of progress over time. Can be compared to estimated number of stories to be completed during each sprint for variance analysis.</td>
<td>Burn up chart</td>
</tr>
<tr>
<td>Retrospective</td>
<td>Final review of what was accomplished during a sprint and documentation of lessons learned (Agile team and customers/stakeholders attend)</td>
<td></td>
</tr>
<tr>
<td>Velocity</td>
<td>The rate of progress accomplished by the team during a sprint (measures number of story points delivered per sprint to better estimate future work). Velocity reflects a team’s cadence and will vary among teams.</td>
<td>Cadence</td>
</tr>
</tbody>
</table>

*Ref:  GAO Scheduling Assessment Guide (DRAFT), May 2012; Agile Appendix (Sept 2015); 10 Best Scheduling Practices.
Additional Definitions

• Velocity is a “team” measurement – not the individual
• Iteration Duration / Completed Total SP = Velocity
• Iterations needed = Total SP / Velocity
• Don’t change the duration and use the same result
• Velocity Measure provides a way to translate a Size into a duration
  – Every estimate starts with a Size estimate (Lines of Code, Function Points, Use Case Points, Ideal Days, Story Points, Hot Dogs in a bucket!)
• Size / Velocity = Duration
• Every estimation process requires a relationship between a volume measure (Size) and productivity – how much size can be done over time
• Velocity can be sued as a TEAM productivity measure.
More Definitions

Agile Concepts and Terms

- **User Stories**: A description of a user-oriented capability containing just enough information so that developers can have a reasonable understanding of how they would need to implement.

- **Complexity Points**: A relative measure of complexity that enables the team to quantify the scope or effort for an issue/artifact (e.g. story).

- **Sprint / Iteration / Release**: *Sprint* - Fixed time-box in which development occurs (usually 2 - 4 weeks); *Iteration* - Collection of sprints that result in prototype software demoed/tested with users; *Release* - Multiple iterations that fulfill a major subset of requirements constituting an operational product released to the users.

- **Velocity**: Performance / productivity measure that indicates progress toward capability delivery (i.e., Complexity Points completed per sprint).

- **Project / Sprint Backlog**: A prioritized database that summarizes the issues/artifacts yet to be completed for a sprint or the entire project.

- **Burndown**: The concept, often shown as a graph over time, of working off or “earning” Complexity Points toward iteration or delivery completion.