Performing Software Feasibility Analysis on Major Defense Acquisition Programs

Mr. James Thompson
Director, Major Program Support
ODDR&E/Systems Engineering

Mr. Christopher Miller
SAIC

Ms. Angela Lungu
QSM

13th Annual NDIA Systems Engineering Conference
San Diego, CA | October 27, 2010
**Mission Context**

**Director, Systems Engineering**
Steve Welby

---

**Systems Analysis**

**“Weapon Systems Acquisition Reform Act of 2009”**

S.454-10; d.(1): The development and tracking of detailed measurable performance criteria as part of the systems engineering master plans...

S.454-10; d.(3): A system for storing and tracking information relating to the achievement of the performance criteria and objectives specified...

S.454-12; SEC. 103.b.(4): Evaluating the utility of performance metrics used to measure the cost, schedule, and performance of [MDAPS], and making such recommendations …to improve such metrics.

---

**Mission Assurance**

**Software Performance Measurement and Analysis**

- DAPS 4.5 Software
- Data Collection based on Best Practices
- Parametric analysis used to assess program feasibility and establish benchmarks

---

**Major Program Support**

James Thompson

- Program Support Reviews
- Systems Engineering Plans
- Program Technical Auditing
- OIPT/DAB Support
- DAES Database Analysis and Support
- Performance Measurement
- Systemic Root Cause Analysis

---

13th Annual NDIA SE Conf
Oct 2010 Page-2
**Initial Metrics Data Call**

- **Metric**
  - Sizing (for each build, broken down by new/modified/reused, in SLOC)

- **Numerical Context (for each build)**
  1. Peak staffing
  2. Effort hours
  3. Duration (start and end dates, both planned and actual)
  4. Software reliability target (Mean Time to Defect, MTTD) or actual defects discovered
Initial Metrics Data Call

• **Descriptive Context (Background)**
  - System type (business, scientific, real time (e.g., avionics))
  - Program briefs that explain the software effort and any discussion of functionality included in each build or release (and which builds are customer releases vs. engineer builds), as well as Software Development Plan (explains their metrics collection plan).
  - Any metrics from previously completed builds/releases
  - Listing of the key PSR event dates (e.g., initial review dates, final brief, etc.)

• **Expected Sources**
  - CARD and Software Resources Data Report (SRDR) appendixes (2630-1 or 2630-2, 2630-3)
  - Software Development Plan (SDP)
  - Contract Deliverables (CDRLs) that contain monthly software performance data
  - Integrated Master Schedule (IMS) if software is shown
  - Integrated Master Plan (IMP)
40 Software Program Analyses Completed/Ongoing (2008-2010)

- **MDD (Materiel Solution Analysis)**
  - CBA
  - ICD
  - CBA
  - ICD

- **Technology Development**
  - PDR
  - CBA
  - ICD
  - ACS
  - EHR
  - GCV
  - KC-X
  - MLP
  - AIAMD
  - SSC
  - AIAMD (Initial CARD)
  - AIAMD (After De-Scope)
  - AOC WS
  - ARH-70A
  - CH-53K
  - DDG 1000
  - LCS
  - SIAP

- **Engineering and Manufacturing Development**
  - CDR
  - BAMS-Com
  - BAMS-MCS
  - BAMS-MR
  - GPS III
  - AEHF
  - Apache Block 3
  - BCS
  - EA-18G
  - E-2D
  - ER/MP
  - Excalibur
  - FCS
  - ISSPAN
  - JSF
  - MPS
  - P-8A
  - VA Class (MS III)

- **Production and Deployment**
  - CPD
  - C-130 AMP
  - C-5 RERP
  - C-5 AMP
  - Global Hawk

- **O&S (Operations and Support)**
  - FRP
  - DR

---

*13th Annual NDIA SE Conf*

10/2010 Page-5

*DISTRIBUTION STATEMENT A -- Cleared for public release by OSR on 19 October 2010 -- SR case number #11-S-0105 applies.*
## Existing Lifecycle Metric Sources

### Currently Leveraging Existing Metrics and Data Sources

### Anticipated Software-Related Actions By Program Phase

<table>
<thead>
<tr>
<th>Notes for reading this matrix: Software-related documents are in bold text. Specific metrics that should be included in the source documents are indented and italicized.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Program briefs explaining program software effort</th>
<th>Pre-MS A</th>
<th>Pre-MS B</th>
<th>Pre-MS C</th>
<th>Pre-ERP</th>
<th>Post-ERP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPE-approved Contract CSDR Plan (SRDRs and standard software metrics DIDs, as listed in CDRD)</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>CARD (with Initial Government Report SRDR, DD 2630-1) (Initial CARD due earlier of 180 days prior to OIPT or 60 days prior to RFP) (Final CARD due earlier of 45 days prior to OIPT or 60 days prior to RFP release) (DTM 09-027 now requires at MS A)</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Estimated/Planned integrated Master Plan listing key software/program events</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>DD 2630-3: Initial Government Report SRDR (estimate) (due same time as draft CARD to OSG, see above) (usually included in CARD)</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Estimated sizing for each build, broken down by new/modified/reused, in row, logical Source Lines of Code (SLLOC), function points, implementation units, requirements, or other standard sizing unit</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Estimated peak staffing by 'increment'</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Estimated Start/End dates by 'increment'</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Estimated software reliability threshold (MTTD)</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>System type (e.g., avionic, engineering)</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>DD 2630-2: Initial Developer Report SRDR (Required at start and completion of each 'increment')</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Planned sizing for each build, broken down by new/modified/reused, in row, logical Source Lines of Code (SLLOC), function points, implementation units, requirements, or other standard sizing unit</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Monthly software metrics CDRs (actuals) [see CAPE-approved contract CSDR plan]</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

**Actual sizing (and re-planned sizing, if changed since submission of DD 2630-2 or since last review) for each build, broken down by new/modified/reused, in row, logical Source Lines of Code (SLLOC), function points, implementation units, requirements, or other standard sizing unit**

- Actual peak staffing by 'increment'
- Actual effort by 'increment'
- Actual start/end dates by 'increment'
- Actual software reliability (MTTD)
- Actual software defects discovered/opened and closed defects, by 'increment' and by category (e.g., P1, P2, P3)

**DD 2630-3: Final Developer Report SRDR (60 days after completion/delivery) (for each 'increment')**

- Final/actual sizing (and re-planned sizing, if changed since last review) for each build, broken down by new/modified/reused, in row, logical Source Lines of Code (SLLOC), function points, implementation units, requirements, or other standard sizing unit
- Final/actual peak staffing by 'increment'
- Final/actual effort by 'increment'
- Final/actual start/end dates by 'increment'
- Final/actual software reliability (MTTD)
Sample Metrics Collected, Normalized, and Modeled

Program Data as Reported

- Metrics are captured as reported by the Program (as Program Artifacts)
  - Identify internal inconsistencies within Program metrics
  - Identify data gaps, and omissions
  - Data validation is necessary to conduct analysis

Normalized & Modeled Data

- Metrics are normalized to enable parametric modeling and benchmark analyses
  - Normalization provides ability use parametric models to assess feasibility
  - Software development effort assessed based on probability of success

Historical Software Performance Data

- Data compiled into historical repository to support benchmark analyses
  - Normalized data allows for benchmarking
  - Unified data set provides ability to assess software performance across portfolios of programs
Analysis Across the Acquisition Lifecycle

Software Metrics and Parametric Modeling and Analysis

C/S/Q/P/R – cost, schedule, quality, performance, & risk

PLANNING

EXECUTION

SUSTAINMENT

SW C/S/Q/P/R Ballpark Estimate

SW C/S/Q/P/R Feasibility

SW C/S/Q/P/R Performance

DoD History (Context & Data)

MDAPS / MAIS Data from Commercial Sources (Benchmarks)

PSTLs

Sw Team

Domain Knowledge

Estimation & Data Collection Plan

Actual/Plan Data Collection Requirements

MDAPS 4.5.2

PSR

PSR

PSR

MDD

Technology Development

A

B

C

ICD

CDD

CPD

FRP

O&S

CBA

PDR

CDR

Materiel Solution Analysis

Engineering and Manufacturing Development

Production and Deployment

Technology Development

Children Development

Materiel Solution

Analysis

PDR

CDR

PSTLs

DAPS 4.5.2

DAPS 4.5.2

DAPS 4.5.2
Software Analysis and Insight

• **MS A: Ballpark Estimate / Feasibility analysis**
  – Is the program’s plan to estimate the software adequate? Is the Acquisition strategy adequately accounting for the software development aspects?
  – What information (granularity) is missing or inadequate at this point in the program? Has the program identified the software metrics/data reporting requirements for the TD & EMD?

• **MS B: Independent Estimate / Plan Feasibility Analysis**
  – Are the software planning artifacts mature enough to support an estimate? What is the probability of meeting the delivery date?
  – How does their software plan compare to similar programs WRT size, complexity, schedule, staffing, & effort/cost?
  – Has the program adequately incorporated data collection & metrics reporting into the RFP and contract deliverables to support OSD program performance assessments?

• **MS C: Benchmark / Software Reliability Assessment**
  – Based on the performance data to date, what is the probability the program will deliver on time and on schedule?
  – What is the software defect density? And how does it compare to other similar programs? Does the data indicate a software quality issue?

• **Nunn-McCurdy/Special Emphasis:**
  – Is the Program’s performance typical? (i.e., was the breach due to inadequate funding/planning or is the breach due to poor performance)?
### Example A/Pre-MS B: “Trade Space”

Interrelationships among size, effort, staffing, duration, and productivity allow decision-makers to see the impact of existing program constraints.

#### Scenario Comparison (80% Assurance)

<table>
<thead>
<tr>
<th>Scenario Assumptions</th>
<th>ESLOC</th>
<th>Cost</th>
<th>Schedule</th>
<th>PI</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Plan</td>
<td>--</td>
<td>$25M</td>
<td>60 mo</td>
<td>--</td>
<td>Program allocated $25M for software, 60 months schedule is not software driven.</td>
</tr>
<tr>
<td>1. Optimized Solution</td>
<td>538K</td>
<td>$76M</td>
<td>86 mo</td>
<td>12.1</td>
<td>Historical industry average; assumes no ESLOC growth; cost overrun 300%; schedule adds 2.1 yrs</td>
</tr>
<tr>
<td>2. Fixed Cost</td>
<td>538K</td>
<td>$25M</td>
<td>114 mo</td>
<td>12.1</td>
<td>Constrained to $25.2M budget; schedule runs 4.5 yrs late</td>
</tr>
<tr>
<td>3. Fixed Schedule</td>
<td>538K</td>
<td>$370M</td>
<td>60 mo</td>
<td>12.1</td>
<td>Constrained to 5-yr schedule; cost is 14.7 times greater than total budgeted</td>
</tr>
<tr>
<td>4. Typical Program Size Growth</td>
<td>700K</td>
<td>$105M</td>
<td>97 mo</td>
<td>12.3</td>
<td>Size growth (80% industry projects typically grow 30% from PDR to delivery); slightly improved productivity index assumed; cost over 420% of budget; schedule takes 3 yrs longer</td>
</tr>
<tr>
<td>5. Reduced Functionality</td>
<td>216K</td>
<td>$25M</td>
<td>58 mo</td>
<td>12.1</td>
<td>Limited functionality/size with budget and schedule constrained</td>
</tr>
<tr>
<td>6. Increased</td>
<td>538K</td>
<td>$25.2M</td>
<td>60 mo</td>
<td>16.0</td>
<td>Increased PI (2 standard deviations higher than</td>
</tr>
<tr>
<td>7. Increased Productivity/Size</td>
<td>700K</td>
<td>$39M</td>
<td>62 mo</td>
<td>16.0</td>
<td>Increased PI (only 2.2% of industry has achieved that PI)</td>
</tr>
</tbody>
</table>

**XXX** = Value constrained (held constant) in scenario run

**PI** = Productivity Index, to include environmental factors for efficiency

**ESLOC** = Effective Logical Source Lines of Code

---

Program Office received trade space analysis

Enabled the program office to select initial planning options in the feasible trade space
Example A/Pre-MS B: Feasibility

- Feasibility analysis of the proposed Government estimate/plan indicated a 7% probability of meeting schedule.
- Excursion analysis showed Program Office feasible options.
- Program Office “de-scoped” and revised plan was assessed near 50% probability of meeting schedule; reducing overall cost & schedule risk to the program.

Excursions (What-if Analysis)

- We performed several excursions (what-if scenarios) to assess whether corrective action can be taken to increase probability of success.
- Each excursion is compared to the planned schedule and budget as well as the Independent Estimate.

<table>
<thead>
<tr>
<th>What-if Scenario</th>
<th>Primary Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excursion 1: Increase Budget</td>
<td>Meet Functionality and Schedule</td>
</tr>
<tr>
<td>Excursion 2: Reduce Functionality</td>
<td>Meet Budget and Schedule</td>
</tr>
<tr>
<td>Excursion 3: Defed Release 2 and Patriot</td>
<td>Meet Schedule and Budget</td>
</tr>
<tr>
<td>Excursion 4: Reduce Quality</td>
<td>Meet Schedule</td>
</tr>
<tr>
<td>Excursion 5: Extend Schedule</td>
<td>Meet Functionality, Budget and Quality</td>
</tr>
<tr>
<td>Excursion 6: Combination</td>
<td>Compromise of the Above Constraints</td>
</tr>
</tbody>
</table>

Excursion 2 Summary: Reduce Functionality

- Objective: Reduce functionality to reach plan release date of Aug 2013.
- Result: 40% of release 2 and 59% of System 1 could be delivered on scheduled date. Integration would require additional 8 months.
- Pros: Deliver code to integration on planned release date.
- Cons: Deliver less functionality, slightly worse quality.
Example MS B: Plan Feasibility

Risk areas identified based on statistical distance from historical program performance.

Scatter plot shows feasibility of planned builds compared to other similar AT&L programs.

Although consistent with AT&L projects, potential risk due to large size and increased defects, impacting reliability and, to lesser degree, schedule (fixing instead of coding).
Example MS C: Software Reliability Modeling

**Reconstruct current/adjusted plan using actual reported metrics**

**Compare and quantify performance to date with similar programs**

**Forecast if acceptable software reliability will be achieved by release date compared to similar programs**

**Using reported defects, calibrate model**
Continuous monthly monitoring provides detailed insight into projected reliability, highlighting any unusual backlog or defect discovery, down to subcomponent level.
Software Observations

• Summary findings from our parametric analyses of software intensive programs at OSD AT&L:
  - Lack of software reliability requirement
  - Missing core metrics needed for monthly tracking / forecasting
  - Inadequate program level estimation and probability assessment
  - Too much schedule compression (high staffing and risk of poor quality)
Path Forward

• **Leveraging existing metrics and data sources**
  – From many potential metrics; initial data call is limited to 5 fundamental metrics which should naturally exist based on existing acquisition policy and guidance.
  – Normalized and validated metrics provide a basis to model the software development effort and provide insight into overall software development feasibility

• **Extending and maturing metric-based oversight**
  – **Time Sensitive**
    – Measures collected during program interactions (e.g., PSRs)
    – Metrics collected to support specific decisions (e.g., Program’s ‘readiness to proceed’)
  – **Milestone Driven**
    – Metrics collected based on a Milestone or Technical Review
    – Metrics provide a static or ‘snapshot’ of program as of a specific date/time
  – **Periodic Data Collection**
    – Metrics collected to show trends or ‘movement’
    – Frequency
      – Monthly (e.g., defect modeling)
      – Quarterly (e.g., requirements stability)
      – Annually (e.g., inputs to congressional report)
Summary of Software Data, Analysis, & Lifecycle Decision Support

Program Artifacts
- CARD
- ConOps
- SRDR-1
- Sys Spec
- SDP
- SRDR-2
- Monthly Status
- Test Plan
- SRDR-3

SW Data/Metrics
- Domain (Context)
  - Ballpark Size, Constraints
  - Schedule, Budget, etc.
- Planned:
  - Size
  - Schedule
  - Staffing/Effort
  - SW Reliability
- Actual:
  - Size
  - Schedule
  - Staffing/Effort
  - Defects
  - Test Points

Parametric Analysis Objectives
- Feasible Trade Space?
- Alternatives?
- Realistic plan?
  - Assumptions
    - Productivity
    - Amount of Reuse
    - Existence of SW Reliability Rqmt

Program health check:
- Actual vs. Plan
- Forecast to Complete
- SW Reliability Modeling
For Additional Information

Jim Thompson
ODDR&E/Systems Engineering
(703) 602.0851 | James.Thompson@osd.mil

Chris Miller
SAIC
(703) 412.3689 | christopher.miller.ctr@osd.mil

Angela Lungu
QSM
(703) 749.3826 | angela.lungu.ctr@osd.mil
Systems Engineering: Critical to Program Success

Innovation, Speed, and Agility

http://www.acq.osd.mil/se