Software and IT-CAST

August 21-23, 2018

Lockheed Martin Global Vision Center
2121 Crystal Drive #100
Arlington, VA 22202
# Table of Contents

<table>
<thead>
<tr>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>1</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td>Overview</td>
<td>3</td>
</tr>
<tr>
<td>Agenda</td>
<td>4</td>
</tr>
<tr>
<td>Keynote Biography</td>
<td>7</td>
</tr>
<tr>
<td>Abstracts</td>
<td>10</td>
</tr>
<tr>
<td>Keynote Address</td>
<td>17</td>
</tr>
</tbody>
</table>

## Presentations

2. **Agile Software Cost Factors Case Study**, Blaze Smallwood (Booze-Allen-Hamilton) 71
3. **Army Software Maintenance Cost Estimating Results**, Cheryl Jones (Army ARDEC), James Doswell & Jenna Myers (ODASA-CE) 89
4. **Agile to DevOPS and Its Impact On Estimation and Measurement**, David Seaver (NSA) 126
5. **Automation In Practice: Assessment of An Agile Database**, Brittany Grissom, Cassie Robbins & Lyle Patashnick (NGA CAPE) 144
6. **Impact of Scope Change on Software Growth**, Dr. Jon Brown & Gail Flynn (NSWC-DD) 157
7. **Cost and Risk analysis of Managing Modernization Projects with Cloud and Open Source Considerations**, Dan Galorath (SEER Galorath) 178
8. **Total Ownership Cost of Cybersecurity in A Cloud Based IT System**, Richard Mabe (Price Systems, LLC) 244
11. **Agile Estimation**, John Sautter & Sarah Nichols (Northrop Grumman) 325
12. **How Bad Are We Estimating Software Growth**, Brittany Grissom (NGA CAPE) 354
13. **Using An Activity-Based Costing Approach to Estimate The Cost of Moving To Cloud**, Kevin Kenney (NGA) Pending Approval
14. **Amazon One-On-One: Cost Optimization on AWS** 391
15. **Boeing One-On-One** Pending Approval
16. **Microsoft One-On-One** Pending Approval
OVERVIEW

The Naval Center for Cost Analysis (NCCA) and the National Geospatial-Intelligence Agency Corporate Assessment and Program Evaluation (NGA CAPE) present the Software and Information Technology Cost Analysis Solutions Team (Software and IT-CAST) meeting from August 22-24, 2017 at the Lockheed Martin Global Vision Center in Crystal City, Virginia. This meeting is organized with the support of Lockheed Martin, and DOD cost agencies.

The Software and IT-CAST meeting is a venue to build coalitions with government and industry, to exchange cost data, share lessons learned, and establish best practices concerning software and information technology cost estimation. Topics include

- Software and Information Technology Cost Estimation
- Software Cost Data Collection and Analysis Best Practices
- Project Cost and Schedule Growth
- Measurements for Agile Software Development
- Measurements for Software Maintenance
- Measurements for Cloud Computing and Cyber Security

The program includes presentations, workshops, and contractor one-on-one discussions. Presentations and workshops are open to all attendees. Contractor one-on-one discussions are restricted to federal employees who have registered.

COMMITTEE

General Chair:
Vjosa Dreshaj (NGA CAPE)
Haset Gebre-Mariam (NCCA)
Lyle Patashnick (NGA CAPE)
Corey Boone (NCCA)

Venue Co-Chair:
Gregory Niemann (Lockheed Martin)

Portal Design Co-Chair:
Don Clarke (NCCA)

ATTENDANCE

General sessions are open to all attendees.
Contractor discussions are restricted to federal government employees who have registered.
<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Speaker(s)</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0800 – 0830</td>
<td>Registration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0830 – 0840</td>
<td>Opening Remarks</td>
<td>Wendy Kunc (Director, NCCA)</td>
<td>Auditorium</td>
</tr>
<tr>
<td>0840 – 0910</td>
<td>Keynote (Unable to attend due to a scheduling conflict)</td>
<td>Edwin Wilson (Deputy Assistant Secretary of Defense, Cyber Policy)</td>
<td>Auditorium</td>
</tr>
<tr>
<td>1000 – 1015</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1015 – 1045</td>
<td>Agile Software Cost Factors Case Study</td>
<td>Blaze Smallwood (Booz Allen)</td>
<td>Auditorium</td>
</tr>
<tr>
<td>1050 – 1120</td>
<td>Army Software Maintenance Cost Estimating Results</td>
<td>James Doswell (ODASA-CE)</td>
<td>Auditorium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cheryl Jones (U.S. Army ARDEC)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Jenna Meyers (ODASA-CE)</td>
<td></td>
</tr>
<tr>
<td>1125 – 1210</td>
<td>Agile to DEVOPS and Its Impact on Estimation and Measurement</td>
<td>David Seaver (NSA)</td>
<td>Auditorium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lyle Patashnick, Brittany Grissom, and Cassandra Robbins (NGA CAPE)</td>
<td></td>
</tr>
<tr>
<td>1210 – 1300</td>
<td>Lunch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1300 – 1330</td>
<td>Impact of Scope Changes on Software Growth</td>
<td>Jonathan Brown (NSWC-DD)</td>
<td>Auditorium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gail Flynn (NSWC-DD)</td>
<td></td>
</tr>
<tr>
<td>1335 – 1405</td>
<td>Cost &amp; Risk Analysis of Managing Modernization Projects With Cloud and Open Source Considerations</td>
<td>Dan Galorath (Galorath)</td>
<td>Auditorium</td>
</tr>
<tr>
<td>1410 – 1440</td>
<td>Total Ownership Cost of Cybersecurity in a Cloud Based IT System</td>
<td>Richard Mabe (PRICE)</td>
<td>Auditorium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>David Cass (IBM)</td>
<td></td>
</tr>
<tr>
<td>1445 – 1630</td>
<td>Amazon One-on-One</td>
<td>Benjamin Kleintank, Roman Rusal, Marc Johnson, and Morteza Zijerdi (Amazon)</td>
<td>Auditorium</td>
</tr>
</tbody>
</table>
# Software and IT-CAST Agenda

**Wednesday, August 22, 2018**

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
<th>Speaker/Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0800 – 0830</td>
<td>Registration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0830 – 0840</td>
<td>Opening Remarks</td>
<td>John Scali (Director, NGA CAPE)</td>
<td>Auditorium</td>
</tr>
<tr>
<td>0840 – 0910</td>
<td>Keynote: How Do We Estimate Agile Development?</td>
<td>Dr. Jeffrey Boleng (Special Assistant for Software Acquisition to the Under Secretary of Defense (A&amp;S))</td>
<td>Auditorium</td>
</tr>
<tr>
<td>0915 – 0945</td>
<td>Keynote: Agile Acquisition at DHS</td>
<td>Chip Fulghum, Deputy Under Secretary for Management, DHS</td>
<td>Auditorium</td>
</tr>
<tr>
<td>0945 – 1000</td>
<td>Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110 – 1140</td>
<td>Agile Estimation – Northrop Grumman Q&amp;A</td>
<td>John Sautter and Sarah Nichols (Northrop Grumman)</td>
<td>Auditorium</td>
</tr>
<tr>
<td>1140 – 1245</td>
<td>Lunch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1245 - 1345</td>
<td>Panel: Software Cost Data Sharing with FFRDC and Support Contractors</td>
<td>Robert M. Flowe (OUSD (A&amp;S)/ARA)); David W. Lyons (OSD CAPE); David J. Nicholls (IDA); Dr. David Zubrow (SEMAI) Moderator: Lyle Patashnick</td>
<td>Auditorium</td>
</tr>
<tr>
<td>1350– 1420</td>
<td>How &quot;Bad&quot; Are We Estimating Software? A Software Growth Study</td>
<td>Brittany Grissom (NGA CAPE)</td>
<td>Auditorium</td>
</tr>
<tr>
<td>1425 – 1455</td>
<td>Using an Activity-Based Costing approach to estimate the cost of &quot;Moving to the Cloud&quot;</td>
<td>Kevin Kenney (NGA)</td>
<td>Auditorium</td>
</tr>
</tbody>
</table>

**Contractor Discussions (Restricted)**

<table>
<thead>
<tr>
<th>Time</th>
<th>Contractor Discussions (Restricted)</th>
<th>Organizer/Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500 – 1630</td>
<td>Boeing one-on-One</td>
<td>Shawn M McCullough (Boeing) Rod Burr (Boeing)</td>
<td>Auditorium</td>
</tr>
</tbody>
</table>
## Software and IT-CAST

### Agenda

**Thursday, August 23, 2018**

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity</th>
<th>Participants</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0800 – 0830</td>
<td>Registration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0830 – 1000</td>
<td>Microsoft One-on-One</td>
<td>Ben Griffith, John-Eric Dyer,</td>
<td>Auditorium</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monica DeZulueta (Microsoft)</td>
<td></td>
</tr>
</tbody>
</table>
Dr. Jeff Boleng

Special Assistant for Software Acquisition to the Under Secretary of Defense for Acquisition and Sustainment

Dr. Jeff Boleng is the Special Assistant for Software Acquisition to the Under Secretary of Defense for Acquisition and Sustainment (USD(A&S)) where he serves as a key member of the Under Secretary’s executive leadership team, providing strategic focus and overall policy guidance on all matters of defense software acquisition. In this role, he leads the formulation of the Department’s software acquisition strategy, advises Department leadership on latest best practices in commercial software development, supports the enterprise to build a team of top-tier software engineers, and works to develop modern software skills in the acquisition workforce.

Jeff has a breadth of experience across the Department of Defense (DOD) and the private sector. Prior to joining DOD, he served as the chief technology officer (acting) and deputy chief technology officer at Carnegie Mellon University Software Engineering Institute. Prior to that, he served more than 21 years in the United States Air Force as a cyberspace operations officer and software engineer. In his final assignment with the Air Force, Jeff served as the deputy department head, Department of Computer Science at the United States Air Force Academy.

Jeff is a senior member of both the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE), and he holds PhD and MS degrees in Mathematical and Computer Sciences from the Colorado School of Mines and a BS in Computer Science from the U.S. Air Force Academy.
Chip Fulghum

*Deputy Under Secretary for Management, Department of Homeland Security*

Chip Fulghum became the Deputy Under Secretary for Management in May 2015. Along with the Under Secretary for Management, Mr. Fulghum oversees all aspects of the Department’s management programs, including financial, human capital, information technology, procurement, security, and asset management. He also provides support and guidance to the Department’s acquisition oversight process and represents DHS in a number of management-related interagency committees.

Mr. Fulghum has twice served as the Senior Official Performing the Duties of the Deputy Secretary. He has also served as the Acting Under Secretary for Management.

Mr. Fulghum joined the Department in October 2012 as the Budget Director within the Office of the Chief Financial Officer (CFO). He later served as the Department’s CFO from September 2014 – January 2017. As CFO, he had stewardship of internal controls to reduce waste, fraud, and abuse across DHS.

Prior to joining the Department, Mr. Fulghum served as a U.S. Air Force officer for 28 years, rising to the rank of Colonel. He is a graduate of the Air War College, the Air Command and Staff College, Professional Military Comptroller School, and Squadron Officer School. He holds a Masters of Business Administration from Golden Gate University in San Francisco, California, and a Bachelor of Arts Degree in History from The Citadel in Charleston, South Carolina.
0840 - 0910: **Keynote Address**  
B. Edwin Wilson, Deputy Assistant Secretary of Defense for Cyber Policy

0915 - 1000: **A Cost Effective Strategy to IT Security**  
Beau Woods, Deputy Director of the Cyber Statecraft Initiative, Atlantic Council

**Abstract**  
Over the next five years, every organization will experience at least one cybersecurity failure; yet global spending on cybersecurity people, products, and technology is expected to exceed $1B. While we may not know all the right things to secure our IT environments, we know many of the failure modes and can avoid them. Smart procurement strategies can increase security effectiveness at decreased or steady cost. This talk will outline both tried and true, and new and novel, approaches to procure secure.

1015 - 1045: **Agile Software Cost Factors Case Study**  
Blaze Smallwood, Booz Allen Hamilton

**Abstract**  
The lack of data on government agile software development programs has made estimating costs for new agile development programs challenging. This presentation seeks to address this challenge through a small study of several completed DoD agile projects with cost, schedule, and performance data. It will examine several relevant metrics, including cost per story point, cost per requirement, scope growth rates, impacts of team size changes on velocity and productivity, and various others.

1050 - 1120: **Army Software Maintenance Cost Estimating Results**  
James Doswell and Jenna Meyers, ODASA-CE

**Abstract**  
The Army has completed an initial analysis of software sustainment cost and performance data collected from ~250 Weapons, C4ISR, and ERP systems. The analysis addresses primary resource distributions and cost estimating relationships across multiple functional domains, and establishes a foundation for efficient resource allocation decisions across the Army systems portfolio, and projected policy and process changes. The results, including the detailed statistical analysis, will be made available for use by participants.
Tuesday, August 21, 2018

1125 - 1210: Agile to DEVOPS and Its Impact on Estimation and Measurement

David Seaver, National Security Agency; Lyle Patashnick, Brittany Grissom, and Cassandra Robbins, National Geospatial-Intelligence Agency

Abstract

NSA and NGA are collaborating on the analysis and measurement of three large projects that have implemented Agile or DEVOPS at scale. 1. This presentation will provide an overview of the approaches we are piloting for the analysis and measurement of the projects; 2. Will provide a snapshot of the analysis that has been completed by the end of July 2018; 3. Initial early lessons learned; 4. Discussion of using an Agile thought process for independent cost estimates; 5. Discussion automation in the measurement and analysis process.

1300 - 1330: Impact of Scope Change on Software Growth

Jonathan Brown and Gail Flynn, NSWCDD

Abstract

The SEI DoD Software Factbook summarizes MDAP/MAIS SRDR data for DoD programs. The mean value reported for ESLOC growth is 106%. While accurate, the SEI’s and other similar analyses capture total software growth, including the impact of scope changes. This paper introduces “Pure Software Growth” which differentiates planned scope changes from traditional software growth. Several programs are analyzed from this perspective to show the difference between pure and total growth and the unexpected impact this could have on estimates.

1335 - 1405: Cost & Risk Analysis of Managing Modernization Projects with Cloud and Open Source Considerations

Dan Galorath, President of Galorath, Inc.

Abstract

Software modernization projects are becoming increasingly critical as mainframe hardware ages out, people retire, and skillsets are lost or forgotten. Concurrently, available resources and budgets are usually tight. This leaves management with the challenges of dealing with the sometimes fragile software iron triangle (scope, resources, schedule and quality.) To further complicate matters, agencies are increasing their cloud consumption as part of modernization. This is fraught with challenges such as getting a viable estimate when cloud vendors make it appear as easy as filling in a web form. Even the definition of cloud must be further quantified to discriminate between public, private, and community clouds while the definitions of Infrastructure as a Service (IaaS) and Platform as a service continue to blur.
1410 - 1440: Total Ownership Cost of Cybersecurity in a Cloud Based IT System

Richard Mabe, Senior Solutions Consultant, PRICE Systems
David A. Cass, VP/CISO & Managing Partner, Global Cloud Security Services, IBM

Abstract
Converging technology trends in XaaS have profound effects on how organizations are evaluating decisions regarding XaaS outsourcing and hybrid deployments as more business functions move to the cloud. Most organizations have a security skills gap that XaaS and moving to the cloud can help solve giving choices on insourcing or outsourcing cybersecurity. This paper explores how XaaS impacts the TCO of cybersecurity and also deliver guidance on the estimating the cost of the DFARS cyber policy to defense programs

1445 - 1630: Amazon One on One Discussion (Restricted)

Benjamin E. Kleintank, Roman B. Rusal, Marc Johnson, Morteza Zijerdi, Amazon
**Wednesday, August 22, 2018**

**0840 - 0910: Keynote: How Do We Estimate Agile Development?**

**Dr. Jeff Boleng, Special Assistant for Software Acquisition, USD(A&S)**

**Abstract**

How much will the software in my new system cost? How long will it take? These are decades old questions that continually defy prediction, yet they are becoming increasingly important in our software defined world. Perhaps we are asking the wrong question, or at least asking the question in the wrong way. Our goal in the DoD is to procure and field capability that provides a decisive combat advantage for our warfighters. More and more of that capability is software defined, but we are not buying software, we are buying warfighting capability. Asking the questions of cost and time in a different way may provide an alternative to effective estimation. In this talk I will explore alternative ways that may help measure and answer these questions.

**0915 - 0945: Keynote: Agile Acquisition at DHS**

**Chip Fulghum, Deputy Under Secretary for Management, DHS**

**Abstract**

More than 50% of DHS major acquisition programs are delivering a solution that is predominantly IT based to fulfill a mission need. IT by its nature is ever evolving and can change before you even have an opportunity to begin developing. As such we are challenged with finding better and faster ways to deliver these solutions while still leveraging the acquisition oversight policy and processes that govern these critical investments. To address these challenges, we are working on a number of initiatives to promote Agile/modular development in accordance with OMB direction. In February 2016 we stood up an Agile Acquisition Working Group to pilot acquisition process improvements that facilitate increased customer value, accountability and oversight, faster time-to-market at reduced cost and risk. Under the Agile premise, balancing a fixed cost and schedule and a varying capability we still must deliver a baseline of capability before the solution becomes obsolete? This engagement will address our experience with these pilots, our challenges and successes and what we’ve done to shorten the time it takes to develop key acquisition documents to include the life cycle cost estimate.

**1000 - 1030: A Probabilistic Method for Predicting Software Code Growth**

**Eric Sommer, US Air Force Space and Missile Command**

**Abstract**

Software estimating is challenging. SMC’s approach has evolved over time to tackle this challenge. Originally based on Mike Ross’s 2011 DSLOC Estimate Growth Model, we’ve updated our model to include more recent SRDR data and an improved methodology (Orthogonal Distance Regression). Discussions will focus on non-linear relationships between size and growth, unique growth for new, modified, and unmodified DSLOC, as well as correlation between DSLOC types and future efforts to include space flight software data.

Neil Albert, CEO, NFA Consulting

Abstract
The Work Breakdown Structure (WBS) is critical to managing, planning, estimating and assessing performance of any project. The WBS development is critical to ensure that all team members, industry and government communicate and coordinate their activities. This presentation discusses the newly revised Information Systems/Defense Business Systems WBS for Investment and Sustainment. The presentation provides the updated concepts and definitions for the use and application of the WBS.

1110 - 1140: Agile Estimation – Northrop Grumman Q&A

John Sautter and Sarah Nichols, Northrop Grumman Technology Services

Abstract
Northrop Grumman Corporation (NGC) will provide answers to a series of previously received questions from the SW IT CAST leadership regarding agile definitions, performance, and estimation. The questions were provided to NGC from the SW and IT CAST IPT organizers at last's year's one-on-one session. Some additional questions have been added this year. NGC will address these questions in an open non-proprietary fashion bringing in historical experiences into the presentation as appropriate.

1245- 1345: Panel: Software Cost Data Sharing with FFRDC and Support Contractors

Robert M. Flowe, Senior Analyst, OSD Studies & FFRDC management, OUSD(A&S)/ARA/OS&FM; David W. Lyons, Operations Research Analyst, OSD CAPE; David J. Nicholls, Director of IDA’s Cost Analysis and Research Division; Dr. David Zubrow, Team Leader for the Software Engineering Measurement and Analysis Initiative

Lyle Patashnick, Panel Moderator, NGA CAPE

Abstract
Section 235 of the National Defense Authorization Act (NDAA) for Fiscal Year 2017, Public Law 114-328 (section 235) (enclosure 1) authorizes the Department of Defense (DoD) to conduct a three-year "Pilot Program on Disclosure of Certain Sensitive Information to Federally Funded Research and Development Centers (FFRDCs)." The intent is to allow FFRDCs to gain access to data needed to fulfill their missions while providing assurances against disclosure or misuse. Invitations went out to FFRDCs in December of 2017. In the intervening months there has been activity to comply with the requirements of the program and to obtain access to data. This panel brings together representatives of the different stakeholders in the process to discuss their perspectives and experiences either providing or gaining access to the data covered by the pilot program.
1350 - 1420: How "Bad" Are We Estimating Software? A Study Software Growth

Brittany Grissom, NGA CAPE

Abstract
DoD's SRDR repository has been the source of many analyses across the community. Drawing on these analyses, NGA developed its own set of software growth factors from the data for both development effort hours and ESLOC. NGA also examined the statistical significance of the data and highlighted areas of concern. The research explores both parametric and non-parametric test and regressions, evaluation the results of both before recommending the final growth factors. This presentation summarizes the methods used and results of the research.

1425 - 1455: Using an Activity-Based Costing Approach to Estimate the Cost of "Moving to the Cloud"

Kevin Kenney, NGA

Abstract
Many agencies across the IC & DOD are transitioning IT services & processes from the data center to "the Cloud". One of the big questions surrounding these transitions is: How much is it going to cost? As part of an ongoing pilot project at NGA, we have adopted an Activity-Based Costing approach as one method of capturing cloud costs. Presentation will highlight the approach, metrics, major cost drivers, and lessons-learned to date.

1500 - 1630: Boeing One on One Discussion (Restricted)

Shawn M McCullough, Boeing Company
Rod L Burr, Boeing Company
Thursday, August 23, 2018

0830 - 1000: Microsoft on One Discussion (Restricted)

Ben Griffith, Microsoft
How do we estimate Agile Software Development?

Dr. Jeff Boleng
Special Assistant for Software Acquisition, OSD(A&S)
Focus on buying capability

We want to buy capability (warfighting capability), not software

Some (much or most) may be realized by software, but we shouldn’t care

However, we do live in a software defined world!
What do we need to know

Is it worth doing?

Was it worth doing?

When do you want capability? As soon as possible!

Not: How much will it cost? or How long will it take?
Old Way

Fix Scope, estimate cost and schedule

We can only control scope

Congress and bidders control budget/cost

Nobody seems to control schedule

We would love to control quality (and security)

An Alternative

Fix schedule and cost

Require frequent deliveries

Evaluate delivered scope/capability and quality via metrics

Start small with minimal risk

Attack highest ROI MVP first

Determine if value delivered justifies continuing

## Jigsaw

### Cost, Aircraft and Personnel

<table>
<thead>
<tr>
<th></th>
<th>MVP</th>
<th>Current Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investment</td>
<td>$1.5M</td>
<td>Ongoing</td>
</tr>
<tr>
<td>Offload Efficiency</td>
<td>2.4%</td>
<td>3.6%</td>
</tr>
<tr>
<td>Savings</td>
<td>$214K/day ($6.4M/mo)</td>
<td>$428K/day ($12.8M/mo)</td>
</tr>
<tr>
<td>Aircraft</td>
<td>1 less tanker/day</td>
<td>2 less tankers/day As many as 4 tails can be allocated to other AORs</td>
</tr>
<tr>
<td>Personnel – Air</td>
<td>5 personnel @ 609th deployed</td>
<td>3 personnel @ 609th deployed</td>
</tr>
<tr>
<td>Coordination Team</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staffing (ACTS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMC Crew Teams</td>
<td>1 tanker/day → 5 less active-duty</td>
<td>2 less tankers/day → 10 less active-duty air crews</td>
</tr>
<tr>
<td></td>
<td>crews</td>
<td></td>
</tr>
</tbody>
</table>
Types of Metrics

Process metrics
  Schedule and cost of development

Quality metrics
  Defects found, etc.

Product metrics
  Context dependent → the hardest to determine
Metrics

Not SLOC! Please stop now...

Errors = (more code)$^2$

SLOC → Penalizes higher level languages and reuse

Cost per defect → Penalizes Quality

Did I mention reuse!

What does automatically generated code imply?

Recommended Reading: “Errors and Omissions in Software Historical Data: Separating Fact from Fiction” Capers Jones
http://www.namcook.com/Articles/MeasurementErrors2008.doc

Google translate “classic” ≈ 500k SLOC

Google translate using TensorFlow ≈ 500 SLOC

Source: https://events.rice.edu/#!view/event/event_id/442

Image source: https://commons.wikimedia.org/wiki/File:Emc2.svg
## Example 1

**Application in Assembly Language**

- **10K SLOC (10 months)**
- **100 pages of documentation (5 months)**
- **15 Months effort = 666 LOC/month**
- **$10K/staff month or $150K**
- **$15/SLOC**

**Application in C++**

- **1K SLOC (1 month)**
- **75 pages of documentation (4 months)**
- **5 Months effort = 200 LOC/month**
- **$10K/staff month or $50K**
- **$50/SLOC**

### Example 1

<table>
<thead>
<tr>
<th>Application in Assembly Language</th>
<th>Application in C++</th>
</tr>
</thead>
<tbody>
<tr>
<td>10K SLOC (10 months)</td>
<td>1K SLOC (1 month)</td>
</tr>
<tr>
<td>100 pages of documentation (5 months)</td>
<td>75 pages of documentation (4 months)</td>
</tr>
<tr>
<td>15 Months effort = 666 LOC/month</td>
<td>5 Months effort = 200 LOC/month</td>
</tr>
<tr>
<td>$10K/staff month or $150K</td>
<td>$10K/staff month or $50K</td>
</tr>
<tr>
<td>$15/SLOC (50 function points)</td>
<td>$50/SLOC (50 function points)</td>
</tr>
<tr>
<td>3.33 function points/month</td>
<td>10 function points/month</td>
</tr>
<tr>
<td>$3k/function point</td>
<td>$1k/function point</td>
</tr>
</tbody>
</table>

Example 2

Poor Quality ($2.5k/week employee)
15 hours writing test cases
10 hours running tests
15 hours fixing 10 bugs
Cost per defect = $250

Good Quality (same cost per week)
15 hours writing test cases
10 hours running tests
5 hours fixing 1 bug
Cost per defect = $1,875

Source: http://www.namcook.com/Articles/MeasurementErrors2008.doc
# Agile Metrics

<table>
<thead>
<tr>
<th>#</th>
<th>Metric</th>
<th>COTS\textsuperscript{ii} apps</th>
<th>Custom-\textsuperscript{iii}ized SW</th>
<th>COTS HW/OS\textsuperscript{iv}</th>
<th>Real-time HW/SW\textsuperscript{v}</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>&lt;1 mo</td>
<td>&lt;3 mo</td>
<td>&lt;6 mo</td>
<td>&lt;1 yr</td>
<td>3-5 yrs</td>
</tr>
<tr>
<td>2</td>
<td>Time to field high priority fcn (spec $\rightarrow$ ops) or fix newly found security hole (find $\rightarrow$ ops)\textsuperscript{vi}</td>
<td>N/A</td>
<td>&lt;1 mo</td>
<td>&lt;3 mo</td>
<td>&lt;3 mo</td>
<td>1-5 yrs</td>
</tr>
<tr>
<td>3</td>
<td>Time from code committed to code in use</td>
<td>&lt;1 wk</td>
<td>&lt;1 hr</td>
<td>&lt;1 da</td>
<td>&lt;1 mo</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\textsuperscript{vii}</td>
<td>N/A</td>
<td>&lt;1 da</td>
<td>&lt;1 da</td>
<td>&lt;1 wk</td>
<td>2 yrs</td>
</tr>
<tr>
<td>5</td>
<td>Time required to restore service after outage</td>
<td>&lt;1 hr</td>
<td>&lt;6 hr</td>
<td>&lt;1 day</td>
<td>N/A</td>
<td>?</td>
</tr>
<tr>
<td>6</td>
<td>Automated test coverage of specs / code</td>
<td>N/A</td>
<td>&gt;90%</td>
<td>&gt;90%</td>
<td>100%</td>
<td>?</td>
</tr>
<tr>
<td>7</td>
<td>Number of bugs caught in testing vs field use</td>
<td>N/A</td>
<td>&gt;75%</td>
<td>&gt;75%</td>
<td>&gt;90%</td>
<td>?</td>
</tr>
<tr>
<td>8</td>
<td>Change failure rate (rollback deployed code)</td>
<td>&lt;1%</td>
<td>&lt;5%</td>
<td>&lt;10%</td>
<td>&lt;1%</td>
<td>?</td>
</tr>
<tr>
<td>9</td>
<td>% code available to DoD for inspection/rebuild</td>
<td>N/A</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>10</td>
<td>Complexity metrics</td>
<td>#/type of specs</td>
<td>#/type of platforms</td>
<td># programmers</td>
<td>Partial/manual tracking</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Development plan/environment metrics</td>
<td>/skill level of teams</td>
<td>#/type deployments</td>
<td>/type deployments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>“Nunn-McCurdy” threshold (for any metric)</td>
<td>1.1X</td>
<td>1.25X</td>
<td>1.5X</td>
<td>1.25X Total $</td>
<td></td>
</tr>
</tbody>
</table>

[https://media.defense.gov/2018/Jul/10/2001940937/-1/-1/0/DIB_METRICS_FORSOFTWARE_DEVELOPMENT_V0.9_2018.07.10.PDF](https://media.defense.gov/2018/Jul/10/2001940937/-1/-1/0/DIB_METRICS_FORSOFTWARE_DEVELOPMENT_V0.9_2018.07.10.PDF)
Agile Metrics

My top four + n (unapologetically borrowed from Kessel Run)

- Release Cadence (process metric)
- Time from code commit to release candidate (process metric)
- Change fail percentage (quality metric)
- Time to roll back (recover from release error) (process and quality metric)

Product Metrics (n number of them)

Please don’t forget product metrics, **which are the most important**
Product Metrics

Reduction in time to complete task

Increase in accuracy for computation or task

Reduction in resources (personnel, cost, etc.) to complete task

Increase in safety

Increase in security
An alternative

1 agile team - 6-10 people, 1 PM/product owner, 1 designer, 4-8 developers

$2-$4M per year

Include funding to cultivate the codebase and refactor, continually

Short sprints (2-4 weeks) and short epochs (3-6 months)

Evaluate capability delivered at the end of every sprint and epoch, use metrics

Learn → we’ll learn to be predictive

Examples: ARCI, Aegis, NSA, JIDO, AOC Pathfinder, F-22
Summary

Take smaller bites
Deliver capability as early as possible
Continually evaluate quality, product, progress, and cost
Learn as we go
Code “fly-offs” should be encouraged
Architecture is still key!
A Cost-Effective Strategy for IT Security

This document is a size-minimized, partially modified version of the presentation given on August 21, 2018 at the IT-CAST meeting hosted by NGA and Navy.
IT Security Failures have become Mission Failures
In 2017, we conducted the world’s first clinical hacking SIMULATIONS (https://iatc.me/cybermed). In 3 separate scenarios, 3 patients died from hacked medical devices.
In 2017, WannaCry impacted patient care in about 30% of UK hospitals, for hours to weeks.
In 2018, TSMC, one of the world's largest technology suppliers, shut down production because of malware, which will cost them up to $1.5B USD and may delay semiconductor shipments.
Global stockpiles of Hepatitis vaccine dipped when Merck was forced to shut down production because of the NotPetya virus in 2017.
The shipping giant Maersk had to shut down ports and logistics due to a virus in 2017. FEDEX and UPS were also affected.
Examining the Adversaries
The following slide is a framework for understanding relative capabilities and intent/willingness of adversaries (and accidents). During the session, we walked through this graphic with examples.
Increasingly willing

Nation State
- IR
- IL
- RU
- NK
- US
- SK
- UK
- CN
- AU

Professional
- Exploit Dev
- Coders
- Criminals
- DDoS
- Blackhat SEO

Accident
- Operators
- Social Bots
- Hosting
- Ransomware
- Botnets

Ideological
- Hacktivists
- Terrorists

Increasingly capable

Increasingly hard to distinguish Accident from Adversary
Nation State
- IR
- RU
- US
- UK
- FR
- IL
- NK
- SK
- CN
- AU

Professional
- Exploit Dev
- Coders
- Criminals
- DDoS
- Blackhat SEO

Accident
- Operators
- Social Bots
- Hosting
- Ransomware
- Botnets

Typical Defensive Level
- Hacktivists
- Terrorists

Capabilities
- Willingness
Nation State
● IR • IL
● RU • NK
● US • SK
● UK • CN
● FR

Professional
• Exploit Dev
• Coders
• Criminals
• DDoS
• Blackhat SEO
• Operators
• Social Bots
• Hosting
• Ransomware
• Botnets

Ideological
• Hacktivists
• Terrorists

Typical Defensive Level
• Known Good Defensive Practices

Willingness

Capabilities

Deterrence

Accident
IT Security Cost/Benefit
Forecasted Global Cybersecurity Spending, 2017-2021:

$1 Trillion
ONE HUNDRED PERCENT of FORTUNE 500 companies will be hacked over the same time period.
The Food Pyramid

Foods and drinks high in fat, sugar and salt

NOT every day

Fats, spreads and oils

In very small amounts

Meat, poultry, fish, eggs, beans and nuts

2 servings a day

Milk, yogurt and cheese

3 servings a day

Wholemeal cereals and breads, potatoes, pasta and rice

3-5 servings a day

Vegetables, salad and fruit

5-7 servings a day

For adults, teenagers and children aged five and over

Maximum once or twice a week
Defensible Infrastructure

- Secure by Design
- Secure Baseline Configurations
- Secure Deployment Guidance
- Operating System and Software Support Lifetimes

VS

- Software Updateable
- Software Ingredients or Components List
- Evidence Capture and Logging
- ...
Defensible Infrastructure

Operational Excellence

• Coordinated Vulnerability Disclosure
  • DevSecOps
  • Visible Ops
• Vulnerability Management
  • Change Management
  • Egress Filtering
• Network Admission Control
  • ...

• Secure by Design
  • Secure Baseline Configurations
  • Secure Deployment Guidance
  • Operating System and Software Support Lifetimes
• Software Updateable
  • Software Ingredients or Components List
  • Evidence Capture and Logging
  • ...

Regular maintenance

Normal (secure) operations
Identifying the threats around you

- Penetration Testing
- Threat Intelligence
- Security Monitoring
- Threat Hunting
- ...
Defensible Infrastructure

Operational Excellence

Situational Awareness

Countermeasures

- Secure by Design
  - Secure Baseline Configurations
  - Secure Deployment Guidance
  - Operating System and Software Support Lifetimes

- Coordinated Vulnerability Disclosure
  - DevSecOps
  - Visible Ops

- Network Admission Control
  - ...)

- Penetration Testing
- Threat Intelligence
- Security Monitoring
- Threat Hunting
  - ...

- Securitvility Management
  - Change Management
  - Egress Filtering
  - ...

- Endpoint Security
- Active Defense
- Intrusion Prevention
- Anti-Everything
  - ...

- Countermeasures
  - Secure by Design
  - Secure Baseline Configurations
  - Secure Deployment Guidance
  - Operating System and Software Support Lifetimes

- Software Updateable
  - Software Ingredients or Components List
  - Evidence Capture and Logging
  - ...

- Specific defenses against specific threats

You might expect spending patterns would look like this.
Actual spending patterns tend to look like this. It’s upside down. Don’t do this.
Security Indicators in Procurement
Automotive 5-Star Cyber Safety Framework

All systems fail. What is your ready posture toward failure?

★ Safety by Design – Anticipate and avoid failure
★ 3rd Party Collaboration – Engage willing allies to avoid failure
★ Evidence Capture – Observe and learn from failure
★ Security Updates – Correct failure conditions once known
★ Segmentation & Isolation – Prevent cascading failure

Connections and Ongoing Collaborations

- Security Researchers
- Automotive Engineers
- Policy Makers
- Insurance Analysts
- Accident Investigators
- Standards Organizations
- Government Agencies

https://iamthecavalry.org/5star/
Secure by Design
(Built In vs Bolt On)

As an example, fire escapes tend to be much less expensive, more effective, and easier to maintain when BUILT IN during construction as opposed to BOLTED ON later.
Collaboration with Security Researchers

Vendors that offer a welcome mat tend to identify and remediate issues sooner and at lower cost and risk than those who threaten vulnerability reporters.
DoD’s Vulnerability Disclosure Policy Results

- Total valid reports resolved: 2,837
- Participating hackers: 645+
- High or critical severity vulnerabilities: 100+

Hackers from 50 countries including: India, Great Britain, Pakistan, Philippines, Egypt, Russia, France, Australia and Canada
Software Security Updatability

Increasing Agility & Decreasing Cost

- Hardware Replacement
- Connected Updates
- Remote Updates
- Automatic Updates
We look for nutrition labels and bills of materials on physical goods. Increasingly, these same concepts are applicable – and available – for software.
Anything sold to the US Government must:

A. Provide a software component list
   Software Bill of Materials or Food Label
B. Disclose known vulnerabilities
C. Be software updateable
Anything sold to the US Government must:

A. Disclose known vulnerabilities
B. Be software updateable
C. Avoid hard-coded credentials
D. Have a coordinated disclosure policy
Coordinated Vulnerability Disclosure

- US Department of Commerce, NTIA Template

- ISO/IEC 29147 Standard for Vulnerability Disclosure
  https://www.iso.org/standard/45170.html

- ISO/IEC 30111 Standard for Vulnerability Handling Processes
  https://www.iso.org/standard/53231.html
## Procurement Guidance

<table>
<thead>
<tr>
<th>4. System Information:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- List of 3rd Party Software</td>
</tr>
<tr>
<td>- List of Accounts</td>
</tr>
<tr>
<td>- List of Network Ports</td>
</tr>
<tr>
<td>- List of firewall rules (if applicable)</td>
</tr>
<tr>
<td>- Documentation of Security Capabilities/Configurations for System Hardening</td>
</tr>
<tr>
<td>- Scanning Requirements</td>
</tr>
</tbody>
</table>

Provides more granular information as to how the system is setup and managed within the Mayo Clinic environment.

<table>
<thead>
<tr>
<th>5. Vulnerability Assessment, including:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Testing Results</td>
</tr>
<tr>
<td>- Remediation Tracking</td>
</tr>
</tbody>
</table>

Provides an in-depth vulnerability assessment, outstanding vulnerabilities and appropriate remediation plans and timelines to resolve the issues. This provides Mayo Clinic with appropriate information on risks that may be introduced into the patient care environment and allows for collaborative mitigation strategies to be detailed.

<table>
<thead>
<tr>
<th>6. Mayo Clinic Information Security Schedule</th>
</tr>
</thead>
</table>

Provides advanced copy of Mayo Clinic's Information Security Schedule that Supply Chain Management will negotiate as part of the purchase contract or vendor agreements.

Provide vendor documentation (i.e. Bill of Materials) for the bulleted items. Template provided.

Complete a vulnerability assessment as detailed in the Vendor Assessment Book (pdf). Once testing is completed, complete the VA Statement of Methodology and document findings and remediation plans in a report. Example VA Statement of Methodology (pdf) and Vulnerability Assessment Template report provided.

Ensure appropriate vendor internal staff receives Mayo's Information Security Schedule for review.

Perform review and prepare any proposed redline items.

Provide vendor contact to the Mayo proponent for the redlined ISS negotiation.

Deliverable 4 - System Information

Deliverable 6 - Information Security

---

https://www.mayoclinic.org/documents/medical-device-vendor-instructions/doc-20389647
Software Component Transparency (Software Bill of Materials)
https://www.ntia.doc.gov/SoftwareTransparency

Coordinated Security Vulnerability Disclosure

Device Upgradeability and Patching
https://www.ntia.doc.gov/IoTSecurity

President’s Commission Report on Enhancing National Cybersecurity
https://www.nist.gov/cybercommission
<table>
<thead>
<tr>
<th>UL 2900-1</th>
<th>Software Cybersecurity for Network-Connectable Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>2900-2-1</td>
<td>Healthcare Systems</td>
</tr>
<tr>
<td>2900-2-2</td>
<td>Industrial Control Systems</td>
</tr>
<tr>
<td>2900-2-3</td>
<td>Life Safety &amp; Physical Security</td>
</tr>
</tbody>
</table>

https://industries.ul.com/cybersecurity
AGILE COST FACTORS CASE STUDY

Blaze Smallwood
Software and IT-CAST 2018

AUGUST 2018
PURPOSE

• Explore cost, schedule, performance metrics for a small collection of DoD agile software projects
• Determine if any trends exist and any rules of thumb can be derived
• Highlight major takeaways
CASE STUDY PROJECTS

- Completed DoD Automated Information System (AIS) software development/integration projects

<table>
<thead>
<tr>
<th>Project / Marker</th>
<th>ACAT</th>
<th>Performer (GOV/KTR)</th>
<th>ALM Tool Used</th>
<th>Cost ($M) ***</th>
<th>Schedule (Months)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>III</td>
<td>KTR</td>
<td>Forge</td>
<td>$5.6</td>
<td>20</td>
</tr>
<tr>
<td>B</td>
<td>III</td>
<td>KTR</td>
<td>Jira</td>
<td>$4.0</td>
<td>21</td>
</tr>
<tr>
<td>C</td>
<td>I</td>
<td>GOV</td>
<td>Jira</td>
<td>$21.2</td>
<td>18</td>
</tr>
<tr>
<td>D</td>
<td>III</td>
<td>KTR</td>
<td>TFS</td>
<td>$10.2</td>
<td>19</td>
</tr>
<tr>
<td>E</td>
<td>N/A**</td>
<td>KTR</td>
<td>Jira</td>
<td>$1.3</td>
<td>14</td>
</tr>
<tr>
<td>F*</td>
<td>N/A**</td>
<td>GOV</td>
<td>Jira</td>
<td>$7.4</td>
<td>11</td>
</tr>
</tbody>
</table>

* Project had no specific end date; schedule indicates # of months data was collected
** Pre-Acquisition risk reduction projects
*** Full cost of the software development/integration project; excludes non-PMP costs, like PMO costs

Acronyms: ACAT = Acquisition Category; ALM = Application Lifecycle Management; TFS = Team Foundation Server
### PROJECT DATA SUMMARY

- Projects had varying levels of data available

<table>
<thead>
<tr>
<th>Metric</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost per Point</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Hours per Point</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cost per Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Hours per Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cost Variance</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Schedule Variance</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Scope Variance</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Team Composition</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Buffering Percentages</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Most projects defined a planned point as 8 developer hours. Actuals indicate more cost/effort per point due to overhead and points taking more effort than expected to finish.
Smaller projects (<$10M) tended to spend less resources per requirement – their requirements were generally less complex and defined at a more granular level.
### DELTAS AT PROJECT END

<table>
<thead>
<tr>
<th>Project</th>
<th>Cost (% Delta – Plan minus Actual)</th>
<th>Schedule (% Delta – Plan minus Actual)</th>
<th>Scope (% of Planned Scope Not Completed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project A</td>
<td>0%</td>
<td>(11%)</td>
<td>8%</td>
</tr>
<tr>
<td>Project B</td>
<td>1%</td>
<td>0%</td>
<td>30%</td>
</tr>
<tr>
<td>Project C</td>
<td>27%</td>
<td>0%</td>
<td>41%</td>
</tr>
<tr>
<td>Project D</td>
<td>(2%)</td>
<td>(17%)</td>
<td>32%</td>
</tr>
<tr>
<td>Project E</td>
<td>1%</td>
<td>21%</td>
<td>2%</td>
</tr>
</tbody>
</table>

Negative numbers indicate cost/schedule overruns.

All projects deferred at least some scope to later releases.

- Project D had major performance issues, while Project E had atypically good performance; Projects A-C were fairly typical.
- Most agile projects treat scope as variable with mostly fixed cost and schedule.
  - Most finish at planned cost and schedule, but defer some scope to future releases, likely impacting future cost/schedule.
Projected to-complete cost/schedule overruns of 20-40% seem to be typical for agile projects.

Projected overruns caused by a combination of performance issues and prioritizing scope from agile activities:
- In-process testing
- User evaluations
- Requirements discovery
BUFFERING

- % of completed non-feature scope (bugs, usability, etc.) of completed feature scope (defined by functional requirements)

- Expected part of agile software development process, and should be included in estimates
  - Major component of expected cost/schedule overruns

- All projects in this case study between 15% and 40%

- Good rule of thumb: 20-30%
Team Composition

- Two larger projects collected data needed for this metric
- Qualitative observation on other, smaller projects: they had more developers as % of total (less overhead), likely ~70-80%

Average: 
~60/40 split between development/integration and overhead (PM, business support, functional SMEs, etc.)
Team Size Growth and Productivity

- Project C more than doubled its team size, mainly in an attempt to get back on schedule after falling behind.
- This strategy failed when productivity decreased significantly as the team size grew.
AGENDA

INTRODUCTION

METRICS DETAILS

SUMMARY
TAKEAWAYS

• Monetizing points or requirement counts is difficult and entails large uncertainty ranges

• For most agile projects, scope is the variable
  - Most finish at planned cost and schedule, but defer some scope to future releases, possibly impacting future cost/schedule
  - Without scope deferral, our “normal” case study data points projected cost/schedule overruns at ~20-40% of original plan

• Good rule of thumb for buffering: Add 20-30% to requirements/feature-driven estimates for bugs, etc.

• Rules of thumb for team composition:
  - Project Cost > $10M: 60% development/integration; 40% overhead
  - Project Cost < $10M: 75% development/integration; 25% overhead
NEXT STEPS

• Further analyze existing data for other useful metrics
  - More detailed analysis of how team size changes impacts productivity
  - Correlation between cost/schedule/scope deltas
  - Metric correlation to high-level project aspects (size, performer, etc.)
  - EVM-like metrics

• Collect/organize additional data points
SUMMARY

• Agile projects can be planned and measured
• Data analysis can yield useful metrics for cost estimating
• As usual, more data collection and analysis is needed
THANK YOU

For more information, contact . . .

Blaze Smallwood
Lead Associate

Booz Allen Hamilton Inc.
Office 309.359.3160
Mobile 619.850.6123
smallwood_blaze@bah.com
Using Army Software Sustainment Cost Estimating Results
DASA-CE

Presented to
IT-CAST
August 21, 2018
SWM Initiative Objective and Strategy

Accurately estimate Army system software sustainment costs to:
- Effectively project and justify software and system life cycle costs
- Objectively evaluate Army system software sustainment execution costs
- Inform and optimize the allocation of available sustainment resources across the Army

Collect and evaluate SWS cost and technical data for all Army operational systems (Phase I and Phase II data call)

Generate and validate cost estimating relationships from Phase I and Phase II data collection

Implement systemic Army SWS data collection via the SRDR-M. Populate cost and technical data repository

Improve Army SWS policy, business, and technical requirements

Effective software sustainment cost estimation is the basis for Army system software life cycle cost management
Data Demographics

- **193 Programs**
- **1,036 Total Releases**
- **3,434 Licenses**
- **411K Data Fields**

- Largest DoD Software Sustainment database
- Total Dollar Value Captured: $3.1B
- Programs collected ranged from ACAT I to Non-Program of Records
Software Sustainment Data Evaluation

Availability
- Completeness of required data set
- Underlying SWS business and technical processes are well enough defined to produce objective data on a periodic and/or event driven basis
- IT systems and tools exist to enable systematic and timely data collection

Integrity
- Data are derivatives of actual SWS technical and management processes
- All data (measures) are explicitly defined - measurement contexts are known
- Cost data is directly correlated with the WBS defined output products and activities
- Data is consistent - methods exist to address system conflicts (normalization)

Usability
- Data is aligned with stakeholder decision information needs
- Data can be objectively characterized and interpreted
- Mapping and aggregation structures and methods exist to combine data
- Potential emerging information requirements have been considered
Data Quality Evaluation
Annual Cost Level

<table>
<thead>
<tr>
<th>Color</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Red indicates there is no planning or actual data reported</td>
<td>0</td>
</tr>
<tr>
<td>Y</td>
<td>Yellow indicates FTE or partial, actual data was reported</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>Green indicates that actual data was reported</td>
<td>2</td>
</tr>
</tbody>
</table>

### System Level Annual

<table>
<thead>
<tr>
<th>Initial System Overall</th>
<th>Detailed System Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating</td>
<td>WBS 2-8</td>
</tr>
<tr>
<td>R</td>
<td>25</td>
</tr>
<tr>
<td>Y</td>
<td>79</td>
</tr>
<tr>
<td>G</td>
<td>76</td>
</tr>
<tr>
<td>N/A</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>190</td>
</tr>
</tbody>
</table>

- Data was collected from 190 programs
  - 174 programs provided total system SWM costs (G, Y)
  - 16 programs could not provide even planned total cost

- A lot of programs could not articulate how much was spent for licenses or facilities, often because these are paid for by enterprise or overhead funds
## Data Quality Evaluation

### Capability Releases

<table>
<thead>
<tr>
<th>Initial Release Overall</th>
<th>Detailed Release Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating</td>
<td>CER Usability</td>
</tr>
<tr>
<td>R</td>
<td>270</td>
</tr>
<tr>
<td>Y</td>
<td>89</td>
</tr>
<tr>
<td>G</td>
<td>348</td>
</tr>
<tr>
<td>N/A</td>
<td>6</td>
</tr>
</tbody>
</table>

| Total | 713 | 713 | 713 | 713 | 713 | 713 | 713 | 713 | 713 | 713 |

- Data was collected from 713 capability releases
  - 437 releases had sufficient data to use in CER cost calculations (G, Y)
  - Size data was not always consistently tracked and generally was not mapped to resource (effort/cost) information
    - 532 releases tracked some sort of software change counts (defects, PTRs)
    - Many of the capability releases did not track the number of IAVAs addressed
    - Effort was often not tracked at the release level
    - Systems in different super-domains used different size measures
      - Software changes was the most commonly used size measure
Data Quality Evaluation
IAVA Only Releases

<table>
<thead>
<tr>
<th>Rating</th>
<th>CER Usability</th>
<th>SER Usability</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>87</td>
<td>30</td>
</tr>
<tr>
<td>Y</td>
<td>147</td>
<td>169</td>
</tr>
<tr>
<td>G</td>
<td>89</td>
<td>124</td>
</tr>
<tr>
<td>N/A</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>323</td>
<td>323</td>
</tr>
</tbody>
</table>

Release Level (IAVA Releases Only)

<table>
<thead>
<tr>
<th>Schedule (WBS-1)</th>
<th>Effort (WBS-1)</th>
<th>Size: Req’ts</th>
<th>Size: External Interfaces</th>
<th>Size: SLOC</th>
<th>Size: Non-SLOC</th>
<th>Size: SW Changes</th>
<th>IAVAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>27</td>
<td>68</td>
<td>116</td>
<td>99</td>
<td>0</td>
<td>103</td>
<td>70</td>
</tr>
<tr>
<td>0</td>
<td>170</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>312</td>
<td>126</td>
<td>59</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>31</td>
<td>253</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>196</td>
<td>197</td>
<td>216</td>
<td>321</td>
<td>189</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>323</td>
<td>323</td>
<td>323</td>
<td>323</td>
<td>323</td>
<td>323</td>
<td>323</td>
</tr>
</tbody>
</table>

- Many programs reported IAVA only releases which are releases that address known cybersecurity vulnerabilities
- Data was collected from 323 IAVA only releases
  - 236 releases have sufficient data to use in CER cost calculations (G, Y)
  - Programs sized IAVA releases by the count of IAVAs information assurance vulnerability alerts
## Super Domain Definitions

### Real-Time

Real-Time is the most constrained type of software. These are specific solutions limited by system characteristics such as memory size, performance, or battery life. These projects take the most time and effort due to constraints.

<table>
<thead>
<tr>
<th>Application Domains</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcode &amp; Firmware</td>
<td>Field Programmable Gate Arrays, Flight Control, Missile Control, Radar</td>
</tr>
<tr>
<td>Signal Processing</td>
<td>Attimeter, Network Operations, Signal Electronics, Tracking Sensors,</td>
</tr>
<tr>
<td>Vehicle Control/</td>
<td>Encryption, Radio Networks, Propulsion</td>
</tr>
<tr>
<td>Vehicle Payload</td>
<td></td>
</tr>
<tr>
<td>Other Real-Time</td>
<td></td>
</tr>
<tr>
<td>Embedded Command &amp;</td>
<td></td>
</tr>
<tr>
<td>Control Communications</td>
<td></td>
</tr>
</tbody>
</table>

### Engineering

Engineering software operates under less severe constraints than real-time software. This software may take real-time software outputs and further process them to provide human consumable information or automated control of devices. Or the software may perform transformation and aggregation / distribution of data.

<table>
<thead>
<tr>
<th>Application Domains</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Process Control</td>
<td>Operating Systems, Image processing, Simulation &amp; Modeling, Test</td>
</tr>
<tr>
<td>Control</td>
<td>Equipment, File Management, Artificial Intelligence, Manufacturing</td>
</tr>
<tr>
<td>Scientific and Simulation</td>
<td>Process Control</td>
</tr>
<tr>
<td>Test, Measurement</td>
<td></td>
</tr>
<tr>
<td>Diagnostic and</td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td></td>
</tr>
</tbody>
</table>

### Support

Support software assists with operator training and software testing. This software has few constraints.

<table>
<thead>
<tr>
<th>Application Domains</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Software</td>
<td>Computer Based Training, Compilers, Programming Aids, Code Generators,</td>
</tr>
<tr>
<td>Tools</td>
<td>Assemblers, Courseware, Test case generation, Linker/loaders, Code</td>
</tr>
<tr>
<td>Auditors</td>
<td></td>
</tr>
</tbody>
</table>

### AIS

Automated information system software provides information processing services to humans or software applications. These applications allow the designated authority to exercise control and have access to typical business / intelligence processes and other types of information access. These systems also includes software that facilitates the interface and control among multiple COTS / GOTS software applications.

<table>
<thead>
<tr>
<th>Application Domains</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission Planning</td>
<td>Scenario Generators, Target Planning, Enterprise Service Management</td>
</tr>
<tr>
<td>Custom AIS Software</td>
<td></td>
</tr>
<tr>
<td>Enterprise Service</td>
<td></td>
</tr>
<tr>
<td>Systems</td>
<td></td>
</tr>
<tr>
<td>Enterprise Information Systems</td>
<td></td>
</tr>
<tr>
<td>Systems</td>
<td></td>
</tr>
</tbody>
</table>

---

**UNCLASSIFIED**

Distribution Statement A: Approved for Public Release; Distribution is Unlimited
Distributions and Benchmarks
SWS Total Annual Cost Distributions
Annual Cost by Super Domain

RT - Annual Total

- Mean: $4,742,426
- Median: $2,409,532

ENG - Annual Total

- Mean: $5,982,711
- Median: $2,715,008

AIS - Annual Total

- Mean: $8,888,236
- Median: $6,582,205

SUP - Annual Total

- Mean: $4,717,051
- Median: $4,381,673

<table>
<thead>
<tr>
<th>Count</th>
<th>Min</th>
<th>1st Quartile</th>
<th>Median</th>
<th>3rd Quartile</th>
<th>Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>219</td>
<td>$16,047</td>
<td>$992,952</td>
<td>$2,409,532</td>
<td>$5,780,354</td>
<td>$36,278,114</td>
<td>$4,742,426</td>
</tr>
<tr>
<td>133</td>
<td>$35,841</td>
<td>$1,005,502</td>
<td>$2,715,008</td>
<td>$7,579,732</td>
<td>$40,141,088</td>
<td>$5,982,711</td>
</tr>
<tr>
<td>91</td>
<td>$168,409</td>
<td>$1,899,573</td>
<td>$6,582,205</td>
<td>$10,648,955</td>
<td>$51,797,592</td>
<td>$8,888,236</td>
</tr>
<tr>
<td>40</td>
<td>$38,203</td>
<td>$891,637</td>
<td>$4,381,673</td>
<td>$7,073,955</td>
<td>$15,319,248</td>
<td>$4,717,051</td>
</tr>
</tbody>
</table>
Number of SW Changes/Release can be used to size future releases when program specific data is unknown. The resulting size can be used with the associated cost benchmark or put into a CER.
### Cost per SW Change

#### Capability Releases

#### Cost per Software Change

<table>
<thead>
<tr>
<th>Super Domain</th>
<th>Count</th>
<th>Min</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>Max</th>
<th>Mean</th>
<th>Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT</td>
<td>82</td>
<td>$488</td>
<td>$15,019</td>
<td>$39,219</td>
<td>$129,365</td>
<td>$307,244</td>
<td>$74,082</td>
<td>$76,622</td>
</tr>
<tr>
<td>ENG</td>
<td>176</td>
<td>$354</td>
<td>$4,774</td>
<td>$9,885</td>
<td>$39,472</td>
<td>$343,344</td>
<td>$37,350</td>
<td>$58,891</td>
</tr>
<tr>
<td>AIS</td>
<td>78</td>
<td>$187</td>
<td>$5,871</td>
<td>$14,547</td>
<td>$26,412</td>
<td>$181,292</td>
<td>$22,763</td>
<td>$30,236</td>
</tr>
<tr>
<td>SUP</td>
<td>13</td>
<td>$2,106</td>
<td>$3,290</td>
<td>$12,603</td>
<td>$35,735</td>
<td>$116,673</td>
<td>$30,954</td>
<td>$41,022</td>
</tr>
</tbody>
</table>
Software Change Definition Variability

- Within WBS 1.0, the effort associated with software releases is captured. A software release can be sized using the count of the number of software changes.

- A software change describes a change where source code/script is altered whether it be added, deleted or modified. Respondents defined a software change as:
  - Enhancement
  - New Requirements – Change or clarification of a requirement that results in a source code modification
  - New Capability: Addition of a new capability
  - Improvement: Enhancement to an existing capability
  - Issues
  - “Bug” fix: defect
  - Change or clarification of a design that results in a source code modification
  - Change request: changes to the requirements and the corresponding implementation.
  - Defect report: Defects are changes to the software to make them meet the requirements.
  - Problem Change Reports
  - Modification requests

Since there was significant variability across the programs in the definition of a software change, a more in-depth analysis was conducted to understand the costs of different types of software changes.
## Unit Cost Grouping Levels: Hrs/SC

<table>
<thead>
<tr>
<th>Release Hrs per Software Change</th>
<th>1-VL (Count: 49)</th>
<th>2-L (Count: 46)</th>
<th>3-N (Count: 41)</th>
<th>4-H (Count: 48)</th>
<th>5-VH (Count: 39)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>24.1</td>
<td>55.4</td>
<td>110.3</td>
<td>228.8</td>
<td>581.4</td>
</tr>
<tr>
<td>Median</td>
<td>23.6</td>
<td>56.3</td>
<td>107.2</td>
<td>219.6</td>
<td>568.5</td>
</tr>
<tr>
<td>Min Value</td>
<td>2.0</td>
<td>41.0</td>
<td>70.5</td>
<td>162.2</td>
<td>286.8</td>
</tr>
<tr>
<td>Max Value</td>
<td>40.1</td>
<td>69.5</td>
<td>161.8</td>
<td>324.3</td>
<td>961.3</td>
</tr>
</tbody>
</table>
Category Analysis Exploration

- Maintenance Organization (17)
- Location of Maintenance Organization (11)
- Commodities (10)
- Super Domains (RT, ENG, SUP, AIS)
- Change types (Enhanced, Maintenance, Cybersecurity)
- Business models (Government, Contractor, Integrated)
- Maintenance Phase (MS-C LRP, MS-C FRP, O&S)/Time in Phase
- ACAT Level
- Number of Software variants
- Number of Platform variants
- Number of Users
- Number of Licenses
- Number of Inter-Services Partners
- Release/Total Cost

To determine which cost grouping (1-VL through 5-VH) a program will fall, a number of characteristics were examined for significance.
Unit Cost Level One-Category Criteria

- Each slide presents Unit Cost levels by a category criteria
- There are two tables:
  - Top table are the counts of each Release’s Unit Cost at a level
  - Bottom table are the percentages of the counts
- The bottom table is examined for a “percentage” or “adjacent sum of percentages” greater than or equal to 50% (green highlight)
- For example, **Business Model**:

Release Unit Cost Level count by **Business Model**

<table>
<thead>
<tr>
<th>Business Model</th>
<th>Count</th>
<th>1-VL</th>
<th>2-L</th>
<th>3-N</th>
<th>4-H</th>
<th>5-VH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Government</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrated</td>
<td>77</td>
<td>19</td>
<td>26</td>
<td>12</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>Contractor</td>
<td>142</td>
<td>25</td>
<td>18</td>
<td>31</td>
<td>35</td>
<td>33</td>
</tr>
</tbody>
</table>

Release Unit Cost Level count % by **Business Model**

<table>
<thead>
<tr>
<th>Business Model</th>
<th>Count</th>
<th>1-VL</th>
<th>2-L</th>
<th>3-N</th>
<th>4-H</th>
<th>5-VH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Government</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrated</td>
<td>77</td>
<td>24.7%</td>
<td>33.8%</td>
<td>15.6%</td>
<td>11.7%</td>
<td>14.3%</td>
</tr>
<tr>
<td>Contractor</td>
<td>142</td>
<td>17.6%</td>
<td>12.7%</td>
<td>21.8%</td>
<td>24.6%</td>
<td>23.2%</td>
</tr>
</tbody>
</table>
## ACAT & Inter-Services

### Release Unit Cost Level Count % by ACAT

<table>
<thead>
<tr>
<th>ACAT</th>
<th>Count</th>
<th>1-VL</th>
<th>2-L</th>
<th>3-N</th>
<th>4-H</th>
<th>5-VH</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACAT I</td>
<td>38</td>
<td>5.3%</td>
<td>15.8%</td>
<td>26.3%</td>
<td>18.4%</td>
<td>34.2%</td>
</tr>
<tr>
<td>ACAT II</td>
<td>41</td>
<td>31.7%</td>
<td>4.9%</td>
<td>9.8%</td>
<td>24.4%</td>
<td>29.3%</td>
</tr>
<tr>
<td>ACAT III</td>
<td>101</td>
<td>24.8%</td>
<td>31.7%</td>
<td>16.8%</td>
<td>13.9%</td>
<td>12.9%</td>
</tr>
<tr>
<td>Non PoR</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>100.0%</td>
<td></td>
</tr>
</tbody>
</table>

### Release Unit Cost Level Count % by Inter-Service

<table>
<thead>
<tr>
<th>Inter-Service</th>
<th>Count</th>
<th>1-VL</th>
<th>2-L</th>
<th>3-N</th>
<th>4-H</th>
<th>5-VH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Army Only</td>
<td>165</td>
<td>24.8%</td>
<td>23.0%</td>
<td>19.4%</td>
<td>18.2%</td>
<td>14.5%</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>9.1%</td>
<td>36.4%</td>
<td>9.1%</td>
<td>27.3%</td>
<td>18.2%</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td></td>
<td>42.9%</td>
<td>14.3%</td>
<td></td>
<td>42.9%</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>14.3%</td>
<td>14.3%</td>
<td>28.6%</td>
<td></td>
<td>42.9%</td>
</tr>
<tr>
<td>5</td>
<td>33</td>
<td>6.1%</td>
<td>6.1%</td>
<td>24.2%</td>
<td>27.3%</td>
<td>36.4%</td>
</tr>
</tbody>
</table>
Super Domain

Release Unit Cost Level Count % by Super Domain

<table>
<thead>
<tr>
<th>Super Domain</th>
<th>Count</th>
<th>1-VL</th>
<th>2-L</th>
<th>3-N</th>
<th>4-H</th>
<th>5-VH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Time</td>
<td>115</td>
<td>27.0%</td>
<td>28.7%</td>
<td>10.4%</td>
<td>18.3%</td>
<td>15.7%</td>
</tr>
<tr>
<td>Engineering</td>
<td>54</td>
<td>3.7%</td>
<td>13.0%</td>
<td>29.6%</td>
<td>27.8%</td>
<td>25.9%</td>
</tr>
<tr>
<td>AIS</td>
<td>49</td>
<td>18.4%</td>
<td>10.2%</td>
<td>34.7%</td>
<td>16.3%</td>
<td>20.4%</td>
</tr>
<tr>
<td>Support</td>
<td>6</td>
<td>50.0%</td>
<td></td>
<td></td>
<td>16.7%</td>
<td>33.3%</td>
</tr>
</tbody>
</table>

• Since the previous results were inconclusive, a more detailed analysis was conducted
  – Software changes were characterized based on contextual comments in the questionnaire and by Super Domain
Lessons Learned/Next Steps
## Software Sustainment Estimating Framework

<table>
<thead>
<tr>
<th>1.0 Software Change Product</th>
<th>5.0 System Facilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activities</strong></td>
<td>License Cost</td>
</tr>
<tr>
<td>IAVAs, SW Changes (defects/enhancements)</td>
<td><strong>Performing Org.</strong></td>
</tr>
<tr>
<td>Contractor</td>
<td><strong>Challenges</strong></td>
</tr>
<tr>
<td>Use of inconsistent size measures; effort not generally tracked by release</td>
<td>Facilities paid by various sources; inheriting hardware from other sources</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2.0 Project Management</th>
<th>6.0 Sustaining Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activities</strong></td>
<td>Help Desk, Delivery/Installation, Test Support</td>
</tr>
<tr>
<td>CM, Execution, Project/Engineering Leads</td>
<td><strong>Performing Org.</strong></td>
</tr>
<tr>
<td>Government/Contractor</td>
<td><strong>Challenges</strong></td>
</tr>
<tr>
<td>Roles/Responsibilities spread throughout WBS; contractor generally paid by overhead</td>
<td>Inconsistent/varying activities reported; category generally misunderstood</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.0 Software Licenses</th>
<th>7.0 Field Software Engineers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activities</strong></td>
<td>Field Maintenance, Installation, Troubleshooting</td>
</tr>
<tr>
<td>License Cost</td>
<td><strong>Performing Org.</strong></td>
</tr>
<tr>
<td>Government/Contractor/Outside Organization (enterprise licenses)</td>
<td><strong>Challenges</strong></td>
</tr>
<tr>
<td>Paid for by multiple sources; licenses generally underreported; not always tracked</td>
<td>Difficult to estimate required support; shared between multiple programs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4.0 Certification and Accreditation</th>
<th>8.0 Operational Management</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activities</strong></td>
<td>Enterprise Management, Business Management</td>
</tr>
<tr>
<td>DIACAP/RMF, STIGs</td>
<td><strong>Performing Org.</strong></td>
</tr>
<tr>
<td>Government/Outside Organization</td>
<td><strong>Challenges</strong></td>
</tr>
<tr>
<td>Differs between types of C&amp;A’s, Difficult to track prep vs certification vs fixes post certification</td>
<td>Generally treated as overhead, spread across programs</td>
</tr>
</tbody>
</table>
Conclusion & Next Steps

Importance of Data Collection

- Consistent and accurate technical/cost data allows for more meaningful CERs that are relevant to the changing environment of software sustainment
- Software sustainment data can be used to better inform design decisions and cost analysis
  - DASA-CE and the Army cost community are now able to develop cost products that use analogous program data and technical output to estimate software maintenance. This facilitates major milestone estimates, O&S cost targets, Operation Sustainment Reviews, and yearly POM reviews
  - Phase I dataset is hosted on CADE under “Library”

Next Steps

- Additional analysis of data, including:
  - Refined CERs/SERs by appropriate categories (application domain, organization, operating environment, etc.)
  - Cost of impacts of DIACAP vs RMF
  - Cost of Cybersecurity
  - Release rhythm analysis
- Systemic data collection
  - The Software Resources Data Reporting for Maintenance (SRDR-M*) closely aligns to the DASA-CE SWM WBS and data requirements
  - Moving forward, the SRDR-M will be utilized to collect SWM data from a large number of programs across the Army
  - Ongoing analysis will be performed as data is made available through the SRDR-M

*See [http://cade.osd.mil/policy/dids](http://cade.osd.mil/policy/dids) for more information
Contributors

James Judy
NISEC Division Chief
ODASA-CE
703-697-1612

Jenna Meyers
Senior Operations Research Analyst
ODASA-CE
703-697-1645

Cheryl Jones
Software Measurement Analyst
US Army ARDEC
973-724-2644

Brad Clark
Vice President
Software Metrics, Inc.
(703) 754-0115

Robert Charette
President
ITABHI Corporation
(540) 972-8150

James Doswell
Senior Operations Research Analyst
ODASA-CE
703-697-1572

John McGarry
Software Measurement and Analysis Lead
US Army ARDEC
973-724-2644
Backup
Army Software Maintenance Definition

For this effort, software maintenance is defined as:

- Software maintenance includes all software change activities and products associated with modifying a software system after EMD has completed and a software release has been provided to an external party
- The release is the primary SWM change product - a composite of one or more changes - it can be either a formal release or an engineering release
- SWM includes software enhancements and software corrections/adaptations
- SWM includes activities and change products funded by multiple funding sources
- Fixed and Variable costs accrued at both the system and organizational levels by both organic and contractor resources
- Software maintenance and software sustainment are considered to be synonymous
Data Collection Process

PEOs/SECs/SEDs identified all programs with software efforts for Phase II

DASA-CE met with program/system representative to explain data collection questionnaire and clarify requirements

System representative completed and submitted initial draft of questionnaire

DASA-CE team reviewed questionnaire, identified questions, and met with representative to discuss context and issues

System representative updated questionnaire based on DASA-CE findings

DASA-CE reviewed submission and continued to rework with system representative as necessary

Final data submission was accepted and evaluated for availability, integrity, and usability
# Data Fields in Questionnaire

## System Level Context (1 of 3)

<table>
<thead>
<tr>
<th>System Description</th>
<th>System Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>System Description</td>
</tr>
<tr>
<td></td>
<td>Services (Army, Navy, AF, etc)</td>
</tr>
<tr>
<td></td>
<td>ACAT Level</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase / Milestone</th>
<th>Current Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Start Date of Phase</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Context Information</th>
<th># of Baselines</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of Systems Fielded</td>
</tr>
<tr>
<td></td>
<td># of Variants</td>
</tr>
<tr>
<td></td>
<td># of Users</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maintenance Activities Performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance Process</td>
</tr>
<tr>
<td>Operational Tempo</td>
</tr>
<tr>
<td>Software Process Maturity</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Rights</th>
<th>Data Rights Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Data Rights Cost</td>
</tr>
<tr>
<td></td>
<td>Data Rights Ownership</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Organization Information</th>
<th>Analogous Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Funding Appropriations Used</td>
</tr>
<tr>
<td></td>
<td>Collection Date</td>
</tr>
<tr>
<td></td>
<td>POC Information</td>
</tr>
<tr>
<td></td>
<td>PEO &amp; SEC</td>
</tr>
<tr>
<td></td>
<td>Transition to SEC Date</td>
</tr>
<tr>
<td></td>
<td>Developers &amp; Current Maintainers</td>
</tr>
</tbody>
</table>
### WBS Element - Cost and Effort

<table>
<thead>
<tr>
<th>WBS Element</th>
<th>Cost and Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Total</td>
<td></td>
</tr>
<tr>
<td>Software Change Product (SW Releases)</td>
<td></td>
</tr>
<tr>
<td>Project Management</td>
<td></td>
</tr>
<tr>
<td>Software Licenses</td>
<td></td>
</tr>
<tr>
<td>Certification and Accreditation</td>
<td></td>
</tr>
<tr>
<td>System Facilities</td>
<td></td>
</tr>
<tr>
<td>Sustaining Engineering</td>
<td></td>
</tr>
<tr>
<td>Field SW Engineers</td>
<td></td>
</tr>
<tr>
<td>Operational Management</td>
<td></td>
</tr>
<tr>
<td>Labor Hours per Year &amp; Labor Rate</td>
<td></td>
</tr>
</tbody>
</table>

### License Questions

<table>
<thead>
<tr>
<th>Questions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>License Name</td>
<td></td>
</tr>
<tr>
<td>Company name</td>
<td></td>
</tr>
<tr>
<td>Quantity</td>
<td></td>
</tr>
<tr>
<td>Entitlement</td>
<td></td>
</tr>
<tr>
<td>Total Cost</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td></td>
</tr>
<tr>
<td>Award Date</td>
<td></td>
</tr>
</tbody>
</table>

- Programs were requested to report 3 years of cost and effort data broken out by the WBS as well as license information, certification frequency, and certification type (DIACAP, RMF, NSA, etc.)
- Data from government and contractor activities
## Data Fields in Questionnaire

### Software Release Level (3 of 3)

<table>
<thead>
<tr>
<th>Report Context</th>
<th>SLOC</th>
<th>Non-SLOC Sizing</th>
<th>Requirements / Interface Size</th>
<th>IAVAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% Enhancements</td>
<td>Software Language</td>
<td></td>
<td></td>
<td>Number of IAVAs Addressed</td>
</tr>
<tr>
<td>% Maintenance</td>
<td>Baseline Code Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% Cybersecurity</td>
<td>New Code Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% Other</td>
<td>Modified Code Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release Status</td>
<td>Reuse Code Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW Release Anomalies</td>
<td>Auto-Generated Code Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating Environment</td>
<td>Comments Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manned vs Unmanned</td>
<td>Deleted Code Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Domain / Super Domain</td>
<td>Delivered Code Count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release Schedule</td>
<td>Non-SLOC Sizing</td>
<td>Sizing Method</td>
<td>Total Number of Changes</td>
<td></td>
</tr>
<tr>
<td>Start Date</td>
<td>Total Size</td>
<td></td>
<td>Priority 1</td>
<td></td>
</tr>
<tr>
<td>End Date</td>
<td>Count of Size Type</td>
<td></td>
<td>Priority 2</td>
<td></td>
</tr>
<tr>
<td>Release Effort &amp; Cost</td>
<td>Number Implemented</td>
<td></td>
<td>Priority 3</td>
<td></td>
</tr>
<tr>
<td>Government Cost &amp; Hours</td>
<td></td>
<td></td>
<td>Priority 4</td>
<td></td>
</tr>
<tr>
<td>Contractor Cost &amp; Hours</td>
<td></td>
<td></td>
<td>Priority 5</td>
<td></td>
</tr>
<tr>
<td>Requirements / Interfaces Description</td>
<td>Software Changes</td>
<td>Total Number of Changes</td>
<td>Priority 5</td>
<td></td>
</tr>
<tr>
<td>Requirements at Release Start</td>
<td></td>
<td>Priority 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements Affect in Release</td>
<td></td>
<td>Priority 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total System Interfaces</td>
<td></td>
<td>Priority 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interfaces Affected in Release</td>
<td></td>
<td>Priority 4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

UNCLASSIFIED
Distribution Statement A: Approved for Public Release; Distribution is Unlimited
WBS 4.0 – Certification & Accreditation
Annual Cost by Super Domain: All Years (FY13-FY17)

RT - C&A Total
- Mean: $329,254
- Median: $179,478

<table>
<thead>
<tr>
<th>Count</th>
<th>Min</th>
<th>1st Quartile</th>
<th>Median</th>
<th>3rd Quartile</th>
<th>Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>93</td>
<td>$1,589</td>
<td>$74,146</td>
<td>$179,476</td>
<td>$368,363</td>
<td>$1,481,020</td>
<td>$329,254</td>
</tr>
</tbody>
</table>

ENG - C&A Total
- Mean: $425,711
- Median: $85,488

<table>
<thead>
<tr>
<th>Count</th>
<th>Min</th>
<th>1st Quartile</th>
<th>Median</th>
<th>3rd Quartile</th>
<th>Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>62</td>
<td>$3,070</td>
<td>$36,196</td>
<td>$95,488</td>
<td>$569,705</td>
<td>$2,600,117</td>
<td>$425,711</td>
</tr>
</tbody>
</table>

AIS - C&A Total
- Mean: $666,447
- Median: $425,550

<table>
<thead>
<tr>
<th>Count</th>
<th>Min</th>
<th>1st Quartile</th>
<th>Median</th>
<th>3rd Quartile</th>
<th>Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>$5,149</td>
<td>$178,445</td>
<td>$425,550</td>
<td>$878,470</td>
<td>$2,719,843</td>
<td>$666,447</td>
</tr>
</tbody>
</table>

SUP - C&A Total
- Mean: $447,996
- Median: $129,463

<table>
<thead>
<tr>
<th>Count</th>
<th>Min</th>
<th>1st Quartile</th>
<th>Median</th>
<th>3rd Quartile</th>
<th>Max</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>$17,262</td>
<td>$42,449</td>
<td>$129,463</td>
<td>$940,186</td>
<td>$1,463,958</td>
<td>$447,996</td>
</tr>
</tbody>
</table>
• Higher cost of C&A’s in more recent years reflects the transition period of moving from DIACAP certification to Risk Management Framework (RMF) certification which generally requires more effort.
Cost per IAVA can be used to bound the number of IAVAs a program can expect to do given a fixed budget.
Lessons Learned

Data Collection and Normalization:

- Numerous iterations were required for every data submission (average 4 submissions/program) to ensure data was accurate
- Data cleansing and normalization consumed significantly more time than expected
- Automation/use of macros streamlined data quality checks and consolidation
- Lack of standardized naming conventions extended data merging effort

Data Analysis Findings:

- Need better measures of size (output) for software sustainment
- Cybersecurity releases for many Army programs are done very frequently (monthly/weekly)
- Release descriptions indicate that COTS changes and interfaces are a prominent cause of software changes
- “Percent Enhancement” of maintenance releases is a good predictor of Software Change Product

Observations Informed by Interviews:

- Many programs did not track actual costs in detail
- There is a lack of standardized processes across the SECs/PEOs
- Delayed retirement of legacy systems generates resource/overhead burden
- Multiple funding streams limit total system cost traceability
Core Truths of Cost Estimation

- No cost estimation decision is better than the data that supports it
- If you don’t collect execution data, your cost estimate will be unreliable
- If you don’t own the data, your cost estimate will likely be untrustworthy
- If your data is not related to actual performance, your cost data will be incomplete
- If you don’t have a good software sustainment process, your cost data will be inconsistent
- If you don’t compare planned to actual performance, you can’t improve your cost estimates
- If no one asks for or uses the data, it will not exist
- If the quality of software sustainment data doesn't match that of acquisition development data, it will never be used by senior decision makers

Software is not static: it has to be continually monitored and updated to address cybersecurity issues, COTS changes, new/revised interfaces, changing platforms, platform capability shortfalls, new parameters, emerging threats, etc.
Agile to DEVOPS and its impact on estimation and measurement
Agile Software

- **Agile** software *development* refers to a group of software *development* methodologies based on iterative *development*, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

- **DevOps** Continuous software delivery that unites development and operations teams for faster business results.
Agile Manifesto

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.
The need for DevOps arose from the increasing success of agile software development, as that led to organizations wanting to release their software faster and more frequently.

As they sought to overcome the strain this put on their release management processes, they had to adopt patterns such as application release automation, continuous integration tools, and continuous delivery.

The need for DevOps has been complimented by the introduction of numerous tools that support the automation of development, deployment, operations and monitoring.
DevOps

- **DevOps** is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops).

- The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure.

- DevOps aims at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives.
Minimal Viable Product

- Minimal Viable Product (MVP): Development technique in which a new product is developed with sufficient features for early adopters
DevOps ala Seaver

- DevOps = Agile ++++
  - Development integration with operations is a key change, this usually requires organizational change to successfully implement. Get users, operations and development in synch.
  - Plus Cloud technology: cheap easily deployed development and test environment. Automated software factory that can construct and deploy tested and integrated software solutions.
  - Newer and better tools to manage information and project management of projects (this is not just a DevOps thing)
DevOps Tools

- As DevOps is intended to be a cross-functional mode of working, rather than a single DevOps tool there are sets or toolchains of multiple tools. Such DevOps tools are expected to fit into one or more of the categories listed below, reflective of key aspects of the development and delivery process:
  - **Plan** — requirements development, review and management
  - **Code** — code development and review, source code management tools, code merging
  - **Build** — continuous integration tools, build status
  - **Test** — continuous testing tools that provide feedback on business risks
  - **Package** — artifact repository, application pre-deployment staging
  - **Release** — change management, release approvals, release automation
  - **Configure** — infrastructure configuration and management, Infrastructure as Code tools
  - **Monitor** — applications performance monitoring, end-user experience
Impact on the Cost Estimation Community

- The information flow is changing, need to estimate high level capability needs statements not requirements.
  - Often buying FTE not requirements
  - Requirements not created till post contract award
  - Can collect, quantify and measure functionality as projects proceed
  - Tracking user stories provides an accurate inventory of delivered capability

- The activities and resources included in estimation need to be adjusted.
  - Systems Engineering activities moved into Software Development (much like commercial systems)
  - Testing folded into software development
  - Data Science/Data Engineering can have an increased role, particularly if analytics are involved
  - Maintenance activities are part of development project
  - Operations staff involved actively with development and test
Impact on the Cost Estimation Community (2)

- Best practices in private sector see PMO & QA functions merging into development.
  - Don’t expect these cost to totally disappear for our community, but the potential for increased efficiency does exist.
  - Acquisition change/legislative change may be required
Current plan

- Collaborate to collect data, analyze data, produce measures and recommend changes to business practices related to the estimation and measurement of Agile at Scale and or DevOps projects
Plan details

- Analysis of 4 projects
  - Estimating in parallel 3 other DevOps programs
- Collecting metrics on user stories
  - Relationships between Epics, features and stories (every project is different so far).
  - Using the transaction count from the key word scan and the SFP count that follows to normalize the story relationship model
  - Attempting to develop a model/relationship between capability need statements by domain and the number of user stories
- Develop a schema to categorize user stories
  - Functional
  - Testing
  - Task/Activity
  - Maintenance & bug fixes
<table>
<thead>
<tr>
<th>Add/Chage/Delete</th>
<th>Report</th>
<th>Query/Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept</td>
<td>Analyze</td>
<td>Browse</td>
</tr>
<tr>
<td>Add</td>
<td>Combine</td>
<td>Display</td>
</tr>
<tr>
<td>Adjust</td>
<td>Correlate</td>
<td>Enquire</td>
</tr>
<tr>
<td>Allocate</td>
<td>Detect</td>
<td>Extract</td>
</tr>
<tr>
<td>Apply</td>
<td>Distribution</td>
<td>Inquire</td>
</tr>
<tr>
<td>Assign</td>
<td>Export</td>
<td>List</td>
</tr>
<tr>
<td>Associate</td>
<td>Generate</td>
<td>Pick List</td>
</tr>
<tr>
<td>Change</td>
<td>Identify</td>
<td>View</td>
</tr>
<tr>
<td>Create</td>
<td>Inform</td>
<td></td>
</tr>
<tr>
<td>Data Source</td>
<td>Interface</td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td>Knowledge</td>
<td></td>
</tr>
<tr>
<td>Enrich</td>
<td>Measure</td>
<td></td>
</tr>
<tr>
<td>Enter</td>
<td>Outputting</td>
<td></td>
</tr>
<tr>
<td>Import</td>
<td>Provenance</td>
<td></td>
</tr>
<tr>
<td>Ingest</td>
<td>Provide</td>
<td></td>
</tr>
<tr>
<td>Inputs</td>
<td>Report</td>
<td></td>
</tr>
<tr>
<td>Interface</td>
<td>Tabulate</td>
<td></td>
</tr>
<tr>
<td>Link</td>
<td>Track</td>
<td></td>
</tr>
<tr>
<td>Log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Make Inactive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provenance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Purge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smart Data Tagging</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Estimation/Measurement Process DevOps

If ongoing project, use user story (from project) historical data to develop functionality building block patterns (typical transactions per user story normalized), use historical staffing to complete calculation.

Collect and analyze Source Code per release:
- Logical Source Code Counts
- ID GOTS/COTS FOSSCAST
- Automated Function Point Counts
- Code type metrics (Developed, Test, Duplicate...)

If new project (no history and no user/developer/operations team to work with) use organizational averages to estimate how much functionality can be implemented based on headcounts estimates.

Analyze user stories from Jira (or like):
- Function Point Size
- User story counts
- Maintenance task counts
- Project Task counts
Functionality Building Blocks

- We are attempting to identify building blocks of functionality
- Building Blocks will vary by domain
- For example, a business intelligence system might have the following building blocks
  - Data Ingest
    - Content
    - Meta data
    - Reference Data
  - Data enrichment (user interaction with data)
  - Analytics
    - Basic analytic
    - Medium analytic
    - Complex analytic
    - Machine learning/AI analytic
## Building Blocks

**Data Ingest**

<table>
<thead>
<tr>
<th></th>
<th>Add</th>
<th>Change</th>
<th>Delete</th>
<th>Query</th>
<th>Report</th>
<th>Save</th>
<th>Transactions</th>
<th>Files</th>
<th>SFP</th>
<th>Comment</th>
<th>Hours</th>
<th>FTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>16</td>
<td></td>
<td>adding basic mission data to the system</td>
<td>470</td>
<td>1.0</td>
</tr>
<tr>
<td>Meta data</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>16</td>
<td></td>
<td>meta data about mission data, time location....................................</td>
<td>470</td>
<td>1.0</td>
</tr>
<tr>
<td>Reference Data</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>16</td>
<td></td>
<td>adding additional content to mission data to make it more relevant</td>
<td>470</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Data enrichment**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>analyst add data to mission data based upon analysis and interpretation.</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Analytics**

<p>| | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic analytic</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>9</td>
<td></td>
<td></td>
<td>table and chart of table data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium analytic</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>39</td>
<td>ability for analyst to add or subtract data sets from reports. Query additional data sources</td>
<td>1,137</td>
<td>2.4</td>
</tr>
<tr>
<td>Complex analytic</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>12</td>
<td>1</td>
<td>62</td>
<td>ability for analyst to add or subtract data sets from reports. Query numerous data sources produce multiple report types</td>
<td>1,804</td>
<td>3.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Machine learning/AI analytic</th>
<th>tbd</th>
<th>tbd</th>
<th>tbd</th>
<th>tbd</th>
<th>tbd</th>
<th>tbd</th>
<th>tbd</th>
<th>tbd</th>
<th>tbd</th>
<th>tbd</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(Image by: http://www.lokeshdhakar.com/)
Automation in Practice: Assessment of an Agile Database
21 August 2018
Brittany Grissom
Cassie Robbins
Lyle Patashnick
Agenda

Background of Data
Simplified Function Point Analysis
Agile SW Growth Analysis
Schedule Analysis
Areas for Investigation
Data Background

Database contains records with over 24,000 records, with 22 fields, which include story description, estimated and actual hours, and schedule data.

<table>
<thead>
<tr>
<th>Data Hierarchy</th>
<th># of Records</th>
<th>Record Types within the hierarchy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Epic</td>
<td>28</td>
<td>Epic</td>
</tr>
<tr>
<td>Sub-Epic</td>
<td>33</td>
<td>Initiative, Capability</td>
</tr>
<tr>
<td>Feature Group</td>
<td>45</td>
<td>Initiative, Capability, Feature Group</td>
</tr>
<tr>
<td>Feature</td>
<td>4,797</td>
<td>Initiative, Capability, Feature Group, Feature, Sub-Feature</td>
</tr>
<tr>
<td>Backlog</td>
<td>19,586</td>
<td>Stories</td>
</tr>
<tr>
<td>Total</td>
<td>24,512</td>
<td></td>
</tr>
</tbody>
</table>

Contractor Mix:
- 4 small businesses
- 9 large businesses
Simplified Function Point Analysis
Simplified Function Points: Data Structure and Manipulation

Data was structured, analyzed, and counted by dividing the stories using key : value dictionary pairs of terms Dave Seaver identified for Simplified Function Points (SFPs)
Snapshots reduced data by zero value SFP’s, no support tasks, and no DRWO’s
Some “support” tasks are definitely included based on a random selection of stories but does not explain all behavior of data
Lowest level of data Illustrates need to refine counting process
► Some effort obviously not captured in counting method
► Some data has internal errors
Need to develop common lexicon
► Update to other “key” terms
► Use NLTK in Python to develop synonyms to manipulate story data
**Simplified Function Points: Higher Level Analysis**

- **Feature Group**
  
  \[ y = 3.7631x - 143.73 \]
  
  \[ R^2 = 0.6403 \]

- **Sub-Epic**
  
  \[ y = 3.1435x - 205.11 \]
  
  \[ R^2 = 0.767 \]

No multi-variate analysis

- Necessary step especially if using the lower components of function points to predict future efforts

May need to include titles in text analysis to incorporate effort worked

Data grouped in higher parent categories shows promise for counting methodology

Approved for Public Release, 18-808
Agile SW Growth Analysis
Agile SW Growth Analysis

![Scatter plot showing the relationship between estimate hours and done hours. The coefficient of determination (R²) is 0.6803.]

![Histogram showing the distribution of growth percentage bins.]

**Hours Growth Stats**
- **Mean**: 1.03
- **Median**: 0.97
- **Mode**: 1
- **Std Dev**: 0.83
- **Minimum**: -0.03
- **Maximum**: 20.97
- **Count**: 13193

---

Approved for Public Release, 18-808
Schedule
Schedule

8.5% of initially planned stories were delayed
  ▶ Some stories continue to be delayed up to 6 times (story is complete with 7 parts)
Delays appear to be addressed in consecutive sprints
2-3 week sprints could mean delays up to 3-4 months from original delivery date
Areas of Further Investigation

Multivariate Analysis for SFP
SFP impact on code growth
Different ways to slice the data into smaller, understandable chunks
Capacity limits that impact SW development
Schedule impact of multiple teams working on a feature
Impact of Scope Changes on Software Growth

Dr. Jon Brown
Gail Flynn

Naval Surface Warfare Center, Dahlgren Division
Cost Engineering and Analysis

August 2018
Agenda

- Background
- Software Growth Defined
- Analysis Methods and Results
- Model Description and Results
- Summary and Q&A
Software Growth

- Survey of recent studies measuring software growth
- Most calculated growth using initial and final reported source lines of code (SLOC) or equivalent SLOC (ESLOC)
- Method captures total growth including any growth owing to scope increases

What is the magnitude of the impact of scope growth on reported software growth?

Sources:
- Average of SRDR Data Compilation Pairs, dated 16 OCT 2017
- ICEAA June 2015 NCCA Software Growth Analysis (SW15) - Logical SLOC only
- SEI DoD SW Factbook, 2017 (CMU/SEI-2017-TR-004)
Software Growth Example

Initial Software Size

400K

Final Software Size

800K

400K ESLOC

800K ESLOC

100% Growth

- Direct comparison of final to initial ESLOC includes all sources of growth if not adjusted
Software Growth Example (cont’d)

To differentiate between the two growth metrics, we need to define some terms.

- Adjusting for scope growth would give a truer picture of the actual growth of initial software scope.
- Requires information not currently captured in SRDRs.

<table>
<thead>
<tr>
<th>Initial Software Size</th>
<th>Final Software Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial Scope</strong></td>
<td><strong>Final Scope</strong></td>
</tr>
<tr>
<td>400K ESLOC</td>
<td>800K ESLOC</td>
</tr>
<tr>
<td>400K</td>
<td>500K</td>
</tr>
</tbody>
</table>

25% Growth

New Scope

300K

500K

Initial Scope

800K
Software Size Growth (cont’d)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Underestimating required SLOC</td>
<td>Size projection errors</td>
<td>Underestimating the amount of new SLOC</td>
</tr>
<tr>
<td>Poor understanding of initial requirements</td>
<td>Requirements volatility</td>
<td>Underestimating the software complexity</td>
</tr>
<tr>
<td>Code reuse optimism</td>
<td>Product functionality changes</td>
<td>Overestimating the expected use of existing SLOC, i.e. modified and unmodified SLOC</td>
</tr>
<tr>
<td>New requirements added during development</td>
<td>Human errors</td>
<td></td>
</tr>
</tbody>
</table>

Software Growth Definition:
- Underestimating required SLOC
- Poor understanding of initial requirements
- Code reuse optimism
- New requirements added during development
Definition of Pure Software Growth

Total Growth

Pure Growth + Scope Growth

- Underestimating required SLOC
- Poor understanding of initial requirements
- Code reuse optimism
- New requirements added during development

Completely unrelated scope additions should be estimated separately and adjusted for in historical data.

Software Growth Definition:
- Underestimating required SLOC
- Poor understanding of initial requirements
- Code reuse optimism
- New requirements added during development
Examples of Pure vs Total Growth

- Four large DoD software programs were selected based on relevance and for availability of data.
- Scope changes were determined using data outside available SRDRs, which included:
  - Monthly or quarterly ESLOC reports
  - Systems Engineering Technical Review briefs
  - Program schedules
  - Software metric reports
  - Identified and interviewed subject-matter experts when possible to validate interpretations of data.
Pure vs Total Growth Program 1 (cont’d)

Program Description
- Real time
- Command and control
- Combat Management System (CMS) upgrade
- Software program: ~5000K DSLOC

<table>
<thead>
<tr>
<th>Component</th>
<th>Growth Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure Growth</td>
<td>28%</td>
</tr>
<tr>
<td>Scope Growth</td>
<td>51%</td>
</tr>
<tr>
<td>Total Growth</td>
<td>79%</td>
</tr>
</tbody>
</table>

**Large Scope Added = 140K ESLOC**
Pure vs Total Growth Program 2 (cont’d)

Program Description
- Real time
- Command and control
- CMS upgrade
- Software program: ~4000K DSLOC

Pure Growth: 20%
Scope Growth: 0%
Total Growth: 20%
Pure vs Total Growth Program 3 (cont’d)

Program Description
- Real time
- Command and control
- CMS upgrade
- Software program: ~4000K DSLOC

<table>
<thead>
<tr>
<th>Growth Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure Growth</td>
<td>24%</td>
</tr>
<tr>
<td>Scope Growth</td>
<td>37%</td>
</tr>
<tr>
<td>Total Growth</td>
<td>61%</td>
</tr>
</tbody>
</table>

Scope Added = 340K ESLOC

Contract Award

ESLOC

Reports

300,000  400,000  500,000  600,000  700,000  800,000  900,000  1,000,000  1,100,000  1,200,000  1,300,000  1,400,000  1,500,000
Pure vs Total Growth Program 4 (cont’d)

Program Description
- Real time
- Command and control
- CMS upgrade
- Software program: ~2000K DSLOC

Scope Added = 50K ESLOC

<table>
<thead>
<tr>
<th>Growth Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure Growth</td>
<td>46%</td>
</tr>
<tr>
<td>Scope Growth</td>
<td>18%</td>
</tr>
<tr>
<td>Total Growth</td>
<td>64%</td>
</tr>
</tbody>
</table>
Pure vs Total Growth Comparison

Scope growth is likely a large contributor to total software growth. What is the magnitude of this difference on software cost estimates?
Example Software Development Model: Variable Method

Design, Code, Test and Integration (DCTI)
= ESLOC X (1+ SW Growth) X Labor Rate
Productivity Rate

Non DCTI
= CER X (DCTI)

Government
= CER X (Non DCTI + DCTI)

Linear CER of DCTI

Linear CER of Non DCTI and DCTI

Cost Model

Software Development Cost Estimate

Variable Method: Functionally Correlated
Example Software Development Model: Fixed Method

Design, Code, Test and Integration (DCTI)
= ESLOC \times (1 + \text{SW Growth}) \times \text{Labor Rate} \times \text{Productivity Rate}

Non DCTI
= \text{FTEs} \times \text{Labor rate}

Government
= \text{FTEs} \times \text{Labor rate}

Cost Model

Software Development Cost Estimate

Fixed Method: FTE-Based
Impact of Pure Growth on Model Results

- Given the large impact on a software development estimate, documenting whether pure or total growth is used is critical.

Using total vs. pure software growth can result in 15–40% difference in software development cost.
Summary

- Pure growth + Scope growth = Total Growth
- Initial vs final comparisons of ESLOC measure total software growth
  - Examples demonstrate that scope growth likely contributes a large amount to total software growth and to variance in the historical dataset
- The choice of pure vs total software growth can impact your software development model 15–40%
  - Given the impact, it is crucial to document your assumption on what is included
  - Using total software growth without adjustment is equivalent to assuming estimate includes software scope growth
- The choice of risk boundaries will impact your software development estimate
  - It is essential to document your risk boundaries and assumptions to support them.
Questions, Answers, and Discussion

NSWCDD V11
Cost & Schedule Engineering & Analysis Branch
Dahlgren Virginia

Dr. Jon Brown
(540) 653-9461
jonathan.d.brown@navy.mil

Gail Flynn
(540) 653-3316
gail.flynn@navy.mil
Impact of Pure Growth on Uncertainty

Risk applied to variable method, with 50% probability

<table>
<thead>
<tr>
<th>Pure Growth</th>
<th>Total Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>A (min)</td>
</tr>
<tr>
<td></td>
<td>D</td>
</tr>
<tr>
<td>Most Likely</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>E</td>
</tr>
<tr>
<td>High</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>F (max)</td>
</tr>
</tbody>
</table>

Some Options for Risk Distribution

Option 5 = + Risk Event: Scope Increase
### Impact of Pure Growth on Uncertainty (cont’d)

Risk applied to variable method, with 50% probability

<table>
<thead>
<tr>
<th>Pure Growth</th>
<th>Total Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>A (min) D</td>
</tr>
<tr>
<td>Most Likely</td>
<td>B E</td>
</tr>
<tr>
<td>High</td>
<td>C F (max)</td>
</tr>
</tbody>
</table>

Some Options for Risk Distribution

- Option 1
- Option 2
- Option 3
- Option 4

Option 5 = Option 1 + Risk Event: Scope Increase

Your choice of pure or total software growth and risk boundaries will impact your estimate and should be documented.

![SW Development Cost Estimate ($M)](image_url)

- 5%
- 30%
Cost & Risk Analysis of Managing Modernization Projects With Cloud and Open Source Considerations

IT CAST 2018
Key Points

Modernization can be costly but is often worth it versus starting over.

Modernization approaches can reduce cost & ongoing risk.

Open source, Agile, Cloud & other technologies can help... But they come at a cost (not free).
Best Analysis of Modernization Approach Looks at Value & Time To Value to the Business

It shouldn’t be just how long and how much… Should include Business Case “WHY”
# Cost & Technical R’s of modernization

(Adapted from Microsoft & Gartner)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Retire</strong></td>
<td>• Decommission if legacy app providing little value&lt;br&gt;• Possibly roll some legacy functionality into consolidated modern application</td>
</tr>
<tr>
<td><strong>Refactor</strong></td>
<td>• Preserve behavior by improving existing code&lt;br&gt;• Possibly execute on new infrastructure (PaaS)</td>
</tr>
<tr>
<td><strong>Replace</strong></td>
<td>• If legacy app providing value but commercial alternative can be better</td>
</tr>
<tr>
<td><strong>Retain &amp; Wrap</strong></td>
<td>• RETAIN if inexpensive or impractical to modernize&lt;br&gt;• WRAP: modern wrapper around app - additional value &amp; benefits e.g. C# Java wrapper around COBOL app</td>
</tr>
<tr>
<td><strong>Rehost</strong></td>
<td>• Viable functionality buy Expensive to run&lt;br&gt;• Move VM from on-premises to new environment E.g. IaaS</td>
</tr>
<tr>
<td><strong>Redevelop</strong></td>
<td>• Application providing value but legacy language, environment&lt;br&gt;• Rewrite a new application that meets the current and upcoming requirements</td>
</tr>
</tbody>
</table>

© 2018 Copyright Galorath Incorporated
Modernization Requires “As Is” Model Discovery

• “As Is” usually requires discovery to mine knowledge
  • Business processes
  • Business rules & vocabulary
  • Logical data model models
  • Application logic
  • Physical data model
  • Program logic

Trying to change the organization processes just because of new software can be disaster
Modernization & Value: Software & IT Should Both Be Estimated (Adapted from IBM)
Modernization Costs Go Far Beyond Just Implementation: Software Total Ownership Cost Allocation

IT Services & Infrastructure Are Situational but Generally 60% of TOC

Development = Biggest Risk

- Software Development
- Software Maintenance
- IT Infrastructure
- IT Services

Software Development is about 6-10% of total ownership cost... But much more of the risk Assume $10m development could be over $100m total ownership...
Legacy Systems Have Substantial Costs That Modernization May Offset

![Staff Vs Maintenance Rigor Graph]

- **Develop**: Blue line
- **Rigor vhi+**: Red line
- **Rigor nom**: Yellow line
- **Rigor vlo**: Green line

The graph illustrates the comparison between staff and maintenance rigor over time, showing the impact of modernization on reducing costs.
Open Source
Open Source Software (OSS)

Computer software that is available in source code form:

Source code and certain other rights normally reserved for copyright holders are provided under a software license that permits users to study, change, improve and at times also to distribute the software.

- Term (OSS) now used for many license types
- Open Use
- Black Box Use
- Black Box from Vendor
- Open Use developmental
<table>
<thead>
<tr>
<th>License Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Public Domain</strong></td>
<td>• Anyone can do anything • Doesn't mean it is safe</td>
</tr>
<tr>
<td><strong>Permissive</strong></td>
<td>• Minimal requirements on software modification or redistribution • AKA: Apache Style or BSD Style or MIT license</td>
</tr>
<tr>
<td><strong>LGPL</strong></td>
<td>• GNU Lesser General Public License • Any user must be given the right to modify so your developed code might have to be exposed</td>
</tr>
<tr>
<td><strong>Copyleft</strong></td>
<td>• End user can modify and distribute new works based on your work • Derived works reside under the same license</td>
</tr>
<tr>
<td><strong>Proprietary</strong></td>
<td>• All rights reserved • Software may not be modified or redistributed</td>
</tr>
</tbody>
</table>
US Law Considers Open Source Software Commercial

- Requires application of laws, regulations, policies, and so on regarding commercial software.
- In particular, U.S. law (10 USC 2377) requires a preference for commercial items for procurement of supplies or services. 10 USC 2377 requires that the head of an agency shall ensure that procurement officials in that agency, to the maximum extent practicable:
US OMB M-16-21 Promote Reuse & Open Source.. But

• **M-16-21**, OMB’s Federal Source Code Policy: Achieving Efficiency, Transparency, and Innovation through Reusable and Open Source Software requirements

• (1) all custom-developed code must be available for reuse within the government subject to limited exceptions (e.g., national security) and

• (2) under a three-year pilot program, federal agencies must release at least 20 percent of their custom-developed code to the public as OSS.

• Goal is to promote reuse as a cost saving measure to reduce redundant coding

• Problem: Up to 63% increase in initial development effort to make software reusable in the first place
Open Source Selection Process

1. Systems Engineering:
   • Choose Open Source Candidates
   • ID obsolescence risk

2. Functional verification

3. Type & licensing choices

Vendor verification where appropriate

Cost analysis
   • Development
   • Licensing

Static Quality verification

Static code sizing (where source available)
   • Where source available

Prototype proof

When source available static analysis can provide quality and size indications
Estimate Open Source Costs

A. Estimate Selection Systems Engineering
B. Estimate Open source development cost
C. Estimate Open Source maintenance & obsolesne cost
D. Estimate open source operational license costs
## Open Source Summarized Costing Process

<table>
<thead>
<tr>
<th></th>
<th>X.1 Systems Engineering</th>
<th>X.2 Development</th>
<th>X.3 Maintenance</th>
<th>X.4 Additional Costs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Open Use</strong></td>
<td>Compute Effective Size, Functionality or SLOC,</td>
<td>Use Effective Size</td>
<td>Cost Model with Use Total or Effective Size</td>
<td>Licensing Cost</td>
</tr>
<tr>
<td></td>
<td>or use Systems Engineering model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Black Box Use</strong></td>
<td>Compute Effective Size, Functionality or SLOC</td>
<td>Similar to Open Source Open Use</td>
<td>Same as Open Use</td>
<td>Licensing Cost</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Black Box Use from Vendor</strong></td>
<td>Compute Effective Size, Functionality or SLOC</td>
<td>Various, good approach is function points</td>
<td>Same as Open Use</td>
<td>Licensing Cost plus Support</td>
</tr>
<tr>
<td><strong>Open Use Developmental</strong></td>
<td>Compute Total, Effective, New Size</td>
<td>Estimate as Development</td>
<td>Same as Open Use</td>
<td>May have licensing cost</td>
</tr>
</tbody>
</table>
Static Code Analysis Can Help Quantify Open Source Quality (Source Cast Software)

- **Reliable measurement:** CISQ Software Sizing and Quality Standards.
- **Automated:** Sizing AFP and AEFP by a tool which remove subjectivity.
- **Consistent:** Same rules and assumption from version to version.
- **Business relevant:** Risk adjusted Productivity with normalization for trending.
- **Fact based measurement:** All metrics quality, quality or complexity should be accessible by both side (client and vendor).
- **SLA or KPI:** All metrics quality, quality or complexity can be reuse in some contract focus on the evolution.
Coverity is providing a free service for open source projects

- 741 projects
- 2.5M LOC
- 44,641 defects are fixed

(Only 10.2% of identified defects are false positives in 2013)
# How To Compute Effective Size For Open Source

## Step 1: Set Redesign Factors

**Redesign Breakdown**

<table>
<thead>
<tr>
<th>Weight</th>
<th>Redesign Component</th>
<th>Formula</th>
<th>Percentage of the existing software that...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.22</td>
<td>Architectural Design Change</td>
<td>0.22<em>A + 0.78</em>B + 0.5<em>C + 0.3</em>(1-(0.22<em>A + 0.78</em>B))<em>(3</em>D+E)/4</td>
<td>requires architectural design change</td>
</tr>
<tr>
<td>0.78</td>
<td>Detailed Design Change</td>
<td></td>
<td>requires detailed design change</td>
</tr>
<tr>
<td>0.5</td>
<td>Reverse Engineering Required</td>
<td></td>
<td>requires reverse engineering</td>
</tr>
<tr>
<td>0.225</td>
<td>Redocumentation Required</td>
<td></td>
<td>requires redocumentation</td>
</tr>
<tr>
<td>0.075</td>
<td>Revalidation Required</td>
<td></td>
<td>requires revalidation with the new design</td>
</tr>
</tbody>
</table>

## Step 2: Set Reimplementation Factors

**Reimplementation Breakdown**

<table>
<thead>
<tr>
<th>Weight</th>
<th>Inputs</th>
<th>Formula</th>
<th>Percentage of the existing software that...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.37</td>
<td>Recoding Required</td>
<td>.37<em>A + .11</em>B + .52*C</td>
<td>requires actual code changes</td>
</tr>
<tr>
<td>0.11</td>
<td>Code Review Required</td>
<td></td>
<td>requires code reviews</td>
</tr>
<tr>
<td>0.52</td>
<td>Unit Testing Required</td>
<td></td>
<td>requires unit testing</td>
</tr>
</tbody>
</table>

## Step 3: Set Retest Factors

**Retest Breakdown**

<table>
<thead>
<tr>
<th>Weight</th>
<th>Inputs</th>
<th>Formula</th>
<th>Percentage of the existing software that...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Test Plans Required</td>
<td>.10<em>A + .04</em>B + .13<em>C + .25</em>D + .36<em>E + .12</em>F</td>
<td>requires test plans to be rewritten</td>
</tr>
<tr>
<td>0.04</td>
<td>Test Procedures Required</td>
<td></td>
<td>requires test procedures to be identified and written</td>
</tr>
<tr>
<td>0.13</td>
<td>Test Reports Required</td>
<td></td>
<td>requires documented test reports</td>
</tr>
<tr>
<td>0.25</td>
<td>Test Drivers Required</td>
<td></td>
<td>requires test drivers and simulators to be rewritten</td>
</tr>
<tr>
<td>0.36</td>
<td>Integration Testing</td>
<td></td>
<td>requires integration testing</td>
</tr>
<tr>
<td>0.12</td>
<td>Formal Testing</td>
<td></td>
<td>requires formal demonstration testing</td>
</tr>
</tbody>
</table>
Open Source Obsolescence Is A Cost / Schedule Risk

• OpenOffice... Open Source Competitor to Microsoft Office

• Developers moved to LibreOffice

• Openoffice seeing little development and potentially drawing potential LibreOffice users to “a defunct piece of software” PC World


• A post on the Apache OpenOffice blog from back in April, 2015 pleads for more developers. “OpenOffice is currently in the need to expand the number of its developers,” it says. “We believe that seeing our release cycle slow down would damage the whole OpenOffice ecosystem.”

For Non-Mainstream Open Source Opsolensence risk is high and must be costed

© 2018 Copyright Galorath Federal Incorporated
Open Source / COTS Cognition:

**COGNITION:**
Architectural and implementation details, not necessarily invoked, but knowledge is needed

**INSTANTIATED:**
Application

**UNINVOLVED:**
Not necessary to know (does not contribute to size)

**INSTANTIATED:**
Open Source Component
Open Source Classification & Estimation Approach

- If used as is
  - Unchanged Non-Developmental Software (NDI) need
    - Selection
    - COTS Cognition
    - Integration & Test
    - Maintenance

- If customized needs to be estimated as developmental software
  - BEWARE the cost of Government Furnished Data (GFD) open source. May not be well suited to new application

Availability does not guarantee suitability, reliability, or information assurance

© 2018 Copyright Galorath Federal Incorporated
Estimating Reused Open Source

If source available run code counter or automated function point
• Establish effective size
• Use a cost model like SEER-SEM estimate the OSS effort & RISK to understand and maintain

If source code not available
• Use function point analysis to count / estimate effort comprehending, applying, and testing the OSS.
• Use a cost model like SEER-SEM that supports function point estimating and RISK for development and maintenance
• Or Use Galorath COTS cognition to understand how much needs to be understood, used and tested

Estimate support costs from vendors

For OSS obsolescence
• Assume how many years the OSS will remain viable... then use size measures from initial sizing to cost replacement + process to estimate rework to surrounding existing systems

© 2018 Copyright Galorath Incorporated
Estimating Developmental (Modified or Total Responsibility) Open Source

Same as any reused software

Run USC code counter
  • Establish effective size
  • Use a cost model or simple $ per line to estimate

Or use Functional sizing to scope the effort
  • Then establish effective function points

For OSS obsolescence
  • Assume how many years the OSS will remain viable...
  • Then use size measures from initial sizing to cost replacement
  • + Galorath process to estimate rework to surrounding existing systems
Costing Open Source Source
System Engineering

1. Identify requirement

2. Determine acceptable open source licensing alternatives (include compatibility with your PaaS if applicable)

3. Select alternatives

4. Evaluate viable alternatives
   - Technically
   - Licenses an support cost
   - Computing resources (if major component)
   - Obsolescence risk wise

Remember: Free in Open Source doesn’t necessarily mean no cost
Cloud Costing
NIST - Cloud Service Models

“Application” Cloud SaaS

May Build On

“Platform” Cloud Web Services, Components PaaS

May Build On

“Infrastructure” (Network, Compute, Storage) IaaS

Service Models Have Blurred Together and are no longer a valuable cost driver

SaaS: Buy and use “complete apps”

PaaS: Reusability

IaaS: Environment
We Know How To Estimate Cloud Costs and ROI

- Cloud isn't so different that alternate approaches to cost, ROI or business case are needed
- Important to identify costs that will increase as well as decrease. E.g. bandwidth
- Risk must be factored in
  - E.g. data inaccessibility
- SaaS and on-premises setup costs could be similar...
  - No SaaS savings
- Measurement, estimation and ROI processes are essential to make the most viable decisions

When cloud computing is perceived as a panacea, with assumed savings, it’s buyer beware. Jobs Changing, NOT Disappearing

© 2018 Copyright Galorath Incorporated
On Premises Often Cheaper (IaaS Example)

**In-house (Buy) $8,873 total 5 years**

Replacement Server: Dell PowerEdge T430 - $3,943

Back-up Software License and agents (2 options)
1. Symantec Back-up Exec: $2,822 (includes 2014 vr and 4 agents)
2. Dell NetVault - $2,108 (includes 1 TB capacity)

Note: Costs Here EXCLUDE IT Support Costs

**Cloud (Rent) $6,423 Annual**

$535.68 * 12

Note: Costs Here EXCLUDE IT Support Costs
Cloud Solutions Still Have Major Organizational Responsibilities & Costs

<table>
<thead>
<tr>
<th></th>
<th>IaaS</th>
<th>PaaS</th>
<th>SaaS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate Data</td>
<td>Organization</td>
<td>Organization</td>
<td>Organization</td>
</tr>
<tr>
<td>Archival Backups</td>
<td>Organization</td>
<td>Organization</td>
<td>Organization</td>
</tr>
<tr>
<td>Local user support</td>
<td>Organization</td>
<td>Organization</td>
<td>Organization</td>
</tr>
<tr>
<td>Source Code</td>
<td>Organization</td>
<td>Organization</td>
<td>Vendor</td>
</tr>
<tr>
<td>Application Configuration</td>
<td>Organization</td>
<td>Organization</td>
<td>Maybe</td>
</tr>
<tr>
<td>Programming Languages</td>
<td>Organization</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td>Frameworks</td>
<td>Organization</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td>Containers</td>
<td>Organization</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td>Operating System</td>
<td>Vendor</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td>Hardware</td>
<td>Vendor</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td>Service level agreements</td>
<td>Difficult or impossible</td>
<td>Difficult or impossible</td>
<td>Difficult or impossible</td>
</tr>
</tbody>
</table>

Note The Line between IaaS and PaaS is blurring to the point that is generally NOT a cost driver
Cloud Selection & Costing Process

1. Systems Engineering:
   - Identify platform (e.g. Private, hybrid, private)
   - Number VM’s
   - Bandwidth
   - Service level
   - Additional resources
   - Identify security considerations

2. Estimate Migration Costs
   - Software Development
   - Conversion
   - Operations

3. Obtain initial supplier costs
   - Virtual machines
   - Storage costs
   - Bandwidth
   - Backup (hot, automatic, rollover)

4. Estimate cost range
Some Gottchas in Cloud Costing

• Reliability requirements can double cloud resources needed
• Security
• Hot backup can double cloud resources needed
• Is backup in cloud sufficient
• Will timing work with application being modernized
• $6.19 per hour may sound like a bargain... but that can be $54k per year
estimate

Cyber Security
Cybersecurity Costing Includes Software, Hardware, IT & Policy

Hardening of Cyber products (SW & HW)

Ongoing cyber related policies & practices

Hardening of an IT network

Above costs don’t include cost impact of breaches (Galorath studying costing breach impact)
• **Product hardening**
  
  • Estimate through software & Hardware cost models
    
    • costs of management
    
    • systems engineering
    
    • Design
    
    • testing
    
    • and qualification of systems to ensure that they meet cybersecurity requirements.

• **Existing Product cybersecurity hardening retrofit**
  
  • Includes after-the-fact product retrofit for cybersecurity
    
    • Physical (anti-tampering, physical enclosure designs, etc.)
    
    • Software side (making the code more secure).
Building Secure Systems Is Very Costly

• Building software cost can be massive...
  • Depending on the cloud supplier’s investment in security and your organization's investment
• Cost of breaches can be worse
IT System Network Hardening

• On-Premesis: Enhancing the security posture of the network through the purchase, configuration and qualification of mostly COTS items

• Cloud: Enhancing the security posture of the network through the licensing, configuration and qualification of cloud platform... Possibly hardening mostly COTS items

• Common cost activities include:
  • Research and architectural analysis
  • Network Product Purchases (firewalls, servers, IDS, IPS, etc)
  • Installation and Configuration
  • Qualification and Checkout
  • Training
  • Monitoring
Cyber Example Cost Breakdown
(Deployment of an Intrusion Prevention System)

Σ 1 Intrusion Prevention System Deployment

Σ 1.1 Research, Architecture, Analysis
  1.1.1 Business Case & Research
  1.1.2 Systems Engineering

Σ 1.2 Purchases
  1.2.1 IPS System
  1.2.2 Supporting Network Devices
  1.2.3 Licensing

Σ 1.3 Installation and Configuration
  1.3.1 IPS Hardware Installation & Configuration
  1.3.2 Supporting Network Enhancement
  1.3.3 IPS SW Installation & Configuration
  1.3.4 Event Log Analyzer
  1.3.5 Data Migration
  1.3.6 Event Log Database

Σ 1.4 Qualification & Check out (Optional)
  1.4.1 IPS Qualification

Σ 1.5 Training
  1.5.1 Admin Users
  1.5.2 IPS Operators

Σ 1.6 Monitoring
  1.6.1 Event Log Monitoring (12x5) - Gold SLA
  1.6.2 Event Log Monitoring (12x5) - Silver+ SLA
<table>
<thead>
<tr>
<th>Category</th>
<th>Sub Category</th>
<th>Application</th>
<th>Human/Technical/Both (0/1/2)</th>
<th>Identify</th>
<th>Protect</th>
<th>Detect</th>
<th>Respond</th>
<th>Recover</th>
<th>Least</th>
<th>Likely</th>
<th>Most</th>
<th>Unit (cost per)</th>
<th>Least</th>
<th>Likely</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Security</td>
<td>Data Encryption</td>
<td>Portable Encrypting Hard Drive 10T1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>$189.00</td>
<td>$1,106.50</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Email Security</td>
<td>Email Encryption</td>
<td>DomainKeys Identified Mail (DKIM)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Email Security</td>
<td>Email Encryption</td>
<td>Integrated Data Protection</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Email Security</td>
<td>Email Encryption</td>
<td>SaaS Delivery and Integrations</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Email Security</td>
<td>Email Encryption</td>
<td>Secure Email Gateway (SEG)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Endpoint Security</td>
<td>Application Control</td>
<td>Anti-Spam</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Endpoint Security</td>
<td>Application Control</td>
<td>Anti-Virus</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Endpoint Security</td>
<td>Biometric (Biological)</td>
<td>Fingerprint</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$39.99</td>
<td>$637.50</td>
<td>$2,260.00</td>
<td>$580.00</td>
<td>machine</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Endpoint Security</td>
<td>Biometric (Biological)</td>
<td>Palm</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$190.00</td>
<td>$1,259.67</td>
<td>$2,275.00</td>
<td>$914.00</td>
<td>machine</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Endpoint Security</td>
<td>Biometric (Biological)</td>
<td>Basic</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$369.99</td>
<td>$7,900.00</td>
<td>$13,770.35</td>
<td>$2,000.00</td>
<td>$500,000</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Identity Governance</td>
<td>Federated Identity</td>
<td>Single Sign on Service</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>$32 1-10 seats: 1001 seats 952.64 1001 seats 952.64</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Sub Category</th>
<th>Protected Systems</th>
<th>Computer</th>
<th>Printer</th>
<th>Cloud</th>
<th>Phone</th>
<th>Tablet</th>
<th>Server</th>
<th>Embedded</th>
<th>Threats Addressed</th>
<th>Malware</th>
<th>Trojan Horse</th>
<th>Credential Theft</th>
<th>Phishing</th>
<th>Hacking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Security</td>
<td>Data Encryption</td>
<td>Portable Encrypting Hard Drive 10T1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Email Security</td>
<td>Email Encryption</td>
<td>DomainKeys Identified Mail (DKIM)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Email Security</td>
<td>Email Encryption</td>
<td>Integrated Data Protection</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Email Security</td>
<td>Email Encryption</td>
<td>SaaS Delivery and Integrations</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Email Security</td>
<td>Email Encryption</td>
<td>Secure Email Gateway (SEG)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Key Points

Modernization can be risky & substantially misestimated

Modernization approaches can reduce cost & risk

Open source, Agile, Cloud & other technologies can help... But they come at a cost (not free)
Poorly defined applications (miscommunication between business and IT) contribute to a 66% project failure rate, costing U.S. businesses at least $30 billion every year (Forrester Research).

60% – 80% of project failures can be attributed directly to poor requirements gathering, analysis, and management (Meta Group).

50% are rolled back out of production (Gartner).

40% of problems are found by end users (Gartner).

25% – 40% of all spending on projects is wasted as a result of re-work (Carnegie Mellon).

Up to 80% of budgets are consumed fixing self-inflicted problems (Dynamic Markets Limited Study).
### Technical Debt Is Exponential When Shipping Early

#### Example early ship shows 400%+ more defects than recommended.

#### Example deferred ship shows fewer defects. Can’t get to zero.
Reuse: Watch Out For Low Cost Assumptions on “Heritage”

- Reuse or Heritage: applying existing software to a new mission (or additional innovation in its current mission)

- Effort to reuse software is routinely under estimated

Why should we care: Bad heritage assumptions often cause major schedule / cost overruns
You May Be Liable for Open Source Licensing / Costs Even If Modified

Volunteer
- Much open source is developed by volunteers
- Someone (copyright holder) controls the baseline
- E.g. Apache web server, Linux

Corporate Backed
- Commercial organizations provide support
- May be its own developed open source or leveraging off a product created by volunteer community
- Usually supported with a service level agreement
- Multiple organizations may support it
- E.G Oracle web server based on Apache

Commercial Open Source
- Open source developed or supported by a single corporation
- E.g. Oracle OpenSolaris
Continuous Development / Delivery

Design → Develop → Test → Release → Continuous Delivery
Key Components Of A Software Project That Uses Off the Shelf

- **Developmental Software:**
  - Functionality developed specifically for the project at hand
  - May include customization of COTS

- **“Glue” Code:**
  - Code written to bind COTS to developmental software
  - Development effort must be captured

- **Install & configure**

- **COTS Software:**
  - Purchased functionality
    - **Direct Cost** component of COTS integration

- **COTS Cognition:**
  - Required functionality within the COTS software that must be understood
    - **Effort** component of COTS integration
DevOps For Continuous Delivery

Software Development

Quality Assurance

IT Operations

© 2013 Copyright Galorath Incorporated
DevOps (Demand Model Can Control & Reduce Costs) Adapted from The Phoenix Project

**Project Work**
- Projects the business is demanding
- What business is asking development or IT for
- Usually managed in some way

**Internal Work**
- Housekeeping: Any internally focused activity
- E.g. installing devices, decommissioning datacenter
- Usually with little oversight or visibility and consume untold resources

**Operational Changes**
- Every day IT operations
- Planning, assessing, building, testing and deploying changes
- May include managing the for above

**Unplanned Work**
- Major source of IT technical debt
- 25% – 40% of all spending on projects is wasted as a result of re-work (Carnegie Mellon)
- Recovery work: Can put everything else on the backburner,
- Usually takes you away from meeting your goals

© 2018 Copyright Galorath Incorporated
Iaas Vs PaaS

Amazon EC2 (IaaS)

- This is because Amazon takes the responsibility of networking, storage, server and virtualization and the user is responsible for managing the Operating System, middleware, runtime, data and application.

Amazon as PaaS

- Amazon now offering managed services

- E.G. WS Lambda: Your code snippets invoked by external event
Cloud Service Breakdown

Cloud Computing

Infrastructure

IaaS

Platform

PaaS

Application

SaaS

© 2009 Copyright Galorath Incorporated
SEER Modeling Can Estimate Total Ownership Costs For Software & IT

Size, approach, help desk, training, and many more IT / software cost modeling functions
Estimating Non Developmental Open Source

1. If source available run code counter or automated function point
   • Establish effective size
   • Use a cost model like SEER-SEM estimate the OSS effort & RISK to understand and maintain

2. If source code not available
   • Use function point analysis to count / estimate effort comprehending, applying, and testing the OSS.
   • Use a cost model like SEER-SEM that supports function point estimating and RISK for development and maintenance
   • Or Use Galorath COTS cognition to understand how much needs to be understood, used and tested

3. Estimate support costs from vendors

4. For OSS obsolescence
   Assume how many years the OSS will remain viable... then use size measures from initial sizing to cost replacement + process to estimate rework to surrounding existing systems
Risk Analysis Is Critical To Understanding Full Modernization Costs

Schedule Probability

Example Application 1

Risk Estimate
Hours 2,266.02
Effort 14.91
Schedule 8.21
Cost 219,148.00
Defects 33
Probability 20.00%

Risk Estimate
Hours 4,567.54
Effort 30.05
Schedule 10.56
Cost 441,728.00
Defects 14
Probability 80.00%

Risk Estimate
Hours 3,170.89
Effort 20.86
Schedule 9.33
Cost 306,658.00
Defects 22
Probability 50.00%

© Galorath Incorporated 2017
Nearly every U.S. weapons program tested in fiscal 2014 showed “significant vulnerabilities” to cyber attacks, including misconfigured, unpatched and outdated software, the Pentagon’s chief weapons tester said in his annual report.

- Reuters.com, Jan. 2015

“The [in]ability to preserve and run software over long periods of time may make the 21st century an information black hole.”

- Google VP Vint Cerf, Feb. 2015
• Open source companies can't compete
  • E.x: Red Hat OS & server virtualization open source poster child
    • 1/3 revenue of VMware
    • 1/40th of Microsoft
• Open source companies products are competing with the free versions AND similar proprietary products

Open Source Companies Can’t be as successful as proprietary... Some source of risk
Functional Sizing For Cloud Process

Identify Objectives
- Why are we counting / estimating functions

Identify counting boundaries
- Review objectives & goals
- Discussion from reference architectures and cloud eco-system models discussions
- Business application boundaries
- Middleware boundary approaches
- IaaS and PaaS can be initiated by a “user” or “machine” (even SLA- SaaS for example)

Identify scope
- what is in and out of scope
- E.g. force.com count features and functions you are developing but not all the features of force

Identify data sets ILF & EIF
- for PaaS & SaaS what DETS are in & out
  - e.g. source code library if you are making updates to it
  - Interpretation of rules PaaS.. Design specs saved, Test docs ILFs
  - IaaS Controlled sets of data such as policy.. E.g. storage cant exceed 5 petabytes.
Categorize Software Capabilities Around The 3 Service Models

One Product Can Cross Services

Business Facing Functions

Development Platform Functions

Infrastructure/Resource Functions

IaaS

$200/FP
e.g. update bandwidth
Add more storage
Remove storage

PaaS

$800/FP
developing new app functionality

SaaS

$60/FP
No customization, simple configuration

No customization, simple configuration

Service Facing Functions
Agile Modernization Needs Estimates
#noestimates Viable For Detailed Development - Should Not Abdicate In Substantial Developments

For substantial systems

- **Business Case**
- **Evaluation of alternatives**
- **Agile or Hybrid Agile Software Development**
- **System Test (when appropriate)**
- **Maintenance & Support**

How Much?
How long?
Ownership
Cost
Go / no go

Hybrid Agile: Requirements & Design

Agile development = root level software development management...

Story point estimating is short term productivity management

It is not a business decision making process

© 2018 Copyright Galorath Incorporated
Direct, indirect & opportunity costs from cyber crime (Source HP)

Internal cost activity centers:
- Detection
- Investigation & Escalation
- Containment
- Recovery
- Ex-Post Response

Direct, indirect & opportunity costs from cyber crime

External Consequences & Costs:
- Information loss or theft
- Business disruption
- Equipment Damage
- Revenue Loss
Bad Estimates Are A Root Cause of Modernization Project Failure

- An *estimate* is the most knowledgeable statement you can make *at a particular point in time* regarding:
  - Effort / Cost
  - Schedule
  - Staffing
  - Risk
  - Reliability

- Estimates more precise with progress

- **A WELL FORMED ESTIMATE IS A DISTRIBUTION**
Cloud Labor Costs

Cloud Does Not Relieve the Organization of Costs & Responsibilities

<table>
<thead>
<tr>
<th></th>
<th>Integration as a Service (Iaas)</th>
<th>Platform as a Service (PaaS)</th>
<th>Software as a Service (SaaS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migration (One-Time) Costs</td>
<td>Possibly Development</td>
<td>Development in new platform</td>
<td>Data conversion</td>
</tr>
<tr>
<td></td>
<td>Data conversion</td>
<td>Data conversion</td>
<td>Configuration</td>
</tr>
<tr>
<td></td>
<td>Training</td>
<td>Training</td>
<td>Training</td>
</tr>
<tr>
<td>Corporate Data</td>
<td>Organization</td>
<td>Organization</td>
<td>Organization</td>
</tr>
<tr>
<td>Archival Backups (some new cloud tech may mitigate this)</td>
<td>Organization</td>
<td>Organization</td>
<td>Organization</td>
</tr>
<tr>
<td>Local user support</td>
<td>Organization</td>
<td>Organization</td>
<td>Organization</td>
</tr>
<tr>
<td>Source Code</td>
<td>Organization</td>
<td>Organization</td>
<td>Vendor</td>
</tr>
<tr>
<td>Application Configuration</td>
<td>Organization</td>
<td>Organization</td>
<td>Either</td>
</tr>
<tr>
<td>Programming Languages</td>
<td>Organization</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td>Frameworks</td>
<td>Organization</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td>Containers</td>
<td>Organization</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td>Operating System</td>
<td>Vendor (Usually)</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td>Hardware</td>
<td>Vendor</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
</tbody>
</table>
DISA Cloud Question

• DISA started offered a brokered service (not specifying the provider), but realized the complexity was overwhelming and moved to a model where you must specify the provider.

• Link to the MilCloud


• The theory is of course lower costs to acquire the open services, however integration and testing can quickly outpace the savings if the fit is not a good one, insecure, low quality, or under-performing.
Virtualization (Mark Baker)

- **Virtual workspaces:**
  - An abstraction of an execution environment that can be made dynamically available to authorized clients by using well-defined protocols,
  - Resource quota (e.g. CPU, memory share),
  - Software configuration (e.g. O/S, provided services).

- **Implement on Virtual Machines (VMs):**
  - Abstraction of a physical host machine,
  - Hypervisor intercepts and emulates instructions for VMs, and allows management of VMs,
  - VMWare, Xen, etc.

- **Provide infrastructure API:**
  - Plug-ins to hardware/support structures
Cascading Service Level Agreements (AND NEED FOR DUPLICATE SYSTEM)

Cascading relationships make SLAs nearly impossible AND you may need to have duplicate system running for hot backup.
Total Ownership Cost of Cybersecurity in a Cloud Based IT System

Software and IT – Cost Analysis Solutions Team
23 August 2018

Zachary Jasnoff
VP, Business Services
Price Systems, LLC

David Cass
VP/CISO
IBM Global Cloud Security Services

Richard Mabe
Sr. Solutions Consultant
Price Systems, LLC
Overview

- Cloud solutions for IT
- Cloud Security Concerns
- Cloud Migration Approach
- Total Cost of Ownership (TCO)
- An Integrated Framework for Cybersecurity Related TCO
- Cybersecurity cost trade-offs for business systems migration
Cloud is a means to an end, enabling many benefits ...

- Faster to market
- Higher Quality
- Cost ➣ Flexibility
- Repeatable & Scalable
- Secure & Compliant

... that require organizations to transform, and re-think –

- How to deliver capabilities while improving quality
- How to interact and react with clients
- How to resolve technical debt
- How to ensure **cybersecurity requirements** are met and cyber threats are mitigated
Integration of Roles, Processes, Information, and Technology requires additional cloud service management

Additional Service Management Needed

Provided by Cloud Provider
Management Concerns:

- Are we protected from the latest threats?
- Have we protected our most critical data?
- Do we have access to the right security skill sets?
- Are we adapting to changing platforms?
- Are we operating at an appropriate maturity level for our industry?
- Are we communicating our risks clearly to our customers and our board?
- Are we maximizing the value of our security investments?
Cloud Security

Industry compliance standards and data protection are the main inhibitors to adopting a cloud solution, including:

- **Privacy and Compliance:**
  - adapting to a threat-aware, risk-based approach vs. a compliance based, box checking approach

- **Data Protection:**
  - the personal data of millions has been compromised in data breaches

- **Human error/Insider threat:**
  - more than half of data breaches are caused by *insiders*, including employees, third-party contractors and partners

- **Security skills gap:**
  - experts predict a shortage of 1.5 million open and unfilled security positions by 2020
  - Additionally, more than 209,000 cybersecurity jobs in the U.S. are unfilled, and postings are up 74% over the past five years (Bureau of Labor Statistics by Peninsula Press, 2015)

- **Innovations:**
  - cloud, mobile, and IOT apps create unprecedented risks to organizations
  - 44% of security leaders expect a major cloud provider to suffer a significant security breach
  - 33% of organizations don’t even test their mobile apps
  - CISCO estimates that by 2020 there’ll be 50 billion devices connected

- **Advanced Attacks:**
  - more than 80% involve cyber gangs, a global business that accounts for $400B+ a year
Cloud adoption and business value is driven by workloads

- Archive
- HR / Workforce
- Mature workloads
- Isolated workloads ( Classified )
- Applications with Sensitive Data
- Applications with complex processes & transactions
- Not yet virtualized applications
- Not Ready for Cloud
- Highly customized applications
- Lowy for Cloud
- Moving to Cloud
- Database Workloads
- DevOps
- Disaster Recovery
- Regulation
- Intensive Applications
- Risk & Compliance
- Web Applications
- Collaboration
- Big Data & Analytics
- Customer Service
- Front Office / Desktop
- 3 rd Party Applications
- ERP / CRM
- High Performance Computing
- Social Business
- Mobile
- Development & Test Workloads
- Compute Workloads
- Business Processes ( e.g. Expense Reporting )
- Storage Workloads
- Information Intensive Applications
- Batch processing
- Travel / Expense Management
Cloud Transition Estimating Process

**As Is System**
(User Data Cntr)
- Operate
- Sustain

**Plan for Transition:**
- Business Case
- Change Mgmt
- Svc Level Agreement

**Transition:**
- Software
- Data
- Interfaces

**To Be System**
(Cloud Host)
- IaaS
- PaaS
- SaaS

**Recurring Costs:**
- Labor
- Materials
- Overhead
- ODCs
- Facilities
- PM/SE

**Non-Recurring Costs:**
- Modify/Refactor SW apps
- Prep data for migration
- Develop new middleware Interfaces
- Adapt to Cloud OS and Middleware Services
- PM/SE

**Execute Plan:**
- SW Porting
- Data Migration
- User Training

**Recurring Costs:**
- Fees
- Licenses
- Subscriptions
For:
- Infrastructure
- Run Time Env
- SW Services
- Access
- Cybersecurity
- PM/SE

**To Be System**
(Cloud Host)
- IaaS
- PaaS
- SaaS

**Recurring Costs:**
- Migrate
- Instantiate
- Test/Verify
- Parallel Ops
- Changeover
- Go Live
Cloud Transition Planning

As Is: Data Center
User Owned
Vertical Integration

To Be: XaaS
Fee for Svc
Virtual Domain

Analysis:
Metrics
Tools
Methods

Phase 1: Project Initiation: collect and review data; prepare transition team and assets

Phase 2: Assess the As Is Security Posture; catalog current cloud use; prepare assessment report for the client

Phase 3: Define the “target” To Be state; Analyze Requirements for the To Be Domain (Gap Analysis); present cloud security maturity framework

Phase 4: Recommend a Cloud Solution Roadmap and (potentially) a Business Case for the level of Cloud service
Cloud Transition Security Strategy

As Is: Data Center
User Owned Vertical Integration

To Be: XaaS
Fee for Svc Virtual Domain

Analysis: Metrics Tools Methods

- Requires a cooperative effort to identify, evaluate, implement and enforce security policies
- Organizations establish cloud-specific security policies that are often an extension of their corporate security policy
- A successful cloud adoption requires both cloud service consumers and cloud service providers to establish and follow their respective cloud security policies
- Security policies are often aligned to the cloud consumption and delivery model:
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)
Cloud Transition Security Approach

As Is: Data Center
User Owned Vertical Integration

To Be: XaaS
Fee for Svc Virtual Domain

Analysis:
Metrics Tools Methods

Cloud Security & Regulatory Compliance Accelerators:

• Asses the maturity and effectiveness of the current security program in-place at the client’s organization

• Manage and govern information security more effectively and efficiently at all levels of the Cloud stack

• Identify and effectively manage security and regulatory compliance requirements while driving growth of programs

• Build a more risk aware culture through education and awareness

• Improve operational security for critical infrastructure
Total Cost of IT Ownership

- Total Cost of Ownership (TCO) measures the direct and indirect costs of IT Infrastructure and Services over the life cycle of systems

\[
\text{TCO} = \text{Capital Expenses} + \text{Operational Expenses} + \text{IT Governance/Sys Mgmt}
\]

\[
\begin{align*}
\text{(Direct)} & \quad \text{(Direct + Indirect)} & \quad \text{(Overhead/Admin)} \\
\text{(Infrastructure)} & \quad \text{(Services)} & \quad \text{(PM, FM, SE, Cyber Mgmt)}
\end{align*}
\]

The transition to Cloud services:

- Change budgeting from a CAPEx focus to an OPEx focus
  - Introduces uncertainty since resource consumption is determined by workload
  - Difficult to estimate cost effective options and cost of bandwidth

- Impacts All Aspects of The Organization
  - Changes the acquisition model: infrastructure outsourced; not procured
  - Changes the compliance / security model: Cloud provider security services
  - Changes the management model: Cloud provider systems management

In calculating TCO, organizations estimate and optimize cost based on workload
Framework to Evaluate Cybersecurity Costs

**Cost Elements**

Mil-Std-881D
Cybersecurity Focus

### Business System - Cyber Specific LCC

**Capital Expenses**
- Cybersecurity Integration - Governance and Org
- Custom Workload
- Cybersecurity Services (SW)
- Cyber End User Device (HW)
- Cyber Data
- System Level Technology
- Dedicated Cyber Comm
- Infrastructure Services
- Systems Engineering (RMF)
- Cyber Test and Evaluation

**Operations Expenses**
- Cybersecurity Services - Governance and Org
- System/Services Operations
- Cybersecurity Services
- Cyber Data Services
- End User Device Support Services
- Training Services Operations
- System/Services Mgmt
- Communications Services
- Infrastructure Services
- Cyber SW Maintenance/Modification
- Managed Services Operations
- Systems Engineering (RMF)
- Recurring Cyber Tests

- Organized with Mil-Std-881D WBS, App J
  - Highlights Cybersecurity costs for trade-off analysis
  - Includes Operating and Support costs

- Cybersecurity costs do not all carry equal weight
- Drivers include:
  - Systems Engineering Labor (RMF)
  - Support Engineering Labor (RMF)
  - Initial and Recurring Cybersecurity Tests
  - Life Cycle Risk Management
    - High replacement rate for vulnerable SW/HW
    - Continuous monitoring and threat analysis
    - Continuous validation of controls related to confidentiality, availability and integrity requirements
# Map to a Common Program WBS

<table>
<thead>
<tr>
<th>Business System</th>
<th>1.1 Development/Procurement</th>
<th>1.1.1 Custom Application Services Elements</th>
<th>1.1.4 System Level Hardware</th>
<th>1.1.2 System Level Integration</th>
<th>1.1.3.1 Cyber Systems Engineering</th>
<th>1.1.4.1 Cyber Program Management</th>
<th>1.1.5 Change Management</th>
<th>1.1.6 Data Management</th>
<th>1.1.7.1 Cybersecurity Test and Evaluation</th>
<th>1.1.12 Operational Site Infrastructure</th>
<th>1.1.12.1 Hardware</th>
<th>1.1.12.2 Software Licenses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Capital Expenses</strong></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cybersecurity Integration - Governance and Org</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Custom Workload</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cybersecurity Services (SW)</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cyber End User Device (HW)</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cyber Data</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>System Level Technology</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Dedicated Cyber Comm</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Infrastructure Services</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Systems Engineering (RMF)</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cyber Test and Evaluation</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

© 2016 PRICE Systems, LLC All Rights Reserved | Decades of Cost Management Excellence
# Map to a Common Program WBS

## Cybersecurity CES

- Cost WBS (Mil-Std-881D, App J)
- Map to a Common Program WBS

## Operations Expenses

<table>
<thead>
<tr>
<th>Category</th>
<th>1 Business System</th>
<th>1.2 Recurring Annual Business System Sustainment</th>
<th>1.2.1 Program Management</th>
<th>1.2.2 Systems/Sustainment Engineering</th>
<th>1.2.3 Change Management</th>
<th>1.2.4 Help Desk</th>
<th>1.2.5 Data Cleansing/Data Mgmt</th>
<th>1.2.6 System Data Base Admin</th>
<th>1.2.7 IT Infrastructure/Network Maintenance</th>
<th>1.2.7.3 Management</th>
<th>1.2.8 HW Tech Refresh</th>
<th>1.2.8.1 Cybersecurity Equipment</th>
<th>1.2.9 SW Licenses Refresh/Update</th>
<th>1.2.9.1 Cybersecurity SW License</th>
<th>1.2.10 Cybersecurity Management</th>
<th>1.2.10.1 Compliance Operations and Tracking (RMF)</th>
<th>1.2.11 Certification/Validation</th>
<th>1.2.12 Software (Includes Cybersecurity and IA/VA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cybersecurity Services - Governance and Org</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System/Services Operations</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cybersecurity Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cyber Data Services</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>End User Device Support Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Services Operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System/Services Mgmt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communications Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infrastructure Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cyber SW Maintenance/Modification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Managed Services Operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Systems Engineering (RMF)</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recurring Cyber Tests</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2016 PRICE Systems, LLC All Rights Reserved | Decades of Cost Management Excellence
# Trade-Offs by Cloud Model: IaaS

## Cost Elements
- **Mil-Std-881D Cybersecurity Focus**

## As Is: Data Center
- User Owned
- Vertical Integration

## To Be: IaaS
- Fee for Svc
- Virtual Domain

## Business System - Cyber Specific LCC

<table>
<thead>
<tr>
<th>Capital Expenses</th>
<th>As Is Total Cost of Ownership</th>
<th>To Be Total Cost of Ownership (IaaS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cybersecurity Integration - Governance and Org</strong></td>
<td>User Funded Program Mgmt (Governance)</td>
<td>(Basic Infrastructure)</td>
</tr>
<tr>
<td>Custom Workload</td>
<td>Workload</td>
<td></td>
</tr>
<tr>
<td>Cybersecurity Services (SW)</td>
<td>User Owned/Managed App SW</td>
<td></td>
</tr>
<tr>
<td>Cyber End User Device (HW)</td>
<td>User Owned/Managed App HW</td>
<td></td>
</tr>
<tr>
<td>Cyber Data</td>
<td>User Owned Data Services</td>
<td>Fee for Cloud Provided Data Storage</td>
</tr>
<tr>
<td>System Level Technology</td>
<td>Technology</td>
<td></td>
</tr>
<tr>
<td>Dedicated Cyber Comm</td>
<td>User Owned/Managed Comm</td>
<td></td>
</tr>
<tr>
<td>Infrastructure Services</td>
<td>User Owned/Managed Infrastructure</td>
<td>Fee for Cloud Virtual Domain Services</td>
</tr>
<tr>
<td>Systems Engineering (RMF)</td>
<td>User Funded Systems Engineering</td>
<td></td>
</tr>
<tr>
<td>Cyber Test and Evaluation</td>
<td>User Funded Systems Test/Eval</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operations Expenses</th>
<th>As Is Total Cost of Ownership</th>
<th>To Be Total Cost of Ownership (IaaS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cybersecurity Services - Governance and Org</strong></td>
<td>User Funded Program Mgmt (Governance)</td>
<td>(Basic Infrastructure)</td>
</tr>
<tr>
<td>System/Services Operations</td>
<td>Workload Management and Operations</td>
<td></td>
</tr>
<tr>
<td>Cybersecurity Services</td>
<td>Data Center/Corporate Staff</td>
<td></td>
</tr>
<tr>
<td>Cyber Data Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fee for Recurring Cloud Data Storage/Management</td>
</tr>
<tr>
<td>End User Device Support Services</td>
<td>Data Center/Corporate Staff</td>
<td></td>
</tr>
<tr>
<td>Training Services Operations</td>
<td>Data Center/Corporate Staff</td>
<td></td>
</tr>
<tr>
<td>System/Services Mgmt</td>
<td>Technology Management and Operations</td>
<td></td>
</tr>
<tr>
<td>Communications Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fee for Recurring Cloud Provided/Managed Infr Services</td>
</tr>
<tr>
<td>Infrastructure Services</td>
<td>Data Center/Corporate Staff</td>
<td></td>
</tr>
<tr>
<td>Cyber SW Maintenance/Modification</td>
<td>SW Maintenance and Modifications</td>
<td>费 for Recurring Cloud Managed Help Desk/User Services</td>
</tr>
<tr>
<td>Managed Services Operations</td>
<td>User Owned/Managed SW Services</td>
<td>Fees for Recurring Cloud Provided Infrastructure Service</td>
</tr>
<tr>
<td>Systems Engineering (RMF)</td>
<td>User Funded RMF for Apps/Data</td>
<td>Fees for Recurring Cloud Provided Infrastructure Service</td>
</tr>
<tr>
<td>Recurring Cyber Tests</td>
<td>User Funded RMF for Apps/Data</td>
<td></td>
</tr>
</tbody>
</table>
### Trade-Offs by Cloud Model: PaaS

#### Cost Elements

**Cybersecurity Focus**
- Mil-Std-881D

**Vertical Integration**
- User Owned

#### As Is: Data Center

**User Owned**
- Vertical Integration

#### To Be: PaaS

**Fee for Svc**
- Virtual Domain

---

<table>
<thead>
<tr>
<th>Business System - Cyber Specific LCC</th>
<th>As Is Total Cost of Ownership</th>
<th>To Be Total Cost of Ownership (PaaS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Capital Expenses</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cybersecurity Integration - Governance and Org</td>
<td>User Funded Program Mgmt (Governance)</td>
<td>(Add Platform Services)</td>
</tr>
<tr>
<td>Custom Workload</td>
<td>Workload</td>
<td></td>
</tr>
<tr>
<td>Cybersecurity Services (SW)</td>
<td>User Owned/Managed App SW</td>
<td></td>
</tr>
<tr>
<td>Cyber End User Device (HW)</td>
<td>User Owned/Managed App HW</td>
<td></td>
</tr>
<tr>
<td>Cyber Data</td>
<td>User Owned Data Services</td>
<td></td>
</tr>
<tr>
<td><strong>System Level Technology</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dedicated Cyber Comm</td>
<td>User Owned/Managed Comm</td>
<td></td>
</tr>
<tr>
<td>Infrastructure Services</td>
<td>User Owned/Managed Infrastructure</td>
<td></td>
</tr>
<tr>
<td>Systems Engineering (RMF)</td>
<td>User Funded RMF for Apps/Data</td>
<td></td>
</tr>
<tr>
<td>Cyber Test and Evaluation</td>
<td>User Funded RMF for Apps/Data</td>
<td></td>
</tr>
<tr>
<td><strong>Operations Expenses</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cybersecurity Services - Governance and Org</td>
<td>User Funded Program Mgmt (Governance)</td>
<td>(Add Platform Services)</td>
</tr>
<tr>
<td>System/Services Operations</td>
<td>Workload Management and Operations</td>
<td></td>
</tr>
<tr>
<td>Cybersecurity Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fee for Recurring Cloud Cyber Services</td>
</tr>
<tr>
<td>Cyber Data Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fee for Recurring Cloud Data Storage/Management</td>
</tr>
<tr>
<td>End User Device Support Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fee for Recurring Cloud Platform Mgmt and Ops</td>
</tr>
<tr>
<td>Training Services Operations</td>
<td>Data Center/Corporate Staff</td>
<td>Fee for Recurring Cloud Platform Training Svcs</td>
</tr>
<tr>
<td><strong>System/Services Mgmt</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communications Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fee for Recurring Cloud Provided/Managed Comm Svcs</td>
</tr>
<tr>
<td>Infrastructure Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fees for Recurring Cloud Provided/Managed Infr Services</td>
</tr>
<tr>
<td>Cyber SW Maintenance/Modification</td>
<td>SW Maintenance and Modifications</td>
<td></td>
</tr>
<tr>
<td>Managed Services Operations</td>
<td>User Owned/Managed SW Services</td>
<td>Fee for Recurring Cloud Managed Help Desk/User Services</td>
</tr>
<tr>
<td>Systems Engineering (RMF)</td>
<td>User Funded RMF for Apps/Data</td>
<td>Fee for Recurring Cloud Provided Platform Service</td>
</tr>
<tr>
<td>Recurring Cyber Tests</td>
<td>User Funded RMF for Apps/Data</td>
<td>Fee for Recurring Cloud Provided Platform Service</td>
</tr>
</tbody>
</table>
# Trade-Offs by Cloud Model: SaaS

## Cost Elements

### Mil-Std-881D Cybersecurity Focus

### As Is: Data Center
- User Owned
- Vertical Integration

### To Be: SaaS
- Fee for Svc
- Virtual Domain

## Business System - Cyber Specific LCC

<table>
<thead>
<tr>
<th>Business System - Cyber Specific LCC</th>
<th>As Is Total Cost of Ownership</th>
<th>To Be Total Cost of Ownership (SaaS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Capital Expenses</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cybersecurity Integration - Governance and Org</td>
<td>User Funded Program Mgmt (Governance)</td>
<td>Cloud Provided Program Mgmt (Governance)</td>
</tr>
<tr>
<td>Custom Workload</td>
<td>Workload</td>
<td></td>
</tr>
<tr>
<td>Cybersecurity Services (SW)</td>
<td>User Owned/Managed App SW</td>
<td>Cloud Provided Application SW (License/Fee)</td>
</tr>
<tr>
<td>Cyber End User Device (HW)</td>
<td>User Owned/Managed App HW</td>
<td>Cloud Provided Run Time Services (License/Fees)</td>
</tr>
<tr>
<td>Cyber Data</td>
<td>User Owned Data Services</td>
<td>Fee for Cloud Provided Data Storage</td>
</tr>
<tr>
<td>System Level Technology</td>
<td>Technology</td>
<td></td>
</tr>
<tr>
<td>Dedicated Cyber Comm</td>
<td>User Owned/Managed Comm</td>
<td></td>
</tr>
<tr>
<td>Infrastructure Services</td>
<td>User Owned/Managed Infrastructure</td>
<td>Cloud Provided/Managed Platform Services (License/Fees)</td>
</tr>
<tr>
<td>Systems Engineering (RMF)</td>
<td>User Funded Systems Engineering</td>
<td>Fee for Cloud Provided Virtual Capability</td>
</tr>
<tr>
<td>Cyber Test and Evaluation</td>
<td>User Funded Systems Test/Eval</td>
<td>Cloud Provided Service for Platform HW/App SW (License/Fee)</td>
</tr>
<tr>
<td><strong>Operations Expenses</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cybersecurity Services - Governance and Org</td>
<td>User Funded Program Mgmt (Governance)</td>
<td>Cloud Provided Program Mgmt (Governance)</td>
</tr>
<tr>
<td>System/Services Operations</td>
<td>Workload Management and Operations</td>
<td>Fee/License for Recurring Cloud HW/SW Cyber Services</td>
</tr>
<tr>
<td>Cybersecurity Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fee/License for Recurring Cloud Data Storage/Management</td>
</tr>
<tr>
<td>Cyber Data Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fee/License for Recurring Cloud Platform Mgmt and Ops</td>
</tr>
<tr>
<td>End User Device Support Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fee/License for Recurring Cloud Platform Training Svc</td>
</tr>
<tr>
<td>Training Services Operations</td>
<td>Data Center/Corporate Staff</td>
<td></td>
</tr>
<tr>
<td>System/Services Mgmt</td>
<td>Technology Management and Operations</td>
<td>Fee/License for Recurring Cloud Provided/Managed Comm Svcs</td>
</tr>
<tr>
<td>Communications Services</td>
<td>Data Center/Corporate Staff</td>
<td>Fees for Recurring Cloud Provided/Managed Infra Services</td>
</tr>
<tr>
<td>Infrastructure Services</td>
<td>Data Center/Corporate Staff</td>
<td></td>
</tr>
<tr>
<td>Cyber SW Maintenance/Modification</td>
<td>SW Maintenance and Modifications</td>
<td>Fee/License for Recurring Cloud Managed Help Desk/User Svcs</td>
</tr>
<tr>
<td>Managed Services Operations</td>
<td>User Owned/Managed SW Services</td>
<td>Fee/License for Recurring Cloud Provided Platform Service</td>
</tr>
<tr>
<td>Systems Engineering (RMF)</td>
<td>User Funded Systems Engineering</td>
<td>Fee/License for Recurring Cloud Provided Platform Service</td>
</tr>
<tr>
<td>Recurring Cyber Tests</td>
<td>User Funded Systems Engineering</td>
<td>Fee/License for Recurring Cloud Provided Platform Service</td>
</tr>
</tbody>
</table>
Evaluating Cost Trade-Offs*

- The key cost-related question: how well does the cloud perform in the context of real workloads and business requirements?
  - It’s not just price, but price-performance that matters (bang-for-buck)
  - Analysis should take every cost driver into account

What to Consider:

- **Capability**: Innovation, Speed, Insight, Security
  - What are the real requirements for applications, workloads, security and service levels?
  - Can the provider meet your requirements for security and compliance (Confidentiality, Availability, Integrity)?

- **Performance**: Flexibility to position workloads, Access to new technology, Scalability
  - Can the provider’s cloud deliver the secure speed and throughput that individual workloads require?
  - Are secure choices available that deliver higher levels of performance and service?

- **Economics**: Choice of technologies, Cost/optimal ROI, Visibility and control
  - How much will it cost to achieve the needed performance/security—initially, and in the future?
  - If upgrades are needed, what will they cost?
  - Are there hidden costs?

*Cloud IT Economics, What you don’t know about TCO can hurt you. IBM Corp., 2018*
Evaluating Cost Trade-Offs

- **Compare the most meaningful Measures**

  - **Web application measures**: Computer-intensive; response, throughput and scalability
    - How many user requests are processed per second on average?
    - Do alternative environments deliver better price-performance for the required Confidentiality, Availability and Integrity required by the Application?

  - **Analytic measures**: Storage-intensive; traditional business analytics, innovative cognitive apps
    - How many input-output queries per hour can the cloud securely handle?
    - How costly is storage?

  - **Network-intensive workload measures**: Inter-application messaging; cloud-to-cloud, cloud-to-data center, data center-to-data center
    - How much cost-of-security does a messaging-intensive workload add?
    - How cost-efficient is the cloud at securely moving data and workloads?

  - **Hosted cloud measures**: move from on-premises to hosted cloud with speed and efficiency
    - How much does it cost to migrate a virtual machine to the cloud?
Trade-Off Evaluation Methodologies

- **Example 1:** The “Cloud Price Index (pCPI)*
  - Support Labor vs Utilization of Capacity and Capability

- Derive the average price of a Cloud solution using a 'basket of goods' approach:
  - Determine the total cost of a “bundle” of services, infrastructure, software and operating systems
  - Then estimate the average “price per VM-hour” and “price per GB month” for compute and storage requirements

- Evaluate to Identify Labor Efficiency and VM use:
  - The greater the number of VMs an administrator/engineer can successfully manage (i.e., its labor efficiency), the lower the unit cost per VM hour
  - The better-utilized the cloud solution (i.e., VM use), the lower the unit cost per GB month

* Total cost of ownership in private cloud: guidelines for buyers. O. Rogers and J. Atelsek, 451 Research, Sept 2017
Cloud Price Index

Commodity Scale
(User Data Center Economies)

Standard Scale
(Commercial Cloud Economies)
Trade-Off Evaluation Methodologies

- **Example 2: Predictive Analytics**
  - Encompasses a variety of statistical techniques from modeling, machine learning, and data mining that analyze current and historical facts to make predictions about future, or otherwise unknown, events (Wikipedia 2015)

- **Applied to Cloud Workloads – Industry Focus**
  - Must take into account control requirements, technical issues and business risks *(Control Objectives for Information and Related Technology)* *(CobiT)*
  - Must take into account governance best practices for information technology-enabled business investments. *(value from IT investments)* **VAL IT**

- **Best Practices – Cloud Workload Optimization Framework**
  - Frameworks such as CobiT 5.0 and Val-IT 2.0 aligns IT Strategy to Business Strategy within a compliance, governance, operational risk management context
  - Extending CAIV best practices is a useful framework applied to cloud workloads.
  - Takes into account both TCO and Workload Performance Objectives and Threshold
Optimized Cloud TCO Analysis Model

Extending CobiT* and Val-IT+ into a CAIV Framework

CobiT DS4  Ensure Continuous Service
Ensure that IT service and infrastructure can resist and recover from failures...

The Optimized TCO provides the essential “best value” framework for the strategic decision process

*Control Objects for Information and Related Technologies
+Value from IT Investments

Typical KPI
- Time to Market
- Patching (IAVA)
- SLA

Val-IT
IM4 Perform Alternative Analysis
IM7 Identify Full Life Cycle Costs and Benefits
Wrap Up

- Cybersecurity related costs are included in a number of places in a system TCO Cost Element Structure: HW, SW, Infrastructure, Governance, Operations/Sustainment/Modifications

- Cost drivers are likely Labor costs for Systems Engineering labor and Test events supporting Risk Based Management of Cybersecurity requirements for the system’s life cycle

- The optimal TCO solution is likely an affordable mix of user owned and managed applications that employ Cloud Infrastructure and Virtual Platforms
  - The User maintains responsibility for the Application Cybersecurity Assessment
  - The Cloud provider accepts responsibility and maintains authority for their Infrastructure and Virtual Domains/Platforms

- Use of predictive analytics, combined with modeling approaches like CobiT, VAL-IT and pCPI provides a consistent framework to holistically and consistently calculate TCO on a lifecycle basis

- The process is a life cycle team effort supported by the User and by the Cloud Provider
Back-Up Slides
Considerations using COBIT

In building an cloud workload optimization framework, it is important to select the aspects of CobiT that addresses the key elements of cloud workload optimization

- Minimizing service interruptions / continuous service
- Moving to cloud must insure availability and recoverability

**CobiT DS4 Ensure Continuous Service**

- The need for providing continuous IT services requires developing, maintaining and testing IT continuity plans, utilizing offsite backup storage and providing periodic continuity plan training.
- An effective continuous service process minimizes the probability and impact of a major IT service interruption on key business functions and processes.

See more at: http://www.itgovernanceblog.com/ds4-ensure-continuous-service-250.htm#sthash.qH4Jf6Ar.dpuf
Considerations using VAL-IT

In building an cloud workload optimization framework, it is important to select the aspects of VAL-IT that addresses the key elements of cloud workload optimization

- Evaluate TCO over the full life cycle

- IM4 Develop full life-cycle costs and benefits.
  - Prepare a program budget based on full economic life-cycle costs. List all intermediate and business benefits in a benefits
  - Register, and plan how they will be realized. Identify and document targets for key outcomes to be achieved, including the
  - Method for measuring and the approach for mitigating non-achievement. Submit budgets, costs, benefits and associated plans for review, refinement and sign-off.

Importance of Understanding Difference between life cycle costs between Cloud and Traditional Approaches
Mr. Cass is the VP/CISO & Managing Partner, Global Cloud Security Services for IBM. He has global responsibility for all aspects of cloud security practices, processes, and policies across the IBM Cloud & Security Services Unit. Mr. Cass serves as a regulatory SME and an Executive Steering committee member for IBM’s International Banking Customers. David is an active contributor to the FS-ISAC on Cloud Compliance and Security for financial services firms, and works closely with U.S., and International Regulators.

Previously Mr. Cass served as the SVP & Chief Information Security Officer for Elsevier. Where he lead an organization of experienced legal, risk and security professionals that provided data protection, privacy, security, and risk management guidance on a global basis for Elsevier.

David has extensive experience in IT security, risk assessment, risk management, business continuity and disaster recovery, developing security policies and procedures. He has played a key role in leading and building corporate risk & governance and information security organizations in the financial sector. As the Senior Director of Information Security Risk and Governance for Freddie Mac, David rebuilt the risk and governance function and developed a team to provide risk assessments, methodologies, tools, services, and training to improve the organization’s capabilities and maturity. Prior to that he was Vice President of Risk Management for JPMorgan Chase, and was responsible for providing an accurate assessment of the current risk management state, contributing to the future direction of risk management, continuity and disaster recovery capabilities for the organization.

David has a MSE from the University of Pennsylvania, and a MBA from MIT. He is also a frequent speaker at high profile industry conferences, and serves on the Board of Directors for PixarBio Corporation.
Zachary Jasnoff is Vice President, Professional Services for PRICE Systems, LLC. Mr. Jasnoff has over 25 years’ experience in Life Cycle Cost estimating on a wide range of defense programs and is an acknowledged expert in Affordability Management. Mr. Jasnoff began his career at the United States Government Accountability Office (GAO) where he was responsible for independent audits and investigations of defense acquisition programs.

Mr. Jasnoff then broadened his career in parametric lifecycle estimating while serving in various positions at Boeing and Lockheed-Martin. At Lockheed-Martin he was responsible for managing the Affordability Analysis group, and was the "Cost as an Independent Variable" (CAIV) author for the Littoral Combat Ship Proposal. Mr. Jasnoff also served as Vice President/Director of Business Resiliency at JPMorganChase. In this position, Mr. Jasnoff managed a staff responsible for developing best practices for measuring resiliency, value-at-risk and Total Cost of Ownership.

He has won several awards from the International Society of Parametric Analysts (ISPA) for various presentations on CAIV and advanced estimating methodologies. Mr. Jasnoff is also a firm believer in lifelong learning and, in August 2006, received his M.S.E in Technology Management from Penn Engineering and The Wharton School at the University of Pennsylvania. While at Wharton, Mr. Jasnoff was part of a team that developed intellectual property for the financial sector in Business Resiliency. He also holds an M.B.A from American University and B.A. from Villanova University.
Richard D. Mabe  
Solutions Consultant; Price Systems, LLC  

Contact:  
(856) 651-8567  
richard.mabe@pricesystems.com  

Mr. Mabe is a Senior Solutions Consultant within the Services Group of Price Systems, LLC. In this role, Mr. Mabe conducts research and develops modeling tools for a variety of programs within the federal government. Mr. Mabe also helps True Planning users develop custom solutions for life cycle cost estimates and other cost analysis products.

Mr. Mabe has over 40 years of experience as an operations analyst, focusing on logistics analysis and cost estimating for the Air Force and other government programs. Prior to his current position with Price Systems, LLC, Mr. Mabe was a Business Area Manager for Quantech Systems, Inc. at Hanscom AFB, managing a team of 20 analysts developing cost estimating products for Air Force C4I, Cyber and Networking system programs. Prior to his work at Quantech, Mr. Mabe was the Technical Advisor for the IT and Electronics Systems Division of the Air Force Cost Analysis Agency (AFCAA), providing cost research, databases and tailored tools to support independent cost estimates of AF acquisition programs. Mr. Mabe also supported several AF and DOD working groups focused on methods to apply industry best practices for SW development, cybersecurity and C4I systems integration to DOD programs.  

Prior to working for AFCAA, Mr. Mabe provided cost estimating and cost analysis support to multiple C4I, Cyber and Networking programs at Hanscom AFB, MA, - for 2 years as a PEO level Cost Chief, and for 13 years as a Technical Expert for Tecolote Research, Inc. Many of these were Joint Service programs, sharing systems and equipment with Army and Navy C4I programs. Prior to working at Tecolote, Mr. Mabe spent 6 years with TASC in Reading, MA managing a team of systems engineers and logistics analysts developing readiness based supply and logistics models for the Air Force. Prior to TASC, Mr. Mabe was an Air Force supply and logistics officer, providing hands-on support to Air Force operations in the CONUS and in USAFE. He completed his active Air Force duties by serving as an Assistant Professor for Inventory Management at the Air Force Institute of Technology.

Mr. Mabe holds a BS Degree in Geology from Boise State University, and an MS in Logistics Management from AFIT. He received a Level 3 DAWIA certification in Business-Cost Estimating, and also a Level 3 DOD Financial Management certification in Cost. He is a recipient of the AF Outstanding Civilian Career Service Award.
A Probabilistic Method for Predicting Code Growth - 2018 Update

ERIC M. SOMMER
BOPHA SENG
DAVID L. LAPORTE
MICHAEL A. ROSS

DISTRIBUTION A: Approved for public release; distribution unlimited
Outline

- Software Cost Estimating Process
- What is code growth?
- Existing Methodology - DSLOC Estimate Growth Model v7 (DEGM7)
- New Methodology - DSLOC Estimate Growth Model v8 (DEGM8)
  - Equations and Explanations
  - Technical Baseline Estimates (TBE)
  - Baseline Growth Amounts
    - Orthogonal Distance Regression (ODR)
  - Maturity
  - SRDR Filtering
  - Outputs
- Conclusion
- Contact Information
- References
- Backup
  - Variable Definition
  - Methodology Based on Specific Operating Environments
Outline

• Software Cost Estimating Process
  • What is code growth?
  • Existing Methodology - DSLOC Estimate Growth Model v7 (DEGM7)
  • New Methodology - DSLOC Estimate Growth Model v8 (DEGM8)
    • Equations and Explanations
    • Technical Baseline Estimates (TBE)
    • Baseline Growth Amounts
      • Orthogonal Distance Regression (ODR)
    • Maturity
    • SRDR Filtering
    • Outputs
  • Conclusion
  • Contact Information
  • References
  • Backup
    • Variable Definition
    • Methodology Based on Specific Operating Environments
Software Cost Estimating Process

r2-v2 SEF Process Flow:
Creating New CDERs for the Library

CARD

рин

Size
New
(DSLOC)

Size
Mod
(DSLOC)

Size
Unmod
(DSLOC)

Maturity
(0%-100%)

DSLOC Growth Model

Historical Data

Size
Growth Adj
New
(DSLOC)

Size
Growth Adj
Mod
(DSLOC)

Size
Growth Adj
Unmod
(DSLOC)

Rework
Characteristics

Cost Analyst

R_D, R_I, R_T

ACEIT

Effective Size
(ESLOC)

CARD

Acquisition GR&As & System Description

Cost Analyst

CDER Library

Joint CDER Calibrate

Historical Data

Excel

Joint CDER Model

Effective Size
(ESLOC)

ACEIT

Duration

Effort

Cost

Cost Analyst

ACEIT
• Software Cost Estimating Process
• What is code growth?
• Existing Methodology - DSLOC Estimate Growth Model v7 (DEGM7)
• New Methodology - DSLOC Estimate Growth Model v8 (DEGM8)
  • Equations and Explanations
  • Technical Baseline Estimates (TBE)
  • Baseline Growth Amounts
    • Orthogonal Distance Regression (ODR)
  • Maturity
  • SRDR Filtering
  • Outputs
• Conclusion
• Contact Information
• References
• Backup
  • Variable Definition
  • Methodology Based on Specific Operating Environments
What is code growth?

• Code growth is the difference between actual Delivered Source Lines of Code (DSLOC) of a completed software development project and its previously estimated DSLOC amount.

  Actual DSLOC > Estimated DSLOC → Growth
  Actual DSLOC < Estimated DSLOC → Growth (Shrink)

• Reasons for Code Growth:
  • The customers didn’t know what they wanted at the start of the program
  • The mission/requirements (REQTS) changed (requirements volatility)
  • The vendor finished early so the customer thought up a few things to add
  • Software regulations have changed
  • Optimistic (e.g. overestimate of unmodified DSLOC)
  • Poor DSLOC TBE
Outline

- Software Cost Estimating Process
- What is code growth?
- **Existing Methodology - DSLOC Estimate Growth Model v7 (DEGM7)**
- **New Methodology - DSLOC Estimate Growth Model v8 (DEGM8)**
  - Equations and Explanations
  - Technical Baseline Estimates (TBE)
  - Baseline Growth Amounts
    - Orthogonal Distance Regression (ODR)
  - Maturity
  - SRDR Filtering
  - Outputs
- Conclusion
- Contact Information
- References
- Backup
  - Variable Definition
  - Methodology Based on Specific Operating Environments
Existing Growth Methodology
DSLOC Estimate Growth Model v7 (DEGM7)

What we are currently using:

- Step 1: Baseline Growth (w/ uncertainty) applied to Technical Baseline
  - Based on DSLOC Estimate Growth Methodology (Ross, v07) using 2011 SRDR data
  - Factored Based Model
- Step 2: Total growth discounted based on maturity
  - Barry Boehm’s “Cone of Uncertainty”
  - Unchanged for DEGM8
Outline

- Software Cost Estimating Process
- What is code growth?
- Existing Methodology - DSLOC Estimate Growth Model v7 (DEGM7)
- New Methodology - DSLOC Estimate Growth Model v8 (DEGM8)
  - Equations and Explanations
  - Technical Baseline Estimates (TBE)
  - Baseline Growth Amounts
    - Orthogonal Distance Regression (ODR)
  - Maturity
  - SRDR Filtering
  - Outputs
- Conclusion
- Contact Information
- References
- Backup
  - Variable Definition
  - Methodology Based on Specific Operating Environments
This model represents a significant update and modernization of the DSLOC Estimate Growth Model version 7 (DEGM7) (Ross, 2011) in that:

- It is based on additional data from the 2015 SRDR database.
- It is based on a better method of regressing the historical data.
- It recognizes non-linear relationships between size and growth.
- It introduces error on the independent variable (DSLOC).
- It decomposes the version 7 notion of Pre-existing reused software into Modified software and Unmodified software.
- It recognizes correlation between New, Modified, and Unmodified growth.
Outline

- Software Cost Estimating Process
- What is code growth?
- Existing Methodology - DSLOC Estimate Growth Model v7 (DEGM7)
- New Methodology - DSLOC Estimate Growth Model v8 (DEGM8)
  - Equations and Explanations
  - Technical Baseline Estimates (TBE)
  - Baseline Growth Amounts
    - Orthogonal Distance Regression (ODR)
  - Maturity
    - SRDR Filtering
    - Outputs
- Conclusion
- Contact Information
- References
- Backup
  - Variable Definition
  - Methodology Based on Specific Operating Environments
The DEGM8 equations for applying growth and uncertainty to TBE New, Modified, and Unmodified DSLOC are shown in Figure 1.

\[ S_{DGANew} \triangleq S_{DNew} + e^{-(\text{Decay})(\text{Maturity})} \left( \tilde{b}_{GN} \epsilon_{GN} \left( \frac{S_{DNew}}{K_N} \right)^{a_{GN}} K_N - S_{DNew} \right) \]

\[ S_{DGAMod} \triangleq S_{DMod} + e^{-(\text{Decay})(\text{Maturity})} \left( \tilde{b}_{GM} \epsilon_{GM} \left( \frac{S_{DMod}}{K_M} \right)^{a_{GM}} K_M - S_{DMod} \right) \]

\[ S_{DGAMod} \triangleq S_{DUmod} + e^{-(\text{Decay})(\text{Maturity})} \left( \tilde{b}_{GU} \epsilon_{GU} \left( \frac{S_{DUmod}}{K_U} \right)^{a_{GU}} K_U - S_{DUmod} \right) \]

**Figure 1** DEGM8 equations yield the sum of the appropriate TBE DSLOC value and its calculated DSLOC growth amount. The calculated DSLOC Growth amount is the product of the baseline DSLOC growth amount (zero maturity) and the calculated estimate maturity adjustment factor.
DEGM8 Growth Equations

TBE DSLOC

The DEGM8 accepts, as input, Technical Baseline Estimate (TBE) amounts for New \( S_{D_{\text{New}}} \), Modified \( S_{D_{\text{Mod}}} \), and Unmodified DSLOC \( S_{D_{\text{Umod}}} \).

They are rendered at various times during the program; Based on some combination of engineering analysis, relevant past program experience, and expert judgment.

These estimates represent the technical team’s best guess as to what the final outcome New, Modified, and Unmodified DSLOC values will be when the system is delivered and accepted.
DEGM8 Growth Equations
Baseline Growth Amounts (DSLOC)

- DEGM8 introduces a new regression technique (ODR)
  - Baseline Growth Equation is now a power function rather than a factor
  - Historically, DoD SW intensive programs experience significant growth; this technique allows us to model error on the initial SLOC input
- $\tilde{b}_G, a_G, \varepsilon_G \rightarrow$ calculated as part of the regression technique
- $S_D, K \rightarrow$ Inputs into Baseline Growth Equation
Orthogonal Distance Regression (ODR)

• What is ODR?
  • A process for finding a “best fit” line (an estimator) through a multi-dimension set of data points (observations) by minimizing the sum of the squared orthogonal (shortest) distances between each data point and that line.

**Ordinary Least Squares (OLS)**
Minimizes the *vertical* distance between each data point and the regression line.

**Orthogonal Distance Regression (ODR)**
Minimizes the *Orthogonal* distance between each data point and the regression line.
Why is ODR better than Ordinary Least Squares (OLS) regression and its variants?

- Works in situations where there are more than two dimensions (measures) without making assumptions about which measure is dependent and which are independent (example: Space Flight Software)
- Accounts the existence of measurement error in all dimensions; not just in the “dependent” variable
Singular Value Decomposition (SVD) for DEGM8 Equation Coefficients

- To find the system of equations that define an ODR best fit line we center the data set by using the data set centroid and then applying the SVD.

\[
S_{DGANewBL} = b_{GN} S_{DNew} a_{GN}
\]

Note: Please see paper for details on SVD and transformation of SVD results to ODR line.
Maturity Adjustment Factor = $e^{-(\text{Decay})(\text{Maturity})} = e^{-(3.466)(\text{Maturity})}$

**Growth Decay:** Based on Boehm’s (1981 pp. 310-311) Cone of Uncertainty. Given the limited amount of granular, periodic, and relevant historical DSLOC estimate data available, we used Boehm’s (1981 pp. 310-311) Cone of Uncertainty as the DEGM8’s default position.
Outline

• Software Cost Estimating Process
• What is code growth?
• Existing Methodology - DSLOC Estimate Growth Model v7 (DEGM7)
• New Methodology - DSLOC Estimate Growth Model v8 (DEGM8)
  • Equations and Explanations
  • Technical Baseline Estimates (TBE)
  • Baseline Growth Amounts
    • Orthogonal Distance Regression (ODR)
  • Maturity
  • SRDR Filtering
  • Outputs
• Conclusion
• Contact Information
• References
• Backup
  • Variable Definition
  • Methodology Based on Specific Operating Environments
The baseline (default) instance of the DEGM8 equation parameter values for New, Modified, and Unmodified DSLOC is based on a subset of 2015 Software Resources Data Report (SRDR) data collected and archived by the U.S. Department of Defense’s Defense Cost and Resource Center (DCARC).

Filter criteria:

- SI: TRUE – the observation must represent a Computer Software Configuration Item (CSCI)-like Software Item (SI) (i.e., not a collection, summary, or roll-up of multiple CSCIs)
- Nonphysical: TRUE – the observation’s DSLOC values must not be measured in units of straight physical lines of code (i.e., they must be measured in logical lines of code (language statements) or non-comment physical lines of code)
- GFValid: TRUE – the observation must contain DSLOC values to calculate New, Modified, and/or Unmodified DSLOC growth factors that are all inside three geometric standard deviations from their respective population (entire database) geometric mean (see Table 2 on the next slide)

Database exhibit some CSCI’s with unrealistic growth; they are obvious outliers in the database:

- 1 Example showed a CSCI with >100x’s growth
- Filtering out data at +/- 3 Geometric SD’s is an attempt to unbiasedly remove those outliers
### SRDR Filtering

**Table 2** Statistical outlier filtering comparison; regression JCDER349 with 3 geometric standard deviation statistical outlier filtering was chosen as the basis for the DEGM8

<table>
<thead>
<tr>
<th>Statistical Outlier Filtering:</th>
<th>New DSLOC</th>
<th>Mobile DSLOC</th>
<th>Unmodified DSLOC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>JCDER345</td>
<td>JCDER349</td>
<td>JCDER346</td>
</tr>
<tr>
<td>Number of Data Points (observations):</td>
<td>302</td>
<td>225</td>
<td>213</td>
</tr>
<tr>
<td>Geometric (log space) mean of b:</td>
<td>0.7947</td>
<td>1.2084</td>
<td>1.1137</td>
</tr>
<tr>
<td>Arithmetic (unit space) mean of b:</td>
<td>3.4927</td>
<td>1.7360</td>
<td>1.4867</td>
</tr>
<tr>
<td>Standard deviation of b:</td>
<td>19.1832</td>
<td>1.8493</td>
<td>1.3418</td>
</tr>
<tr>
<td>Coefficient of Variation (CV) of b:</td>
<td>5.49</td>
<td>1.07</td>
<td>0.90</td>
</tr>
<tr>
<td>Arithmetic (unit space) mean of e:</td>
<td>1.9368</td>
<td>1.3665</td>
<td>1.2819</td>
</tr>
<tr>
<td>Standard deviation of e:</td>
<td>3.2795</td>
<td>1.2238</td>
<td>0.9590</td>
</tr>
<tr>
<td>Coefficient of Variation (CV) of e:</td>
<td>1.69</td>
<td>0.90</td>
<td>0.75</td>
</tr>
<tr>
<td>Mean Magnitude of the Relative Error:</td>
<td>61%</td>
<td>44%</td>
<td>39%</td>
</tr>
<tr>
<td>Implied Growth Factor at data set arithmetic mean baseline DSLOC:</td>
<td>96% at 74,958</td>
<td>53% at 56,443</td>
<td>49% at 60,213</td>
</tr>
<tr>
<td>Implied Growth Factor at data set geometric mean baseline DSLOC:</td>
<td>80% at 25,635</td>
<td>50% at 23,038</td>
<td>45% at 23,672</td>
</tr>
</tbody>
</table>
Outline

• Software Cost Estimating Process
• What is code growth?
• Existing Methodology - DSLOC Estimate Growth Model v7 (DEGM7)
• New Methodology - DSLOC Estimate Growth Model v8 (DEGM8)
  • Equations and Explanations
    • Technical Baseline Estimates (TBE)
    • Baseline Growth Amounts
      • Orthogonal Distance Regression (ODR)
  • Maturity
  • SRDR Filtering
  • Outputs
• Conclusion
• Contact Information
• References
• Backup
  • Variable Definition
  • Methodology Based on Specific Operating Environments
Output (Default: All Paired Data – Filtered)

ODR Equation

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>New</th>
<th>Modified</th>
<th>Unmodified</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{G_{-}}$</td>
<td>1.021</td>
<td>0.913</td>
<td>1.044</td>
</tr>
<tr>
<td>$b_{G_{-}}$</td>
<td>1.208</td>
<td>2.651</td>
<td>0.6199</td>
</tr>
</tbody>
</table>

**Baseline Growth (New DSLOC)**

- $b_{GN}\varepsilon_{GN}\left(\frac{S_{DNew}}{K_N}\right)^{a_{GN}} K_N - S_{DNew} \rightarrow 1.208\varepsilon_{GN}\left(\frac{S_{DNew}}{K_N}\right)^{1.021} K_N - S_{DNew}$

**Baseline Growth (Mod DSLOC)**

- $b_{GM}\varepsilon_{GM}\left(\frac{S_{DMod}}{K_M}\right)^{a_{GM}} K_M - S_{DMod} \rightarrow 2.651\varepsilon_{GM}\left(\frac{S_{DMod}}{K_M}\right)^{0.913} K_M - S_{DMod}$

**Baseline Growth (Unmod DSLOC)**

- $b_{GU}\varepsilon_{GU}\left(\frac{S_{DUmod}}{K_U}\right)^{a_{GU}} K_U - S_{DUmod} \rightarrow 0.6199\varepsilon_{GU}\left(\frac{S_{DUmod}}{K_U}\right)^{1.044} K_U - S_{DUmod}$
### Output (Default: All Paired Data – Filtered) Cumulative Distribution Function (CDF)

- **$\varepsilon_{GN}$, $\varepsilon_{GM}$, $\varepsilon_{GU}$**

<table>
<thead>
<tr>
<th>Percentile</th>
<th>$JCDER349$ (Custom Growth CDF)</th>
<th>$JCDER349_{e_GN_CDF}$</th>
<th>$JCDER349_{e_GM_CDF}$</th>
<th>$JCDER349_{e_GU_CDF}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td>0.22780002</td>
<td>0.19797695</td>
<td>0.31476088</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.29379456</td>
<td>0.26905482</td>
<td>0.52151790</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>0.42005378</td>
<td>0.37202772</td>
<td>0.69719375</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>0.51067177</td>
<td>0.44448984</td>
<td>0.80244290</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>0.58960123</td>
<td>0.58049465</td>
<td>0.87663594</td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>0.72495221</td>
<td>0.74853580</td>
<td>0.93359197</td>
</tr>
<tr>
<td>35</td>
<td></td>
<td>0.86357696</td>
<td>0.88755418</td>
<td>0.95855498</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>0.94805461</td>
<td>0.96684746</td>
<td>0.97517640</td>
</tr>
<tr>
<td>45</td>
<td></td>
<td>1.07607675</td>
<td>1.04115081</td>
<td>0.99675890</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>1.13572756</td>
<td>1.12987711</td>
<td>1.01372680</td>
</tr>
<tr>
<td>55</td>
<td></td>
<td>1.21842846</td>
<td>1.17893913</td>
<td>1.02592816</td>
</tr>
<tr>
<td>60</td>
<td></td>
<td>1.33350780</td>
<td>1.20158562</td>
<td>1.04526460</td>
</tr>
<tr>
<td>65</td>
<td></td>
<td>1.44006179</td>
<td>1.29116776</td>
<td>1.09504214</td>
</tr>
<tr>
<td>70</td>
<td></td>
<td>1.48139681</td>
<td>1.37838414</td>
<td>1.13383614</td>
</tr>
<tr>
<td>75</td>
<td></td>
<td>1.51552892</td>
<td>1.51895071</td>
<td>1.16555251</td>
</tr>
<tr>
<td>80</td>
<td></td>
<td>1.63383151</td>
<td>1.67307957</td>
<td>1.26139028</td>
</tr>
<tr>
<td>85</td>
<td></td>
<td>1.86949605</td>
<td>2.17763327</td>
<td>1.39181571</td>
</tr>
<tr>
<td>90</td>
<td></td>
<td>2.68118931</td>
<td>4.02790186</td>
<td>1.77923586</td>
</tr>
<tr>
<td>95</td>
<td></td>
<td>3.85981488</td>
<td>5.56055883</td>
<td>2.52099764</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>8.80095370</td>
<td>8.68058303</td>
<td>6.98289033</td>
</tr>
</tbody>
</table>

CDFs above are abbreviated for this presentation
Output (Default: All Paired Data – Filtered)

Correlation

- Correlation between DSLOC type
  - New & Modified: 0.00257
  - New & Unmodified: 0.302
  - Modified & Unmodified: 0.0745

- For this particular subset, correlation between growth is weak and will have little impact on result

- When we start to investigate growth by operating environment, there is evidence of stronger correlations

- Interesting to note that negativity correlation may exists
Assume estimating NAV CSCI for Ground System

- TBEs for New, Modified, and Unmodified software size are 25,000 DSLOC, 50,000 DSLOC, and 100,000 DSLOC respectively
- 1 CSCI (normalization of the TBEs to the historical data is unnecessary)
- Assume SLOC estimate rendered at SwRR (20% maturity)
- Assume based on Boehm’s (1981 pp. 310-311) Cone of Uncertainty
- Assume methodology based on Default Methodology (All Paired Filtered Data)

* Represents mean growth at SwRR for Notional Program
Outline

• Software Cost Estimating Process
• What is code growth?
• Existing Methodology - DSLOC Estimate Growth Model v7 (DEGM7)
• New Methodology - DSLOC Estimate Growth Model v8 (DEGM8)
  • Equations and Explanations
  • Technical Baseline Estimates (TBE)
  • Baseline Growth Amounts
    • Orthogonal Distance Regression (ODR)
  • Maturity
  • SRDR Filtering
  • Outputs
• Conclusion
• Contact Information
• References
• Backup
  • Variable Definition
  • Methodology Based on Specific Operating Environments
Conclusions/Way Ahead

• Our latest methodology (DEGM8) is based on a better method of regressing the historical data.
  - It recognizes non-linear relationships between size and growth.
  - Decomposes modified and unmodified software growth methodologies.
  - It accounts for correlation between New, Modified, and Unmodified growth.

• Way Ahead
  - Update database with 2017 SRDR
  - Continue Flight Software data collection efforts
  - Rerun the data analysis for additional software operating environments, application domains, and other characteristics of interest.
  - Create a specific growth model for each Joint Cost and Duration Estimating Relationship (JCDER)
Contact Information

Eric M. Sommer  
Operations Research Analyst  
Space and Missile Systems Center (SMC/FMCR)  
eric.sommer@us.af.mil

Bopha Seng  
Senior Analyst  
Tecolote Research, Inc.  
bseng@Tecolote.com

Mike Ross  
President/CEO  
r2Estimating, LLC  
mike.ross@r2estimating.com
References

DEGM8 Growth Equations
Baseline Growth Amounts (DSLOC)

\[
S_{DGANew} = S_{DNew} + e^{-\text{Decay}(\text{Maturity})} \left( \frac{S_{DNew} \times K_{N}}{K_{N} - S_{DNew}} \right)^{\alpha_{GN}} \cdot K_{N} - S_{DNew}
\]

\[
S_{DGAMod} = S_{DMod} + e^{-\text{Decay}(\text{Maturity})} \left( \frac{S_{DMod} \times K_{M}}{K_{M} - S_{DMod}} \right)^{\alpha_{GM}} \cdot K_{M} - S_{DMod}
\]

\[
S_{DGAMod} = S_{DUnmod} + e^{-\text{Decay}(\text{Maturity})} \left( \frac{S_{DUnmod} \times K_{U}}{K_{U} - S_{DUnmod}} \right)^{\alpha_{GU}} \cdot K_{U} - S_{DUnmod}
\]

- **Input** – Technical Baseline Estimate (TBE) of Unmodified DSLOC
- **Model** – Decay constant; default is 3.466 based on Boehm’s (1981 pp. 310–311) Cone of Uncertainty
- **Maturity** – Estimate Maturity Parameter; (SDLCA Begin (ATP, Contract Award) = 0%; SyRR = 10%; SwRR = 20%; SwPDR = 40%; SwCDR = 60%; SwTRR = 80%; SwAccept = 100%)
- **\( \varepsilon_{GN} \)** – Model – Baseline (SDLCA Begin) New DSLOC growth error factor distribution of outcomes with associated attainment probability; approximated by Custom CDF in Appendix A
- **\( \varepsilon_{GM} \)** – Model – Baseline (SDLCA Begin) Modified DSLOC growth error factor distribution of outcomes with associated attainment probability; approximated by Custom CDF in Appendix A
- **\( \varepsilon_{GU} \)** – Model – Baseline (SDLCA Begin) Unmodified DSLOC growth error factor distribution of outcomes with associated attainment probability; approximated by Custom CDF in Appendix A
- **\( \alpha_{GN}, \alpha_{GM}, \alpha_{GU} \)** – Estimator equality symbol; the left expression estimates the right expression
- **\( \delta_{GN}, \delta_{GM}, \delta_{GU} \)** – Exponent parameters for New, Modified, and Unmodified DSLOC growth estimating relationships that are calculated by the regression process
- **\( K_{N}, K_{M}, K_{U} \)** – Geometric mean (arithmetic mean in log space) scale factor parameters for New, Modified, and Unmodified DSLOC growth estimating relationships that are calculated by the regression process
- **Input** – Software Item (SI) to Computer Software Configuration Item (CSCI) normalization factors for New, Modified, and Unmodified DSLOC
Output (Default: All Pair Data – Filtered)

ODR Equation

$$\varepsilon_G - \begin{array}{c|c|c|c}
\text{Percentile} & JCDER349\_e\_GN\_CDF & JCDER349\_e\_GM\_CDF & JCDER349\_e\_GU\_CDF \\
\hline
5 & 0.22780002 & 0.19797695 & 0.31476088 \\
10 & 0.29379456 & 0.26905482 & 0.52151790 \\
15 & 0.42005378 & 0.37202772 & 0.69719375 \\
20 & 0.51067177 & 0.4448984 & 0.8024290 \\
25 & 0.58960123 & 0.58049465 & 0.87663594 \\
30 & 0.72495221 & 0.7485358 & 0.93359197 \\
35 & 0.86357696 & 0.88755418 & 0.95855498 \\
40 & 0.94805461 & 0.96684746 & 0.97517640 \\
45 & 1.07607675 & 1.044 & 0.99675890 \\
50 & 1.13577256 & 1.12987711 & 1.01372680 \\
55 & 1.21842846 & 1.17893913 & 1.02592816 \\
60 & 1.3350780 & 1.20158622 & 1.04526460 \\
65 & 1.44006179 & 1.29116776 & 1.09504214 \\
70 & 1.48139681 & 1.37838414 & 1.13383614 \\
75 & 1.51552892 & 1.51895071 & 1.16555251 \\
80 & 1.63383151 & 1.67307957 & 1.26139028 \\
85 & 1.86949605 & 2.17763327 & 1.39181571 \\
90 & 2.68118931 & 4.02790186 & 1.77923586 \\
95 & 3.8581488 & 5.56055883 & 2.52099764 \\
100 & 8.80095370 & 8.68058303 & 6.98289033 \\
\end{array}$$

Correlation

New to Modified DSLOC Correlation: 2.570E-03
New to Unmodified DSLOC Growth Correlation: 3.025E-01
Operating Environment: Fixed Ground

### Joint Cost and Duration Estimating Relationship (JCDER) Data Sheet (continued)

<table>
<thead>
<tr>
<th>Percentile</th>
<th>JCDER351_e_GN_CDF</th>
<th>JCDER351_e_GM_CDF</th>
<th>JCDER351_e_GU_CDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.22762555</td>
<td>0.28879716</td>
<td>0.20892398</td>
</tr>
<tr>
<td>10</td>
<td>0.29074110</td>
<td>0.37224510</td>
<td>0.34600877</td>
</tr>
<tr>
<td>15</td>
<td>0.32450063</td>
<td>0.50217307</td>
<td>0.52600805</td>
</tr>
<tr>
<td>20</td>
<td>0.49757245</td>
<td>0.53306236</td>
<td>0.68244218</td>
</tr>
<tr>
<td>25</td>
<td>0.56743434</td>
<td>0.56872118</td>
<td>0.70682042</td>
</tr>
<tr>
<td>30</td>
<td>0.65072762</td>
<td>0.58978237</td>
<td>0.86921910</td>
</tr>
<tr>
<td>35</td>
<td>0.75314420</td>
<td>0.59989374</td>
<td>0.95937548</td>
</tr>
<tr>
<td>40</td>
<td>0.89216252</td>
<td>0.60525967</td>
<td>1.07448472</td>
</tr>
<tr>
<td>45</td>
<td>0.92656909</td>
<td>0.73663852</td>
<td>1.08934477</td>
</tr>
<tr>
<td>50</td>
<td>1.00538050</td>
<td>0.81376043</td>
<td>1.10352074</td>
</tr>
<tr>
<td>55</td>
<td>1.19923930</td>
<td>1.08173528</td>
<td>1.11561548</td>
</tr>
<tr>
<td>60</td>
<td>1.28910046</td>
<td>1.19859686</td>
<td>1.14117023</td>
</tr>
<tr>
<td>65</td>
<td>1.59552201</td>
<td>1.34534346</td>
<td>1.16954904</td>
</tr>
<tr>
<td>70</td>
<td>1.71076921</td>
<td>1.50988568</td>
<td>1.31184404</td>
</tr>
<tr>
<td>75</td>
<td>1.76029032</td>
<td>1.57758170</td>
<td>1.4501684</td>
</tr>
<tr>
<td>80</td>
<td>1.86955357</td>
<td>1.76417333</td>
<td>1.49489554</td>
</tr>
<tr>
<td>85</td>
<td>2.23053443</td>
<td>2.4555265</td>
<td>1.64792676</td>
</tr>
<tr>
<td>90</td>
<td>2.86419186</td>
<td>4.20737683</td>
<td>1.74796409</td>
</tr>
<tr>
<td>95</td>
<td>4.32474823</td>
<td>5.51392208</td>
<td>2.87021541</td>
</tr>
<tr>
<td>100</td>
<td>8.46006991</td>
<td>6.46453042</td>
<td>4.08600963</td>
</tr>
</tbody>
</table>

| New to Modified DSLOC Correlation: | 1.399E-01 |
| New to Unmodified DSLOC Growth Correlation: | -2.361E-01 |

### DSLOC Estimate Growth Model Equations and Variables

#### New DSLOC Growth Equation:

\[ S[D\text{GA}_\text{New}] \equiv \exp(-(\text{Decay} \times \text{Maturity}))(b[\text{GN}] \times \varepsilon[\text{GN}]) \]

\[ S[D\text{New}] = K[N] / a[GN] \times K[N] - S[D\text{New}]) / S[D\text{New}] \]

#### Modified DSLOC Growth Equation:

\[ S[D\text{GA}_\text{Mod}] \equiv \exp(-(\text{Decay} \times \text{Maturity}))(b[\text{GM}] \times \varepsilon[\text{GM}]) \]

\[ S[D\text{Mod}] = K[M] / a[GM] \times K[M] - S[D\text{Mod}]) / S[D\text{Mod}] \]

#### Unmodified DSLOC Growth Equation:

\[ S[D\text{GA}_\text{Unmod}] \equiv \exp(-(\text{Decay} \times \text{Maturity}))(b[\text{GU}] \times \varepsilon[\text{GU}]) \]

\[ S[D\text{Unmod}] = K[U] / a[GU] \times K[U] - S[D\text{Unmod}]) / S[D\text{Unmod}] \]

where:

- \( a[\text{GN}] = 1.050 \)
- \( a[\text{GM}] = 0.743 \)
- \( a[\text{GU}] = 1.275 \)
- \( \varepsilon = 2.7183 \)
- \( \text{Decay} = 3.466 \)

### List Statistics

<table>
<thead>
<tr>
<th></th>
<th>[GN]</th>
<th>[GM]</th>
<th>[GU]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Data Points (observations):</td>
<td>46</td>
<td>23</td>
<td>27</td>
</tr>
<tr>
<td>Geometric (log space) mean of k:</td>
<td>1.001E+00</td>
<td>1.373E+01</td>
<td>4.149E-02</td>
</tr>
<tr>
<td>Arithmetic (unit space) mean of k:</td>
<td>1.537E+00</td>
<td>1.844E+01</td>
<td>5.440E-02</td>
</tr>
<tr>
<td>Standard deviation of k:</td>
<td>1.742E+00</td>
<td>1.332E+01</td>
<td>5.191E-02</td>
</tr>
<tr>
<td>Coefficient of Variation (CV) of k:</td>
<td>1.13</td>
<td>0.72</td>
<td>0.95</td>
</tr>
<tr>
<td>Arithmetic (unit space) mean of ( \varepsilon ):</td>
<td>1.474E+00</td>
<td>1.489E+00</td>
<td>1.189E+00</td>
</tr>
<tr>
<td>Standard deviation of ( \varepsilon ):</td>
<td>1.524E+00</td>
<td>1.591E+00</td>
<td>7.854E-01</td>
</tr>
<tr>
<td>Coefficient of Variation (CV) of ( \varepsilon ):</td>
<td>1.03</td>
<td>1.07</td>
<td>0.66</td>
</tr>
</tbody>
</table>

Mean Magnitude of the Relative Error: 48% 53% 29%
Operating Environment: Mobile Ground

<table>
<thead>
<tr>
<th>Percentile</th>
<th>JCDER501_e_GN_CDF</th>
<th>JCDER501_e_GM_CDF</th>
<th>JCDER501_e_GU_CDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.45013479</td>
<td>0.14330357</td>
<td>0.14007866</td>
</tr>
<tr>
<td>10</td>
<td>0.49553145</td>
<td>0.27018380</td>
<td>0.15824144</td>
</tr>
<tr>
<td>15</td>
<td>0.58140578</td>
<td>0.30099591</td>
<td>0.21667083</td>
</tr>
<tr>
<td>20</td>
<td>0.63885153</td>
<td>0.51185844</td>
<td>0.46130186</td>
</tr>
<tr>
<td>25</td>
<td>0.79090244</td>
<td>0.53448611</td>
<td>0.47783022</td>
</tr>
<tr>
<td>30</td>
<td>0.90596191</td>
<td>0.56632174</td>
<td>0.51610214</td>
</tr>
<tr>
<td>35</td>
<td>1.00962111</td>
<td>0.57671453</td>
<td>0.62626453</td>
</tr>
<tr>
<td>40</td>
<td>1.05811515</td>
<td>0.60395873</td>
<td>0.68075791</td>
</tr>
<tr>
<td>45</td>
<td>1.08394523</td>
<td>1.02760795</td>
<td>0.71319113</td>
</tr>
<tr>
<td>50</td>
<td>1.12029311</td>
<td>1.12956599</td>
<td>0.77768603</td>
</tr>
<tr>
<td>55</td>
<td>1.13988871</td>
<td>1.28950264</td>
<td>1.04813803</td>
</tr>
<tr>
<td>60</td>
<td>1.15677483</td>
<td>1.46473355</td>
<td>1.30497673</td>
</tr>
<tr>
<td>65</td>
<td>1.17698810</td>
<td>1.57600172</td>
<td>1.42833668</td>
</tr>
<tr>
<td>70</td>
<td>1.22701029</td>
<td>1.63317056</td>
<td>1.63640911</td>
</tr>
<tr>
<td>75</td>
<td>1.24858575</td>
<td>1.69296394</td>
<td>2.11570968</td>
</tr>
<tr>
<td>80</td>
<td>1.32829118</td>
<td>1.77517725</td>
<td>2.77984142</td>
</tr>
<tr>
<td>85</td>
<td>1.38402221</td>
<td>2.06913012</td>
<td>4.41452196</td>
</tr>
<tr>
<td>90</td>
<td>1.51039009</td>
<td>6.37208295</td>
<td>6.97308778</td>
</tr>
<tr>
<td>95</td>
<td>1.61373561</td>
<td>7.53914949</td>
<td>10.00927893</td>
</tr>
<tr>
<td>96</td>
<td>3.56984543</td>
<td>7.64541359</td>
<td>10.62341473</td>
</tr>
</tbody>
</table>

List Statistics:

<table>
<thead>
<tr>
<th></th>
<th>GN</th>
<th>GM</th>
<th>GU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Data Points (observations)</td>
<td>24</td>
<td>18</td>
<td>13</td>
</tr>
<tr>
<td>Geometric (log space) mean of $t_c$</td>
<td>2.483E+00</td>
<td>1.548E+00</td>
<td>6.739E-01</td>
</tr>
<tr>
<td>Arithmetic (unit space) mean of $t_c$</td>
<td>3.692E+00</td>
<td>2.727E+00</td>
<td>7.831E-01</td>
</tr>
<tr>
<td>Standard deviation of $t_c$</td>
<td>3.937E+00</td>
<td>2.605E+00</td>
<td>4.552E-01</td>
</tr>
<tr>
<td>Coefficient of Variation (CV) of $t_c$</td>
<td>1.07</td>
<td>1.32</td>
<td>0.58</td>
</tr>
<tr>
<td>Arithmetic (unit space) mean of $\varepsilon$</td>
<td>1.402E+00</td>
<td>1.443E+00</td>
<td>1.237E+00</td>
</tr>
<tr>
<td>Standard deviation of $\varepsilon$</td>
<td>1.238E+00</td>
<td>9.067E-01</td>
<td>1.097E+00</td>
</tr>
<tr>
<td>Coefficient of Variation (CV) of $\varepsilon$</td>
<td>0.88</td>
<td>0.63</td>
<td>0.89</td>
</tr>
<tr>
<td>Lean Magnitude of the Relative Error</td>
<td>47%</td>
<td>55%</td>
<td>22%</td>
</tr>
</tbody>
</table>

New to Modified DSLOC Correlation: 3.601E-02
New to Unmodified DSLOC Growth Correlation: 5.369E-02
Operating Environment: Unmanned Space

- **DEGM8SV (Space Vehicle)**
- Lack of Flight Software data in SRDR database
- Performed data collection
  - Insufficient ATP DSLOC estimates
  - Modified DEGM8 to account insufficient data
Military Standard 881D
Work Breakdown Structures for Defense Materiel Items

Automated Information Systems
vs.
Information Systems/Defense Business Systems

Presented By:
Neil F. Albert
NFA Consulting, LLC
(In Support of PARCA)
22 August 2018
Agenda

- Overview
- Key Takeaways
- Changes from MIL-STD-881C to 881D
- MIL-STD 881D Addition - Sustainment
- Summary
Overview

• Mil-STD 881C published October 2011
• Changes to Standard based on Government need and Industry concurrence and recommendation
• MIL STD-881D published April 2018
• New inclusions in the MIL-Std-881D
  – Cybersecurity identification
  – Expanded Common Element definitions
  – Improved Information Technology definitions
  – Improved Strategic Missile definitions
  – Life Cycle Approach
Key Takeaways

• Showing WBS numbering for each commodity:
  – Provides clarity regarding level of indenture and parent-child content
  – Maintaining the WBS numbering not essential requirement

• Extension of the WBS to lower levels may be necessary to get needed visibility
  – Only those elements that define the system will be used
  – WBS should be the same level for cost estimating and EVM reporting before extensions of the WBS are required. (i.e., If Cost Estimating reporting goes to level 5 and EVM reporting goes to level 3, the WBS should be the same for cost estimating and EVM from levels 1 through 3.)
  – Extensions for commodities can be found at http://cade.osd.mil/policy/csd-rc-plan

• Critical to understand cybersecurity cost of each system (i.e., hardware, software, program management, systems engineering, and system test and evaluation), MIL-STD provides:
  – Structure to identify, collect and report many of these critical costs (recognizing that collecting all this information is nearly impossible)
  – Where cybersecurity related costs can be easily accounted for, they should be called out as a WBS element.

• “Release” in agile development terms has a different definition; not be misinterpreted within the MIL-STD
  – Release of one or more EPIC level CSCIs, is equivalent to a release in MIL-STD

• Considered a Life Cycle Approach
Comparison of 881C to 881D

MIL-STD 881C

- Appendix K – Automated Information Systems (AIS).
- Includes the complex of enterprise elements, equipment (hardware), software, legacy systems, users, business rules, data and facilities required to develop, test and deploy an automated information system.
- The systems can be Custom Application, Enterprise Service Elements, Enterprise Information Systems, and/or External Interface Development.
- Level 2 WBS Elements generally traditional (reflects Investment only approach)
- Appendix defines the Investment structure only.

MIL-STD 881D

- The title of the Appendix J has changed from Appendix K (AIS) in 881C to reflect the current definition and purpose of the systems developed and delivered.
- Same inclusion of complex equipment, legacy systems, users, business rules, etc.
- The systems can be Custom Application, Enterprise Service Elements, Enterprise Information Systems, and/or External Interface Development.
- The Investment Level 2 WBS definitions have changed to reflect Investment only.
- Appendix J also has both an Investment WBS and a Sustainment WBS with related definitions.
## MIL-STD 881C
### AIS Work Breakdown Structure

<table>
<thead>
<tr>
<th>WBS #</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Automated Information System (AIS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Automated Information System Prime Mission Product Release/Increment X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.1</td>
<td>Custom Application Software 1…n (Specify)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.1.1</td>
<td>Subsystem Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.1.2</td>
<td>Subsystem Software CSCI 1…n (Specify)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.1.3</td>
<td>Subsystem Software Integration, Assembly, Test and Checkout</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2</td>
<td>Enterprise Service Element 1…n (Specify)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2.1</td>
<td>Enterprise Service Element Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2.2</td>
<td>Enterprise Service Element Software CSCI 1…n (Specify)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2.3</td>
<td>Enterprise Service Element Integration, Assembly, Test and Checkout</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.3</td>
<td>Enterprise Information System 1…n (Specify)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.3.1</td>
<td>Business Area Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.3.2</td>
<td>Business Area Software CSCI 1…n (Specify)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.3.3</td>
<td>Business Area Integration, Assembly, Test and Checkout</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.4</td>
<td>External System Interface Development 1…n (Specify)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.4.1</td>
<td>External System Interface Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.4.2</td>
<td>External System Interface Software CSCI 1…n (Specify)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.4.3</td>
<td>External System Interface Integration, Assembly, Test and Checkout</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.5</td>
<td>AIS Platform Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.6</td>
<td>System Level Integration</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## MIL-STD 881D IS/DBS Work Breakdown Structure

<table>
<thead>
<tr>
<th>WBS #</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Information Systems (IS)/Defense Business Systems (DBS) (Investment)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>IS/DBS Development/Customization/Configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.1</td>
<td>Custom Application 1...n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.1.1</td>
<td>Subsystem Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.1.2</td>
<td>Subsystem Software CSCI 1...n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.1.3</td>
<td>Subsystem Software Level Integration, Assembly, Test, and Checkout</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2</td>
<td>Enterprise Service Element 1...n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2.1</td>
<td>Enterprise Service Element Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2.2</td>
<td>Enterprise Service Element Software CSCI 1...n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2.3</td>
<td>Enterprise Service Element Integration, Assembly, Test, and Checkout</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.3</td>
<td>Enterprise/Management Information System 1...n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.3.1</td>
<td>Business Area Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.3.2</td>
<td>Business Area Software CSCI 1...n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.3.3</td>
<td>Business Area Integration, Assembly, Test, and Checkout</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.4</td>
<td>External System Interface Development 1...n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.4.1</td>
<td>External System Interface Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.4.2</td>
<td>External System Interface Software CSCI 1...n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.4.3</td>
<td>External System Interface Integration, Assembly, Test, and Checkout</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.5</td>
<td>System Level Hardware</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>System Level Integration</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MIL-STD 881C
AIS Work Breakdown Structure (Level 2+)

1.2 System Engineering
1.3 Program Management
1.4 Change Management
1.5 System Test and Evaluation
  1.5.1 Development Test and Evaluation
  1.5.2 Operational Test and Evaluation
  1.5.3 Mock-ups / System Integration Labs (SILs)
  1.5.4 Test and Evaluation Support
  1.5.5 Test Facilities
1.6 Training
  1.6.1 Equipment
  1.6.2 Services
  1.6.3 Facilities
1.7 Data
  1.7.1 Technical Publications
  1.7.2 Engineering Data
  1.7.3 Management Data
  1.7.4 Support Data
  1.7.5 Data Depository
MIL-STD 881D
IS/DBS Work Breakdown Structure (Level 2+)

1.3 Systems Engineering
   1.3.1 Software Systems Engineering
   1.3.2 Integrated Logistics Support (ILS) Systems Engineering
   1.3.3 Cybersecurity Systems Engineering
   1.3.4 Core Systems Engineering
   1.3.5 Other Systems Engineering 1...n (Specify)

1.4 Program Management
   1.4.1 Software Program Management
   1.4.2 Integrated Logistics Support (ILS) Program Management
   1.4.3 Cybersecurity Program Management
   1.4.4 Core Program Management
   1.4.5 Other Program Management 1...n (Specify)

1.5 Change Management

1.6 Data Management

1.7 System Test and Evaluation
   1.7.1 Development Test and Evaluation
   1.7.2 Operational Test and Evaluation
   1.7.3 Cybersecurity Test and Evaluation
   1.7.4 Mock-ups/System Integration Labs (SILs)
   1.7.5 Test Facilities

1.8 Training
   1.8.1 Equipment
   1.8.2 Services
   1.8.3 Facilities
   1.8.4 Training Software 1...n (Specify)

1.9 Data
   1.9.1 Data Deliverables 1...n (Specify)
   1.9.2 Data Repository
   1.9.3 Data Rights 1...n (Specify)
MIL-STD 881C
AIS Work Breakdown Structure (Level 2+)

1.8  Peculiar Support Equipment
1.8.1  Test and Measurement Equipment
1.8.2  Support and Handling Equipment
1.9  Common Support Equipment
1.9.1  Test and Measurement Equipment
1.9.2  Support and Handling Equipment
1.10  Operational/Site Activation
1.10.1  Site Type 1
1.10.1.1  Deployment Hardware and Software
1.10.1.2  User Documentation
1.10.1.3  Site Activation
1.10.1.4  User Training
1.10.1.5  Data Migration
1.10.1.6  Management/Engineering Support
1.10.1.7  Interim Logistics Support
1.11  Industrial Facilities
1.11.1  Construction/Conversion/Expansion
1.11.2  Equipment Acquisition or Modernization
1.11.3  Maintenance (Industrial Facilities)
1.12  Initial Spares and Repair Parts
MIL-STD 881D
IS/DBS Work Breakdown Structure (Level 2+)

1.10 Peculiar Support Equipment
1.10.1 Test and Measurement Equipment
1.10.2 Support and Handling Equipment
1.11 Common Support Equipment
1.11.1 Test and Measurement Equipment
1.11.2 Support and Handling Equipment
1.12 Operational Infrastructure/Site Activation By Site 1…n (Specify)
1.12.1 Initial Hardware Procurement
1.12.1.1 End User Equipment
1.12.1.2 Cybersecurity Equipment
1.12.1.3 IT Infrastructure and Enterprise Software Equipment
1.12.1.4 Other 1…n (Specify)
1.12.2 Initial Software License Procurement
1.12.2.1 End User Software License
1.12.2.2 Cybersecurity Software Licenses/Services
1.12.2.3 IT Infrastructure and Equipment
1.12.2.4 Other 1…n (Specify)
1.12.3 Initial Software Release (Pre-IOC) Modification/Enhancement
1.12.3.1 Routine Fixes/Deficiency Correction
1.12.3.2 Deployment Independent Verification and Validation
1.12.3.3 Installation/Test
1.12.4 Site Activation
1.12.4.1 Data Migration
1.12.4.2 User Training
1.12.4.3 User Documentation
1.12.4.4 Management/Engineering Support
1.12.4.5 Site Installation, Test, and Checkout
1.12.5 Interim Operations and Support (Pre-IOC)
1.12.5.1 Help Desk
1.12.5.2 System Database Administrator
1.12.5.3 Installation, Test, and Checkout
1.12.5.4 IT Equipment Maintenance
1.13 Industrial Facilities
1.13.1 Construction/Conversion/Expansion
1.13.2 Equipment Acquisition or Modernization
1.13.3 Maintenance (Industrial Facilities)
1.14 Initial Spares and Repair Parts
## J.4 WORK BREAKDOWN STRUCTURE LEVELS (SUSTAINMENT)

<table>
<thead>
<tr>
<th>WBS #</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>Information Systems/Defense Business Systems (IS/DBS) (Sustainment)</td>
<td>2.1 Program Management</td>
<td>2.2 Systems/Sustainment Engineering</td>
<td>2.3 Change Management</td>
<td>2.4 Help Desk</td>
</tr>
<tr>
<td>2.5</td>
<td>Data Cleansing/Data Maintenance</td>
<td>2.6 System/Database Administration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.7</td>
<td>IT Infrastructure/Network Maintenance Support</td>
<td>2.7.1 IT Infrastructure Hardware/Equipment Maintenance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.7.2</td>
<td>IT Infrastructure Software License Support Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.7.3</td>
<td>IT Infrastructure Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.7.4</td>
<td>Other IT Infrastructure Support 1...n (Specify)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8</td>
<td>Operational Hardware Refresh/Upgrade</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8.1</td>
<td>End-User Equipment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8.2</td>
<td>Cybersecurity Equipment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8.3</td>
<td>IT Infrastructure and Enterprise Services Equipment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.8.4</td>
<td>Other (Specify)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.9</td>
<td>Operational Software License Refresh/Update</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.9.1</td>
<td>End-User Software License</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.9.2</td>
<td>Cybersecurity Software Licenses/Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.9.3</td>
<td>IT Infrastructure and Enterprise Software Licenses/Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.9.4</td>
<td>Other (Specify)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.10</td>
<td>Cybersecurity Maintenance Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.10.1</td>
<td>Cybersecurity Compliance Operations and Tracking</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.10.2</td>
<td>Follow-on Cybersecurity Test and Evaluation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.10.3</td>
<td>Cybersecurity and IT Certification and Accreditation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.11</td>
<td>Follow-on User Training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.12</td>
<td>System Independent Verification and Validation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.13</td>
<td>Continuing System Improvements</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.13.1</td>
<td>Operational Hardware Modification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.13.1.1</td>
<td>Mod Kit Hardware Development</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.13.1.2</td>
<td>Mod Kit Hardware Procurement</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.13.1.3</td>
<td>Mod Kit Hardware Installation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.13.2</td>
<td>Software Release Modification/Enhancement 1...n (Specify)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.13.2.1</td>
<td>Modifications/Enhancements</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.13.2.2</td>
<td>Routine Fixes and Deficiency Correction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.13.2.3</td>
<td>Installation/Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.13.2.4</td>
<td>IAVAs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.14</td>
<td>AW/Safety/Networthiness Certification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.15</td>
<td>Facilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MIL-STD 881D Addition

• Appendix L – Relationship of the Sustainment Cost Reporting Structure to the Work Breakdown Structure
  – Provides Cost Assessment Program Evaluation CAPE Cost Reporting Structure (CRS)
  – Includes discussion on how to integrate the CAPE Sustainment CRS with the WBS for
    ▪ Interim Contractor Support (ICS)
    ▪ Contractor Logistics Support (CLS) reporting
  – CAPE CRS insufficient (at the time) to support IS/DBS sustainment activities at this time
  – IS/DBS Sustainment Structure identified in Appendix J
    ▪ Provides sustainment reporting structure
    ▪ Should be used in lieu of CAPE CRS for IS/DBS type systems
Summary

- Maintaining numbering not required
- Extension to lower level WBS elements linked to CADE website
- Identify cybersecurity elements when and where appropriate
- Expansion of definitions to support initial site activation and other support activities
- Included Hardware/Software procurement
- Added software/hardware enhancement/upgrade (pre-IOC activities)
- Interim Operations and Support
- Added Sustainment structure for IS/DBS programs
Agile Estimation
Northrop Grumman Q&A

08.22.2018

John Sautter
Sarah Nichols

Copyright 2018 Northrop Grumman Systems Corporation
John Sautter is a Lead Software Estimator working in the Northrop Grumman Technology Services SMS Division engineering staff.

- Serves as task lead in the collection and evaluation of project historical data.
- Serves as the lead facilitator of the Northrop Grumman Cost Estimation Community of Practice.
- Trained function point specialist and is the corporate liaison to the International Function Point Users Group (IFPUG).
- TS Sector member of the Software Center of Excellence and the Agile Center of Excellence with a focus on software metrics and estimation.
- Over 36 years of experience working in software engineering, project management, and organizational process improvement.
- BS degree in Computer Science and a MS in Organizational Performance.
Sarah Nichols leads the Enterprise Agile and DevSecOps transformation team within Northrop Grumman Technology Services (TS) and is the lead TS representative within the Northrop Grumman Agile Center of Excellence.

- 25+ years experience as a certified Quality and Project/Program Manager and Enterprise Agile coach.
- Strong background in integrating Agile Scrum, Scaled Agile Framework (SAFe), Lean, TDD, FDD and Kanban frameworks/methods aligned with High Maturity CMMI practice areas, Affordability and Risk Management for all types of organizations.
- Agile champion for several Fortune 50/100 companies
- Certified as a PMP, CQM/OE, CSP, CSPO, CCA, SA, SPC4, and RTE.
- BS degree in Corporate Finance/Statistics.
Agenda: Agile Estimation and Q&A

- Agile Primer
- Agile Estimation Nuances and Perceptions
- Agile Estimation Cases
- Other Insights
- Summary
Agile Primer
Initial NGC Agile Framework with Outputs

- Identify team
- Product Vision
- Training
- Initial schedule/cadence
- Initial architecture
- Capabilities/Features
- Product Roadmap
- Infrastructure setup

- Features to Stories
- Stories estimated
- Release Plan
- Release Definition of Done

- Stories to tasks with hours
- Story Definition of Done

Capture and Proposal

Release 0 (Project Startup)

Release Planning

Sprint Planning

Sprint Execution

Sprint Retrospective

Release Closeout

Sprint Review

Demo completed Stories
- Stories accepted
- Review status
- Potentially Shippable

Project Closeout

Features complete and ready for release

Prioritized actions for improvement

Tasks completed
- Regular collaboration

Daily Scrum

Framework is designed to be a customizable combination of many different Agile methods

Copyright 2018 Northrop Grumman Systems Corporation
Capabilities-Epics-Features
Definitions according to Northrop Grumman

- Product Backlog may contain hierarchy
  - Breaking down big items into smaller ones
    - *Example*: Capabilities -> Epics -> Features -> User Stories
- Capabilities account for higher-level behaviors of the solution
- Features are derived from a given Capability
- Generally, scenarios and workflows can help the team understand how the user will use the system and then generate Features
- A Feature should be completed within one Release cycle
- Features and Epics have benefits and Acceptance Criteria just like Stories
- Features are very large User Stories which are eventually broken into smaller Stories

**NOTE**: For Large Scale Agile Implementations, business Epics identify significant work (themes) which help guide value streams toward the larger aim of the portfolio. They require a formulation and analysis of cost, impact and opportunity in a lightweight business case as well as financial approval before implementation.
Product Roadmap

Roadmap is needed to plan new projects

Planning is crucial to assist the estimation effort and for reporting.
Agile Estimation Nuances and Perceptions
Estimating - New Projects vs. Projects “In-flight”

- This presentation addresses estimates for new projects or products that have the full iron triangle: Scope, Cost, Schedule
- The #NoEstimates social conversation assumes a technical staff is in place
  - Agile estimates are needed to set budgets to allow the technical staff to exist
  - These estimates also set Time & Material funding

Strong tendencies to use Story Points as software size still exists

- There are concerns in using “hours per Story Point” to use for future project estimates
- Story Points account for Effort, Complexity, Risk, and Experience level of the estimators, “not just a Size attribute”
- Recommend avoiding using non-standard sizing from historical projects
- Story Points can be used successfully to estimate future Stories if the same team who defined the historical Story Point sizing is performing the current estimating and work
Northrop Grumman uses SEER For Software (formally SEER-SEM) in new contracts needing Agile estimations.

A standard parametric tool helps answer these questions:

- What is the overall budget of this project or contract?
- What are the time spans for the Roadmap Epics?
- Are the Features broken down enough to fit into a Release cycle (i.e. quarterly or monthly)?
- How many defects would be expected when the work is done?
- What is the impact of team co-location or teams working in different time zones?
- What is the impact of adding new Capabilities or new Features to the contract?
- How can the work contribution to system-level waterfall milestones be estimated while still utilizing Agile time boxes (i.e., Sprints or Iterations)?
- How can historical data be effectively captured to use for future contracts?

A standard tool allows for normalization, comparison of projects, and encourages using the same terminology.
Agile Specific Knowledge Base Definitions in SEER for Software

- **AGILE FULL**: This knowledge base is used to describe the impacts of deploying an Agile software development life cycle approach.
  
  This methodology is independent of the numerous Agile implementation methods (Scrum, XP, ASD, etc) and considers the generic set of Agile characteristics. This methodology assumes the development team is motivated, has strong programming skills, has previously performed an Agile project, and the project will have a certified facilitator – such as a "Scrum Master." Software will be delivered using a series of incremental deliveries, where the requirements-design-code-integration process sequence is repeated using short delivery cycles, until full functionality has been reached. The primary criterion for determining the content of each repetition (often referred to as a "delivery," "block," or "release") is customer need (and is typically customer driven). The focus of this strategy is for the software to evolve as the customer requirements are interpreted and implemented over time.

- **AGILE NOVICE**: This knowledge base is similar to the Agile-Full development method, however it is used for a development team's first or initial attempt at using an Agile software development. This methodology assumes the development team is motivated, has strong programming skills, but has little to nominal experience in an Agile process or does not have a certified and experienced facilitator. The learning curve for the process is expected to increase during the project life however the team velocity will be less than optimal. Quality assurance oversight during the implementation of this new methodology will be slightly greater than for the Agile-Full methodology.

Knowledge Bases set the Overall “Tone” of the model.
Customize the model with user-defined activity names

Align activities to things that must be completed. Mapping will assist effective project reporting.
Customize the labor categories to fit the program needs.

### Labor Category Naming Scheme

<table>
<thead>
<tr>
<th>Standard Labor Category Name</th>
<th>User-Defined Labor Category Name</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>Scrum Master and MGT</td>
<td>ScrumMSTR</td>
</tr>
<tr>
<td>Software/Business Analyst</td>
<td>Product Owner</td>
<td>ProdOwner</td>
</tr>
<tr>
<td>Software Architect/Designer</td>
<td>Designer</td>
<td>Designr</td>
</tr>
<tr>
<td>Programmer</td>
<td>Programmer</td>
<td>Programr</td>
</tr>
<tr>
<td>Data Analyst/Architect</td>
<td>Data Analyst/Architect</td>
<td>DataENG</td>
</tr>
<tr>
<td>Quality Assurance/Tester</td>
<td>Quality Assurance/Tester</td>
<td>Test/QA</td>
</tr>
<tr>
<td>Configuration/Release Manager</td>
<td>Configuration/Release Manager</td>
<td>CM/RM</td>
</tr>
<tr>
<td>Quality Control Lead</td>
<td>Quality Control Lead</td>
<td>QC Lead</td>
</tr>
</tbody>
</table>

Generalize the cross-functional scrum teams. It is not an exact science.
Agile Estimation Nuances and Perceptions in Agile Estimation

Agile Sizing Options with SEER for Software

**Parameters**

**PROXY SIZING**

**Proxy Description**

This parameter allows you to select an existing proxy set for use as a sizing metric. When you choose a proxy set, the parameters below will change to reflect the choice. A proxy set can consist of anywhere from one to ten parameters, depending on how the set has been defined.

Proxies can be used together with the other sizing measures in SEER-SEM, which are source lines of code and function points. If a proxy set has already been chosen, selecting a new proxy set will overwrite the already-selected one.

Your custom proxy definitions may be added to a Microsoft Word document, "Proxy Definitions.doc", in the Tools folder.

See: [SEER-SEM Document and Application Data File Locations](#)

**Proxy Sizing**

**Story Points**

User Stories can be used to identify an initial estimate of effort required to complete an Agile project. A scoring system is used to determine the complexity of a story and to reflect how long that story will take to develop and deliver. Each user story is read in a team setting, and each member rates it using a value of: 1/2, 1, 2, 3, 5, 8, 13, 20, 40, or 100. After each team member discloses their score the differences are discussed and the team works to select the most likely value. It should be noted that a User Story with a value of 2 means that story will likely take twice as long to develop as a story with a value of 1. Likewise, a user story of 13 is NOT twice the effort of an 8.

**Using the Story Point Proxy**

Select the total number of user stories that fall within a point value. A least, likely and most range can be used to distribute the probability of total User Stories falling into any specific point range.

Story points are relative to the team expectations. The underlying effort associated with the values provided in this proxy can be adjusted using the Proxy sizing definitions option under the Tools menu item. To reduce some of the ambiguity associated with story points they have been grouped into the following categories:

<table>
<thead>
<tr>
<th>Category</th>
<th>Story Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extra Small</td>
<td>1/2 - 1</td>
</tr>
<tr>
<td>Small</td>
<td>2 - 3</td>
</tr>
<tr>
<td>Medium</td>
<td>5 - 8</td>
</tr>
<tr>
<td>Large</td>
<td>13 - 20</td>
</tr>
<tr>
<td>Extra Large</td>
<td>40</td>
</tr>
<tr>
<td>Huge</td>
<td>100</td>
</tr>
</tbody>
</table>

**T-Shirt Sized Stories**

*NOTE: Most parametric tools have similar sizing options*
SEER for Software will translate into effective size. Historical ESLOC productivity can be used to validate a function based estimate.
### Function Points

- **Notes on Function based sizing early in Life Cycle**
  - Consider using Early and Quick Function Points
  - Develop an Excel lookup table for tagging functional requirements
    - Include Simple Function Points

#### Level 1 – Standard IFPUG Set

<table>
<thead>
<tr>
<th>Basic Functional Component - Transactions</th>
<th>Min</th>
<th>ML</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIL</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>EIA</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>EIM</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>EQL</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>EQA</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>EQH</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>BOL</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>BOA</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>BOH</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>ILFL</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>ILFM</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>ILFH</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>EILF</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>EIFM</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>EIFH</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Basic Functional Component - Data</th>
<th>Min</th>
<th>ML</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEOH</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>ILFL</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>ILFM</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>ILFH</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>EIFL</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>EIFM</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>EIFH</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

#### Level 2

<table>
<thead>
<tr>
<th>Unclassified Elementary Process</th>
<th>Min</th>
<th>ML</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEI</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>GEO</td>
<td>3.7</td>
<td>3.9</td>
<td>4.1</td>
</tr>
<tr>
<td>GEOA</td>
<td>4.9</td>
<td>5.2</td>
<td>5.4</td>
</tr>
<tr>
<td>UGO</td>
<td>4.1</td>
<td>4.6</td>
<td>5</td>
</tr>
<tr>
<td>UGP</td>
<td>4</td>
<td>4.4</td>
<td>4.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Unclassified Logical File</th>
<th>Min</th>
<th>ML</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>GILF</td>
<td>7.4</td>
<td>7.7</td>
<td>8</td>
</tr>
<tr>
<td>FEIF</td>
<td>5.2</td>
<td>5.4</td>
<td>5.7</td>
</tr>
<tr>
<td>UGDG</td>
<td>6.4</td>
<td>7.1</td>
<td>7.8</td>
</tr>
</tbody>
</table>

### Level 3

<table>
<thead>
<tr>
<th>Typical Process</th>
<th>Min</th>
<th>ML</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPS</td>
<td>14.1</td>
<td>16.5</td>
<td>19</td>
</tr>
<tr>
<td>TPM</td>
<td>17.9</td>
<td>21.1</td>
<td>24.3</td>
</tr>
<tr>
<td>TPL</td>
<td>22.3</td>
<td>26.3</td>
<td>30.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General Process</th>
<th>Min</th>
<th>ML</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS</td>
<td>26.4</td>
<td>35.2</td>
<td>44</td>
</tr>
<tr>
<td>GFM</td>
<td>42.9</td>
<td>57.2</td>
<td>71.5</td>
</tr>
<tr>
<td>GPL</td>
<td>59.4</td>
<td>78.3</td>
<td>98.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General Data Group</th>
<th>Min</th>
<th>ML</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDGS</td>
<td>15</td>
<td>21.4</td>
<td>27.8</td>
</tr>
<tr>
<td>GDGM</td>
<td>32.4</td>
<td>46.3</td>
<td>60.2</td>
</tr>
<tr>
<td>GDGL</td>
<td>54.6</td>
<td>78.3</td>
<td>101.8</td>
</tr>
</tbody>
</table>

#### Level 4

<table>
<thead>
<tr>
<th>Macro Process</th>
<th>Min</th>
<th>ML</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPS</td>
<td>111.5</td>
<td>171.5</td>
<td>231.5</td>
</tr>
<tr>
<td>MPM</td>
<td>185.6</td>
<td>285.9</td>
<td>385.9</td>
</tr>
<tr>
<td>MPL</td>
<td>297.8</td>
<td>457.4</td>
<td>617.4</td>
</tr>
</tbody>
</table>

**Add-on – Simple Function Points – only two types.**

- Simple File
  - SFILE: 6, 7, 8
- Simple EP
  - SEP: 4, 4.6, 5

---

Function Based Sizing Early in the Life Cycle is most effective

Copyright 2018 Northrop Grumman Systems Corporation
Agile Estimation Use Cases
Example Early Estimations with Minimal Data
FP Early and Quick Function Point Example

Large architecture sized with Function Points and reuse plan.

Large SEER for Software model: Time phased the hours into a Mil-STD-881c Product Based WBS. Proposal team planned the Releases and Features in the Integrated Master Schedule, aligned with Agile Earned Value requirements.
The Case Study

- New contract, software intensive, legacy software existed
- Completion contract, fixed scope (Feature list provided by customer)

Phase 1

- Counted the legacy code and used SEER for Software to get to Product Based Structure (PBS) with Equivalent Source Lines of Code (ESLOC)
- SEER for Software provided effort months and labor category map
  - Placed time phased effort into a schedule

Phase 2 – After award, additional task orders were received

- Used SEER for Software with ESLOC estimates to get effort months
  - Used the list of desired Features to estimate ESLOC
- In parallel, developed Stories and Story Points against the desired list of Features and software changes
- Accepted the SEER for Software output and time phased the labor
Phase 3 used no more parametric tools

- The project estimated new work on their own
- Stories were created to do the actual estimation work
- Features were broken down into products
- Members from the various Agile teams were pulled together so all Agile teams understood the Roadmap and the effort needed
- Using Agile friendly contracting, Government customer had insight to the size per Feature and could move Features in and out of the contractual obligations
- If new work was contractually added and could not be accomplished with the existing set of Agile teams, new Agile teams were added
The Case Study

1) Pure Agile – no parametrics – Story Points of the work to be done
   - Used Velocity with same teams to discover end date

2) SEER for Software with ESLOC
   - Project estimated ESLOC and predicted both effort and schedule using SEER for Software

3) SEER for Software with historic ESLOC and historic Story Points
   - Captured a ratio of ESLOC per Story Point
   - Entered ratio into SEER for Software as a size proxy
   - Used team’s estimate of future Story Points to drive a new effective size and an effort and schedule estimate

Results

- Project determined that the proxy based estimate (3) was the closest to the actuals once the work was completed
Customer provided requirements in “Story” format. NGC estimated the function points.

SME’s Conducted GAP Analysis against a OTS Solution – Summarized Solution into Percentages (12% Custom, 27% OTS, 60% Configurations)

Tallied the FPs into the Features with the Reuse %s

Structured the SEER for Software model to match customer’s Capabilities, loaded the size, iterated the estimate.
Other Insights
## Testing a Tool to Count Function Points from User Stories

**LGAPPS MARINE tool**

<table>
<thead>
<tr>
<th>Re..No.</th>
<th>Requirement Text</th>
<th>Objects</th>
<th>Actions</th>
<th>Component</th>
<th>Functs</th>
<th>FP Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>As a CFT user, I want to receive a confirmation message asking if I want to create a new Record Set, if one or more do not exist when I enter valid information to identify an employee from the main page, in order to create new Record Sets if this is the first one being created for the employee.</td>
<td>a</td>
<td>VIEW</td>
<td>EQ, EI, ILF</td>
<td>3</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>As a CFT User, I want to be prompted to enter a new Record Setname, for the purpose of uniquely identifying this record set.</td>
<td>a new Record Set</td>
<td>CREATE</td>
<td>EI, ILF, EO</td>
<td>3</td>
<td>19</td>
</tr>
<tr>
<td>11</td>
<td>As a CFT user, I want to be able to create a new Record Set so that I can enter my adjustments information, save that information, and create a new adjustment.</td>
<td>a new Record Setname</td>
<td>ENTER</td>
<td>EI, EO</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>that information</td>
<td>SAVE</td>
<td>EI, ILF</td>
<td>2</td>
<td>14</td>
</tr>
</tbody>
</table>

**MARINE = Machine Assisted Requirements Inspection and Evaluation**

http://logapps.com/tools/marine/

Copyright 2018 Northrop Grumman Systems Corporation
Summary
1. Map product and architecture requirements to pre-existing solutions, including COTS
2. Find the gaps – estimate new and pre-existing size using SLOC or Function Points or parametric tool size alternative
3. Estimate the changes to the pre-existing solution
4. Map sizes into Capabilities or Features
5. Develop a Product Roadmap with Features and time based Releases
6. Load Feature based structure into the parametric estimation tool
7. Set Knowledge Bases (Agile) and parameters to match the planned reality of the new project
8. Match results against internal historical productivity and conduct a cross-check – e.g., Alternate Sizing, Bottoms-up Activity Based or Top Down
9. Make sure the Features in the schedule are achievable as compared to the parametric schedule predictions – iterate with Roadmap Time boxes
10. Perform risk and opportunity analysis
11. If needed, set margins (hardening Sprint) using parametric tool confidence levels
12. Time-Phase the hours into the Features - PBS aligned with the schedule

Agile Projects need sound estimation activities, building upon proven techniques
Q&A
THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN
HOW “BAD” ARE WE AT ESTIMATING SOFTWARE EFFORT?:
A SOFTWARE GROWTH STUDY
22 August 2018
Brittany Grissom
NGA SW Growth Study Background

Background: The NGA CAPE Cost Assessment Division wanted a data driven software growth factor, created in house, using timely and relevant development data.

Scope: Analysis focused on developing SW growth factors at the two levels NGA typically receives inputs for:
- Release/Computer Software Configuration Item (CSCI)
- Program Summary

Data: NGA analysis relies on DoD Software Resource Data Reports (SRDRs) because access to IC SW growth data is very limited.

Analysis aims to produce a rigorously developed & defensible SW growth factor.
Study Background: Analysis Overview

Dataset Overview
- 274 DoD SRDR CSCI records
- Investigate overall characteristics

Statistical Testing
- Investigate statistical significance and correlation among parameters
- Parametric and non-parametric tests

Analysis
- Regress and analyze results for both ESLOC and development hours
- Perform outlier analysis and analyze results

Results Summary
- Summarize selected SW growth factors
- Summarize usage recommendations

Further Investigation
- Address areas for further investigation
SRDR Dataset Characteristics
Dataset Characteristics: SRDR Dataset Overview

Dataset taken from the SRDR workbook managed by NAVAIR

- Determined that less than 20% of all SRDRs within the database are suitable data points (e.g., initial/final paired report, hours and code counts align, etc.)
- NGA’s research uses the October 2017 version of the database

The 274 CSCI records have been aggregated into two natural groupings for further analysis

- **Program**
  - 45 DoD programs
  - Program views created by summing ESLOC counts and SW development hours from all CSCIs in each program

- **Commodity**
  - 10 DoD commodity types
  - Commodity grouping used for statistical tests (no growth factor derived from this grouping)

<table>
<thead>
<tr>
<th>Commodity Type</th>
<th># of CSCIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft</td>
<td>84</td>
</tr>
<tr>
<td>BMDS</td>
<td>7</td>
</tr>
<tr>
<td>Electronic</td>
<td>65</td>
</tr>
<tr>
<td>Missile</td>
<td>6</td>
</tr>
<tr>
<td>Ordnance</td>
<td>7</td>
</tr>
<tr>
<td>Ship</td>
<td>17</td>
</tr>
<tr>
<td>Space</td>
<td>13</td>
</tr>
<tr>
<td>Surface Vehicle</td>
<td>4</td>
</tr>
<tr>
<td>System of Systems</td>
<td>45</td>
</tr>
<tr>
<td>UAV</td>
<td>26</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>274</strong></td>
</tr>
</tbody>
</table>

Approved for Public Release, 18-787
Several datasets can be formed from the SRDR records, as each contains multiple SW parameters.

The scope of this analysis covers two datasets:
- ESLOC counts
- Development hours counts

These two datasets are then analyzed at the CSCI level and the program level, creating four SW growth analysis cases:
- ESLOC growth at CSCI level
- Development hours growth at CSCI level
- ESLOC growth at program level
- Development hours growth at program level
Dataset Characteristics: SRDR Data Visualization and Depiction

Plotting estimated values against actual values of all 274 CSCI data points for both ESLOC and hours displays key characteristics of the data sets:

- Both data sets have very wide ranges
- A few data points appear to be potential outliers
ESLOC growth factors are not normally distributed

- The histogram below displays the skewed distribution of all 274 ESLOC growth factors (calculated using the reported estimated and actual ESLOC counts).
- The large difference between the mean and median growth values also highlights the skew of the data set.

Distribution of CSCI ESLOC Growth

**ESLOC Growth**
- Mean: 0.857
- Median: 0.312
- Mode: 0.000
- Std Dev: 2.130
- Minimum: -1.000
- Maximum: 17.818
- Count: 274
Dataset Characteristics: Hours Growth Distribution

Development hours are also not normally distributed

- The histogram below displays the skewed distribution of all 274 hours growth factors (calculated using the reported estimated and actual hour counts)
- The large difference between the mean and median growth values also highlights the skew of the data set

```
<table>
<thead>
<tr>
<th>Hours Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Median</td>
</tr>
<tr>
<td>Mode</td>
</tr>
<tr>
<td>Std Dev</td>
</tr>
<tr>
<td>Minimum</td>
</tr>
<tr>
<td>Maximum</td>
</tr>
<tr>
<td>Count</td>
</tr>
</tbody>
</table>
```
Analysis Part I: Statistical Tests
Growth Statistical Tests: Test Overview

Because the data sets are not normally distributed, a series of non-parametric tests were performed

- Non-parametric analyses are statistical methods used when the data is not required to fit a normal distribution
- Uses ranks or orders instead of values

Kruskal-Wallis Test – Statistical significance
- Uses ranks to test samples for statistical differences
- If no statistical differences, samples can be combined into one larger data set and used together

Spearman’s Rank – Correlation
- Uses the rankings of data to test for correlation
- Differs from standard Pearson’s correlation in that correlation is not limited to a linear relationship

Depending on the results, non-parametric analysis enables further parametric analysis
- Parametric correlation analysis (Pearson’s) was also performed to test strictly for linear relationships
Statistical Tests: Statistical Significance

Test to determine if growth factors from different commodity types are analogous to one another (e.g., are growth factors from ship systems analogous to factors from space systems?)

To confirm that all 274 CSCIs can be used as one data set, the non-parametric Kruskal-Wallis test was used

- Tested for statistical significance among growth factors of each commodity type
- Test employed for the 4 analysis cases: ESLOC and hours at CSCI and program level

In all four cases, no statistical differences were found – all data can be combined into one population
Statistical Tests: Correlation

Spearman’s Rank identifies non-parametric correlation between two paired variables

► Similar to more commonly used Pearson’s product moment which only measures linear correlation
► Uses ranks to determine correlation values and is therefore a robust way to identify potential outliers

Pearson’s correlation matrix also developed to determine possible linear relationships

Based on these correlation results, both linear and nonlinear regressions were performed
Analysis Part II: Regression Analysis
Regression Analysis: Process Overview

Three base regressions: linear with intercept, linear without intercept, and log linear (power curve)

► Initial ESLOC and initial hours used as independent variables to predict final ESLOC and final hours, respectively
► Regressed at the CSCI and program level

Analysis of potential outliers

► Identification of influential observations
► Evaluation of level of influence on regression equations
► Evaluation of value in removing influential observations

Analysis of results

► Comparison of regression statistics ($R^2$, standard errors, etc.)
  • $R^2$ for nonlinear models have been shown to be artificial and inaccurate stats
    Analysis charts will show $R^2$ for power curves, but caution is advised
  • Standard errors are a main factor in deciding the best-fit regression
► Sanity checks on regression equations (evaluation of intercepts, predicted values, etc.)
Regression Analysis: Process Overview – Outlier Analysis Details

Identification of influential data points

- Data points need to be evaluated for leverage against the three components of regressions
  - Independent value (points more than 2 standard deviations away from the mean of x)
  - Dependent value (points more than 2 standard deviations away from the mean of y)
  - Regression line (points more than 2 standard errors away from the regression line)

Evaluation of level of influence on regression equations

- Data points proven to be influential against any of the three components need to be evaluated for level of impact
- Regression is then run again without each individual influential data point to determine the effect on the resulting equation
- Once the point is removed, the resulting slope coefficient in the equation is investigated for significant changes from the original slope coefficient

Evaluation of value in removing influential observations

- Without programmatic information to explain potential outliers, need strong statistical justification to exclude data points
Analysis Part II: Regression Analysis
ESLOC Growth Analysis
Regression Analysis: **CSCI** Level ESLOC Results Overview

Data set does not produce strong regressions

- Linear without intercept is the most acceptable of all three
- No recommendation for code growth factor

### Overview of Three Base Regressions

<table>
<thead>
<tr>
<th>Variable</th>
<th>Regression Type</th>
<th>Equation</th>
<th>Adjusted $R^2$</th>
<th>SE (unit space)</th>
<th>CV (unit space)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESLOC</td>
<td>Linear with Intercept</td>
<td>$\text{ESLOC}<em>{\text{Act}} = 10,870 + 1.187 \times \text{ESLOC}</em>{\text{Est}}$</td>
<td>58.90%</td>
<td>75,226</td>
<td>100.86%</td>
</tr>
<tr>
<td></td>
<td>Linear without Intercept</td>
<td>$\text{ESLOC}<em>{\text{Act}} = 1.255 \times \text{ESLOC}</em>{\text{Est}}$</td>
<td>70.35%</td>
<td>75,612</td>
<td>101.37%</td>
</tr>
<tr>
<td></td>
<td>Power</td>
<td>$\text{ESLOC}<em>{\text{Act}} = 4.598 \times \text{ESLOC}</em>{\text{Est}}^{0.8761}$</td>
<td>74.46%</td>
<td>78,096</td>
<td>104.70%</td>
</tr>
</tbody>
</table>
Regression Analysis: CSCI Level ESLOC Outlier Analysis

Identification of influential data points
- All three base regressions have data points with leverage
  - Linear with intercept – 16 data points
  - Linear without intercept – 27 data points
  - Power – 21 data points

Evaluation of influence on regression equations
- Regressions re-run without each identified point
- Some points do effect the regression equation (changes in slopes);
  - Example shown reflects slope change from 1.187 to 1.117
- Minimal improvement on regression statistics

Evaluation of value in removing influential observations
- Again, the new regressions also contain data points with leverage
- No programmatic information to explain potential outliers

Example Comparison of Two Linear Regressions

\[ y = 1.1872x + 10870 \]
\[ R^2 = 0.5906 \]
\[ y = 1.1174x + 12609 \]
\[ R^2 = 0.6059 \]

After substantial outlier analysis, ultimately no data points were removed from the original data set.
Graphical Depiction of Prediction Error of Both Linear Regressions for Initial ESLOC Less than 30K

With intercept mostly overestimates

Without intercept is more balanced, but favors underestimating

Approved for Public Release, 18-787
Graphical Depiction of Prediction Error of Both Linear Regressions for Initial ESLOC Greater than 30K

Both estimate much more similarly for this block of data
Regression Analysis: **CSCI** Level ESLOC Analysis of Results Continued

Predicted growth factors using regressions and SRDR estimated ESLOC data points

- **Linear with Intercept**
  - Yields factors ranging from 21% to 14,136%
  - Average factor is 219%
  - Growth percentage decreases as the ESLOC estimates increase
    - For small estimates, the intercept is a large percentage of growth
    - Begins to level off near 20% for large ESLOC estimates

- **Linear without Intercept**
  - Growth always 25.5%

- **Power**
  - Yields factors ranging from -8% to 168%
  - Average factor is 35.7%
  - Growth decreases as ESLOC estimates increase

<table>
<thead>
<tr>
<th>Variable</th>
<th>Regression Type</th>
<th>Equation</th>
<th>Adjusted $R^2$</th>
<th>SE (unit space)</th>
<th>CV (unit space)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESLOC</td>
<td>Linear with Intercept</td>
<td>$ESLO\text{C}<em>{Act} = 10,870 + 1.187 \cdot ESLO\text{C}</em>{Est}$</td>
<td>58.90%</td>
<td>75,226</td>
<td>100.86%</td>
</tr>
<tr>
<td>ESLOC</td>
<td>Linear without Intercept</td>
<td>$ESLO\text{C}<em>{Act} = 1.255 \cdot ESLO\text{C}</em>{Est}$</td>
<td>70.35%</td>
<td>75,612</td>
<td>101.37%</td>
</tr>
<tr>
<td>ESLOC</td>
<td>Power</td>
<td>$ESLO\text{C}<em>{Act} = 4.598 \cdot ESLO\text{C}</em>{Est}^{0.8761}$</td>
<td>74.46%</td>
<td>78,096</td>
<td>104.70%</td>
</tr>
</tbody>
</table>

Considering all statistics and factors, no recommendation of these regressions can be made.
Regression Analysis: Program Level ESLOC Results Overview

Recommended regression: Linear without intercept

- Lowest SE, highest adjusted $R^2$
- Linear regressions nearly equivalent
- Power curve nearly 1

### Relevant Range

<table>
<thead>
<tr>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>8,585</td>
<td>2,700,903</td>
</tr>
</tbody>
</table>

### Total Data Points

45

### Overview of Three Base Regressions

<table>
<thead>
<tr>
<th>Variable</th>
<th>Regression Type</th>
<th>Equation</th>
<th>Adjusted $R^2$</th>
<th>SE (unit space)</th>
<th>CV (unit space)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESLOC</td>
<td>Linear with Intercept</td>
<td>$ESLOC_{\text{Act}} = -108.5 + 1.387 \times ESLOC_{\text{Est}}$</td>
<td>95.17%</td>
<td>145,113</td>
<td>32.07%</td>
</tr>
<tr>
<td>ESLOC</td>
<td>Linear without Intercept</td>
<td>$ESLOC_{\text{Act}} = 1.387 \times ESLOC_{\text{Est}}$</td>
<td>96.74%</td>
<td>143,455</td>
<td>31.70%</td>
</tr>
<tr>
<td>ESLOC</td>
<td>Power</td>
<td>$ESLOC_{\text{Act}} = 2.066 \times ESLOC_{\text{Est}}^{0.9656}$</td>
<td>88.95%</td>
<td>161,558</td>
<td>35.70%</td>
</tr>
</tbody>
</table>
Analysis Part II: Regression Analysis
Development Hours Growth Analysis
Regression Analysis: **CSCI** Level Hours Results Overview

Recommended regression: Linear without intercept

- Both linear regressions are very similar, but the equation without an intercept has the better $R^2$
- SE’s are nearly the same

<table>
<thead>
<tr>
<th>Variable</th>
<th>Regression Type</th>
<th>Equation</th>
<th>Adjusted $R^2$</th>
<th>SE (unit space)</th>
<th>CV (unit space)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours</td>
<td>Linear with Intercept</td>
<td>$\text{Hours}<em>{\text{Act}} = 6.636 + 1.238 \times \text{Hours}</em>{\text{Est}}$</td>
<td>77.93%</td>
<td>44,230</td>
<td>72.22%</td>
</tr>
<tr>
<td></td>
<td>Linear without Intercept</td>
<td>$\text{Hours}<em>{\text{Act}} = 1.283 \times \text{Hours}</em>{\text{Est}}$</td>
<td>84.27%</td>
<td>44,497</td>
<td>72.65%</td>
</tr>
<tr>
<td>Power</td>
<td></td>
<td>$\text{Hours}<em>{\text{Act}} = 5.393 \times \text{Hours}</em>{\text{Est}}^{0.8633}$</td>
<td>76.64%</td>
<td>49,838</td>
<td>81.37%</td>
</tr>
</tbody>
</table>

**Relevant Range**

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Hours</td>
<td>320</td>
<td>578,567</td>
</tr>
</tbody>
</table>

**Total Data Points**

274

---

Approved for Public Release, 18-787
Regression Analysis: **Program** Level Hours Results Overview

Recommended regression: Linear without intercept

- Lowest SE, highest adjusted $R^2$
- Linear regressions nearly equivalent
- Power curve nearly 1

<table>
<thead>
<tr>
<th>Relevant Range</th>
<th>Total Data Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>18,500</td>
<td>2,054,088</td>
</tr>
<tr>
<td>45</td>
<td></td>
</tr>
</tbody>
</table>

Overview of Three Base Regressions

<table>
<thead>
<tr>
<th>Variable</th>
<th>Regression Type</th>
<th>Equation</th>
<th>Adjusted $R^2$</th>
<th>SE (unit space)</th>
<th>CV (unit space)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours</td>
<td>Linear with Intercept</td>
<td>$\text{Hours}<em>{\text{Act}} = 1.947 + 1.381 \times \text{Hours}</em>{\text{Est}}$</td>
<td>96.19%</td>
<td>97,503</td>
<td>26.15%</td>
</tr>
<tr>
<td></td>
<td>Linear without Intercept</td>
<td>$\text{Hours}<em>{\text{Act}} = 1.384 \times \text{Hours}</em>{\text{Est}}$</td>
<td>97.57%</td>
<td>96,402</td>
<td>25.85%</td>
</tr>
<tr>
<td></td>
<td>Power</td>
<td>$\text{Hours}<em>{\text{Act}} = 1.536 \times \left(\text{Hours}</em>{\text{Est}}^{0.9902}\right)$</td>
<td>88.81%</td>
<td>100,426</td>
<td>26.93%</td>
</tr>
</tbody>
</table>
Summary of Results

<table>
<thead>
<tr>
<th>Variable</th>
<th>Level</th>
<th>Regression Type</th>
<th>Factor</th>
<th>Relevant Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESLOC</td>
<td>CSCI</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
</tr>
<tr>
<td></td>
<td>Program</td>
<td>Linear without Intercept</td>
<td>1.387</td>
<td>8,585 - 2,700,903</td>
</tr>
<tr>
<td>Hours</td>
<td>CSCI</td>
<td>Linear without Intercept</td>
<td>1.283</td>
<td>320 - 578,567</td>
</tr>
<tr>
<td></td>
<td>Program</td>
<td>Linear without Intercept</td>
<td>1.384</td>
<td>18,500 - 2,054,088</td>
</tr>
</tbody>
</table>
Further Investigations and Questions
Are “Outliers” Making Our Regressions?

\[ y = 1.2378x + 6636.1 \]
\[ R^2 = 0.7801 \]

\[ y = 1.1872x + 10870 \]
\[ R^2 = 0.5906 \]
Regressing Initial Data Points Less than 20K

\[ y = 1.2733x + 3729.9 \]

\[ R^2 = 0.3294 \]

\[ y = 1.1358x + 6532.6 \]

\[ R^2 = 0.1186 \]
Zooming In…

CSCI ESLOC < 20K

\[ y = 1.1358x + 6532.6 \]
\[ R^2 = 0.1186 \]

CSCI Hrs < 20K

\[ y = 1.2733x + 3729.9 \]
\[ R^2 = 0.3294 \]
Are “Outliers” Making Our Regressions?

**CSCI ESLOC ALL**

$$y = 1.1872x + 10870$$  \[ R^2 = 0.5906 \]

**CSCI Hrs ALL**

$$y = 1.2378x + 6636.1$$  \[ R^2 = 0.7801 \]
Graphical Depiction of Prediction Error of Both Linear Regressions for Initial ESLOC Less than 30K

With intercept mostly overestimates

Without intercept is more balanced, but favors underestimating
Back Up
Regression Analysis: **CSCI** Level ESLOC Analysis of Results

With no data points deemed statistically significant enough to remove, analysis returns to original three base regressions.

Comparison of regression statistics

- **R2**
  - All regressions have a low adjusted R2
  - Linear without intercept line has the highest

- **SE and CV**
  - Minimal difference between either linear SEs
  - All three regressions have an extremely high CV; predicted values are far from the actual values

Sanity checks on equations and intercepts

- Intercept in the first linear regression is very large; leads to large predicted ESLOC values for small estimated ESLOC points
- Linear without intercept slope coefficient yields a growth factor lower than expected
  - Mean growth factor of data set is 1.86 and median is 1.31
- Power model exponent is close to 1 – results in only a slight power curve

Considering all statistics and factors, no recommendation of these regressions can be made
Regression Analysis: CSCI Level ESLOC using Program Level Factor

Without a strong regression equation from the CSCI data, the program level factor of 1.385 was evaluated for viability

- Standard error: 79,938
  - Not significantly higher than the lowest SE of 76,663 from the ‘linear with intercept’ equation
  - Falls between median (1.28) and mean (1.84) of CSCI growth factors – more reasonable factor than the three base regressions

Some values of the CSCI data are out of the relevant range of the program factor

- Program level range begins near 17,000 ESLOC, while CSCIs begin near 255
- 81 CSCI data points fall below 17,000 estimated ESLOC – 38% of the data set
- Risk of applying factor outside its relevant range is accepted
  - CSCI data is available to depict how the data behaves between 255 and 17,000 ESLOC
  - A factor of 1.386 for these points is as suitable as the factors produced through regression

Program factor includes more risk than ‘linear without intercept’ factor or median

- Analysis shows estimates for ESLOC contain large amounts of variation (66% of data points were underestimated)
- Program factor will estimate a higher value final ESLOC than the median and ‘linear without intercept’ factors

For ESLOC at release level, recommend applying program level factor of 1.385
Cost Optimization on AWS

Marc Johnson
Benjamin Kleintank
Roman Rusal
Morteza Zijerdi

Cloud Economics, AWS
The AWS Difference
What sets AWS apart?

- **Experience**: Building and managing cloud since 2006

- **Service Breadth & Depth**: 90+ services to support any cloud workload

- **Global Footprint**: 18 regions, 54 availability zones, 114 edge locations

- **Pricing Philosophy**: 65 proactive price reductions to date (as of 03/2018)

- **Ecosystem**: 10,000’s of partners; 4,000+ Marketplace products
Why choose AWS?

**Traditional Infrastructure**
- Equipment
- Resources and Administration
- Contracts
- Cost

**AWS Cloud**
- No Up Front Expense
- Pay for what you Use
- Improve Time to Market & Agility
- Scale Up and Down
- Self-Service Infrastructure
Total Cost of Ownership
What is TCO?

- **Comparative Total Cost of Ownership analysis**
  - (acquisition and operating costs)
  - for running an infrastructure environment
  - end-to-end on-premises vs. AWS.
Cloud Value Framework

Cost Savings (TCO)

- Infrastructure cost savings/avoidance from moving to the cloud
- 50%+ reduction in TCO (GE)

Staff Productivity

- Efficiency improvement by function on a task-by-task basis
- Over 500 hours per year of server configuration time saved (Sage)

Operational Resilience

- Benefit of improving SLAs and reducing unplanned outages
- Critical workloads run in multiple AZs and Regions for robust DR (Expedia)

Business Agility

- Deploying new features/applications faster and reducing errors
- Launch of new products 75% faster (Unilever)

What is it?

Examples

Typical Focus

Most Compelling Cloud Benefits
## Typical Focus

**Cost Savings (TCO)**

<table>
<thead>
<tr>
<th>Cost</th>
<th>Savings</th>
<th>Impr</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>$1,093,514</td>
<td>30%</td>
<td>$405,000</td>
</tr>
<tr>
<td>Storage</td>
<td>$688,514</td>
<td>80%</td>
<td>$270,000</td>
</tr>
<tr>
<td>Network</td>
<td>$154,640</td>
<td>20%</td>
<td>$337,500</td>
</tr>
</tbody>
</table>

**TOTAL**  

$1,936,668  

$1,012,500

**TOTAL ANNUAL BENEFIT**  

$8,731,433

## Most Compelling Cloud Benefits

### Staff Productivity

<table>
<thead>
<tr>
<th>Cost</th>
<th>Impr</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>30%</td>
<td>$405,000</td>
</tr>
<tr>
<td>Storage</td>
<td>80%</td>
<td>$270,000</td>
</tr>
<tr>
<td>Network</td>
<td>20%</td>
<td>$337,500</td>
</tr>
</tbody>
</table>

**TOTAL**  

$1,432,343

### Operational Resilience

<table>
<thead>
<tr>
<th>Uptime</th>
<th>Before</th>
<th>After</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>eCommerce 1</td>
<td>99.00%</td>
<td>99.50%</td>
<td>$1,432,343</td>
</tr>
</tbody>
</table>

### Business Agility

<table>
<thead>
<tr>
<th>KPI</th>
<th>Impr</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to Deploy</td>
<td>22.0%</td>
<td>$656,232</td>
</tr>
<tr>
<td>Total Defects</td>
<td>35.0%</td>
<td>$640,360</td>
</tr>
<tr>
<td>Customer NPS</td>
<td>50.0%</td>
<td>$1,302,274</td>
</tr>
<tr>
<td>Employee NPS</td>
<td>33.3%</td>
<td>$1,751,055</td>
</tr>
</tbody>
</table>

**TOTAL ANNUAL BENEFIT**  

$8,731,433
Examples of Value Realized

**COST SAVINGS**

Cost savings of $20M p.a. (FINRA)

CIA used to build to max peak, only utilizing 12% of their infrastructure, now scale on demand as needed in seconds.

- DC footprint from 13 to 6 (The Weather Co.)
- DC footprint reduced from 8 to 3 by 2018 (CapitalOne)
- Over 50% reduction in TCO (GE)
- DC footprint from 45 to 6 (News Corp)
- 50% reduction in app costs (Time Inc.)

- Computational cost reduced by 20%+ (ENEL)
- Cloud deployment has saved US$34 million in CAPEX and reduced OPEX by 85% (Samsung)
- Cost reduction of $40k p.a. (Dow Jones)

**STAFF PRODUCTIVITY**

CIA – in ’17, post transition to C2S over 4000 developers were all writing code in an agile simultaneous environment in the cloud. Deploying live (within seconds Vs months).

- Average annual staffing savings of $3m (Adroll)
- Energy Marketing business prepared for acquisition in only 6 months rather than 12 (Hess Corp)
- Half the infrastructure team required to manage infrastructure (2C2P)
- Performance targets over-achieved by 43–66% (McDonalds)
- IT Infra consolidation completed in 20% of expected time (Hearst)
- 60% of IT working on data proliferation, lack of standards, security hardening all of which AWS is addressing. (Intuit)
- Over 500 hours per year of server configuration time saved (Sage)
- 39 years of Computational chemistry condensed into 9 hours (Novartis)

**OPERATIONAL RESILIENCE**

Processing over 75 billion market events daily (FINRA)

Per CIO of CIA: “we are now in the most invincible environment possible, with security measures in place that wouldn’t have been possible in an on-prem environment, or without AWS”.

- Scaled to handle a 400% increase in page views (Kurt Geiger)
- Improved security posture (CapitalOne)
- 8600 transactions/second (McDonalds)
- Transfer of over 750 TB of data from pipeline inspection machinery (GE)
- Critical applications run in multiple AZs, x-Regions for robust disaster recovery (Expedia)
- Supports over 300,000 requests per minute to its API (Easy Taxi)
- 60% reduced downtime (Trainline)
- Migration of SAP on Oracle to AWS with zero unplanned downtime across five countries (Kellogg’s)

**BUSINESS AGILITY**

CIA – Applications selection cut down from a lead time of 9 months to minutes (Marketplace and Partner ecosphere), Due to AI/ML, searches that used to take weeks, are now done in seconds, Cloud usage growth is at 208% YOY.

- 80% reduction in software R&D times (Apeejay Stya & Srivan)
- Time to launch digital campaigns cut from weeks to 24 hours (91App)
- Calc and reporting time cut from 10 days to 10 minutes (Aon Benfield)
- Time to market cut from weeks to hours (FlyDubai)
- Clinical simulations 98% faster than on-premise (Bristol-Myers Squibb)
- Time to deploy IT compute reduced to <5 minutes (Alcatel-Lucent)
- R&D RFS times reduced from 6 months to 1 day (NewsCorp)
- Provisioning time cut from 3–4 weeks to 2 days (ENEL)
- Test-run time cut to 10 minutes, from up to 2 hours (Yelp)
TCO for On-premises v AWS

Traditional Data Centre & Co-Location ≠ aws

Comparing TCO isn’t easy
## Typical TCO Considerations

<table>
<thead>
<tr>
<th></th>
<th>Server Costs</th>
<th>Storage Costs</th>
<th>Network Costs</th>
<th>IT Labor Costs</th>
<th>Extras</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Server Costs</strong>&lt;br&gt;Hardware – Server, Rack Chassis PDUs, ToR Switches (+Maintenance)&lt;br&gt;Software - OS, Virtualization Licenses (+Maintenance)</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Server Admin, Virtualization Admin, Storage Admin, Network Admin, Support Team</strong></td>
<td><strong>Project planning, Advisors, Legal, Contractors, Managed Services, Training, Cost of capital</strong></td>
</tr>
<tr>
<td>2</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Server Admin, Virtualization Admin, Storage Admin, Network Admin, Support Team</strong></td>
<td><strong>Project planning, Advisors, Legal, Contractors, Managed Services, Training, Cost of capital</strong></td>
</tr>
<tr>
<td>3</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Server Admin, Virtualization Admin, Storage Admin, Network Admin, Support Team</strong></td>
<td><strong>Project planning, Advisors, Legal, Contractors, Managed Services, Training, Cost of capital</strong></td>
</tr>
<tr>
<td>4</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Server Admin, Virtualization Admin, Storage Admin, Network Admin, Support Team</strong></td>
<td><strong>Project planning, Advisors, Legal, Contractors, Managed Services, Training, Cost of capital</strong></td>
</tr>
<tr>
<td>5</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Facilities Cost</strong>&lt;br&gt;Space</td>
<td><strong>Server Admin, Virtualization Admin, Storage Admin, Network Admin, Support Team</strong></td>
<td><strong>Project planning, Advisors, Legal, Contractors, Managed Services, Training, Cost of capital</strong></td>
</tr>
</tbody>
</table>

Diagram doesn’t include every cost item. E.g. software costs can include database, management, middle tier software costs. Facilities cost can include costs associated with upgrades, maintenance, building security, taxes etc. IT labor costs can include security admin and application admin costs.
What’s included in a typical TCO?

<table>
<thead>
<tr>
<th>Economic Criteria</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure Comparison</td>
<td>✔️</td>
</tr>
<tr>
<td>Capacity Planning Benefits</td>
<td>❌</td>
</tr>
<tr>
<td>Financial Benefits of Innovation</td>
<td>❌</td>
</tr>
<tr>
<td>Cost Avoidance</td>
<td>✔️</td>
</tr>
<tr>
<td>Workforce Productivity</td>
<td>✔️</td>
</tr>
<tr>
<td>Accelerated Time To Value/Market</td>
<td>❌</td>
</tr>
<tr>
<td>Cost to Achieve (Migration, Platform, Training)</td>
<td>❌</td>
</tr>
<tr>
<td>Legacy Constraints</td>
<td>❌</td>
</tr>
</tbody>
</table>

- ✔️ Included
- ✔️ Partially Included
- ❌ Not Included

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
A typical on-premises compute environment is massively underutilized.

Studies by Gartner, McKinsey and the Uptime Institute have stated that typical data centers are on average less than 50% utilized.

www.uptimeinstitute.org


Why is on-premises so under-utilised?

Part of this can be explained by buying for “peak load” requirements with inflexible infrastructure.

- **Fluctuating/“Spiky”**
- **Part-time**
- **Cyclical**
Why is on-premises built for peak?

- Unused Capacity = Wasted $
- Downtime, Lost Customers, Lost Revenue (Impossible to measure)
- More Wasted $
- Increased Capacity, again
- New purchase after "Don't Let it happen again"
- Yet More Wasted $
Initial questions to consider when exploring TCO

1. **Capacity Planning**
   - How do you **plan** for capacity?
   - How many servers have you added in the past year? Anticipating next year?
   - Can you switch your hardware **on and off** and only pay for what is used?

2. **Utilization**
   - What is your average **server utilization**?
   - How much do you **overprovision** for peak load?

3. **Operations**
   - Will you run out of data center space some time in the future?
   - What was your last year power utility bill for the Data Center(s)?
   - Have you budgeted for both **average** and **peak power** requirements?

4. **Optimization**
   - Are you on AWS today?
   - Are you **cost-optimized** (Auto Scaling, RIs, Spot, Instances turn on/off)?
Addressing TCO in AWS
How do customers lower their TCO with AWS?

1. Remove over provisioning and move to a “pay for what you use” model
2. Economies of scale allow AWS to continually lower costs
3. Pricing model choice to support variable & stable workloads
4. Save more money as you grow bigger

“Customers will have spent 63.4% more on average on-prem or in co-location”

Traditional approaches to capacity management

Build to peak load

Build to average load
Lower over-provisioning via elasticity

Auto Scaling allows you to:
• React dynamically to changes in load
• Schedule regular workloads
• Optimise your instance usage
• Reduce over-provisioning
• Free service!
AWS Economies of Scale

Reduced Prices

More Customers

More AWS Usage

More Kit

Economies of Scale

Lower Costs

Continually lowering prices for customers is in our DNA

Infrastructure Innovation

Ecosystem
Global Footprint
New Features & Services

We pass the savings along to our customers in the form of low prices and continuous reductions

(65 reductions to-date)
Amazon EC2 Pricing Models

On-demand

Reserved

Spot
When to use Reserved Instances?
Amazon EC2 Reserved Instances

Up to 75%+ savings* (and capacity reservation)

Commitment level
1 year
3 year

AWS services offering RIs
Amazon EC2
Amazon RDS
Amazon DynamoDB
Amazon Redshift
Amazon ElastiCache

* Dependent on specific AWS service, size/type, and region
Amazon EC2 Spot Instances

Allow you to bid on spare Amazon EC2 computing capacity for **up to 90% off** the normal On-Demand price.

Applications that have flexible start and end times

Applications that are only feasible at very low compute prices

Users with urgent computing needs for large amounts of additional capacity
With spot, the rules are simple

Markets where the price of compute changes based on supply and demand

You’ll never pay more than your bid. When the market exceeds your bid you get 2 minutes to wrap up your work
Volume Tiered Pricing Discounts

First 50 TB per month

0.024 GB/month

Next 450 TB per month

0.023 GB/month

Over 500 TB per month

0.022 GB/month
Cost Optimization
Modelling Cost Optimization

Traditional TCO Comparisons

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Five Pillars of Cost Optimization

1. Right-Sizing Your Instances
2. Increase Elasticity
3. Pick the Right Pricing Model
4. Match usage to storage class
5. Measuring & Monitoring
**Right-sizing instances**

Selecting the cheapest instance available while meeting performance requirements

Look at CPU, RAM, storage, and network Utilization to identify potential instances that can be downsized

Leveraging Amazon CloudWatch metrics and setting up custom RAM metrics

**Rule of thumb:** Right size, then reserve. (But if you’re in a pinch, reserve first.)
Right-sizing & elasticity to reduce cost

More smaller instances vs. fewer larger instances

- 29 m5.large @ $0.111 /hr
  $2,349.87 / mo*

- 59 t2.medium @ $0.052/hr
  $2,239.64 / mo*

*Assumes Linux instances in the EU (London) Region at 730 hours per month
Workload Scheduling

% Running Time

- 100.0 for 24 x 7
- 71.4 for 24 x 5
- 35.7 for 12 x 5
- 29.8 for 10 x 5

Up to 70% savings for non-production workloads
AWS Instance Scheduler

- AWS-provided solution
- Custom start & stop schedules
- Works with EC2 & RDS instances
- Deploy using CloudFormation
- Selectively tag instances to schedule
- Multiple schedules per instance
- 5-minute granularity

Match usage to storage classes

Amazon S3
Designed to store and access any type of data over the Internet

Amazon Glacier
Low-cost and highly durable storage service for long-term backup and archive of any type of data

Amazon Elastic Block Storage
Block-level storage that serves as a virtual hard drive for your Amazon EC2 instance

AWS Storage Gateway
Seamlessly links your on-premises environment to Amazon cloud storage

Amazon Elastic File System
Simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud

Amazon CloudFront
Amazon CloudFront is a global content delivery network (CDN) service
Serverless Architecture

- **No Server Management**
- **Flexible Scaling**
- **High Availability**
- **No Idle Capacity**
Amazon CloudWatch

Monitor AWS Resources
Set Alarms
Monitor Custom Metrics

View Graphs and Statistics
Monitor and React to Resource Changes
Metrics & Targets

Set up metrics to define success and track progress

- % Instances turned off daily
- % of Instances right-sized
- % Always-on Resources covered by RIs
- % RI utilization

What KPI makes sense for this workload?
AWS Trusted Advisor

As an AWS customer, you want the most value from your investment. Trusted Advisor can help.

Trusted Advisor scans your AWS infrastructure and provides recommended actions.

...to help secure and optimize your infrastructure, and save money.

RDS Idle DB Instances
- 11 of 15 RDS DB instances appear to be idle.
- Recommended Action: Annual savings of up to $2,738 are available by minimizing the idle RDS DB instances.

Service Limits
- 0 of 24 service limits are over 80% capacity.

RDS Security Group Access Risk
- 9 of 26 RDS Security Group rules create potential security vulnerabilities.
- Recommended Action: Configure Security Groups to restrict access to only specified users or computers.

Amazon EC2 Availability Zone Balance
- 0 regions have an imbalanced instance distribution across Availability Zones.

Recommendations include links to take direct action...
AWS Cost Explorer
Cost Conscious Design
Cost Conscious Design

Example: Should I use Amazon S3 or Amazon DynamoDB?

https://calculator.s3.amazonaws.com/index.html
Amazon S3 or Amazon DynamoDB

<table>
<thead>
<tr>
<th>Request rate (Writes/sec)</th>
<th>Object size (Bytes)</th>
<th>Total size (GB/month)</th>
<th>Objects per month</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>2,048</td>
<td>1,483</td>
<td>777,600,000</td>
</tr>
</tbody>
</table>
Amazon S3 or Amazon DynamoDB

### Table

<table>
<thead>
<tr>
<th>Request rate (Writes/sec)</th>
<th>Object size (Bytes)</th>
<th>Total size (GB/month)</th>
<th>Objects per month</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>2,048</td>
<td>1,483</td>
<td>777,600,000</td>
</tr>
</tbody>
</table>

### DynamoDB Details

**Indexed Data Storage:**
- Dataset Size: 1,483 GB

**Provisioned Throughput Capacity:**
- Item Size (All attributes): 2 KB
- Number of items read per second: 0 Reads/Second
- Read Consistency: Strongly Consistent
- Number of items written per second: 300 Writes/Second

### S3 Details

**Amazon S3 Service (US-East):**
- Provisioned Throughput Capacity: $261.69
- Indexed Data Storage: $382.61
- Put/List Requests: $3,888.00

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Amazon S3 or Amazon DynamoDB

“…but what happens if I change the object size to 32 KB?”

<table>
<thead>
<tr>
<th>Request rate (Writes/sec)</th>
<th>Object size (Bytes)</th>
<th>Total size (GB/month)</th>
<th>Objects per month</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>32,768</td>
<td>23,730</td>
<td>777,600,000</td>
</tr>
</tbody>
</table>
### Amazon S3 or Amazon DynamoDB

<table>
<thead>
<tr>
<th>Request rate (Writes/sec)</th>
<th>Object size (Bytes)</th>
<th>Total size (GB/month)</th>
<th>Objects per month</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>32,768</td>
<td>23,730</td>
<td>777,600,000</td>
</tr>
</tbody>
</table>

**Indexed Data Storage:**
- Dataset Size: 23,730 GB

**Provisioned Throughput Capacity:**
- Item Size (All attributes): 32 KB
- Number of items read per second: 0 Reads/Second
- Read Consistency: Strongly Consistent
- Number of items written per second: 300 Writes/Second

**Amazon S3 is storage for the Internet. It is designed to make web-scale data accessible anywhere in the world in seconds.**

**Standard Storage:**
- PUT/COPY/POST Requests: 7,776,000 Requests
- GET and Other Requests: 0 Requests

### Amazon DynamoDB Service (US East (N. Virginia))
- Provisioned Throughput Capacity: $4,555.79
- Indexed Data Storage: $5,944.36
- Total: $10,500.15

### Amazon S3 Service (US East (N. Virginia))
- Standard Storage: $545.79
- Standard Put Requests: $388.80
- Total: $4,433.79

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
## Amazon S3 or Amazon DynamoDB

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Request rate (Writes/sec)</th>
<th>Object size (Bytes)</th>
<th>Total size (GB/month)</th>
<th>Objects per month</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scenario 1</strong></td>
<td>300</td>
<td>2,048</td>
<td>1,483</td>
<td>777,600,000</td>
</tr>
<tr>
<td><strong>Scenario 2</strong></td>
<td>300</td>
<td>32,768</td>
<td>23,730</td>
<td>777,600,000</td>
</tr>
</tbody>
</table>

**Use**

### Amazon S3 Service (US-East)

- Storage: $44.27
- Put/List Requests: $3888.00

### Amazon DynamoDB Service (US-East)

- Provisioned Throughput Capacity: $261.69
- Indexed Data Storage: $32.61
- DynamoDB Streams: $0.00

**Use**

### Amazon S3 Service (US East (N. Virginia))

- Standard Storage: $545.79
- Standard Put Requests: $3888.00

### Amazon DynamoDB Service (US East (N. Virginia))

- Provisioned Throughput Capacity: $4555.79
- Indexed Data Storage: $5944.36

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
What Next?
What benefits do I get by moving to AWS?

- Trade capital expense for variable expense
- Benefit from massive economies of scale
- Stop guessing capacity
- Increase speed and agility, reduce time-to-market
- Focus money on product development, not data centres
- Go global in minutes
And for existing customers: a call to action!

- How many instances could I right-size?
- What benefits could I get from using reserved instances?
- How many of my instances need to be running 24x7?
- How many instances are configured for auto-scaling?
Useful Resources

- AWS Pricing
  - https://aws.amazon.com/pricing/
- Online TCO Calculator:
  - https://awstcocalculator.com
- AWS Cloud Economics Center:
  - https://aws.amazon.com/economics/
Thank you!