# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title Page</td>
<td>1</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>2</td>
</tr>
<tr>
<td>Welcome</td>
<td>3</td>
</tr>
<tr>
<td>Agenda</td>
<td>4</td>
</tr>
<tr>
<td>Keynote Biographies</td>
<td>12</td>
</tr>
<tr>
<td>Speaker Biographies</td>
<td>14</td>
</tr>
<tr>
<td>Presentations</td>
<td></td>
</tr>
<tr>
<td><strong>August 20, 2019</strong></td>
<td></td>
</tr>
<tr>
<td>1) Tradecraft Evolution for Emerging Needs; Jen Rose (National Reconnaissance Office)</td>
<td>Pending Approval</td>
</tr>
<tr>
<td>2) Living in the Clouds; Michael Cassidy (Department of Justice)</td>
<td>19</td>
</tr>
<tr>
<td>3) Forecasting Future Amazon Web Services Pricing; James Smirnoff and Hassan Souiri (NRO Cost and Acquisition Assessment Group (CAAG))</td>
<td>32</td>
</tr>
<tr>
<td>4) Causality and Uncertainty: A New Wave for Cost Estimation; Robert Stoddard, Mike Konrad (Software Engineering Institute)</td>
<td>63</td>
</tr>
<tr>
<td>5) Better Data Communication; Jon Schwabish (The Urban Institute)</td>
<td>Pending Approval</td>
</tr>
<tr>
<td>6) Core Principles of Data Visualization and Presentation Skills; Jonathan Schwabish (PolicyViz)</td>
<td>91</td>
</tr>
<tr>
<td>7) Data Exploitation – More Value, Less Time; Adam H. James (Technomics, Inc.)</td>
<td>92</td>
</tr>
<tr>
<td>8) Calibrating COCOMO® II for Functional Size Metrics; Anandi Hira, Brad Clark and Barry Boehm (University of Southern California)</td>
<td>121</td>
</tr>
<tr>
<td>9) Cost of Software Obsolescence Resolution of Real-Time Software; Sanathanan Rajagopal (QinetiQ)</td>
<td>129</td>
</tr>
<tr>
<td>10) Army Software Sustainment Cost Estimating Results; Cheryl Jones and James Doswell (United States Army)</td>
<td>167</td>
</tr>
<tr>
<td>11) Agile Projects and GAO Best Practices; Jennifer Leotta (Government Accountability Office)</td>
<td>212</td>
</tr>
<tr>
<td><strong>August 21, 2019</strong></td>
<td></td>
</tr>
<tr>
<td>1) Defense Innovation Board, Public Meeting Presentation</td>
<td>241</td>
</tr>
<tr>
<td>2) Generating a Cost Estimate for Agile Software Development; Matthew R. Kennedy, PhD (United States Treasury)</td>
<td>265</td>
</tr>
<tr>
<td>3) Adaptive Acquisition Framework; Dr. Jeff Boleng (Special Assistant for Software Acquisition to the Under Secretary of Defense for Acquisition and Sustainment)</td>
<td>284</td>
</tr>
<tr>
<td>4) A Foundation for Software Acquisition Decisions; Tory Cuff (Tested Tech Solutions)</td>
<td>314</td>
</tr>
<tr>
<td>5) Living in a World Without Source Lines Of Code (SLOC); John Sautter (Northrop Grumman)</td>
<td>348</td>
</tr>
<tr>
<td>6) Tracking Software Development: An example of feature-based estimating; Capt. John Rosson (United States Air Force, AFCAA)</td>
<td>378</td>
</tr>
<tr>
<td>7) Implementing Continuous Iterative Development in DoD Acquisition; Cheryl Jones (United States Army)</td>
<td>396</td>
</tr>
<tr>
<td>8) Automated Requirements Analysis – Why and How To Use COSMIC FP Effectively On Agile Development; Colin Hammond and Lonnie Franks (ScopeMaster)</td>
<td>409</td>
</tr>
<tr>
<td>9) Estimating SW Costs from Requirements using Objective Function Points Research; Office of the Director of National Intelligence (ODNI)</td>
<td>428</td>
</tr>
<tr>
<td>10) Joint Agile Software Innovation (JASI) Cost IPT, Katherine “Kammy” Mann (Department of Homeland Security, CAD)</td>
<td>447</td>
</tr>
</tbody>
</table>
WELCOME

It's a pleasure to welcome you to the 2019 Software and IT-CAST Symposium! It comes as no surprise that innovation in software and information technology — whether it be new products, processes or services — marches along. The United States Government, long past the time of dictating and leading innovation, faces a crossroads; how does it keep pace to this new reality?

Adding to the troubles is that cost has the dubious problem of being an unrecognized stakeholder. Resource allocation, financial information, and performance-based tracking is imperative for success and is a tenuous balancing act in acquisition, cost, and program management activities. Luckily, through this symposium and other venues we are tackling these inequities and searching for solutions that are aligned with the more traditional stakeholders.

Over the next two days, you will find a program that explores how cost is addressing some of our most significant challenges. And through the plenary sessions and discussions that follow, we will explore how our tradecraft should evolve and where we can improve our profession.

We encourage you to take advantage of the opportunities available at the IT-CAST Symposium to connect with others and let new partnerships grow. You will find that many attendees share similar challenges, and others who have found viable solutions are here to share what they did. We will have one happy hour event and numerous breaks and we hope you will be able to make the most of these sessions and meet with many of those in attendance.

Thank you for joining us this year! And a very special thank you to all of our speakers! We hope the next few days are a unique and rewarding experience that helps you meet all of your business, educational and networking objectives.

Sincerely,

Vjosa Dreshaj and Lyle Patashnick
The Software and IT-CAST Executive Team
8:30-8:35 a.m. ............. Announcements
Gregory Niemann

8:35-8:45 a.m. ............. Introduction
Mr. John Scali

8:45-9:15 a.m. ............. KEYNOTE: Tradecraft Evolution for Emerging Needs
Ms. Jen Rose

9:20-9:50 a.m. ............. Living in the Clouds
Walking through the Department of Justice (DOJ) United States Trustees Program (USTP) Cloud journey over the last three years, I will delve into Cloud cost optimization techniques that can be applied in your agencies. Cloud cost optimization requires a solid understanding of a Cloud’s offerings combined with meaningful data analysis and visualization that is continually providing value in the Cloud and impacting your agency’s mission. This requires a solid understanding and knowledge of the Cloud services offerings, licensing, discounts, automation options and proper configuration of their services. I will break down specific case studies to show how the migrating to the Cloud can not only save money, but provide better value to users and enhance the mission, enabling IT resources to focus on higher-value work. The goal of this talk is to provide specific and detailed examples of cost optimization strategies for Cloud services that can be applied in each of your organizations.

   Key Presenter
   Michael Cassidy
   Chief Technology Officer, Department of Justice
   United States Trustees Program

9:55-10:10 a.m. | Break

10:10-10:40 a.m. .......... Forecasting Future Amazon Web Services Pricing
The National Reconnaissance Office (NRO) Cost and Acquisition Assessment Group (CAAG) produces independent cost estimates to support decision making, budgeting, and trade studies. Cloud service costs procured from Amazon Web Services (AWS) are becoming increasingly scrutinized. We conducted a thorough analysis to collect historical AWS prices and modeled the downward trend. Autoregressive time series models were fit to storage and compute service prices, resulting in annual price reduction rates to be applied in future estimates.

   Authors
   James Smirnoff and Hassan Souiri
   National Reconnaissance Office

10:45-11:15 a.m. .......... Causality and Uncertainty: A New Wave for Cost Estimation
SEI research in the past seven years has progressed methods and tooling for early life cycle software cost estimation. The early life cycle cost estimation method and tooling known as QUELCE (Quantifying Early Lifecycle Cost Estimation) combines scenario planning workshops with Bayesian Belief probabilistic models and Monte Carlo simulation to model uncertainty as front-end inputs to existing cost estimation machinery. To enable the QUELCE framework to guide stakeholders in interventions for cost containment, reduction, and price negotiation, recent SEI research into causal modeling of observational data is being used to distinguish correlated factors from causal factors of program performance that affect
software cost. This talk describes the practical aspects of QUELCE and the ability, using open-source tooling from Carnegie Mellon University, to supplement QUELCE with causal search. The newly added causal search step better controls the exploding probabilistic model derived from expert opinion by reducing the number of factors included into the software cost model. Participants will take away job aids including process flowcharts for the complete methodology of QUELCE and conducting causal search. Participants will be encouraged to use QUELCE and conduct their own research using the causal learning tools and methods.

Authors

Bob Stoddard and Dr. Mike Konrad
Software Engineering Institute (SEI)

11:55 a.m.-1:00 p.m. | Lunch

1:00-1:30 p.m. .......... Better Data Communication

A first step to improving the way you communicate data and analysis is to have some basic understanding of best practices and strategies. In this talk, I lay out three principles for better data visualization: Show the Data, Reduce the Clutter, and Integrate Graphics and Text. I also lay out three principles for better presentations: Visualize, Unify and Focus. Together, with the help of examples, both good and bad, I demonstrate how anyone can more effectively communicate their data and elicit insight.

Key Presenter
Jon Schwabish
The Urban Institute

1:35-2:05 p.m. .......... Data Exploitation — More Value, Less Time

Cost analysts have historically operated on small, manageable datasets. Preparing data for a task may be tedious and time consuming but overall achievable. Those days are over. More and more data is being collected, yet that collection and storage is being handled under the same strategies as yesteryear. The unfortunate result is datasets and databases which are difficult, if not impossible, to work with at scale.

This presentation discusses an approach to normalize, categorize and analyze data as applied to a larger dataset. The resulting product is a compact, usable database that can answer any question the data itself is capable of answering, without additional prep work. Such a solution would take an order of magnitude more time to develop using lesser methods and tools.

The methodologies presented are consistent with data science best practices — formalized by the popular “grammar of data manipulation” implemented in R statistical software and the related “tidyverse” packages. The goal is to communicate a flexible, scalable thought process which can be applied to any dataset.

Author
Adam James
Technomics

2:10-2:30 p.m. .......... Calibrating COCOMO® II for Functional Size Metrics

While COCOMO® II provides generalizable effort estimates, the accuracy of the estimates depends on the accuracy of the input parameters — namely, size in Source Lines of Code (SLOC), personnel, product and environmental attributes. Since SLOC is nearly impossible to estimate accurately until the project is nearly completed, it would be desirable to have a generalizable effort estimation model that allows size to be represented by functional size metrics, such as IFPUG (FPs) and COSMIC (CFPs) Function Points.
Since SLOC represents software size at a much lower level of granularity compared to functional size metrics, the effects various effort factors have on effort may need to be adjusted — particularly the 5 Scale Factors (Precedentedness, Development Flexibility, Team Cohesion, Risk and Architecture Resolution, and Process Maturity), which affect the rate at which effort grows with respect to size, and product characteristics, such as Product Complexity. We invite expert input on how the COCOMO® II ratings should be adjusted via a Wideband Delphi being held at the 20th Practical Software and Systems Measurement (PSM) User's Group Workshop September 16–20. Additionally, we will have handouts to get initial parameter ratings during the breaks and lunch. The insights gained may eventually be included in COCOMO® III.

Author
Anandi Hira
University of Southern California

2:30-2:45 p.m. | Break

2:45-3:15 p.m. ............. Cost of Software Obsolescence Resolution of Real-Time Software

Software obsolescence happens when the original developer and authorized third party cease to provide support with regular updates, upgrades, and fixes or due to changes in the target environment, systems, and hardware, which makes software unusable (S. Rajagopal et al, 2014). It has been identified by the means of literature reviews and various interviews with the project teams in Ministry of Defence and Defence Industries that there is a requirement for developing a systematic framework that allows the forecasting and estimation of software obsolescence at a very early stage of the projects. It is even more important to develop a framework for identification, resolution and mitigation of Software Obsolescence issues that may arise during the lifetime of the software.

Author
Rajagopal Sanathanan
QinetiQ

3:20-3:50 p.m. ............. New Army Software Sustainment Cost Estimating Results

The Army has conducted a study over the past six years to improve the estimation accuracy of software sustainment systems cost. Based on an extensive data call of 192 Army systems, data analysis revealed several types of cost estimating relationships based on release type, release rhythm and three categories of data. Analysis of a sustainment cost risk model was also conducted. This presentation will show the study results, including what worked and did not work. A paper providing additional detail on this presentation is available.

Authors
Cheryl Jones and James Doswell
United States Army

3:55-4:30 p.m. ............. Agile Projects and GAO Best Practices

This presentation will provide an overview of the GAO Agile Assessment Guide (currently being developed). It will focus on Chapter 9 (program controls), which ties the best practices established in previous GAO guides to Agile development efforts. Additionally, the presentation will provide a more detailed look into areas of consideration currently discussed in the draft guide related to cost estimating, scheduling and earned value management.

Author
Jennifer V Leotta
Government Accountability Office
8:30-8:35 a.m. .......... Announcements
Gregory Niemann

8:40-9:10 a.m. .......... KEYNOTE — Software is Never Done: Refactoring the Acquisition Code for Competitive Advantage

U.S. national security increasingly relies on software to execute missions, integrate and collaborate with allies, and manage the defense enterprise. The ability to develop, procure, assure, deploy and continuously improve software is thus central to national defense. At the same time, the threats that the United States faces are changing at an ever-increasing pace, and the Department of Defense's (DoD's) ability to adapt and respond is now determined by its ability to rapidly develop and deploy software to the field. The current approach to software development is broken and is a leading source of risk to DoD: it takes too long, is too expensive, and exposes warfighters to unacceptable risk by delaying their access to tools they need to ensure mission success. Instead, software should enable a more effective joint force, strengthen our ability to work with allies, and improve the business processes of the DoD enterprise.

The Defense Innovation Board (DIB) recently presented a report to Congress and DoD arguing that DoD and industry must change the practice of how software is procured and developed by adopting modern software development approaches, prioritizing speed as the critical metric, ensuring cybersecurity is an integrated element of the entire software life cycle, and purchasing existing commercial software whenever possible. In this talk, I will briefly survey the recommendations from this report, focusing on those aspects of the report related to cost assessment and performance estimation of software programs (and software-intensive components of larger systems). This includes the use of modern metrics for tracking performance for software programs and driving improvement in cost, schedule and performance.

Keynote Speaker
Dr. Richard M. Murray
California Institute of Technology

9:15-9:45 a.m. .......... Estimating and Tracking Agile Software Development Projects

Generating accurate cost estimates for software development efforts has always been problematic, and no universal solution exists. What are the primary factors that make software cost estimation challenging, and what can be done to improve our cost estimates and accurately track the progress of agile software development? This discussion will focus on one approach to develop software cost estimates to better align with agile software development methodologies and accurately track software development programs throughout their life cycle.

Key Presenter
Matt Kennedy
United States Treasury

9:50-10:10 a.m. | Break
10:10-10:40 a.m. .......... KEYNOTE — Software Acquisition: Pathway, Appropriation, and Value of Enterprise Infrastructure

Keynote Speaker
Dr. Jeff Boleng
Special Assistant for Software Acquisition to the Under Secretary of Defense for Acquisition and Sustainment

10:45-11:15 a.m. .......... A Foundation for Software Acquisition Decisions

Electricity and the telephone took over 30 years to be adopted by more than 25 percent of U.S. households, while the smartphone was adopted in less than five. The advent of cloud computing has provided the ability to access computer services over the internet and significantly changed the initial costs of companies. Currently, the Fortune 500 list is overturning 20–50 companies annually. As George W. Bush stated: “You can't do today's job with yesterday's methods and be in business tomorrow.” The challenge that industry and the government is facing is how to stay relevant in this environment. Our foundational business practices and decision-making processes will determine our ability to compete. Today, the projected cost of the weapon system or modernization effort is one of the primary facts that makes it all the way to Congress. When software is malleable and continuously delivered, how do we provide foundations for decisions?

Author
Tory Cuff
Tested Tech Solutions

11:20-11:50 a.m. .......... Living in a World Without SLOC

With the recent publications from the Defense Science Board and Defense Innovation Board (DIB), department and federal agency leaders are pushing us away from using Source Lines of Code (SLOC) as a primary software size or estimates and actuals. This presentation will review some of the alternative software sizes. We will review the alternate sizing available in the more popular software parametric tools. This presentation will also include an overview introduction to Simple Function Points as an alternative to SLOC.

Author
John Sautter
Northrop Grumman

11:55 a.m.–1:00 p.m. | Lunch

1:00-1:30 p.m. .......... Tracking Software Development: An Example of Feature-Based Estimating

AFCAA is working to develop a feature-based software cost and schedule estimating approach that can be used to track the execution of software development for DoD acquisition programs already underway. The goal is to develop metrics that will measure the progress of software development as it advances through three phases — code development, software integration, and test — and use those metrics to estimate the remaining cost and schedule of the effort.

Our preliminary approach consists of two stages: 1) map system or subsystem features to a program WBS at the control account level and 2) gather labor hour data spent on code development, software integration, and test for each of the system or subsystem features. In addition to estimating the remaining effort, this methodology will be useful in identifying both the features and software development phases that are the most effort intensive,
helping to develop an understanding of how each phase scales with content to allow for more effective cost and schedule estimates. As data is collected, productivity metrics can be developed using labor hours and the progression of features through each phase. As this effort continues, this methodology should identify which existing metrics data will be most useful to the cost community.

Author
John Rosson, Capt.
United States Air Force

1:35-1:50 p.m. .......... A Path Toward Consensus Measures for Iterative Software Development

Traditional measures used to plan and manage software programs based largely on waterfall development and software lines of code-based estimates are not keeping pace with trends in the defense industry toward methods based in a software factory environment, including automated testing, continuous integration, and rapid iterative development and deployment of new capabilities. The Defense Science Board (DSB) and Defense Innovation Board (DIB) recommended measures for continuous iterative development and agile programs. A joint NDIA, INCOSE and PSM working group surveyed the community for feedback on the usefulness and effectiveness of these measures and has been developing a framework based on information needs to help reach industry consensus on candidate measures. This presentation will summarize current recommendations, feedback from the community, and a path forward on a consensus measurement framework.

Author
Cheryl Jones
United States Army

1:55-2:25 p.m. .......... Why and How to Use COSMIC FP Effectively on Agile Development Contracts

In this presentation, Colin and Lonnie will be presenting the use of automated analysis of software requirements for both functional sizing and early requirements quality improvement. The presentation is based in part on the effectiveness of ScopeMaster, the world’s first automated software requirements analyzer. ScopeMaster estimates functional size (both COSMIC and IFPUG) and identifies potential errors in software requirements at very high speed, directly from the language of user stories. It helps the user remove around 50 percent of requirements defects in a fraction of the time it would take to do manually. With over 75 years of software project management between them and substantial experience in both IFPUG and COSMIC, they will highlight the merits of using COSMIC Function Points and its particular suitability to Agile and embedded systems work. They will show the importance of knowing the functional size early and how it lets you manage cost, velocity and quality throughout an Agile project. They will also show how this can contain technical debt and reduce project risk, especially on larger software projects.

Authors
Colin Hammond and Lonnie Franks
ScopeMaster

2:30-2:45 p.m. | Break

2:45-3:15 p.m. .......... Automatic Objective Function Points

Searching for better methods to take advantage of latest technologies, we investigated how to automatically capture Function Points from actual SW code. We proposed using Cyclomatic Complexity to calculate Function Points by selecting the appropriate Function
Point standard tables. This method enables the ability to capture “Objective” Function Points directly from SW code. While investigating this process and latest technologies, we experimented with the Natural Language Toolkit (NLTK) to parse through textual definitions to determine if it is possible to estimate “Objective” Function Points directly from a SW document such as Agile “Features.”

Author
Paul Cymerman
Quaternion Consulting

3:20-3:40 p.m. ............. JASI Cost IPT – Join the Band!

The Joint Agile Software Innovation (JASI) Cost IPT was established in 2018 as a cross-government agency team with the purpose of exchanging cost data and information related to Agile software development. The goal of this Cost IPT is to improve the cost community’s ability to estimate the cost of software development in an Agile software development environment and track progress to successful completion using Simplified Function Points. This presentation will discuss the origins of JASI, what we do as an IPT, a brief introduction to Simplified Function Points, and how you can join this innovative and dynamic team!

Author
Katharine “Kammy” Mann
Department of Homeland Security

3:45-4:30 p.m. ............. Agile Centers Panel Discussion

Come pop the “Agile Bubble” as we explore topics on: agile measures, which data is most useful for answering RFPs, how far we should go in standardization of agile, which data is best used for performance tracking, and others that may come to pass. This panel is meant to be interactive; arrive prepared to ask questions of this very knowledgeable group.

Panel
Sarah Nichols
Northrup Grumman
(Representative)
Boeing
Robin Yeman
Lockheed Martin

Facilitator
Lyle Patashnick
National Geospatial-Intelligence Agency
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.

Dr. Richard M. Murray
California Institute of Technology

Dr. Richard M. Murray is the Thomas E. and Doris Everhart Professor of Control & Dynamical Systems and Bioengineering at the California Institute of Technology (Caltech). He received the B.S. degree in Electrical Engineering from California Institute of Technology in 1985 and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California, Berkeley, in 1988 and 1991, respectively. Murray served on the Air Force Scientific Advisory Board from 2002-2006 and has served on advisory committees for the Jet Propulsion Laboratory (JPL), the Pacific Northwest National Laboratory (PNNL) and the Defense Advanced Research Projects Agency (DARPA). Murray is an elected member of the National Academy of Engineering (2013) and the recipient of the 2017 IEEE Control Systems Award. He is a current member of the Defense Innovation Board, where he co-chairs the Science and Technology Subcommittee, and served as co-chair of the Software Acquisition and Practices (SWAP) report.

Ms. Jennifer Rose
Director, Cost and Acquisition Assessment Group, National Reconnaissance Office (NRO)

Ms. Jennifer Rose was selected as the Director, Cost and Acquisition Assessment Group (CAAG), within the National Reconnaissance Office (NRO) Business Plans and Operations (BPO) Directorate in November 2017. With this selection, Ms. Rose joined the Defense Intelligence Senior Leader (DISL) ranks. As the Director, CAAG, Ms. Rose manages cost analysis and integrated performance management (IPM) functions for the agency.

Prior to joining the NRO, Ms. Rose led the National Geospatial-Intelligence Agency (NGA) Cost Assessment Division within the Corporate Assessment and Program Evaluation (CAPE) Office beginning in January 2015. Her group conducted unbiased, independent cost and resource analysis in support of NGA planning and programming activities, acquisitions, and CAPE studies.

Ms. Rose worked for defense contractor TASC from 1996 – 2015, holding numerous positions reflecting growth in technical knowledge and responsibility over the 19-year period. Her technical expertise ranges across cost analysis; data analysis; budget planning, programming, budgeting, and execution (PPBE); portfolio management; earned value management and analysis; program justification and advocacy; systems engineering; and acquisition support. She applied these skills as a direct support analyst for NGA, NRO, Defense Finance and Accounting Service, Army, Navy and Air Force. Ms. Rose concurrently held corporate leadership positions while providing direct support. In her last formal leadership role at TASC, Ms. Rose served as the Cost and Risk Analysis Division Director within TASC’s Financial Business Analytics Center of Excellence. She was responsible for over 50 analysts and approximately $12M of revenue based on cost analysis services across various customers in the Intelligence Community, Department of Defense and Civilian Agencies.

Ms. Rose began her career as a cost analysis intern with the Naval Center for Cost Analysis in 1994. Ms. Rose graduated with a Bachelor of Arts degree with a double major in Mathematics and English from Washington College, Chestertown, MD. She earned a Master of Science degree in Operations Research from the College of William and Mary, Williamsburg, VA. She is a Certified Cost Estimator/Analyst (CCE/A) with the International Cost Estimating and Analysis Association (ICEAA) and a Level III certified DAU Business Cost Estimator.
Michael Cassidy  
*Chief Technology Officer, Department of Justice, United States Trustees Program*

Mike Cassidy is the Chief Technology Officer at the Department of Justice United States Trustees Program (USTP) focusing on modernizing the USTP IT services using Agile and DevOps practices on modern Cloud services platforms (e.g., IaaS, PaaS and SaaS). Over the past three years Mike has focused within USTP performing a “lift and shift” migration to the Cloud. During this three-year journey, Mike has lead the effort to capture detailed cost analysis on this migration identifying cost savings, cost avoidance and enhanced opportunity costs that have been realized. USTP was recognized by GAO as one of the leaders in the Cloud Cost savings in their report “Agencies Have Increased Usage and Realized Benefits, but Cost and Savings Data Need to Be Better Tracked” (Refer to: https://www.gao.gov/products/GAO-19-58).

Mike has a B.S. in Consumer Economics from University of Maryland College Park and a M.S. in Information Technology from University of Maryland University College. Mike also holds a number of Cloud professional certifications including CompTIA Cloud+, CompTIA's Cloud Essentials, ISC's Cloud Certified Security Professional (CCSP) and Microsoft's Azure Architect.

Matt Kennedy  
*United States Treasury*

Matthew R. Kennedy is a Senior IT Program Manager and Contracting Officer Representative (COR) at the Office of the Comptroller of the Currency (OCC). Formerly, Matt was a Program Manager at the Army’s Program Executive Office - Enterprise Information Systems (PEO-EIS) and was a Professor of Software Engineering at Defense Acquisition University (DAU) where he specialized in agile acquisition. Matt served as the Associate Director of Engineering at the National Cancer Institute's Center for Biomedical Informatics and Information Technology and served in the U.S. Air Force as a network intelligence analyst. He has worked both inside and outside of the government on various IT projects over the last 18 years. Matthew holds a Bachelors in Computer Science and a masters and Ph.D. in Computer Science and Software Engineering from Auburn University. He is Defense Acquisition Workforce Improvement Act (DAWIA) Level III certified in Program Management, Systems Engineering, and Information Technology (IT) and a SAFe® 4.5 Program Consultant and Certified Scrum Professional (CSP).

Jon Schwabish  
*The Urban Institute*

Jon is currently a Senior Fellow in The Urban Institute's Income and Benefits Policy Center. He is also a member of the Institute's Communication team where he specializes in data visualization and presentation design. Prior to the time at Urban, Jon spent the previous 9 years at the Congressional Budget Office conducting research in such areas as earnings and income inequality, immigration, disability insurance, retirement security, data measurement, the Supplemental Nutrition Assistance Program (SNAP), and other aspects of public policy.

Jon created a number of policy-relevant data visualization products; wrote widely on data visualization and presentation techniques; and offer public workshops on those areas. Jon is generally known for calling for clarity and accessibility in research and wrote on various aspects of how to best visualize data including technical aspects of creation, design best practices, and how to communicate social science research in more accessible ways. Jon was named a “visualization thought leader” by AllAnalytics in 2013. Jon's book, Better Presentations: A Guide for Scholars, Researchers, and Wonks is designed to help presenters of scholarly or data-intensive content develop clear, sophisticated, and visually captivating presentations.

In addition to his efforts to improve how researchers communicate their findings to a wider audience, Jon continues to pursue his existing research portfolio. Ongoing and future work includes investigation of child nutrition programs; long-term earnings patterns among SNAP recipients; the relationship between state-level SNAP policies and individual participation decisions; coincident retirement-disability Social Security benefit claiming behavior; and patterns in earnings inequality and volatility within and across groups of workers.

Jon earned an M.A. and Ph.D. in Economics from Syracuse University and an undergraduate degree in Economics from the University of Wisconsin at Madison.
Tory Cuff  
Tested Tech Solutions

Victoria ‘Tory’ Cuff is the CEO and founder of Tested Tech Solutions, a consultancy for the DoD on software delivery. Tested Tech was founded to empower the government to deliver—based on the belief that potential solutions for current and emerging capability gaps are a combination of technological advancements, key partnerships, and investments in the current workforce. As a previous AF civilian, she began as a cost estimator that had worked on various weapon systems, including: aircraft, tactical data links, satellites, sensors and then turned her focus to software cost estimating. She became the Chief of the Agile Acquisitions branch supporting Kessel Run as it morphed from the modernization of the Air Operations Center (AOC) system modernization to the entirety of the HBB division plus the addition F-35 ALIS program. That role expanded her responsibility to include overseeing program management, contracting, financial management and cost estimating. Since her AF departure, she is now supporting the Joint Artificial Intelligence Center (JAIC) to support the acceleration and adoption of AI across the DoD.

Paul Cymerman  
Quaternion Consulting

Paul Cymerman works for Quaternion Consulting Inc. (QCI) supporting the Office of Director of National Intelligence (ODNI). He has over 32 years of experience as a software cost analyst, aerospace engineer, and computer programmer. He is currently supporting the ODNI in developing independent cost estimates and researching new estimating methods. Back in 2001, Paul proposed code counting standards using the University of Southern California (USC) code counter. In 2004, he proposed and developed the original “Diff” capability to USC tool to help analyze SW development process. This was a first in the SW cost community to be able to extract the actual changes in the SW development.

James Doswell  
United States Army

James Doswell is a Senior Operations Research Analyst in the Networks, Information, Software & Electronics Costing (NISEC) Division of the Office of the Deputy Assistant Secretary of the Army for Cost and Economics (ODASA-CE). James is a technical advisor and is responsible for software and electronics estimating for major Army programs. He has also been an instrumental part of the Army software maintenance data collection initiative for the last 5 years.

Lonnie Franks  
ScopeMaster

Lonnie is a Senior Executive Consultant with extensive knowledge and experience in the delivery of large-scale IT projects over the last 35 years, for financial institutions, leading blue chip companies and the US government, involving both on-shore and off-shore development. His key skills include sizing, estimating, planning, organizing and controlling large-scale IT development projects using proven metrics; ensuring that projects have the expected and required functionality; that projects are done within a reasonable (and predictable) schedule; that projects meet cost targets and benchmarks; and that projects are delivered with exemplary quality. In the area of quality management, Lonnie’s skills include setting quality targets and measuring the actual quality achieved for all work products; analyzing quality variances between expected and actual quality; doing root cause analysis and providing closed loop corrective actions for quality issues; ensuring that the right product is developed in the right way with outstanding quality the first time; and minimizing rework because of quality issues. Lonnie has used IFPUG function point analysis and, more recently, COSMIC FSM.

Colin Hammond  
ScopeMaster

Colin Hammond is a software project leader, innovator and entrepreneur. He is the inventor of ScopeMaster, the world’s first requirements analyzer that simultaneously performs both automated functional sizing and quality assurance by interpreting the functional intent of written software requirements. Colin is a certified IFPUG and certified COSMIC Function Point specialist. For 30 years he worked as a software project, program and portfolio manager at numerous large organizations in the UK. An engineer by training he combines technical skills with management skills to bringing certainty to software projects using effective metrics and techniques. Colin now works full time at ScopeMaster helping software project leaders achieve greater certainty and faster success through early sizing and improved requirements quality work.

Anandi Hira  
University of Southern California

Anandi Hira is currently a PhD student under Dr. Barry Boehm at University of Southern California’s (USC’s) Computer Science Department. Her research interests lie in cost estimation and models. She has been a part of the Unified Code Count (UCC) development effort at USC CSSE for the past 6 years and has been collecting and analyzing the data to improve the development processes and the product’s quality. Anandi has also joined the effort within USC CSSE to develop COCOMO® II (Constructive COst MOdel) as an update from COCOMO® II.
Adam James  
Technomics

Adam James is a Senior Analyst at Technomics, Inc., where he serves as a strategic leader and data scientist for the newly established Technomics Innovation Lab (TIL). In this role, Adam provides his expertise to develop modern, innovative solutions to address new – and old – cost analysis problems. Adam’s current focus is helping clients extract value from datasets of varying complexity. Recently, he helped the Army analyze traditional cost estimating data sources such as the Cost and Software Data Reports (CSDRs) and Contracts/Mods using modern tools and data science techniques. Prior work includes serving as a lead author of the Joint Agency Cost Estimating Relationship (CER) Development Handbook.

Adam earned a M.S. in Statistics from Virginia Tech in 2012. He also has a B.S. from Virginia Tech with a double major in Mathematics and Statistics. He received the International Cost Estimating and Analysis Association (ICEAA) Technical Achievement Award in 2016. Adam also has been active in contributing to the community knowledge base, winning best paper in both the ‘Analysis & Modeling, Machine Learning” and “Methods, Data Collection & Management” categories at ICEAA in 2019.

Cheryl Jones  
United States Army

Cheryl Jones works in the Strategic Quality initiatives Group at FCDD-ACE-QSA at Picatinny Arsenal, New Jersey. Ms. Jones is a technical lead for the Army Software Sustainment Cost Estimation initiative. The objective of this project is to provide the Department of the Army with the ability to accurately estimate, budget, allocate, and justify the software maintenance resources required to meet evolving mission and service affordability requirements across the system life-cycle.


Dr. Mike Konrad  
Software Engineering Institute (SEI)

Michael Konrad is a principal researcher at the Software Engineering Institute (SEI), currently using causal and machine learning and simulation to help achieve improved analytic capabilities in: software estimation, engine heat, and video understanding. From 1998 to 2013, he contributed to CMMI in many technical roles. From 1988 to 1998, Konrad was a member of the teams that developed the original Software CMM and ISO 15504. He is coauthor of the CMMI for Development (CMMI-DEV) books. Konrad received his PhD in mathematics from Ohio University in 1978; and is a Senior Member of the IEEE.

Jennifer V Leotta  
Government Accountability Office

Jennie Leotta is an Assistant Director for the Government Accountability Office (GAO). She is responsible for performing cost, schedule, and Earned Value Management analyses to support audits on a wide range of government programs. Before joining the GAO, Jennie worked for the Department of Homeland Security and the Navy, developing and analyzing cost estimates. Jennie holds a B.S. in Quantitative Finance from James Madison University and an M.A. in Economics from George Mason University.

Katharine “Kammy” Mann  
Department of Homeland Security

Ms. Kammy Mann is an Operations Research Analyst for the Department of Homeland Security (DHS), Cost Analysis Division (CAD). She has supported numerous projects and programs in the DoD and DHS and for the NATO Communication and Information Agency (NCIA) in Brussels, Belgium. Ms. Mann is a member of the Agile IT Software Development commodity team at DHS CAD responsible for approving all Software Major Acquisition Programs Life Cycle Cost Estimates. She is the current Secretary for the Joint Agile Software Innovation (JASI) Cost IPT and Membership Chair for the Washington Chapter of ICEAA. She holds B.S. and M.S. degrees in Industrial and Systems Engineering from Virginia Tech, is a Certified Cost Estimator/Analyst (CCEA®), and is a member of the International Function Point Users Group (IFPUG).

Sarah Nichols  
Northrup Grumman

Sarah Nichols has 25 years of experience in the government consulting and banking industries in the management and training fields. She is currently a Lead Enterprise Agile and DevOps Coach and trainer. She received a Master’s in Adult Education for Wayland Baptist University, a BS in Finance from Oklahoma Christian University, and an Associates in Journalism and Math from York College. She holds certifications with the American Society of Quality Certified Quality Manager/Organizational Excellence (CQM/OE) and is a PMI Project Management Professional (PMP), as well as a certified CMMI associate.
Lyle Patashnick  
National Geospatial-Intelligence Agency

Lyle Patashnick has over 15 years' experience in the government cost tradecraft. A serial connector and collaborator, Lyle is passionate on pursuing joint ventures, whether it be as a founding member of the JASI CIPT or an integral member for the present and several of the past Software and IT-CAST Symposia. Since 2017, Lyle has supported NGA's Corporate Assessment and Program Evaluation Division, working on Automatic Simplified Function Point Estimation and Automatic Traceability in programmatic and engineering artifacts. Prior to that, Lyle was a cost analyst for several DoD and non-DoD federal agencies. In addition to cost, Lyle has a Juris Doctor from Northwestern University School of Law and is licensed to practice in both Illinois and Washington, D.C.

John Rosson  
Capt, United States Air Force


Rajagopal Sanathanan  
QinetiQ

Sanath is a highly capable and internationally recognized Cost Engineer/Cost Estimator specializing in Software Estimating, Software Obsolescence Management, Software Obsolescence Cost Estimating and Parametric Estimating. Sanath is an ISPA (International Society of Parametric Association) certified parametric estimator who worked in various high-profile Defence projects in ISS/ISTAR, Air, Maritime and Land domains, providing costing services at all stages of the procurement cycle in setting budgets, historic trends and Independent Cost Estimates assurances for government business cases. Currently, Sanath is working as a Cost Estimating Manager for Cyber, Information and Training in QinetiQ. Sanath is a Fellow of QinetiQ and Fellow of Defence Equipment and Support. Sanath served as a UK Director for ICEAA, and he is the current Chairman of SCAF (Society of Cost Analysis and Forecasting), UK.

John Sautter  
Northrop Grumman

John Sautter is a Level 6 Software Estimator working in the Northrop Grumman Technology Services Global Services Division. Mr. Sautter serves as task lead in the collection and evaluation of project historical data and is often assigned to high-priority large new business acquisitions and also helps perform independent cost evaluations, non-advocate project reviews, and startup reviews. Mr. Sautter serves as the lead facilitator of the Northrop Grumman Cost Estimation Community of Practice. Mr. Sautter is a trained function point specialist and is the corporate liaison to the International Function Point Users Group (IFPUG). Mr. Sautter is a TS member of the software Center of Excellence and the Agile Center of Excellence with a focus on software metrics and estimation. Mr. Sautter has over 37 years of experience working in software engineering, project management, and organizational process improvement. Mr. Sautter holds a BS degree in Computer Science and an MS in Organizational Performance.

James Smirnoff  
National Reconnaissance Office

Hassan Souiri  
National Reconnaissance Office

Bob Stoddard  
Software Engineering Institute (SEI)

Robert Stoddard is a principal researcher within the Software Engineering Institute focused on research and customer work involving causal and machine learning. Recent projects include jet engine predictive maintenance, test and simulation of weapon systems, and early lifecycle cost estimation. The SEI cost research transcends traditional cost estimation to actual causal modeling of program software costs, moving from prediction modeling to uncertainty modeling to prescriptive modeling. Robert Stoddard is a Fellow of the American Society for Quality and Senior Member of the IEEE with 24 years industrial experience and 14 years SEI research.

Robin Yeman  
Lockheed Martin

Robin Yeman works for Lockheed Martin in Northern Virginia as a Lockheed Martin Fellow, Certified Enterprise Coach, and an Agile Champion. She has over 20 years of experience in software and IT, across multiple business areas building everything from Satellites to Submarines. She has been actively supporting and leading Agile programs at Scale both domestically and internationally for the last 13 years with multiple certifications including SPC, CSM, CSP, PSM, PMP, PMI-ACP, INCOSE CSEP, and ITIL Practitioner. Robin has been actively driving DevOps into Lockheed Programs for the last 3 years and recently collaborated across all lines of business to develop an integrated delivery pipeline toolset known as Sparta to give new programs a jumpstart. She actively coaches and trains teams through in person coaching, Agile workshops and virtual training classes. She leads the Lockheed Martin's Agile Community of Practice and Center of Excellence and speaks at multiple conference engagements each year. Robin received her Master's Degree in Software Engineering from Rensselaer Polytechnic Institute.
Living in the Clouds

Michael Cassidy
CTO United States Trustees Program (USTP)
Department of Justice
Agenda

- Cloud Services & Billing Overview
- Training Staff
- Licensing Challenges & Tips
- Cost Saving Tips
- Cost Saving Examples
Cloud Billing Tips

- **Start Early** – Fully Understand Cloud Bill *Immediately*
- **Identify Resource** – Needs Strong Cloud Knowledge
- Many **Cloud Billing Tools** – Experiment
  - Larger Organizations May Need 3rd Party Solutions
  - Smaller Organizations – Keep Simple
- **Monthly Review** with Key Stakeholders
# Billing Overview

<table>
<thead>
<tr>
<th>Service</th>
<th>Unit of Measure</th>
<th>Included Quantity</th>
<th>Part Number</th>
<th>Unit Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Academic Knowledge - Standard - Transactions - Preview</td>
<td></td>
<td>10000</td>
<td>AAA-17686</td>
<td>$2.60</td>
</tr>
<tr>
<td>Advanced Data Security for SQL Database - Standard Node</td>
<td>1 /Month</td>
<td>0</td>
<td>AAA-54514</td>
<td>$15.60</td>
</tr>
<tr>
<td>Advanced Data Security for SQL Database - Standard Node - US DoD</td>
<td>1 /Month</td>
<td>0</td>
<td>AAD-46271</td>
<td>$21.84</td>
</tr>
<tr>
<td>Advanced Data Security for SQL Database - Standard Node - US Gov</td>
<td>1 /Month</td>
<td>0</td>
<td>AAD-46269</td>
<td>$19.50</td>
</tr>
<tr>
<td>Advanced Data Security for SQL Database - Standard Trial Node</td>
<td>1 /Month</td>
<td>0</td>
<td>AAA-54515</td>
<td>$0.00</td>
</tr>
<tr>
<td>Advanced Data Security for SQL Database - Standard Trial Node - US DoD</td>
<td>1 /Month</td>
<td>0</td>
<td>AAD-46272</td>
<td>$0.00</td>
</tr>
<tr>
<td>Advanced Data Security for SQL Database - Standard Trial Node - US Gov</td>
<td>1 /Month</td>
<td>0</td>
<td>AAD-46270</td>
<td>$0.00</td>
</tr>
<tr>
<td>Advanced Threat Protection - Azure Database for MySQL - Standard Nodes - Preview</td>
<td>1 /Month</td>
<td>0</td>
<td>AAD-31471</td>
<td>$7.80</td>
</tr>
<tr>
<td>Advanced Threat Protection - Azure Database for MySQL - Standard Trial Nodes - Preview</td>
<td>1 /Month</td>
<td>0</td>
<td>AAD-31472</td>
<td>$0.00</td>
</tr>
<tr>
<td>Advanced Threat Protection - Azure Database for PostgreSQL - Standard Nodes - Preview</td>
<td>1 /Month</td>
<td>0</td>
<td>AAD-31470</td>
<td>$7.80</td>
</tr>
<tr>
<td>Windows Server - 6 vCPU VM License</td>
<td>10 Hours</td>
<td>0</td>
<td>AAA-70516</td>
<td>$2.87</td>
</tr>
<tr>
<td>Windows Server Burst - 16 vCPU VM License</td>
<td>100 Hours</td>
<td>0</td>
<td>AAA-70572</td>
<td>$6.66</td>
</tr>
<tr>
<td>Windows Server Burst - 2 vCPU VM License</td>
<td>100 Hours</td>
<td>0</td>
<td>AAA-70526</td>
<td>$0.83</td>
</tr>
<tr>
<td>Windows Server Burst - 20 vCPU VM License</td>
<td>100 Hours</td>
<td>0</td>
<td>AAA-70571</td>
<td>$8.32</td>
</tr>
<tr>
<td>Windows Server Burst - 4 vCPU VM License</td>
<td>100 Hours</td>
<td>0</td>
<td>AAA-70510</td>
<td>$1.66</td>
</tr>
<tr>
<td>Windows Server Burst - 8 vCPU VM License</td>
<td>10 Hours</td>
<td>0</td>
<td>AAA-70527</td>
<td>$0.33</td>
</tr>
<tr>
<td>Workload Protection for App Services - Standard - Nodes</td>
<td>100 /Hour</td>
<td>0</td>
<td>AAA-97326</td>
<td>$2.08</td>
</tr>
<tr>
<td>Workload Protection for App Services - Standard - Nodes - US Gov</td>
<td>100 /Hour</td>
<td>0</td>
<td>AAA-97328</td>
<td>$2.60</td>
</tr>
<tr>
<td>Workload Protection for App Services - Standard Trial - Nodes</td>
<td>100 /Hour</td>
<td>0</td>
<td>AAA-97327</td>
<td>$0.00</td>
</tr>
<tr>
<td>Workload Protection for App Services - Standard Trial - Nodes - US Gov</td>
<td>100 /Hour</td>
<td>0</td>
<td>AAA-97329</td>
<td>$0.00</td>
</tr>
<tr>
<td>Xamarin University - Online Training - Users - Promo</td>
<td>1</td>
<td>0</td>
<td>AAA-35100</td>
<td>$57.20</td>
</tr>
</tbody>
</table>
### Billing Example: Legacy Server vs Cloud Server

<table>
<thead>
<tr>
<th>Legacy Server</th>
<th>Cloud Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>One Time Purchase w/ Long Lead Time</td>
<td>Instant Access w/ Constant Tuning</td>
</tr>
<tr>
<td>Manuel Time Consuming Setup</td>
<td>Automated Setup</td>
</tr>
<tr>
<td>Over “Engineered” w/ Many Upgrades</td>
<td>Fit for Purpose – Start Small</td>
</tr>
<tr>
<td>Annual Maintenance Agreement</td>
<td>Pay for Use- Turn Off w/ No Use</td>
</tr>
<tr>
<td>Limited Scalability</td>
<td>Auto Scale Up or Down</td>
</tr>
<tr>
<td>Less “Pieces” to Manage</td>
<td>Many “Pieces” to a Cloud Server</td>
</tr>
<tr>
<td>Pay Upfront w/ Less Long Term Monitoring</td>
<td>Constant Monitoring of Bill - Clean Up is Critical</td>
</tr>
</tbody>
</table>
Re-Skill-ing People to be “Cloud Smart”

- Identify Primary Cloud Providers
  - Define Training Plan for Cloud Providers
    - Examples: AWS, Azure, ServiceNow, Office 365, Salesforce, etc.

- On-Site Training – Easy to Train Staff On-Site

- Conferences -
  - Attend Cloud Provider Conferences (e.g., Microsoft Ignite, AWS Invent, etc.)
  - Gartner Peer Forums – Research Analysts – Conferences
  - Covers Entire Cloud Ecosystem for All Vendors
  - Get Outside “Beltway” for Conferences
Training Tips for becoming “Cloud Smart”

- **Identify** Cloud Training Tracks
  - Clearly Outline Training Tracks for Staff in various Clouds

- **Continually...**
  - Follow Blogs, Newsletters, Conferences, etc.
  - Regular **Meetings/Discussions** on Cloud Services w/ Vendor

- **Hands-On** Classroom Training & Conferences

- **Cloud Access** - CRITICAL
  - Example – Visual Studio Premium for Azure
Licensing Challenges

- Many Discounts and Packages = Complexity
  - Example: Azure Monitoring
- Hybrid Use Benefit
- Reserved or Spot Instances
  - Ensure Strong Baseline Data Set and Foundational Servers
- 3rd Party Applications – Cloud Marketplace & Legacy
Cost Saving Tips

Next Steps

- Automation – Move Beyond Basic Automation
  - Setup & Removal of Cloud Services
  - Enforce Tagging & Labeling
- Develop Near Real Time Cloud Bill Alerting
- Continually Review Licensing & Discounts
- Work Closely with Cloud Service Providers
- Monitor Cloud Workloads – Proper Sizing & Cost Efficiencies
- Eliminate On-Premise Hardware when Feasible
Cost Saving Example #1

**USTP Print & File Server Cloud Migration**

- **Print Migration** – Migrated 515 Print Queues
  - Challenges: Drivers and Older Printers -
- **Regional Print Servers** – East & Mid-West
  - Based on Network Latency Testing
- **File Server Migration** Completed Last Week
  - Lots of Planning, Coordination & Communication
  - Network Upgrade to Ethernet Key Foundational Piece
- **Significant Cost Savings** - $800,000 vs $18,000
Network Flow Migration

- Legacy Network Hardware Solution:
  - End of Life Coming Up
  - Unable to Handle Network Flows Capacity
  - Unable to Monitor/Integrate with Cloud Traffic
  - Maintenance Costs Higher

- Cloud Solution
  - Ability to handle more flows at 50% of Cost
  - Reduces Legacy Hardware Footprint
  - Dashboards More Effective – Already Identified/Solved Multiple Issues

Moderate Cost Savings - $280,000 vs $88,000 over 5 Years
Next Steps

- Ensure Staff is Properly Trained & Understands Cloud Services
  - Continual Learning CRITICAL
- Set Up Cloud Billing Reporting & Analysis Process
- Identify Cost Saving Projects – Start Small and Build Up
- Identify Resources to Read Daily on Cloud Services
  - Focus on Major Providers for Your Organization
References

1. Cloud Cost Tools Analysis: [Computer World Cloud Cost Overview](#)
2. Cloudability Cloud Costs: [Cloudability White Paper](#)
3. OMB Cloud Smart: [Cloud Smart Strategy](#)
4. GAO Cloud Cost Paper: [GAO Cloud Cost Study](#)
5. GSA DCOI Resources: [GSA Data Center Cloud Services](#)
7. NIST Evaluation of Cloud Computing Services: [NIST SP 500-322](#)
8. Cloud Certified Security Professional: [CCSP Certification Information](#)
9. Meetups – Cloud Provider Meetups Monthly
10. LinkedIn: [Mike Cassidy LinkedIn](#)
Introduction

- **Cost and Acquisition Assessment Group (CAAG)**
  - Critical role: Agency Cost Positions – “how much will it cost?”
  - Key responsibility: provide independent cost and schedule analyses for program planning, budget, acquisition decisions and design reviews

- NRO utilizes cloud services provided by Amazon Web Services (AWS)
  - Dedicated region for Intelligence Community (IC)
  - Operating since late 2014
  - Utilization and therefore cost have been steadily increasing
AWS Pricing Study
Study Outline

- Task: “To find and compile commercial AWS price history and see the pattern of price decreases in order to estimate future pricing”
  - Storage – Simple Storage Service (S3)
  - Compute – Elastic Compute Cloud (EC2)

- AWS prides itself on offering periodical cuts to service pricing
  - 66 total price reductions to commercial AWS since 2006

- Multiple factors influence cloud pricing
  - Competition
  - Economies of scale
  - Hardware costs
  - License costs

How can 12 years of AWS pricing data be leveraged to model future prices?
Study Impact

- Just as omitting inflation impacts can lead to a large estimating error (under-estimate), ignoring cloud service price decreases will likely produce an over-estimate.

![Annual Price Decrease Impact](chart)

Example with $100 first year cost and constant usage

- Prior to the AWS Price Study, CAAG estimates assumed annual cloud price decreases based on BLS metrics and SME judgement.

Study defends application of annual price reduction factors with rigorous analysis.
Approach

- Collected historical commercial AWS storage (S3) and compute (EC2) pricing
  - Raw data from Amazon to retrieve 2015-2018 pricing information
  - Archival website to find archived AWS S3 pricing dated back to 2006 and EC2 pricing to 2010
- Normalized pricing for analysis*
- Developed and evaluated candidate models to estimate future cloud service prices
- Calibrated model to AWS region utilized by NRO
  - Note: this presentation does not share the NRO model, but calibrates the commercial AWS results to AWS GovCloud using the same approach

*Due to the tiered S3 pricing structure, one storage level had to be assumed to analyze costs over time. Based on internal analysis, models assumed 7.11 PB.
To investigate an alternative method where the current price is effectively constrained to its actual value, an Autoregressive (AR) time series model was considered.

The AR model predicts a month’s price based on a certain number of previous month’s prices (referred to as order \( p \), where \( p \) is the number of previous months considered).

The resulting model coefficients produce an exponential curve that predicts future prices.

An annualized factor is then calculated from the continuous exponential curve to produce a step function.

![AR(p) Model](image)

**AR(p) Model**

\[
Z_t = c + \sum_{i=1}^{p} \varphi_i Z_{t-i} + \varepsilon_t
\]

- \( Z_t \) = price at time \( t \)
- \( c \) = constant, set equal to zero for our models
- \( p \) = order of model; number of previous months considered
- \( \varphi \) = parameters or weights of the model
- \( \varepsilon \) = error term
AR Model Order Selection

+ Time series models are typically fit to data that sporadically increase and decrease, such as stock market indices, home prices or the unemployment rate.

+ Given the monotonic behavior of S3 prices and near-monotonic behavior of EC2 prices, this price analysis differs from other time series models.

+ The table below shows the number of months between price reductions and the number of months between two consecutive price reductions.

<table>
<thead>
<tr>
<th>Months Between Price Reductions</th>
<th>32</th>
<th>14</th>
<th>10</th>
<th>15</th>
<th>10</th>
<th>14</th>
<th>2</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Months Between Consecutive Price Reductions</td>
<td>46</td>
<td>24</td>
<td>25</td>
<td>25</td>
<td>24</td>
<td>16</td>
<td>34</td>
<td></td>
</tr>
</tbody>
</table>

+ There was one price reduction that was followed two months later by another price reduction. Otherwise, all successive price reductions were separated by at least ten months.

+ AR(10), AR(14) and AR(15) seem like logical choices given the distances between price reductions.
The time series model fits the actual cost function very closely, as expected due to the stable, non-increasing behavior of the cost function.

- Predicts 14.3% annual price reductions

- Reverse fitting the exponential curve appears to fit the general shape of the data, but generally underestimates actual cost
AR(14) model was considered using the same constraints and objectives as the AR(10) model.

The back forecasted exponential curve is slightly higher than in the AR(10) model and represents a closer fit to the actual historical costs.

The model includes a longer price history covering more price reductions at some points and the back forecast curve aligning closer to the data, the AR(14) model is preferred over the AR(10) model.

14 month autoregressive (AR) time series model predicts 14.9% annual price reductions on commercial S3.
AWS GovCloud S3 Model

- Commercial AWS faces more pressure from competition to reduce prices and benefits from larger economies of scale than AWS GovCloud.

- GovCloud S3 was introduced in August 2011.

- 14 month AR hybrid model generated using commercial AWS S3 pricing up to August ‘11 and GovCloud S3 pricing since.

- Model forecasts a future GovCloud S3 annual price decrease of 11.6%.

Model result of 11.6% is reasonable given BLS PPI for storage curve (11.4%).
AWS EC2 offers thousands of combinations of service offerings
  - e.g. m5.xlarge Linux 1 Year Standard Reserved All Upfront
  - Analysis focused on popular services used

AWS EC2 prices have decreased 41% over the past five years (11% annually)
  - Includes explicit service price reductions and new generation compute instances at lower prices compared to predecessors

10 month AR proved best fit of AWS EC2 pricing data, forecasting 8.2% annual price reductions
To generate forecasted GovCloud EC2 prices, we apply the GovCloud to AWS S3 model ratio to the commercial EC2 model

- Results in projected annual price reductions of 6.4% on GovCloud EC2

\[
\frac{11.6\%}{14.9\%} \times 8.2\% = 6.4\%
\]
Summary

- Commercial AWS has seen significant price reductions over the past decade across storage and compute services
- AWS GovCloud realizes similar effects as the commercial region that contribute to price decreases, namely economies of scale, improvements in hardware technology and need to be competitive in an active market
- Price reductions and new billing options that lower costs have been introduced in the more limited history of GovCloud
- Cost estimates and budget forecasts must model both price decreases and usage growth over time
- Prices will be monitored and models regularly adjusted

CAAG models forecast annual price decreases of 11.6% for S3 and 6.4% for EC2
CAAG AWS Cost Model
AWS Monthly Pricing Calculator

Strengths:
- Precise
- Frequently updated
- Includes many services
- Captures small cost drivers
- Best for estimating next month's cost

Weaknesses:
- Does not accommodate changing requirements or prices
- Requires many detailed inputs, often unknown
- Does not account for enterprise tiered pricing
- Only estimates one month

140+ Inputs
CAAG AWS Cost Model (C3M)

- Addresses the major weaknesses of the AWS Monthly Calculator
  - Produces long term estimates
    - User selects start/end dates for individual services, up to 10 years
    - Monthly and annual phased estimate
    - Accommodates storage growth
    - Price reductions are modeled based on historical trends
  - Fewer inputs
    - 34 readily known inputs (22 have default values)
    - Only requires the major cost drivers for compute, storage and database
  - Captures effective enterprise tiered storage pricing

C3M produces accurate long term estimates from fewer inputs
CAAG Datasheet: Compute

- CAAG C2S datasheet simplifies EC2 into 5 use-cases vice 64-different server sizes on the AWS monthly calculator
  - General Purpose
  - Memory Optimized
  - Storage Optimized
  - Dense Storage
  - GPU instance
- The datasheet inputs for quantity of CPUs, Memory per CPUs, and server use-case enables the CAAG to accurately predict AWS pricing
- When hourly usage specified for on demand instances exceeds the break-even point a reserve instance is automatically estimated
- Estimate is time-phased by fiscal year based on user provided start and end dates
**CAAG Datasheet: Storage**

- CAAG datasheet simplifies storage into 6 storage types
  - Block Storage – General Purpose
  - Block Storage – High I/O
  - Block Storage – Infrequent Access
  - Object Storage – General Purpose
  - Object Storage – Warm Archive
  - Object Storage – Cold Archive

- CAAG uses historical pricing for consumed services to estimate costs and does not require detailed performance data for sizing
  - IOPS inputs
  - Snapshot Types
  - Put / Copy / Post / List
  - Get Requests

- Usage profile can be either a constant size or linear growth over time (e.g. an archive that grows by 500GB per month)

<table>
<thead>
<tr>
<th>IC Cloud Provider</th>
<th>Project</th>
<th>Item Description</th>
<th>DEV / TEST / PROD</th>
<th>Cloud Service Type</th>
<th>Begin Date</th>
<th>End Date</th>
<th>Usage Profile</th>
<th>yCPUs</th>
<th>Memory / CPU (GB)</th>
<th>Use Case</th>
<th>Hours / Day Usage</th>
<th>Storage Type</th>
<th>Storage (GB)</th>
<th>Deployment</th>
</tr>
</thead>
</table>
CAAG Datasheet: Database

- RDS is the database service offering and is composed of a compute and storage component
- The CAAG datasheet enables users to accurately describe RDS requirements for compute and storage
- **Compute Inputs**
  - Use Case
    - General Purpose
    - Memory Optimized
  - # CPUs and Memory / CPU (can be found in Tool Tip provided in Amazon AWS calculator)
- **Storage Inputs**
  - Block Storage – General Purpose
  - Block Storage – High /IO
  - Block Storage – Infrequent Access
- **Deployment**
  - Single Availability-Zone
  - Multiple Availability-Zone

<table>
<thead>
<tr>
<th>IC Cloud Provider</th>
<th>Project</th>
<th>Item Description</th>
<th>DEV / TEST / PROD</th>
<th>Cloud Service Type</th>
<th>Begin Date</th>
<th>End Date</th>
<th>Usage Profile</th>
<th>Compute vCPUs</th>
<th>Memory / CPU (GB)</th>
<th>Use Case</th>
<th>Hours / Day Usage</th>
<th>Storage Type</th>
<th>Storage (GB)</th>
<th>Deployment</th>
</tr>
</thead>
</table>

| Compute | Storage | Database |
Other Amazon Services

- Compute, storage, and RDS comprise >95% of GED AWS cost to date
- The AWS Monthly Calculator has a plethora of service options beyond Compute, Storage, and RDS
- The CAAG applies a factor to the compute and storage estimates to account for these other Amazon services
  - AWS CloudTrail
  - AWS Training
  - AWS Key Management Service
  - AWS Direct Connect
  - AWS Diode
  - AWS DB Migration Service
  - Amazon Cloud Watch
Other C3M Assumptions

22 inputs are prepopulated in C3M but user may tailor as appropriate.

- The CAAG has analyzed historical commercial AWS and AWS pricing and has developed rates at which service prices are expected to decrease (next slide)

- Additionally, the NRO may negotiate bulk buy discounts
  - Common in commercial AWS
  - Heavy consumers may negotiate discounts based on levels of usage (e.g. if Company ABC spends $M, AWS provides a 15% discount)
  - Independent from service price reductions

See User Documentation for more guidance on assumptions
Try C3M for Free Today

✿ C3M IntelDocs Links

Model: https://go.ic.gov/_fpzGHSL

User Documentation: https://go.ic.gov/maVaFKo
Contact Information

✦ James Smirnoff
✦ 571-304-8875
✦ smirnoff@nro.mil
✦ smirnoff@nro.ic.gov

✦ Hassan Souiri
✦ 571-304-8902
✦ souiriha@nro.mil
✦ souiriha@nro.ic.gov
Backup
### S3 Model Assumptions

- Due to the tiered S3 pricing structure, one storage level had to be assumed to analyze costs over time.
- Based on internal analysis, models assumed 7.11 PB.

<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>
</tr>
</thead>
<tbody>
<tr>
<td>0-1 TB</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.14</td>
<td>0.125</td>
<td>0.095</td>
<td>0.085</td>
<td>0.03</td>
<td>0.023</td>
<td></td>
</tr>
<tr>
<td>1-50 TB</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.125</td>
<td>0.11</td>
<td>0.08</td>
<td>0.075</td>
<td>0.0295</td>
<td>0.023</td>
<td></td>
</tr>
<tr>
<td>50-100 TB</td>
<td>0.15</td>
<td>0.14</td>
<td>0.14</td>
<td>0.11</td>
<td>0.095</td>
<td>0.07</td>
<td>0.06</td>
<td>0.029</td>
<td>0.022</td>
<td></td>
</tr>
<tr>
<td>100-500 TB</td>
<td>0.15</td>
<td>0.13</td>
<td>0.13</td>
<td>0.11</td>
<td>0.095</td>
<td>0.07</td>
<td>0.06</td>
<td>0.029</td>
<td>0.022</td>
<td></td>
</tr>
<tr>
<td>500-1000 TB</td>
<td>0.15</td>
<td>0.12</td>
<td>0.105</td>
<td>0.095</td>
<td>0.09</td>
<td>0.065</td>
<td>0.055</td>
<td>0.0285</td>
<td>0.021</td>
<td></td>
</tr>
<tr>
<td>1000-5000 TB</td>
<td>0.15</td>
<td>0.12</td>
<td>0.08</td>
<td>0.08</td>
<td>0.08</td>
<td>0.06</td>
<td>0.051</td>
<td>0.028</td>
<td>0.021</td>
<td></td>
</tr>
<tr>
<td>5000 TB or More</td>
<td>0.15</td>
<td>0.12</td>
<td>0.055</td>
<td>0.055</td>
<td>0.055</td>
<td>0.055</td>
<td>0.043</td>
<td>0.0275</td>
<td>0.021</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Storage (TB)</th>
<th>Effective Price ($ per GB/Month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7,110</td>
<td>0.1500  0.1209  0.0781  0.0759  0.0744  0.0596  0.0496  0.0280  0.0211</td>
</tr>
</tbody>
</table>
Data Collection

- Collected raw data from Amazon to retrieve 2015-2018 pricing information
- Used archival website to find archived AWS S3 pricing dated back to 2006 and EC2 pricing to 2010
- Compiled the EC2 data by OS, Term Type and Product Family
- Normalized the data in order to find the annual percent price change
- Narrowed dataset to service options most relevant to NRO
Calculate Annual Price Change

- Compare today’s price to the S3 introductory price and find annual percent change from S3 introduction to today
  \[
  \left( \frac{p_{\text{today}}}{p_0} \right)^{\frac{1}{t_0-t_{\text{today}}}} - 1 = -14.9\%
  \]

- Prices would be underestimated in almost the entire timeframe

- Next the underestimate bias was addressed by not constraining the introductory price

- This model resulted in an estimated 2006 price 25% higher than the actual price
Constrain Model Area and $p_0$

- Area between est. and act. curves constrained to 0; $p_0$ constrained to 0.15
- This method is always time biased (underestimates beginning years; overestimates later years)
C3M Definitions

Usage Profile:
Constant usage indicates that the C2S service will generally be utilized at a constant or uniform rate from the Begin Date to End Date.
Linear Growth (storage only) indicates that the service will grow at a constant rate over time. The inputs for linear growth should correspond to the first 30 days of usage. For example, if a program expects 1 TB of storage in the first month, 2 TB in the second, 3 TB in the third, etc., linear growth would be selected and 1,000 GB would be entered into the Storage (GB) input cell.

(Other usage profiles are possible. Use multiple rows and a combination of constant and linear growth profiles to describe a custom profile.)

Compute Use Cases:
General Purpose provides a balanced set of resources and a high level of processing performance. Ideal for applications that require balanced CPU and memory performance. Example applications include small and mid-sized databases, data processing, SAP, Microsoft SharePoint, encoding, high traffic content management systems, and memcached.
Memory Optimized offers large memory sizes for high throughput applications, including high performance databases, memory caching, in-memory analytics, and larger deployments of SAP, Microsoft SharePoint and other enterprise applications.
Storage Optimized provides Intel Xeon processors and direct-attached storage options optimized for applications with specific storage capacity and random I/O requirements. Use cases include MapReduce-based workloads, distributed file systems such as HDFS and MapR-FS, network file systems, log or data processing applications such as Apache Kafka, and big data workload clusters.
Dense Storage features HDD-based local storage and deliver high disk throughput. Example applications include Massively Parallel Processing (MMP), data warehousing, MapReduce and Hadoop distributed computing, distributed file systems, network file systems, and log or data processing.
GPU Instance is intended for graphics and general purpose GPU compute applications such as 3D application streaming, machine learning, and video encoding.

Storage Types:
Block Storage - General Purpose volumes are backed by SSDs and are suitable for small to medium-sized databases, development and test environments and boot volumes. Designed to offer single digit millisecond latency and deliver consistent baseline IOPS performance.
Block Storage - High I/O volumes are also backed by SSDs but offer a greater number of IOPS than Block Storage - General Purpose and therefore are suitable for applications with I/O-intensive workloads such as databases where performance consistency and latency is critical.
Block Storage - Infrequent Access volumes are backed by magnetic drives and are ideal for workloads where data is accessed infrequently.
Object Storage - General Purpose provides highly durable and available storage for a variety of content.
Object Storage - Warm Archive provides the same durable and available storage as Object Storage - General Purpose but is intended to be accessed infrequently. Intended for long-term storage, backups and as a data store for disaster recovery.
Object Storage - Cold Archive provides durable archival for infrequently accessed data where a retrieval time of several hours to days is suitable, depending on retrieval amount. Intended to be an archival storage solution where less than 5% of the data will be accessed per month.

Database includes SQL or NoSQL, fully managed and relational databases. Web service allowing quick set up, operation and scaling of databases in the cloud.
For database services, populate all fields for the Compute, Storage and Database sections.

Provide storage in GB (1 TB = 1,024 GB; 1 PB = 1,048,576 GB)
Causality and Uncertainty: A New Wave for Cost Estimation

Robert Stoddard
Mike Konrad

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Document Markings

Copyright 2019 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.


DM19-0804
Agenda

Original SEI Cost Research Motivation

The QUELCE Research Project Solution

SEI Causal Learning Research

Call to Action
Original SEI Cost Research Motivation

**Challenges:**
1) change and uncertainty
2) optimistic judgment

**Cost Estimate**
Based on:
- Limited Information
- Expert Judgment
- Analogies

**Acquisition Phases and Decision Milestones**

- Materiel Solution
- Technology Development
- Engineering & Manufacturing
- Production & Deployment

**Approval**

**Delay**

**Ground Combat Vehicle Delay Due to Reconciling Cost Estimates**
- 4 months delay in obtaining approval to proceed
- 30% discrepancy in cost estimates, Army v independent cost estimate
- Rework to conduct a new Analysis of Alternatives and to produce a new cost estimate

**FCS Program 2003 vs 2009**
- Status – program terminated
- Cost estimate grew by $70B
- Schedule grew from 7.5 to 12.3 yrs
- Lines of code grew from 34M to 114M

Source: GAO-10-406

**Weapon Systems Acquisition Reform Act (WSARA)**
2009 requirements for Milestone A approval.
## The QUELCE Solution

### Step 1: Identify Change Drivers and States

<table>
<thead>
<tr>
<th>Change Driver</th>
<th>Nominal State</th>
<th>Alternative States</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope Definition / Mission / CONOPS</td>
<td>Stable</td>
<td>Users added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Additional (foreign) customer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Additional deliverable (e.g. training &amp; manuals)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Production downsized</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scope Reduction (funding reduction)</td>
</tr>
<tr>
<td>Mission / CONOPS Definition</td>
<td>As defined</td>
<td>New condition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New mission</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New echelon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Program becomes Joint</td>
</tr>
<tr>
<td>Capability Definition</td>
<td>Stable</td>
<td>Addition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subtraction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Variance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Trade-offs (performance vs affordability, etc.)</td>
</tr>
<tr>
<td>Funding Schedule</td>
<td>Established</td>
<td>Funding delays tie up resources (e.g. operational test)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FFRDC ceiling issue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Funding change for end of year</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Funding spread out</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Obligated vs. allocated funds shifted</td>
</tr>
<tr>
<td>Advocacy Change</td>
<td>Stable</td>
<td>Joint service program loses participant</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Senator did not get re-elected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change in senior pentagon staff</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Advocate requires change in mission scope</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Service owner different than CONOPS users</td>
</tr>
<tr>
<td>Closing Technical Gaps (CBA)</td>
<td>Stable</td>
<td>Selected Trade studies are sufficient</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Technology does not achieve satisfactory performance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Technology is too expensive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Selected solution cannot achieve desired outcome</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Technology not performing as expected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New technology not testing well</td>
</tr>
</tbody>
</table>

### Domain-Specific Program Change Drivers Identified

- ~~~~ ~~~~ ~~~~ ~~~~ ~~~~ ~~~~ ~~~~ ~~~~
## The QUELCE Solution

### Step 2: Reduce Cause and Effect Relationships via Design Structure Matrix Techniques

1. Identify Change Drivers & States
   - Mission / CONOPS
   - Change in Strategic Vision
   - Capability Definition
   - Advocacy Change

2. Reduce complexity of Cause and Effect relationships via matrix techniques
   - Building Technical Capability & Capacity (CBA)
   - Interoperability
   - Systems Design
   - Interdependency
   - Functional Measures
   - Scope Definition
   - Functional Solution Criteria (measure)
   - Funding Schedule
   - Acquisition Management
   - Program Mgt - Contract Rel (Office)
   - Prop Mgt - DSEV
   - Prop Mgt - Structure
   - Manning at program office

3. Assign Conditional Probabilities to BBN Model

4. Calculate Cost Factor Distributions for Program Execution Scenarios

5. Monte Carlo Simulation to Compute Cost Distribution

### Change Drivers - Cause & Effects Matrix

<table>
<thead>
<tr>
<th>Causes</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission / CONOPS</td>
<td>Capability Definition</td>
</tr>
<tr>
<td>Change in Strategic Vision</td>
<td></td>
</tr>
<tr>
<td>Interoperability</td>
<td>Systems Design</td>
</tr>
<tr>
<td>Interdependency</td>
<td>Functional Solutions</td>
</tr>
<tr>
<td>Functional Measures</td>
<td>Functional Solution Criteria</td>
</tr>
<tr>
<td>Scope Definition</td>
<td>Funding Schedule</td>
</tr>
<tr>
<td>Funding Schedule</td>
<td>Program Mgt - Contract Rel (Office)</td>
</tr>
<tr>
<td>Program Mgt - DSEV</td>
<td>Program Mgt - Structure</td>
</tr>
<tr>
<td>Manning at program office</td>
<td></td>
</tr>
</tbody>
</table>

Capturing interrelationships among change drivers and reducing the complexity of the network
The QUELCE Solution

Step 3: Assign Conditional Probabilities to BBN Model
The QUELCE Solution

Step 4: Calculate Cost Factor Distributions for Program Execution Scenarios

1. Identify Change Drivers & States
2. Reduce Cause and Effect Relationships via Dependency Structure Matrix techniques
3. Assign Conditional Probabilities to BBN Model
4. Calculate Cost Factor Distributions for Program Execution Scenarios
5. Monte Carlo Simulation to Compute Cost Distribution

An example scenario with 4 drivers in nominal state

BBN model enables computation of different scenarios of program execution on cost model factors
The QUELCE Solution

Step 5a: Connecting BBNs to Cost Estimation Models

1. Identify Change Drivers & States
2. Reduce complexity of Cause and Effect relationships via matrix techniques
3. Assign Conditional Probabilities to BBN Model
4. Calculate Cost Factor Distributions for Program Execution Scenarios
5. Monte Carlo Simulation to Compute Cost Distribution

Understand and analyze cost model input factors
Group similar input factors based on empirical analysis in task 3.

Use empirical analysis from Repository as basis to map scale (XL ... EH) of original cost model input factors to scale (1...5) of BBN output factors

<table>
<thead>
<tr>
<th>COCOMO Parameter</th>
<th>Scale Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREC</td>
<td></td>
</tr>
<tr>
<td>FLEX</td>
<td></td>
</tr>
<tr>
<td>RESL</td>
<td></td>
</tr>
<tr>
<td>TEAM</td>
<td></td>
</tr>
<tr>
<td>PMAT</td>
<td></td>
</tr>
<tr>
<td>PERS</td>
<td></td>
</tr>
<tr>
<td>RCPX</td>
<td></td>
</tr>
<tr>
<td>PDFIF</td>
<td></td>
</tr>
<tr>
<td>PREX</td>
<td></td>
</tr>
<tr>
<td>FCIL</td>
<td></td>
</tr>
<tr>
<td>RUSE</td>
<td></td>
</tr>
<tr>
<td>SCED</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COCOMO Parameter</th>
<th>Scale Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREC</td>
<td>1 3 5</td>
</tr>
<tr>
<td>FLEX</td>
<td>1 2 3 5</td>
</tr>
<tr>
<td>RESL</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>TEAM</td>
<td></td>
</tr>
<tr>
<td>PMAT</td>
<td></td>
</tr>
<tr>
<td>PERS</td>
<td></td>
</tr>
<tr>
<td>RCPX</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>PDFIF</td>
<td>1 5</td>
</tr>
<tr>
<td>PREX</td>
<td></td>
</tr>
<tr>
<td>FCIL</td>
<td></td>
</tr>
<tr>
<td>RUSE</td>
<td>1 3 5</td>
</tr>
<tr>
<td>SCED</td>
<td></td>
</tr>
</tbody>
</table>

Product Challenge factors (1=low ... 5=high)

Project Challenge factors (1=low ... 5=high)
The QUELCE Solution

Step 5b: Monte Carlo Simulation to Compute Cost Distribution

1. Identify Change Drivers & States
2. Reduce complexity of Cause and Effect relationships via matrix techniques
3. Assign Conditional Probabilities to BBN Model
4. Calculate Cost Factor Distributions for Program Execution Scenarios
5. Monte Carlo Simulation to Compute Cost Distribution

Monte Carlo simulation using program change factor distributions uses uncertainty on the input side to determine the cost estimate distribution
QUELCE Application and Challenge

Space program piloted QUELCE just after a recent cost estimate

- Anticipated 66 change drivers
- Realized 33 change drivers not considered in latest basis of estimate (BOE)
- Reported only 2/3 of change drivers in BOE were expected to have off-nominal performance
- SEI concluded at least 90% of historical cost growth events could have been identified and prevented by QUELCE

QUELCE workshops in past several years:

- Produced 200-400 change drivers
- Confirmed complexity explosion due to human judgement issue

Experts tend to attribute correlation as cause-effect!
SEI Causal Learning Research Addresses the Challenge

Beginning in 2016, SEI research focused on causal learning

- Causal learning appeared capable of distinguishing “spurious” correlation from causal-based correlation
- Causal learning did not require experimentation and thus, could operate on observational data such as historical cost research data
- Belief was that causal learning could help trim down the overwhelming list of software cost change driver relationships hypothesized by the experts

Causal learning was a novel leap from predominant use in medical research to use in software cost research

Causal learning has grown in use to six SEI research projects in past 3 years
SEI Research Journey from Uncertainty to Causality

PREDICTION
(Correlation & Regression)

What are the subjective uncertainties to model in cost estimates?

WSARA 2009
QUELCE
Affordability

PRESCRIPTION
(Causal Search Collaboration with CMU Dept of Philosophy)

How to control cost through...

... team and leadership factors?

... organizational and enterprise factors?

... acquisition and technology factors?

SCOPE LSI (3 Year LSI)

Why Does Software Cost So Much? (LENS)

Pre-2017
2017
2018
2019
2020
Motivation for Causal Learning

Controlling costs requires knowing which “independent factors” actually cause cost outcomes, so that we may change cost in a predictable manner.

Just as correlation may be fooled by spurious association, so can regression.

We must move beyond correlation to causation, if we want to make use of cause and effect relationships.

We can now evaluate causation without expensive and difficult experiments.

Establishing causation with observational data remains a vital need and a key technical challenge, but is becoming more feasible and practical.
Often, an excluded common cause results in a misinterpretation of correlation!

So...to prevent shark attacks, we should limit the number of ice cream cones sold, right?

Does high correlation imply causation?
Different Uses for Correlation versus Causation

<table>
<thead>
<tr>
<th>Correlation</th>
<th>Causation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifying &amp; identifying</td>
<td>Influencing &amp; acting</td>
</tr>
<tr>
<td>Informational value of different</td>
<td>Using evidence to guide policy or actions</td>
</tr>
<tr>
<td>evidence</td>
<td></td>
</tr>
<tr>
<td>Prediction &amp; reasoning given</td>
<td>Prediction &amp; reasoning given given</td>
</tr>
<tr>
<td>observations</td>
<td>interventions</td>
</tr>
<tr>
<td>Probable explanations for some</td>
<td>Ways to produce or prevent an</td>
</tr>
<tr>
<td>event or issue</td>
<td>event or problem</td>
</tr>
</tbody>
</table>
Landscape of Causal Learning

**Causal Discovery**
using CMU Tetrad, which implements a variety of algorithms

**Directed Acyclic Graph Model**

- A → B → C → D → E → F

**Prior Knowledge & Observational Data**

**Formulate Hypotheses**
using domain knowledge and prior scholar publication

**Estimated SEM Model**

- A → −2.75 → B → +3.19 → C → +1.02 → D → +6.51 → F
- Y
Causal Learning as a Discipline
Causal Search Algorithms and Tooling

Step 4: Initiate a causal search by inserting a Search box with a connection from the Data box.

Select from over a dozen different algorithms for causal discovery.

The PC algorithm, named for Peter Spirtes and Clark Glymour (CMU), its creators, is one of the most widely used and known causal search algorithms. https://www.ccd.pitt.edu/
Causal Search Outputs

Comparison of causal graphs unhealthy vs. healthy for a cyber physical system (CPS)

- In an unhealthy CPS, outcomes have many internal causes, including wear and tear from earlier missions
- In a healthy CPS, outcomes are almost entirely driven by the environment

Key:
- **O1**: Outcome
- **E2**: Environment variable
- **F3**: Internal Performance factor
Causal Estimation Techniques

**Structural Equation Modeling**
Multivariate modeling: may involve measured and latent factors; with a simultaneous set of regression equations; where factors can be independent and dependent within the same model

**Instrumental Variables**
Finding a third factor correlated to your independent factor but not directly related to your dependent factor; useful when unmeasured confounders might exist between your independent and dependent factors

**Propensity Scoring**
An approach to matching and trimming within a data set to evaluate cause effect relationships

**Causal Algebra Do-Calculus**
A new causal algebra by Pearl et al to compute causal effects using three new rules of do-calculus in addition to traditional probability manipulation rules
Example SEI Causal Learning Cost Research Results

Controlling Size: Only 2 of 4 code size measures appear causal on effort and quality

Controlling Complexity: Only 1 of 3 factors appears causal on performance and quality

Controlling Architecture Violations: Only 1 of 4 violation factors appears causal on quality

Controlling Team Performance: Only 1 of 20+ factors appears causal on quality and cost

Causal search may provide useful feedback:
- Presence of causal links
- Absence of causal links
Other SEI Causal Learning Research: Simulation and Test

**Problem**
Lack of accredited simulators

**Technical Challenge**
Experts unsure of the expected result for a given simulated scenario

**Research Questions**
1. Scale up metamorphic testing to test very complex DoD systems?
2. Machine learning to identify metamorphic relations for testing?
3. Causal learning to drive metamorphic relations testing?
Other SEI Causal Learning Research: Sustainment

**Problem**
Unscheduled maintenance creates unacceptable costs

**Technical Challenge**
Traditional statistical approaches helpful, but insufficient

**Research Questions**
1. Machine learning of engine sensor and control data improve scheduled maintenance?
2. Causal learning integrated with machine learning add value?
Future SEI Causal Learning Research: Examples

**Affordable**
- Acquisition practice improved using causal models
- Cost estimates and budget execution using causal models
- Simpler but more effective ROI models based on causal factors (e.g. Model Based Engineering, Architecture practice, Technical Debt)

**Trustworthy**
- Causal factors threatening cyber defenses
- Causal factors limiting resilience
- CL combined with ML tools for more affordable and trustworthy SW technologies (e.g. DOD initiative in Digital Engineering)
- Expected behavior from autonomous systems (e.g. “Explainable AI”; Jensen, UMass)

**Capable**
- Causal drivers of workforce performance
- SW architecture strategies and tactics driving system performance
- More efficient experimentation of technical solutions
- Increased realism of complex system simulation
- Autonomous systems controlling consequences
- Machine learning with human-like intelligence (e.g. “Strong AI”; Pearl, “The Book of Why”)

**Timely**
- Causal structures from DevOps information stream to control process and lifecycle
- Agile causal systems situationally prescribe practices aligned with goals
- Project risks controlled through causal structures of project parameters
A Vision for Software Cost Research

Prescriptive Cost Guidance

• Guide cost and pricing negotiations
• Identify interventions during program execution
• Formulate causal-based lessons learned

Explainable AI Cost Estimation

• Increase transparency of future AI-based cost estimation and management
• Increase trust in such solutions and reduce bias

Cost Model Transferability

• Determine when a cost model may be safely used in a new context or situation
• Identify in advance when cost models may not be trusted
Call to Action

Demand causal knowledge to guide interventions

Engage with SEI causal researchers studying software cost

Motivate data collection and sharing for more repeatable and reproducible causal studies

Build causal learning competency in your organization
Contact Information

Presenter / Point(s) of Contact

Robert Stoddard
Dr. Mike Konrad

Email:
rws@cmu.edu
mdk@sei.cmu.edu

Telephone:
+1 412.268.1121 (Bob)
+1 412.268.5813 (Mike)

Other SEI Cost Research Team Members
Chris Miller
Bill Nichols
Sarah Sheard
Dave Zubrow

Other CMU Cost Research Contributors
Madelyn Glymour
Joe Ramsey
Kun Zhang

USC Cost Research Contributors
Jim Alstad
Barry Boehm
Anandi Hira
Core Principles of Data Visualization and Presentation Skills

I’m here to help you do a better job to analyze, visualize, and present your data and research. I conduct on-site and online workshops and webinars, detailed consulting and coaching, and keynote presentations to organizations around the world. My data visualization workshops offer attendees the core skills needed to create better visualizations and how to do so in software tools like Excel, PowerPoint, R, and Tableau.

Jonathan Schwabish
@jschwabish
jschwabish@gmail.com
www.policyviz.com
Who am I?

Adam James is a Senior Analyst at Technomics, Inc., where he serves as a strategic leader and data scientist for the newly established Technomics Innovation Lab (TIL). In this role, Adam provides his expertise to develop modern, innovative solutions to address new – and old – cost analysis problems.

Adam’s current focus is helping clients extract value from datasets of varying complexity. Recently he helped the Army analyze traditional cost estimating data sources such as the Cost and Software Data Report (CSDR) and Contracts using modern tools and data science techniques. Prior work includes sharing lead authorship of the Joint Agency CER Development Handbook.

Adam earned a M.S. in Statistics from Virginia Tech in 2012. He also has a B.S. from Virginia Tech with a double major in Mathematics and Statistics. He received the International Cost Estimating and Analysis Association’s (ICEAA) 2016 Technical Achievement Award. Adam also has been active in contributing to the community knowledge base, winning best paper in both the “Analysis & Modeling, Machine Learning” and “Methods, Data Collection & Management” categories at ICEAA in 2019.1,2

The Future is Data

1. Data are not being utilized to their full potential
   - Too voluminous to handle in spreadsheets
   - Too diverse to quickly understand
   - New data are being collected at an exponential rate

2. How we think of data must change
   - Machine readable is more desirable than human readable
   - The right tool for the right job – out of spreadsheets into the future

3. All analysts must understand data structure
   - Efficient for a machine to process
   - Enables rapid exploitation of data
   - Necessary for quantitative analysis
What is Structure?

**structured data**
aligns nicely and fits into fixed tables

**unstructured data**
does not (e.g., text blocks, images, audio, video)
This presentation is **not about data storage**
- Many products, tools, techniques and companies focus on this topic
- Very important, but largely an IT function

This presentation is through the view of the “**desktop user**”
- Convert data from a raw storage format into something usable
- Analytically focused

- Tips and best practices for **working with structured data**

- Approaches to converting **unstructured (text) data** into structured data
Structured Data

best practices, tips, and tricks
Standards help prevent a continual reinvention of the wheel.

There are many examples of data structure standards:
- Database engineers have standards of working with **SQL**
- **Templates** provide consistent ways of working with specific data

The **tidy** data philosophy has become a standard in the R universe
- Forms the foundation of most “data science” online courses
- Creates a robust standard of how **any data** should be formatted in **any tool**
- Does not necessarily make for efficient data storage
- Does not eliminate all other post processing (though usually makes it much easier)

---

2 [https://r4ds.had.co.nz/tidy-data.html](https://r4ds.had.co.nz/tidy-data.html)
Rows and Columns

**Rows** represent unique data observations (“records”), **columns** represent variables (“fields”).

<table>
<thead>
<tr>
<th>System</th>
<th>WBS Element</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truck A</td>
<td>Engine Cost</td>
<td>50,000</td>
</tr>
<tr>
<td></td>
<td>Remaining Cost</td>
<td>150,000</td>
</tr>
<tr>
<td></td>
<td>PM Cost</td>
<td>1,500</td>
</tr>
<tr>
<td></td>
<td>Number of Units</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Total Cost</td>
<td>2,015,000</td>
</tr>
<tr>
<td>Truck B</td>
<td>Engine Cost</td>
<td>40,000</td>
</tr>
<tr>
<td></td>
<td>Remaining Cost</td>
<td>120,000</td>
</tr>
<tr>
<td></td>
<td>PM Cost</td>
<td>2,500</td>
</tr>
<tr>
<td></td>
<td>Number of Units</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Total Cost</td>
<td>812,500</td>
</tr>
</tbody>
</table>

### Bad

### Better

<table>
<thead>
<tr>
<th>System</th>
<th>Metric</th>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truck A</td>
<td>Unit Cost</td>
<td>Engine</td>
<td>50000</td>
</tr>
<tr>
<td>Truck A</td>
<td>Unit Cost</td>
<td>Remaining</td>
<td>150000</td>
</tr>
<tr>
<td>Truck A</td>
<td>Unit Cost</td>
<td>PM</td>
<td>1500</td>
</tr>
<tr>
<td>Truck A</td>
<td>Unit Cost</td>
<td>Surface Vehicle</td>
<td>201500</td>
</tr>
<tr>
<td>Truck A</td>
<td>Quantity</td>
<td>Surface Vehicle</td>
<td>10</td>
</tr>
<tr>
<td>Truck B</td>
<td>Unit Cost</td>
<td>Engine</td>
<td>40000</td>
</tr>
<tr>
<td>Truck B</td>
<td>Unit Cost</td>
<td>Remaining</td>
<td>120000</td>
</tr>
<tr>
<td>Truck B</td>
<td>Unit Cost</td>
<td>PM</td>
<td>2500</td>
</tr>
<tr>
<td>Truck B</td>
<td>Unit Cost</td>
<td>Surface Vehicle</td>
<td>162500</td>
</tr>
<tr>
<td>Truck B</td>
<td>Quantity</td>
<td>Surface Vehicle</td>
<td>5</td>
</tr>
</tbody>
</table>
A column (or variable) should only contain **one** piece of information.

**Bad**

1. Surface Vehicle System
   1.1 Variant A
      1.1.1 Surface Vehicle
         1.1.1.1 Engine
         1.1.1.2 Remaining Vehicle
   1.2 Variant B
      1.2.1 Surface Vehicle
         1.2.1.1 Engine
         1.2.1.2 Remaining Vehicle

**Better**

<table>
<thead>
<tr>
<th>Original WBS</th>
<th>Modified WBS</th>
<th>Element</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Surface Vehicle System</td>
<td></td>
</tr>
<tr>
<td>1.1.1</td>
<td>1.1</td>
<td>Surface Vehicle</td>
<td>Variant A</td>
</tr>
<tr>
<td>1.1.1.1</td>
<td>1.1.1</td>
<td>Engine</td>
<td>Variant A</td>
</tr>
<tr>
<td>1.1.1.2</td>
<td>1.1.2</td>
<td>Remaining Vehicle</td>
<td>Variant A</td>
</tr>
<tr>
<td>1.2.1</td>
<td>1</td>
<td>Surface Vehicle</td>
<td>Variant B</td>
</tr>
<tr>
<td>1.2.1.1</td>
<td>1.1.1</td>
<td>Engine</td>
<td>Variant B</td>
</tr>
<tr>
<td>1.2.1.2</td>
<td>1.1.2</td>
<td>Remaining Vehicle</td>
<td>Variant B</td>
</tr>
</tbody>
</table>
Other Data Tips (1 of 2)

Use **variable names** that any tool can use

<table>
<thead>
<tr>
<th>Bad Name</th>
<th>Better Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Breakdown Structure</td>
<td>WBS</td>
</tr>
<tr>
<td>% Complete</td>
<td>PercComp</td>
</tr>
<tr>
<td>Cost (TY $K)</td>
<td>CostTY_K</td>
</tr>
<tr>
<td>Unit Cost (FY18)</td>
<td>UnitCost_FY18</td>
</tr>
<tr>
<td>1970</td>
<td>Cost_1970</td>
</tr>
</tbody>
</table>

Be explicit in preserving **order**

<table>
<thead>
<tr>
<th>Month</th>
<th>ID_Month</th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>1</td>
<td>January</td>
</tr>
<tr>
<td>February</td>
<td>2</td>
<td>February</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>November</td>
<td>11</td>
<td>November</td>
</tr>
<tr>
<td>December</td>
<td>12</td>
<td>December</td>
</tr>
</tbody>
</table>

Store data **values** and report data **formats**

<table>
<thead>
<tr>
<th>Value</th>
<th>Formatted</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000</td>
<td>100,000</td>
</tr>
<tr>
<td>2018-10-05</td>
<td>October 5, 2018</td>
</tr>
<tr>
<td>2695.255648</td>
<td>$2,695.26</td>
</tr>
</tbody>
</table>

Take advantage of **sparse data** when possible

<table>
<thead>
<tr>
<th>wbs</th>
<th>rec</th>
<th>non_rec</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.3</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.4</td>
<td>2000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>wbs</th>
<th>r_nr</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3</td>
<td>rec</td>
<td>1000</td>
</tr>
<tr>
<td>1.4</td>
<td>non_rec</td>
<td>2000</td>
</tr>
</tbody>
</table>
Other Data Tips (2 of 2)

Avoid storing **redundant information**

- ✓ Only store child elements
- ✗ Do not store subtotals / totals
- ✗ Do not store calculated variables

Use **intermediate** tables for calculations

Be mindful of **data types**

- Numeric, date, and text
- Excel will make (sometimes wrong) assumptions

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Text Format</td>
<td>General Format</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>4</td>
<td>1.1.1</td>
<td>1.1.1</td>
</tr>
<tr>
<td>5</td>
<td>1.1.2</td>
<td>1.1.2</td>
</tr>
<tr>
<td>7</td>
<td>1.9</td>
<td>1.9</td>
</tr>
<tr>
<td>8</td>
<td>1.10</td>
<td>1.1</td>
</tr>
<tr>
<td>9</td>
<td>1.11</td>
<td>1.11</td>
</tr>
</tbody>
</table>
A grammar is the rules of a language – or a set of instructions
- Data forms a collection of nouns
- A small set of verbs operate on the data

dplyr is a package in R which proposes and implements the grammar
- Only 5 (+1) primary verbs create a powerful, flexible framework assuming “tidy” data

The grammar outlines a thought process transferable to any tool (even Excel!)

1 https://dplyr.tidyverse.org/
Primary Verbs

1) **mutate** – adds new variables that are functions of existing variables
2) **select** – picks variables based on their name
3) **filter** – picks cases based on their values
4) **summarize** – reduces multiple values down to a single summary
5) **arrange** – changes the ordering of rows

And the bonus verb **group_by** performs any operation “by group”

---

1 https://dplyr.tidyverse.org/
A Simple Example

**Goal:** to view average dollars (applying escalation) and hours

<table>
<thead>
<tr>
<th>Obs</th>
<th>fiscalyear</th>
<th>metric</th>
<th>engine</th>
<th>value</th>
<th>escalation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2010</td>
<td>cost</td>
<td>diesel</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1992</td>
<td>cost</td>
<td>gas</td>
<td>8</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>2011</td>
<td>cost</td>
<td>diesel</td>
<td>2</td>
<td>1.02</td>
</tr>
<tr>
<td>4</td>
<td>2006</td>
<td>cost</td>
<td>diesel</td>
<td>19</td>
<td>0.96</td>
</tr>
<tr>
<td>5</td>
<td>2008</td>
<td>hours</td>
<td>diesel</td>
<td>22</td>
<td>n/a</td>
</tr>
<tr>
<td>6</td>
<td>1989</td>
<td>hours</td>
<td>diesel</td>
<td>14</td>
<td>n/a</td>
</tr>
<tr>
<td>7</td>
<td>2016</td>
<td>hours</td>
<td>diesel</td>
<td>16</td>
<td>n/a</td>
</tr>
<tr>
<td>8</td>
<td>2014</td>
<td>hours</td>
<td>gas</td>
<td>9</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**Steps:**

1) Divide the dataset into cost and hours
2) For each of the cost values, adjust for escalation
3) Calculate the average for the escalation adjusted costs
4) Calculate the average for the hours
A Simple Example – Applying the Grammar

**Goal:** to view average dollars (applying escalation) and hours

<table>
<thead>
<tr>
<th>Obs</th>
<th>Fiscal Year</th>
<th>Metric</th>
<th>Engine</th>
<th>Value</th>
<th>Escalation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2010</td>
<td>Cost</td>
<td>Diesel</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1992</td>
<td>Cost</td>
<td>Gas</td>
<td>8</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>2011</td>
<td>Cost</td>
<td>Diesel</td>
<td>2</td>
<td>1.02</td>
</tr>
<tr>
<td>4</td>
<td>2006</td>
<td>Cost</td>
<td>Diesel</td>
<td>19</td>
<td>0.96</td>
</tr>
<tr>
<td>5</td>
<td>2008</td>
<td>Hours</td>
<td>Diesel</td>
<td>22</td>
<td>n/a</td>
</tr>
<tr>
<td>6</td>
<td>1989</td>
<td>Hours</td>
<td>Diesel</td>
<td>14</td>
<td>n/a</td>
</tr>
<tr>
<td>7</td>
<td>2016</td>
<td>Hours</td>
<td>Diesel</td>
<td>16</td>
<td>n/a</td>
</tr>
<tr>
<td>8</td>
<td>2014</td>
<td>Hours</td>
<td>Gas</td>
<td>9</td>
<td>n/a</td>
</tr>
</tbody>
</table>

1: Vehicle data
2: mutate value_adj =
3: if (metric = cost) then (value / escalation)
4: else value
5: group_by metric
6: summarize avg =
7: mean of value_adj
Unstructured Data

*working with blobs of text*
In this presentation we are focusing on **unstructured text**
- Text mining can uncover a lot of information very quickly
- In the software engineering and cost estimating world, unstructured data are everywhere

**How can we better use these data?**
- Can we read requirements documents automatically?
- Can we analyze user stories for predictive purposes?

There are many steps in this process
- The following slides highlight some fundamental techniques
- Software tools applied to this problem can vary – we will focus on a **tidy representation**

Will use data from the PURE dataset as an example (specifically KeePass)

---

Some Basics

- Collections of text are stored in **documents**
- Include **ID fields** for useful information (e.g., document number, chapters, etc.)

<table>
<thead>
<tr>
<th>doc_id</th>
<th>chapter_id</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>This document includes software requirements for KeePass Password Safe, release number 1.10. KeePass…</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>KeePass consists of a database which contains data for one or more users. Each user’s data are divided in…</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>System features are organized by use cases and functional hierarchy so that the main functions of the sys…</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>User interface includes various forms and windows. The main database window consists of the main menu b…</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>When a password is copied, it remains on memory for only 10 seconds. If in the meanwhile it i…</td>
</tr>
</tbody>
</table>
Cleaning up Data

- Remove **special characters** (e.g., line breaks, strange encodings from a PDF)
- Replace **symbols** (e.g., @, #, $, %)
- Replace **contractions** (e.g., “isn’t” to “is not”)
- Convert **numbers** and ordinal terms (e.g., “1” to “one”, “2nd” to “second”)
- Remove **punctuation**

Original Text:

Our 1st text cleaning (e.g., clean-up, transforming, standardizing, # conversion) isn't hard w/ the right tool.

Clean Text:

Our **first** text cleaning (eg, clean-up, transforming, **standardizing**, **number** conversion) **is not** hard **with** the right tool
This document includes software requirements for KeePass Password Safe, release number 1.10. KeePass Password Safe is an OSI Certified Open Source Software distributed under the terms of the GNU General Public License Version 2 or under.

**sentences** have meaning as independent statements

<table>
<thead>
<tr>
<th>doc_id</th>
<th>chapter_id</th>
<th>sent_id</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>this document includes software requirements for keepass password safe, release number</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>keepass password safe is an osi certified open source software distributed under the terms of the gnu general public license version or under</td>
</tr>
</tbody>
</table>
## Tokenization (Words/N-Grams)

### Words are a common token

<table>
<thead>
<tr>
<th>doc_id</th>
<th>chapter_id</th>
<th>sent_id</th>
<th>word</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>this</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>document</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>release</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>number</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>keepass</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>password</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

When using words as the token, we often remove **stop words** (e.g., the, and, it, etc.)

### N-grams are groups of “n” words

<table>
<thead>
<tr>
<th>doc_id</th>
<th>chapter_id</th>
<th>id_sent</th>
<th>ngram_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>this document</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>document includes</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>release number</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>keepass password</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>password safe</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>safe is</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Standardizing the Language (Example 1)

- **Stemming** truncates to a base and then completes the base using a word vector
- **Lemmatization** returns a word to its dictionary form

<table>
<thead>
<tr>
<th>Original Word</th>
<th>Stem</th>
<th>Completed Stem</th>
<th>Lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>add</td>
<td>added</td>
<td>add</td>
</tr>
<tr>
<td>added</td>
<td>ad</td>
<td>added</td>
<td>add</td>
</tr>
<tr>
<td>adding</td>
<td>ad</td>
<td>added</td>
<td>add</td>
</tr>
<tr>
<td>adds</td>
<td>add</td>
<td>added</td>
<td>add</td>
</tr>
</tbody>
</table>
Standardizing the Language (Example 2)

- **Stemming** is usually performed algorithmically
- **Lemmatization** requires a dictionary (not a problem) but is generally superior

<table>
<thead>
<tr>
<th>Original Word</th>
<th>Stem</th>
<th>Completed Stem</th>
<th>Lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>copied</td>
<td>copi</td>
<td>copied</td>
<td>copy</td>
</tr>
<tr>
<td>copy</td>
<td>copi</td>
<td>copied</td>
<td>copy</td>
</tr>
<tr>
<td>copies</td>
<td>copi</td>
<td>copied</td>
<td>copy</td>
</tr>
<tr>
<td>recopy</td>
<td>recopi</td>
<td>copy</td>
<td></td>
</tr>
</tbody>
</table>
Example Visualizations (1 of 3)
These graphics show visually how the computer “reads” the documents.

- Structuring data is a **starting point** to more advanced natural language processing.
Prepare for Modeling

A **document-term matrix** is a way to represent text numerically.

<table>
<thead>
<tr>
<th>doc_id</th>
<th>sent_id</th>
<th>audience</th>
<th>backup</th>
<th>can</th>
<th>case</th>
<th>certify</th>
<th>class</th>
<th>computer</th>
<th>contain</th>
<th>convention</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</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>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>3</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>
</tr>
<tr>
<td>1</td>
<td>4</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>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>6</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>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>0</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>
</tr>
<tr>
<td>1</td>
<td>8</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>1</td>
</tr>
<tr>
<td>1</td>
<td>9</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>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

*not the complete matrix*
Tip of the Iceberg

- This is a very basic and introductory approach

- More complex strategies are used in practice
  - Syntax parsing
  - Semantics
  - Machine learning enhancements

- Data structure is important – and even the unstructured has structure!
Closing Thoughts

Until the community as a whole commits to better data management, the true power and value of data science will never be realized.

Data source: Google Trends (https://www.google.com/trends)
Calibrating COCOMO® II for Functional Size Metrics

ANANDI HIRA, BRAD CLARK, BARRY BOEHM
COCOMO® II Model

Software product size estimate (in KSLOC)

Software product, process, computer, and personal attributes

Software reuse, maintenance, and increment parameters

Software Project data

COCOMO® Estimates:
• Resource
• Equivalent Size
• Reuse impact
• Re-Engineering or conversion
• Maintenance

Local calibration to organization’s data

Software development and maintenance:
• Costs (effort)
• Schedule estimates
• Distributed by phase, activity, increment

COCOMO® is an open and free model

COCOMO® is an open and free model
## Size Metrics’ Level of Abstraction

<table>
<thead>
<tr>
<th>Requirement Levels</th>
<th>Size Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary Goals</td>
<td>Story Points</td>
</tr>
<tr>
<td>User Goals</td>
<td>Use Cases</td>
</tr>
<tr>
<td></td>
<td>Use Case Points (UCPs)</td>
</tr>
<tr>
<td>Sub-Functions</td>
<td>IFPUG Function Points (FPs)</td>
</tr>
<tr>
<td></td>
<td>COSMIC Function Points (CFPs)</td>
</tr>
<tr>
<td></td>
<td>Source Lines of Code (SLOC)</td>
</tr>
</tbody>
</table>
Considering 2 Prominent Functional Size Methods

**IFPUG SOFTWARE MODEL**

**COSMIC SOFTWARE MODEL**
PM = A × Size\(^{(B + 0.1 \times (\sum SF))}\) × \(\prod EM\)

Exponent ranges from 0.9 to 1.2, with 1.0991 as default

Where
- PM = Software development effort (in Person-months)
- Size = Size in Thousand Equivalent Source Lines of Code (KESLOC)
- A = Calibrated Productivity constant (ESLOC/PM)
- B = Calibrated Exponent constant
- SF = Scale Factors – have exponential effect
- EM = Effort Multipliers – have multiplicative effect
Objective/Goal of My Research

Adjust COCOMO® II parameters to result in a model that gives accurate estimates using either IFPUG or COSMIC Function Points

• Scale Factors – how quickly effort grows with respect to size
  • Precedentedness, Development Flexibility, Team Cohesion, Risk and Architecture Resolution, and Process Maturity

• Effort Multipliers – if necessary
  • Perhaps Product drivers, such as Product Complexity (CPLX)?

2 Steps:
• Opinions of improved parameter values
• Bayesian Analysis to combine opinion and regression
Participation Requirements

• Familiar with software development at project level, either as project lead, estimator, or engineer.

• Experience with either or both IFPUG/COSMIC Function Points
  • Or other types of functional size metric

• Experience estimating software development cost is very helpful

• Experience with COCOMO® II or other software estimation models is helpful.
Please Participate by Doing 1 or More of These:

- Meet Barry Boehm, Brad Clark, or myself during the breaks or lunch to provide feedback.
- Meet Barry Boehm, Brad Clark, or myself to fill out a voting sheet for parameter values.
  - Wednesday, September 18 1:30-5pm: Calibrating COCOMO® II for Functional Size Metrics
Cost of Software Obsolescence Resolution of Real-Time Software

20th August 2019
Sanathanan Rajagopal – QinetiQ Fellow
Estimating Manager- Cyber, Information and Training

- ICEAA Certified Cost Estimator / Analyst with the Parametric Specialism (ISPA)
- Chairman and member of the board of the Society for Cost Analysis and Forecasting (SCAF)
- Fellow of Defence Equipment and Support, MoD, UK
- Recipient of the ICEAA Technical Achievement of the year 2017 award
- Member of EPSRC Centre of Innovative Manufacturing (Through Life Engineering Services, Cranfield University)

Publications:
- Supported in the development of modules for ICEAA Software CEBoK
- Contributor to a study undertaken by Institute for Defence Analysis (US DoD) sponsored by Defence Logistics Agency (DLA, DoD) on Obsolescence titled “A Research and Development Investment Portfolio for Diminishing Manufacture Sources and Material shortages”
COMMERCIAL IN CONFIDENCE

12 Empire Test Pilot School students have become astronauts including Major Tim Peake.

85+
locations worldwide

1,300+
patents (including 300+ pending)

5
5 fundamental mobile phone technologies including Touch Screen and Liquid Crystal Displays (LCD) developed by QinetiQ experts

40
40 organisations, including the Royal Navy involved in 6 weeks of operations during the Unmanned Warrior Exercise

50+
including 50+ unmanned vehicles operating in the air, land and sea

We are QinetiQ

£833m
FY2018 revenue

6,000+
people with unique science and engineering expertise

3 secs
Every 3 seconds a Boeing aircraft takes off or lands that has been tested in QinetiQ’s low speed Wind Tunnel

1,850
1,850km of the TANAP pipeline will be protected by OptaSense®

16
Our Ocean Basin in Gosport, UK contains enough water to fill 16 Olympic swimming pools
## Agenda

1. Research Aim
2. Definitions
3. Introductions
4. Software Obsolescence Cost Analysis Framework
5. Software Obsolescence Resolution Cost Optimization Model
6. Summary
7. Conclusion
Software Obsolescence

Research Aim
“To develop a cost analysis framework to estimate the cost of Software Obsolescence Resolution of a bespoke real-time software in defence and aerospace”
Software Obsolescence

Definitions
Software Obsolescence is defined as “what happens when the original and authorised third party ceases to provide support with regular update, upgrade, fixes or due to the changes in target or operating environment, systems or hardware which makes the software unusable”

-S Rajagopal et al; (2014)
## Software Obsolescence vs Software Maintenance

<table>
<thead>
<tr>
<th>Software Maintenance</th>
<th>Software Obsolescence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug fixes</td>
<td>Replacement of entire application if need be to a new one</td>
</tr>
<tr>
<td>To address fault/Failures, security patches etc.</td>
<td>To address the issues with the application in totality</td>
</tr>
<tr>
<td>Maintenance is the review of the stored files to ensure they are still useable</td>
<td>Solves unavailability of fixes, licenses, permission and upgrades</td>
</tr>
<tr>
<td>Software maintenance takes care of the current versions to ensure that its up and running and meeting the requirements</td>
<td>Software Obsolescence management looks forward the industry standards and other software to continue supportability of the software</td>
</tr>
<tr>
<td>Maintenance deals with the upgrading the software to enhance capability</td>
<td>Obsolescence management deals with enforced changes in the environment</td>
</tr>
</tbody>
</table>
Software Obsolescence

Introduction
Introduction

The need for a Software Obsolescence Cost Analysis Framework

- High Dependency on Software in Defence and Aerospace
- Constant Changes in Hardware
- Software Obsolescence is unavoidable
- Software is a key(cost and Schedule) Driver
- Long Support Contract
Software Obsolescence Cost Analysis Framework
Following process was undertaken to develop the Framework

- 6 x Defence Case Studies
- 120 Responses from Online Survey
- Literature Searches
- Over 20 SME Interviews
- 6 x Cognitive Case Studies
Software Obsolescence Cost Analysis Framework

The framework has the following attributes

– This framework is in its final iterations.
– This framework’s foundation is based on the Literature Searches, Case Studies, Online Survey results, SME Interviews and Cognitive Case Studies.
– This framework has several attributes that can be mapped across from and to, to the software estimating principals.
– This framework looks at the Cost Risk and Uncertainty.
Software Obsolescence Cost Analysis Framework
1. Identification of the software Obsolescence Resolution Approach
2. Compile Software Obsolescence Strategy
3. Generates Key Cost Drivers and Potential Risk/Uncertainties
Software Obsolescence Management Level

Reactive

- S/W Obs Mgt Level 1
  - Deal with Software Obsolescence Reactively
  - No Obsolescence Management Strategy
  - Freeze and do nothing
  - CMMI Level 1
  - Low TOMCAT Score

- S/W Obs Mgt Level 2
  - No Software Obsolescence Management Strategy
  - Reactive but dealing with Software Obsolescence by reverse engineering and code conversion
  - CMMI Level 2
  - Low TOMCAT Score

- S/W Obs Mgt Level 3
  - Deploy software Obsolescence Monitoring process or tool if available
  - Monitoring software Supply Chain
  - Monitoring skills and technological insertions
  - Deploy software Obsolescence professional
  - Monitoring software Obsolescence Proactively
  - CMMI Level 3
  - Medium TOMCAT Score

- S/W Obs Mgt Level 4
  - Deploy S/W Obs Mgt Strategy
  - Proactive Mgt of update, upgrade and migration
  - Employ and deploy appropriate skills in-house
  - Mgt of Software Supply Chain and monitoring any technology insertions
  - Escrow agreement in place or third party partnership in place
  - CMMI Level 4
  - Medium TOMCAT Score

- S/W Obs Mgt Level 5
  - Proactive Mgt of S/W Obs
  - Deploy effective Mgt of S/W Obs Mgt Strategy
  - Continuous Monitoring of S/W Obs
  - Management of S/W obsolescence as BAU
  - Considering software Obsolescence at the design and development stages
  - CMMI Level 5
  - High TOMCAT Score

Proactive

Transitions
Software Obsolescence Complexity Level

Proactive

High Complexity
- Custom Software
- Real Time Software
- Custom Middleware
- Custom Glue Code
- Safety Critical Software
- High Requalification and testing requirement
- Require high end hardware
- Single Source
- Low Reliable suppliers
- Machine Code, 1st and 5th generation language
- No backward or forward compatibility
- Not easy to emulate

Medium Complexity
- Medium level of requalification and testing
- Non Safety Critical Software
- 2nd and 3rd generation language
- Readily available but requires minor re-design
- Easy to adapt
- Easy to emulate

Low Complexity
- Standard software
- Standard middleware
- Low requalification and testing requirement
- 4th generation language
- Readily available
- Backward and Forward compatible

Reactive

High Risk Software

Low Risk Software
Software Obsolescence Key Cost Drivers

Key drivers are determined by the following

- The resolution approach
- The level of integration (these are both software to software and software to hardware integration)
- Software/System multipliers
- Type of platforms
- Testing and requalification
- Level of modification required
Software Obsolescence and Maintenance
Types of Software Maintenance

There are four different types of Software Maintenance. (from ISBSG)

• Perfective Maintenance: - Perfective maintenance is the modification of a software application, after delivery, to improve performance or maintainability

• Preventative Maintenance: - The modification of a software application after delivery to detect and correct latent faults in the software product before they become effective faults

• Corrective Maintenance: - The reactive modification of a software product performed after delivery to correct discovered problems.

• Adaptive Maintenance: - Enhancements necessary to accommodate changes in the environment in which a software product must operate
Maintenance Vs Obsolescence

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventative Maintenance

- Reactive Management
  - High Risk Software

- Proactive Management
  - Low Risk Software
Case Study – Software Maintenance Vs Software Obsolescence

• On Software with larger applications, about 95% of the time is spent on corrective maintenance.
• This indicates that more time is spent on reactive management of the software.
• In order to reduce software obsolescence, more time should be spent on preventive and perfective maintenance.
Software Obsolescence Resolution Cost Optimisation Model (SRCOM)
Integration of all the Models

Software Obsolescence Data Collection Form

Software Obsolescence Resolution Model

Software Obsolescence Resolution Cost Model

Software Obsolescence Optimization Model

Optimized Through Life Resolution Profile with Cost
Model build ups

• The model is in three parts
  – Data Collections
  – Resolution model
  – Cost model /Optimisation Model

• The data collection form is based on the back of the literature review, case studies, expert interviews and cognitive case studies.

• There is about 150 questions been asked on these forms.

• High level validation on these questions have been undertaken by,
  – Cross checking these with the outputs from Case studies and online surveys
  – Questions were send out to the Software Project managers in QinetiQ
This model will recommend the best resolution approach based on the cost.

Optimisation is undertaken using MatLab Optimisation toolbox.

This model is at a very early stages of its development and

This methodology involve diagrammatical representation of the cost.

Diagram represents decompositions of cost from output node through successive levels to individual inputs nodes.

This decomposition is the focus of the diagram and technique.

The diagram will be converted into a set of inputs and calculations.

Each nodes have inputs (Min, ML and Max).
Software Obsolescence Resolution Cost Optimisation Model - Nodes

- System/WBS/Component/PBS Nodes
- Resolution Profile Level 1
- Resolution Profile Level 2
- Resolution profile Level 3/Component Nodes
- Notes and Audit Nodes
- Information only Nodes
Software has several components and each component undergoes different obsolescence resolution profiles which can be diagrammatically represented using above methodology.

Diagram represents decomposition of cost from output node (on the left) through successive levels to individual input nodes (on the right);

This decomposition is the focus of the diagram and technique.
• Input values are entered on the right of the diagram and aggregate through the model to give the vignette cost.

• Each node has a specific formula for combining inputs, which are defined in the diagram.

• Module nodes allow repeated use of common elements, such as Software Engineers rates.
Software Obsolescence Resolution Cost Optimisation Model – Example
# Optimisation – Test Case

<table>
<thead>
<tr>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of Platform</td>
<td>Air</td>
</tr>
<tr>
<td>Level of Integration</td>
<td>Medium</td>
</tr>
<tr>
<td>Testing and Requalification</td>
<td>High</td>
</tr>
<tr>
<td>Level of Modification</td>
<td>Medium</td>
</tr>
<tr>
<td>Software Complexity</td>
<td>Low</td>
</tr>
<tr>
<td>Software Dependency</td>
<td>Low</td>
</tr>
<tr>
<td>Number of Applications</td>
<td>2</td>
</tr>
<tr>
<td>Software Contract Length</td>
<td>6-10 years</td>
</tr>
<tr>
<td>Software Language</td>
<td>2nd Generation</td>
</tr>
<tr>
<td>Software Development Cycle</td>
<td>Stable</td>
</tr>
<tr>
<td>Software Development environment</td>
<td>Stable</td>
</tr>
<tr>
<td>Target Environment</td>
<td>Stable</td>
</tr>
<tr>
<td>Operating environment</td>
<td>stable</td>
</tr>
</tbody>
</table>
Optimisation – Results

- Total Resolution Cost
- Corrective
- Perfective
- Update
- Upgrade
- 1st Line
- 2nd Line

£6,200,000
£6,400,000
£6,600,000
£6,800,000
£7,000,000
£7,200,000
£7,400,000
£7,600,000
£7,800,000
£8,000,000
£8,200,000
£8,400,000
£8,600,000
£8,800,000
£9,000,000
£9,200,000
£9,400,000
£9,600,000
£9,800,000
£10,000,000

£7,600,000
£7,400,000
£7,200,000
£7,000,000
£6,800,000
£6,600,000
£6,400,000
£6,200,000
Summary

Software Obsolescence is an emerging issue and it is important to understand how much Software Obsolescence is going to cost at a very early stage of the development life cycle. In order to do so, we need to:

- Define what Software obsolescence is
- Understand the difference between Software Maintenance and Obsolescence
- Identify how Software Obsolescence is triggered
- Have a framework to manage software obsolescence proactively
- Identify the key Software Obsolescence Resolution approaches
Conclusions

• Software plays an important role in defence. Almost every project in defence has software elements with various degrees of complexity and dependencies.

• In order to understand and see the bigger picture and challenges; software developers and the customers need to foresee the following issues that drive the whole life cost and should be in a position to develop innovative means to mitigate these issues by:

  – Anticipation of the Software Obsolescence at a very early stage of projects.
  – Understanding the technology insertion, technology update requirement.
  – Understanding the relationship between Software Maintenance and Software Obsolescence.
  – Anticipation of future capability integration to the existing platforms taking into account systems of systems, software to software and software to hardware integrations.
  – Formulation and evaluation of alternative architectural framework to inform the software designers that recognises the key market and cost drivers.
Thank you – Any Questions?
Army Software Sustainment Cost Estimating Results
DASA-CE

IT CAST
August 2019
SWS Initiative Objective and Strategy

Accurately estimate Army system Software Sustainment (SWS) 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

Aug 2019

Unclassified
Distribution Statement A: Approved for Public Release Distribution is Unlimited
Executive Summary - Accomplishments

• Established Software Sustainment Data Collection Mechanisms
  – Army Software Data Collection Questionnaire
  – SRDR for Maintenance
  – Software Sustainment WBS Used to Collect Sustainment Costs

• Created Comprehensive Software Sustainment Data Repository
  – 192 Systems
  – 700+ Capability Releases
  – 300+ IAVA Releases
  – 3,200+ records on software license data

• Established Robust Foundation for Software Sustainment Fact-Based Decisions
  – Allocations of Costs by WBS Elements
  – Cost & Schedule Estimating Relationships
  – Benchmarks

• Data and Analysis Results provided to DoD and Army Community
  – Benchmarks and CERs Ready for Use
Decision Information

- Decision information must objectively tie investment costs to software product mission capability

- Program-level management must decide
  - Which baseline change requirements to implement
  - Prioritization of capability, maintenance, and security changes
  - Delivery strategy for incremental software releases

- Enterprise-level management must decide
  - Prioritization of resources across the operational system portfolio
  - Tradeoffs between funding and associated mission capability
DASA-CE SWS WBS

**Software Sustainment**

**1.0 Software Change Product**
- 1.1 Capability Changes
  - Change requirements
  - Change development
  - B/L Integration & Test
  - IV&V
- 1.2 IAVA Changes

**2.0 Project Management**
- Planning
- Execution management
- Configuration management
- Resource & team management
- Contracting management
- Measurement - reporting

**3.0 Software Licenses**
- License - Right to Use
  - COTS/GOTS
  - NDI
  - Open Source
  - Other
- License - Maintenance

**4.0 Certification & Accreditation**
- Security
- Safety
- Networthiness
- Airworthiness

**5.0 System Facilities**
- Hardware
  - Software development
    - assets/workstations
    - System integration & test facilities
    - Test equipment - tools
- Facility Operations

**6.0 Sustaining Engineering**
- 6.1 Help desk
- 6.2 Hosting
- 6.3 Engineering and User Support
  - Test Support
  - Software Delivery
  - Technical Studies
  - Training

**7.0 Field Software Eng.**
- On-Site technical assistance
- Problem Troubleshooting
- S/W Installation
- Operational Assistance
- On-site Training

**8.0 Other**
- Operations
- Organization management
- Personnel management
- Financial management
- Information management
- Process management
- Change management

Version 5.0

Aug 2019

Unclassified

Distribution Statement A: Approved for Public Release Distribution is Unlimited
Army Software Sustainment Definition

- Software sustainment (SWS) includes all software change activities and products associated with modifying a software system after a software release has been provided to an external party.
- The release is the primary SWS change product - a composite of one or more changes - it can be either a formal release or an engineering release.
- SWS includes software enhancements, software maintenance, and cybersecurity updates.
- Software maintenance includes defect repair, rehosting, adaptations, updates, and reconfiguration.
- SWS may be funded by multiple funding sources.
- Costs include both Fixed and Variable costs accrued at both the system and organizational levels.
- Costs include both organic (government) and contractor resources.
Software Sustainment Data Characterization
Data Demographics

Overview
- **192** Systems
- **1,040** Releases
- **3,434** License Records
- **411K** Data Fields

Systems by Super Domain
- Real Time
  - 93
- Engineering
  - 47
- Automated Information Systems
  - 33
- Support
  - 13
- DBS
  - 6

Distribution of System Age

Releases by Change Type
- All Releases: **1040**
  - Maintenance: 318
  - Capability: 718
  - Hybrid: 195
  - Enhancement: 170
  - Other: 16

  - IAVAs: 322
  - Cyber: 341

Releases by Size Measure
- Software Changes: **571**
  - IAVAs: 420
  - Requirements: 224
  - SLOC: 152
  - Other: 39
  - Agile Story Pts: 11

Each release can have multiple size measures.
Annual Cost Distribution
Average Annual Cost by WBS & Cost Allocation by WBS

WBS 1.0 Software Changes $369.8M
WBS 2.0 Project Mgmt $136.5M
WBS 3.0 Licenses $150.2M
WBS 4.0 C&A $60.0M
WBS 5.0 System Facilities $78.3M
WBS 6.0 Sustaining Engineering $248.7M
WBS 7.0 FSEs $291.4M
WBS 8.0 Ops & Mgmt $74.0M

$1.41B Average Annual Cost (BY18)

Represents $56 Million Dollars

Annual Cost Allocation by WBS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RT</td>
<td>33%</td>
<td>11%</td>
<td>4%</td>
<td>4%</td>
<td>9%</td>
<td>12%</td>
<td>25%</td>
<td>2%</td>
</tr>
<tr>
<td>ENG</td>
<td>22%</td>
<td>9%</td>
<td>21%</td>
<td>7%</td>
<td>4%</td>
<td>9%</td>
<td>25%</td>
<td>3%</td>
</tr>
<tr>
<td>AIS</td>
<td>17%</td>
<td>9%</td>
<td>15%</td>
<td>4%</td>
<td>4%</td>
<td>13%</td>
<td>31%</td>
<td>8%</td>
</tr>
<tr>
<td>SUP</td>
<td>24%</td>
<td>15%</td>
<td>17%</td>
<td>8%</td>
<td>4%</td>
<td>21%</td>
<td>5%</td>
<td>7%</td>
</tr>
<tr>
<td>DBS</td>
<td>32%</td>
<td>9%</td>
<td>11%</td>
<td>2%</td>
<td>5%</td>
<td>32%</td>
<td>2%</td>
<td>8%</td>
</tr>
</tbody>
</table>
## Total Annual Cost Distribution

**By Super Domain (BY18$)**

Distributions were developed for each WBS element.

<table>
<thead>
<tr>
<th>Super Domain</th>
<th>RT</th>
<th>ENG</th>
<th>AIS</th>
<th>SUP</th>
<th>DBS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Count</strong></td>
<td>240</td>
<td>136</td>
<td>91</td>
<td>40</td>
<td>17</td>
</tr>
<tr>
<td><strong>Mean</strong></td>
<td>$6,351,776</td>
<td>$6,202,735</td>
<td>$8,572,860</td>
<td>$5,411,627</td>
<td>$59,221,413</td>
</tr>
<tr>
<td><strong>Median</strong></td>
<td>$2,237,870</td>
<td>$2,547,140</td>
<td>$5,972,457</td>
<td>$6,285,354</td>
<td>$58,334,452</td>
</tr>
</tbody>
</table>

* Up to 3 FYs per system
Specific Analysis Overview
(full detail is available)

- **Benchmarks for capability releases by super domain and commodity**
  - # of software changes/release
  - Hours per software change
  - By WBS
  - PDSS vs. PPSS for application super-domains
  - By sustaining organization
  - DSLOC per FTE

- **Cost estimating relationships (CERs) for capability releases**
  - Evaluated meta data for impacts on CERs: commodity, change type, # of inter-service partners, and ACAT levels had an impact
  - Data Trimmed and CERs developed
    - CERs for Software Changes (most effective), Requirements, and Lines of Code

- **Schedule estimating relationships (SERs) for capability releases**
  - Initial SERs had low correlation
  - Data segmented into schedule approaches:
    - Cyclic, Sequential, and Concurrent

- **IAVA Release Analysis**
  - Most data is Level-of-Effort
  - Data best described by median by grouping for # of IAVAs and hours per IAVA
Benchmarks for Capability Releases
Number of Software Changes per Release
By Super Domain

<table>
<thead>
<tr>
<th>Super Domain</th>
<th>Count</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT</td>
<td>89</td>
<td>44</td>
<td>13</td>
</tr>
<tr>
<td>ENG</td>
<td>171</td>
<td>50</td>
<td>33</td>
</tr>
<tr>
<td>AIS</td>
<td>79</td>
<td>29</td>
<td>11</td>
</tr>
<tr>
<td>SUP</td>
<td>14</td>
<td>91</td>
<td>49</td>
</tr>
<tr>
<td>DBS</td>
<td>17</td>
<td>118</td>
<td>83</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
Hours per Software Change
By Super Domain

<table>
<thead>
<tr>
<th>Super Domain</th>
<th>Count</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT</td>
<td>101</td>
<td>1481.6</td>
<td>407.4</td>
</tr>
<tr>
<td>ENG</td>
<td>186</td>
<td>629.8</td>
<td>102.5</td>
</tr>
<tr>
<td>AIS</td>
<td>79</td>
<td>244.9</td>
<td>111.5</td>
</tr>
<tr>
<td>SUP</td>
<td>14</td>
<td>490.7</td>
<td>197.0</td>
</tr>
</tbody>
</table>
Hours per Software Change by Commodity

<table>
<thead>
<tr>
<th>Commodity</th>
<th>Count</th>
<th>Mean</th>
<th>St Dev</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>1</td>
<td>16.1</td>
<td>--</td>
<td>16.1</td>
</tr>
<tr>
<td>ChemBio</td>
<td>4</td>
<td>57.5</td>
<td>58.8</td>
<td>41.7</td>
</tr>
<tr>
<td>Aviation</td>
<td>87</td>
<td>227.2</td>
<td>360.3</td>
<td>58.8</td>
</tr>
<tr>
<td>Simulation</td>
<td>20</td>
<td>103.3</td>
<td>84.1</td>
<td>78.4</td>
</tr>
<tr>
<td>Business</td>
<td>32</td>
<td>154.3</td>
<td>118.4</td>
<td>115.6</td>
</tr>
<tr>
<td>CSISR</td>
<td>52</td>
<td>270.0</td>
<td>312.7</td>
<td>138.2</td>
</tr>
<tr>
<td>Intel</td>
<td>37</td>
<td>351.3</td>
<td>432.7</td>
<td>197.8</td>
</tr>
<tr>
<td>Missiles</td>
<td>17</td>
<td>475.0</td>
<td>456.5</td>
<td>198.8</td>
</tr>
<tr>
<td>Network</td>
<td>14</td>
<td>418.4</td>
<td>459.0</td>
<td>203.3</td>
</tr>
<tr>
<td>Fires</td>
<td>25</td>
<td>307.3</td>
<td>291.0</td>
<td>206.1</td>
</tr>
<tr>
<td>Vehicles</td>
<td>14</td>
<td>201.6</td>
<td>160.4</td>
<td>211.7</td>
</tr>
<tr>
<td>Comms</td>
<td>3</td>
<td>327.3</td>
<td>42.9</td>
<td>345.1</td>
</tr>
<tr>
<td>DBS</td>
<td>10</td>
<td>467.0</td>
<td>377.3</td>
<td>420.4</td>
</tr>
<tr>
<td>SATCOM</td>
<td>8</td>
<td>794.6</td>
<td>471.8</td>
<td>764.4</td>
</tr>
<tr>
<td>MSN CMD</td>
<td>5</td>
<td>922.5</td>
<td>406.5</td>
<td>1005.7</td>
</tr>
</tbody>
</table>

Aug 2019

Distribution Statement A: Approved for Public Release Distribution is Unlimited
**DSLOC per FTE**

*By Super Domain*

<table>
<thead>
<tr>
<th>Super Domain</th>
<th>AIS</th>
<th>ENG</th>
<th>RT</th>
<th>SUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>4</td>
<td>8</td>
<td>23</td>
<td>2</td>
</tr>
<tr>
<td>Mean</td>
<td>69,492</td>
<td>73,166</td>
<td>38,306</td>
<td>139,953</td>
</tr>
<tr>
<td>Min</td>
<td>34,307</td>
<td>16,424</td>
<td>3,496</td>
<td>63,754</td>
</tr>
<tr>
<td>1Q</td>
<td>35,486</td>
<td>20,138</td>
<td>9,892</td>
<td>101,853</td>
</tr>
<tr>
<td>Median</td>
<td>48,094</td>
<td>56,181</td>
<td>21,221</td>
<td>139,953</td>
</tr>
<tr>
<td>3Q</td>
<td>82,099</td>
<td>126,185</td>
<td>44,916</td>
<td>178,052</td>
</tr>
<tr>
<td>Max</td>
<td>147,473</td>
<td>164,340</td>
<td>240,813</td>
<td>216,151</td>
</tr>
</tbody>
</table>

- DSLOC represents Delivered Source Lines of Code which counts all code equally
- The earliest baseline size reported was used to represent DSLOC
- Full Time Equivalent (FTE) counts were derived by including the following WBS Elements: SW Change Product (1.0), Program Management (2.0), Certification and Accreditation (4.0), and Sustaining Engineering (5.0)
- FTEs were derived by using labor hours per man-year and labor rate reported for each program

For a 100,000 DSLOC baseline, the estimated FTEs are RT=4.7, ENG=1.8, AIS=2.1 & SUP=NA
Cost and Schedule Estimating Relationships (CER/SER) Capability Releases
All Data CER

Equation

\[
\log(\text{Hours}) = 0.703 \log(\text{SC}) + 6.2438
\]

R-Squared 36.0%

Adj. R-Squared 35.8%

Observations 397

<table>
<thead>
<tr>
<th>Variable</th>
<th>Coef.</th>
<th>Std Err</th>
<th>T-Value</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>6.2438</td>
<td>0.172</td>
<td>36.231</td>
<td>0.000</td>
</tr>
<tr>
<td>Log(SC)</td>
<td>0.0730</td>
<td>0.047</td>
<td>14.900</td>
<td>0.000</td>
</tr>
</tbody>
</table>

High amount of variance in the Data
Assumptions

• Removed records with:
  − Defense Business Systems (DBS) super domain
  − Hour data outliers or missing data
  − Records with no dependent variable, e.g., SW Change (SC) counts
  − Upper & lower 10% of records based on unit cost

• Both Dependent and Independent variables were transformed using $\log_{10}$
  − Zeros were represented with 0.1

• All categorical variables were represented as dummy variables (0,1)

• Adjusted $R^2$ was used for model performance comparisons
Trimmed Data CER*

Equation

\[ \log(\text{Hours}) = 0.7981 \log(\text{SC}) + 5.905 \]

R-Squared 57.2%
Adj. R-Squared 57.1%
Observations 317

<table>
<thead>
<tr>
<th>Variable</th>
<th>Coef.</th>
<th>Std Err</th>
<th>T-Value</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>5.9052</td>
<td>0.145</td>
<td>40.618</td>
<td>0.000</td>
</tr>
<tr>
<td>Log(SC)</td>
<td>0.7981</td>
<td>0.039</td>
<td>20.532</td>
<td>0.000</td>
</tr>
</tbody>
</table>

* Data records trimmed by 10%
## Total Hours vs SW Changes -1

<table>
<thead>
<tr>
<th>Model</th>
<th>Conditions</th>
<th>Obs</th>
<th>Adj $R^2$</th>
<th>SEE (Hrs)</th>
<th>PRED(30)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{THrs} = 463 \times (\text{TSC})^{0.69}$</td>
<td>All data</td>
<td>329</td>
<td>0.36</td>
<td>48,385</td>
<td>17.3%</td>
</tr>
<tr>
<td>$\text{THrs} = 341 \times (\text{TSC})^{0.79}$</td>
<td>10% trimmed data</td>
<td>263</td>
<td>0.57</td>
<td>44,842</td>
<td>23.6%</td>
</tr>
<tr>
<td>$\text{THr}_\text{AIS} = 242 \times (\text{TSC})^{0.73}$</td>
<td>10% trimmed &amp; Super Domains (Categorical)</td>
<td>263</td>
<td>0.62</td>
<td>39,330</td>
<td>20.2%</td>
</tr>
<tr>
<td>$\text{THr}_\text{ENG} = 386 \times (\text{TSC})^{0.73}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THr}_\text{RT} = 736 \times (\text{TSC})^{0.73}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THr}_\text{SUP} = 698 \times (\text{TSC})^{0.73}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THrs} = 1,452 \times \text{TSC}^{0.66}$</td>
<td>10% trimmed &amp; Commodities (Categorical)</td>
<td>263</td>
<td>0.68*</td>
<td>40,886</td>
<td>23.2%</td>
</tr>
<tr>
<td>$\text{THrs} = 301 \times \text{TSC}^{0.66}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THrs} = 364 \times \text{TSC}^{0.66}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THrs} = 182 \times \text{TSC}^{0.66}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THrs} = 1,531 \times \text{TSC}^{0.66}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THrs} = 1,114 \times \text{TSC}^{0.66}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THrs} = 577 \times \text{TSC}^{0.66}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THrs} = 1,005 \times \text{TSC}^{0.66}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THrs} = 1,742 \times \text{TSC}^{0.66}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THrs} = 425 \times \text{TSC}^{0.66}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{THrs} = 608 \times (\text{TSC})^{0.98} / (\text{TReqts})^{0.21}$</td>
<td>10% trimmed</td>
<td>32</td>
<td>0.84</td>
<td>32,228</td>
<td>25.0%</td>
</tr>
<tr>
<td>$\text{THrs} = 330 \times (\text{TSC})^{0.97} / (\text{TReqts}_{\text{Imp}})^{0.11}$</td>
<td>10% trimmed</td>
<td>65</td>
<td>0.74</td>
<td>63,904</td>
<td>23.1%</td>
</tr>
</tbody>
</table>

* High $P$-Values for one or more coefficients
## Total Hours vs SW Changes -2

<table>
<thead>
<tr>
<th>Model</th>
<th>Conditions</th>
<th>Obs</th>
<th>Adj R²</th>
<th>SEE (Hrs)</th>
<th>PRED(30)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_{\text{Hrs}} = 296 \times (TSC)^{0.94} / (\text{EI}_\text{Mod})^{0.11} )</td>
<td>10% trimmed</td>
<td>41</td>
<td>0.74*</td>
<td>47,326</td>
<td>22.0%</td>
</tr>
<tr>
<td>( T_{\text{Hrs}} = 1,219 \times (TSC)^{0.75} / (\text{SWBase})^{0.04} )</td>
<td>10% trimmed</td>
<td>69</td>
<td>0.61*</td>
<td>36,567</td>
<td>26.1%</td>
</tr>
<tr>
<td>( T_{\text{Hrs}} = 757 \times (TSC)^{1.02} / (\text{BL})^{0.36} )</td>
<td>10% trimmed</td>
<td>45</td>
<td>0.74</td>
<td>81,719</td>
<td>15.6%</td>
</tr>
<tr>
<td>Cyber Enhance Hybrid Maint Other</td>
<td>10% trimmed &amp; Change Type (Categorical)</td>
<td>263</td>
<td>0.59*</td>
<td>39,573</td>
<td>21.3%</td>
</tr>
<tr>
<td>( T_{\text{Hrs}} = 332 \times TSC^{0.79} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_{\text{Hrs}} = 531 \times TSC^{0.79} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_{\text{Hrs}} = 382 \times TSC^{0.79} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_{\text{Hrs}} = 281 \times TSC^{0.79} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_{\text{Hrs}} = 284 \times TSC^{0.79} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_{\text{Hrs}} = 338 \times TSC^{0.77} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>* Enh%0.10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>* Maint%0.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>* Cyber%0.03</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>* Other%0.01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_{\text{Hrs}} = 281 \times TSC^{0.79} )</td>
<td>10% trimmed &amp; percentages of Change Types</td>
<td>263</td>
<td>0.60*</td>
<td>26,494</td>
<td>6.8%</td>
</tr>
</tbody>
</table>
IAVA Release Analysis
There is a central tendency across segmentation groups using the Median

IAVA data is better estimated using descriptive statistics i.e. average cost (hours per IAVA) as compared to regression
IAVAs per Release

### IAVA Releases

<table>
<thead>
<tr>
<th>Super Domain</th>
<th>Count</th>
<th>Mean</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT</td>
<td>63</td>
<td>55</td>
<td>37</td>
</tr>
<tr>
<td>ENG</td>
<td>116</td>
<td>38</td>
<td>24</td>
</tr>
<tr>
<td>AIS</td>
<td>43</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>SUP</td>
<td>7</td>
<td>21</td>
<td>30</td>
</tr>
</tbody>
</table>

Aug 2019

Distribution Statement A: Approved for Public Release Distribution is Unlimited
Cost per IAVA can be used to bound the number of IAVAs a program can expect to do given a fixed budget.
Lessons Learned
Hierarchy of Use Cases

1. Utilize this analysis structure and findings to support your data collection and analysis
   - Build predictive models based on historical actual data collected to predict future efforts

2. Utilize data SWM PH 2 data repository to filter for analogous systems
   - Develop custom regression models based on systems within the analogous set

3. Use the regression models developed from the SWM PH 2 effort to estimate future effort
   - Utilize the benchmarks presented to estimate or compare against your program
Why Software Sustainment Cost is Difficult to Track

1. System Segmentation

- 9 Sub-Systems to Sustain
  - Common SW
  - CDSS
  - DE-SS
  - IFS
  - Fixed
  - F-MFWS
  - P-MFWS
  - GWS
  - IPC-1,2,3

- System segmentation makes it difficult to track software size, licenses, requirements, across the lifecycle
- Each sub-system is separately managed

2. Contracting (Omnibus Contracts)

- Current Omnibus contracts do not provide a means to track cost to an individual system or WBS activity
- Contract CLINs can contain multiple systems and multiple funding streams
- Current contracts do not align to software sustainment activities

3. Activity Segmentation

- Release activities get split across multiple performing organizations
- Annual funding makes it difficult to track release costs since releases may take from one month to several years

- 1 Program of Record (PoR)
- System 1
- System 2
- System 3
- PMO System 3
- CLIN 0001
- CLIN 0002

- Activities:
  - Bug Fixes
  - Enhancements
  - IAVAs
  - Integration / Test

- Performing Organizations:
  - SEC Contract
  - PMO Contract
  - Army Depot
  - SEC / Government

Aug 2019

Unclassified

Distribution Statement A: Approved for Public Release Distribution is Unlimited
Lessons Learned From Analysis

• It often takes multiple iterations with the data provider to clean up the data provided – this may be caused by a misunderstanding of what data is being requested or a lack of complete data
  – It is worth the effort to clean up the submitted data

• Data for some of the WBS elements was reported “unavailable” because the work was funded by different organizations, because costs were applicable to multiple systems, or because data was not tracked at lower WBS levels

• Release data was collected for a full release – yet it is tracked annually
  – Future analysis will evaluate annual release data and aggregate release data that spans multiple fiscal years

• Inner program CERs and SERs show significantly better statistics
  – Project leads at LCMC’s can use same methodology to develop estimates for program funding
Next Steps & Future Research

• Annual data collection
  – The Software Resources Data Reporting for Maintenance (SRDR-M*) closely aligns to the DASA-CE SWS WBS and data requirements
    o Moving forward, the SRDR-M will be utilized to collect SWS data from Army programs and perform analysis

• Additional analysis of data, including:
  – Cost impact of cybersecurity framework (DIACAP vs RMF)
  – Cost of Cybersecurity
  – Analysis of annualized release data

• Expand SER analysis to include all systems in each release duration category (Cyclic, Sequential, Concurrent)

• Additional license analysis
  – Does higher license costs correlate to higher sustainment costs?
  – Does using COTS software save money in sustainment?

• Impact of budget reductions on fixed-cost versus variable-cost funding

• Iterative/Agile versus traditional development is being explored for differences

• New FY18 PPSS data being collected
Concluding Remarks

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 sustainment
  – This facilitates major milestone estimates, O&S cost targets, Operation Sustainment Reviews, and yearly POM reviews
  – Dataset is hosted on CADE under “Library”

*See http://cade.osd.mil/policy/dids for more information
Contributors

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

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

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

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

Dr. Robert Charette
President
ITABHI Corporation
(540) 972-8150

Paul Janusz
Software Measurement Analysis
US Army AFC
973-724-2644
# Acronyms -1

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACAT</td>
<td>Acquisition Category</td>
</tr>
<tr>
<td>AIS</td>
<td>Automated Information System super domain</td>
</tr>
<tr>
<td>BL</td>
<td>Software Change Backlog</td>
</tr>
<tr>
<td>BY</td>
<td>Base Year</td>
</tr>
<tr>
<td>C&amp;A</td>
<td>Certification and Accreditation</td>
</tr>
<tr>
<td>C5ISR</td>
<td>Command, Control, Communications, Computers, Cyber, Intelligence,</td>
</tr>
<tr>
<td></td>
<td>Surveillance, and Reconnaissance</td>
</tr>
<tr>
<td>CADE</td>
<td>Cost Assessment Data Enterprise</td>
</tr>
<tr>
<td>CER</td>
<td>Cost Estimating Relationship</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off The Shelf</td>
</tr>
<tr>
<td>CRED</td>
<td>Uncertainty Estimation Determination</td>
</tr>
<tr>
<td>CSCI</td>
<td>Computer Software Configuration Item</td>
</tr>
<tr>
<td>Cyber%</td>
<td>Percent of the release that is Cybersecurity updates</td>
</tr>
<tr>
<td>DASA-CE</td>
<td>Deputy Assistant to the Secretary of the Army for Cost and Economics</td>
</tr>
<tr>
<td>DBS</td>
<td>Defense Business System commodity</td>
</tr>
<tr>
<td>DIACAP</td>
<td>DoD Information Assurance Certification and Accreditation Process</td>
</tr>
<tr>
<td>DISA</td>
<td>Defense Information Systems Agency</td>
</tr>
<tr>
<td>DoD</td>
<td>US Department of Defense</td>
</tr>
<tr>
<td>DSLOC</td>
<td>Delivered Source Lines of Code</td>
</tr>
<tr>
<td>ECP</td>
<td>Engineering Change Proposal</td>
</tr>
<tr>
<td>El_Mod</td>
<td>External Interfaces Modified</td>
</tr>
<tr>
<td>ENG</td>
<td>Engineering super domain</td>
</tr>
<tr>
<td>Enh%</td>
<td>Percent of the release that is Enhancements to the system</td>
</tr>
<tr>
<td>EW</td>
<td>Electronic Warfare</td>
</tr>
</tbody>
</table>

**Aug 2019**

*Distribution Statement A: Approved for Public Release Distribution is Unlimited*
## Acronyms -2

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSE</td>
<td>Field Software Engineering</td>
</tr>
<tr>
<td>FTE</td>
<td>Full Time Equivalent</td>
</tr>
<tr>
<td>IAVA</td>
<td>Information Assurance Vulnerability Alert</td>
</tr>
<tr>
<td>IAVM</td>
<td>Information Assurance Vulnerability Management</td>
</tr>
<tr>
<td>ICEAA</td>
<td>International Cost Estimating and Analysis Association</td>
</tr>
<tr>
<td>Maint%</td>
<td>Percent of the release that is Maintenance changes</td>
</tr>
<tr>
<td>NVD</td>
<td>National Vulnerability Database</td>
</tr>
<tr>
<td>O&amp;S</td>
<td>Operations and Sustainment</td>
</tr>
<tr>
<td>ODC</td>
<td>Other than Direct Costs</td>
</tr>
<tr>
<td>OMA</td>
<td>Operations and Maintenance Army funding</td>
</tr>
<tr>
<td>OPA</td>
<td>Other Program Army funding</td>
</tr>
<tr>
<td>OSMIS</td>
<td>Operation/Sustainment Management Information System</td>
</tr>
<tr>
<td>PDSS</td>
<td>Post-Deployment Software Support</td>
</tr>
<tr>
<td>PEO</td>
<td>Program Executive Office</td>
</tr>
<tr>
<td>POM</td>
<td>Program Objective Memorandum</td>
</tr>
<tr>
<td>PPSS</td>
<td>Post-Production Software Support</td>
</tr>
<tr>
<td>PTR</td>
<td>Problem Trouble Report</td>
</tr>
</tbody>
</table>
## Acronyms - 3

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDT&amp;E</td>
<td>Research, Development, Testing, and Evaluation</td>
</tr>
<tr>
<td>RMF</td>
<td>Risk Management Framework</td>
</tr>
<tr>
<td>RT</td>
<td>Real-Time super domain</td>
</tr>
<tr>
<td>SC</td>
<td>Software Changes</td>
</tr>
<tr>
<td>SEC</td>
<td>Software Engineering Center</td>
</tr>
<tr>
<td>SER</td>
<td>Schedule Estimating Relationship</td>
</tr>
<tr>
<td>SLOC</td>
<td>Source Lines of Code</td>
</tr>
<tr>
<td>SRDR</td>
<td>Software Resources Data Report</td>
</tr>
<tr>
<td>SRDR-M</td>
<td>Software Resources Data Report for Maintenance</td>
</tr>
<tr>
<td>STIG</td>
<td>Security Technical Implementation Guides</td>
</tr>
<tr>
<td>SUP</td>
<td>Mission Support super domain</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>SWBase</td>
<td>Software Baseline SLOC</td>
</tr>
<tr>
<td>SWS</td>
<td>Software Sustainment</td>
</tr>
<tr>
<td>TDEV</td>
<td>Time to Develop</td>
</tr>
<tr>
<td>THrs</td>
<td>Total release hours</td>
</tr>
<tr>
<td>TReqts</td>
<td>Total Requirements in a system</td>
</tr>
<tr>
<td>TReqts_Imp</td>
<td>Total Requirements Implemented in a release</td>
</tr>
<tr>
<td>TSC</td>
<td>Total Software Changes for a release</td>
</tr>
<tr>
<td>WBS</td>
<td>Work Breakdown Structure</td>
</tr>
</tbody>
</table>
Backup
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:
  - Enhancements
    - New capability: ECPs, new requirements
    - Redesign / rewrite: 100% new code, new architecture
  - Maintenance
    - Defect repair: bug fixes, PTR fixes
    - Reconfiguration: threat loads, EW parameters
    - Rehost: migration from Windows to Linux
    - Testing: interoperability testing
    - Update: weapon tables, switch configurations, Operating System
    - Update, Defect repair (see above)
    - Upgrade: upgrade the v “n” to v “n+1”, upgrading applications
  - Cyber
    - Vulnerabilities: enhance security posture not resolved

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
Benchmarks for Capability Releases
Software Changes per Release
By Sustaining Organization

Aug 2019

Distribution Statement A: Approved for Public Release Distribution is Unlimited
Software Changes per Release
By Sustaining Organization

Count
Org-4  1  2  2  21  22  131  9  25  61  12  18  32  21  4  9  3  23  3
Mean
St Dev
N/A  1.039  1.399  1.281  1.312  1.823  1.037  1.88  1.402  1.327  1.675  1.995  1.646  1.802  1.438  1.474  0.349
Median

Aug 2019
Distribution Statement A: Approved for Public Release Distribution is Unlimited
Hours per Software Change
By Sustaining Organization

Aug 2019

Distribution Statement A: Approved for Public Release Distribution is Unlimited
IT CAST 2019: Agile Projects and GAO Best Practices

Jennifer Leotta, CCEA
Outline

- Overview: GAO Best Practice Guides
  - Characteristics of reliability in current guides
- GAO Agile Assessment Guide
  - Guide’s Contents
  - Chapter 9: Program Controls
    - Work Breakdown Structure (WBS)
    - Cost Estimating
    - Scheduling
    - Earned Value Management
OVERVIEW:
GAO BEST PRACTICE GUIDES
To develop these guides, GAO compiles a list of experts in that area to discuss topics.

For example, the Cost Expert Group was established in 2005 and has since grown to include experts on schedule analysis and earned value management. Group meets twice a year to discuss a variety of related issues. Contributions have been invaluable both in providing historical information and experience and keeping the Guide current with industry trends.

GAO has currently assembled an Agile Expert Group. The group’s first meeting occurred on August 30, 2016.
GAO Guides and Best Practices

Purpose of these documents is two-fold:

• Provide criteria for GAO to use when performing audits
• Provide guidance for agencies
• **Cost Estimating and Assessment Guide** (March 2009, GAO-09-3SP)
• **Schedule Assessment Guide** (December 2015, GAO-16-89G)
• **Technology Readiness Assessment (TRA) Guide** (August 2016, GAO-16-410G)
Characteristics of Reliability

- The guides lay out characteristics of a reliable cost estimate, schedule, and EVM system.
- The following slides provide an overview of what GAO has published in the Cost Guide and Schedule Guide.
- The Agile Guide builds on the program control best practices established in these guides.

To download these guides:
Four Characteristics of a Reliable Estimate

**Comprehensive**
- Develop the estimating plan
- Determine the estimating approach

**Well-Documented**
- Define the program
- Identify ground rules and assumptions
- Obtain data
- Document the estimate
- Present estimate to management

**Accurate**
- Develop the point estimate
- Compare the point estimate to an independent estimate
- Update the estimate with actual costs

**Credible**
- Create an independent cost estimate
- Conduct sensitivity analysis
- Conduct risk and uncertainty analysis

- Are all costs included?
- Can the estimate be recreated?
- Is the estimate unbiased?
- What is the uncertainty?
Four Characteristics of a Reliable Schedule

**Comprehensive**
- Capture all activities
- Assign resources to all activities
- Establish durations for all activities

**Well Constructed**
- Sequence all activities
- Confirm the critical path
- Confirm reasonable float (slack)

**Credible**
- Confirm vertical and horizontal traceability
- Conduct a schedule risk analysis

**Controlled**
- Update the schedule with progress
- Maintain a schedule baseline

Is all effort included?
Is the network logical?
What is the uncertainty?
Is progress measured?
Three Characteristics of a Reliable EVM System

- **Comprehensive**
  - Certified EVM system
  - IBR conducted
  - Reliable schedule
  - EVM surveillance

- **Accurate**
  - No data anomalies
  - Consistent data
  - Realistic EAC

- **Informative**
  - Regular reviews conducted
  - Corrective action plans
  - Updated PMB
GAO Agile Assessment Guide

- Chapter 1 – Background
- Chapter 2 – Compliance and Past Work
- Chapter 3 – Agile Adoption Best Practices
  - Team activities, Program processes, and Organizational Environment
- Chapter 4 – Agile Implementation Challenges
- Chapter 5 - Agile Metrics
- Chapter 6 – Requirements Decomposition
- Chapter 7 – Agile and the Federal Acquisition Process
  - Agile and Federal Contracting Process / Budget Process
- Chapter 8 - Agile and Program Management Factors
  - Program Planning and Tradeoffs, Team composition
- Chapter 9 – Agile Program Control Best Practices
  - Cost estimating, Scheduling, and Earned Value Management

Appendices:
- Agile Glossary
- Effects of not following best practices
- Agile Methodologies
- Debunking Agile Myths
- Questions for Auditors and Managers
- Case Study Descriptions

GAO is currently working to develop the Agile Assessment Guide exposure draft. Expected out on the GAO webpage: Fall 2019
The chapter discusses developing a work breakdown structure (WBS) used by management and Agile teams to provide a clear picture of the total scope of work necessary to meet a program’s requirements.

Currently the guide shows the following break-out for the WBS:

- **Epic**: A large duration effort that captures high-level capabilities. An epic usually spans many releases.

- **Feature**: A specific amount of work that can be developed within one or two reporting periods. It can be further segmented into stories from the user’s point of view. The functionality is described with enough detail that it can remain stable throughout its development and integration into working software.

- **Story**: Smallest level of detail in an Agile program; is subject to change based on user feedback. For this reason, a story can be added to or deleted without altering the overall scope of the features. A story is weighted for complexity using story points.

Recommend working at the feature level.
## Cost Best Practices in an Agile Environment

### GAO Cost Guide: Comprehensive

- Cost estimates should be structured in sufficient detail to ensure that cost elements are neither omitted nor double-counted.
- The cost estimate should be based on a product-oriented work breakdown structure (WBS) that allows a program to track cost and schedule by defined deliverables, such as hardware or software components.
- Where information is limited and judgments must be made, the cost estimate should document all cost-influencing ground rules and assumptions.

### Agile Example

- Roadmap and prioritized backlog indicate “must have” features to be developed with input from stakeholders and subject matter experts.
- Roadmap or Vision aligned and traceable to program requirements.
- Backlog queues, unfinished work, and any defects are listed in priority order.
Agile and Program Controls

Chapter 9: Cost Estimating

• Cost estimating best practices established in the GAO Cost Guide are still applicable to Agile projects but these projects should use Agile documentation and artifacts produced during Agile development to develop and update cost estimates.

• The guide highlights cost estimating concerns:
  • Consistent sizing
  • Expertise of the development team
  • Cost estimating benefits
Agile and Program Controls
Chapter 9: Cost Estimating

- Consistent sizing
  - Developers often use relative sizing techniques to estimate effort; however, it is important for estimators to both understand terms used by the developers and use consistent sizing/metrics to develop cost estimates

- Expertise of the development team
  - Due to the use of relative sizing, it is important for estimators to understand the composition and expertise of the development team. Estimators may take a more iterative, integrated, and collaborative approach to estimate Agile projects
Agile and Program Controls
Chapter 9: Cost Estimating

- Cost estimating benefits
  - As with traditional projects, estimates can be used to support the government budgeting process and justify management decisions. As more time passes, estimators can update the estimate with new data discovered during development to decrease the uncertainty surrounding cost.
Scheduling Best Practices in an Agile Environment

**Step 4: Establish Durations**

- The schedule should realistically reflect how long each activity will take. When the duration of each activity is determined, the same rationale, historical data, and assumptions used for cost estimating should be used. Durations should be reasonably short and meaningful and should allow for discrete progress measurement. Schedules that contain planning and summary planning packages as activities will normally reflect longer durations until broken into work packages or specific activities.

**Agile Example**

- Durations are time-boxed in Agile, which makes each release a consistent duration in the schedule. However, since requirements can fluctuate, it is important to track what work has been accomplished for each release in the schedule.
- Can use the prioritized backlog, release plans, and roadmap.
- Additionally, Agile tools can help track durations.
Agile and Program Controls
Chapter 9: Schedule

- Scheduling best practices established in the GAO Schedule Guide are still applicable to Agile projects but these projects should use Agile documentation and artifacts produced during Agile development to develop and update cost estimates.

- The guide will highlight scheduling concerns:
  - Planning for all activities
  - Minimize the use of constraints
  - Assign resources
  - Conduct a schedule risk analysis (SRA)
  - Develop and use a schedule baseline
Agile and Program Controls
Chapter 9: Schedule

• Planning for all activities
  • While near term work is subject to change, Agile projects define their goals in a product vision and typically plan releases to meet this vision

• Minimize the use of constraints
  • While it may be tempting to put time-boxed iterations with date constraints in the schedule, this will reduce the utility of the schedule as a coordination tool among Agile teams

• Assign resources
  • Often done in Agile tools as part of iteration planning, understanding resource availability has a major impact on estimates of work and duration
• Conduct a schedule risk analysis (SRA)
  • While Agile emphasizes that teams will uncover risk via early and frequent delivery of software, the potential impact of some issues (e.g. technical debt or team size) should be considered earlier rather than later

• Develop and use a schedule baseline
  • Welcoming change does not mean that software is developed and delivered in an undisciplined manner. Critical features identify the projects’ schedule baseline and, as a result, product owners have the ability to reprioritize work at the end of each iteration
EVM in an Agile Environment

**GAO Cost Guide**

- Step 4: Estimate the labor and material required to perform the work and authorize the budgets, including management reserve

**Agile example**

- Features should be the basis for identifying work package scope and budget
Agile and Program Controls
Chapter 9: EVM

• EVM best practices established in the GAO Cost Guide are still applicable to Agile projects but these projects should use Agile documentation and artifacts produced during Agile development to develop and update cost estimates.

• The guide will highlight EVM concerns:
  • WBS detail
  • Measuring earned value
  • Calculating variances
  • Controlling baseline changes

• The EVM portion of the chapter relies heavily on the March 2018 NDIA EVM Desk Guide.
Agile and Program Controls
Chapter 9: EVM

- WBS detail
  - Recommend tracking at the feature or epic level since Agile is dynamic and tracking at too low a level will not yield valuable data

- Measuring earned value
  - Recommend tracking features using the percent complete method and their associated stories at 0/100 method
Agile and Program Controls
Chapter 9: EVM

• Calculating variances
  • Every project needs a method to measure performance. This can be tricky in Agile since cost and schedule are considered fixed in each iteration. However, features can often cross iterations and variances can occur in Agile projects

• Controlling baseline changes
  • A properly designed change process will not restrict the Agile process while maintaining a credible baseline.
Conclusions

• Best practice guides help establish a consistent methodology that can be used across the federal government
• Cost estimating, scheduling, and earned value management (EVM) best practices established in earlier guides are still applicable to Agile projects
Thank you

Guides Available Online and Downloadable in PDF:

  GAO Cost Estimating and Assessment Guide:
  http://www.gao.gov/products/GAO-09-3SP

  GAO Schedule Assessment Guide:
  http://www.gao.gov/products/GAO-12-120G

  GAO Technology Readiness Assessment Guide:
Agile Work

- Vision
- Roadmap
- Release
- Iteration
- Daily Work
EVM Relationships in Agile Projects

- **Traditional Earned Value Management**
  - Control account
  - Work package
  - Planning package
  - Quantifiable Back-up data

- **Alignment to Agile Project**
  - Epic/capability
  - Feature

- **Alignment to Agile Time-Boxes**
  - Release
  - Iteration
  - Story
  - Prioritized Backlog

Source: Goes here | Product #
SOFTWARE ACQUISITION AND PRACTICES (SWAP) STUDY
Directed by Section 872 of the 2018 National Defense Authorization Act

"To undertake a study on streamlining software development and acquisition regulations..."
The FY18 National Defense Authorization Act (NDAA) directs the Secretary of Defense to task the Defense Innovation Board "to undertake a study on streamlining software development and acquisition regulations." The NDAA further stipulates that the study must:

1. Review **acquisition processes** and organizational structures to improve the efficiency and effectiveness of software adoption **to maintain defense technology advantage**

2. Review a cross section of ongoing software development and acquisition programs to **identify case studies of best and worst practices** within the Department

3. Produce **specific and detailed recommendations** for any **legislative and non-legislative reforms**

4. Produce **additional recommendations** for legislation as such members consider appropriate
DELIVERABLES

1. Proposed **statutory changes** and suggestions for new language to modernize and streamline software acquisition and processes

2. Proposed necessary **regulatory changes** and suggestions for new language to modernize and streamline software acquisition and processes

3. Identification of **organizational and cultural practices** that impede the Department's adoption of software and proposals for improving software acquisition and practices

4. Identification of the **appropriate type of data to collect** for effective decision-making and oversight of software acquisition
DIB SWAP STUDY MEMBERSHIP

Defense Innovation Board Subcommittee Members

Dr. J. Michael McQuade
Co-Chair
Vice President for Research
Carnegie Mellon University

Dr. Richard M. Murray
Co-Chair
Prof of Control and Dynamical Systems and Bioengineering
California Institute of Technology

Mr. Gilman Louie
(Consultant)
Founder and Partner of Alsop-Louie Partners

Mr. Milo Medin
Vice President, Access Services
Google

Ms. Jennifer Pahlka
Founder and Executive Director
Code for America

Mr. Trae’ Stephens
(Consultant)
Partner at Founders Fund
**APPROACH**

**Key Deliverables**
- Interim Report (15 May)
- Final Report Due 5 April to A&S

**DIBPublicMtgs**
- Boston 4/26
- Silicon Valley 7/11
- DC 10/10

**SPG/WGMtgs**
- SPG 5/1
- WG 6/28
- WG 7/26
- WG 8/30
- WG 9/27
- WG 10/31
- WG 11/29
- WG 12/19
- WG 1/24
- SPG 2/17
- WG 3/26

**SiteVisits**
- Conduct Site Visits
  - Visit 1
  - Visit 2
  - Visit 3/4
  - Visit 5
  - Visit 6
  - Visit 7

**Background Research**
- Initial Research
- 2nd Iteration

**Collect Feedback**
- Gather/incorporate academic, industry, government, and congressional feedback on draft products (e.g., surveys)

**Release Iterative Draft Products**
- Ancillary work products (e.g., DIB and WG products, vignettes, concept papers)
  - 10 Commandments
    - 4/20
  - DIB Metrics
    - 7/10
  - Is Your Compute Holding You Back;
  - Is Your Development Environment Holding You Back;
  - Program Observations;
  - Detecting Agile BS
    - 10/10
  - TL; DR
  - README
  - Draft Main Body
  - Draft Supporting
  - Vignettes
    - 3/6

**SPG/WGMtgs:**
- SPG: Senior Principals Group
- WG: Working Group

---

**UNCLASSIFIED**
PATH TO ACTION

3 Themes → describe software capabilities world which must be improved by focusing on...

4 Lines of Effort → describe the main areas that must be change, by acting on...

10 / 16 Recommendations → describe the specific changes that need to be made, via...

Initial Implementation Plans comprised of...

Actions
THREE THEMES

Speed and cycle time are the most important metrics for software

Software is made by people and for people, so digital talent matters

Software is different than hardware (and not all software is the same)
FOUR LINES OF EFFORT

A. Refactor statutes, regulations, and processes for software

B. Create and maintain cross-program/cross-service digital infrastructure

C. Create new paths for digital talent (especially internal talent)

D. Change the practice of how software is procured and developed
DIB TOP TEN RECOMMENDATIONS

THINGS TO DO STARTING...NOW!
## Recommendation A1

### Line of Effort | Refactor statutes, regulations, and processes for software
---|---
**Recommendation** | Establish new acquisition pathway(s) for software that prioritizes continuous integration and delivery of working software in a secure manner, with continuous oversight from automated analytics.

### Draft Implementation Plan

| A1.1 | (optional) Submit legislative proposal using Sec 805 to propose new acquisition pathways for two or more classes of software (e.g., application, embedded), optimized for DevSecOps | USD(A&S), in coordination with USD(C) and CAPE | Q3 FY19 |
| A1.2 | Create new acquisition pathway(s) for two or more classes of software, optimized for DevSecOps (based on A2c.1 or Appendix B.1) | HASC, SASC | FY20 NDAA |
| A1.3 | Develop and issue a Directive Type Memorandum (DTM) for the new software acquisition pathway | USD(A&S) | Q1 FY20 |
| A1.4 | Issue Service level guidance for new acquisition pathway | SAE | Q2 FY20 |
| A1.5 | Select pilot programs using DevSecOps to convert to or utilize new software acquisition pathway | USD(A&S), with SAEs | Q2 FY20 |
| A1.6 | Develop and implement training at Defense Acquisition University on new software acquisition pathway for all acquisition communities (FM, Costing, PM, IT, SE, etc.) | USD(A&S) | Q3 FY20 |
| A1.7 | Convert DTM to DoD Instruction (5000.SW?), incorporating lessons learned during pilot program implementation | USD(A&S) | Q4 FY20 |
## RECOMMENDATION A2

### Line of Effort | Refactor statutes, regulations, and processes for software
---|---
### Recommendation
Create a new appropriations category that allows (relevant types of) software to be funded as a single budget item, with no separation between RDT&E, production, and sustainment.

<table>
<thead>
<tr>
<th>Draft Implementation Plan</th>
<th>Lead Stakeholder</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A2.1</strong> (optional) Submit legislative proposal using Sec 805 to create a new appropriations category for software and software-intensive programs</td>
<td>USD(A&amp;S), with USD(C) and CAPE</td>
<td>Q3 FY19 for FY20 NDAA</td>
</tr>
<tr>
<td><strong>A2.2</strong> Create new appropriation category for software-intensive programs, with appropriate reporting and oversight for software (based on Action A2.1 or Appendix B.1)</td>
<td>HAC-D, SAC-D, with OSD, HASC, SASC</td>
<td>FY20 NDAA, FY20 budget</td>
</tr>
<tr>
<td><strong>A2.3</strong> Select initial programs using DevSecOps to convert to or use new SW Appropriation in FY20</td>
<td>USD(A&amp;S), with Service Acquisition Executives</td>
<td>Q4 FY19</td>
</tr>
<tr>
<td><strong>A2.4</strong> Define budget exhibits for new software appropriation (replacement for P- and R-forms; see Appendix C)</td>
<td>USD(A&amp;S), with USD(C), CAPE, HAC-D, SAC-D</td>
<td>Q4 FY19</td>
</tr>
<tr>
<td><strong>A2.5</strong> Change audit treatment of software with these goals: (1) separate category for software instead of being characterized as property, plant, and equipment; (2) default setting that software is an expense, not an investment; and (3) there is no “sustainment” phase for software</td>
<td>FASAB, with USD(A&amp;S) and USD(C)</td>
<td>End FY20</td>
</tr>
<tr>
<td><strong>A2.6</strong> Make necessary modifications in supporting PPB&amp;E systems to allow use and tracking of new software appropriation</td>
<td>USD(C) and CAPE</td>
<td>Q1 FY21</td>
</tr>
<tr>
<td><strong>A2.7</strong> Ensure programs using new software appropriation submit budget exhibits in the approved format</td>
<td>SAE with USD(C), CAPE</td>
<td>FY 22 POM</td>
</tr>
</tbody>
</table>
# RECOMMENDATION B1

**Line of Effort** | Create and maintain cross-program/cross-service digital infrastructure
---|---
**Recommendation** | Establish and maintain digital infrastructure within each Service or Agency that enables rapid deployment of secure software to the field and incentivizes its use by contractors.

## Draft Implementation Plan

<table>
<thead>
<tr>
<th>Draft Implementation Plan</th>
<th>Lead Stakeholder</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>B1.1</strong> Designate organization(s) responsible for creating and maintaining the digital infrastructure for each Service’s digital infrastructure. Explore the use of tiered approaches with infrastructure at Service or Program level, as appropriate</td>
<td>DoD CIO, USD(C) and Services (SAE and Service CIO)</td>
<td>Q3 FY19</td>
</tr>
<tr>
<td><strong>B1.2</strong> Designate organization responsible for creating and maintaining digital infrastructure for DoD agencies and organizations, including joint digital infrastructure available to the Services</td>
<td>USD(A&amp;S), with CIO, CMO</td>
<td>Q3 FY19</td>
</tr>
<tr>
<td><strong>B1.3</strong> Provide resources for digital infrastructure, including cloud solutions, pre-approved “drop-ship” local compute capability, approved development environments (see DIB Compute Environment concept paper, Appendix I [Glossary])</td>
<td>USD(A&amp;S), SAE with CAPE, USD(C)</td>
<td>FY20 budget</td>
</tr>
<tr>
<td><strong>B1.4</strong> Define baseline digital infrastructure systems and implement procurement and deployment processes and capability</td>
<td>Responsible organizations from B1.1, B1.2</td>
<td>Q2 FY20</td>
</tr>
<tr>
<td><strong>B1.5</strong> Implement digital infrastructure and provide access to ongoing and new programs</td>
<td>Responsible organizations from B1.1, B1.2</td>
<td>Q3 FY20</td>
</tr>
<tr>
<td><strong>B1.6</strong> Identify acquisition programs to transition to digital infrastructure</td>
<td>SAE</td>
<td>Q2 FY20</td>
</tr>
<tr>
<td><strong>B1.7</strong> Transition programs to digital infrastructure</td>
<td>SAE, PEO, PM</td>
<td>Q4 FY20</td>
</tr>
</tbody>
</table>
## RECOMMENDATION B2

<table>
<thead>
<tr>
<th>Line of Effort</th>
<th>Create and maintain cross-program/cross-service digital infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommendation</td>
<td>Create, implement, support, and use fully automatable approaches to testing and evaluation (T&amp;E), including security, that allow high confidence distribution of software to the field on an iterative basis.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Draft Implementation Plan</th>
<th>Lead Stakeholders</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>B2.1</strong> Establish procedures for fully automated testing on digital infrastructure (Rec B1), updating DoDI 5129.47 and Service equivalents</td>
<td>USD(A&amp;S), DOT&amp;E, with Service Testers</td>
<td>Q1 FY20</td>
</tr>
<tr>
<td><strong>B2.2</strong> Establish processes for automated security testing, including zero-trust assumptions, automated penetration testing, and red teams for vulnerability scanning</td>
<td>USD(A&amp;S), DOT&amp;E, with Service Testers</td>
<td>Q1 FY20</td>
</tr>
<tr>
<td><strong>B2.3</strong> Identify initial programs to use tools and workflows</td>
<td>SAE</td>
<td>Q1 FY20</td>
</tr>
<tr>
<td><strong>B2.4</strong> Implement minimum viable product (MVP) tools and workflows on digital infrastructure (Rec B1)</td>
<td>SAE, DOT&amp;E, with PMOs</td>
<td>Q2 FY20</td>
</tr>
<tr>
<td><strong>B2.5</strong> Migrate initial programs to digital infrastructure using automated T&amp;E</td>
<td>PEO, with Responsible Organizations</td>
<td>Q3 FY20</td>
</tr>
<tr>
<td><strong>B2.6</strong> Use tools and workflows, identify lessons learned and improvements (using DevSecOps iterative approach)</td>
<td>Service Testers, with PEO/PM</td>
<td>Q4 FY20</td>
</tr>
<tr>
<td><strong>B2.7</strong> Modify tools and workflows, document procedures</td>
<td>Responsible Organizations, Service Testers</td>
<td>Q4 FY20</td>
</tr>
</tbody>
</table>
# Recommendation B3

<table>
<thead>
<tr>
<th>Line of Effort</th>
<th>Create and maintain cross-program/cross-service digital infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommendation</td>
<td>Create a mechanism for Authority to Operate (ATO) reciprocity within and between programs, Services, and other DoD agencies to enable sharing of software platforms, components, and infrastructure, and rapid integration of capabilities across (hardware) platforms, (weapons) systems, and Services.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Draft Implementation Plan</th>
<th>Lead Stakeholder</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>B3.1</strong> Issue guidance making reciprocity the default practice in DoD with limited exceptions and update DoDI 8510.01 to reflect updated risk management framework. Exceptions should require signoff by the DoD CIO to discourage their use</td>
<td>DoD CIO, with Service CIOs</td>
<td>Q3 FY19</td>
</tr>
<tr>
<td><strong>B3.2</strong> Establish DoD-wide repository for ATO artifacts with tools and access rules that enable Services to identify existing ATOs and utilize them when possible</td>
<td>DoD CIO, with Service CIOs, DISA</td>
<td>Q4 FY19</td>
</tr>
<tr>
<td><strong>B3.3</strong> Implement procedures and access controls so that Authorizing Officials have visibility over other programs that are using compatible ATOs</td>
<td>DoD CIO, with Service CIOs, DISA</td>
<td>Q2 FY20</td>
</tr>
<tr>
<td><strong>B3.4</strong> Implement mechanisms to allow FedRAMP and other non-DoD security certifications to be used for DoD ATO when appropriate based on intended use and environment</td>
<td>DoD CIO, with FedRAMP</td>
<td>Q4 FY20</td>
</tr>
</tbody>
</table>
## RECOMMENDATION C1

**Line of Effort**: Create new paths for digital talent (especially internal talent)

**Recommendation**: Create software development units in each Service consisting of military and civilian personnel who develop and deploy software to the field using DevSecOps practices.

<table>
<thead>
<tr>
<th>Draft Implementation Plan</th>
<th>Lead Stakeholders</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C1.1</strong> Exercise existing acquisition and cybersecurity hiring authorities to increase the number of software developers in DoD programs with vacant positions</td>
<td>SAE, PEO, with CIO (cyber excepted service ability)</td>
<td>Immediately</td>
</tr>
<tr>
<td><strong>C1.2</strong> Create new military occupational specialty (MOS) and core occupational series plus corresponding career tracks for each Service; use to grow digital talent for DevSecOps</td>
<td>J1 and comparable X1 for each Service with USD(P&amp;R)</td>
<td>Q1 FY20</td>
</tr>
<tr>
<td><strong>C1.3</strong> Create regulations to allow standard identification, recruitment, and onboarding of experienced civilian software talent, especially on rotation from private sector roles</td>
<td>USD(P&amp;R)</td>
<td>Q1 FY20</td>
</tr>
<tr>
<td><strong>C1.4</strong> Create mechanism for tracking software development expertise and use as preferred experience for promotion into software engineer and acquisition roles</td>
<td>A&amp;S, CIO</td>
<td>Q2 FY20</td>
</tr>
<tr>
<td><strong>C1.5</strong> Obtain additional manpower authorizations for military and civilian software developers</td>
<td>USD(A&amp;S), with USD(P&amp;R), SAE</td>
<td>FY20, FY21</td>
</tr>
<tr>
<td><strong>C1.6</strong> Stand up one or more software factories within each Service, tied to field needs that can be satisfied through organic software development groups</td>
<td>SAEs, with PEOs Digital</td>
<td>FY20 (pilot), FY21 (scale)</td>
</tr>
</tbody>
</table>
**RECOMMENDATION C2**

<table>
<thead>
<tr>
<th>Line of Effort</th>
<th>Create new paths for digital talent (especially internal talent)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Recommendation</strong></td>
<td>Expand the use of (specialized) training programs for CIOs, SAEs, PEOs, and PMs that provide (hands-on) insight into modern software development (e.g., agile, DevOps, DevSecOps) and the authorities available to enable rapid acquisition of software.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Draft Implementation Plan</th>
<th>Lead Stakeholders</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C2.1</strong></td>
<td>USD(A&amp;S), SAEs with DAU</td>
<td>Q4 FY19</td>
</tr>
<tr>
<td>Leverage existing training venues to add content about modern software development practices</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>C2.2</strong></td>
<td>A&amp;S with SAEs, USD(P&amp;R)</td>
<td>FY20 (MVP), FY21 (scale)</td>
</tr>
<tr>
<td>Create and provide training opportunities via boot camps and rotations for acquisition professionals to obtain hands-on experience in DevSecOps programs</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>C2.3</strong></td>
<td>USD(A&amp;S), SAE</td>
<td>Q2 FY20</td>
</tr>
<tr>
<td>Develop additional training opportunities for key leaders about modern software development practices</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>C2.4</strong></td>
<td>A&amp;S, DAU</td>
<td>Q3 FY20</td>
</tr>
<tr>
<td>Create software continuing education programs and requirements for CIOs, SAEs, PEOs and PMs modeled after MCLE (Minimum Continuing Legal Education) for lawyers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## RECOMMENDATION D1

**Line of Effort** | **Change the practice of how software is procured and developed**
---|---
**Recommendation** | Require access to source code, software frameworks, and development toolchains – with appropriate IP rights – for DoD-specific code, enabling full security testing and rebuilding of binaries from source.

<table>
<thead>
<tr>
<th>Draft Implementation Plan</th>
<th>Lead Stakeholders</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D1.1</strong></td>
<td>USD(A&amp;S)</td>
<td>Q3 FY19</td>
</tr>
<tr>
<td>Work with industry to modernize policies for software code ownership, licensing, and purchase. See <a href="#">2018 Army IP directive</a> as an example</td>
<td>USD(A&amp;S)</td>
<td>Q3 FY20</td>
</tr>
<tr>
<td><strong>D1.2</strong></td>
<td>USD(A&amp;S)</td>
<td>Q3 FY20</td>
</tr>
<tr>
<td>Modify FAR/DFARS guidance to require software source code deliverables for GOTS and for government-funded software development. Obtain rights for access to source code for COTS wherever possible (and useful)</td>
<td>USD(A&amp;S)</td>
<td>Q3 FY20</td>
</tr>
<tr>
<td><strong>D1.3</strong></td>
<td>USD(A&amp;S)</td>
<td>Q3 FY20</td>
</tr>
<tr>
<td>Modify DoDI 5000.02 and DoDI 5000.75 to make access to code and development environments the default</td>
<td>USD(A&amp;S), with CIO</td>
<td>Q4 FY20</td>
</tr>
<tr>
<td><strong>D1.4</strong></td>
<td>USD(A&amp;S), with CIO</td>
<td>Q4 FY20</td>
</tr>
<tr>
<td>Develop a comprehensive source code management plan for DoD including the safe and secure storage, access control, testing and field of use rights</td>
<td>USD(A&amp;S), with CIO</td>
<td>Q4 FY20</td>
</tr>
</tbody>
</table>
# RECOMMENDATION D2

## Change the practice of how software is procured and developed

### Recommendation

Make security a first-order consideration for all software-intensive systems, recognizing that security at the border is not enough.

<table>
<thead>
<tr>
<th>Draft Implementation Plan</th>
<th>Lead Stakeholders</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>D2.1 Adopt standards for secure software development and testing that use a zero-trust security model</td>
<td>CIO, with DDS</td>
<td>Q3 FY19</td>
</tr>
<tr>
<td>D2.2 Develop, deploy, and require the use of IA-accredited (commercial) development tools for DoD software development</td>
<td>CIO, PEO Digital</td>
<td>Q4 FY19</td>
</tr>
<tr>
<td>D2.3 Establish automated penetration testing as part of OT&amp;E evaluation (integrated with program development)</td>
<td>DOT&amp;E</td>
<td>Q1 FY20</td>
</tr>
<tr>
<td>D2.4 Establish red team responsible for ongoing vulnerability testing against any defense software system</td>
<td>CIO with DDS</td>
<td>Q2 FY20</td>
</tr>
<tr>
<td>D2.5 Establish security as part of the selection criteria for software programs</td>
<td>A&amp;S with CIO, SAEs</td>
<td>Q3 FY20</td>
</tr>
</tbody>
</table>
## RECOMMENDATION D3

**Line of Effort**  
Change the practice of how software is procured and developed

### Recommendation

Shift from the use of rigid lists of requirements for software programs to a list of desired features and required interfaces/characteristics, to avoid requirements creep, overly ambitious requirements, and program delays.

### Draft Implementation Plan

<table>
<thead>
<tr>
<th>Draft Implementation Plan</th>
<th>Lead Stakeholders</th>
<th>Target Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify requirements guidance by memo to shift from a list of requirements for software to a list of desired features and required interfaces/characteristics</td>
<td>USD(A&amp;S), CMO</td>
<td>Q4 FY19</td>
</tr>
<tr>
<td>Update CJCSI 3170.01H (JCIDS requirements process) to reflect contents of guidance memos</td>
<td>Joint Staff</td>
<td>Q1 FY20</td>
</tr>
<tr>
<td>Modify DoDI 5000.02 and DoDI 5000.75 (or integrate into new DoDI 5000.SW)</td>
<td>USD(A&amp;S)</td>
<td>Q2 FY20</td>
</tr>
<tr>
<td>Define and use new budget exhibits for software programs using evolving lists of features in place of requirements (see also Rec A2)</td>
<td>USD(A&amp;S), with USD(C), CAPE, HAC-D, SAC-D</td>
<td>Q3 FY20</td>
</tr>
</tbody>
</table>
SUPPORTING INFORMATION

1. Vignettes
2. Draft Implementation Plans
3. Legislative and Regulatory Changes
4. Modern Alternative to P- and R-Forms
5. Frequently Asked Questions
6. DIB Guides for Software
7. SWAP Working Group Reports
8. Data Analysis and Machine Learning
9. Acronyms
10. Required Content
## DIB Metrics for Software

<table>
<thead>
<tr>
<th>#</th>
<th>Metric</th>
<th>Target value (by software type)</th>
<th>Typical DoD values for SW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>COTS\textsuperscript{ii} apps</td>
<td>Custom\textsuperscript{iii} -ized SW</td>
</tr>
<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>
</tr>
<tr>
<td>2</td>
<td>Time to field high priority fcn (spec → ops) or fix newly found security hole (find → ops)\textsuperscript{vi}</td>
<td>N/A</td>
<td>&lt;1 mo</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>
</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>
</tr>
<tr>
<td>5</td>
<td>Time required to restore service after outage</td>
<td>&lt;1 hr</td>
<td>&lt;6 hr</td>
</tr>
<tr>
<td>6</td>
<td>Automated test coverage of specs / code</td>
<td>N/A</td>
<td>&gt;90%</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>
</tr>
<tr>
<td>8</td>
<td>Change failure rate (rollback deployed code)</td>
<td>&lt;1%</td>
<td>&lt;5%</td>
</tr>
<tr>
<td>9</td>
<td>% code available to DoD for inspection/rebuild</td>
<td>N/A</td>
<td>100%</td>
</tr>
<tr>
<td>10</td>
<td>Complexity metrics</td>
<td>#/type of specs structure of code</td>
<td># programmers</td>
</tr>
<tr>
<td>11</td>
<td>Development plan/environment metrics</td>
<td>#/type of platforms</td>
<td>#/skill level of teams type of deployments</td>
</tr>
<tr>
<td>12</td>
<td>“Nunn-McCurdy” threshold (for any metric)</td>
<td>1.1X</td>
<td>1.25X</td>
</tr>
</tbody>
</table>
Generating a Cost Estimate for Agile Software Development

“Succeeding at an Impossible Task”

Matthew R. Kennedy, PhD
Email: MatthewRKennedyEmail@gmail.com
Disclaimer

• The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation

• References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring

• This material is furnished on an "as-is" basis. The author makes no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material
Overview

• Why is cost estimation for software development challenging?
• How can agile development help?
• Example
Traditional Ecosystem

- Requirements
- Funding

Traditional Program Ecosystem

- Process
- Metrics
- Program Management
- Engineering/Development

- Contracts

Tools

Input

Output

Users

Capability
Traditional Ecosystem (Input)

Requirement 1

3.2.5.1 Nuts, almonds, shelled. Shelled almond pieces shall be of the small piece size classification and shall be U.S. No. 1 Pieces of the U.S. Standards for Grades of Shelled Almonds. A minimum of 95 percent, by weight, of the pieces shall pass through a 4/16-inch diameter round hole screen and not more than 5 percent, by weight, shall pass through a 2/16-inch diameter round hole screen. The shelled almonds shall be coated with an approved food grade antioxidant and shall be of the latest season’s crop.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Analysis</th>
<th>Design</th>
<th>Dev</th>
<th>Test</th>
<th>Deploy</th>
<th>TOTAL (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement 1</td>
<td>$10.00</td>
<td>$10.00</td>
<td>$50.00</td>
<td>$30.00</td>
<td>$10.00</td>
<td>$110.00</td>
</tr>
<tr>
<td>Requirement 2</td>
<td>$8.00</td>
<td>$8.00</td>
<td>$40.00</td>
<td>$25.00</td>
<td>$8.00</td>
<td>$89.00</td>
</tr>
<tr>
<td>Requirement n</td>
<td>$15.00</td>
<td>$20.00</td>
<td>$80.00</td>
<td>$60.00</td>
<td>$25.00</td>
<td>$200.00</td>
</tr>
<tr>
<td><strong>TOTAL (m)</strong></td>
<td><strong>$33.00</strong></td>
<td><strong>$38.00</strong></td>
<td><strong>$170.00</strong></td>
<td><strong>$115.00</strong></td>
<td><strong>$43.00</strong></td>
<td><strong>$399.00</strong></td>
</tr>
</tbody>
</table>
Traditional Ecosystem

Traditional Program Ecosystem

Perfect Alignment

Process
Metrics
Program Management

Contracts
Engineering

Tools

Input
Output

Capability

Poor Outcomes

Traditional Ecosystem

Commercial Pesticides and Oils Exempt from the Federal Insecticide, Fungicide, and Rodenticide Act (FIFRA) and the Federal Food, Drug, and Cosmetic Act (FFDCA) until registration is completed. However, the following restrictions apply:

3.2.5.1 Insecticides. Commercial Pesticides and Oils Exempt from the Federal Insecticide, Fungicide, and Rodenticide Act (FIFRA) and the Federal Food, Drug, and Cosmetic Act (FFDCA) until registration is completed. However, the following restrictions apply:

- All registered commercial pesticides and oils exempt from FIFRA and FFDCA must be labeled according to the current registration label and must be used in accordance with the label specifications.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only on labeled crops and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be stored in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only by trained and certified applicators.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be applied in a manner that results in off-target effects.
- Commercial pesticides and oils exempt from FIFRA and FFDCA must be used only in accordance with the label specifications and must not be transported in a manner that results in off-target effects.
- Commercial pesticide...
We Can’t Gather All of the Requirements Up-Front

*The top three indicators found in challenged projects are:
1. Lack of user input
2. Incomplete Requirements & Specifications
3. Changing Requirements & Specifications

Program “B” Requirements Volatility

Volatility metrics track the number and type of changes over time


*Standish report was published in 1994
We Don’t Know the Efficiency of the Workforce
2 (of many) Reasons Cost Estimation for Software Development is Challenging?

1. We can’t gather all of the requirements up front

   Requirements Delivered
   Requirements Locked Down

2. We don’t know the efficiency of the workforce
How Can Agile Development Help?

1) Requirements are continually (Re)assessed

2) Stable Teams

Multidisciplinary-Team Focused
Capacity-Based Estimation

- \#Team \times (#Sprints \times Burn-Rate of Team)
  - Estimation Data
    - 8-person team has a burn rate of $29k/Wk.
    - 2-week sprints ($58k/Sprint)
  
  Capacity Estimate
  3 Teams of 8 x 5 Sprints = $870k
  3 x ($58k \times 5)

Vision

- Capability 1 ($7.5M)
- Capability 2 ($3.75M)
- Feature 1 $870k
- Feature 2 $1.16M
- Feature 3 $1.75M
Tracking Project Progress (Feature 1)

- Refine estimates throughout development based on value delivered vs. estimate
  - Add Teams (Capacity)
  - (Re)move Teams (Capacity)
  - Do Nothing

Think Teams **NOT** Individuals

<table>
<thead>
<tr>
<th>Initial Team Size</th>
<th>New Team Size</th>
<th>Iteration +1</th>
<th>Iteration +2</th>
<th>Iteration +3</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>−20%</td>
<td>−4%</td>
<td>+12%</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>0%</td>
<td>−6%</td>
<td>+15%</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>−12%</td>
<td>−8%</td>
<td>−8%</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>−20%</td>
<td>−20%</td>
<td>−16%</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>−15%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Succeeding with Agile (Mike Cohn)
Summary

- Align your tracking strategy with the Agile Ecosystem
  - All agile ecosystems are different
- Think teams **not** individuals
- Don’t forget about other costs:
  - Hardware
  - Program Management
  - Computing Costs (including cloud services)
  - Licensing
  - Software Development Support (Architecture, etc.)
Matthew R. Kennedy, PhD

• Matthew R. Kennedy is a Senior IT Program Manager and Contracting Officer Representative (COR) at the Office of the Comptroller of the Currency (OCC). Formerly, Matt was a Program Manager at the Army's Program Executive Office - Enterprise Information Systems (PEO-EIS) and was a Professor of Software Engineering at Defense Acquisition University (DAU) where he specialized in agile acquisition. Matt served as the Associate Director of Engineering at the National Cancer Institute’s Center for Biomedical Informatics and Information Technology and served in the U.S. Air Force as a network intelligence analyst. He has worked both inside and outside of the government on various IT projects over the last 18 years.

• Matthew holds a Bachelors in Computer Science, and a masters and PhD in Computer Science and Software Engineering from Auburn University. He is Defense Acquisition Workforce Improvement Act (DAWIA) Level III certified in Program Management, Systems Engineering, and Information Technology (IT).

Contact: MatthewRKennedyEmail@gmail.com
TO BE REMOVED PRIOR TO PRESENTATION
“New Start” Ecosystem

Vision
Funding
Scrum
XP
SAFe
DAD
Kanban
DSDM
Velocity
Burndown
WIP
Cum. Flow
Lead Time
Cycle Time
Docker
Visual Studios
JIRA
Confluence
Jenkins
J-unit
Kubernetes
SQL Server
Input
Process
Contract
Users
Eng/Dev
PM
Output
Capacity
Licenses
Tools
Agile Product Ecosystem
Capability
Comparison (Traditional vs. Agile)

<table>
<thead>
<tr>
<th></th>
<th>Individuals</th>
<th>Teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost per Individual per Hour</td>
<td>![Clock Icon]</td>
<td>![Team Icon]</td>
</tr>
<tr>
<td>Cost per Requirement per Phase</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cost per Team per Sprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product ($5.4M)</td>
</tr>
<tr>
<td>MVP (2.4M)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cost per Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Priority 1</td>
</tr>
<tr>
<td>Feature Priority 2</td>
</tr>
<tr>
<td>Feature Priority 3</td>
</tr>
<tr>
<td>Feature Priority 4</td>
</tr>
<tr>
<td>Feature Priority 5</td>
</tr>
</tbody>
</table>
Adaptive Acquisition Framework

aka “The new 5000.02”

Jeff Boleng, OUSD(A&S), Special Assistant for Software Acquisition

IT-CAST, 21 Aug 2019
FOUR LINES OF EFFORT

A. Refactor statutes, regulations, and processes for software

B. Create and maintain cross-program/cross-service digital infrastructure

C. Create new paths for digital talent (especially internal talent)

D. Change the practice of how software is procured and developed
Software is fundamentally different than hardware
• Infinitely malleable
• Highly automatable
• Free to replicate
• Easy/Free to deploy
• Very low cost to operate
• Can/must be continuously refactored and modernized
Why is this so hard?
Where is the Operational User?
Adaptive Acquisition Framework
Tenets of the Defense Acquisition System

1. Simplify Acquisition Policy
2. Tailor Acquisition Approaches
3. Empower Program Managers
4. Data Driven Analysis
5. Active Risk Management
6. Emphasize Sustainment

DoDD 5000.01: The Defense Acquisition System
DoDI 5000.02: Operation of the Adaptive Acquisition Framework

Legend:
DD: Disposition Decision
OD: Outcome Determination
MDD: Material Development Decision
MS: Milestone
IOC: Initial Operational Capability
FOC: Full Operational Capability
S: Sprint
MVP: Minimum Viable Product
MVCR: Minimum Viable Capability Release
R: Release
ATP: Authority to Proceed

July 2019
DoD 5000 Series Policy Development Process

Current DoDI 5000.02

- CORE A&S ACQUISITION POLICY
  - Policy
  - Responsibilities
  - Procedures
  - Decision Points and Phases

- FUNCTIONAL ENCLOSURES
  - Acquisition Categories and Compliance Requirements
  - Program Management
  - Systems Engineering
  - Developmental T&E
  - Operational & Live Fire T&E
  - Life-Cycle Sustainment
  - Human Systems Integration
  - Affordability Analysis and Investment Constraints
  - Analysis of Alternatives
  - Cost Estimating and Reporting
  - Information Technology
  - Urgent Capability Acquisition
  - Cybersecurity

Revised DoD Directive 5000.01

- Revised DoD Instruction 5000.02, Operation of the Adaptive Acquisition Framework

Separately Published Functional Policies

DAU Website

- Do Directive 5000.01
- DoD Instruction 5000.02
- DoD Instructions 5000.xx, (ea. Pathway)
- Functional Policy Documents
- Tables (Milestone Documentation Identification Tool)
- Defense Acquisition Guidebook
- Other Tools

USD(A&S) Initiates Formal Coordination
Comment Adjudication Complete
USD(A&S) Signature

A&S Development, Internal A&S Coordination, Finalize Draft
WHS Pre-Coordination Review, Revisions, 1st Legal Review
Formal DoD Coordination, Finalize Document for Signature
Pre-Signature Review, Final Legal Review, Security Release
Software Acquisition Pathway – draft/pre-decisional
Software Acquisition Pathway – draft/pre-decisional
Software Acquisition Pathway – draft/pre-decisional
Engagement and feedback

• Engagement
  • May – US Chamber of Commerce
  • May - 16th Annual Acquisition Research Symposium
  • July - feedback session hosted by NDIA, AIA event, quarterly industry association round table
  • August – PEO forum, SW Acq Pathway wargame

• Feedback
  • Need to better describe linkage to system’s engineering process
  • How does this map to embedded software?
  • Where does developmental and operational testing fit in?
  • This will be hard to estimate cost
Can the Independent Cost Assessment become the Independent Capability Assessment?

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

Enterprise DevSecOps
? [SecDevOps | DevSecOps | DevOpsSec] ?
DevSecOps Model: Current

**Planning**
- Customer Involvement
- Issue Tracking
- Threat Modeling
- Release Planning
- Sprint Planning
- Sustainable Velocity

**Development**
- Architecture-First Approach
- Version Control
- Coding Standards
- Dependency Analysis
- Observability
- Test Driven Design
- Peer Review

**Continuous Integration**
- Automated Build
- Automated Unit Testing
- Static Analysis
- Code Quality Metrics
- Release Packaging

**Provision**
- Base Image Provenance
- Infrastructure Automation
- Instance Provisioning
- Credential Management

**Validation**
- Interface Validation
- Integration Testing
- Compliance / Accreditation
- Chaos Engineering
- Dynamic Analysis
- Vulnerability Scanning
- Deployment Validation

**Deployment**
- Deployment Orchestration
- Small Releases
- Canary Deployments
- Rolling Updates
- Instant Rollback

**Operations**
- Continuous Monitoring
- Practice Recovery
- Upstream Feedback

**KEY**
- Existing
- Added
DoD Enterprise DevSecOps Technology Stack (Exemplar)

**Plan & Develop**
- GitHub
- GitLab
- MSBuild
- CMake
- Maven
- Gradle

**Build**
- JUnit
- cucumber

**Test**
- SonarQube
- OWASP
- Qualys
- Contrast Security
- Twistlock
- Archaius
- Aqua

**Secure**
- Nexus
- Archiva
- Nexus

**Store Artifacts**
- Docker
- Kubernetes

**Deploy & Operate**
- Ansible
- SaltStack
- Chef
- Operator SDK
- ELK Stack
- Nagios
- Splunk
- New Relic

**Monitor**
- Prometheus
- Sensu

**Scale**
- AWS GovCloud
- Google Cloud Platform
- Azure
- Amazon Web Services

**Orchestration**
- Jenkins

**Continuous Integration & Continuous Delivery**
What is the DoD Enterprise DevSecOps Initiative?

- Joint Program with OSD A&S, DoD CIO and the DoD Services.

- Technology:
  - Selecting, certifying, and packaging best of breed development tools and services (over 100 options)
  - Creating the Sidecar Container Security Stack (SCSS) for baked-in zero trust security
  - Creating a Centralized artifacts repository of hardened and centrally authorized containers
  - Designing a Scalable Microservices Architecture with Service Mesh/API Gateway and baked-in security
  - Providing on-boarding and support for adoption of Agile and DevSecOps
  - Developing best-practices, training, and support for pathfinding and related activities

- Standardizing metrics and define acceptable thresholds for continuous ATO
- Working with DAU to bring state of the art DevSecOps curriculum
- Creating new contracting language to enable and incentivize the use of Agile and DevSecOps
Value for DoD Programs (1)

- Enables any DoD Program across DoD Services deploy a DoD hardened Software Factory, on their existing or new environments (including classified, disconnected and Clouds), within days instead of a year. Tremendous cost and time savings.

- Multiple DevSecOps pipeline exemplars are available with various options to avoid vendor lock-in and enable true DoD-scale as there is not a one-size-fit-all for CI/CD.

- Enables rapid prototyping (in days and not months or years) for any Business, C4ISR and Weapons system. Deployment in PRODUCTION!

- Enables learning and continuous feedback from actual end-users (warfighters).
Value for DoD Programs (2)

- Enables bug and security fixes in minutes instead of weeks/months.
- Enables automated testing and security.
- Enables **continuous Authorization to Operate (ATO)** process for rapid deployment and scalability. **Authorize ONCE, use MANY times!**
- Brings a holistic and **baked-in cybersecurity stack**, gaining complete visibility of all assets, software security state and infrastructure as code.
- Microservices Architecture to facilitate the adoption of microservices
- Deployed on any environment, including DoD-approved Cloud and Jedi (when available).
Create and Maintain DevSecOps pipelines (and not just DevOps) to avoid each DoD services building their own stack and reinventing the wheel.

Create hardened Container images in a dedicated artifacts repository with security built-in and compliance with FedRAMP/NIST (similar to gold images concept).

Create a Microservice Service Architecture with Service Mesh (ISTIO)

Standardize metrics and define acceptable thresholds for test coverage, security, documentation etc. to enable complete continuous deployment with pre-ATO embedded.

Leverage Kubernetes for Orchestration to ensure automation, rolling-update, scale, security and visibility thanks to the sidecar security container concept.
Cloud One: new Air Force Cloud Offering

- Former Common Computing Environment (CCE), PEO C3I&N
- Cloud One provides:
  - Access to AWS GovCloud and Azure Government on:
    - Impact Level (IL) 2, 4 and 5, today
    - IL 6 and Secret SAP (C2S) within December 2019
  - Cybersecurity Services (CSSP) baked-in
  - Cloud Access Point (CAP) and GCDS baked-in
  - Single Sign On
  - Zero Trust model
- Pay per use scalable model (pay for your compute, storage and shared services), as easy as MIPRing money.
- Enables instantiation of DevSecOps environment in your dedicated VPCs (Development VPC with internet access (including at IL5) and Production VPC) in days with Continuous ATO and full DoD-wide reciprocity
- Building a new multi-award ID/IQ contract vehicle to buy licenses, services (including consultants, FTEs etc.) and Clouds services in bulk (within 90 days).
LevelUP: new centralized Air Force Software Factory Team

- Merged with top talent across U.S. Air Force from various Factories (Kessel Run, Kobayoshi Maru SpaceCAMP and Unified Platform).
- Manages Software Factories for Development teams so they can focus on building mission applications.
- Decouples Development Teams from Factory teams with DevSecOps and Site Reliability Engineer (SRE) expertise.
- Helps instantiate DevSecOps CI/CD pipeline / Software Factories in days at various classification levels.
- Leverages the DoD hardened containers while avoiding one-size-fits-all architectures.
- Fully compliant with the DoD Enterprise DevSecOps Initiative (DSOP) with DoD-wide reciprocity.
- Centralizing the Container Hardening of 172 enterprise containers (databases, development tools, CI/CD tools, cybersecurity tools etc.).
- Launching Software Enterprise Services (within 90 days for first chat tools) with Collaboration tools, Cybersecurity tools, Source code repositories, Artifact repositories, Development tools, DevSecOps as a Service, Chats etc. These services will be MANAGED services on Cloud One by our SRE team so development teams can simply USE those tools and pay per use at scale with bulk licenses.
DoD Enterprise DevSecOps Architecture*

*each DoD Program can have its own instantiation of the DoD Enterprise DevSecOps Platform on any Cloud.
** can be installed with single command and deployed on any Cloud.
*** could be deployed inside an enclave or on-premises
**** gives complete visibilities of assets, security/vulnerability state etc. can be integrated to existing cybersecurity shared services.
Questions and Feedback
A FOUNDATION FOR AGILE ACQUISITION DECISIONS

TORY CUFF

IT CAST 21 AUG 19
information:

knowledge:
A UNIVERSAL APPROACH TO MEASUREMENT

1. Define the decision.
2. Determine what you know now.
3. Compute the value of additional information. (If none, go to step 5.)
4. Measure where information value is high. (Return to steps 2 and 3 until further measurement is not needed.)
5. Make a decision and act on it. (Return to step 1 and repeat as each action creates new decisions.)
THE DECISION
FY 2019 National Defense Budget Request

(Dollars in Billions)

- DoD – Military - Base Budget Request: $617
- DoD - Military - Overseas Contingency Operations (OCO): $69
  - Total DoD Military (051): $686
- Other National Defense (053/054): $30
  - Total National Defense (050): $716

- Budget driven by the National Defense Strategy
- $74 Billion (10% real growth) over DoD’s current CR levels
- Appreciative of the Congress enacting the Bipartisan Budget Act of 2018
- We are committed to being good stewards of the taxpayers’ money
CURRENT STATE
National Defense Strategy

- Compete, Deter, and Win to Preserve Peace through Strength
  - Expand the competitive space leveraging all elements of national power

- Competition with China and Russia is central challenge
  - Continue efforts to deter and counter North Korea, Iran, and terrorists

- Sustain U.S. influence and ensure favorable regional balances of power
  - Build a more lethal, resilient, agile, and ready Joint Force
  - Strengthen alliances and attract new partners
  - Reform the Department’s business practices for greater performance and affordability

Great Power Competition is now the Primary Focus

ASSUMPTIONS

1. Our mission is vital
"Organizations which design systems...are constrained to produce designs which are copies of the communication structures of these organizations."

-M. Conway
During stable times organizations are tempted to build big systems—multi-year projects of brain melting complexity, like the death star. Despite these large programs and projects rarely working they’ve become the standard approach in many organizations.

-GHCQ: Boiling Frogs? Technology organizations need to change to radically survive increasing technical and business disruptive
ASSUMPTIONS

1. Our mission is vital

2. The DoD Innovation Ecosystem is complex
TECHNOLOGY ADOPTION
ASSUMPTIONS

1. Our mission is vital
2. The DoD Innovation Ecosystem is complex
3. Rate of technology adoption and integration is increasing
CURRENT PROJECT TIMELINES (1 OF 2)

Shows cycle times for Major Defense Acquisition Program (MDAP) from initial development to Initial Operational Capability (or equivalent)

*NOTE: does not include timeline for requirement definition/initial funding

Median Cycle Time: 8 Years

CURRENT PROJECT TIMELINES (2 OF 2)

12-16 YEARS
ASSUMPTIONS

1. Our mission is vital
2. The DoD Innovation Ecosystem is complex
3. Rate of technology adoption and integration is increasing
4. Our current acquisition timelines exceed rates of technology change
“We have the money, ladies and gentlemen, but one thing that money can’t buy is time and we don’t have time to do business as usual.”
-SecNav Richard Spencer

“...rapidly integrating more commercial sector technology is paramount for DOD as it seeks to retain a technical edge.”
-Deputy Director of Defense, Patrick Shanahan

“There is an undeniable urgency to develop and deploy software faster, faster than our adversaries, in order to maintain strategic and tactical advantage.”
-Undersecretary of Defense, Ms. Ellen Lord

“Software is our big issue...Software intensive programs are almost all over cost, over schedule.”
-Assistant Secretary of the AF AT&L, Dr. William Roper
ASSUMPTIONS

1. Our mission is vital
2. The DoD Innovation Ecosystem is complex
3. Rate of technology adoption and integration is increasing
4. Our current acquisition timelines exceed rates of technology change
5. Executive support
business is change.
	here is nothing else.

@gapingvoid
A PROPOSED PROCESS
Without defined processes, you can’t scale, you can’t put metrics and instrumentation in place...you can’t manage.

1. Leverage commercial cloud or DoD cloud (if data encryption or operational environment requires)
THERE IS NO CLOUD

IT'S JUST SOMEONE ELSE'S COMPUTER
1. Leverage commercial cloud or DoD cloud (if data encryption or operational environment requires)

2. Breakdown complex systems or projects to smaller products
1. Leverage commercial cloud or DoD cloud (if data encryption or operational environment requires)

2. Breakdown complex systems or projects to smaller products

3. Constrain cost and schedule, i.e. Duration-based cost estimating
   - Current proposal/product: 2 Years and Fixed Funding $5M
1. Leverage commercial cloud or DoD cloud (if data encryption or operational environment requires)

2. Breakdown complex systems or projects to smaller products

3. Constrain cost and schedule, i.e. Duration-based cost estimating
   - Current proposal / product: 2 Years and Fixed Funding $5M

4. Focus on Value Estimate, i.e. impact in operational environment
When we learn to manage value, our workload and costs reduce because we’re managing the output of our processes, not processes themselves.

GHCQ: Boiling Frogs? Technology organizations need to change to radically survive increasing technical and business disruptive
I NEED A BUDGET ESTIMATE FOR MY PROJECT, BUT I DON’T HAVE A SCOPE OR A DESIGN FOR IT YET.

OKAY, MY ESTIMATE IS $3,583,729.

YOU DON’T KNOW ANYTHING ABOUT MY PROJECT. THAT MAKES TWO OF US.
1. Leverage commercial cloud or DoD cloud (if data encryption or operational environment requires)
2. Breakdown complex systems or projects to smaller products
3. Constrain cost and schedule, i.e. Duration-based cost estimating
   - Current proposal / product: Two (2) Years and Fixed Funding $5M
4. Focus on Value Estimate, i.e. impact in operational environment
5. Provide information for decision
QUESTIONS
Living in a World Without Source Lines Of Code (SLOC)

2019 Software and Information Technology Cost Analysis Solutions Team

August 21, 2019

John Sautter
Cost Analyst

Approved for Public Release #19-1455. Distribution Unlimited
Table of Contents

• Background
• Alternate Sizing Choices
• Simple Functional Sizing
• Other Aspects
• Concluding Remarks
Background

• This presentation deals with alternate forms of software sizing other than Source Lines Of Code (SLOC)

• The context of this presentation revolves around software project estimation and the collection of actual historical results from those projects

• “Metrics …Not SLOC…Please stop now”
  – From Dr. Jeff Belong - Special Assistant for Software Acquisition, OSD(A&S)
    • Software and Information Technology Cost Analysis System Team Workshop
      August 22, 2018 – Last Year’s Workshop

  – The current state of practice within DoD is that software complexity is often estimated based on number of source lines of code (SLOC), and rate of progress is measured in terms of programmer productivity. While both of these quantities are easily measured, they are not necessarily predictive of cost, schedule, or performance.
Alternate Sizing Choices
Alternate Sizing Choices - Agenda

- No Software Sizing
- Review of Inputs of three Key Parametric Tools
- Functional Requirements as Sizing
No Software Sizing

• Why not just use hour(s) estimates?
  – Answer: No completion tracking available
  • If the project uses all the hours is the project done?

• Hours per software change – Best Paper Award ICEAA 2019

<table>
<thead>
<tr>
<th>Model</th>
<th>Conditions</th>
<th>Obs</th>
<th>Adj R²</th>
<th>SEE (Hrs)</th>
<th>PRED(30)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T Hrs = 453 * (TSC)^0.69</td>
<td>All data</td>
<td>329</td>
<td>0.36</td>
<td>48,385</td>
<td>17.3%</td>
</tr>
<tr>
<td>T Hrs = 341 * (TSC)^0.79</td>
<td>10% trimmed data</td>
<td>263</td>
<td>0.57</td>
<td>44,842</td>
<td>23.6%</td>
</tr>
</tbody>
</table>

• In-Flight/In-progress projects
  – Use their collected via their Agile Lifecycle Management (ALM) or project management tools to estimate new work and predict “when will it be done” dates
  • More on the next slide
  • Standardization in this area – highly unlikely

Sizing: Need something that can be estimated and used in collection of historical actuals available for other SW projects
Estimating with our Management Tools

Monte Carlo Simulation tells you how long to finish the issues in your backlog or how many issues you can get done by a certain date.

Reference: ActionableAgile for JIRA – AgileMetrics

For Projects In-Flight – Story/Issue tracking can help in estimation and historicals
Aligning to Candidate Software Metrics

1. **Usefulness metrics**
   1.1. List of features implemented

2. **Process efficiency and effectiveness metrics**
   2.1. Release cadence (time between software feature releases into operations)
   2.2. Technology/Hardware refresh cycle (time between major hardware refreshes)
   2.3. Time required to field high priority functions (specifications → operations) or fix newly found security holes (discovery → operations)
   2.4. Time from code committed to code in use
   2.5. Time required for full regression test and cybersecurity audit/penetration testing
   2.6. Time required to restore service after outage

3. **Quality metrics**
   3.1. Change failure rate (frequency of rollback required for deployed code)
   3.2. Percentage test coverage of features
   3.3. Number of automated tests per feature
   3.4. Percentage test coverage of code (by module or line)
   3.5. Percentage of feature/code tests that are automated
   3.6. Number of cybersecurity/penetration tests

Cost Community Needs Size Drivers Early in Life Cycle
Sizing Methods Used at Various Stages of the Software Development Life Cycle

The sizing method used should be based on available information and where you are in the software development life cycle vs. the “Cone of Uncertainty.”

Quantitative Software Management (QSM) identifies some sizing Through the Cone of Uncertainty.

See: https://www.qsm.com/infographic/software-sizing-matters
Sizing in the QSM SLIM-Suite

- Leverages sizing metrics useful at different points in the software development lifecycle
- Supports size metrics that technical & non-technical people can relate to and that current deliverables support
- Supports gearing factors to translate between different size measures
- Sizing methodology supports re-estimation at regular intervals and multiple sizing techniques
The Size Calculator provides multiple sizing approaches that can be combined to refine size estimates and quantify sizing uncertainty.
SLIM Sizing Approaches

Sizing approaches support early high-level estimation while others provide more detail when that information becomes available. All sizing methods are customizable to specific development environments.

QSM SLIM – Parametric Suite
## True Planning SW Size Inputs

<table>
<thead>
<tr>
<th>Software Size</th>
<th>Size Units</th>
<th>Source Lines of Code (SLOC)</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Size</td>
<td></td>
<td>15,000</td>
<td>0.00%</td>
</tr>
<tr>
<td>New Size Non-executable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adapted Size</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adapted Size Non-executable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percent of Design Adapted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percent of Code Adapted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percent of Test Adapted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design Repeat</td>
<td></td>
<td></td>
<td>0.00%</td>
</tr>
<tr>
<td>Reused Size</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Reused Size Non-executable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reuse Factor</td>
<td></td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>Deleted Size</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Removal Complexity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auto Generated Size</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auto Gen Size Non-executable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auto Translated Size</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auto Trans Size Non-executable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auto Translation Tool Efficiency</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Types of “code”:
- New
- Adapted/Modified
- Reused
- Deleted
- Auto Generated
- Auto Translated

### Calculators
- Source Lines of Code (SLOC)
- Function Point Analysis
- Fast Function Points
- RICEFW Objects
- COSMIC Function Points
- Use Case Conversion Points
- Functional Size

Calculators estimate “size” from user inputs for every type; for example:
- Functions
- Features
- RICE-FW

Inputs entered with a selection of Calculators Accounts for New, Adopted, Reuse, AutoGen, Translated
TruePlanning Parametric Inputs

COSMIC Function Points

RICEFW Objects => Function Points
(Reports, Interfaces, Conversions, Enhancements, Forms, Workflows)

Use Case Conversion Points

Fast Function Points

Also provides the full complement of the IFPUG function points

International Function Point Users Group (IFPUG)

© 2019 Northrop Grumman Corporation
The Size Metric dialog allows the modeler to select from a canned set of choice but also can add to these choices with their own definitions.
SEER for Software will translate into effective size. Historical ESLOC productivity can be used to validate a function based estimate.
Using Functional Requirements Counts

- From Past Study of Software Resource Data Reports (SRDR)
  - Derived Hours Per Software Functional Requirement

<table>
<thead>
<tr>
<th>Project Type</th>
<th>Least (Hours)</th>
<th>Mean (Hours)</th>
<th>Most (Hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Resource Planning (ERP)</td>
<td>125</td>
<td>192</td>
<td>325</td>
</tr>
<tr>
<td>Automated Information System (AIS)</td>
<td>50</td>
<td>141</td>
<td>375</td>
</tr>
<tr>
<td>Defense Software</td>
<td>75</td>
<td>154</td>
<td>300</td>
</tr>
</tbody>
</table>

- Extrapolated from Box Plot Chart
- Hours cover IEEE 12207 type activities

Source: November 16, 2015 Presentation – Dr. Wilson Rosa, Navy Center for Cost Analysis (NCCA)
Early and Quick Function Points

- 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

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Min</th>
<th>ML</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Functional</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component -</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transactions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>EL</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>EI</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>EQ</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>EQ</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>EGH</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>EQ</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>EQA</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>FCH</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

| Basic Functional |     |    |      |
| Component -      |     |    |      |
| Data             |     |    |      |
| IFL | 7 | 7 | 7 |
| IF | 10 | 10 | 10 |
| IFH | 15 | 15 | 15 |
| IF | 5 | 5 | 5 |
| IFM | 7 | 7 | 7 |
| IFH | 10 | 10 | 10 |

<table>
<thead>
<tr>
<th>Level 2</th>
<th>Min</th>
<th>ML</th>
<th>Most</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unclassified</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elementary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GEI</td>
<td>4</td>
<td>4</td>
<td>4.4</td>
</tr>
<tr>
<td>GEO</td>
<td>3.7</td>
<td>3.9</td>
<td>4.1</td>
</tr>
<tr>
<td>GEO</td>
<td>4.9</td>
<td>5.2</td>
<td>5.4</td>
</tr>
<tr>
<td>UGD</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>

| Unclassified     |     |    |      |
| Logical File     |     |    |      |
| GLF | 7.4 | 7.7 | 8.1 |
| FED | 5.2 | 5.4 | 5.7 |
| UGDG | 6.4 | 7.1 | 7.8 |

Use higher levels if the requirements are high level capabilities

Reference: Early and Quick Function Point Manual
Google “E&QFP”
Simple Function Points (SiFP)

Background

• Background
  – Original Paper 2014:
    “An Evaluation of Simple Function Point as a Replacement of IFPUG Function Point”
    By Luigi Lavazza and Roberto Meli

• States functional sizing can be broken down into two
  Basic Functional Components (BFCs)
    1. Logical Data Groups
    2. Elementary Logical Processes

  Much simpler and can be performed in less time
SiFP includes the following two Basic Functional Components (BFCs), the two BFCs become

1. Logical Data Groups ➔ Unspecified Generic Data Group (UGDG) worth 7 SiFPs
2. Elementary Logical Processes ➔ Unspecified Generic Elementary Process (UGEP) worth 4.6 SiFPs

In contrast, (International Function Point Users Group (IFPUG) has five types with levels of complexity

1. Internal Logical Files (ILF) and
2. External Interface Files (EIF) – consider Record Element Types (RETs) and Data Element Types (DETs)
3. External Inputs (EI), and
4. External Outputs (EO), and
5. External Inquires (EQ) – consider File Types References (FTRs) and DETs

In addition, analysis must consider distinguishing differences of ILF vs. EIF and the difference between an EO and an EQ.
Simple Function Points (SiFP)
Time Saved

- Also in contrast to IFPUG method, the SiFP counting method saves time.

Authors claim about 60% saving avoiding complexity determination
Simple Function Points (SiFP) Demystified

- Functional sizes like SiFPs are derived by analyzing the Functional User Requirements (FUR)

- Typical function point counts are usually either counting the size of an existing application or estimating the size a software project

- In sizing a software projects one can use the actual written functional requirements
  - These can be documented Capabilities, EPICs, Features or even User Stories

- Analysis: Look for Verbs (AUDIO) or (CRUD) in the text
  - Add or Create
  - Update
  - Delete
  - Inquiry or Read
  - Output

- Count the Data Groups only once
  - Add a hotel reservation or update a hotel reservation, two elementary processes,
    - one data group = hotel reservation
Function Point Automation

• Counting from the Code and Database Baselines
  – Known Tools --> CAST AIP

• The OMG Standard for Automated Function Points (AFP) was approved as an ISO Standard in 2019: ISO 19515:2019
  – https://www.iso.org/standard/65378.html

• Natural Language Processing Automation
  – LogApps MARINE tool
    • Machine Assisted Requirements Inspection and Evaluation (MARINE)
    • http://logapps.com/marine/
  – ScopeMaster Tool
    • Automating function point sizing from written requirements
    • https://www.scopemaster.com/
Technical Size

• Function Points address functional size

• What about Non-functional size or Technical Size?
  1) Address with parametric settings
  2) Address with IFPUG General Systems Characteristics
     • Go from Unadjusted to Adjusted Function Points
  3) Use IFPUG SNAP Method
     • SNAP => Software Non-functional Assessment Process
     • Method provides an assessment framework of four categories and fourteen sub-categories

• Typical historical data consisting of hours per FP accounts for the majority of the technical size
  – If your organization or application has a big emphasis on technical size then you should use one of approaches above: 1), 2) or 3)
## SNAP Framework Breakdown

<table>
<thead>
<tr>
<th>Data Operations</th>
<th>Interface Design</th>
<th>Technical Environment</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Data Entry Validation</td>
<td>2.1 UI (User Interface) Change</td>
<td>3.1 Multiple Platforms</td>
<td>4.1 Component Based Software</td>
</tr>
<tr>
<td>1.2 Logical and Mathematical Operations</td>
<td>2.2 Help Methods</td>
<td>3.2 Database Technology</td>
<td>4.2 Multiple Input / Output Interfaces</td>
</tr>
<tr>
<td>1.3 Data Formatting</td>
<td>2.3 Multiple Input Methods</td>
<td>3.3 Batch Processes</td>
<td></td>
</tr>
<tr>
<td>1.4 Internal Data Movements</td>
<td>2.4 Multiple Output Methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5 Delivering Added Value to Users by Data Configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If SNAP is used then must track hours per SNAP point separately from FP work.
Other Aspects
Other Aspects Considered

• How do we address “Infrastructure as Code” in our cost analysis world?

• Stronger resilience in terms of test harnesses and test applications become a bigger part of definition of done
  – Build up of full automated testing in Continuous Integration and Continuous Delivery (CI/CD) pipelines not reflected in our historical databases

• What is happening with the Low Code No Code movement and how do we estimate and track that?
  – Is this just more autogen of code?
  – Is this a bit of Model Based Engineering (MBE)?

• Can Digital Engineering with Model Based Engineering advance the production of products and capabilities faster, cheaper, better?
  – Can our cost models easily integrate?

• Expanded use of Data Science, Artificial Intelligence, and Machine Learning will become more valuable to our cost community
Concluding remarks
Conclusions

• Sizing: Need independent variables that can be estimated and used in collection of historical actuals available for other SW projects

• Parametric tools have similar alternate sizing choices and offer a good framework of thinking through complex estimation challenges and seem to stay current with evolving industry

• Function points seem to be a viable sizing option but fast wide spread adoption into Defense and other sectors unlikely

• Expecting our transition into data science and machine learning will be become valuable
  – It’s here we can build upon what we have as a cost community
THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN
Tracking Software Development:
An example of feature-based estimating
Purpose:

- Present an AFCAA perspective on collecting software development metrics for use in tracking execution and forecasting remaining cost & schedule.

Goal:

- Build coalitions between government and industry to exchange data, share lessons learned, and establish best practices concerning software and information technology cost estimation.

The terminology referenced within is one example that may not translate directly across software development efforts – more on that later.
Many levels of communication
Information and data needs vary across the levels
Many levels of communication
Information and data needs vary across the levels
Task hierarchy terminology is not consistently defined or applied

Stories/Story Points are relative measures specific to a team

Looking for consistent content metric across programs

Targeting above the story point level and below a system feature or capability level

For presentation purposes: We will call them subsystem features (SSF)
Three primary elements:
- Productivity
  - Code development throughput
  - Integration throughput
  - Test point throughput
- Capacity
  - FTEs available in dev, int, & test
  - SIL & SEL availability
- Total estimated effort
  - Forecast of total effort needed to meet contracted requirements

Time-Phased Data:
- Actuals for labor hrs
- Capacity for development, integration, & test
- Story points
- Subsystem Features
- System Features
- Integration Points
- Test Points
Approach

- Code Development:
  - Productivity
    - Code development throughput
  - Capacity
    - FTEs available in dev, int, & test
  - Total estimated effort
    - System features
    - Subsystem features

Data:
Requirements
Staffing by FTE
Demonstrated Performance

Productivity rates, capacity, & requirements inform schedule!

Cost & Schedule Estimate

Integrity - Service - Excellence
Tracking Subsystem Features

1.2.5 Avionics
  1.2.5.1 Avionics Integration, Assembly, Test, and Checkout
  1.2.5.2 Communication/Identification
    1.2.5.2.1 Communication/Identification Integration
    1.2.5.2.2 Communication/Identification Software Design
  1.2.5.3 Navigation/Guidance
  1.2.5.4 Mission Computer/Processing
    1.2.5.4.1 Mission Management SW Design and Development
    1.2.5.4.2 Mission Management IATC Design and Development
  1.2.5.5 Fire Control
  1.2.5.6 Data Display and Controls
    1.2.5.6.1 DDC SW Design and Development
  1.2.5.7 Survivability
  1.2.5.8 Reconnaissance
  1.2.5.9 Electronic Warfare
    1.2.5.9.1 Electronic Warfare Design and Development
  1.2.5.10 Automatic Flight Control
  1.2.5.11 Health Monitoring System
  1.2.5.12 Stores Management
    1.2.5.12.1 Stores Management Software Design and Development

Features mapped to lowest possible WBS element

NOTIONAL EXAMPLE ONLY
### Tracking Hours

#### 1.2.5 Avionics
- **1.2.5.1** Avionics Integration, Assembly, Test, and Checkout
- **1.2.5.2** Communication/Identification
  - **1.2.5.2.1** Communication/Identification Integration
  - **1.2.5.2.2** Communication/Identification Software Design
- **1.2.5.3** Navigation/Guidance
- **1.2.5.4** Mission Computer/Processing
  - **1.2.5.4.1** Mission Management SW Design and Development
  - **1.2.5.4.2** Mission Management IATC Design and Development
- **1.2.5.5** Fire Control
- **1.2.5.6** Data Display and Controls
  - **1.2.5.6.1** DDC SW Design and Development
- **1.2.5.7** Survivability
- **1.2.5.8** Reconnaissance
- **1.2.5.9** Electronic Warfare
  - **1.2.5.9.1** Electronic Warfare Design and Development
- **1.2.5.10** Automatic Flight Control
- **1.2.5.11** Health Monitoring System
- **1.2.5.12** Stores Management
  - **1.2.5.12.1** Stores Management Software Design and Development

**SSF 1**

**SSF 2**

**SSF 3**

**SSF 4**

**SSF 5**

**SSF 6**

**SSF 7**

**SSF 8**

**SSF 9**

**SSF 10**

**SSF 11**

**SSF 12**

**SSF 13**

**SSF 14**

**SSF 15**

---

*Hours tracked to the lowest possible WBS element

**NOTIONAL EXAMPLE ONLY**

---

**Integrity - Service - Excellence**
Delta Planned to Delivery

Subsystem Features

Planned Increment

Initial Planned SSF Delivery
Original Baseline
SSFs Delivered To-Date
New Baseline
Ind. Estimated Delivery

Time Now

Illustrated Example
Prior to Execution:

<table>
<thead>
<tr>
<th>Previous Increment's Statistics:</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>14.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staffing:</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Developer Planned Performance:</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>13.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staffing:</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planned Delivery</td>
<td>3450</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Independent Estimate:</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>14.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staffing:</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated Delivery</td>
<td>2841</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What Actually Happened:

<table>
<thead>
<tr>
<th>Midway Through Increment</th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>16.1</td>
<td>15.2</td>
<td>13.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delivered SSF:</td>
<td>425</td>
<td>450</td>
<td>500</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Replan:

<table>
<thead>
<tr>
<th>Replan:</th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>16.1</td>
<td>15.2</td>
<td>13.7</td>
<td>12.7</td>
<td>12.4</td>
<td>12.2</td>
</tr>
<tr>
<td>Delivered SSF:</td>
<td>425</td>
<td>450</td>
<td>500</td>
<td>540</td>
<td>550</td>
<td>560</td>
</tr>
</tbody>
</table>

Estimated:

<table>
<thead>
<tr>
<th>Estimated:</th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td></td>
<td></td>
<td></td>
<td>14.0</td>
<td>14.0</td>
<td>14.0</td>
</tr>
<tr>
<td>Delivered SSF:</td>
<td>425</td>
<td>450</td>
<td>500</td>
<td>489</td>
<td>489</td>
<td>489</td>
</tr>
</tbody>
</table>

Replan comes to fruition. 3025 SSFs delivered.

Bottom Line: Using ratio of hours per SSF allows comparative analysis between increments to project future effort.
Prior to Execution:

<table>
<thead>
<tr>
<th>Previous Increment's Statistics:</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>14.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staffing:</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Developer Planned Performance:

<table>
<thead>
<tr>
<th></th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>13.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staffing:</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planned Delivery:</td>
<td>3450</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Independent Estimate:

<table>
<thead>
<tr>
<th></th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>14.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14.0</td>
</tr>
<tr>
<td>Staffing:</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated Delivery:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2841</td>
</tr>
</tbody>
</table>

What Actually Happened:

Midway Through Increment

<table>
<thead>
<tr>
<th></th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>16.1</td>
<td>15.2</td>
<td>13.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delivered SSF:</td>
<td>425</td>
<td>450</td>
<td>500</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Replan:

<table>
<thead>
<tr>
<th></th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>16.1</td>
<td>15.2</td>
<td>13.7</td>
<td>12.7</td>
<td>12.4</td>
<td>12.2</td>
<td>13.7</td>
</tr>
<tr>
<td>Delivered SSF:</td>
<td>425</td>
<td>450</td>
<td>500</td>
<td>540</td>
<td>550</td>
<td>560</td>
<td>3025</td>
</tr>
</tbody>
</table>

Estimated:

<table>
<thead>
<tr>
<th></th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td></td>
<td></td>
<td></td>
<td>14.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.0</td>
</tr>
<tr>
<td>Delivered SSF:</td>
<td>425</td>
<td>450</td>
<td>500</td>
<td>489</td>
<td>489</td>
<td>489</td>
<td>2841</td>
</tr>
</tbody>
</table>

Replan comes to fruition. 3025 SSFs delivered.

Bottom Line: Using ratio of hours per SSF allows comparative analysis between increments to project future effort.
<table>
<thead>
<tr>
<th>Prior to Execution:</th>
<th>What Actually Happened:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous Increment's Statistics:</td>
<td>Midway Through Increment</td>
</tr>
<tr>
<td>Hrs/SSF: 14.0</td>
<td>Period 1</td>
</tr>
<tr>
<td>Staffing: 40</td>
<td>Hrs/SSF: 16.1</td>
</tr>
<tr>
<td></td>
<td>Delivered SSF: 425</td>
</tr>
<tr>
<td>Developer Planned Performance:</td>
<td>Replan:</td>
</tr>
<tr>
<td>Hrs/SSF: 13.4</td>
<td>Period 1</td>
</tr>
<tr>
<td>Staffing: 50</td>
<td>Hrs/SSF: 16.1</td>
</tr>
<tr>
<td>Planned Delivery: 3450</td>
<td>Delivered SSF: 425</td>
</tr>
<tr>
<td>Independent Estimate:</td>
<td>Estimated:</td>
</tr>
<tr>
<td>Hrs/SSF: 14.0</td>
<td>Period 1</td>
</tr>
<tr>
<td>Staffing: 45</td>
<td>Hrs/SSF: 14.0</td>
</tr>
<tr>
<td>Estimated Delivery: 2841</td>
<td>Delivered SSF: 425</td>
</tr>
</tbody>
</table>

Replan comes to fruition. 3025 SSFs delivered.

Bottom Line: Using ratio of hours per SSF allows comparative analysis between increments to project future effort.
Prior to Execution:

<table>
<thead>
<tr>
<th>Previous Increment's Statistics:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF: 14.0</td>
</tr>
<tr>
<td>Staffing: 40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Developer Planned Performance:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF: 13.4</td>
</tr>
<tr>
<td>Staffing: 50</td>
</tr>
<tr>
<td>Planned Delivery: 3450</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Independent Estimate:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF: 14.0</td>
</tr>
<tr>
<td>Staffing: 45</td>
</tr>
<tr>
<td>Estimated Delivery: 2841</td>
</tr>
</tbody>
</table>

What Actually Happened:

<table>
<thead>
<tr>
<th>Midway Through Increment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period 1</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>Hrs/SSF:</td>
</tr>
<tr>
<td>Delivered SSF:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Replan:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period 1</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>Hrs/SSF:</td>
</tr>
<tr>
<td>Delivered SSF:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Estimated:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period 1</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>Hrs/SSF:</td>
</tr>
<tr>
<td>Delivered SSF:</td>
</tr>
</tbody>
</table>

Replan comes to fruition. 3025 SSFs delivered.

Bottom Line: Using ratio of hours per SSF allows comparative analysis between increments to project future effort.
Prior to Execution:

<table>
<thead>
<tr>
<th>Previous Increment's Statistics:</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>14.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Staffing:</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What Actually Happened:

<table>
<thead>
<tr>
<th>Midway Through Increment</th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>16.1</td>
<td>15.2</td>
<td>13.7</td>
<td>12.7</td>
<td>12.4</td>
<td>12.2</td>
</tr>
<tr>
<td>Delivered SSF:</td>
<td>425</td>
<td>450</td>
<td>500</td>
<td>540</td>
<td>550</td>
<td>560</td>
</tr>
</tbody>
</table>

Developer Planned Performance:

| Hrs/SSF:                 | 13.4     |       |       |       |       |       |
| Staffing:                | 50       |       |       |       |       |       |
| Planned Delivery:        | 3450     |       |       |       |       |       |

Replan:

<table>
<thead>
<tr>
<th>Replan:</th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>16.1</td>
<td>15.2</td>
<td>13.7</td>
<td>12.7</td>
<td>12.4</td>
<td>12.2</td>
<td>13.7</td>
</tr>
<tr>
<td>Delivered SSF:</td>
<td>425</td>
<td>450</td>
<td>500</td>
<td>540</td>
<td>550</td>
<td>560</td>
<td>3025</td>
</tr>
</tbody>
</table>

Independent Estimate:

| Hrs/SSF:                 | 14.0     |       |       |       |       |       |
| Staffing:                | 45       |       |       |       |       |       |
| Estimated Delivery:      | 2841     |       |       |       |       |       |

Estimated:

<table>
<thead>
<tr>
<th>Estimated:</th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Period 5</th>
<th>Period 6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/SSF:</td>
<td>14.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.0</td>
<td>14.0</td>
</tr>
<tr>
<td>Delivered SSF:</td>
<td>425</td>
<td>450</td>
<td>500</td>
<td>489</td>
<td>489</td>
<td>489</td>
<td>2841</td>
</tr>
</tbody>
</table>

Replan comes to fruition. 3025 SSFs delivered.

Bottom Line: Using ratio of hours per SSF allows comparative analysis between increments to project future effort.
### Example

**Bottom Line:** Using ratio of hours per SSF allows comparative analysis between increments to project future effort.

<table>
<thead>
<tr>
<th>Prior to Execution:</th>
<th>What Actually Happened:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Previous Increment's Statistics:</strong></td>
<td><strong>Midway Through Increment</strong></td>
</tr>
<tr>
<td>Hrs/SSF: 14.0</td>
<td>Period 1</td>
</tr>
<tr>
<td>Staffing: 40</td>
<td>Hrs/SSF: 16.1</td>
</tr>
<tr>
<td><strong>Developer Planned Performance:</strong></td>
<td>Delivered SSF: 425</td>
</tr>
<tr>
<td>Hrs/SSF: 13.4</td>
<td></td>
</tr>
<tr>
<td>Staffing: 50</td>
<td></td>
</tr>
<tr>
<td>Planned Delivery: 3450</td>
<td></td>
</tr>
<tr>
<td><strong>Independent Estimate:</strong></td>
<td><strong>Replan:</strong></td>
</tr>
<tr>
<td>Hrs/SSF: 14.0</td>
<td>Period 1</td>
</tr>
<tr>
<td>Staffing: 45</td>
<td>Hrs/SSF: 16.1</td>
</tr>
<tr>
<td>Estimated Delivery: 2841</td>
<td>Delivered SSF: 425</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Estimated:</strong></td>
<td><strong>Estimated:</strong></td>
</tr>
<tr>
<td>Hrs/SSF: 14.0</td>
<td>Period 1</td>
</tr>
<tr>
<td>Staffing: 45</td>
<td>Hrs/SSF: 14.0</td>
</tr>
<tr>
<td>Estimated Delivery: 2841</td>
<td>Delivered SSF: 425</td>
</tr>
</tbody>
</table>

Replan comes to fruition. 3025 SSFs delivered.
Challenges

Accumulating enough data for:
- Normalization
- Trend development
- Early development planning

Working across contractors:
- Handling different software development methodologies
- Availability & Fidelity of data
QUESTIONS?
Implementing Continuous Iterative Development in DoD Acquisition

Summary Progress and Status

IT CAST
August 2019
# Summary Progress and Status

**Key Initiatives**

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect industry feedback on agile/CID info needs and measures (DSB, DIB, ...)</td>
<td>Kickoff at PSM User Conf (Sep ’18). Conducted surveys (~60; PSM, INCOSE, NDIA, SERC). • Analysis results: brief at PSM 10/19, NDIA 10/19 (see summary excerpts in backup)</td>
</tr>
<tr>
<td>Develop consensus CID measurement framework (PSM)</td>
<td>Formed PSM/INCOSE/NDIA SME WG. Draft ICM table &amp; indicator specs late Jul for review. Seeking additional reps for core team and reviewers. POC: Cheryl Jones, Geoff Draper, Larri Rosser</td>
</tr>
<tr>
<td>Provide industry feedback to DoD on draft SW acquisition policy (“SW 5000.02”)</td>
<td>Industry review in progress – seeking INCOSE input. Comments due to DoD early August. POC: G.Draper, C.Jones, G.Roedler, R.Yeman</td>
</tr>
<tr>
<td>Industry collaboration and outreach to further consensus on agile/CID development and measures.</td>
<td>• PSM CID WG. PSM User’s Group (Sep ’19). • NDIA SE Conference (Oct ’19)</td>
</tr>
</tbody>
</table>

**Studies and Publications**

- **2019 Reports**
  - [Implementing Continuous Iterative Development and Acquisition](https://innovation.defense.gov/software/)
  - [DSB SW Task Report](https://innovation.defense.gov/software/) and [DIB SWAP Study](https://www.ndia.org/divisions/systems-engineering/studies-and-publications)

- [NDIA, INCOSE and PSM Continuous Iterative Development and Sustainment (CID) Study Group](https://www.ndia.org/divisions/systems-engineering/studies-and-publications)

- [DoD DIB SWAP Study](https://innovation.defense.gov/software/)

- [NDIA SE Conference (Oct ’19)](https://www.ndia.org/divisions/systems-engineering/studies-and-publications)

- [NDIA SE Conferences](https://innovation.defense.gov/software/)
Developing a consensus measurement framework for agile development

PSM is leading an initiative, in conjunction with NDIA and INCOSE, to develop a measurement framework for agile SW development.

- ICM Table (Information Needs, Measurable Concepts, Information Needs, Potential Measures)

- Measurement indicator specifications (draft examples)

<table>
<thead>
<tr>
<th>Velocity</th>
<th>Acceleration</th>
<th>Burndown (Sprint, Release)</th>
<th>Committed vs. Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defect Escapes</td>
<td>Defect Containment</td>
<td>Release Frequency</td>
<td>Automated Test Coverage</td>
</tr>
</tbody>
</table>

First pass draft products are expected to be available for community review in August 2019.

We are seeking motivated volunteers to help as authors and reviewers. Interested? contact: Cheryl Jones, Geoff Draper, Larri Rosser
## ICM Table (Draft)

<table>
<thead>
<tr>
<th>Information Categories</th>
<th>Measurable Concept</th>
<th>Team Information Need</th>
<th>Product Information Need</th>
<th>Enterprise Information Need</th>
<th>Potential Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Are story points delivered as committed?</td>
<td>Are features delivered as committed?</td>
<td>Are capabilities delivered as committed?</td>
<td>(story points, features, capabilities)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Are we still on track to deliver all features per roadmap? (on plan)</td>
<td></td>
<td>Burndown</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Committed vs. Completed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cumulative Flow Diagram (WIP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Feature or Capability Implementation by priority</td>
</tr>
<tr>
<td>Schedule and Progress</td>
<td>Work Unit Progress</td>
<td>Did we deliver expected capabilities / features? Is the roadmap still valid?</td>
<td>Is the user satisfied with the delivered products? Do they provide the desired functionality when needed?</td>
<td></td>
<td>Test Progress (# test run and passed)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Feature or Capability Backlog</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Is the integration and test progress proceeding as planned?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>How much outstanding technical or mission debt exists?</td>
<td>Stories produced (team)</td>
<td>Features</td>
<td>Requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SLOC</td>
<td>Features Delivered</td>
<td>Feature Volatility</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Capabilities Delivered</td>
<td>Capability Volatility</td>
<td>Backlog Volatility</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Capabilities Delivered</td>
<td>Capability Volatility</td>
<td>Backlog Volatility</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Backlog Volatility</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>How big is our system?</td>
<td>How big is our system?</td>
<td>How big is our system?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Physical Size and Stability</td>
<td>How volatile are capabilities or features?</td>
<td>How volatile are capabilities or requirements? What is the ability to accommodate changes in customer desiresments?</td>
<td>Features Delivered</td>
<td>Feature Volatility</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Are we adding more features? What is the ability to accommodate changes in customer desiresments?</td>
<td></td>
<td>Capabilities Delivered</td>
<td>Capability Volatility</td>
</tr>
<tr>
<td></td>
<td>Functional Size and Stability</td>
<td>How much of the product is newly developed vs. reused from other sources?</td>
<td>Reuse of capability, features, stories, code</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Implementing Continuous Iterative Development and Acquisition
Backup –
Selected Excerpts from Briefings and Analyses

Contacts for More Information:
Geoff Draper (geoff.draper@l3harris.com)
Larri Rosser (larri.rosser@Raytheon.com)
Cheryl Jones (cheryl.l.jones128.civ@mail.mil)
Garry Roedler (garry.j.roedler@lmco.com)
Robin Yeman (robin.yeman@lmco.com)

Interesting in getting involved more in consensus community solutions for any of these initiatives?
Let us know – help is welcome!
Summary

The NDIA WG provides an industry perspective on picture of success, current state, obstacles and path forward for each DSB recommendation.

<table>
<thead>
<tr>
<th>DSB Recommendation</th>
<th>NDIA “Path Forward” recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 – Software Factory</td>
<td>14 Contracting, funding, incentives, methods, security, supply chain, and measures</td>
</tr>
<tr>
<td>#2 – Continuous Iterative Development</td>
<td>3 Pilots and continuous improvement</td>
</tr>
<tr>
<td>#3 – Risk Reduction &amp; Metrics</td>
<td>10 Acquisition strategy, competitive prototyping, culture, workforce, IP, and measures</td>
</tr>
<tr>
<td>#4 – Legacy Systems</td>
<td>5 Assessments, supply chain, methods, tools, and modeling</td>
</tr>
<tr>
<td>#5 – Workforce Development</td>
<td>3 Competency models, workforce assessment, workforce management, and training</td>
</tr>
<tr>
<td>#6 – Sustainment</td>
<td>2 Contracting and industry-government transfer of sustainment responsibilities</td>
</tr>
<tr>
<td>#7 – Machine Learning</td>
<td>5 Risk, research, CONOPs, ML data, and Software Factory interactions</td>
</tr>
</tbody>
</table>

Details of each topic and recommendation are provided in the separate report.
1. **Software Factory** – A key evaluation criteria in the source selection process should be efficacy of the offeror’s software factory.

2. **Continuous Iterative Development** – DoD and defense industrial base partners should adopt continuous iterative development best practices for software, including through sustainment.

3. **Risk Reduction and Metrics for New Programs** – For all new programs, starting immediately, implement best practices in formal program acquisition strategies (multiple vendors and down-selects, modernized cost and schedule measures, status estimation framework).

4. **Current and Legacy Programs in Development, Production, and Sustainment** – for ongoing development programs, PMs/PEOs should plan transition to a software factory and continuous iterative development.

5. **Workforce** – The U.S. Government does not have modern software development expertise in its program offices or the broader functional acquisition workforce. This requires Congressional engagement and significant investment immediately.

6. **Software is Immortal: Software Sustainment** – RFPs should specify the basic elements of the software framework supporting the software factory... reflected in source selection criteria.

7. **IV&V for Machine Learning** – Machine learning is an increasingly important component of a broad range of defense systems, including autonomous systems, and will further complicate the challenges of software acquisition.

The NDIA working group developed consensus recommendations responding to each of the 7 DSB findings:

- Assumptions
- Picture of Success (End State)
- Current State
- Description
- Obstacles
- Path Forward

This briefing is an executive summary of those recommendations. Detailed report provided separately.
DSB #3b: Measures for CID
NDIA WG Recommendations

Picture of Success (end state)

Consensus frameworks
- Objectives first - measures aligned and tailored from information needs, goals and constraints, at program and enterprise levels

Modernized measures
- Migration toward consensus alternatives to traditional waterfall and phase-based SW measures (LOC, EVM, milestones, ...)
- Derived from SW factory processes, automated by toolchain
- Basis for measuring cost and schedule vs. plan

History-based estimates
- Repositories collect performance-based measures (e.g., WBS, staff, cost, productivity) supporting future comparisons, basis of estimates, proposals, and program monitoring

Measures for CID should be aligned with information needs and constraints, at program and enterprise levels

Recommendations for Path Forward:

- Objectives first - measures aligned and tailored from information needs, goals and constraints, at program and enterprise levels
- Migration toward consensus alternatives to traditional waterfall and phase-based SW measures (LOC, EVM, milestones, ...)
- Derived from SW factory processes, automated by toolchain
- Basis for measuring cost and schedule vs. plan
- Repositories collect performance-based measures (e.g., WBS, staff, cost, productivity) supporting future comparisons, basis of estimates, proposals, and program monitoring

We are here
DSB Recommendation #3 - Metrics
Frameworks for aligning measures with objectives

Summary of DIB Metrics Categories

Deployment Rate
- Initial launch to deployment of simplest useful functionality [MVP]
- Time to field high priority fn (spec>ops) or security hole (find>ops)
- Time from code committed to code in use

Response Rate
- Time req’d for full regression test (automated) and cyber testing
- Time required to restore service after outage [MTTD, MTTR, MTTA]

Code Quality
- Automated test coverage of specs/code
- Number of bugs caught in testing vs. field use [defect detection %]
- Change failure rate (rollback)

Program Management
- Complexity metrics. Devel plan/env metrics (specs, code, staff, ...)

Industry feedback (usefulness, effectiveness)
Available measures instrumented and automated by the toolchain
Selection of program measures tailored by information needs (with a few primary colors required by the enterprise)
Enterprise measures driven by business performance objectives

Success is measured at multiple levels:
- Mission capability
- Program execution
- Enterprise improvement
- Business results, competitiveness

Adoption
Measures, goals, and priorities are tailored based on program objectives and information needs

The NDIA WG recommends a measurement framework that can be adapted to specifics of the program, domain, or acquisition

Measures for continuous iterative development should be aligned with information needs, objectives and constraints, at program and enterprise levels

The NDIA WG recommends a measurement framework that can be adapted to specifics of the program, domain, or acquisition

References:
- Defense Science Board, Design and Acquisition of Software for Defense Systems, Feb 2018
- Defense Innovation Board Metrics for Software Development, version 0.9, 9 Jul 2018
## Information Needs

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deployment Rate</td>
<td>Response Rate</td>
</tr>
<tr>
<td>Sample Count</td>
<td>56</td>
<td>66</td>
</tr>
<tr>
<td>Smooth</td>
<td>0.70</td>
<td>0.60</td>
</tr>
<tr>
<td>Very High</td>
<td>0.57</td>
<td>0.40</td>
</tr>
<tr>
<td>Very High</td>
<td>13</td>
<td>29</td>
</tr>
<tr>
<td>High</td>
<td>16</td>
<td>26</td>
</tr>
<tr>
<td>Medium</td>
<td>26</td>
<td>24</td>
</tr>
<tr>
<td>Low</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>36</td>
<td>60</td>
</tr>
<tr>
<td>Weighted Score</td>
<td>1.93</td>
<td>1.58</td>
</tr>
</tbody>
</table>

### NDIA Continuous Iterative Development and Sustainment WG

10
<table>
<thead>
<tr>
<th>DSB Measures</th>
<th>DIB Measures</th>
<th>Code Quality Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment Rate</td>
<td>Response Rate</td>
<td>Deployment Rate</td>
</tr>
<tr>
<td>Useful</td>
<td>Effective</td>
<td>Useful</td>
</tr>
<tr>
<td>SPI</td>
<td>FTE</td>
<td>SPI</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>2.21</td>
<td>2.35</td>
<td>2.00</td>
</tr>
<tr>
<td>7.10</td>
<td>7.25</td>
<td>7.00</td>
</tr>
</tbody>
</table>

**Measurement Survey Integration and Analysis**

Evaluation of DSB and DIB Measures – Usefulness and Effectiveness: VH=1; H=2; M=3; L=4

**NDIA Continuous Iterative Development and Sustainment WG**
# Evaluation of DSB and DIB Measures – Usefulness and Effectiveness

<table>
<thead>
<tr>
<th>Category:</th>
<th>Response Rate</th>
<th>Deployment Rate</th>
<th>Evaluation and Ranking of DIB Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category Rank</td>
<td>3 4 5 5 4 1 6 2 1 2 1 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall Rank</td>
<td>8 14 11 9 6 1 13 10 2 4 3 12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Usefulness

| Sample Count | 53 54 53 52 54 56 53 53 53 55 55 53 |
| Mean         | 2.30 2.41 2.55 2.30 2.11 1.72 2.73 2.48 1.70 1.83 1.80 2.55 |
| Median       | 2.00 3.00 3.00 2.00 2.00 1.00 3.00 3.00 2.00 2.00 2.00 3.00 |
| Std Deviations | 1.05 0.99 0.87 0.77 0.87 0.88 0.80 0.76 0.74 0.57 0.87 0.87 |
| Variance     | 1.09 0.98 0.76 0.59 0.75 0.78 0.64 0.57 0.55 0.32 0.75 0.76 |

## Effectiveness

| Sample Count | 33 33 34 31 35 31 30 31 33 33 30 30 |
| Mean         | 2.39 2.62 2.94 2.90 2.89 2.61 2.83 2.87 2.45 2.24 2.20 2.93 |
| Median       | 2.00 3.00 3.00 3.00 3.00 2.00 3.00 3.00 2.00 2.00 2.00 3.00 |
| Std Deviations | 0.90 0.92 0.98 0.98 0.83 1.05 0.91 0.88 1.18 0.97 1.00 1.05 |
| Variance     | 0.81 0.84 0.97 0.96 0.69 1.11 0.83 0.78 1.38 0.94 0.99 1.10 |

<table>
<thead>
<tr>
<th>Code Quality Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automated Test Coverage of Test Specs / Code</td>
</tr>
<tr>
<td>Time Required to Restore Service (MTTR)</td>
</tr>
<tr>
<td>Time for Automatic Regression Test (reg test cycle time)</td>
</tr>
<tr>
<td>Time from Code Commit to Use (factory cycle time)</td>
</tr>
<tr>
<td>Time to Fix New Security Hole (patch cycle time)</td>
</tr>
<tr>
<td>Time to Field High Priority Functions (incr lead time)</td>
</tr>
<tr>
<td>Time from Launch to MVP (initial lead time)</td>
</tr>
<tr>
<td>Cumulative Flow</td>
</tr>
<tr>
<td>Velocity (Control Chart)</td>
</tr>
<tr>
<td>Cycle Time</td>
</tr>
</tbody>
</table>

**Least Favorable (N):**

**Most Favorable (1):**
### Metric Summary: Usefulness vs. Effectiveness (1-4)

#### DSB

<table>
<thead>
<tr>
<th>All</th>
<th>Industry</th>
</tr>
</thead>
</table>

#### DIB

<table>
<thead>
<tr>
<th>All</th>
<th>Industry</th>
</tr>
</thead>
</table>
Automated requirements analysis
Why and How To Use COSMIC FP Effectively On Agile Development

Colin Hammond
Creator of ScopeMaster

Lonnie Franks
Project Assurance expert

80 years and over 1000 projects

Software and IT-CAST, August 2019
Goal

Working software that meets the business need
To time and Cost

Start with

Requirements written in English

Need

Reliable estimates

Avoid

Surprises / scope creep
Delays
Technical debt
Requirements are about communication

Requirements - Precision Matters

Requirements are like Blueprints

...that tell you how deep to dig your foundations, the type of windows to order and how much cabling is needed.

English words translated to Code
1 word or requirements : 25 SLOC*

=> Defects are amplified

*ScopeMaster analysis of 25,000 user stories across 70 projects
Get the requirements as good as you can as early as you can

Cognitively intensive
Changing midstream is disruptive

Evolve
Commonly
2%
Per month

Typical
Committed
Completed
De-scoped
Added
Modified
Unchanged

Goal is to reduce these
Example User Story

Requirements in Agile

“User Stories” are the catalyst of the conversation.

Add Delivery Details

As a … Site visitor
I want … Add my delivery address
So that … I can receive my goods

Acceptance/Test Criteria …
I can click pencil to enter my zip code and full home address

Who & what

Why

Given, when & then

I set out to automate the functional sizing of user stories
Discovering functional intent

Automated Functional Sizing

Delete Profile

As a ... Administrator
I want ... Delete a profile......
So that ... I can receive my goods

Deleting a profile

As an administrator I want to delete a profile. Then the system should send a confirmation email to the team.

Functional Steps
- delete profile
- send confirmation email

Interpretation
- Delete profile
- Read confirmation email

Data Movements
- request delete
- delete profile
- return error/confirmation
- confirmation email id
- read confirmation email from storage
- display confirmation email

Who & what → Functions → Data movements
What the analyser does:

1. Reads the user story, analyses with NLP+
2. Detects the functional intent(s)
3. Detects likely users and objects
4. No training required
5. Compares the story with all the other stories
6. Finds problems and suggests fixes (>50%)
7. Proposes functional test cases
8. Produces clear documentation
9. Takes only 2-4 seconds per story
Case Study
160 defects found and fixed in 16 hours
Intelligent Analysis of User Stories

“As Registered user I want to update my profile”

Benefits:

Sizing
- Reliable, valid estimates
- ±20% accuracy
- 100% consistent
- Estimate faster
- More Reliable planning
- Metrics to manage S,V,Q

Quality
- Fewer ambiguities,
- Fewer omissions, duplicates
- Fewer inconsistencies
- Better documentation
- Reduced scope churn & creep
- Less rework & fewer bad fixes
- Less effort to get good quality

CSV: No setup

ScopeMaster

CSV
COSMIC Functional Sizing - the successor to IFPUG

Different from IFPUG
Evolved from inc improvements
Principles not rules
Suitable for all S/W
Works on incomplete / Agile
Open source

https://cosmic-sizing.org

2016 NIST - canonical reference reference for a FP

Human Users

Application being sized

Other interfacing App(s) or devices

Entry

EXit

Read

Write

Persistant storage

\[ \sum E, X, R, W = CFP \]
# Functional Size Metrics on Software Projects

## Sizing software

<table>
<thead>
<tr>
<th>Agile Story Points</th>
<th>SLOC</th>
<th>RICEFW</th>
<th>IFPUG FP</th>
<th>COSMIC Function Points</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Very Flawed</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Not Valid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Inconsistent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Easy to game</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Flawed</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Not Valid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Inconsistent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Easy to game</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Good</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ISO Standard</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Consistent</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- User stories insufficient</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Not ideal for embedded</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Best</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- ISO Standard</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Incomplete OK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Principle-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Automated</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- US. GAO Recommended</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*US. GAO Recommended*
Case study to compare SP vs CFP

**Story points vs actual effort**

R² = 0.33

**CFP vs actual effort**

R² = 0.97

Conclusion: CFP is a better predictor of effort than story points.

Typical Source of Defects on Software Projects

1.000 FP Application
Source: Capers Jones
Applied Software Measurement, third edition
Consistency and CRUD analysis

Using ScopeMaster, you can fix many requirements problems in minutes, sometimes seconds.
Valid Metrics based on CFP

1. **Scope**  CFP estimated, delivered, removed
2. **Velocity**  Rate of delivery of CFP
3. **Cost**  to develop and test CFP
4. **Quality**  Defects delivered per CFP
Agile development contracts…

Promise

Reality

With CFP based contracts
Value of knowing the size
Given a typical Cost of $2,000 per CFP

Vendor negotiation, reasonable price, quality & schedule
$200 10%

Efficient project management (scope, effort, cost, quality)
$100 5%

Avoid de-scoping and reduce rework by using size to manage & ensure quality of each activity early.
$300 15%

$600 Per CFP 30%
## Functional Sizing Automation is Available

### Tools:

<table>
<thead>
<tr>
<th>Based on</th>
<th>Tool</th>
<th>Additional Benefits</th>
<th>Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-requirements</td>
<td>Namcook Analytics LLC</td>
<td>1. Requirements QA</td>
<td></td>
</tr>
<tr>
<td>High level written requirements</td>
<td>ScopeMaster Ltd</td>
<td>1. Requirements QA</td>
<td></td>
</tr>
<tr>
<td>Matlab Simulink Designs</td>
<td>Proprietary - Renault</td>
<td>1. Requirements QA</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Castsoftware</td>
<td>1. Code quality analysis</td>
<td></td>
</tr>
</tbody>
</table>
1. Knowing the functional size is valuable
2. Early functional sizing analysis leads to better quality
3. Functional sizing automation is here
4. COSMIC FSM is ideal for Agile projects & contracts
Estimating SW Costs from Requirements using Objective Function Points Research
From the Congressional Notification: Data Science Tradecraft and Standards Initiative:

- “This initiative, in concert with ODNI’s Augmenting Intelligence Using Machines (AIM) Strategy, will ensure we maintain the IC’s high standards and preserve confidence in the analytic processes as we increase the use of machines and automated methods”

ODNI is collaborating with community on developing more effective/accurate cost/schedule software estimation using automated methods

- Investigating alternatives to estimating Software (SW) via Source Lines Of Code (SLOC) counting, by exploring alternative estimating methods using Function Points (FP) for estimating SW development efforts
  - This drove the need for automated methods to count/capture FPs
  - Similar to the UCC standards in SLOC counting where tool would be Open Source

- This effort focused the IC to initiate an Objective Function Point (OFP) Counting capability into the government managed tool suite (UCC-G) that is requested for each IC MSA program acquisition via CDRL
  - Automatically calculating OFP’s based on International Function Point User Group (IFPUG) documented standards
    - Currently analyzes C, C++, C#, Java and Java Script languages
What are Objective Function Points (OFPs)?

• Developing an Automated Objective Function Point Counter
  – Developing a standard automated approach to counting OFPs to avoid subjective estimates
    » Standard Function Points (FP) require Function Point experts to derive
    » Used UCC tool baseline since it already parses through most SW languages and is Open Source
  – Current Function Point estimates use IFPUG tables such as:
    1. External Input (EI): Functions that move data into the application without presenting data manipulation.
    2. External Output (EO): Functions that move data to user and presents some data manipulation.
    3. External Inquiries (EQ): Functions that move data to user without presenting data manipulation.
    4. Internal Logical Files (ILF): The logic in the form of fixed data managed by the application using External Input (EI)
    5. External Interface Files (EIF): The logic in the form of fixed data used by the application but did not run in it

  – OFPs capture the sizing needed to assess the effort and are based from the IFPUG weights
    » Cyclomatic Complexity determines which IFPUG table to use
To test the OFPs, we used NASA’s General Missions Analysis Tool (GMAT) software
  – Open Source code

To calculate the OFPs from GMAT code, we can pull out the following data:
  – The Cyclomatic Complexity shows that most of the GMAT code falls less than 10
  – This will drive the OFPs to be the lowest values from the IFPUG tables

Things to consider on OFPs:
  – OFPs are derived from the actual source code
  – Every Function gets an OFP associated to it
  – There are more developed Functions in the code than what the Function Point experts can predict
  – This will lead to higher OFPs than FPs
    » Assuming FPs were generated from requirements by an FP expert

<table>
<thead>
<tr>
<th>McCabe Complexity Values</th>
<th>McCabe Complexity Definitions</th>
<th>Traditional Function Point Mapping Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 – 10</td>
<td>Structured and well written code, High Testability, Cost and Effort is Less</td>
<td>External Input (Ei): Functions that move data into the application without presenting data manipulation.</td>
</tr>
<tr>
<td>11 – 20</td>
<td>Complex Code, Medium Testability, Cost and Effort is Medium</td>
<td>External Output (Eo): Functions that move data to user and presents some data manipulation. External Inquiries (Eq): Functions that move data to user without presenting data manipulation.</td>
</tr>
<tr>
<td>21 – 40</td>
<td>Very Complex Code, Low Testability, Cost and Effort is Medium</td>
<td>External Interface Files (Eif): The logic in the form of fixed data used by the application but did not run in it</td>
</tr>
<tr>
<td>&gt; 40</td>
<td>Difficult to test, Very High Cost and Effort</td>
<td>Internal Logical Files (Ilf): The logic in the form of fixed data managed by the application using External Input (Ei)</td>
</tr>
</tbody>
</table>
OFPs and Requirements

• 5 Different small test cases were performed with a FP expert
  – FP Expert was provided in all cases UML Sequence and Class Model diagrams
  – All 5 cases had less than a 10% error between the automated OFPs and the predicted FPs

• This led to a large test case using a subset of the GMAT Formal SW Requirements document
  – The GMAT Formal SW Requirements document is composed of:
    » Application Control, Resource and Command Objects Functional Requirements
    » External Interface, Environmental, Computer Resource and Test System Requirements
  – **INITIAL TEST**: Using a subset of the GMAT requirements document (see backup slide for details), the FP expert calculated FPs that showed ~ 200% error from the OFPs
  – **FINAL TEST**: After providing the FP expert the Unified Modeling Language (UML) documentation, the FP expert calculated FPs that showed ~ 2% error from the OFPs

• This shows the Uncertainty of estimates are due to the Maturity of the Requirements provided to the estimator
  – Phase A requirements are being refined/defined/matured while some are not specified until the end of Phase B
  – As the requirements get more mature, the Uncertainty of the estimate will go down
Using OFPs to Capture Effort

• OFPs capture the total effort of a baseline as though it was ALL NEW code

• How do we use these if we are trying to capture the effort between baselines or in AGILE’s case “Sprints” or “Increments”?
  – We needed a new metric that can utilize the UCC DIFF capabilities

• Created a measure to capture development called **Effective Objective Function Points (EOFPs)**
  – EOFPs are computed by comparing the source code of two baselines
  – After the code is divided into modules / classes or each code baseline, the code is compared and the number of ADDED, MODIFIED and DELETED Logical Source Lines Of Code (LSLOC) is determined for each function within each module
  – For each function, the number of ADDED + MODIFIED lines (as a percentage of total lines) is multiplied by the OFP to determine the EOFP for that function.
  – The EOFP for the module is simply the sum of the EOFP for all functions
NASA’s General Missions Analysis Tool (GMAT) Example

- Results from comparison between GMAT C++ code baselines 2017a and 2018a

<table>
<thead>
<tr>
<th></th>
<th>2017a</th>
<th>2018a</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Source Lines Of Code (PSLOC)</td>
<td>457,112</td>
<td>535,661</td>
</tr>
<tr>
<td>Logical Source Lines Of Code (LSLOC)</td>
<td>290,674</td>
<td>318,173</td>
</tr>
<tr>
<td>Delta LSLOC</td>
<td>27,499</td>
<td></td>
</tr>
<tr>
<td>Modules (Classes)</td>
<td>3,104</td>
<td>3,191</td>
</tr>
<tr>
<td>Delta Modules</td>
<td>3,146</td>
<td></td>
</tr>
<tr>
<td>EOFPS</td>
<td>70,953</td>
<td>74,099</td>
</tr>
<tr>
<td>Delta EOFPs</td>
<td>3,146</td>
<td></td>
</tr>
</tbody>
</table>

**DIFF Results**

- NEW LSLOC: 9.3%
- DELETED LSLOC: 1.0%
- MODIFIED LSLOC: 0.7%
- UNMODIFIED LSLOC: 89.0%
- EOFPS: 3,023.3

- Other potential metrics:
  - EOFPs / Hours
  - LSLOC / Function
    » Previous UCC counter reported by file and NOT by Function
  - Objective Maintainability Index
    » GMAT = 135.67 **High Maintainability**

- Objective Maintainability Index
  
<table>
<thead>
<tr>
<th>Maintainability Index</th>
<th>Good Maintainability</th>
<th>Moderate Maintainability</th>
<th>Difficult to Maintain</th>
</tr>
</thead>
<tbody>
<tr>
<td>85 and more</td>
<td>Good maintainability</td>
<td>Moderate maintainability</td>
<td>Difficult to maintain</td>
</tr>
<tr>
<td>65-85</td>
<td>Moderate maintainability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>under 65</td>
<td>Difficult to maintain</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

  Bad pieces of code (big, uncommented, unstructured) the MI value can be even negative

- Here are some high level GMAT metrics:
  - Total # EOFPs = 3,023.3
  - # Changed/New Modules = 211
  - # Changed/New Functions = 1,473
  - EOFPs / (Changed/New Modules) = 14.33
  - EOFPs / (Changed/New Functions) = **2.05**
  - (Changed/New Functions)/(Changed/New Modules) = **6.98**

We are opening the door to new metrics to estimate effort
Potential of Using Requirements Documents

• Software Requirement Documents contain nouns and verbs
  – Object Oriented Theory:
    » Nouns become Modules/Classes
    » Verbs become Process Functions

• Using Natural Language Toolkit (NLTK) to automatically extract nouns and verbs
  » This is free open source on the unclass and class side

• NLTK parses out the nouns and verbs from the requirements very well

• In order to identify key words that correlate to effort, we need to calculate weights for the nouns and verbs
  – These weights would be derived by scoring them against the rest of the requirement document

• Due to long runtime with NLTK algorithms when scoring, we investigated using Lucene in place of NLTK to compute Scoring between Module / Class names and individual requirements
  – Lucene combines Boolean model (BM) of Information Retrieval with Vector Space Model (VSM) of Information Retrieval - documents "approved" by BM are scored by VSM
  – Lucene is open source available on high and low side
  – Calibrated Lucene Scoring model to properly map Key Words to Requirements
    » Calibration involved many hours of many different test cases and individually comparing results

• Runtime of Lucene Scoring outperformed NLTK Scoring
What can we do with Scoring?

• How can we use the Scoring of Nouns and Verbs to help estimate the number of Functions?
  – Following the basic principle from Object Oriented Analysis:
    » Nouns are potential Modules (classes)

• By using the Scores from the Unique Nouns, we could estimate the number of Modules (Classes)
  – The # of Unique Nouns equals the # of Modules (Classes) when:
    » Score of 3 or more (see plot of right)
    » As previously noted, there are 211 GMAT Module (Classes)

• # Functions currently do not trend to Unique Verbs
  – Verbs such as “Get or “Set” are used many times
  – Requires more research

• As previously noted, there are **6.98** Functions per Module (Class) in GMAT
  – Currently observing other programs in this range (6 – 7)
  – Multiplying the 211 Total Changed and New Modules by 6.98, we get **1473** GMAT Functions

• Next challenge is to convert to EOFPs
  – As previously noted, there are **2.05** EOFPs/ Total Changed and New Functions in GMAT
    » Currently observing other programs in this range (2 – 3)
  – **Total Estimated EOFPs = 2.05 * 1473 = 3,019.7**
  – **Total UCC EOFPs = 3,023.3**
• This Monte Carlo approach provides a Risk Range for the estimate
  – If Story Points are available from data collection efforts, the Story Points map to each requirement where each EOFP maps to Code
    » Story Points are a subjective means in Agile to relate size to effort
  – If Story Points are NOT available, subjectively map each requirement to the Code Functionality

• Challenge has always been correlating requirements to Function Points

• Scoring paves the way for Monte Carlo approach to correlate requirements to Function Points
  • This allows Automapping between Code to Requirements
    – Each requirement can now map to a section of Code as well as EOFPs
    – Thus producing ranges for each requirement
  • This bounds the estimate with ranges from the Monte Carlo simulation runs
**Summary of Sizing Estimating Process**

Use NLTK to parse nouns and verbs

Use Lucene to score each noun, verb

Use Hypothesis to estimate # Functions

Use Relationships to estimate # EOFPs

Read in Reqts

GMAT FEATURE: The Ground Station object supports a new troposphere model, the Marini model, matching the implementation used in GTDS. One operational advantage of the Marini model is that it doesn’t require input of weather data at the Ground station. (Models that do accept weather data may have more accuracy.)

Y = Xb + c

Y = Xb + c

GMAT FEATURE: The Ground Station object supports a new troposphere model, the Marini model, matching the implementation used in GTDS. One operational advantage of the Marini model is that it doesn’t require input of weather data at the Ground station. (Models that do accept weather data may have more accuracy.)

GMAT FEATURE: The Ground Station object supports a new troposphere model, the Marini model, matching the implementation used in GTDS. One operational advantage of the Marini model is that it doesn’t require input of weather data at the Ground station. (Models that do accept weather data may have more accuracy.)

GMAT FEATURE: The Ground Station object supports a new troposphere model, the Marini model, matching the implementation used in GTDS. One operational advantage of the Marini model is that it doesn’t require input of weather data at the Ground station. (Models that do accept weather data may have more accuracy.)
• Accomplishments:
  1. Need an Automated and Objective method to capture sizing
     Objective Function Point (OFP) tool provides this solution ✓
  2. Need an Automated method to pull nouns and verbs from Requirement documents
     NLTK tool provides this solution ✓
  3. Need to Correlate code to Requirement descriptions
     Lucene Scoring tool provides a quick solution for each word combination ✓
  4. Need to isolate specific development to specific SW Requirements
     OFP DIFF tool isolates the development that relates to stories/specific Requirements ✓
  5. Need to capture new Sizing metric to relate specific development to specific Requirements
     OFP DIFF tool now reports Effective OFPs (EOFPs) ✓
  6. Need to Map Code to Requirements
     Built a Mapping tool to read in Classes and Methods/Functions and map them to Requirements ✓
  7. Need to estimate EOFPs based from specific Requirements
     Investigating various hypotheses that will provide confidence in estimating EOFPs ✓
  8. Need to convert EOFPs to hours
  9. Need to collect more program baselines
POCs

• Govt POC: Michal Bohn  MICHALB6@dni.gov
• Presenter: Paul Cymerman  pcymerman@quaternion-consulting.com
## GMAT Requirements for Unit Level Testing

<table>
<thead>
<tr>
<th>Requirement ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRR-42.1.0</td>
<td>The system shall allow the user to choose among the following objects as the central body of a ground track plot:</td>
</tr>
<tr>
<td>FRR-42.1.1</td>
<td>1) Default Celestial Body</td>
</tr>
<tr>
<td>FRR-42.1.2</td>
<td>2) User-defined Body</td>
</tr>
<tr>
<td>FRR-42.2.0</td>
<td>The Ground Track Plot shall draw the longitude and latitude time-history for the following object types:</td>
</tr>
<tr>
<td>FRR-42.2.1</td>
<td>1) Spacecraft</td>
</tr>
<tr>
<td>FRR-42.2.3</td>
<td>2) Groundstation</td>
</tr>
<tr>
<td>FRR-42.3.0</td>
<td>The system shall display icons on the ground track to indicate the locations of the following object types:</td>
</tr>
<tr>
<td>FRR-42.3.1</td>
<td>1) Spacecraft</td>
</tr>
<tr>
<td>FRR-42.3.2</td>
<td>2) Groundstation</td>
</tr>
<tr>
<td>FRR-42.4.0</td>
<td>The system shall display object labels next to the icons for the following object types:</td>
</tr>
<tr>
<td>FRR-42.4.1</td>
<td>1) Spacecraft</td>
</tr>
<tr>
<td>FRR-42.4.2</td>
<td>2) Groundstation</td>
</tr>
<tr>
<td>FRR-42.5.0</td>
<td>The system shall allow the user to define the data plotting options for a ground track plot:</td>
</tr>
<tr>
<td>FRR-42.5.1</td>
<td>1) The number of integration steps to skip between plot points</td>
</tr>
<tr>
<td>FRR-42.5.2</td>
<td>2) The number of plot points to collect before updating a ground track plot</td>
</tr>
<tr>
<td>FRR-42.5.3</td>
<td>3) The number of plot points to retain and redraw during propagation and animation,</td>
</tr>
<tr>
<td>FRR-42.6.0</td>
<td>The system shall allow the user to specify how data is drawn to Ground Track Plots during iterative processes such as differential correction, optimization, and estimation. The following options shall be available:</td>
</tr>
<tr>
<td>FRR-42.6.1</td>
<td>1) Show all iterations/perturbations</td>
</tr>
<tr>
<td>FRR-42.6.2</td>
<td>2) Show current iteration/perturbation only</td>
</tr>
<tr>
<td>FRR-42.6.3</td>
<td>3) Show solution only</td>
</tr>
<tr>
<td>FRR-42.7.0</td>
<td>The system shall allow the user to specify a texture map using the following options</td>
</tr>
<tr>
<td>FRR-42.7.1</td>
<td>1) Use default texture map for central body</td>
</tr>
<tr>
<td>FRR-42.7.2</td>
<td>2) Use user-defined texture map,</td>
</tr>
<tr>
<td>FRR-42.8</td>
<td>The system shall optionally display or not display a configured ground track plot</td>
</tr>
<tr>
<td>FRR-42.9</td>
<td>The Ground Track Plot shall display the epoch in UTC Gregorian format</td>
</tr>
<tr>
<td>FRR-42.10</td>
<td>The system shall allow the user to animate the Ground Track Plot after a run is complete</td>
</tr>
<tr>
<td>FRR-42.11</td>
<td>The system shall display the latitude and longitude values when the cursor is placed over a GroundTrackPlot.</td>
</tr>
</tbody>
</table>
Objective Function Point Counting Process on Existing Source Code

• Parse source code for relevant metrics:
  – Modules (classes, or file names for non-OO code and code outside of classes)
    » Class inheritance
    » Associations between modules
  – Methods / Functions
    » Cyclomatic Complexity
  – Attributes (class level variables)

• Cyclomatic Complexity for each method / function is used to determine a which OFP table to use
  (EI, EO/EQ, ELF, ILF)
  – Complexity < 11 = EI table
  – Complexity < 21 = EO/EQ table
  – Complexity < 41 = ELF table
  – Complexity >= 41 = ILF table

• Use metrics gathered in step 1 to determine which row to choose in the OFP table
  – Class Inheritances (+1) corresponds to OFP RET (Record Element Type)
  – Class Associations correspond to OFP FTR (File Type Reference)
  – Class Attributes correspond to OFP DET (Data Element Type)
  – The number of inheritances, associations, and attributes for a module tells us which row to select in the
    OFP tables.
    » The average of the RET/DET and FTR/DET ratios gives us a low, average, or high risk, corresponding to
      the three rows in each OFP table, so we simply use that knowledge to pick the FP number from our selected
      table

• Now we have an Objective Function Point number for each method/function. Total them all
together and we have the OFP for the source code
Determine Function Point Complexity

- To account for the interfaces in the design of the code, Function Point Theory captures these interfaces through 5 different pieces of data:
  1. External Input (EI): Functions that move data into the application without presenting data manipulation.
  2. External Output (EO): Functions that move data to user and presents some data manipulation.
  3. External Inquiries (EQ): Functions that move data to user without presenting data manipulation.
  4. Internal Logical Files (ILF): The logic in the form of fixed data managed by the application using External Input (EI)
  5. External Interface Files (EIF): The logic in the form of fixed data used by the application but did not run in it

- Based on the 5 above types and the calculated RET, FTR and DET, the Complexity value can be attained by using the look-up tables on the right
  » These are standard IFPUG tables
Cyclomatic Complexity Approach

- UCC already collects Cyclomatic Complexity (CC)
- Objective Function Point (OFP) uses CC as a proxy for the complexity-related inputs to Function Point calculations

<table>
<thead>
<tr>
<th>McCabe Complexity Values</th>
<th>McCabe Complexity Definitions</th>
<th>Traditional Function Point Mapping Definitions</th>
</tr>
</thead>
</table>
| 1 – 10                   | • Structured and well written code  
                           • High Testability  
                           • Cost and Effort is Less  | • External Input (EI): Functions that move data into the application without presenting data manipulation. |
| 11 – 20                  | • Complex Code  
                           • Medium Testability  
                           • Cost and Effort is Medium  | • External Output (EO): Functions that move data to user and presents some data manipulation.  
                           • External Inquiries (EQ): Functions that move data to user without presenting data manipulation. |
| 21 – 40                  | • Very Complex Code  
                           • Low Testability  
                           • Cost and Effort is Medium  | • External Interface Files (EIF): The logic in the form of fixed data used by the application but did not run in it |
| > 40                     | • Difficult to test  
                           • Very High Cost and Effort  | • Internal Logical Files (ILF): The logic in the form of fixed data managed by the application using External Input (EI) |
Simple Example

- Run counter on a small program that has only 2 Classes resulting in 2 files
- File 1 has 2 Methods/Functions
- File 2 has 1 Method/Function
- Below are results from the Function Point tool:

<table>
<thead>
<tr>
<th></th>
<th>RET</th>
<th>FTR</th>
<th>DET</th>
<th>CC</th>
<th>OFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE/Method1.1</td>
<td>2</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>FILE/Method1.2</td>
<td>3</td>
<td>2</td>
<td>20</td>
<td>20</td>
<td>7</td>
</tr>
<tr>
<td>FILE/Method2.1</td>
<td>5</td>
<td>3</td>
<td>35</td>
<td>45</td>
<td>10</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
</tbody>
</table>

- Where:
  - RET: Record Element Type
  - FTR: File Type Reference
  - DET: Data Element Type
  - CC: Cyclomatic Complexity
  - OFP: Objective Function Points

<table>
<thead>
<tr>
<th>McCabe Complexity Values</th>
<th>McCabe Complexity Definitions</th>
<th>Traditional Function Point Mapping Definitions</th>
</tr>
</thead>
</table>
| 1 – 10                    | • Structured and well written code  
                            | • High Testability  
                            | • External Input (EI): Functions that move data into the application without presenting data manipulation. |
| 11 – 20                   | • Complex Code  
                            | • Medium Testability  
                            | • Cost and Effort is Medium  
                            | • External Output (EO): Functions that move data to user and presents some data manipulation.  
                            | • External Inquiries (EIQ): Functions that move data to user without presenting data manipulation. |
| 21 – 40                   | • Very Complex Code  
                            | • Low Testability  
                            | • Cost and Effort is Medium  
                            | • External Interface Files (EIF): The logic in the form of fixed data used by the application but did not run in it |
| > 40                      | • Difficult to test  
                            | • Very High Cost and Effort  
                            | • Internal Logical Files (ILF): The logic in the form of fixed data managed by the application using External input (EI) |
Joint Agile Software Innovation (JASI) Cost IPT

Join the Band!

Kammy Mann
JASI Cost IPT Secretary
Cost Analysis Division (CAD)
DHS Office of the Chief Financial Officer
Agenda

Origins and Purpose of the JASI Cost IPT
What We Do
Introduction to Simplified Function Points
Who is part of JASI?
How can you join the team?
Origins of JASI

After last year’s IT CAST, representatives from DHS/NGA/NSA discussed forming an IPT to share best practices and lessons learned.

The first kick-off meeting for JASI was held on October 3, 2018.

**Goal:** Improve the US Government Cost Community’s ability to estimate the cost of software development in an Agile environment and track progress to successful completion through collaboration and exchange of terminology, definition, data, techniques, and methods.
After last year’s IT CAST, representatives from DHS/NGA/NSA discussed forming an IPT to share best practices and lessons learned.

The first kick-off meeting for JASI was held on October 3, 2018.

**Goal**: Improve the US Government Cost Community’s ability to estimate the cost of software development in an Agile environment and track progress to successful completion through collaboration and exchange of terminology, definition, data, techniques, and methods.
After last year’s IT CAST, representatives from DHS/NGA/NSA discussed forming an IPT to share best practices and lessons learned.

The first kick-off meeting for JASI was held on October 3, 2018.

**Goal:** Improve the US Government Cost Community’s ability to estimate the cost of software development in an Agile environment and track progress to successful completion through collaboration and exchange of terminology, definition, data, techniques, and methods.
Develop a pragmatic/defendable approach to estimate and measure the cost of software development – Simplified Function Point Analysis (SFPA)

- Deliver SFPA training to the Cost Community as requested
- Standardize and expand the lexicon of verbs for the Pattern Matching Technique used in SFPA

Improve data availability to enhance the credibility of cost estimates and measurement

- Identify data and analysis for inter-agency sharing and use
- Explore machine learning/natural language processing for mining data and sizing software programs

Investigate new approaches to tracking, measuring, and reporting progress of an Agile Program through its development lifecycle
What We Do

Develop a pragmatic/defendable approach to estimate and measure the cost of software development – Simplified Function Point Analysis (SFPA)
  – Deliver SFPA training to the Cost Community as requested
  – Standardize and expand the lexicon of verbs for the Pattern Matching Technique used in SFPA

Improve data availability to enhance the credibility of cost estimates and measurement
  – Identify data and analysis for inter-agency sharing and use
  – Explore machine learning/natural language processing for mining data and sizing software programs

Investigate new approaches to tracking, measuring, and reporting progress of an Agile Program through its development lifecycle
What We Do

Develop a pragmatic/defendable approach to estimate and measure the cost of software development – Simplified Function Point Analysis (SFPA)
  – Deliver SFPA training to the Cost Community as requested
  – Standardize and expand the lexicon of verbs for the Pattern Matching Technique used in SFPA

Improve data availability to enhance the credibility of cost estimates and measurement
  – Identify data and analysis for inter-agency sharing and use
  – Explore machine learning/natural language processing for mining data and sizing software programs

Investigate new approaches to tracking, measuring, and reporting progress of an Agile Program through its development lifecycle
IFPUG* Function Point Analysis

*International Function Point Users Group - www.ifpug.org
Simplified Function Points

Simple Function Points* count two components:

**Elementary Process:** the smallest level of activity that is meaningful to the user (EI, EO, EQ)

**Logical Data Groups:** a user identified group of data or control information maintained by an application (ILF, EIF)

<table>
<thead>
<tr>
<th>IFPUG Components</th>
<th>Low</th>
<th>Average</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Inputs</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>External Outputs</td>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>External Inquiries</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Internal Logical Files</td>
<td>7</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>External Interface Files</td>
<td>5</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SFPA Components</th>
<th>Weighting Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elementary Processes</td>
<td>4.6</td>
</tr>
<tr>
<td>(Create, Update, Delete,</td>
<td></td>
</tr>
<tr>
<td>Report, Read)</td>
<td></td>
</tr>
<tr>
<td>Logical Data Groups</td>
<td>7</td>
</tr>
<tr>
<td>(Saves)</td>
<td></td>
</tr>
</tbody>
</table>

## Simple Function Point Sizing

Size = \sum (4.6 \times \text{Transactions}) + (7.0 \times \text{Save})
Who is part of JASI?

And Growing!
How Can You Join?

We are always looking for new members to join JASI CIPT!

Requirements:

– Government-only, or Support Contractor with Client permission & part of current scope of work
– Involved in Software Cost Estimating
– Willing to attend meetings quarterly and be active members

Our next meeting will be in September/early October!

Email JASI Secretary: Katharine.Mann@hq.dhs.gov