ITCAST

SYMPOSIUM 2019

PROGRAM

SPONSORED BY THE NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY

TABLE OF CONTENTS

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

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 apace 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

AGENDA

TUESDAY, AUGUST 20, 2019

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 agenc'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

24 h

AGENDA

WEDNESDAY, AUGUST 21, 2019

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

KEYNOTE BIOGRAPHIES

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.

KEY PRESENTER BIOGRAPHIES

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.

BIOGRAPHIES

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® III (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.

Ms. Jones is the technical lead and project manager of Practical Software and Systems Measurement and a primary author of Practical Software Measurement: Objective Information for Decision Makers. Ms. Jones is the DoD representative to the US Technical Advisory Group to International Standards Organization SC7, System and Software Engineering.

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 heath, 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 Symposiums. 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

U.S. Air Force Captain John Rosson joined the Air Force Cost Analysis Agency (AFCAA) in April 2017, after completing his Masters of Cost Analysis degree at the Air Force Institute of Technology. During his time at AFCAA, he has worked on Joint Strike Fighter, Light Attack Aircraft, and the B-21 bomber.

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.

NGA
NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY

NGA.mil

Approved for public release, 19-949

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

Billing Overview

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

1	Service	Unit of Measure	Included Quantity	Part Number	Unit Price
2	Academic Knowledge - Standard - Transactions - Preview	10000	0	AAA-17686	$2.60
3	Advanced Data Security for SQL Database - Standard Node	1 /Month	0	AAA-54514	$15.60
4	Advanced Data Security for SQL Database - Standard Node - US DoD	1 /Month	0	AAD-46271	$21.84
5	Advanced Data Security for SQL Database - Standard Node - US Gov	1 /Month	0	AAD-46269	$19.50
6	Advanced Data Security for SQL Database - Standard Trial Node	1 /Month	0	AAA-54515	$0.00
7	Advanced Data Security for SQL Database - Standard Trial Node - US DoD	1 /Month	0	AAD-46272	$0.00
8	Advanced Data Security for SQL Database - Standard Trial Node - US Gov	1 /Month	0	AAD-46270	$0.00
9	Advanced Threat Protection - Azure Database for MySQL - Standard Nodes - Preview	1 /Month	0	AAD-31471	$7.80
10	Advanced Threat Protection - Azure Database for MySQL - Standard Trial Nodes - Preview	1 /Month	0	AAD-31472	$0.00
11	Advanced Threat Protection - Azure Database for PostgreSQL - Standard Nodes - Preview	1 /Month	0	AAD-31470	$7.80
41370	Windows Server - 6 vCPU VM License	10 Hours	0	AAA-70516	$2.87
41379	Windows Server Burst - 16 vCPU VM License	100 Hours	0	AAA-70572	$6.66
41380	Windows Server Burst - 2 vCPU VM License	100 Hours	0	AAA-70526	$0.83
41381	Windows Server Burst - 20 vCPU VM License	100 Hours	0	AAA-70571	$8.32
41382	Windows Server Burst - 4 vCPU VM License	100 Hours	0	AAA-70510	$1.66
41383	Windows Server Burst - 8 vCPU VM License	10 Hours	0	AAA-70527	$0.33
41384	Workload Protection for App Services - Standard - Nodes	100 /Hour	0	AAA-97326	$2.08
41385	Workload Protection for App Services - Standard - Nodes - US Gov	100 /Hour	0	AAA-97328	$2.60
41386	Workload Protection for App Services - Standard Trial - Nodes	100 /Hour	0	AAA-97327	$0.00
41387	Workload Protection for App Services - Standard Trial - Nodes - US Gov	100 /Hour	0	AAA-97329	$0.00
41388	Xamarin University - Online Training - Users - Promo	1	0	AAA-35100	$57.20

Billing Example:

Legacy Server vs Cloud Server

Legacy Server	Cloud Server
One Time Purchase w/ Long Lead Time	Instant Access w/ Constant Tuning
Manuel Time Consuming Setup	Automated Setup
Over "Engineered" w/ Many Upgrades	Fit for Purpose – Start Small
Annual Maintenance Agreement	Pay for Use- Turn Off w/ No Use
Limited Scalability	Auto Scale Up or Down
Less "Pieces" to Manage	Many "Pieces" to a Cloud Server
Pay Upfront w/ Less Long Term Monitoring	Constant Monitoring of Bill - Clean Up is Critical

Training

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

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

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**

Cost Saving Example #2

Network Flow Migration

- Legacy Network Hardware Solution:
  - **End of Life** Coming Up
  - **Unable to Handle** Network Flows Capacity
  - Unable to **Monitor/Integrate** w/ 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**

Technology

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

Cloud References

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**
6. **Doppler Report:** **https://www.cloudtp.com/doppler/**
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**

NATIONAL RECONNAISSANCE OFFICE

Forecasting Future Amazon Web Services Pricing

James Smirnoff, Hassan Souiri

NRO Cost and Acquisition Assessment Group (CAAG)

IT CAST

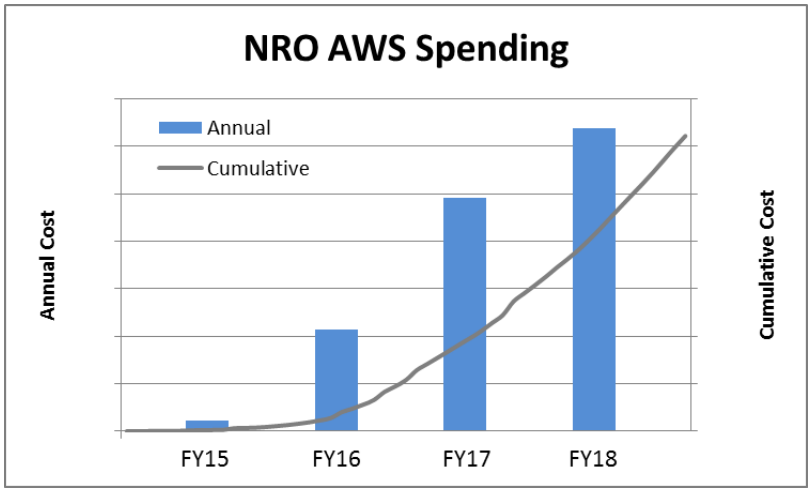
August 2019

SUPRA ET ULTRA

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

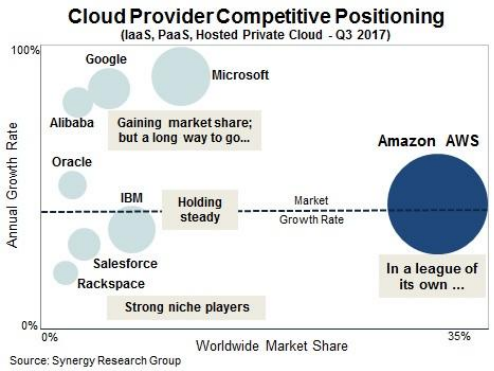


AWS Pricing Study

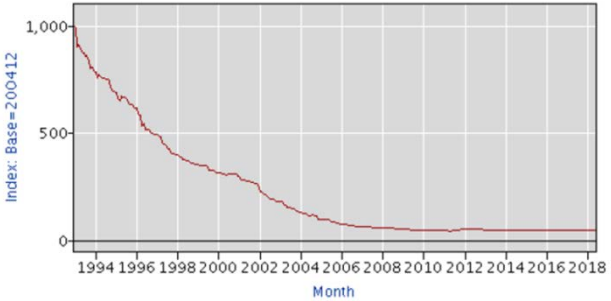
SUPRA ET ULTRA

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



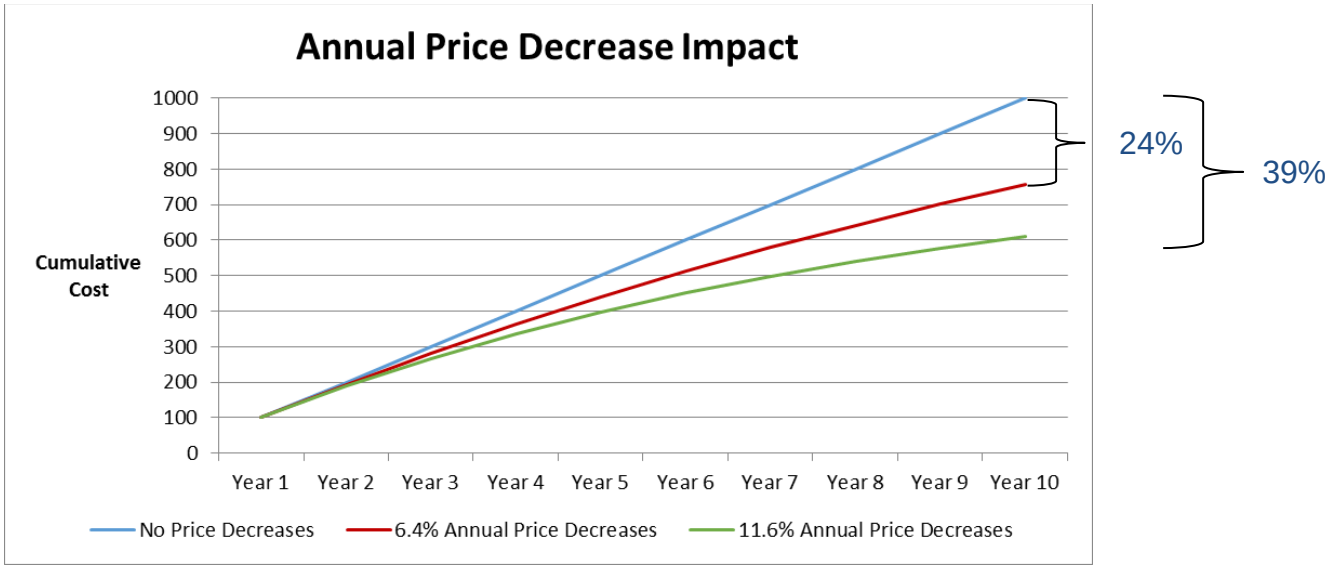
Bureau of Labor Statistics (BLS) Producer Price Index (PPI) – Computer Storage Devices



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



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.

Autoregressive (AR(p)) Time Series Model

- 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

$$z_t = c + \sum_{i=1}^{p} \varphi_i z_{t-i} + \varepsilon_t$$

z_t = price at time t

c = constant, set equal to zero for our models

p = order of model; number of previous months considered

φ = parameters or weights of the model

ε = 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

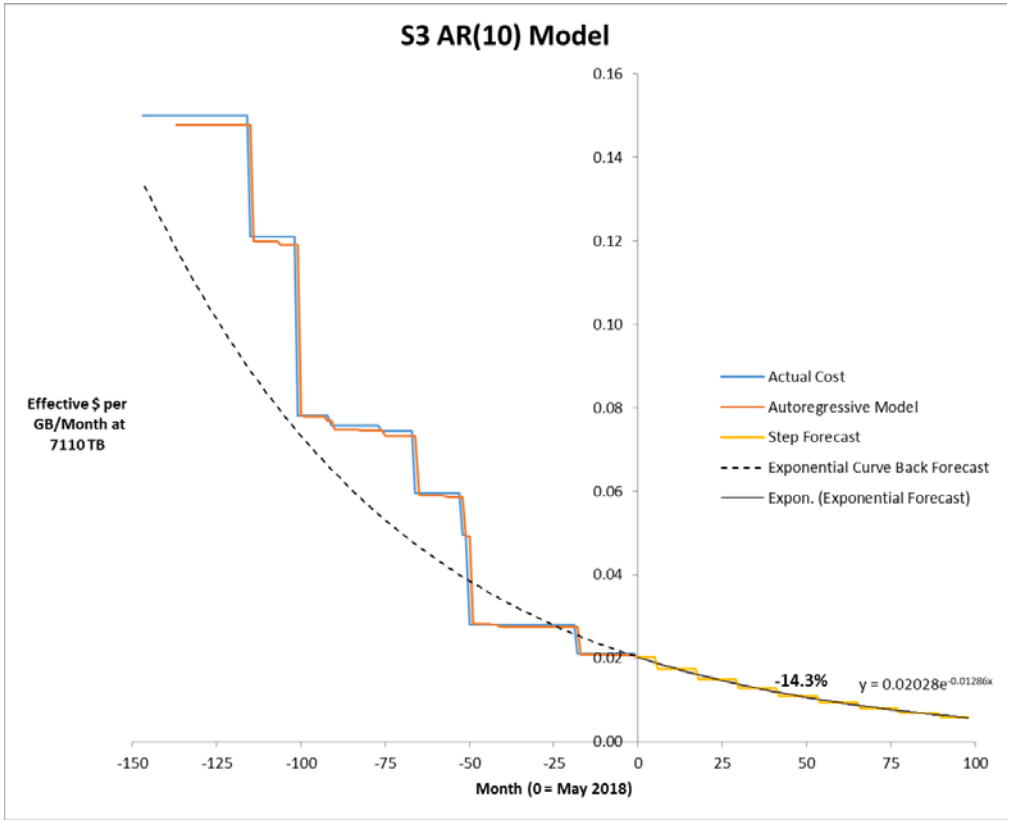
Months Between Price Reductions	32		14		10		15		10		14		2		32
Months Between Consecutive Price Reductions		46		24		25		25		24		16		34	

- 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.

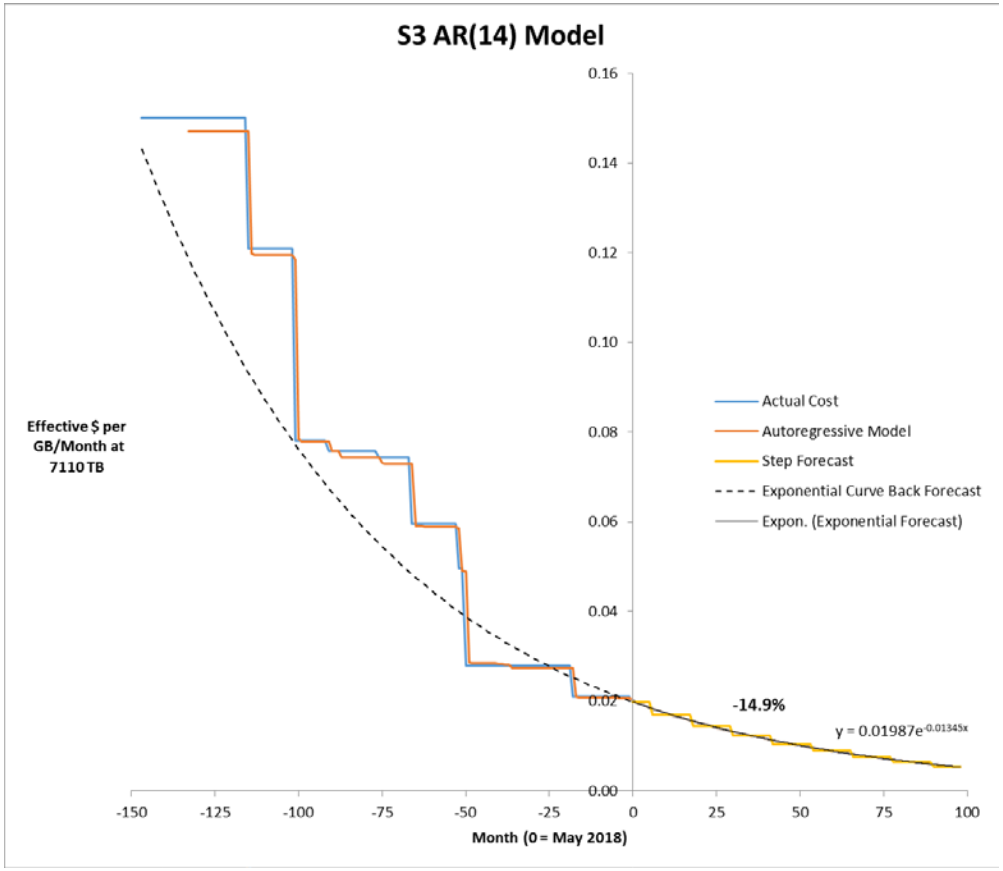
Commercial AWS Storage (S3) AR(10) Model

- 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



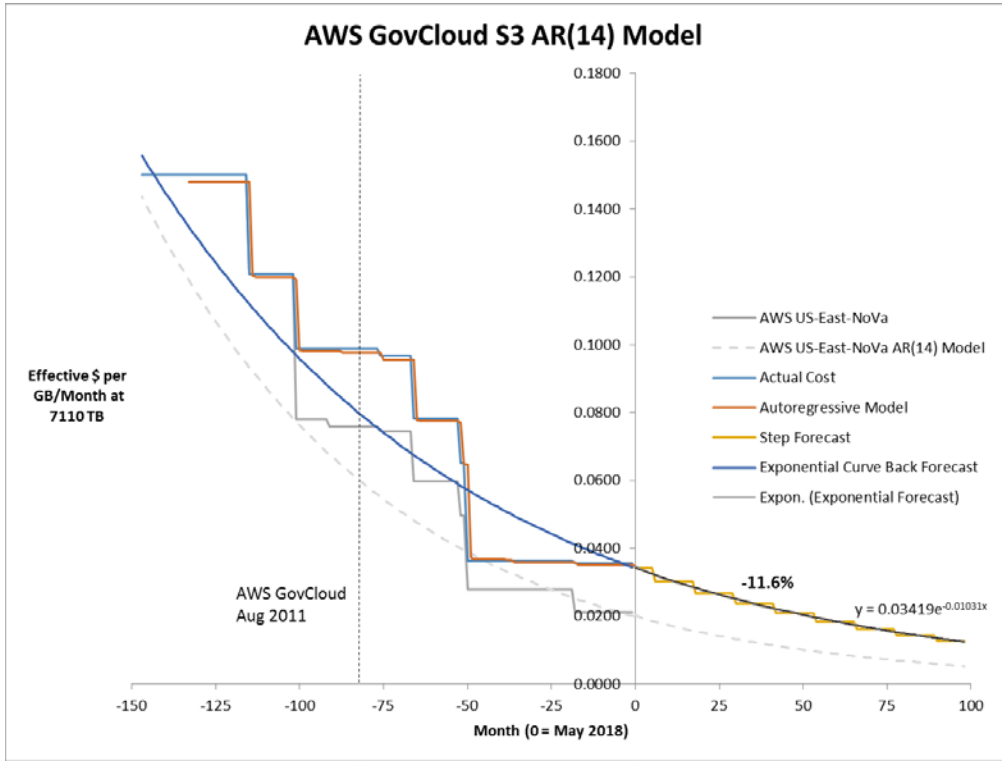
Commercial AWS Storage (S3) AR(14) Model

- 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

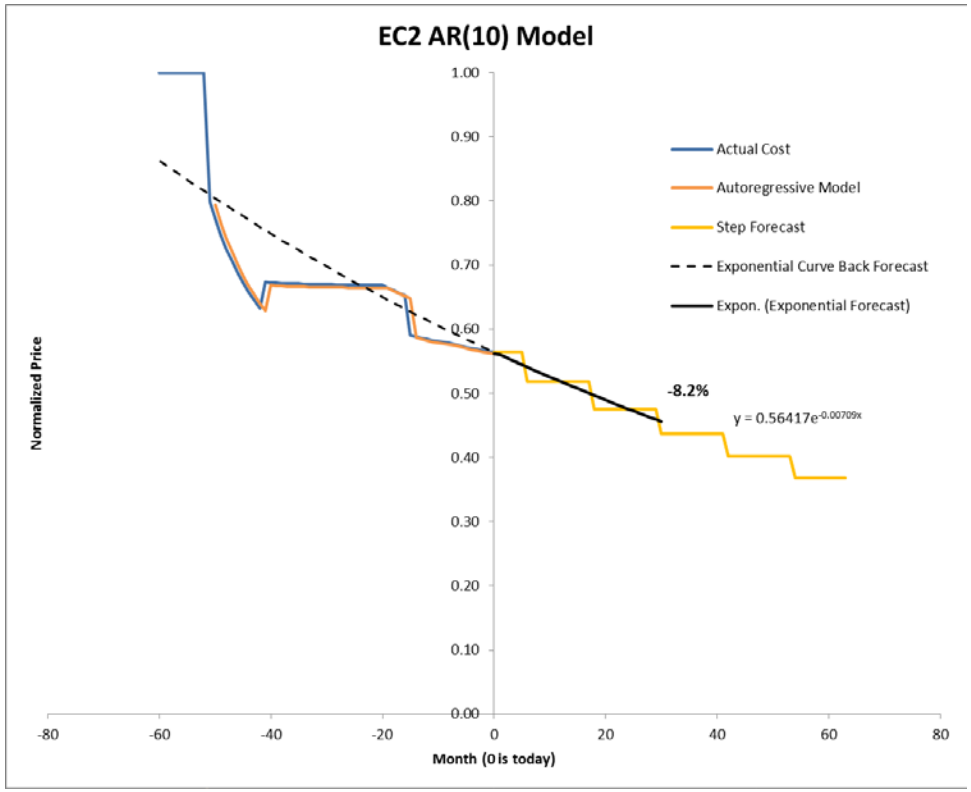
- Commercial AWS faces more pressure from competition to reduce prices and benefits from larger economies of scale than AWS GovCloud
- GovCloud S3 was introduced in August 2011
- 14 month AR hybrid model generated using commercial AWS S3 pricing up to August '11 and GovCloud S3 pricing since
- Model 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%)

Commercial AWS EC2 – Compute Prices

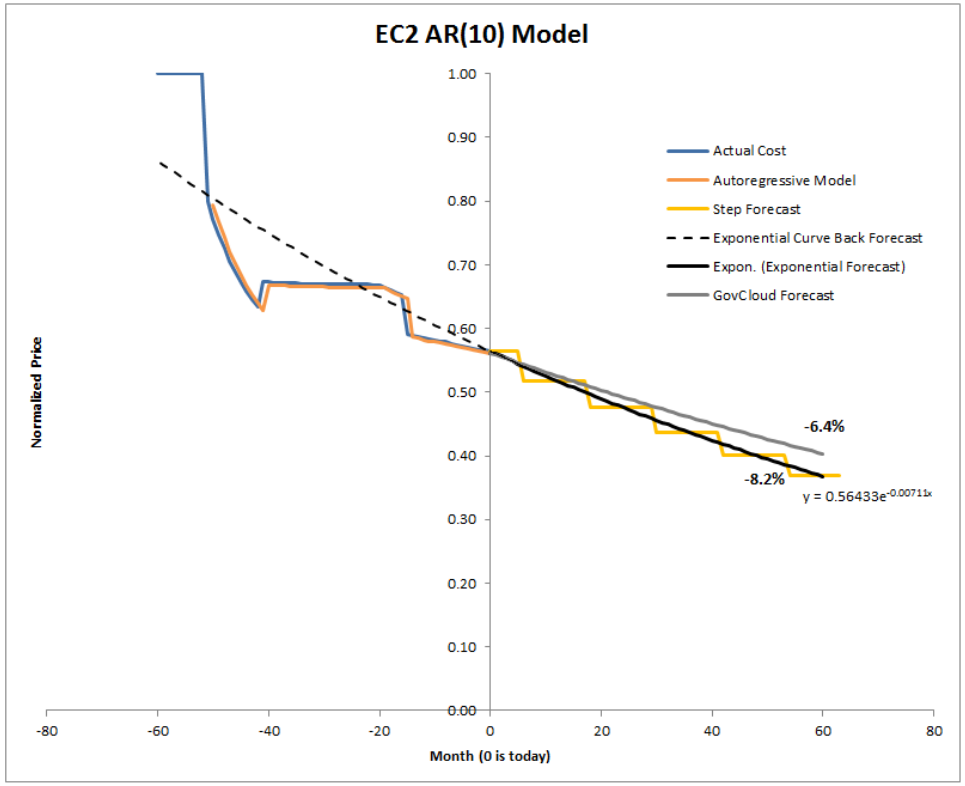
- AWS EC2 offers thousands of combinations of service offerings
  - e.g. m5.xlarge Linux 1 Year Standard Reserved All Upfront
  - Analysis focused on popular services used
- AWS EC2 prices have decreased 41% over the past five years (11% annually)
  - Includes explicit service price reductions and new 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



AWS GovCloud EC2 Model

- 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

- $\frac{11.6\%}{14.9\%} * 8.2\% = 6.4\%$



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

SUPRA ET ULTRA

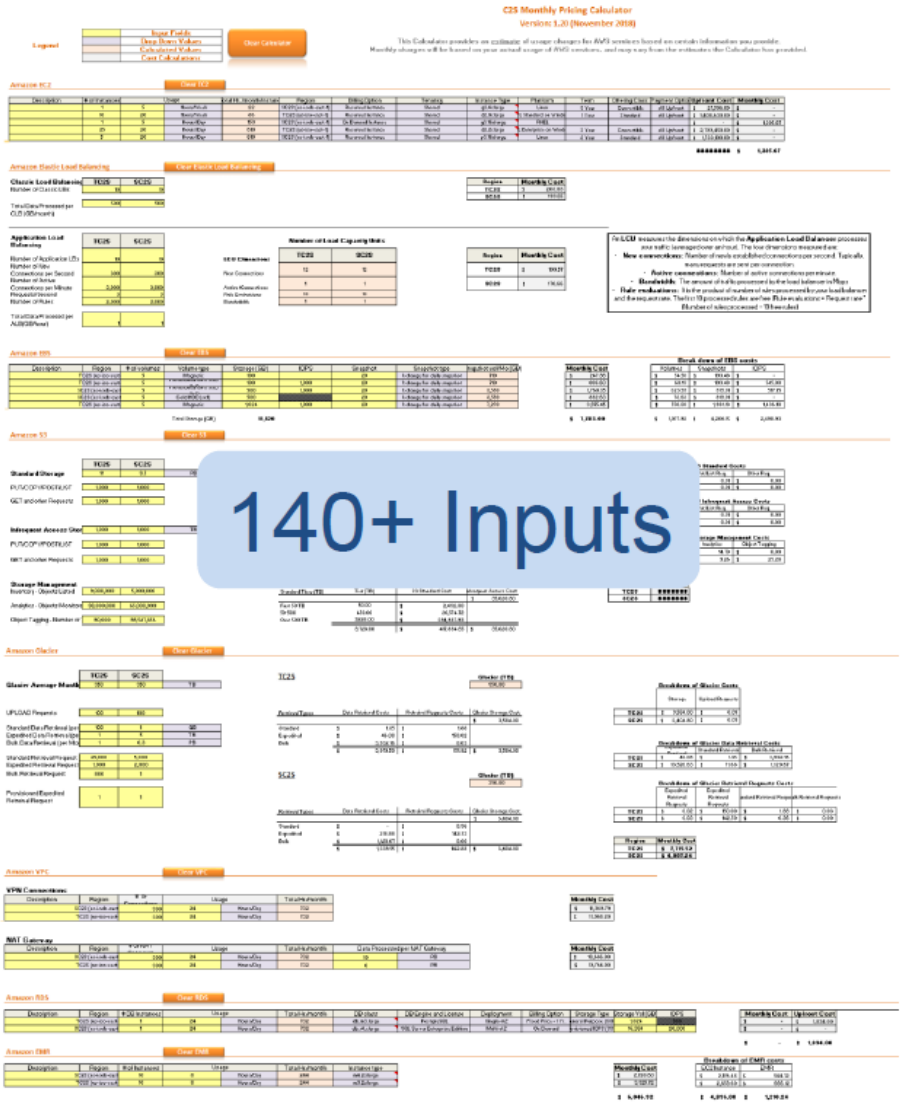
AWS Monthly Pricing Calculator

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**



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

IC Cloud Provider	Project	Item Description	DEV / TEST / PROD	Cloud Service Type	Begin Date	End Date	Usage Profile	Compute				Storage		Database
								vCPUs	Memory / CPU (GiB)	Use Case	Hours / Day Usage	Storage Type	Storage (GB)	Deployment

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

IC Cloud Provider	Project	Item Description	DEV / TEST / PROD	Cloud Service Type	Begin Date	End Date	Usage Profile	Compute				Storage		Database
								vCPUs	Memory / CPU (GiB)	Use Case	Hours / Day Usage	Storage Type	Storage (GB)	Deployment

CAAG Datasheet: Storage

- CAAG datasheet simplifies storage into 6 storage types
  - Block Storage – General Purpose
  - Block Storage – High /IO
  - 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)

IC Cloud Provider	Project	Item Description	DEV / TEST / PROD	Cloud Service Type	Begin Date	End Date	Usage Profile	Compute				Storage		Database
								vCPUs	Memory / CPU (GiB)	Use Case	Hours / Day Usage	Storage Type	Storage (GB)	Deployment

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

IC Cloud Provider	Project	Item Description	DEV / TEST / PROD	Cloud Service Type	Begin Date	End Date	Usage Profile	Compute				Storage		Database
								vCPUs	Memory / CPU (GiB)	Use Case	Hours / Day Usage	Storage Type	Storage (GB)	Deployment

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.

Assumptions

Global	Assumption	Source
FY Start	2019	
First Price Reduction Date	10/1/2019	CAAG AWS Pricing Study

Compute (EC2)	Assumption	Source
OS Type	Red Hat Enterprise Linux	
RI Billing Type	1-year	
RI Adoption %		
Annual Price Reduction	5.8%	CAAG AWS Pricing Study
EC2 Other Factor	0.9%	Actual C2S usage 10/17 - 9/18

Storage (S3, Glacier, EBS)	Assumption	Source
Annual Price Reduction	10.8%	CAAG AWS Pricing Study
Storage Other Factor	8.1%	Actual C2S usage 10/17 - 9/18

Database (RDS)	Assumption	Source
Annual Price Reduction	7.6%	AWS Pricing Study (ext. to RDS)
RDS Other Factor	0.0%	Actual C2S usage 10/17 - 9/18

Other	Assumption	Source
Other AWS Services Factor	2.6%	Actual C2S usage 10/17 - 9/18
Enterprise Support Factor	2.0%	Enterprise Support is 2% of total

Bulk Buy Savings	Assumption	Source
FY19		
FY20		
FY21		
FY22		
FY23		
FY24		
FY25		
FY26		
FY27		
FY28		

See User Documentation for more guidance on assumptions

- 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

Try C3M for Free Today

- C3M IntelDocs Links

Model: htpps://go.ic.gov/fpzGHSI

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

NATIONAL RECONNAISSANCE OFFICE

SUPRA ET ULTRA

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.

Months between price change		32	14	10	15	10	14	2	32	17
	Effective Date	3/14/2006	11/1/2008	1/1/2010	11/1/2010	2/1/2012	12/1/2012	2/1/2014	4/1/2014	12/1/2016
Storage	0-1 TB	0.15	0.15	0.15	0.14	0.125	0.095	0.085	0.03	0.023
	1-50 TB	0.15	0.15	0.15	0.125	0.11	0.08	0.075	0.0295	0.023
	50-100 TB	0.15	0.14	0.14	0.11	0.095	0.07	0.06	0.029	0.022
	100-500 TB	0.15	0.13	0.13	0.11	0.095	0.07	0.06	0.029	0.022
	500-1000 TB	0.15	0.12	0.105	0.095	0.09	0.065	0.055	0.0285	0.021
	1000-5000 TB	0.15	0.12	0.08	0.08	0.08	0.06	0.051	0.028	0.021
	5000 TB or More	0.15	0.12	0.055	0.055	0.055	0.055	0.043	0.0275	0.021

	Storage (TB)	Effective Price ($ per GB/Month)								
Enter storage level ->	7,110	0.1500	0.1209	0.0781	0.0759	0.0744	0.0596	0.0496	0.0280	0.0211

Data Collection

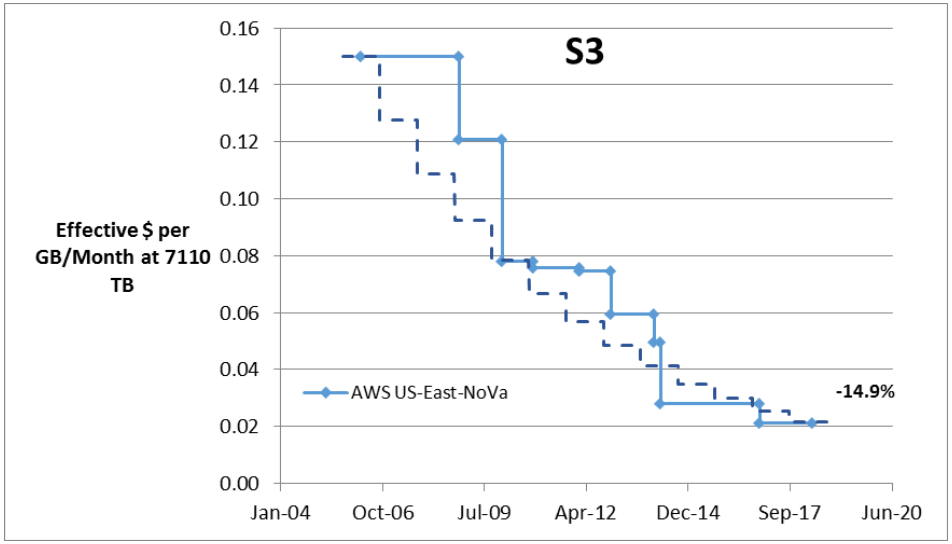
- Collected raw data from Amazon to retrieve 2015-2018 pricing information
- Used archival website to find archived AWS S3 pricing dated back to 2006 and EC2 pricing to 2010
- Compiled the EC2 data by OS, Term Type and Product Family
- Normalized the data in order to find the annual percent price change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

Calculate Annual Price Change

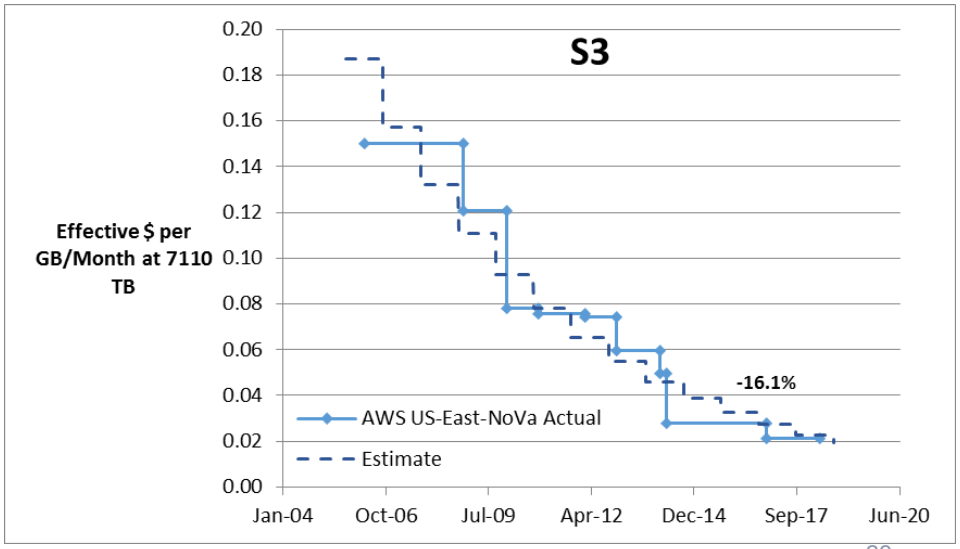
- Compare today's price to the S3 introductory price and find annual percent change from S3 introduction to today

$$\left(\frac{p_{today}}{p_0}\right)^{\left[\frac{1}{t_0 - t_{today}}\right]} - 1 = -14.9\%$$

- Prices would be underestimated in almost the entire timeframe

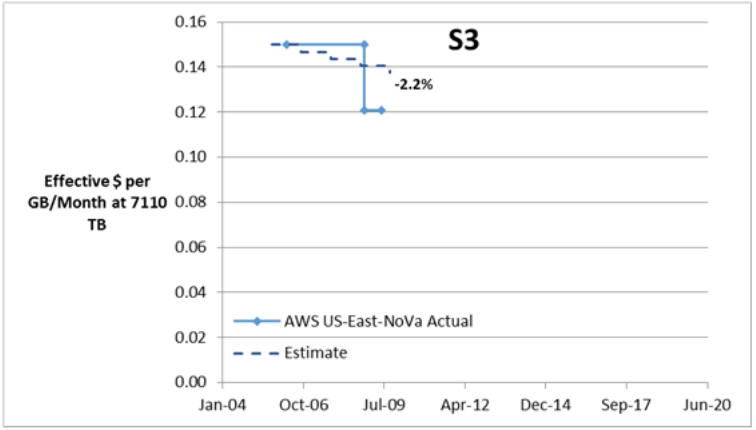


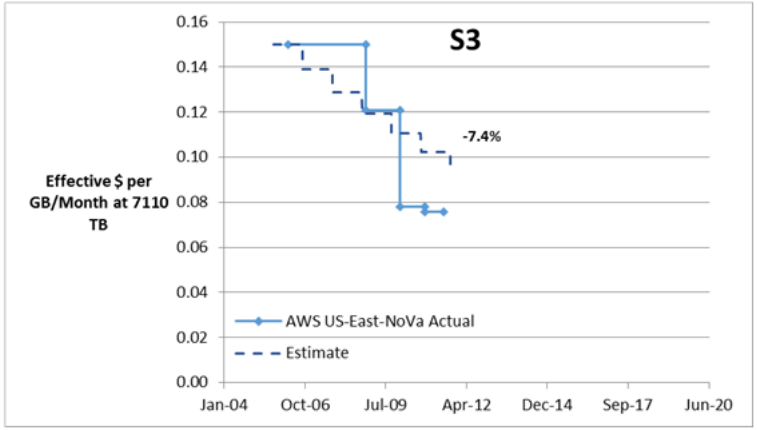
- Next the underestimate bias was addressed by not constraining the introductory price
- This model resulted in an estimated 2006 price 25% higher than the actual price

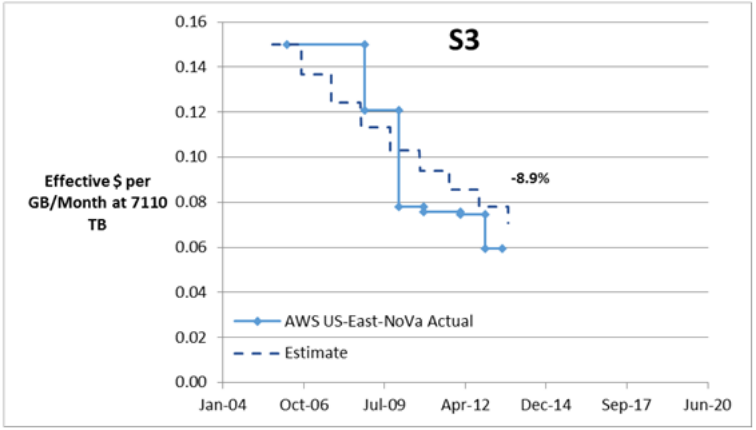


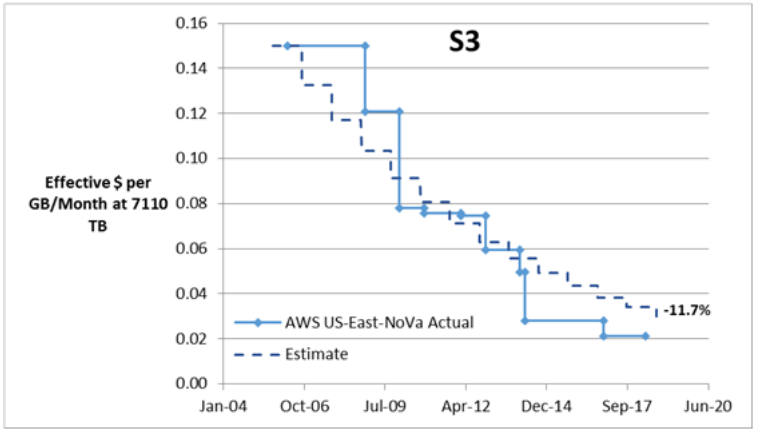
Constrain Model Area and p_0

- 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, 2TB in the second, 3TB 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.

Provide storage in GB (1 TB = 1,024 GB; 1 PB = 1,048,576 GB)

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.

Causality and Uncertainty: A New Wave for Cost Estimation

Robert Stoddard

Mike Konrad

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

[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.

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.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM19-0804

Agenda

Original SEI Cost Research Motivation

The QUELCE Research Project Solution

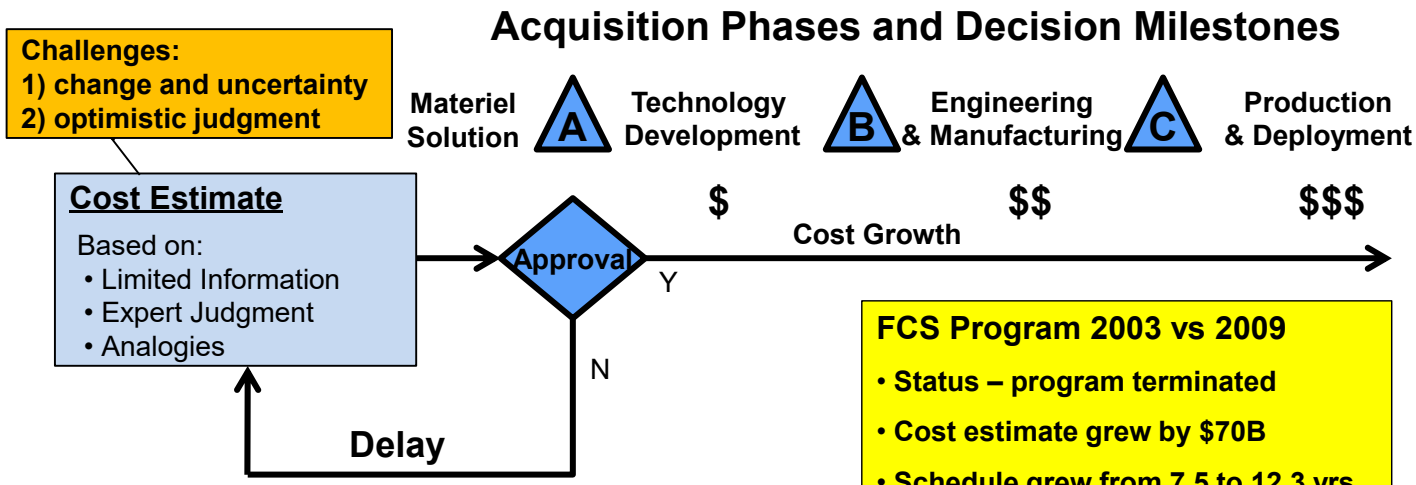
SEI Causal Learning Research

Call to Action

© 2019 Carnegie Mellon University

[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.

Original SEI Cost Research Motivation



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**

Source: GAO-12-181T

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

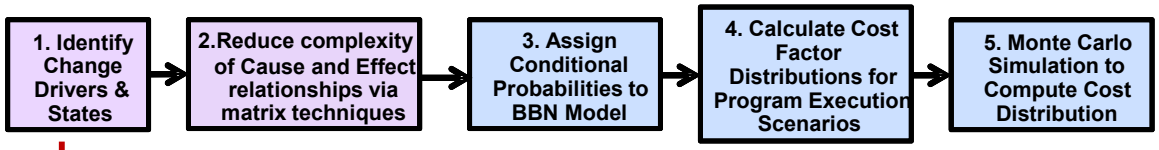
Weapon Systems Acquisition Reform Act (WSARA) 2009 requirements for Milestone A approval.

© 2019 Carnegie Mellon University

[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.

The QUELCE Solution

Step 1: Identify Change Drivers and States



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

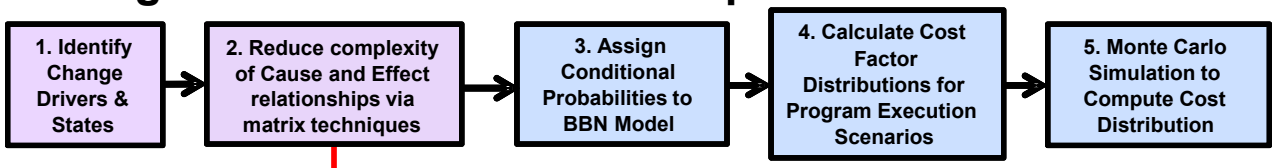
Domain-Specific Program Change Drivers Identified

© 2019 Carnegie Mellon University

[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.

The QUELCE Solution

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



Change Drivers - Cause & Effects Matrix

Causes \ Effects	Mission / CONOPS	Change in Strategic Vision	Capability Definition	Advocacy Change	Closing Technical Gaps (CBA)	Building Technical Capability & Capacity (CBA)	Interoperability	Systems Design	Interdependency	Functional Measures	Scope Definition	Functional Solution Criteria (measure)	Funding Schedule	Acquisition Management	Program Mgt - Contractor Relations	Project Social / Dev Env	Prog Mgt Structure	Manning at program office
Mission / CONOPS			3								3							
Change in Strategic Vision			3	3		3			2				2					
Capability Definition								3					0	2	1	1	0	0
Advocacy Change											2			1			1	1
Closing Tech																		
Building Tech																		
Interoperabili																		
Systems Des																		

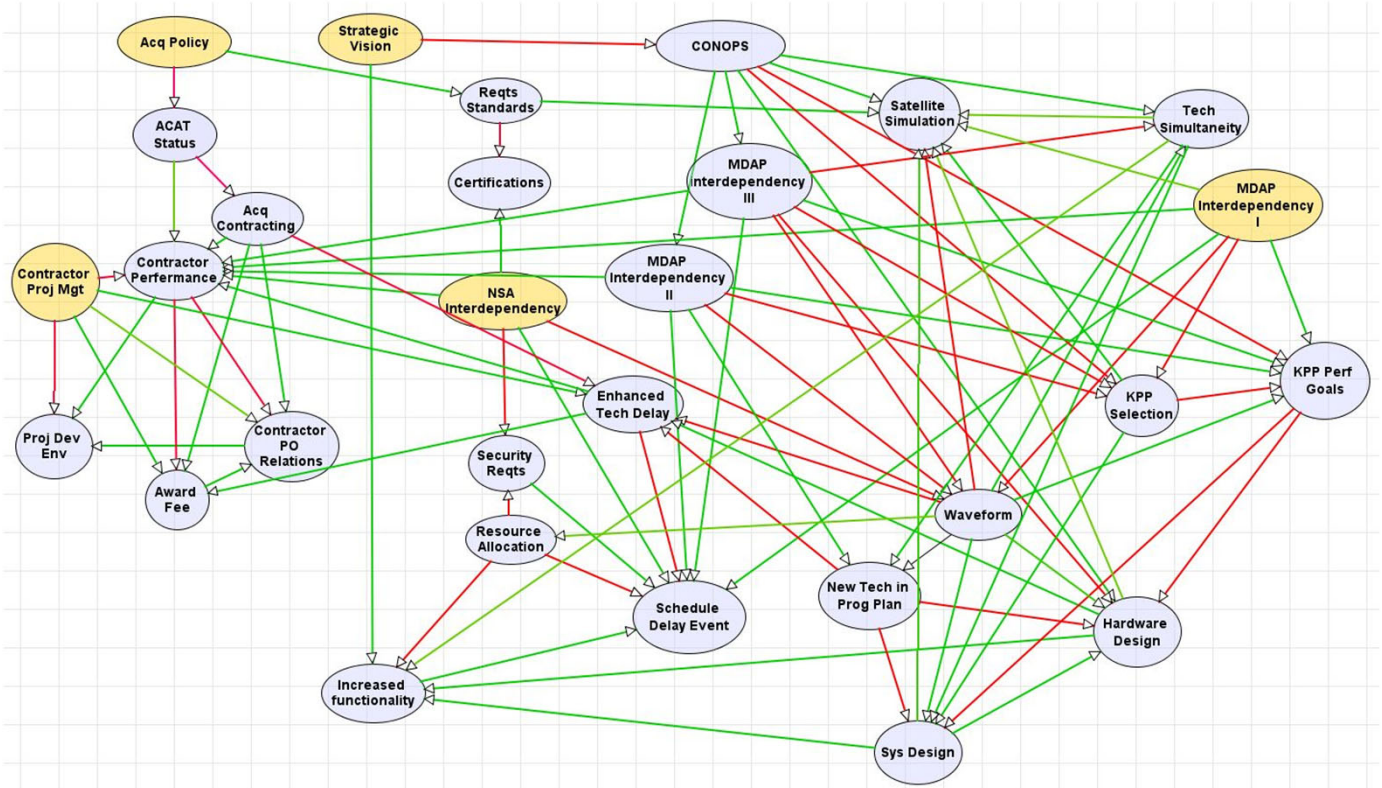
Capturing interrelationships among change drivers and reducing the complexity of the network

© 2019 Carnegie Mellon University

[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.

The QUELCE Solution

Step 3: Assign Conditional Probabilities to BBN Model

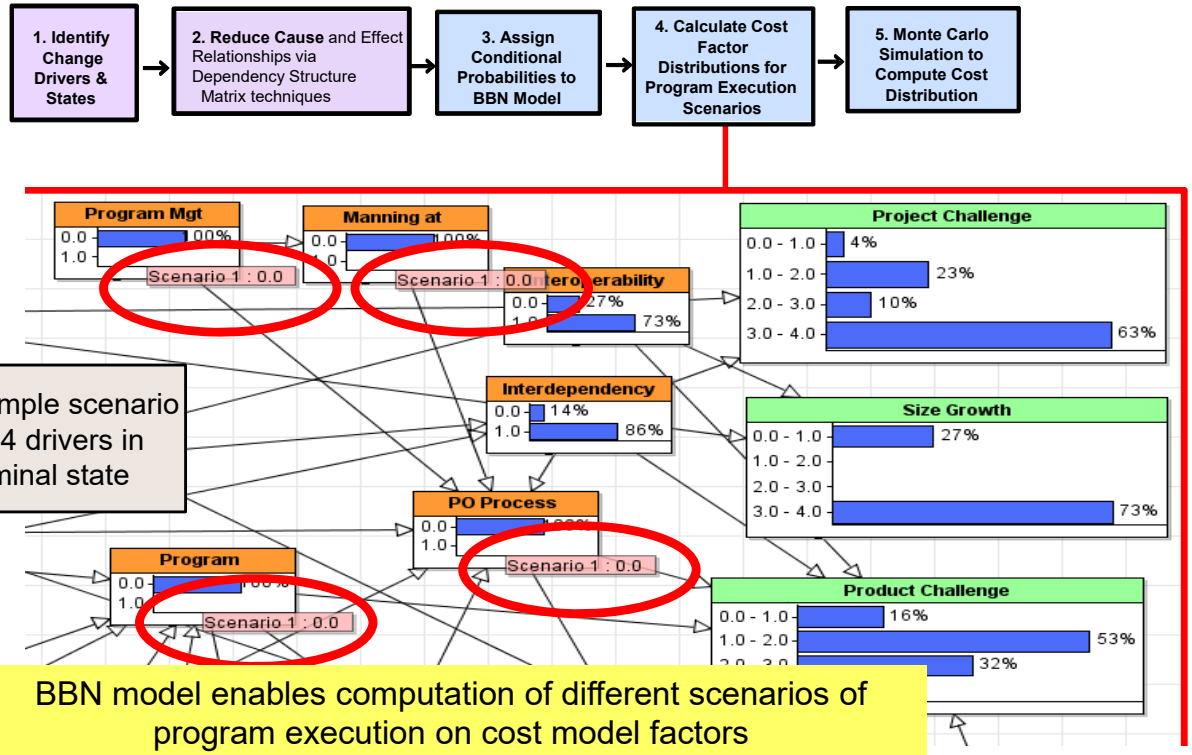


© 2019 Carnegie Mellon University

[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.

The QUELCE Solution

Step 4: Calculate Cost Factor Distributions for Program Execution Scenarios



An example scenario with 4 drivers in nominal state

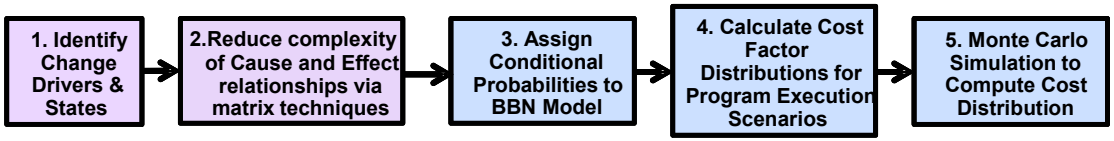
BBN model enables computation of different scenarios of program execution on cost model factors

© 2019 Carnegie Mellon University

[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.

The QUELCE Solution

Step 5a: Connecting BBNs to Cost Estimation Models



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

COCOMO Parameter	
Scale Factors	PREC
	FLEX
	RESL
	TEAM
	PMAT
Effort Multipliers	PERS
	RCPX
	PDIF
	PREX
	FCIL
	RUSE
	SCED

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

COCOMO Parameter		XL	VL	L	N	H	VH	EH
Scale Factors	PREC			1	3	5		
	FLEX		1	2	3	5		
	RESL	1	2	3	4	5		
Effort Multipliers	RCPX			1	2	3	4	5
	PDIF			1	5			
	RUSE				1	3	5	

Project Challenge factors (1=low…5=high)

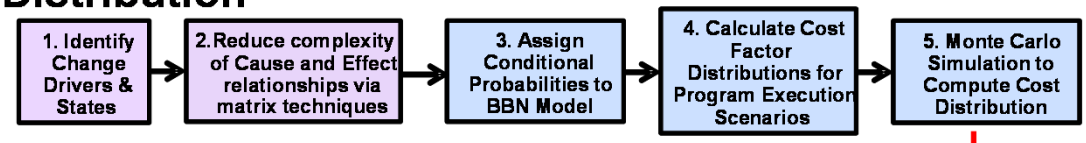
COCOMO Parameter		XL	VL	L	N	H	VH	EH
Scale Factors	TEAM	1	3	5				
	PMAT		1	2	3	4	5	
Effort Multipliers	PERS			1	3	5		
	PREX			1	2	3	4	5
	FCIL				1	3	5	
	SCED	1	3	5				

© 2019 Carnegie Mellon University

[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.

The QUELCE Solution

Step 5b: Monte Carlo Simulation to Compute Cost Distribution

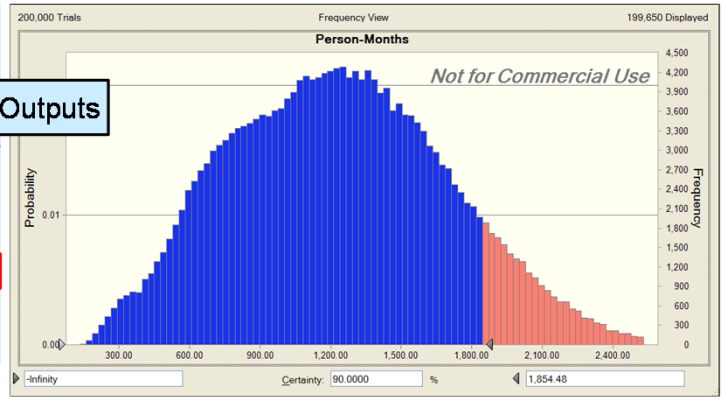


Monte Carlo simulation using program change factor distributions uses uncertainty on the input side to determine the cost estimate distribution

	A	B	C	D
1	Effect			
2	Product Challenge	5		
3	Project Challenge	4		
4	Estimated Size (KSLOC)	50		
5	Product Challenge factors		5	
6	COCOMO Parameter		XL	VL
7	Scale Factors	PREC		4
8		Val		6.2
9		FLEX		5

BBN Outputs

Mapped COCOMO value



© 2019 Carnegie Mellon University

[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.

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!

© 2019 Carnegie Mellon University

[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.

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

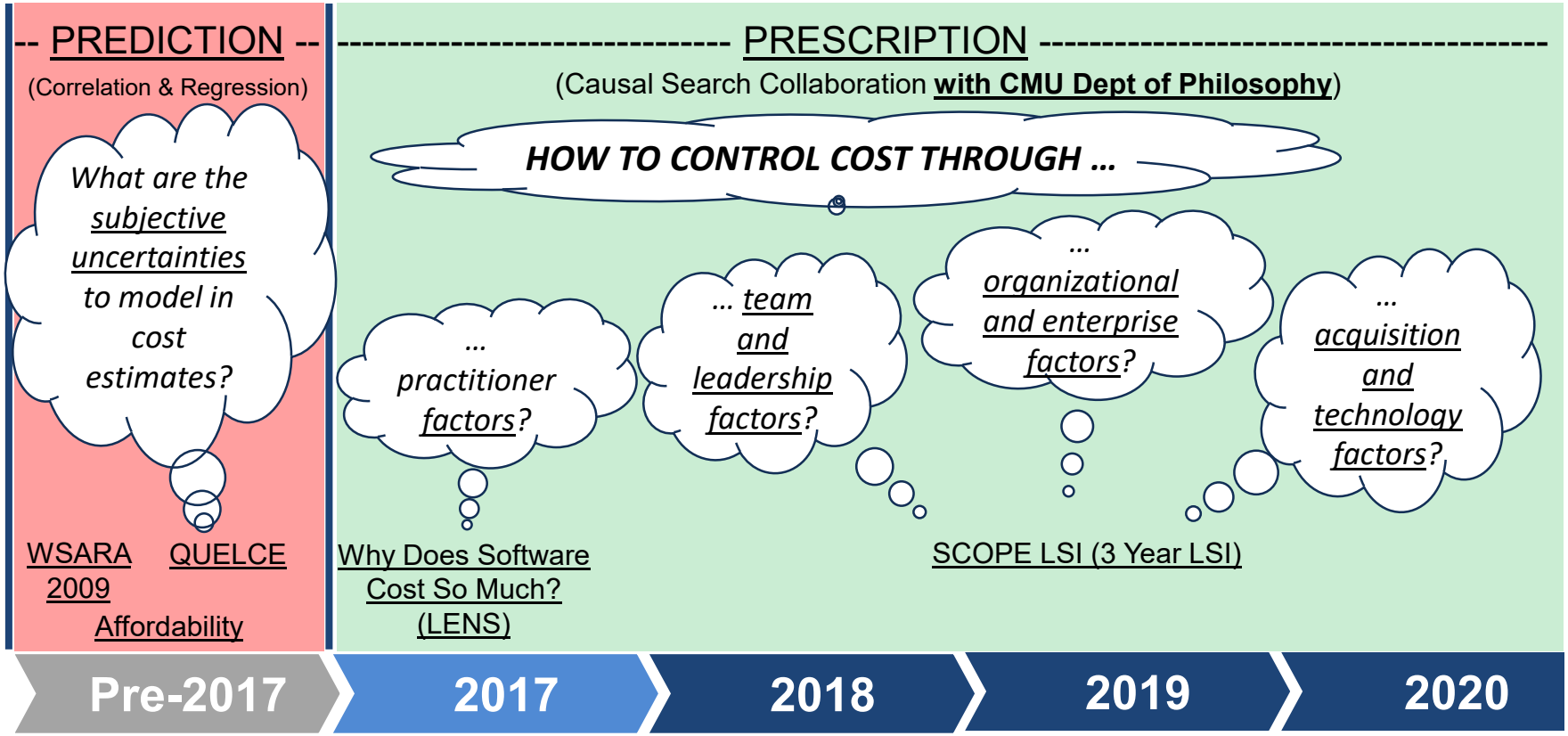
Causal learning was a novel leap from predominant use in medical research to use in software cost research

Causal learning has grown in use to six SEI research projects in past 3 years

© 2019 Carnegie Mellon University

[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.

SEI Research Journey from Uncertainty to Causality



© 2019 Carnegie Mellon University

[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.

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

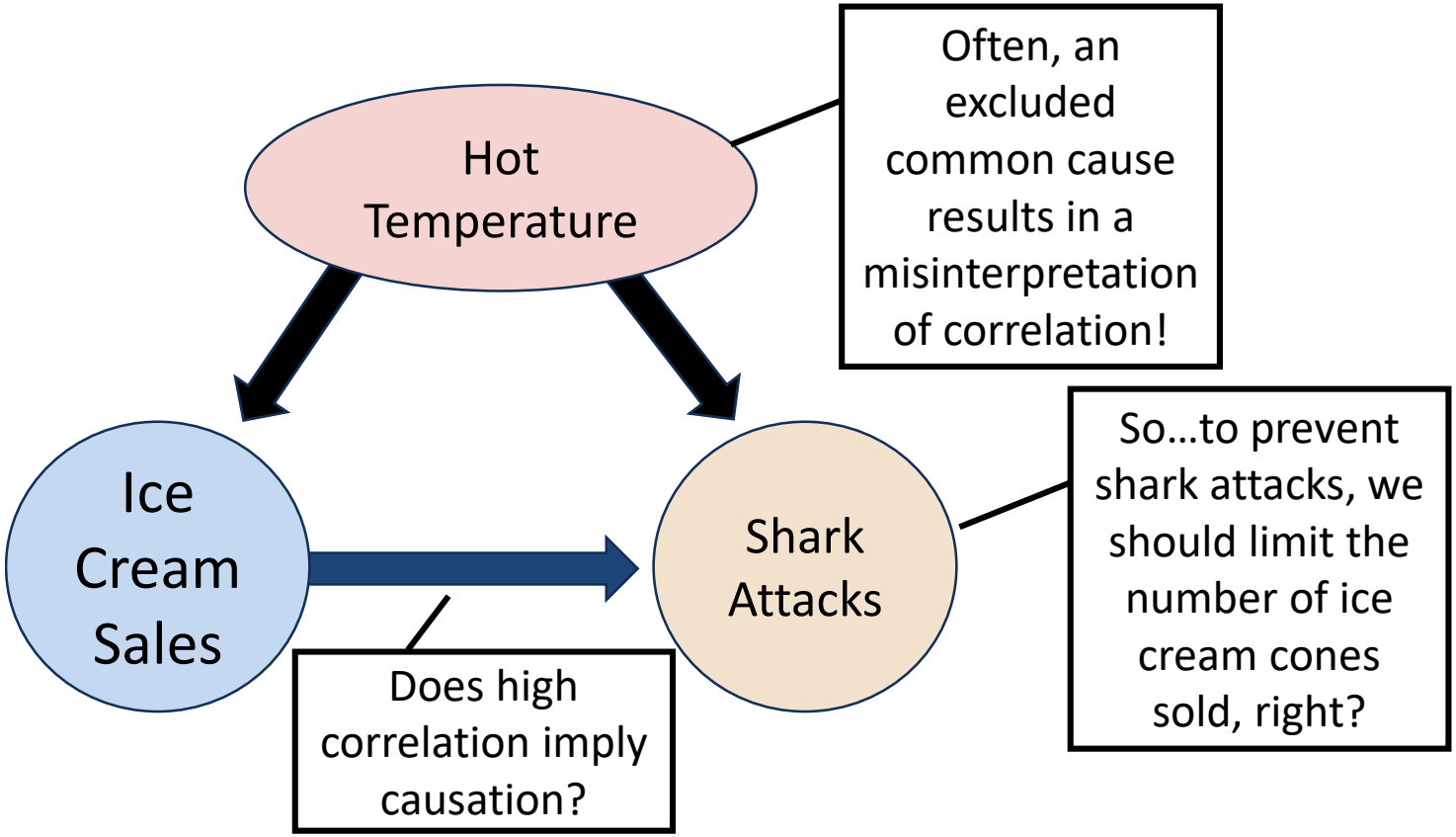
We can now evaluate causation without expensive and difficult experiments

Establishing causation with observational data remains a vital need and a key technical challenge, but is becoming more feasible and practical.

© 2019 Carnegie Mellon University

[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.

Primary Reason for Spurious Association



© 2019 Carnegie Mellon University

[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.

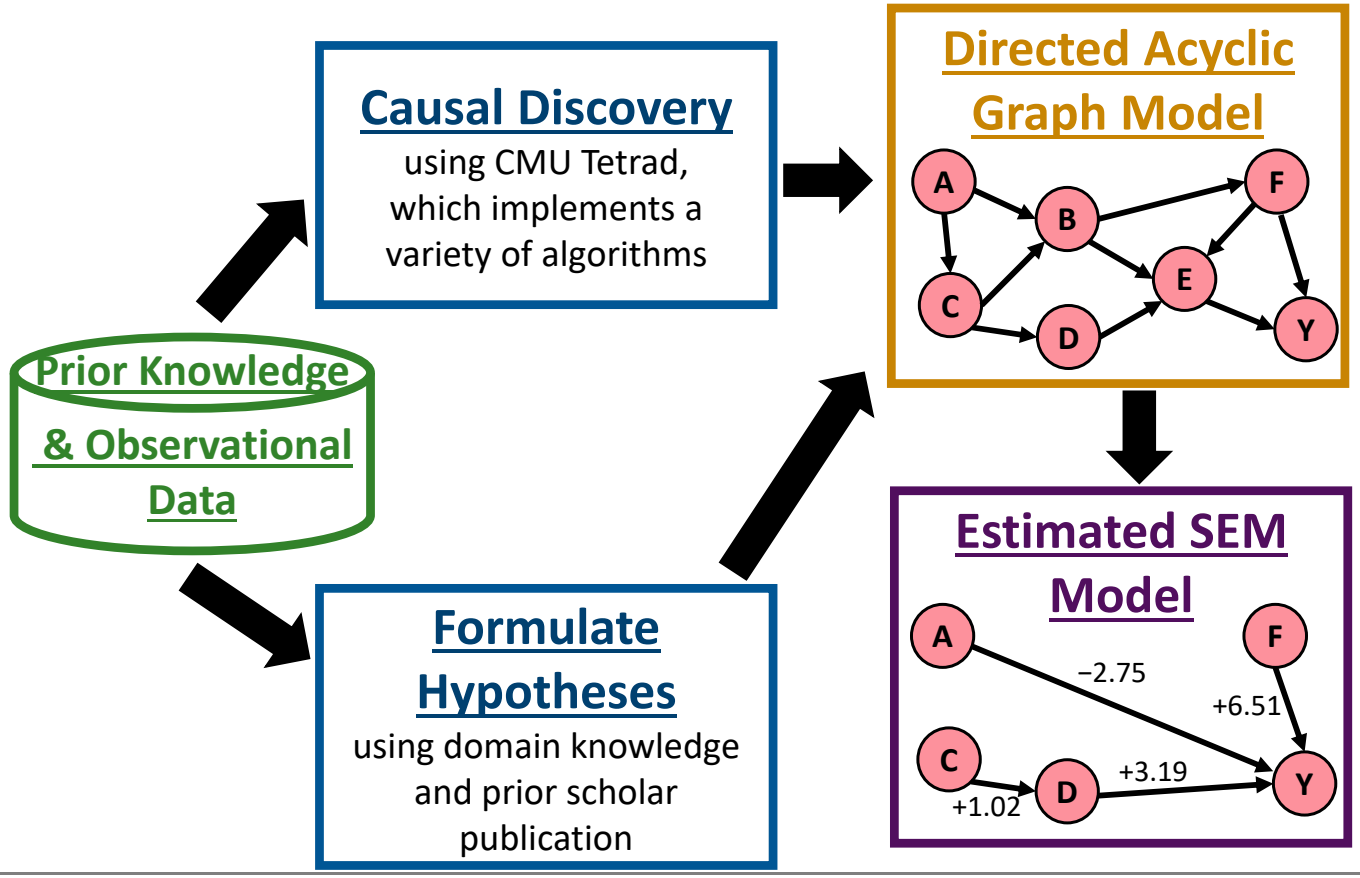
Different Uses for Correlation versus Causation

Correlation	Causation
Classifying & identifying	Influencing & acting
Informational value of different evidence	Using evidence to guide policy or actions
Prediction & reasoning given observations	Prediction & reasoning given interventions
Probable explanations for some event or issue	Ways to produce or prevent an event or problem

© 2019 Carnegie Mellon University

[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.

Landscape of Causal Learning

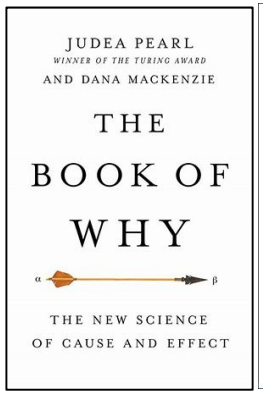


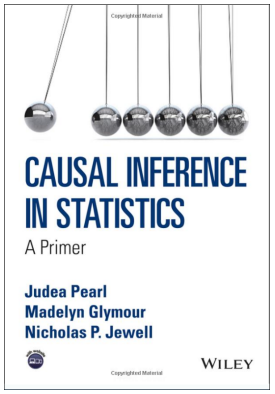
© 2019 Carnegie Mellon University

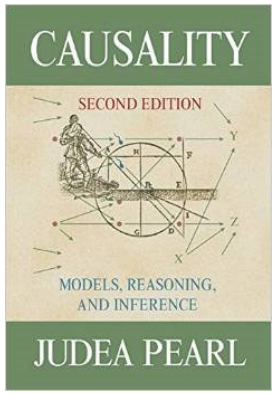
[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

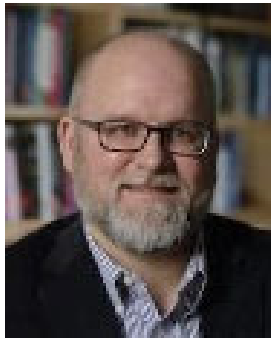
Causal Learning as a Discipline

Judea Pearl

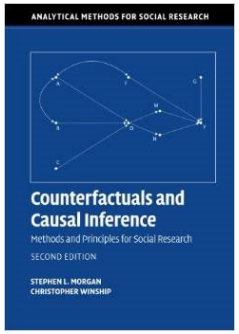


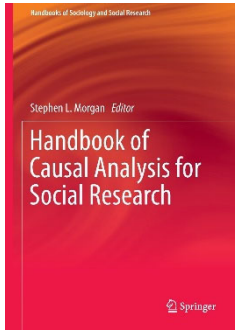






Stephen Morgan





Richard Scheines

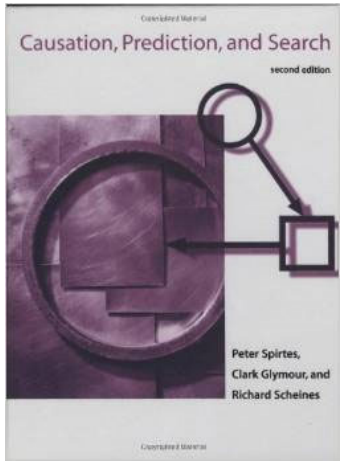
David Danks

Clark Glymour

Peter Spirtes

Joe Ramsey

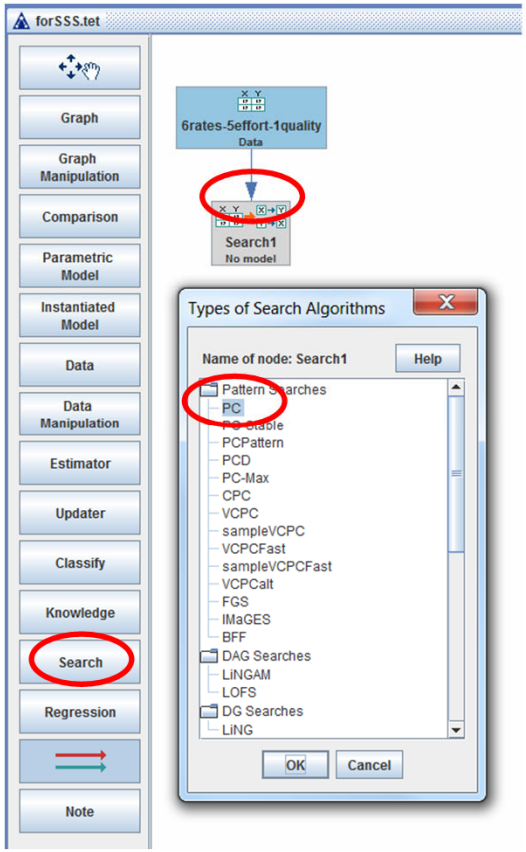
Kun Zhang



© 2019 Carnegie Mellon University

[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.

Causal Search Algorithms and Tooling

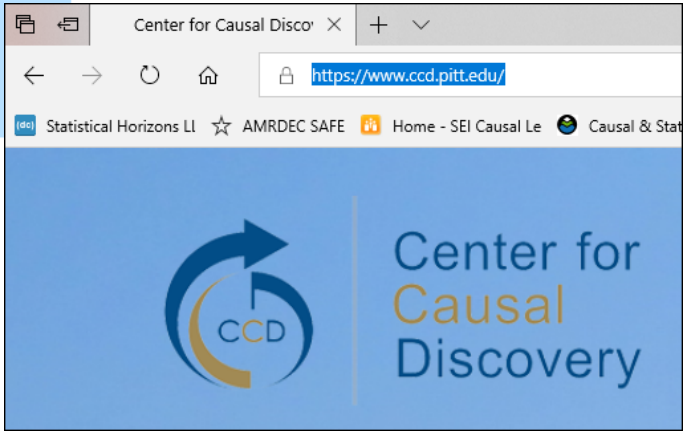


Step 4: Initiate a causal search by inserting a Search box with a connection from the Data box.

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/



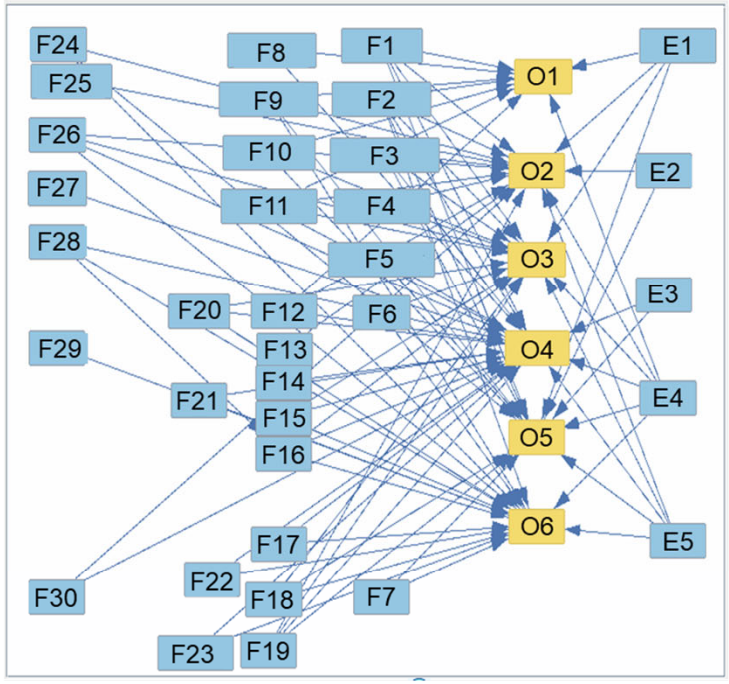
© 2019 Carnegie Mellon University

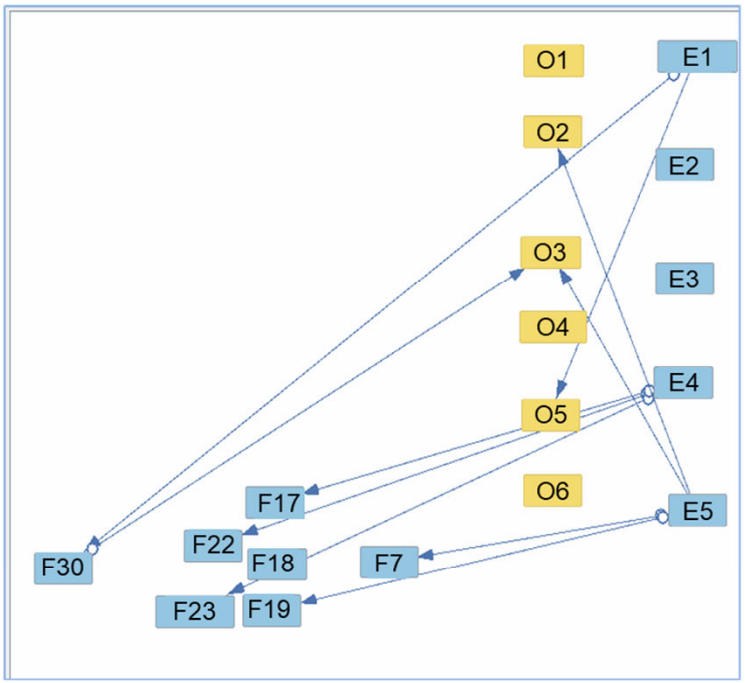
[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

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



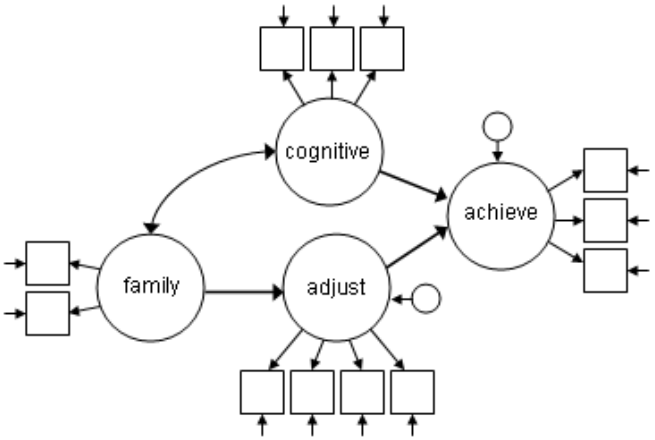


Key: O1 Outcome
E2 Environment variable
F3 Internal Performance factor

© 2019 Carnegie Mellon University

[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.

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

© 2019 Carnegie Mellon University

[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.

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***

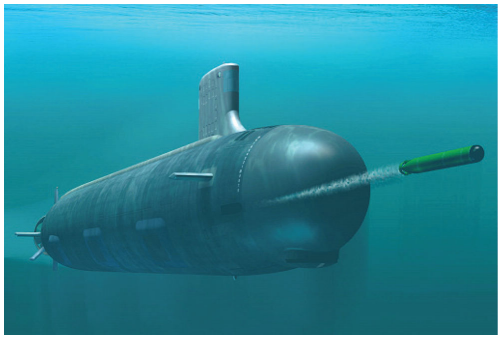
© 2019 Carnegie Mellon University

[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.

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?

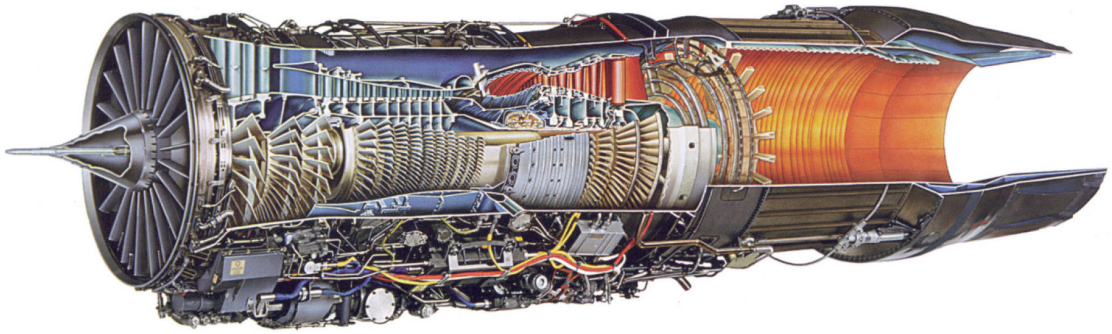
© 2019 Carnegie Mellon University

[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.

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?

© 2019 Carnegie Mellon University

[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.

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

© 2019 Carnegie Mellon University

[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.

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

© 2019 Carnegie Mellon University

[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.

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

© 2019 Carnegie Mellon University

[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.

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

© 2019 Carnegie Mellon University

[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.

PolicyViz

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

Technomics

Better Decisions Faster

Data Exploitation – More Value, Less Time

IT-CAST

August 20 – 21, 2019

Who am I?

Adam H. James

Technomics, Inc.

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]

[1] James, Adam, Jeff Cherwonik, and Brandon Bryant. Don't Just Use Your Data… Exploit It. *ICEEA Professional Development & Training Workshop*, Tampa, FL, May 14-17, 2019. Unpublished Conference Paper. Online.

[2] Bazzy, Rich, Adam James, and Jeff Cherwonik. A New Approach When Cost/Capability Trades Matter Most. *ICEEA Professional Development & Training Workshop*, Tampa, FL, May 14-17, 2019. Unpublished Conference Paper. Online.

The Future is Data

Technomics Innovation Lab

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**

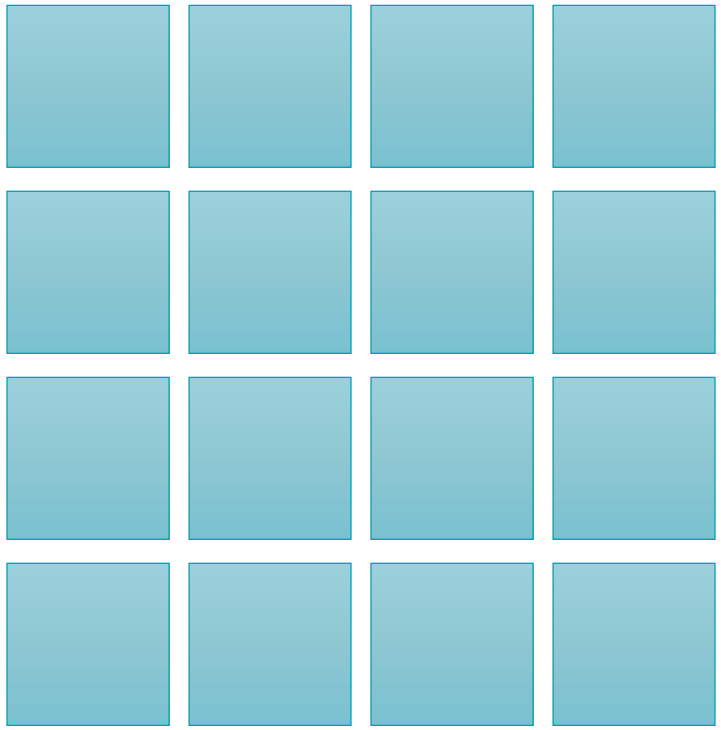
2. How we think of data must change
   - **Machine readable** is more desirable than human readable
   - The right tool for the right job – out of spreadsheets into the future

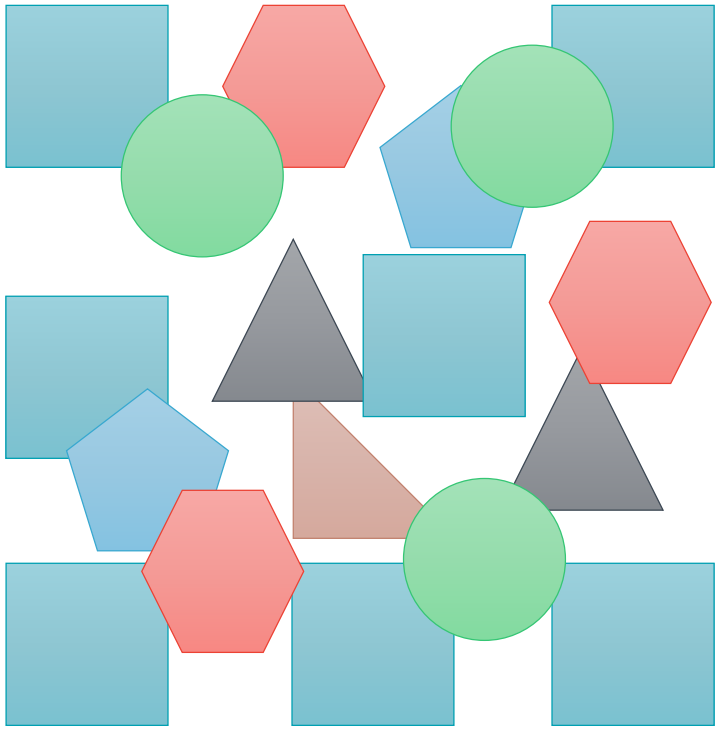
3. All analysts must understand **data structure**
   - Efficient for a machine to process
   - Enables rapid exploitation of data
   - Necessary for quantitative analysis

What is Structure?

structured data

aligns nicely and fits into fixed tables





unstructured data

does not (e.g., text blocks, images, audio, video)

Staying Focused

- 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

Structured Data Standards

- 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[1]
  - Forms the foundation of most “data science” online courses[2]
  - 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)

[1] Wickham, Hadley. (2014). Tidy data. *The American Statistician*. 14. 10.18637/jss.v059.i10.
[2] https://r4ds.had.co.nz/tidy-data.html

Rows and Columns

Technomics Innovation Lab

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

Bad

System	WBS Element	Cost ($)
Truck A		
	Engine Cost	50,000
	Remaining Cost	150,000
	PM Cost	1,500
	Number of Units	10
	Total Cost	2,015,000
Truck B		
	Engine Cost	40,000
	Remaining Cost	120,000
	PM Cost	2,500
	Number of Units	5
	Total Cost	812,500

Better

System	Metric	Element	Value
Truck A	Unit Cost	Engine	50000
Truck A	Unit Cost	Remaining	150000
Truck A	Unit Cost	PM	1500
Truck A	Unit Cost	Surface Vehicle	201500
Truck A	Quantity	Surface Vehicle	10
Truck B	Unit Cost	Engine	40000
Truck B	Unit Cost	Remaining	120000
Truck B	Unit Cost	PM	2500
Truck B	Unit Cost	Surface Vehicle	162500
Truck B	Quantity	Surface Vehicle	5

Single Purpose Variables

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

Original WBS	Modified WBS	Element	Model
1	1	Surface Vehicle System	
1.1.1	1.1	Surface Vehicle	Variant A
1.1.1.1	1.1.1	Engine	Variant A
1.1.1.2	1.1.2	Remaining Vehicle	Variant A
1.2.1	1.1	Surface Vehicle	Variant B
1.2.1.1	1.1.1	Engine	Variant B
1.2.1.2	1.1.2	Remaining Vehicle	Variant B

Other Data Tips (1 of 2)

Technomics Innovation Lab

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

Bad Name	Better Name
Work Breakdown Structure	WBS
% Complete	PercComp
Cost (TY $K)	CostTY_K
Unit Cost (FY18)	UnitCost_FY18
1970	Cost_1970

Be explicit in preserving **order**

Month
January
February
…
November
December

ID_Month	Month
1	January
2	February
…	…
11	November
12	December

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

Value	Formatted
100000	100,000
2018-10-05	October 5, 2018
2695.255648	$2,695.26

Take advantage of **sparse data** when possible

wbs	rec	non_rec
1.1	0	0
1.2	0	0
1.3	1000	0
1.4	2000	0
1.5	0	0
1.6	0	0

wbs	r_nr	value
1.3	rec	1000
1.4	non_rec	2000

Technomics

Other Data Tips (2 of 2)

Technomics Innovation Lab

Avoid storing **redundant information**

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

Use **intermediate** tables for calculations

Data Table(s)

Intermediary Table(s)

Analyses

Be mindful of **data types**

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

	A	B
1	Text Format	General Format
2	1	1
3	1.1	1.1
4	1.1.1	1.1.1
5	1.1.2	1.1.2
7	1.9	1.9
8	1.10	1.1
9	1.11	1.11

Technomics

Grammar of Data Manipulations

"dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges" [1]

- 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]

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

Technomics Innovation Lab

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

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

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

Technomics Innovation Lab

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

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

```
Vehicle data
  mutate value_adj =
    if (metric = cost) then (value / escalation)
    else value
  group_by metric
  summarize avg =
    mean of value_adj
```

Technomics

Unstructured Data

working with blobs of text

Unstructured (Text) Data

- 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)[1]

[1] Ferrari, Alessio & Oronzo Spagnolo, Giorgio & Gnesi, Stefania. (2017). PURE: A Dataset of Public Requirements Documents. 502-505. 10.1109/RE.2017.29.

Some Basics

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

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

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
```

Tokenization (Sentences)

Technomics Innovation Lab

```
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

doc_id	chapter_id	sent_id	sentence
1	1	1	this document includes software requirements for keepass password safe, release number
1	1	2	keepass password safe is an osi certified open source software distributed under the terms of the gnu general public license version or under

Tokenization (Words/N-Grams)

words are a common token

doc_id	chapter_id	sent_id	word
1	1	1	this
1	1	1	document
…	…	…	…
1	1	1	release
1	1	1	number
1	1	2	keepass
1	1	2	password
…	…	…	…

n-grams are groups of “n” words

doc_id	chapter_id	id_sent	ngram_2
1	1	1	this document
1	1	1	document includes
…	…	…	…
1	1	1	release number
1	1	2	keepass password
1	1	2	password safe
1	1	2	safe is
…	…	…	…

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

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

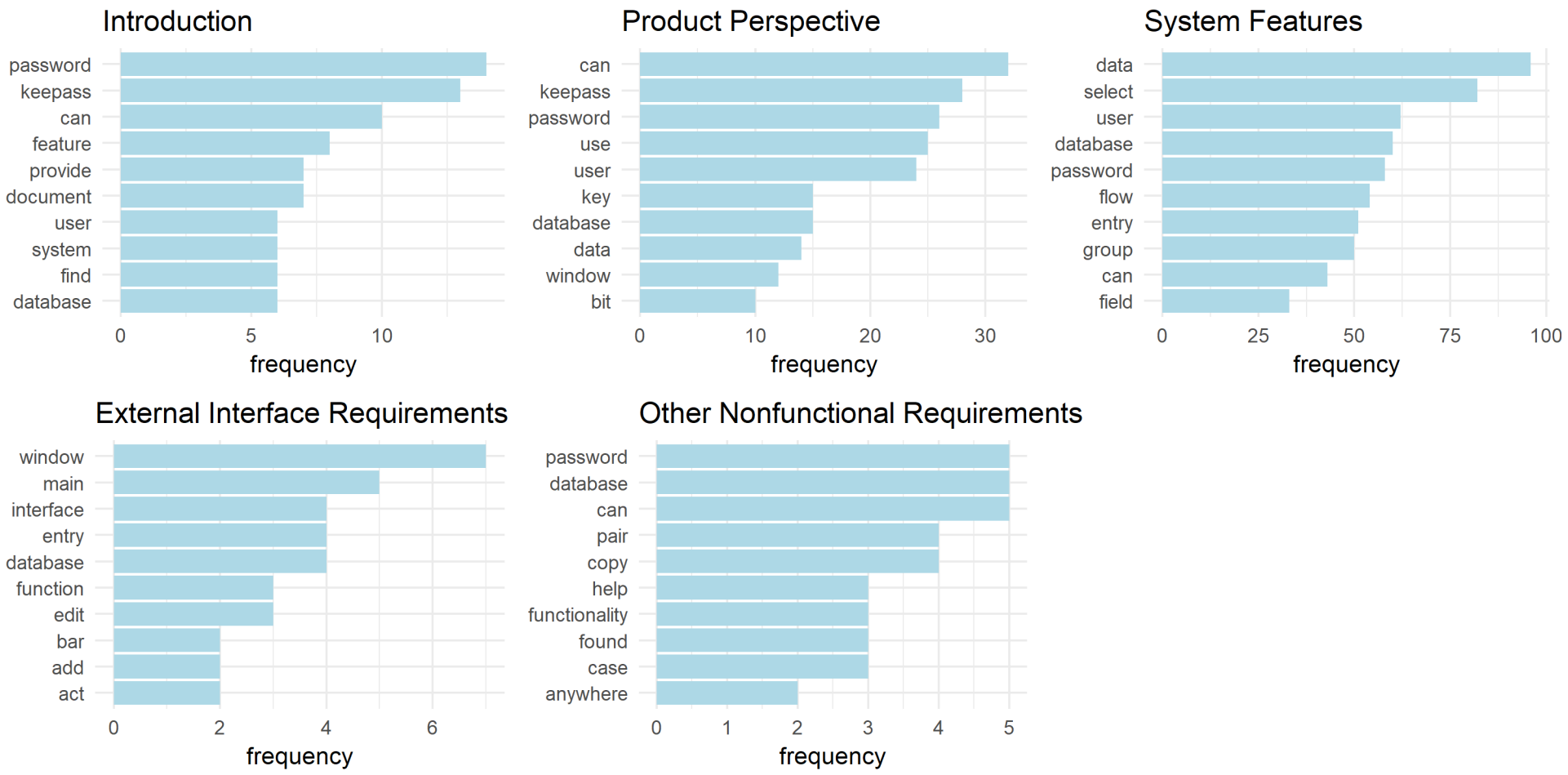
Original Word	Stem	Completed Stem	Lemma
add	add	added	add
added	ad	added	add
adding	ad	added	add
adds	add	added	add

Standardizing the Language (Example 2)

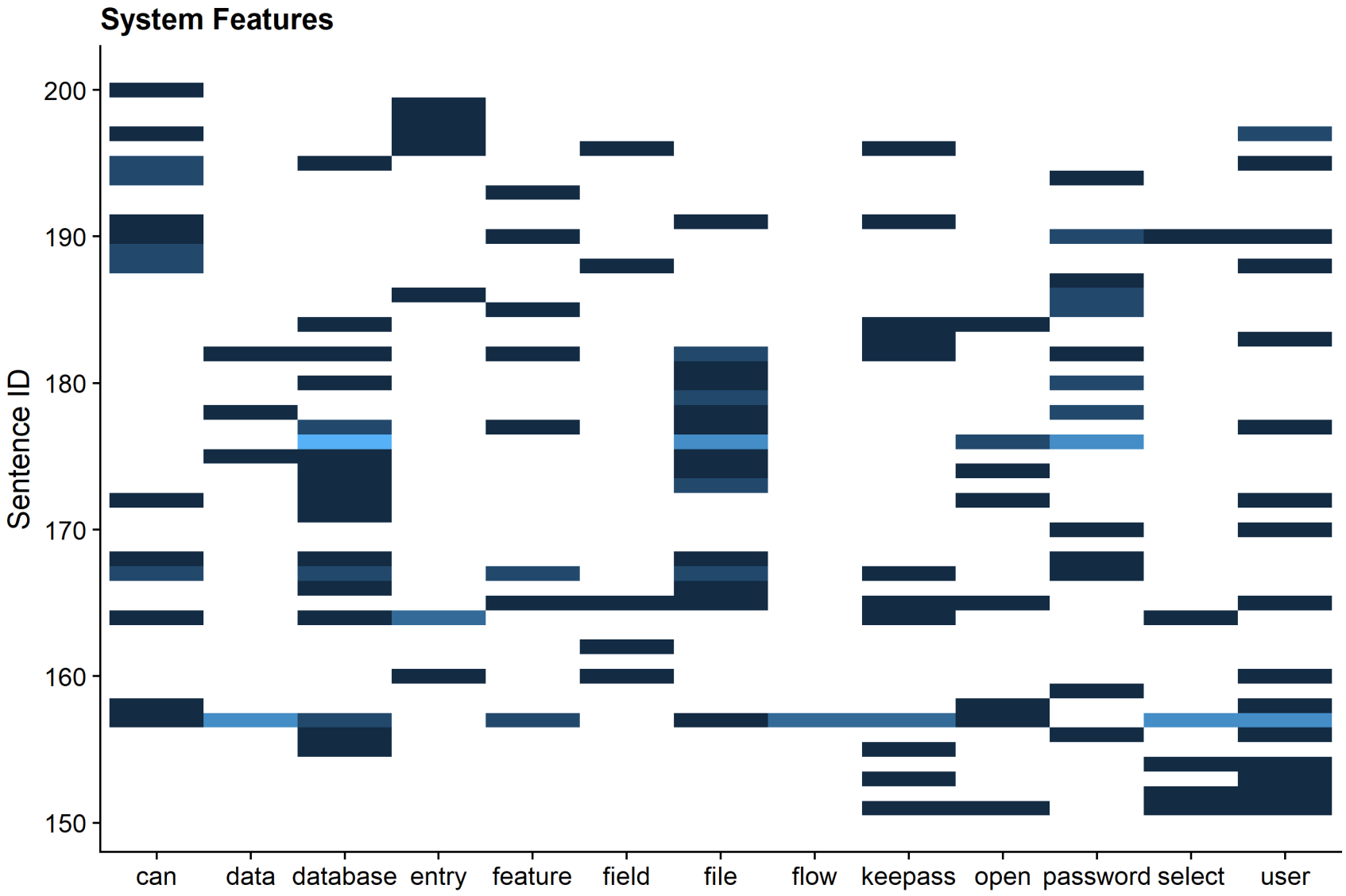
- **Stemming** is usually performed algorithmically
- **Lemmatization** requires a dictionary (not a problem) but is generally superior

Original Word	Stem	Completed Stem	Lemma
copied	copi	copied	copy
copy	copi	copied	copy
copies	copi	copied	copy
recopy	recopi		copy

Example Visualizations (1 of 3)

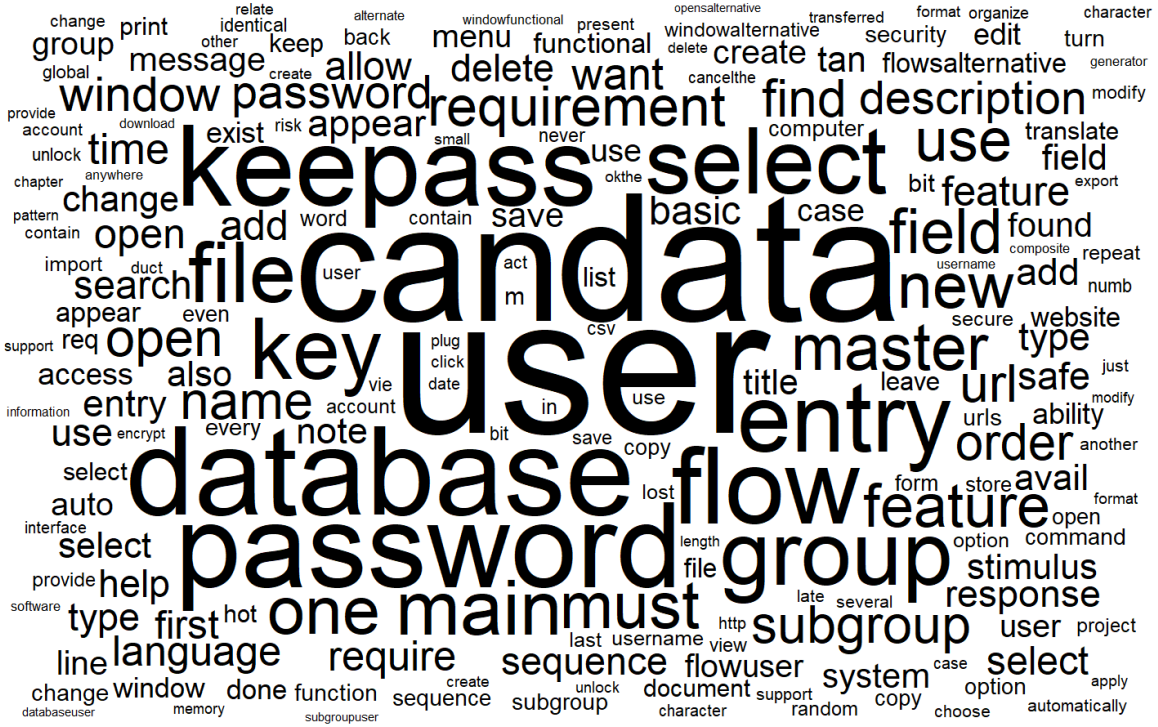


Example Visualizations (2 of 3)



Example Visualizations (3 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

Technomics Innovation Lab

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

doc_id	sent_id	audience	backup	can	case	certify	class	computer	contain	convention
1	**1**	0	0	0	0	0	0	0	0	0
1	**2**	0	0	0	0	1	0	0	0	0
1	**3**	0	0	0	0	0	0	0	0	0
1	**4**	0	0	0	0	0	0	0	0	0
1	**5**	0	0	1	0	0	0	1	0	0
1	**6**	0	0	0	0	0	0	0	0	0
1	**7**	0	1	0	0	0	0	0	0	0
1	**8**	0	0	0	0	0	0	0	0	1
1	**9**	0	0	0	0	0	0	0	0	0
1	**10**	1	0	0	1	0	1	0	1	0

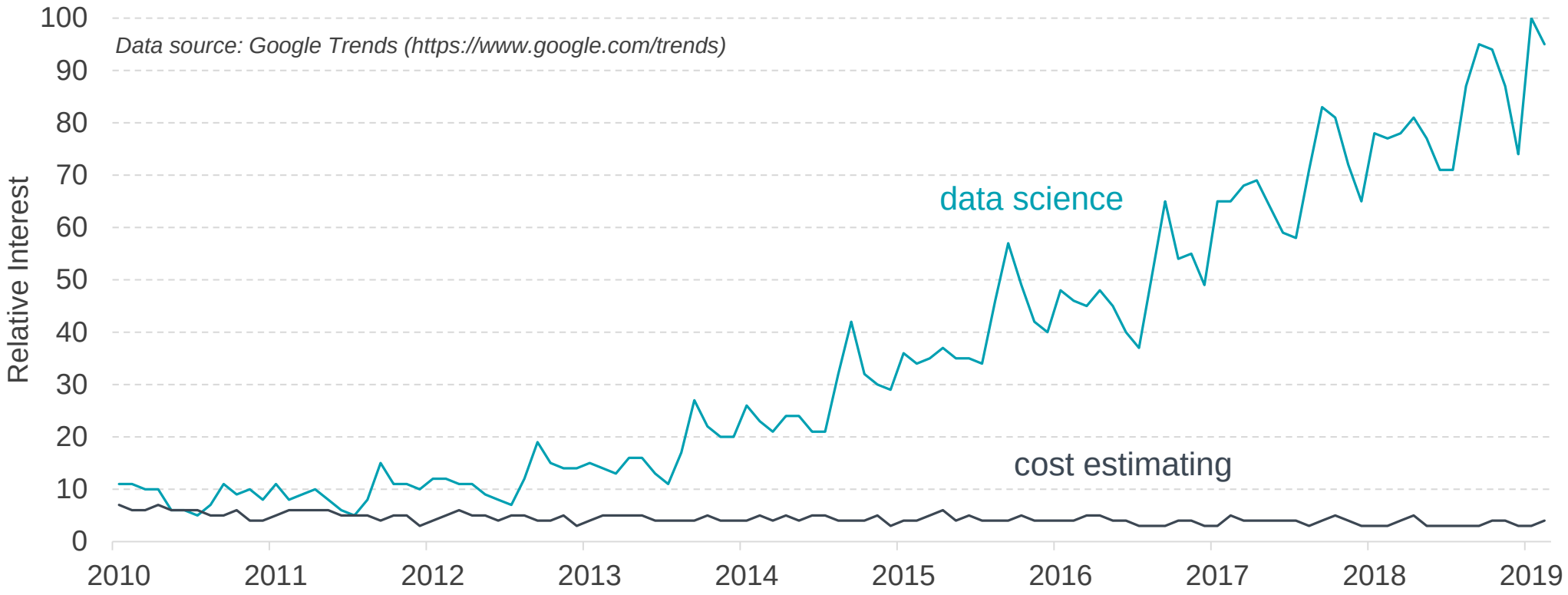
not the complete matrix

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

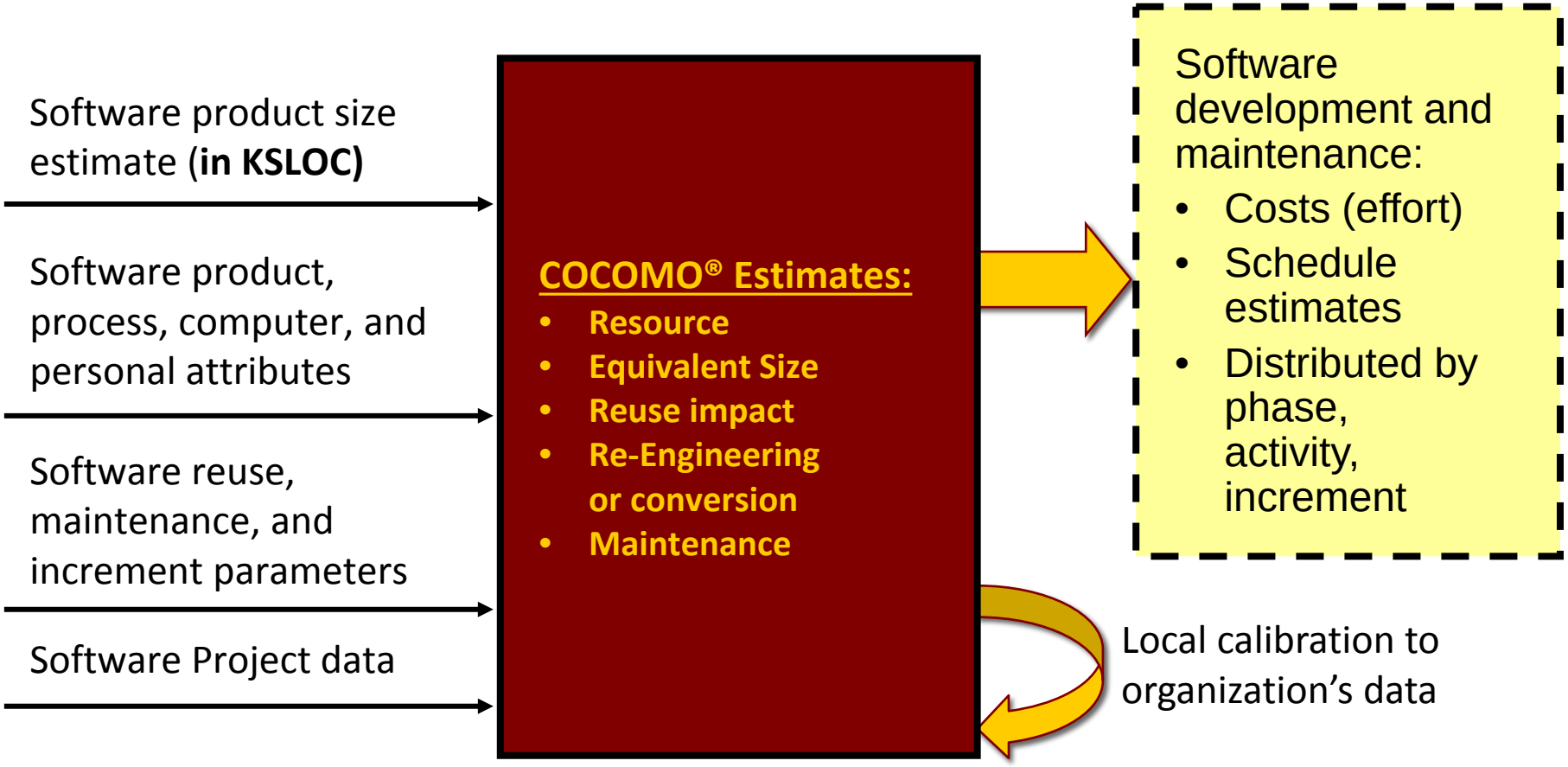
Until the community as a whole commits to better data management, the true power and value of data science will never be realized.



Calibrating COCOMO® II for Functional Size Metrics

ANANDI HIRA, BRAD CLARK, BARRY BOEHM

COCOMO® II Model



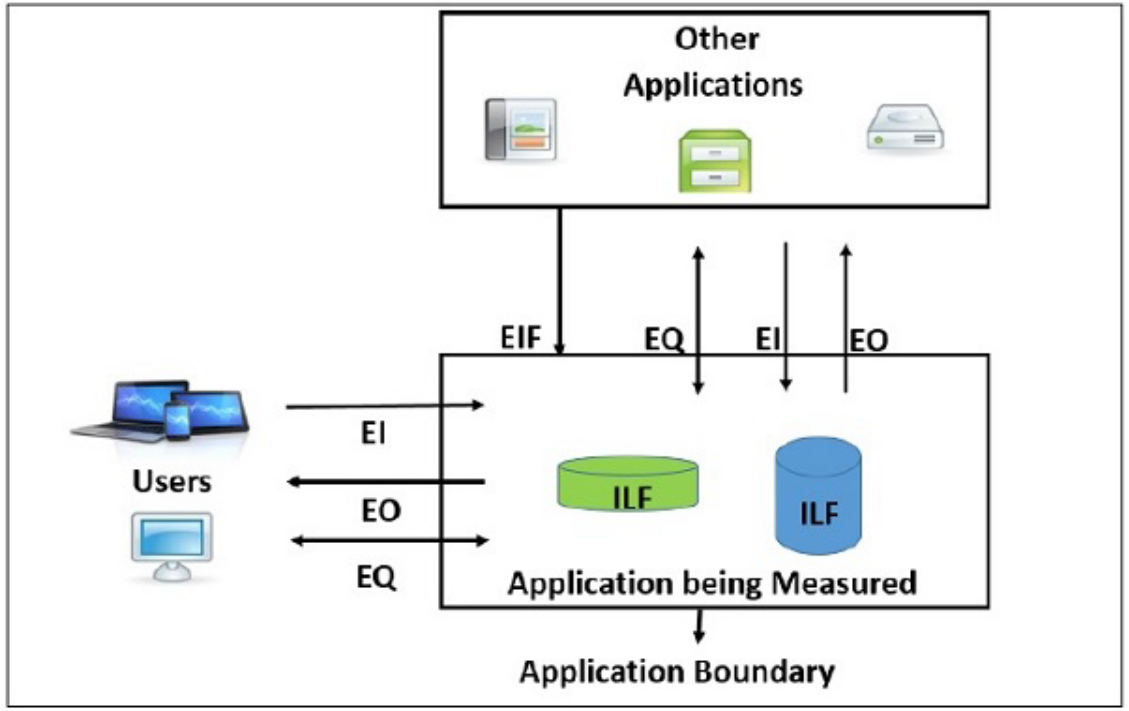
COCOMO® is an open and free model

Size Metrics' Level of Abstraction

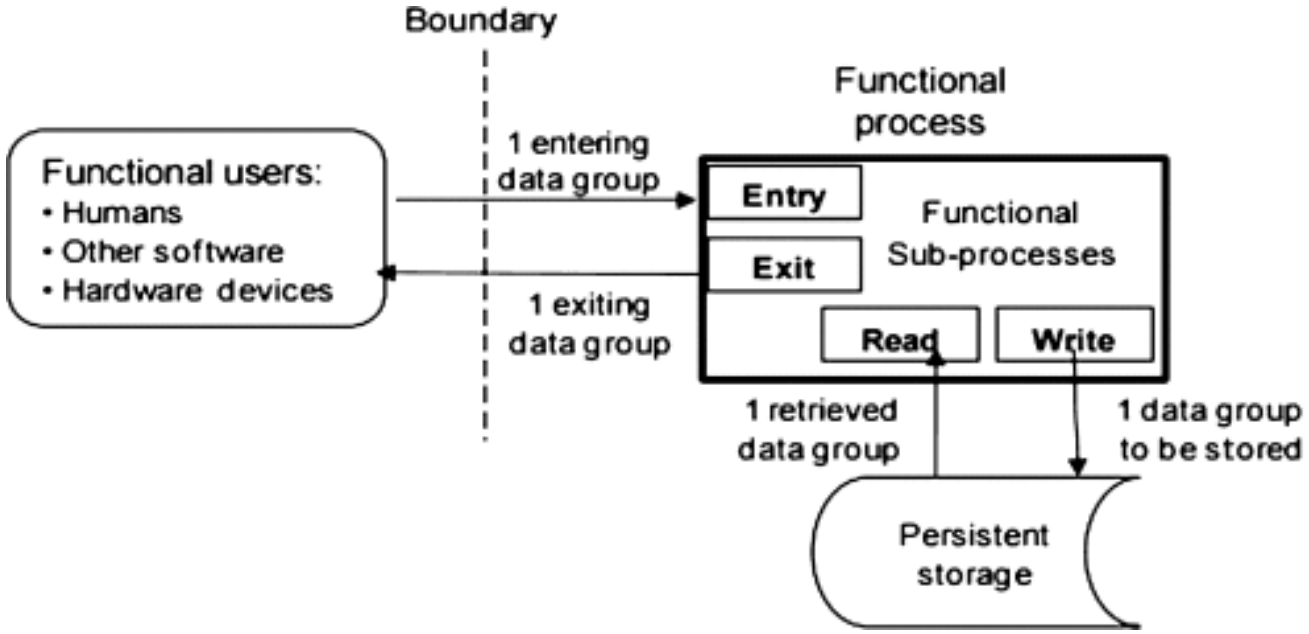
Requirement Levels	Size Metrics
Summary Goals	Story Points
User Goals	Use Cases Use Case Points (UCPs)
Sub-Functions	**IFPUG Function Points (FPs)** **COSMIC Function Points (CFPs)**
	Source Lines of Code (SLOC)

Considering 2 Prominent Functional Size Methods

IFPUG SOFTWARE MODEL



COSMIC SOFTWARE MODEL



COCOMO® II Effort Model Format

$$PM = A \times Size^{\underbrace{(B + 0.1 \times (\sum SF))}} \times \prod EM$$

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.
- Join Workshop at PSM (Practical Software and Systems Measurement) Users' Group Workshop.
  - **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

QINETIQ

The QinetiQ Team

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:

- More than 30 publications in National, International, Scientific Journal papers
- 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

We are QinetiQ

5

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

12

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

3 secs

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

85+

locations woldwide

1,300+

patents (including 300+ pending)

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

£833m

FY2018 revenue

6,000+

people with unique science and engineering expertise

1,850km

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

QINETIQ

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

Unclassified-QinetiQ Proprietary

Software Obsolescence

Research Aim

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"

Unclassified-QinetiQ Proprietary

COMMERCIAL IN CONFIDENCE

Software Obsolescence

Definitions

QINETIQ

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)

Unclassified-QinetiQ Proprietary

Software Obsolescence vs Software Maintenance

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

COMMERCIAL IN CONFIDENCE

Software Obsolescence

Introduction

- Text

QINETIQ

Introduction

The need for a Software Obsolescence Cost Analysis Framework

Unclassified-QinetiQ Proprietary

COMMERCIAL IN CONFIDENCE

Software Obsolescence Cost Analysis Framework

- Text

QINETIQ

Software Obsolescence Cost Analysis Framework

Following process was undertaken to develop the Framework

Unclassified-QinetiQ Proprietary

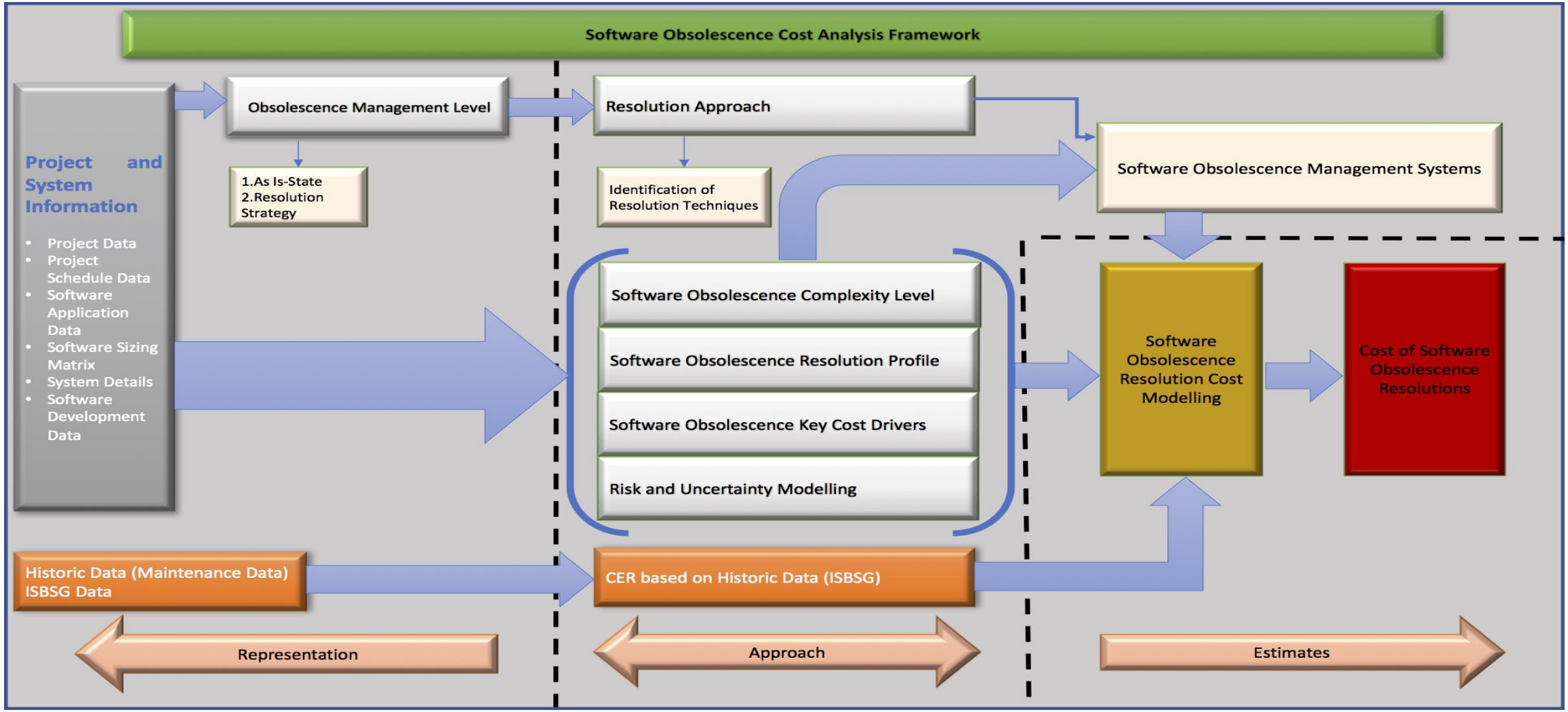
Software Obsolescence Cost Analysis Framework

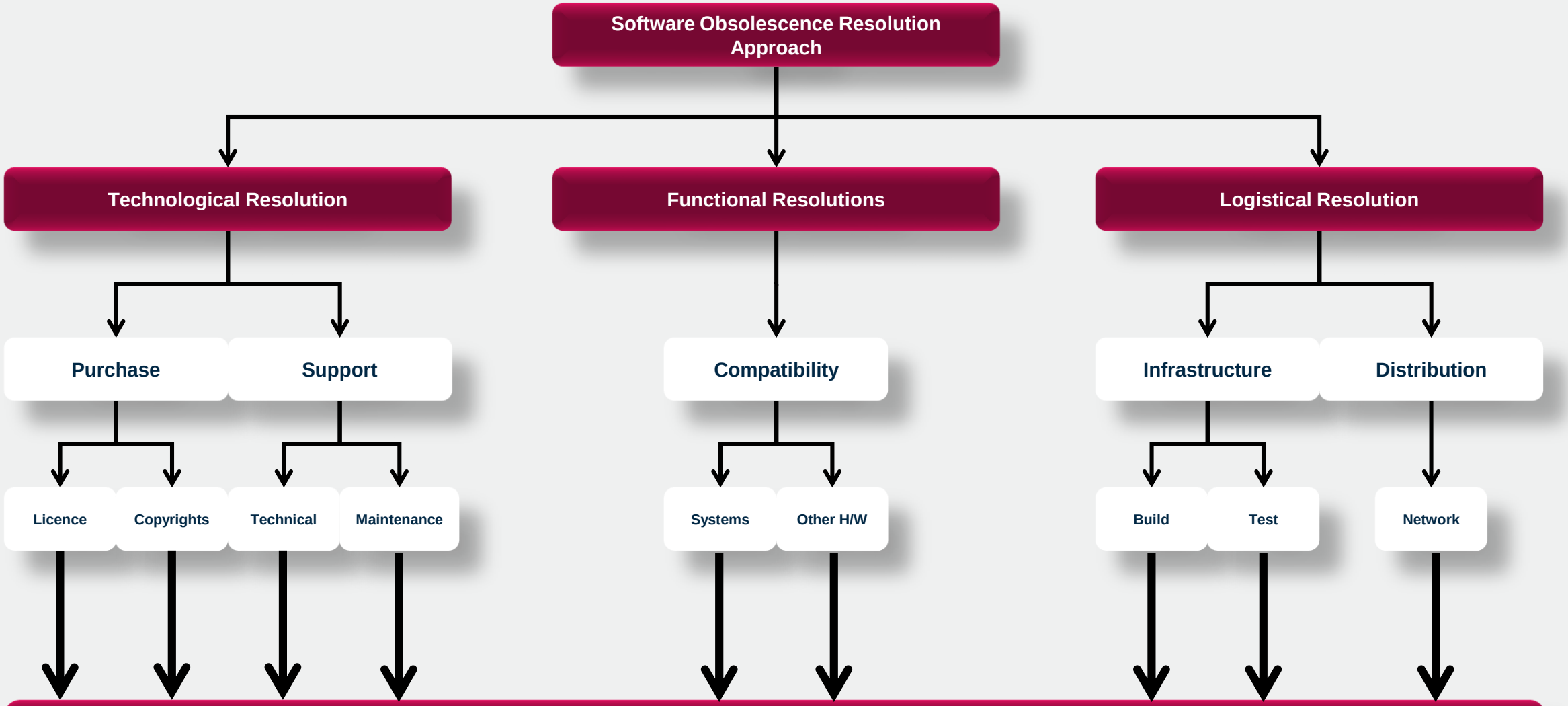
The framework has the following attributes

- 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.

Unclassified-QinetiQ Proprietary

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

QINETIQ/EMEA/CIT/CP1901699| ©QinetiQ

Software Obsolescence Management Level

Reactive----------Transitions----------Proactive

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

QINETIQ/EMEA/CIT/CP1901699 © QinetiQ

Software Obsolescence Complexity Level

Proactive ---------- Reactive

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

High Risk Software ---------- Low Risk Software

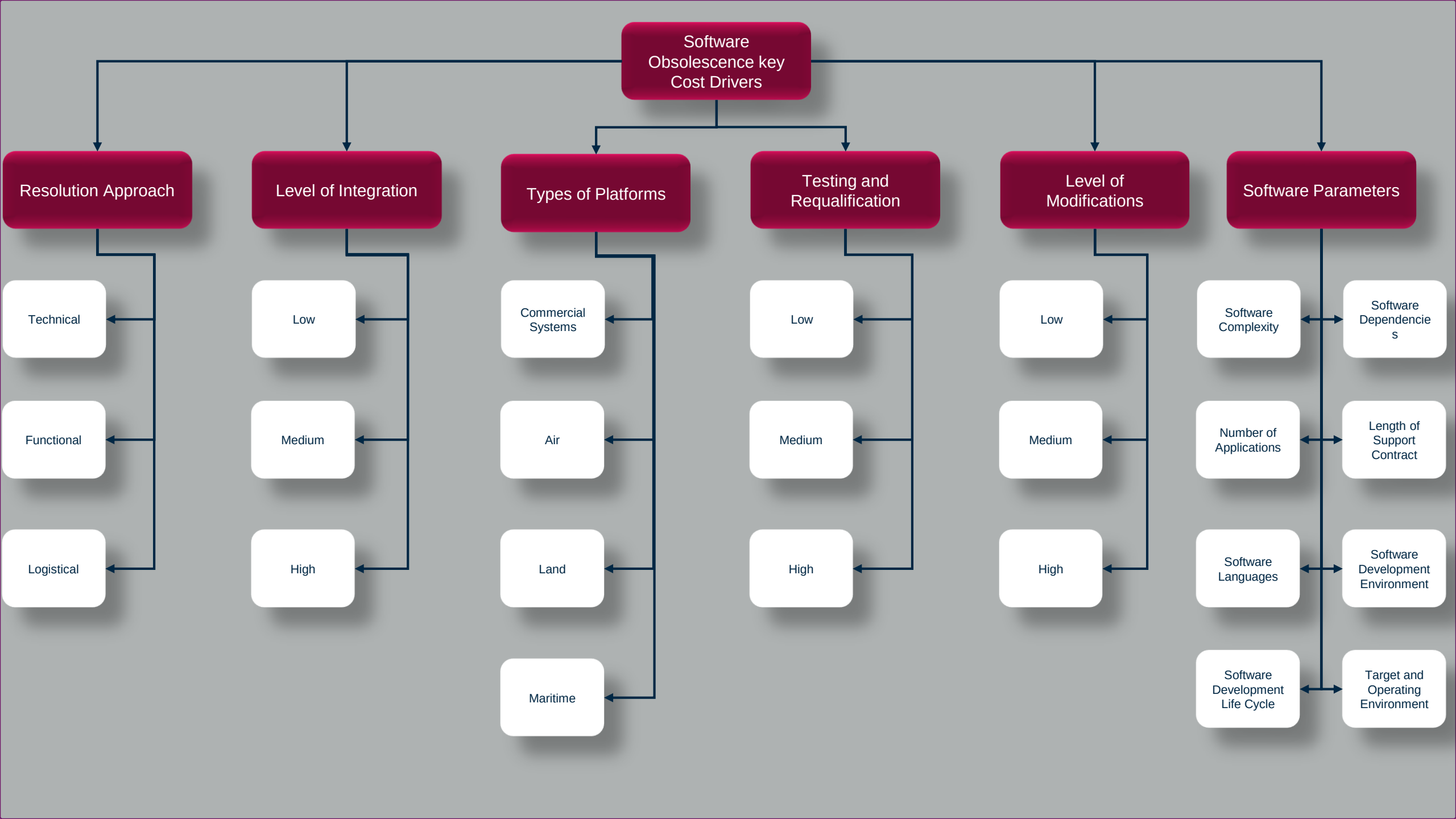
QINETIQ/EMEA/CIT/CP1901699| ©QinetiQ

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

COMMERCIAL IN CONFIDENCE


Software Obsolescence key Cost Drivers
Resolution Approach
Level of Integration
Types of Platforms
Testing and Requalification
Level of Modifications
Software Parameters
Technical
Functional
Logistical
Low
Medium
High
Commercial Systems
Air
Land
Maritime
Low
Medium
High
Low
Medium
High
Software Complexity
Software Dependencies
Number of Applications
Length of Support Contract
Software Languages
Software Development Environment
Software Development Life Cycle
Target and Operating Environment

QINETIQ PROPRIETARY
COMMERCIAL IN CONFIDENCE

Software Obsolescence and Maintenance

QINETIQ

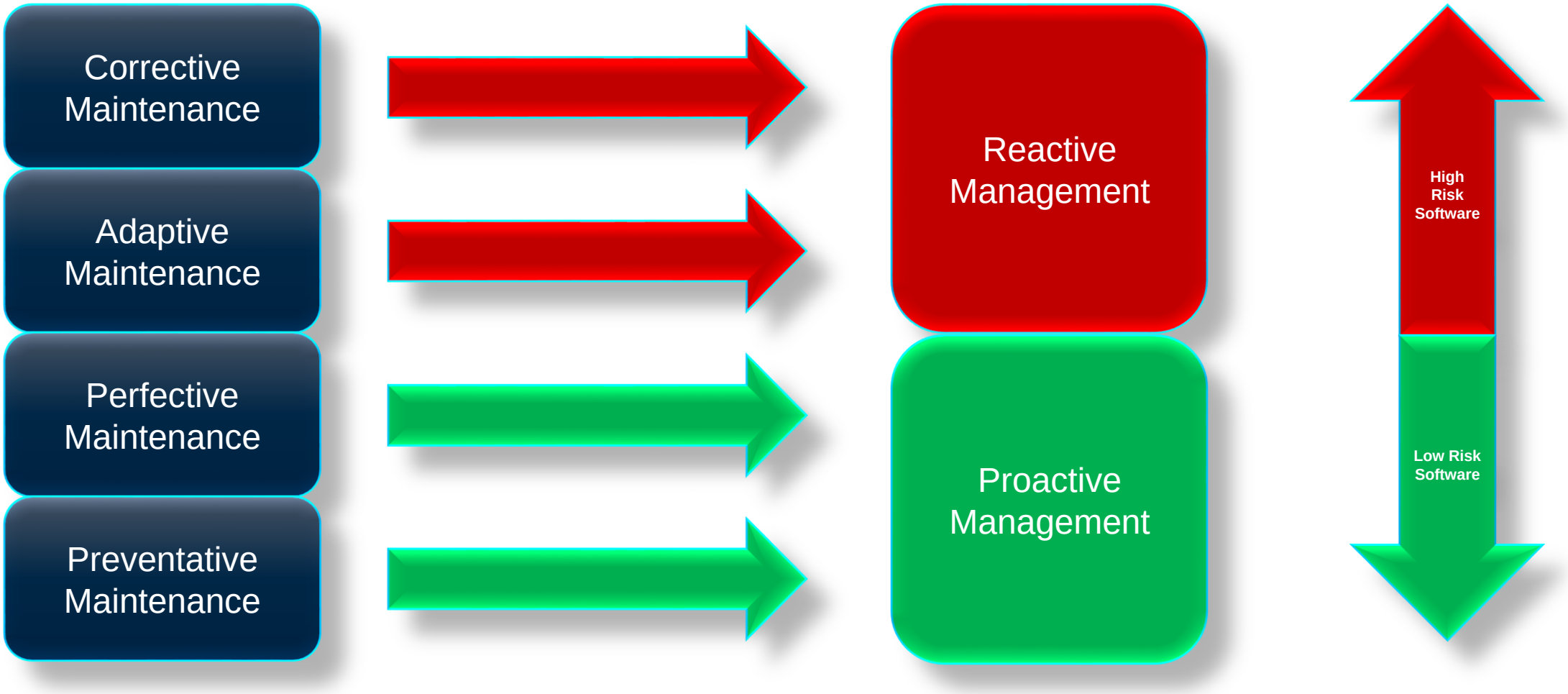
Types of Software Maintenance

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

COMMERCIAL IN CONFIDENCE

Maintenance Vs Obsolescence



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.

COMMERCIAL IN CONFIDENCE

COMMERCIAL IN CONFIDENCE

Software Obsolescence Resolution Cost Optimisation Model (SRCOM)

- Text

Unclassified-QinetiQ Proprietary

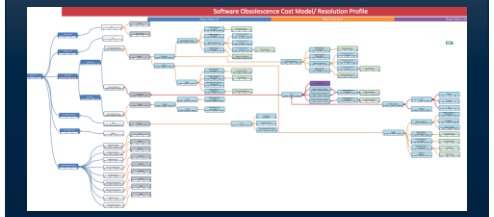
QINETIQ

Integration of all the Models

Software Obsolescence Data Collection Form

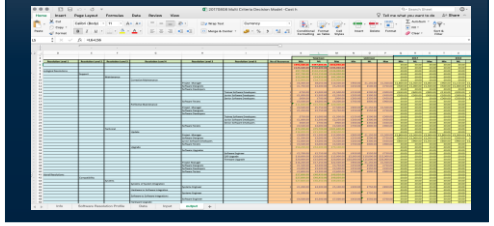
+

Software Obsolescence Resolution Model



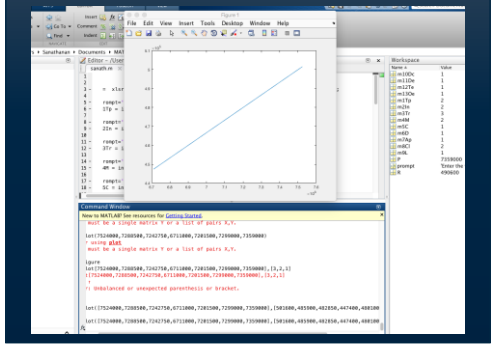
+

Software Obsolescence Resolution Cost Model



+

Software Obsolescence Optimization Model



=

Optimized Through Life Resolution Profile with Cost

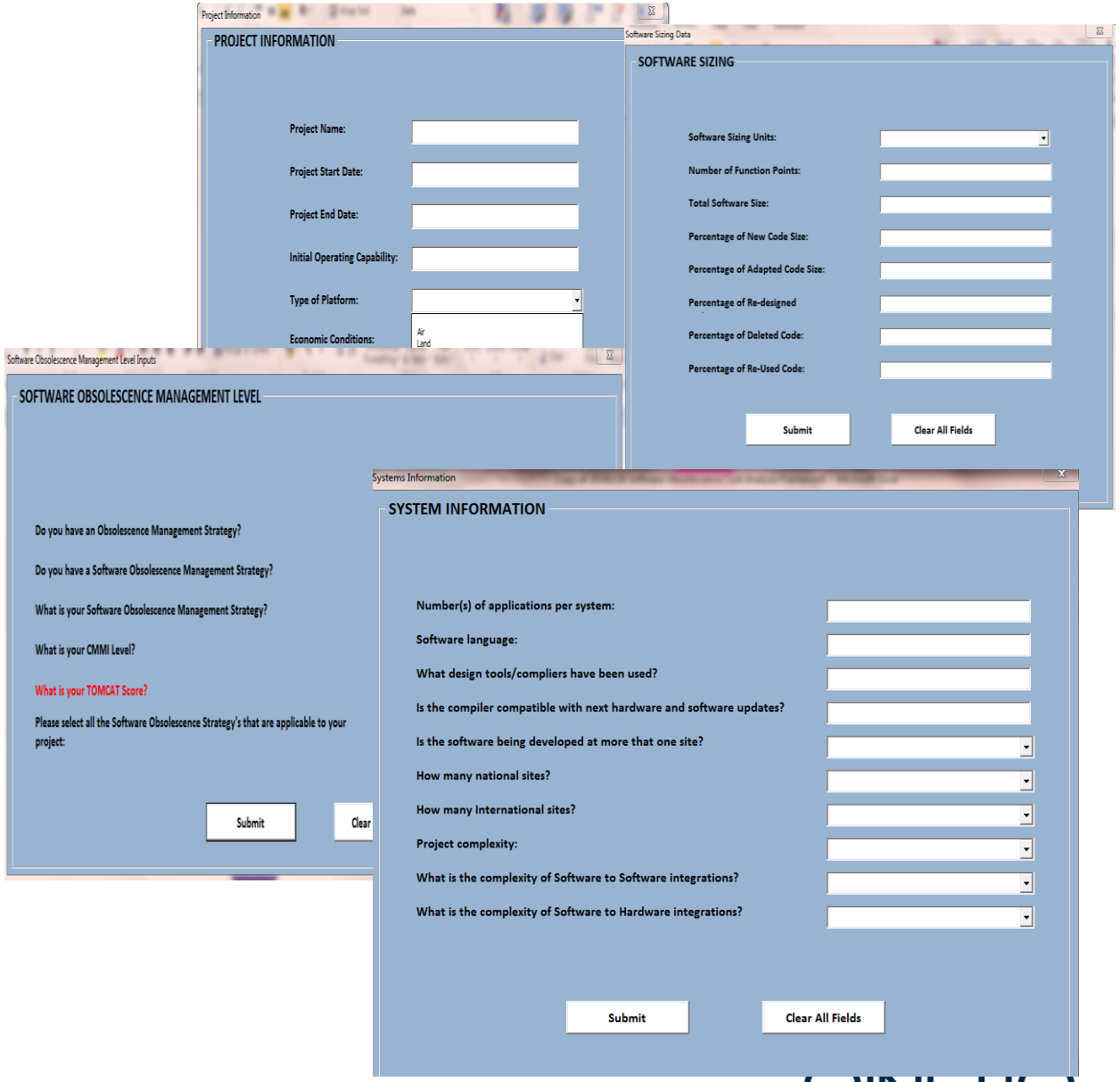
COMMERCIAL IN CONFIDENCE

QINETIQ

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

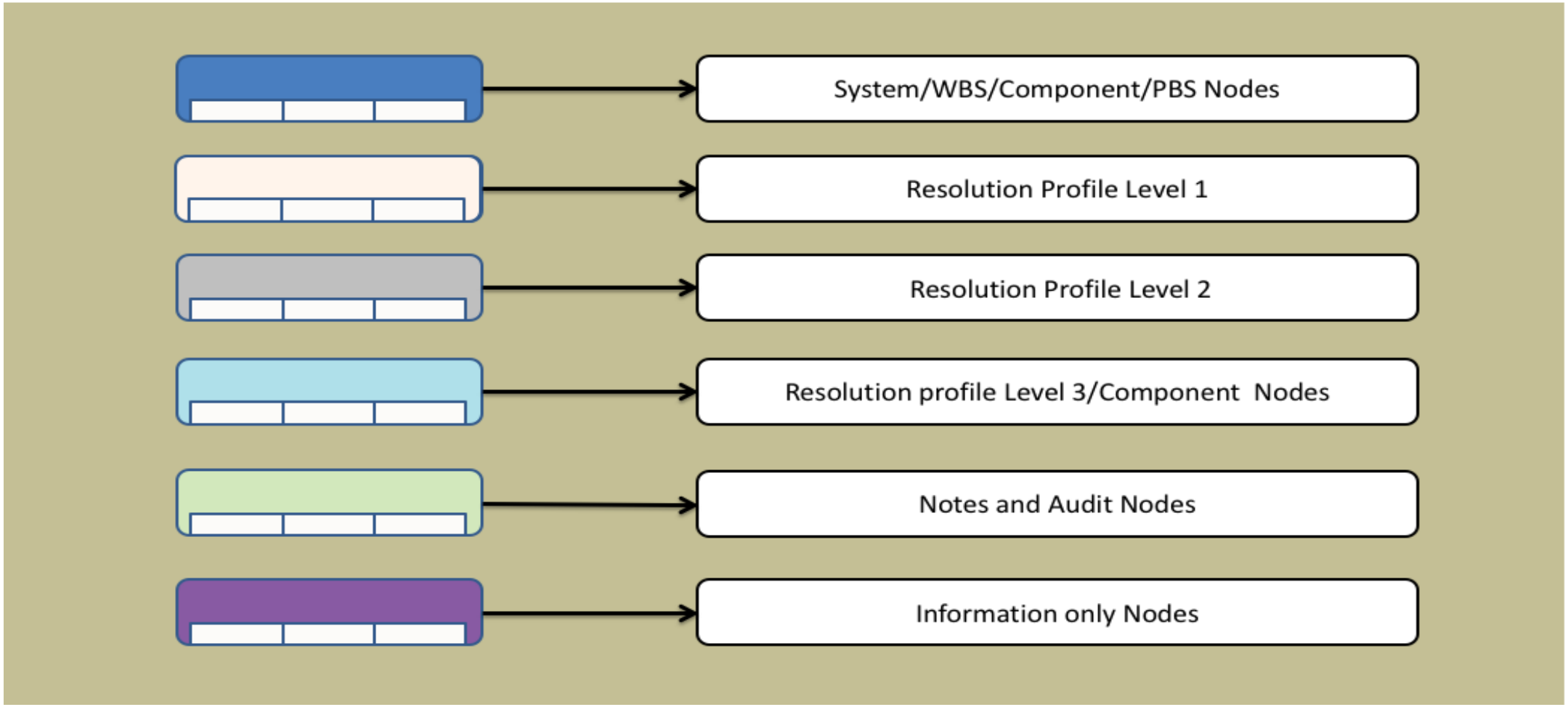


COMMERCIAL IN CONFIDENCE

Software Obsolescence Resolution Cost Optimisation Model (SRCOM)

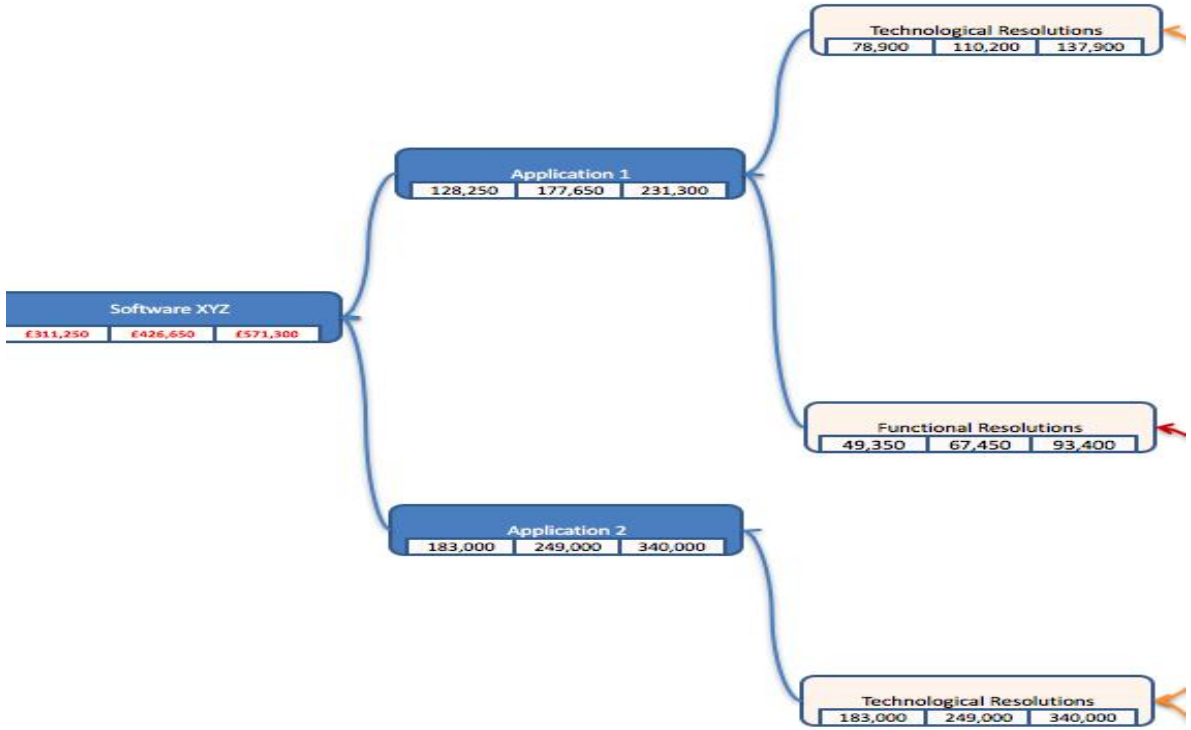
- This model will recommend the best resolution approach based on the cost
- Optimisation is undertaken using MatLab Optimisation toolbox
- This model is at a very early stages of its development and
- This methodology involve diagrammatical representation of the cost
- Diagram represents decompositions 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



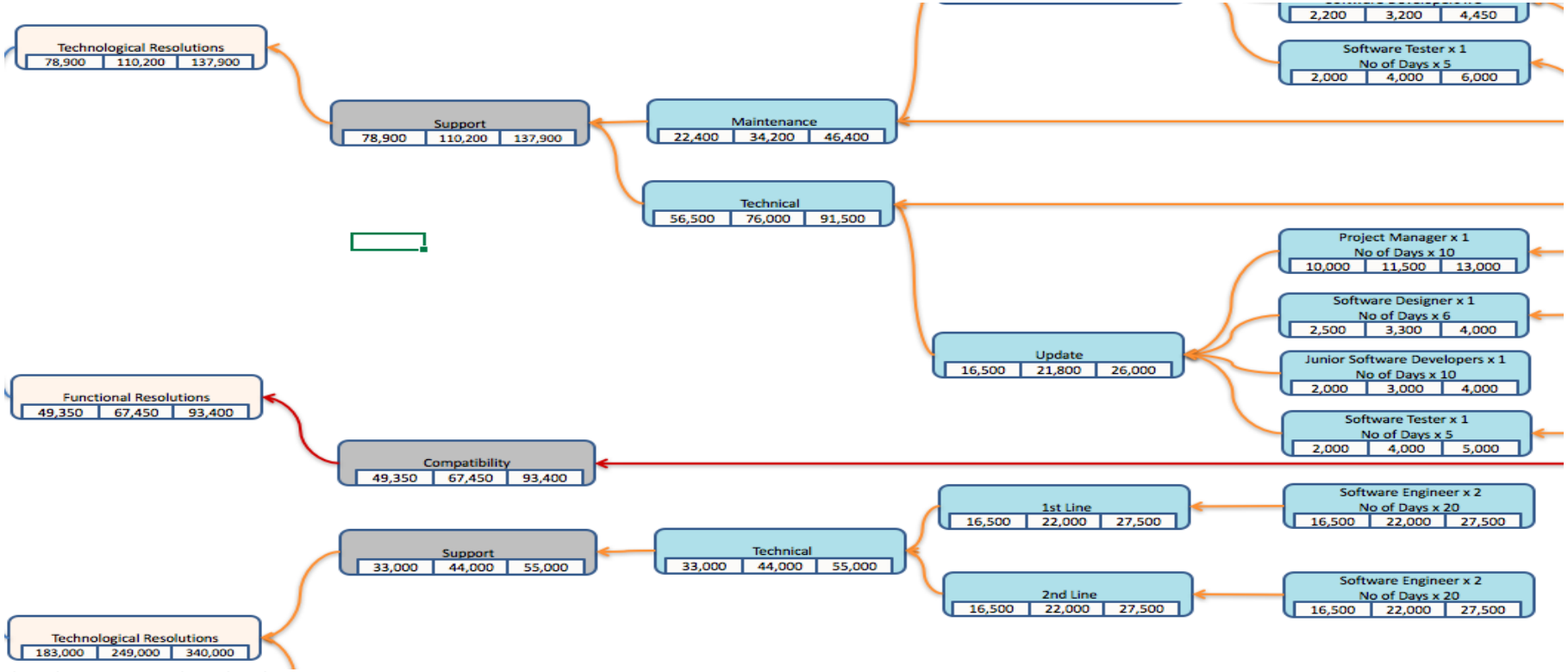
Software Obsolescence Resolution Cost Optimisation Model - Nodes

- 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.

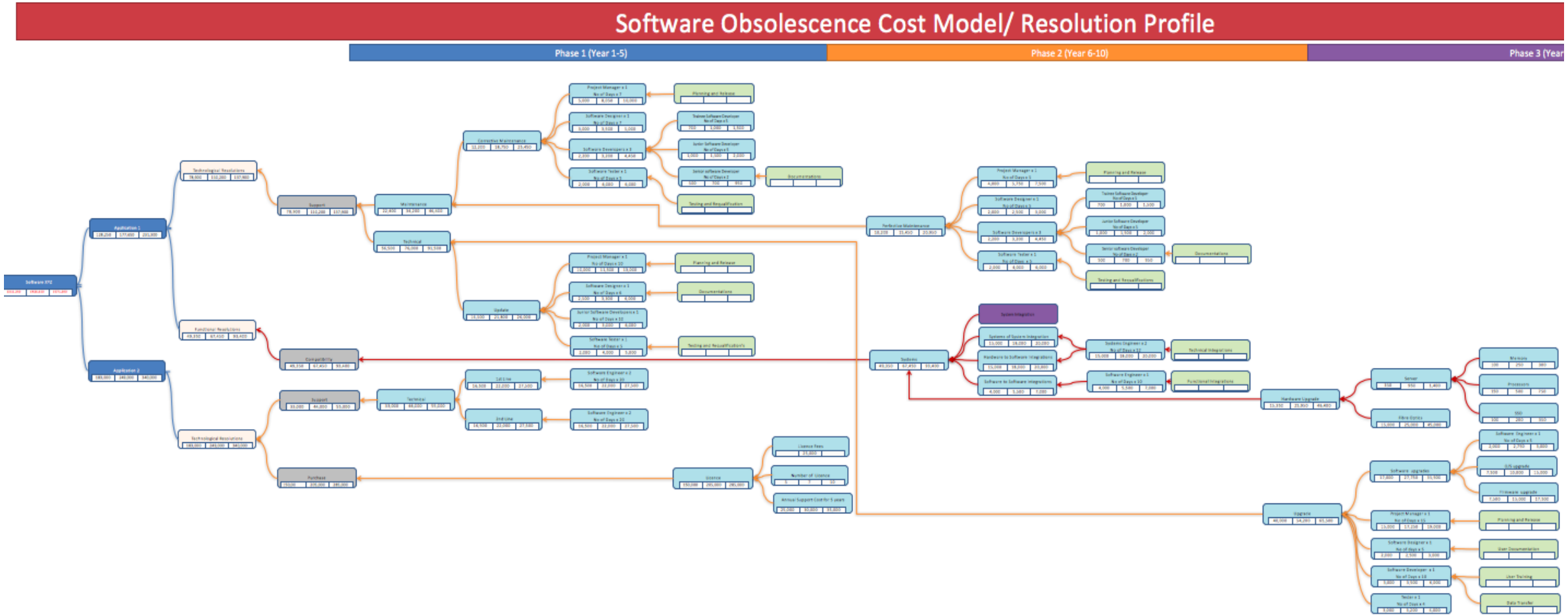


Software Obsolescence Resolution Cost Optimisation Model - Nodes

- 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

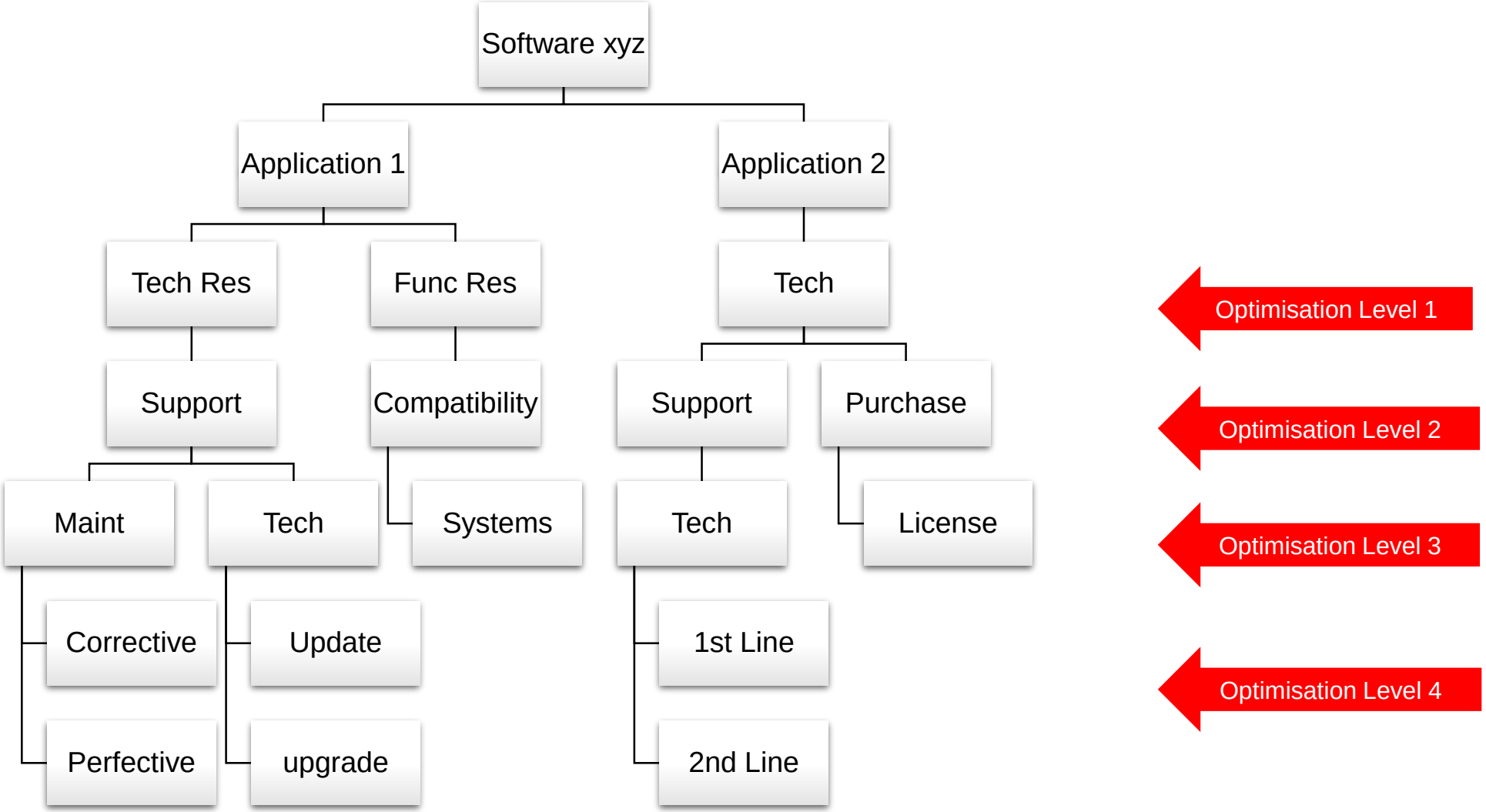


Optimisation –Test Case

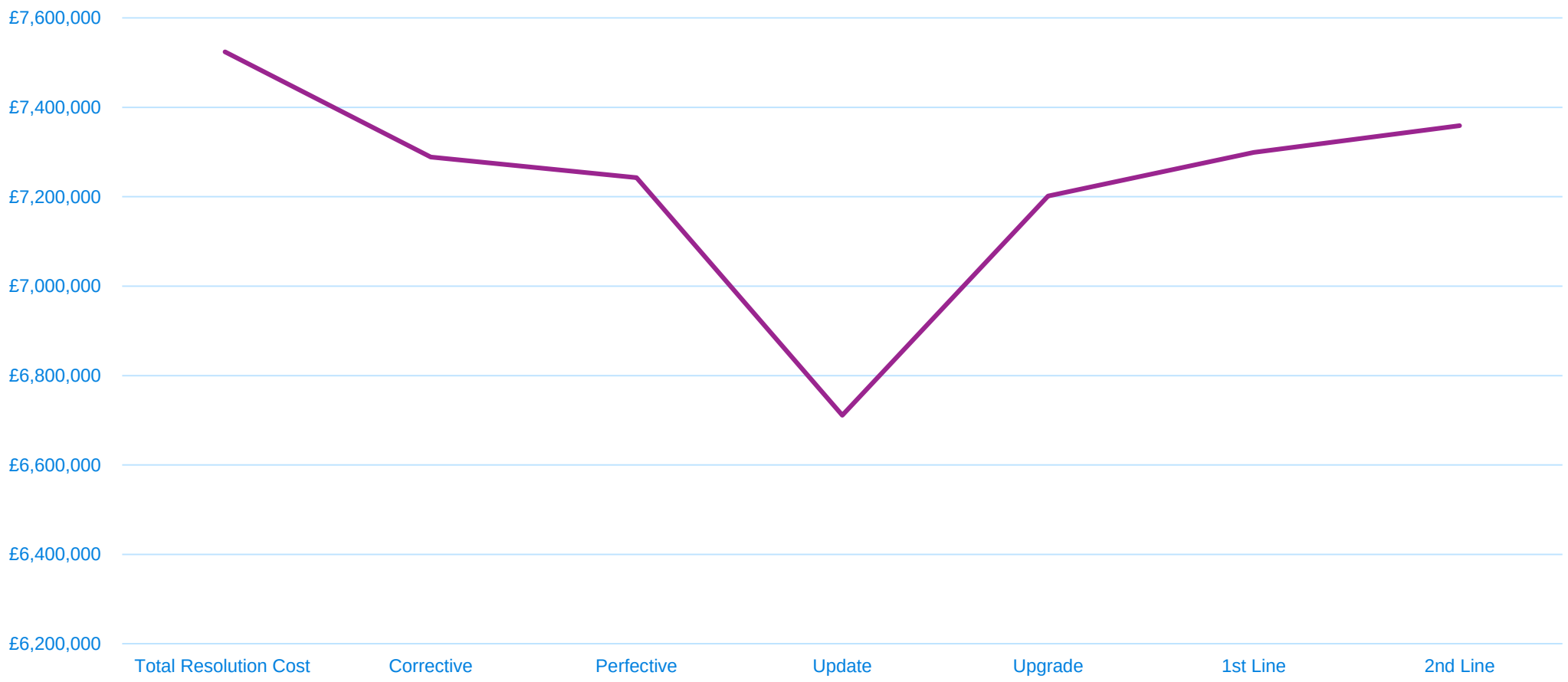
Discription		Values
Type of Platform	Air	2
Level of Integration	Medium	2
Testing and Requalification	High	3
Level of Modification	Medium	2
Software Complexity	Low	1
Software Dependency	Low	1
Number of Applications	2	1
Software Contract Length	6-10 years	2
Software Language	2nd Generation	1
Software Development Cycle	Stable	1
Software Development environment	Stable	1
Target Environment	Stable	1
Operating environment	stable	1

Unclassified-QinetiQ Proprietary

Optimisation –Test Case (Resolution Strategies)



Optimisation –Results



Unclassified-QinetiQ Proprietary

Summary

Software Obsolescence is a an emerging issues and it is important to understand how much Software Obsolescence is going to cost at a very early stages of 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.

Unclassified-QinetiQ Proprietary

QinetiQ
Building 240
The Close
Bristol Business Park
Coldharbour Lane
Bristol BS16 1FJ
United Kingdom

Tel +44 (0)117 3172558
Mobile +44 (0)738 237 044
srajagopal@QinetiQ.com
www.QinetiQ.com

Sanathanan Rajagopal TMIET
QinetiQ Fellow
Estimating Manager –CiT

Thank you –Any Questions ?

Unclassified-QinetiQ Proprietary

Army Software Sustainment Cost Estimating Results

DASA-CE

IT CAST

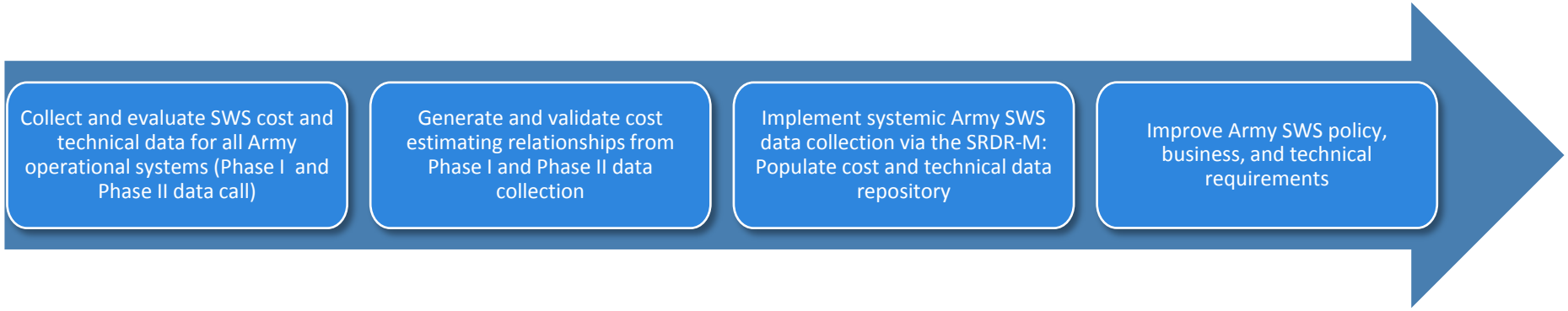
August 2019

Unclassified
Distribution Statement A: Approved for Public Release Distribution is Unlimited

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



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

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

System Specific

1.0 Software Change Product

1.1 Capability Changes
- Change requirements
- Change development
- B/L Integration & Test
- IV&V

1.2 IAVA Changes

System/Non-System Specific

3.0 Software Licenses

License - Right to Use
- COTS/GOTS
- NDI
- Open Source
- Other

License - Maintenance

System/Non-System Specific

5.0 System Facilities

Hardware
- Software development assets/workstations
- System integration & test facilities
- Test equipment - tools

Facility Operations

System/Non-System Specific

7.0 Field Software Eng.

On-Site technical assistance
Problem Troubleshooting
S/W Installation
Operational Assistance
On-site Training

System Specific

2.0 Project Management

Planning
Execution management
Configuration management
Resource & team management
Contracting management
Measurement - reporting

System Specific

4.0 Certification & Accreditation

Security
Safety
Networthiness
Airworthiness

System Specific

6.0 Sustaining Engineering

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

Non-System Specific

8.0 Other

Operations
Organization management
Personnel management
Financial management
Information management
Process management
Change management

Version 5.0

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

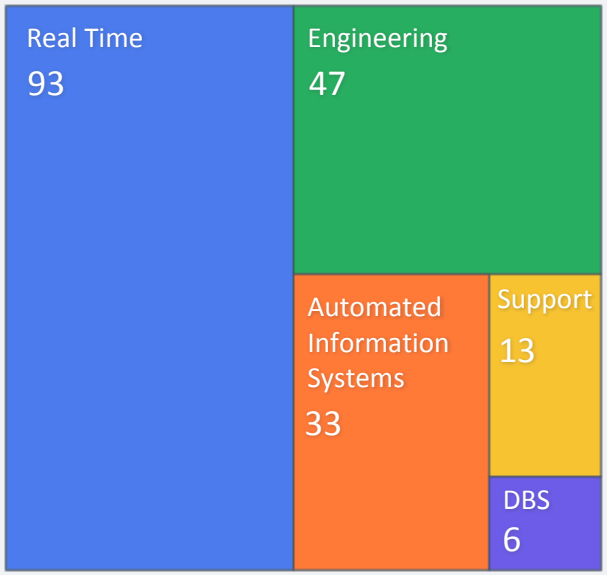
192 Systems

1,040 Releases

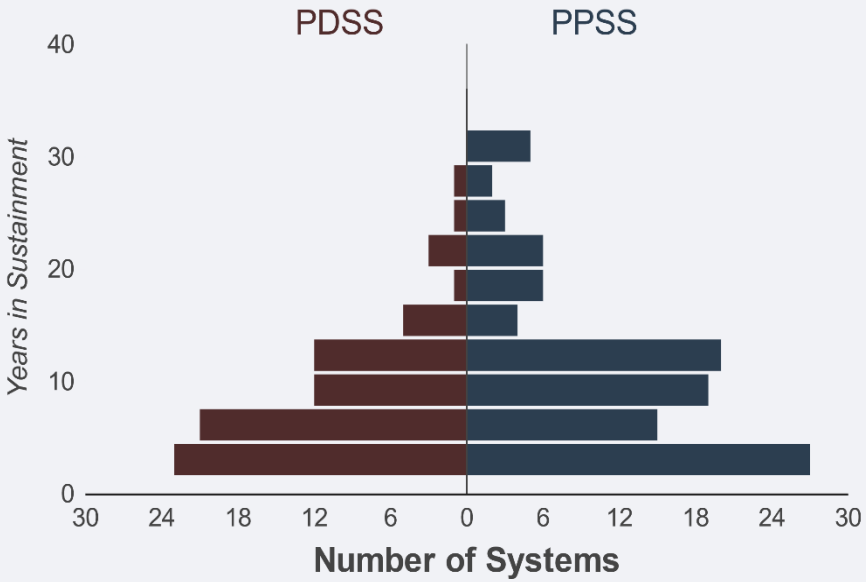
3,434 License Records

411K Data Fields

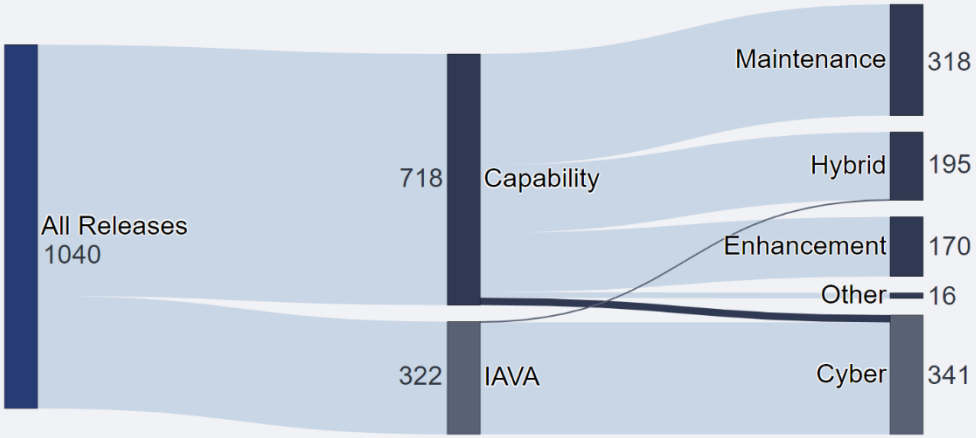
Systems by Super Domain



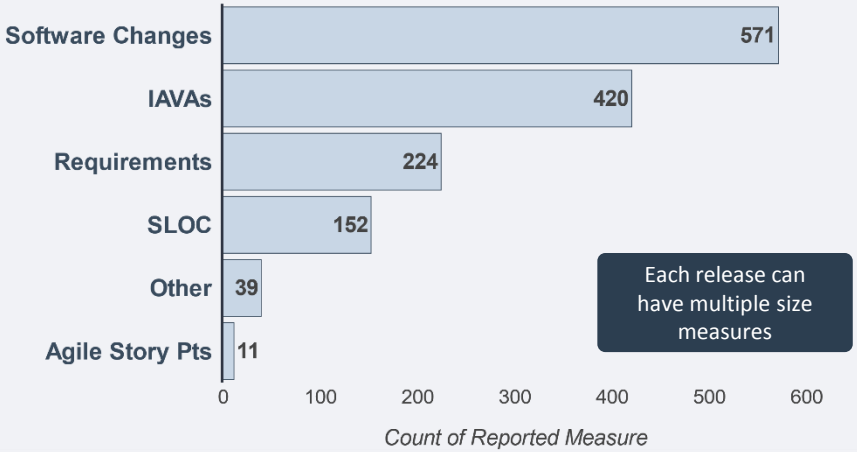
Distribution of System Age



Releases by Change Type

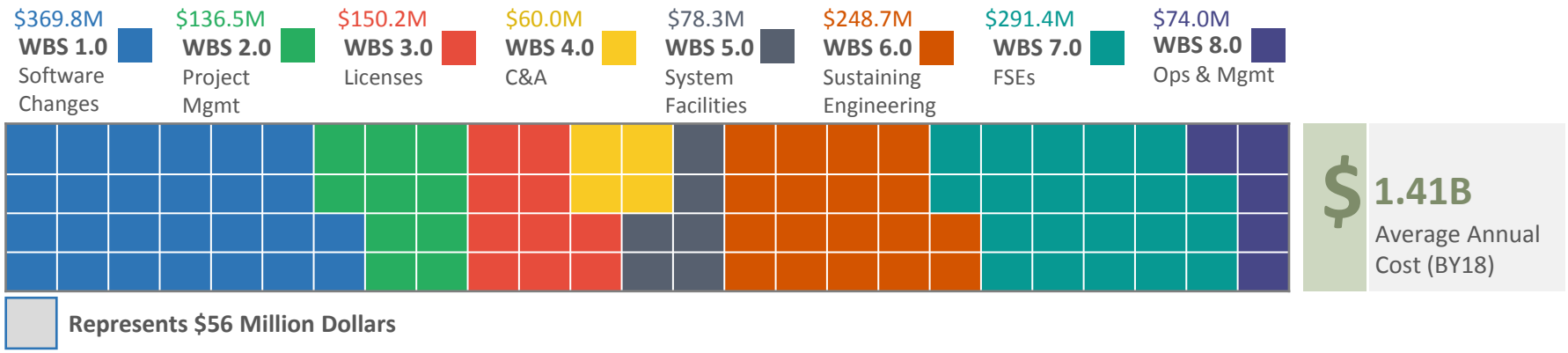


Releases by Size Measure

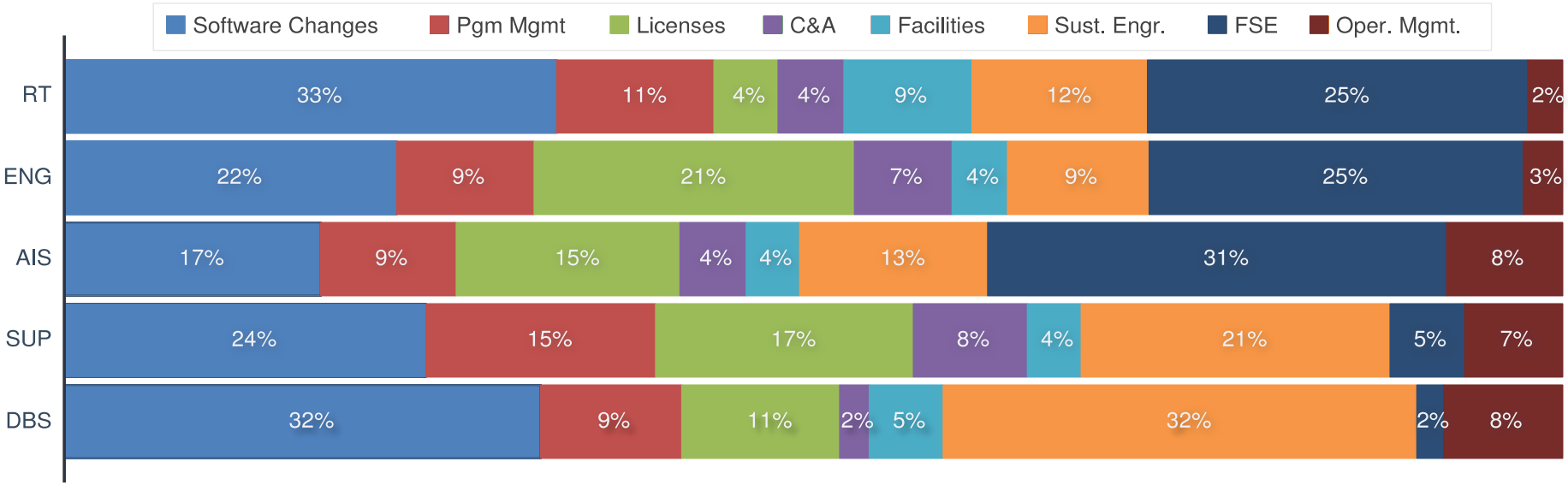


Annual Cost Distribution

Average Annual Cost by WBS & Cost Allocation by WBS



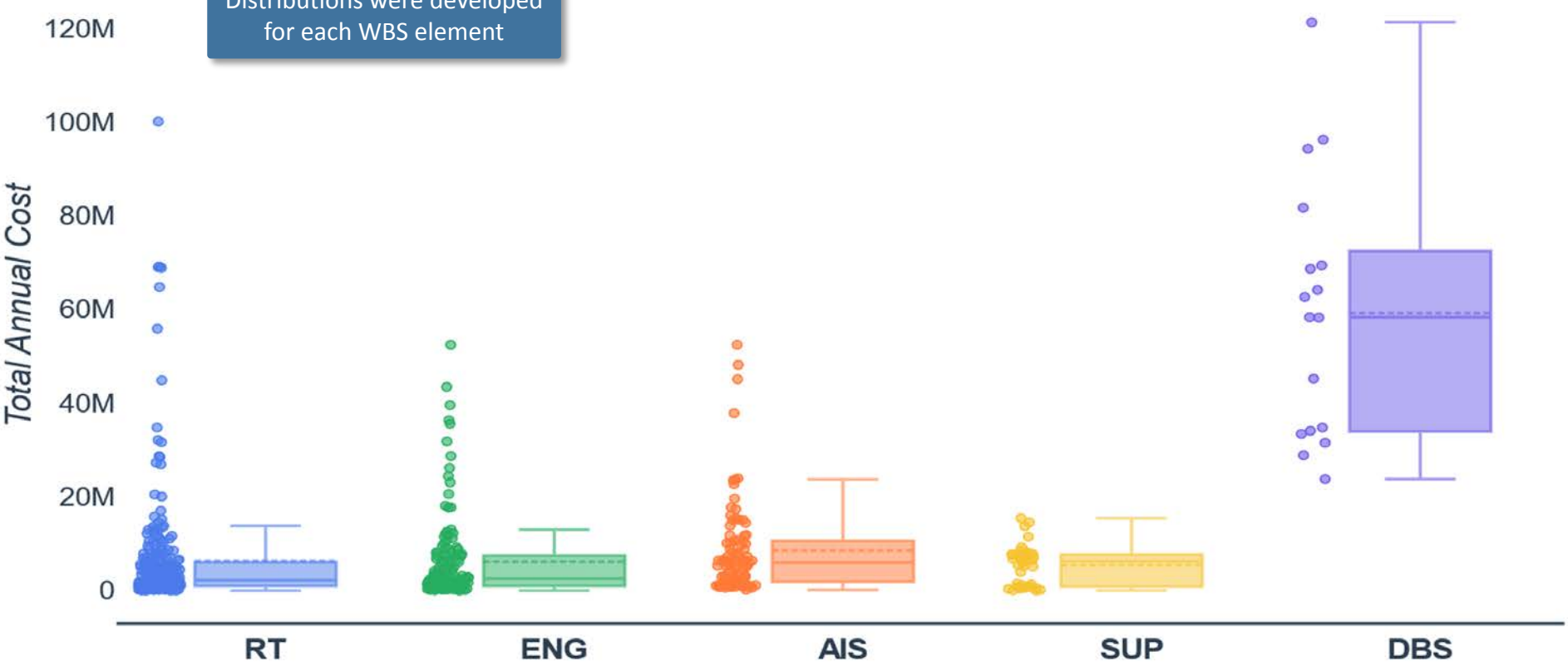
Annual Cost Allocation by WBS



Total Annual Cost Distribution

By Super Domain (BY18$)

Distributions were developed for each WBS element

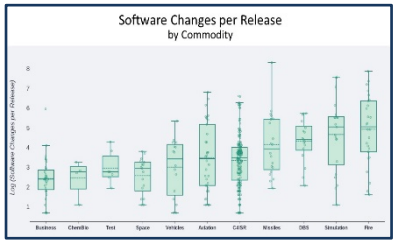


Super Domain	RT	ENG	AIS	SUP	DBS
Count*	240	136	91	40	17
Mean	$6,351,776	$6,202,735	$8,572,860	$5,411,627	$59,221,413
Median	$2,237,870	$2,547,140	$5,972,457	$6,285,354	$58,334,452

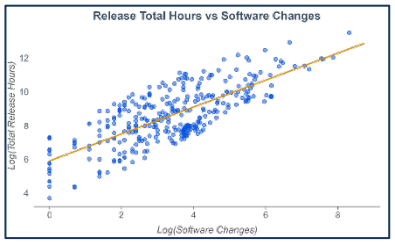
* Up to 3 FYs per system

Specific Analysis Overview
(full detail is available)

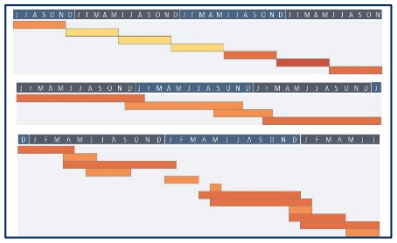
- Benchmarks for capability releases by super domain and commodity
  - # of software changes/release
  - Hours per software change
  - By WBS
  - PDSS vs. PPSS for application super-domains
  - By sustaining organization
  - DSLOC per FTE



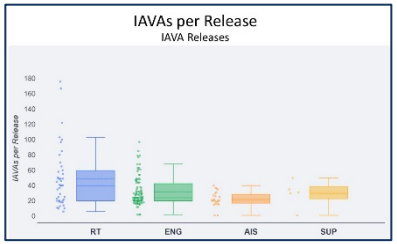
- Cost estimating relationships (CERs) for capability releases
  - Evaluated meta data for impacts on CERs: commodity, change type, # of inter-service partners, and ACAT levels had an impact
  - Data Trimmed and CERs developed
    - CERs for Software Changes (most effective), Requirements, and Lines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



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



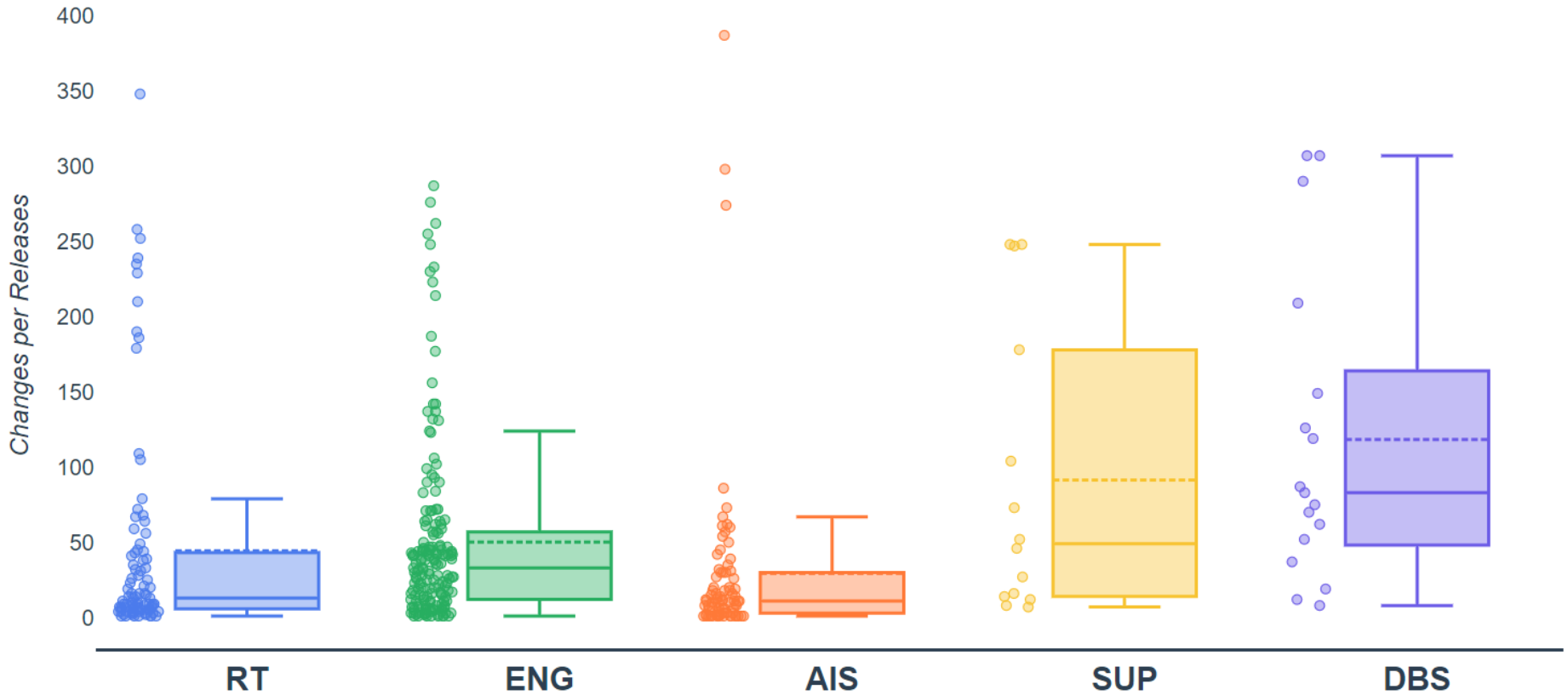
- IAVA Release Analysis
  - Most data is Level-of-Effort
  - Data best described by median by grouping for # of IAVAs and hours per IAVA



Unclassified
Distribution Statement A: Approved for Public Release Distribution is Unlimited

Benchmarks for Capability Releases

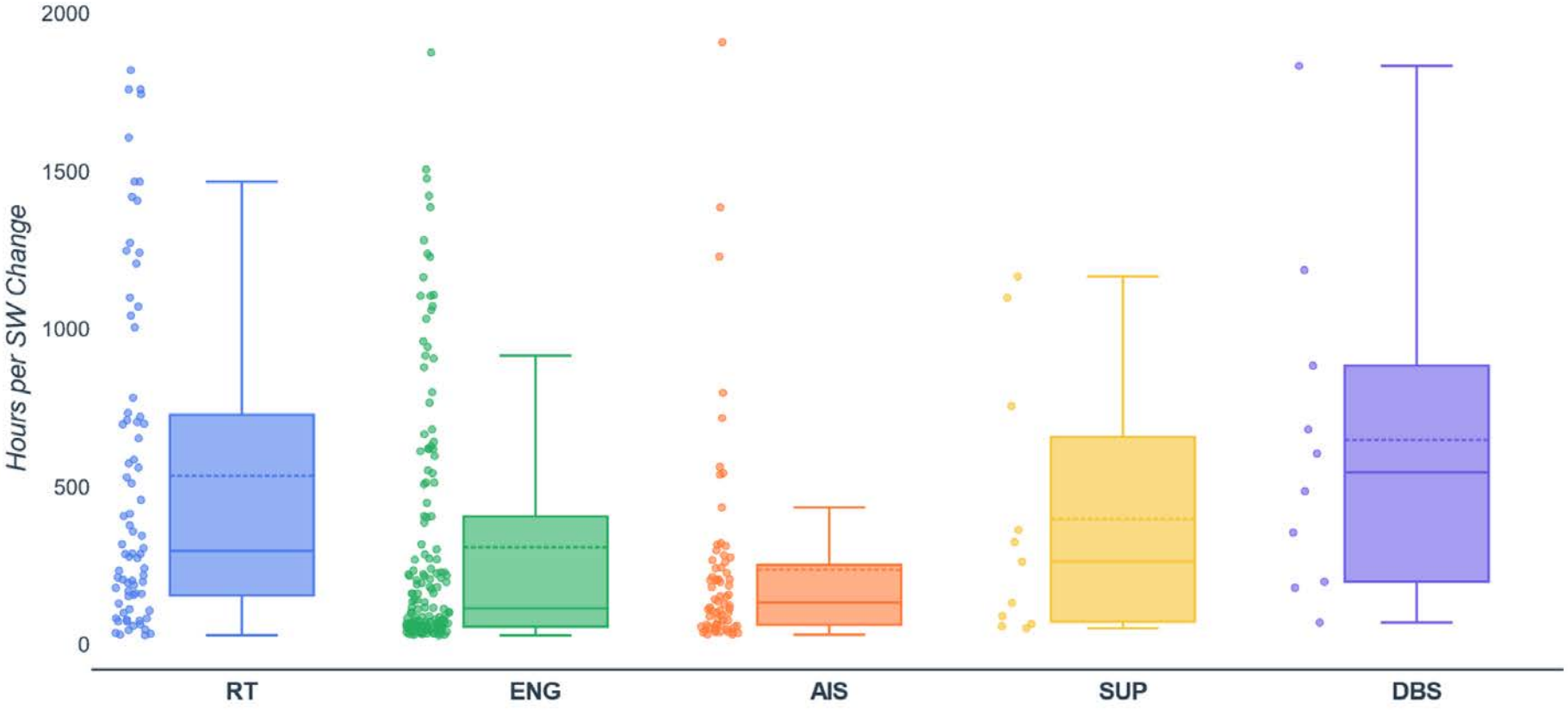
Number of Software Changes per Release
By Super Domain



Super Domain	Count	Mean	Median
RT	89	44	13
ENG	171	50	33
AIS	79	29	11
SUP	14	91	49
DBS	17	118	83

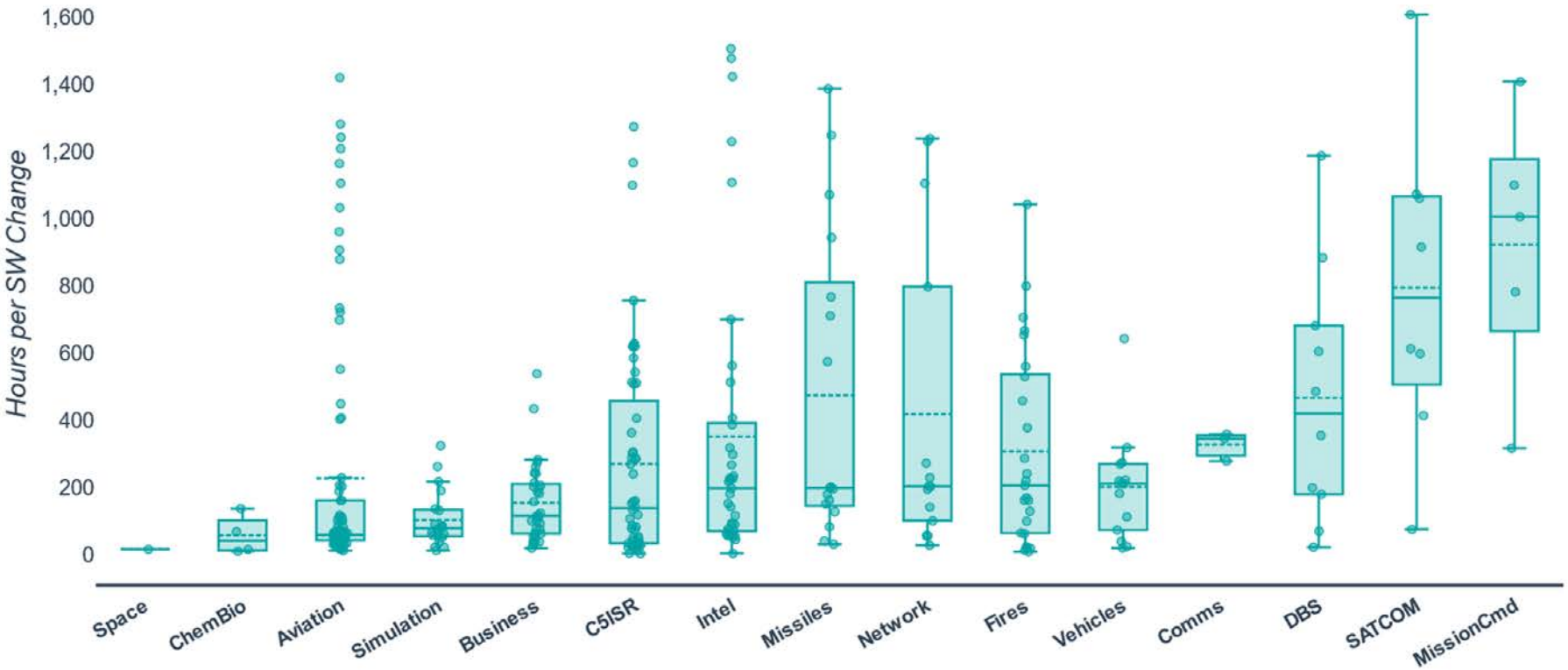
Number of SW Changes/Release can be used to size future releases when program specific data is unknown - The resulting size can be used with the associated cost benchmark or put into a CER

Hours per Software Change
By Super Domain



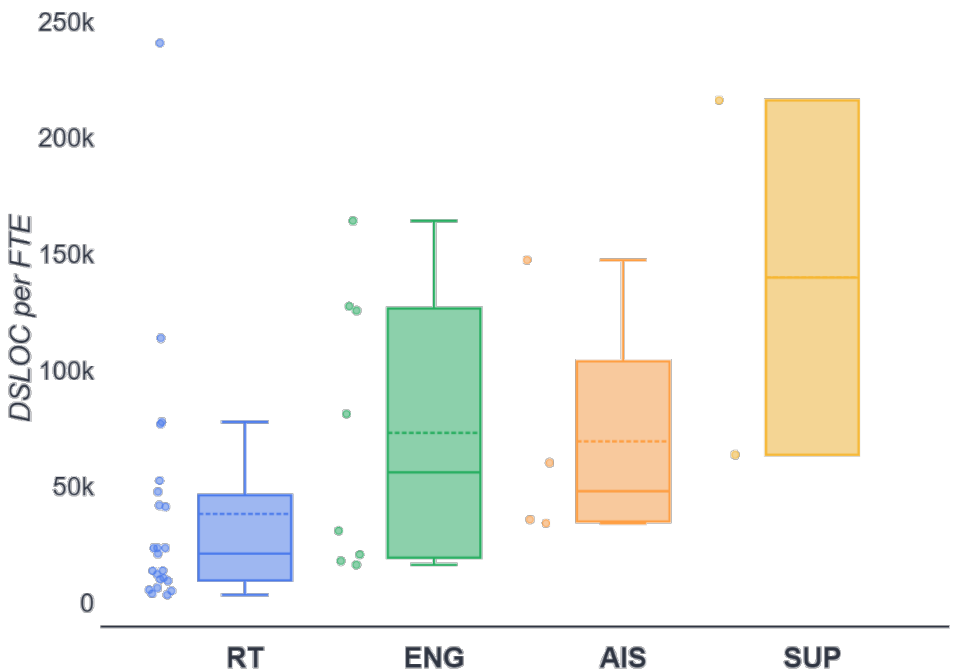
Super Domain	Count	Mean	Median
RT	101	1481.6	407.4
ENG	186	629.8	102.5
AIS	79	244.9	111.5
SUP	14	490.7	197.0

Hours per Software Change
by Commodity



	Space	ChemBio	Aviation	Simulation	Business	C5ISR	Intel	Missiles	Network	Fires	Vehicles	Comms	DBS	SATCOM	MSN CMD
Count	1	4	87	20	32	52	37	17	14	25	14	3	10	8	5
Mean	16.1	57.5	227.2	103.3	154.3	270.0	351.3	475.0	418.4	307.3	201.6	327.3	467.0	794.6	922.5
St Dev	--	58.8	360.3	84.1	118.4	312.7	432.7	456.5	459.0	291.0	160.4	42.9	377.3	471.8	406.5
Median	16.1	41.7	58.8	78.4	115.6	138.2	197.8	198.8	203.3	206.1	211.7	345.1	420.4	764.4	1005.7

DSLOC per FTE
By Super Domain



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

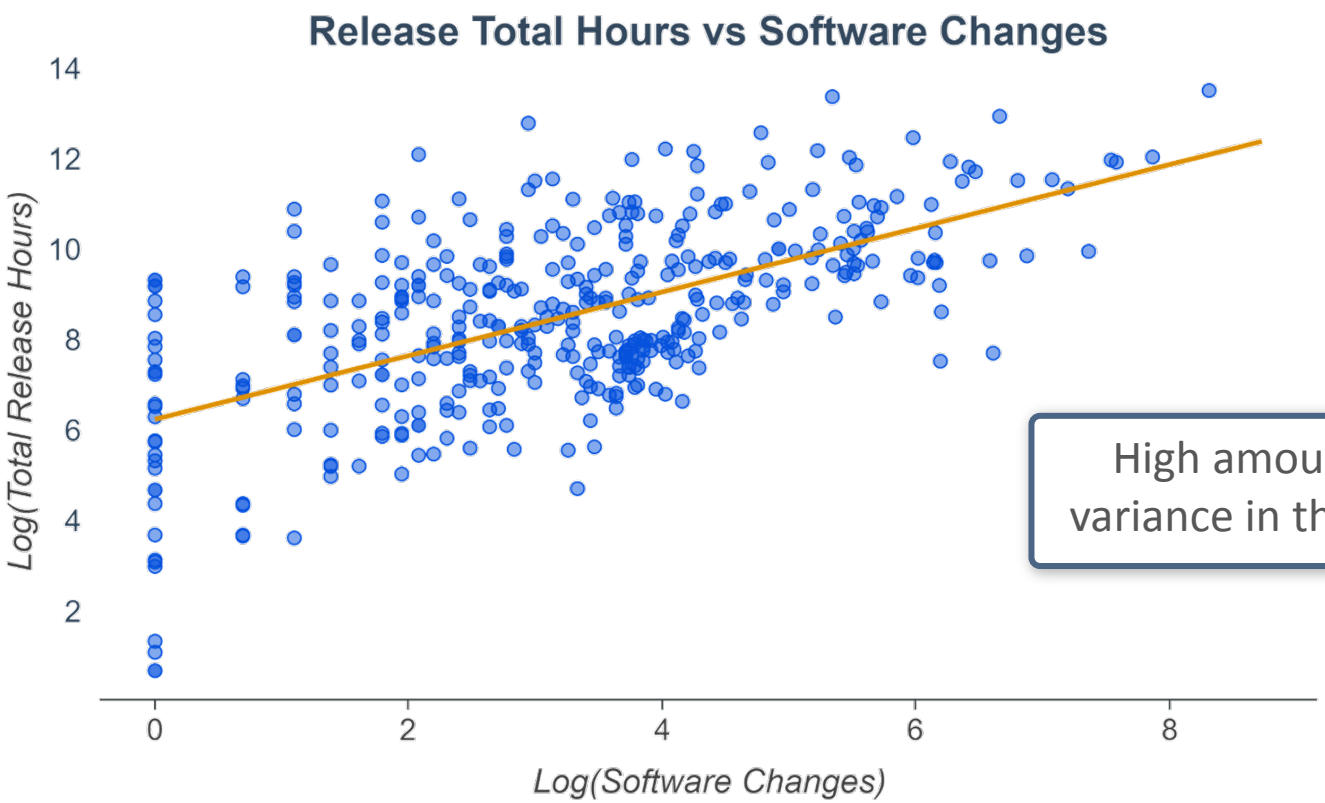
For a 100,000 DSLOC baseline, the estimated FTEs are RT=4.7, ENG=1.8, AIS=2.1 & SUP=NA

- 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

Cost and Schedule Estimating Relationships (CER/SER) Capability Releases

Unclassified
Distribution Statement A: Approved for Public Release Distribution is Unlimited

All Data CER



High amount of variance in the Data

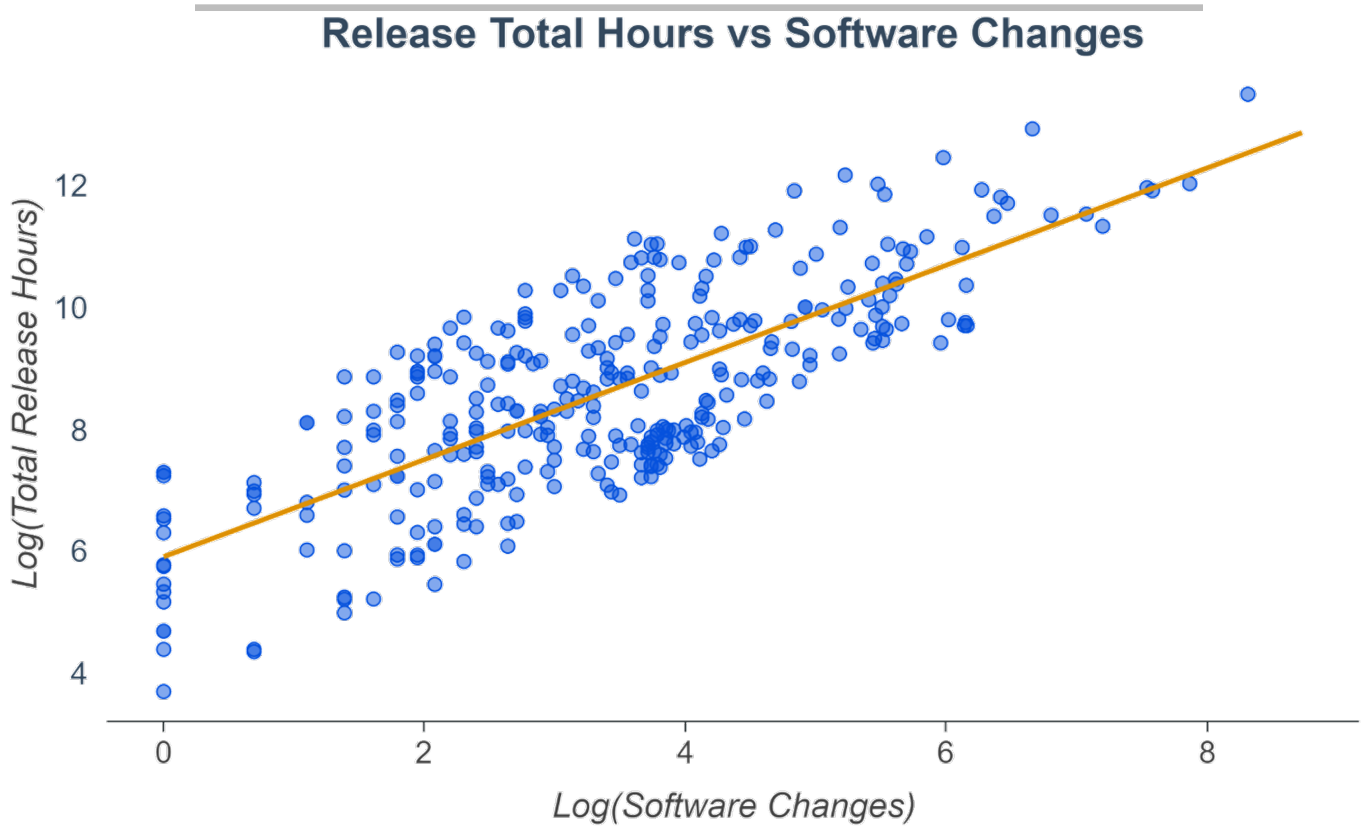
Equation	
$\log(Hours) = 0.703 \log SC + 6.2438$	
R-Squared	36.0%
Adj. R-Squared	35.8%
Observations	397

Variable	Coef.	Std Err	T-Value	P-Value
Constant	6.2438	0.172	36.231	0.000
Log(SC)	0.0730	0.047	14.900	0.000

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	
$\log(Hours) = 0.7981 \log SC + 5.905$	
R-Squared	57.2%
Adj. R-Squared	57.1%
Observations	317

Variable	Coef.	Std Err	T-Value	P-Value
Constant	5.9052	0.145	40.618	0.000
Log(SC)	0.7981	0.039	20.532	0.000

* Data records trimmed by 10%

Total Hours vs SW Changes -1

Model		Conditions	Obs	Adj R^2	SEE (Hrs)	PRED(30)
$THrs = 463 * (TSC)^{0.69}$		All data	329	0.36	48,385	17.3%
$THrs = 341 * (TSC)^{0.79}$		10% trimmed data	263	0.57	44,842	23.6%
AIS	$THr = 242 * (TSC)^{0.73}$	10% trimmed & Super Domains (Categorical)	263	0.62	39,330	20.2%
ENG	$THr = 386 * (TSC)^{0.73}$					
RT	$THr = 736 * (TSC)^{0.73}$					
SUP	$THr = 698 * (TSC)^{0.73}$					
Aviation	$THrs = 1,452 * TSC^{0.66}$	10% trimmed & Commodities (Categorical)	263	0.68*	40,886	23.2%
Business	$THrs = 301 * TSC^{0.66}$					
C5ISR	$THrs = 364 * TSC^{0.66}$					
ChemBio	$THrs = 182 * TSC^{0.66}$					
Fire	$THrs = 1,531 * TSC^{0.66}$					
Missiles	$THrs = 1,114 * TSC^{0.66}$					
Simulation	$THrs = 577 * TSC^{0.66}$					
Space	$THrs = 1,005 * TSC^{0.66}$					
Test	$THrs = 1,742 * TSC^{0.66}$					
Vehicles	$THrs = 425 * TSC^{0.66}$					
$THrs = 608 * (TSC)^{0.98} / (TReqts)^{0.21}$		10% trimmed	32	0.84	32,228	25.0%
$THrs = 330 * (TSC)^{0.97} / (TReqts\_Imp)^{0.11}$		10% trimmed	65	0.74	63,904	23.1%

* High P-Values for one or more coefficients

Total Hours vs SW Changes -2

Model	Conditions	Obs	Adj R^2	SEE (Hrs)	PRED(30)
$THrs = 296 * (TSC)^{0.94} / (EI\_Mod)^{0.11}$	10% trimmed	41	0.74*	47,326	22.0%
$THrs = 1{,}219 * (TSC)^{0.75} / (SWBase)^{0.04}$	10% trimmed	69	0.61*	36,567	26.1%
$THrs = 757 * (TSC)^{1.02} / (BL)^{0.36}$	10% trimmed	45	0.74	81,719	15.6%
Cyber: $THrs = 332 * TSC^{0.79}$ Enhance: $THrs = 531 * TSC^{0.79}$ Hybrid: $THrs = 382 * TSC^{0.79}$ Maint: $THrs = 281 * TSC^{0.79}$ Other: $THrs = 284 * TSC^{0.79}$	10% trimmed & Change Type (Categorical)	263	0.59*	39,573	21.3%
$THrs = 338 * TSC^{0.77} * Enh\%^{0.10} * Maint\%^{0.02} * Cyber\%^{0.03} * Other\%^{0.01}$	10% trimmed & percentages of Change Types	263	0.60*	26,494	6.8%

Unclassified
Distribution Statement A: Approved for Public Release Distribution is Unlimited

IAVA Release Analysis

IAVA Analysis

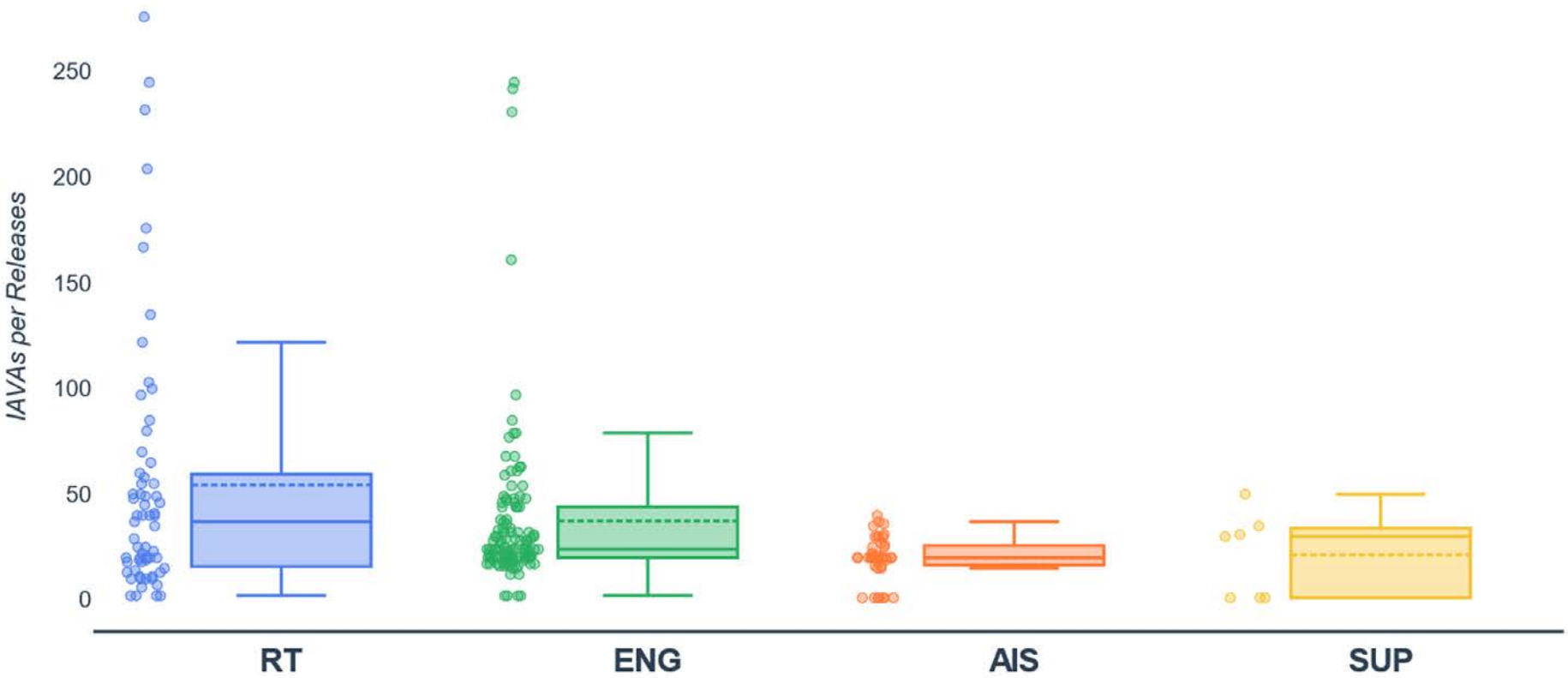
Group	Mean	Median
MaintOrg	243.7	24.2
System	164.0	25.6
Commodity	28.1	30.1
Super Domain	38.0	36.5

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

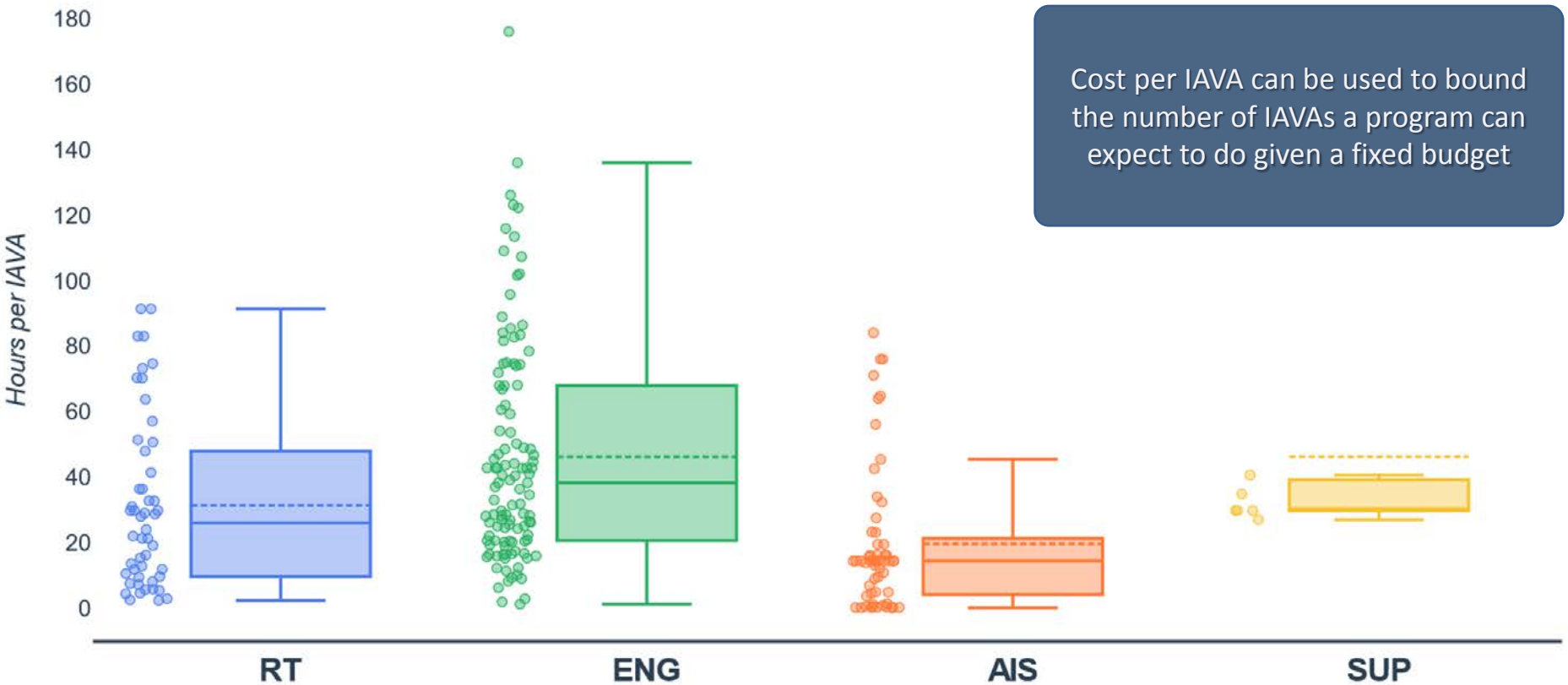
IAVA Releases



Super Domain	Count	Mean	Median
RT	63	55	37
ENG	116	38	24
AIS	43	20	20
SUP	7	21	30

Hours per IAVA

IAVA Releases



Cost per IAVA can be used to bound the number of IAVAs a program can expect to do given a fixed budget

Super Domain	Count	Mean	Median
RT	50	31.6	26.2
ENG	113	46.4	38.5
AIS	56	19.8	14.6
SUP	7	46.4	30.0

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

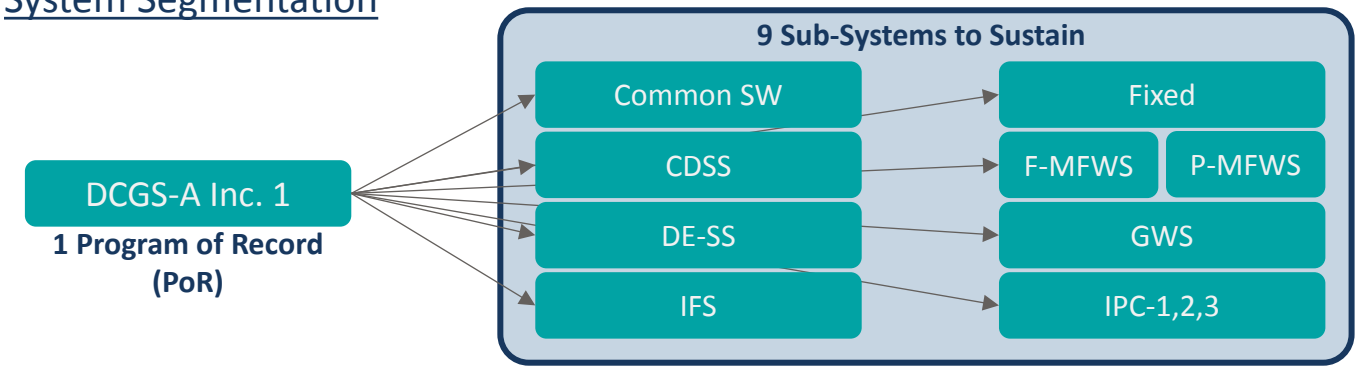
- Utilize data SWM PH 2 data repository to filter for analogous systems
- Develop custom regression models based on systems within the analogous set

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

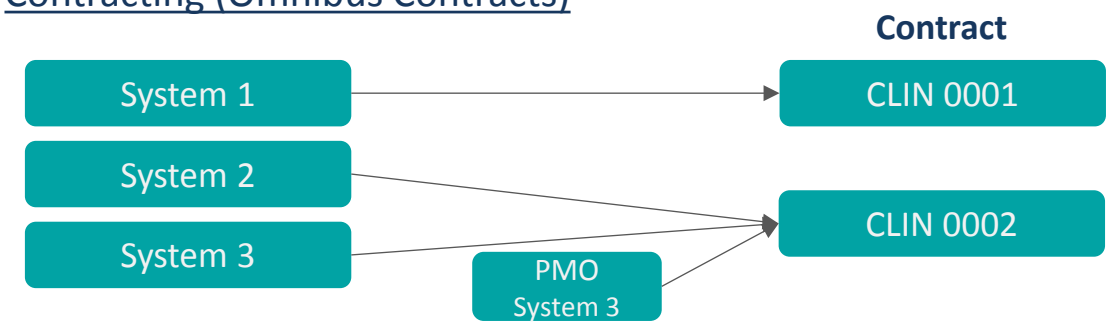
Why Software Sustainment Cost is Difficult to Track

1 System Segmentation



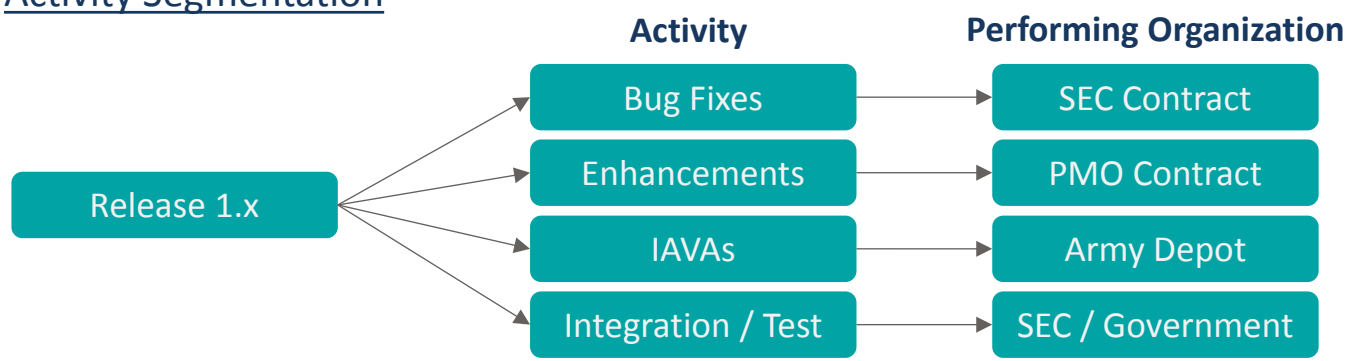
- System segmentation makes it difficult to track software size, licenses, requirements, across the lifecycle
- Each sub-system is separately managed

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

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
    - 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

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

Acronyms -2

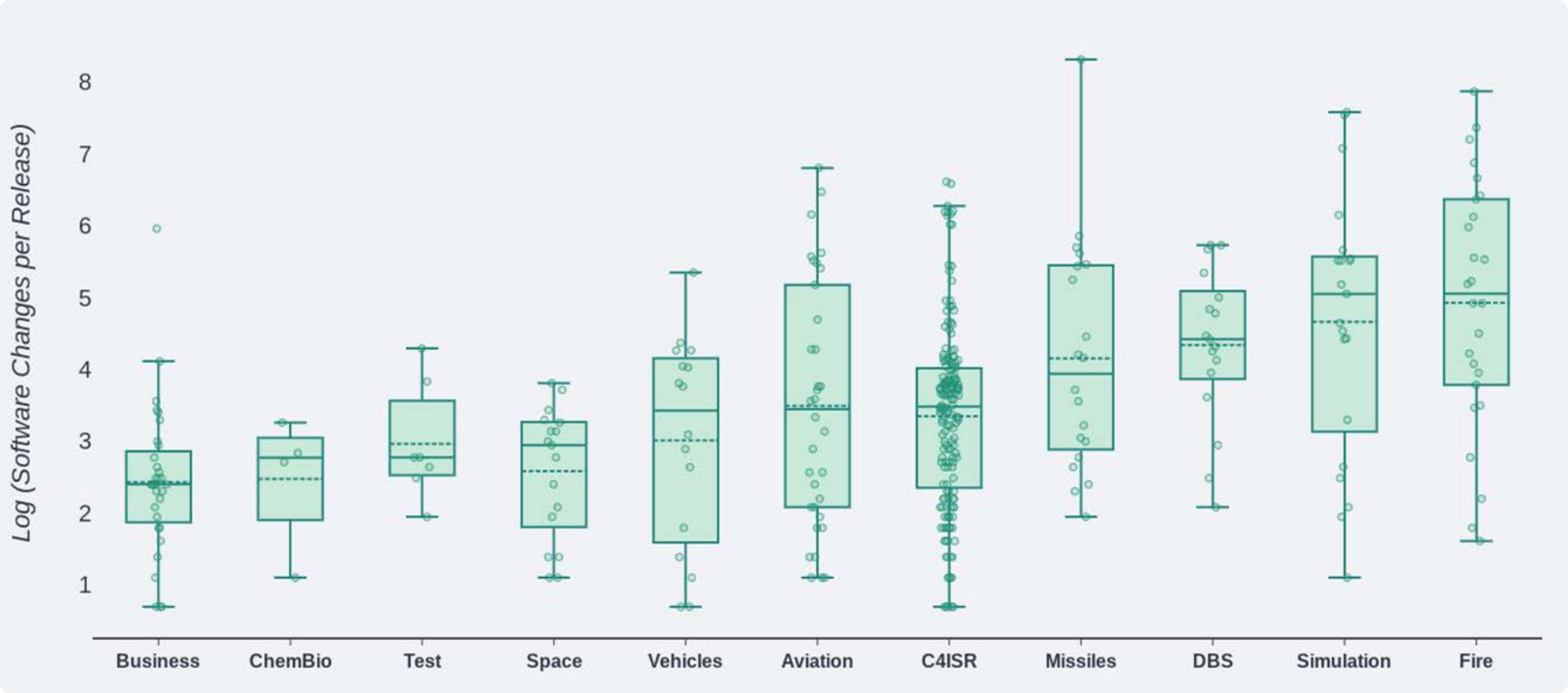
FSE	Field Software Engineering
FTE	Full Time Equivalent
IAVA	Information Assurance Vulnerability Alert
IAVM	Information Assurance Vulnerability Management
ICEAA	International Cost Estimating and Analysis Association
Maint%	Percent of the release that is Maintenance changes
NVD	National Vulnerability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

Acronyms - 3

RDT&E	Research, Development, Testing, and Evaluation
RMF	Risk Management Framework
RT	Real-Time super domain
SC	Software Changes
SEC	Software Engineering Center
SER	Schedule Estimating Relationship
SLOC	Source Lines of Code
SRDR	Software Resources Data Report
SRDR-M	Software Resources Data Report for Maintenance
STIG	Security Technical Implementation Guides
SUP	Mission Support super domain
SW	Software
SWBase	Software Baseline SLOC
SWS	Software Sustainment
TDEV	Time to Develop
THrs	Total release hours
TReqts	Total Requirements in a system
TReqts_Imp	Total Requirements Implemented in a release
TSC	Total Software Changes for a release
WBS	Work Breakdown Structure

Backup

Software Changes per Release
by Commodity



	Business	ChemBio	Test	Space	Vehicles	Aviation	C5ISR	Missiles	DBS	Simulation	Fire
Count	32	4	7	17	16	34	172	20	17	21	26
Mean	2.43	2.47	2.96	2.58	3.01	3.49	3.35	4.15	4.34	4.66	4.93
St Dev	1.05	0.95	0.81	0.91	1.47	1.71	1.28	1.60	1.08	1.83	1.73
Median	2.40	2.77	2.77	2.94	3.43	3.44	3.48	3.94	4.42	5.05	5.05

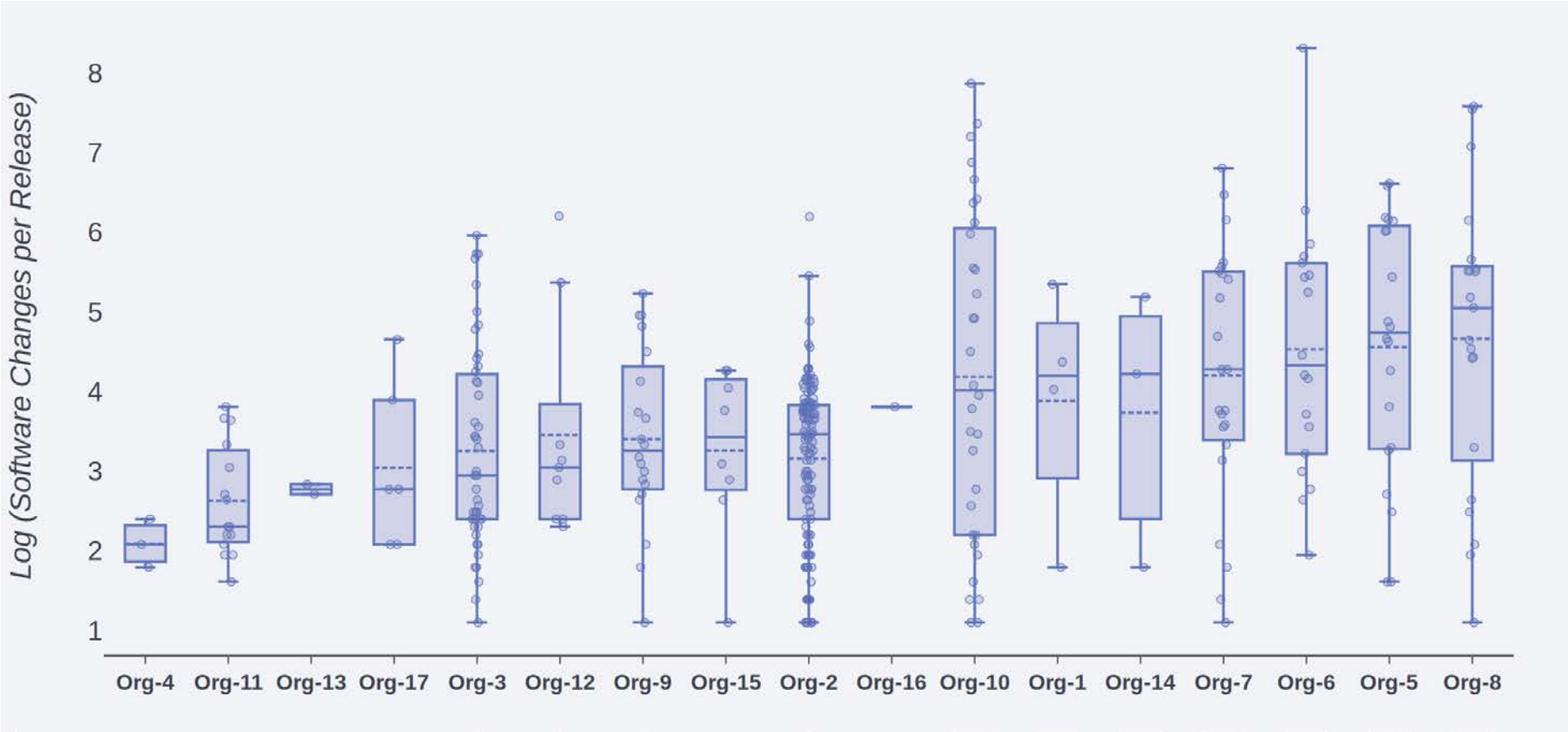
Software Change Definition Variability

- 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

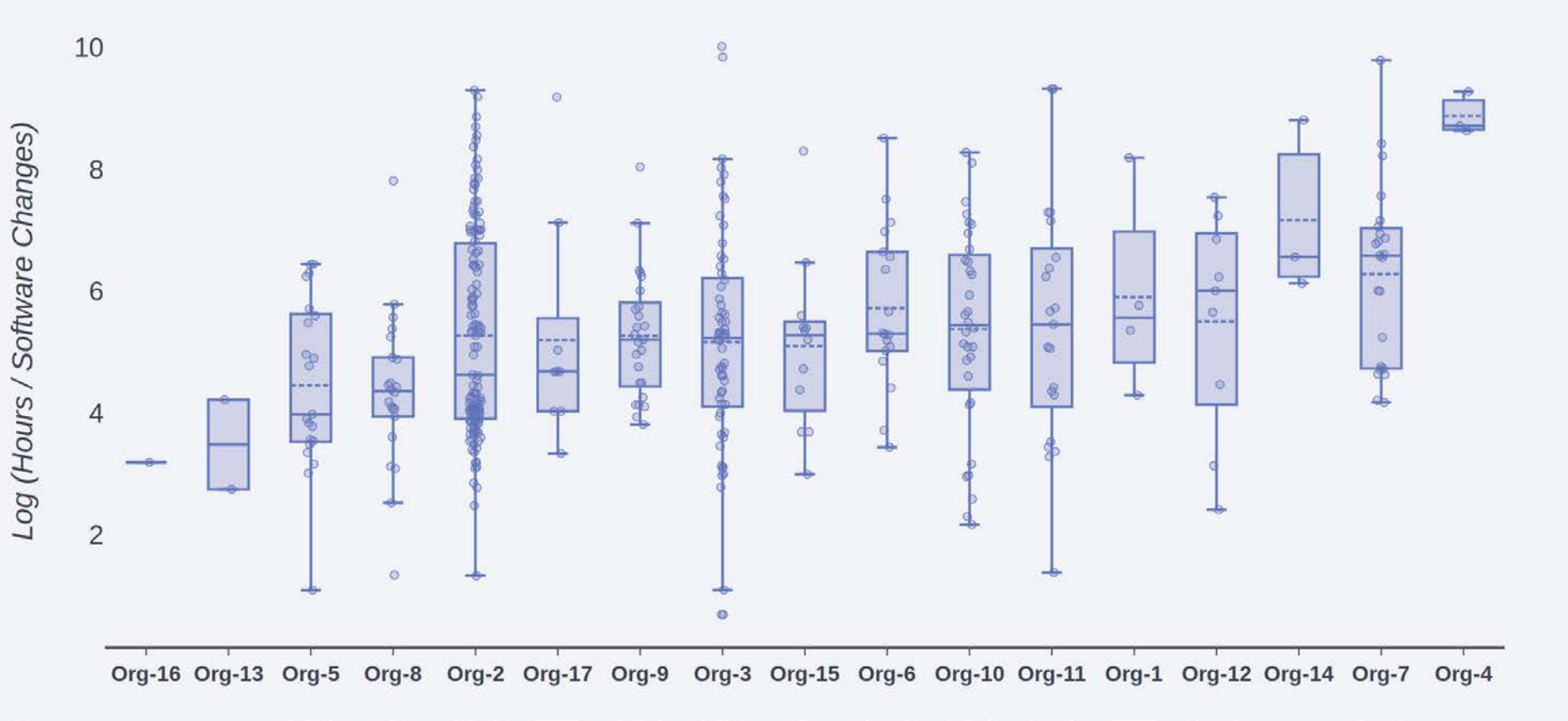
Benchmarks for Capability Releases

Software Changes per Release
By Sustaining Organization



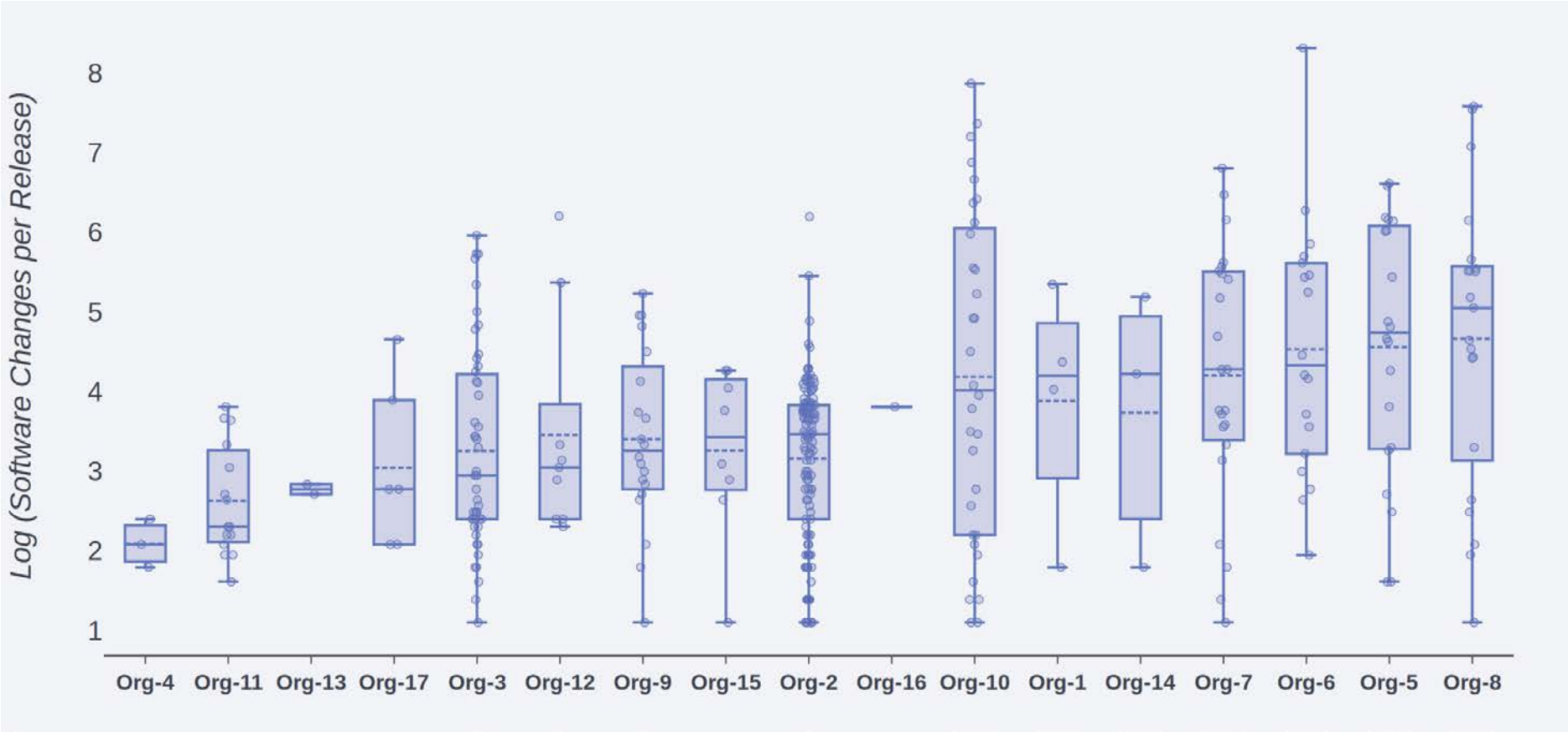
	Org-4	Org-11	Org-13	Org-17	Org-3	Org-12	Org-9	Org-15	Org-2	Org-16	Org-10	Org-1	Org-14	Org-7	Org-6	Org-5	Org-8
Count	1	2	21	22	131	9	25	61	12	18	32	21	4	9	3	23	3
Mean	3.19	3.483	4.454	4.352	5.268	5.197	5.266	5.163	5.099	5.719	5.378	5.456	5.901	5.503	7.164	6.279	8.87
St Dev	N/A	1.039	1.399	1.281	1.712	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	3.19	3.483	3.978	4.361	4.626	4.682	5.202	5.231	5.278	5.3	5.439	5.454	5.56	6.007	6.559	6.582	8.712

Hours per Software Change
By Sustaining Organization



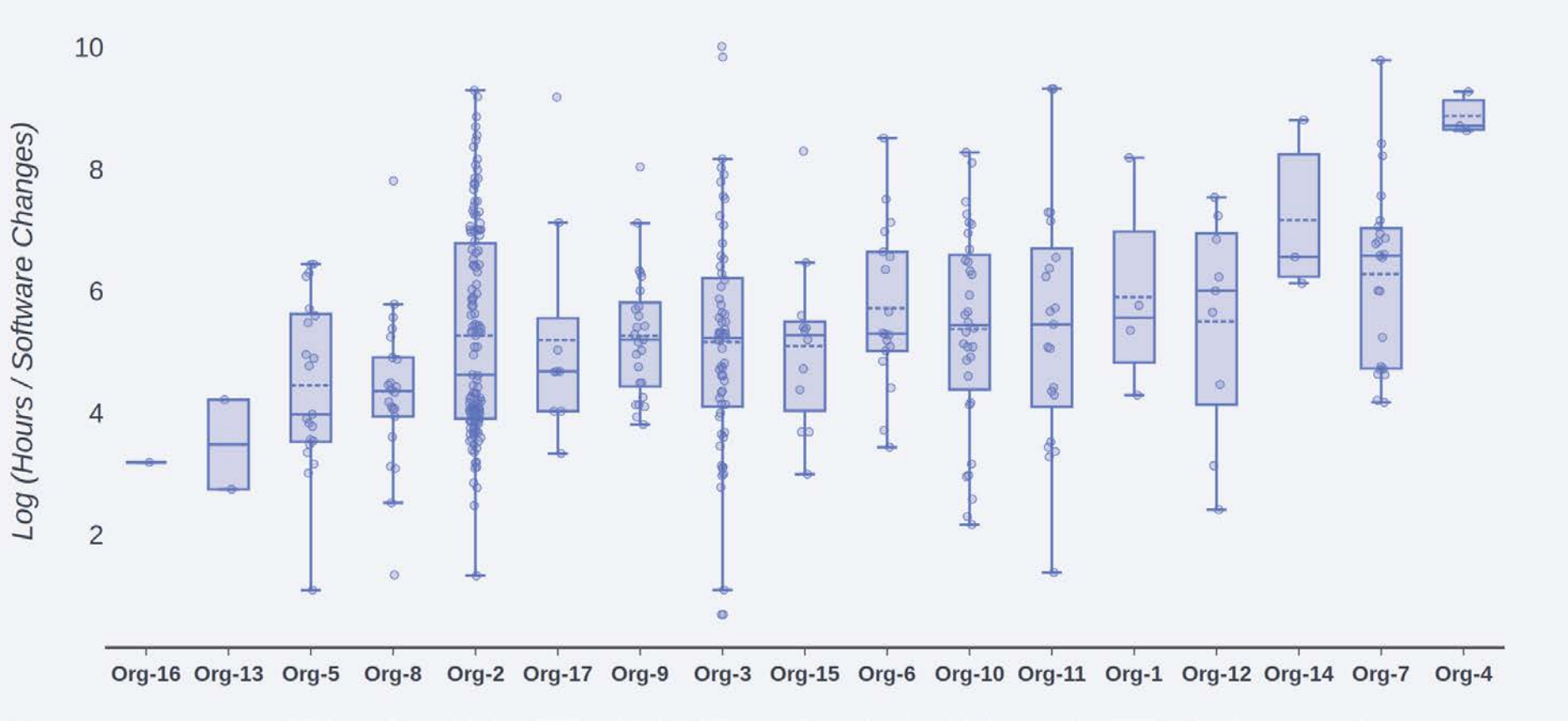
	Org-16	Org-13	Org-5	Org-8	Org-2	Org-17	Org-9	Org-3	Org-15	Org-6	Org-10	Org-11	Org-1	Org-12	Org-14	Org-7	Org-4
Count	1	2	21	22	131	9	25	61	12	18	32	21	4	9	3	23	3
Mean	3.19	3.483	4.454	4.352	5.268	5.197	5.266	5.163	5.099	5.719	5.378	5.456	5.901	5.503	7.164	6.279	8.87
St Dev	N/A	1.039	1.399	1.281	1.712	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	3.19	3.483	3.978	4.361	4.626	4.682	5.202	5.231	5.278	5.3	5.439	5.454	5.56	6.007	6.559	6.582	8.712

Software Changes per Release
By Sustaining Organization



	Org-4	Org-11	Org-13	Org-17	Org-3	Org-12	Org-9	Org-15	Org-2	Org-16	Org-10	Org-1	Org-14	Org-7	Org-6	Org-5	Org-8
Count	1	2	21	22	131	9	25	61	12	18	32	21	4	9	3	23	3
Mean	3.19	3.483	4.454	4.352	5.268	5.197	5.266	5.163	5.099	5.719	5.378	5.456	5.901	5.503	7.164	6.279	8.87
St Dev	N/A	1.039	1.399	1.281	1.712	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	3.19	3.483	3.978	4.361	4.626	4.682	5.202	5.231	5.278	5.3	5.439	5.454	5.56	6.007	6.559	6.582	8.712

Hours per Software Change
By Sustaining Organization



	Org-16	Org-13	Org-5	Org-8	Org-2	Org-17	Org-9	Org-3	Org-15	Org-6	Org-10	Org-11	Org-1	Org-12	Org-14	Org-7	Org-4
Count	1	2	21	22	131	9	25	61	12	18	32	21	4	9	3	23	3
Mean	3.19	3.483	4.454	4.352	5.268	5.197	5.266	5.163	5.099	5.719	5.378	5.456	5.901	5.503	7.164	6.279	8.87
St Dev	N/A	1.039	1.399	1.281	1.712	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	3.19	3.483	3.978	4.361	4.626	4.682	5.202	5.231	5.278	5.3	5.439	5.454	5.56	6.007	6.559	6.582	8.712

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

GAO Guides and Best Practices

Expert Group Process

- 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
    - 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:
Cost Guide: https://www.gao.gov/new.items/d093sp.pdf
Schedule Guide: https://www.gao.gov/assets/680/674404.pdf

Four Characteristics of a Reliable Estimate

Are all costs included?	Can the estimate be recreated?	Is the estimate unbiased?	What is the uncertainty?
Comprehensive	**Well-Documented**	**Accurate**	**Credible**
• Develop the estimating plan • Determine the estimating approach	• Define the program • Identify ground rules and assumptions • Obtain data • Document the estimate • Present estimate to management	• Develop the point estimate • Compare the point estimate to an independent estimate • Update the estimate with actual costs	• Create an independent cost estimate • Conduct sensitivity analysis • Conduct risk and uncertainty analysis

Four Characteristics of a Reliable Schedule

Is all effort included?

Comprehensive

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

Is the network logical?

Well Constructed

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

What is the uncertainty?

Credible

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

Is progress measured?

Controlled

- Update the schedule with progress
- Maintain a schedule baseline

Three Characteristics of a Reliable EVM System

Is the EVM system certified and comprehensive?

Comprehensive

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

Is the EVM data reliable?

Accurate

- No data anomalies
- Consistent data
- Realistic EAC

Is Management using the EVM data?

Informative

- Regular reviews conducted
- Corrective action plans
- Updated PMB

AGILE ASSESSMENT GUIDE

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

Agile and Program Controls

Chapter 9: WBS

- 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	Agile Example
• 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	• 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

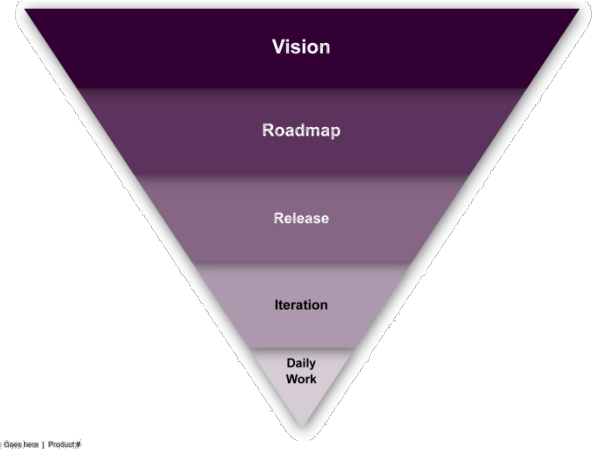
GAO Schedule Guide	Agile Example
• **Step 4: Establish Durations** • The schedule should realistically reflect how long each activity will take. When the duration of each activity is determined, the same rationale, historical data, and assumptions used for cost estimating should be used. Durations should 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	• 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

Agile and Program Controls

Chapter 9: Schedule

- 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	Agile example
• Step 4: Estimate the labor and material required to perform the work and authorize the budgets, including management reserve	• Features should be the basis for identifying work package scope and budget

Agile and Program Controls

Chapter 9: EVM

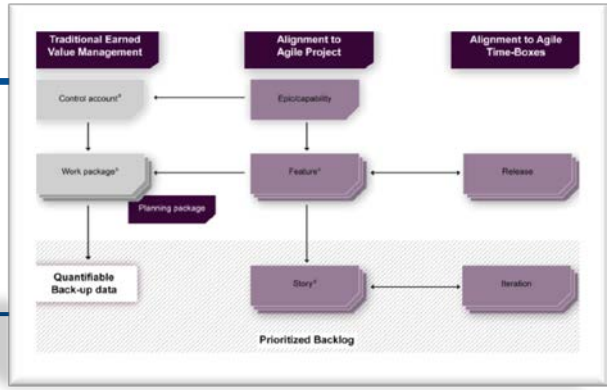
- EVM best practices established in the GAO Cost Guide are still applicable to Agile projects but these projects should use Agile documentation and artifacts produced during Agile development to develop and update cost estimates

- 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:
http://www.gao.gov/products/GAO-16-410G

GAO STAA Engineering Sciences

Jason T. Lee, Assistant Director, leejt1@gao.gov
Jennie Leotta, Assistant Director, leottaj@gao.gov
Brian Bothwell, Assistant Director, bothwellb@gao.gov

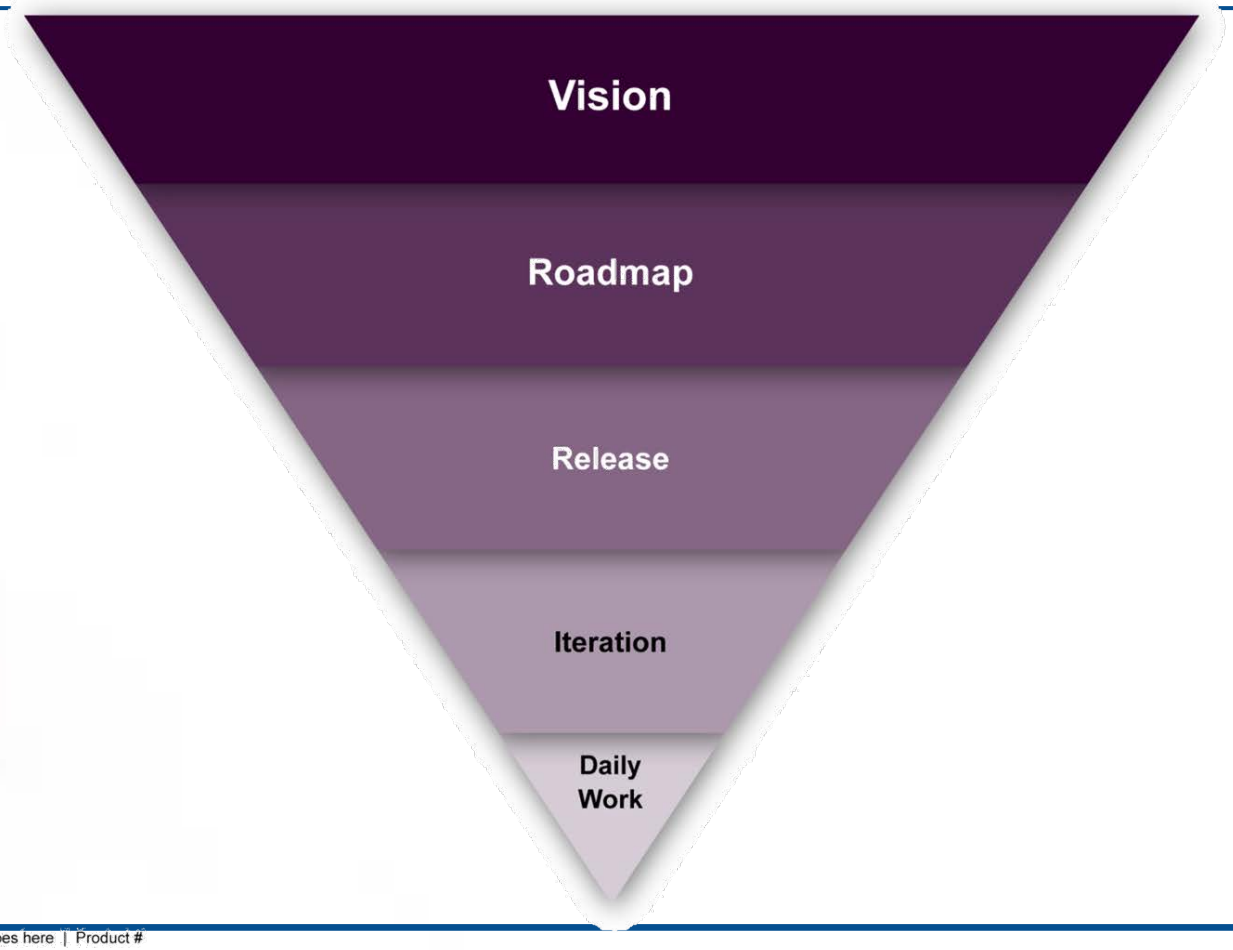
GAO on the Web

Web site: http://www.gao.gov/

Copyright

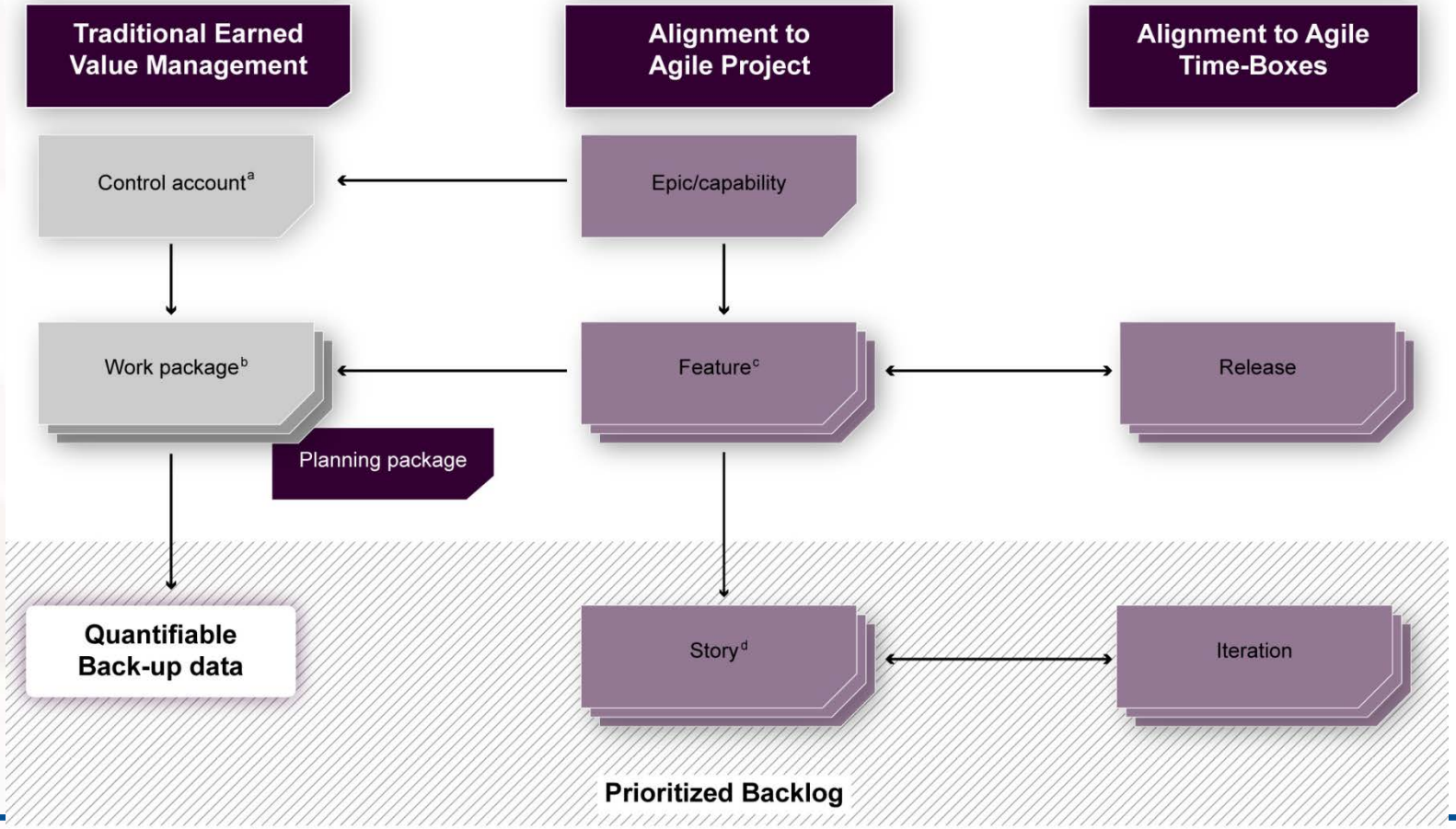
This is a work of the U.S. government and is not subject to copyright protection in the United States. The published product may be reproduced and distributed in its entirety without further permission from GAO. However, because this work may contain copyrighted images or other material, permission from the copyright holder may be necessary if you wish to reproduce this material separately.

Agile Work



Source: Goes here | Product #

EVM Relationships in Agile Projects



Source: Goes here | Product #

PUBLIC MEETING

DEFENSE INNOVATION BOARD

MARCH 21, 2019 | NATIONAL DEFENSE UNIVERSITY

#DefenseInnovationBoard

@InnovationBoard

innovation.defense.gov

osd.innovation@mail.mil

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..."

CONGRESS' CHARGE

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

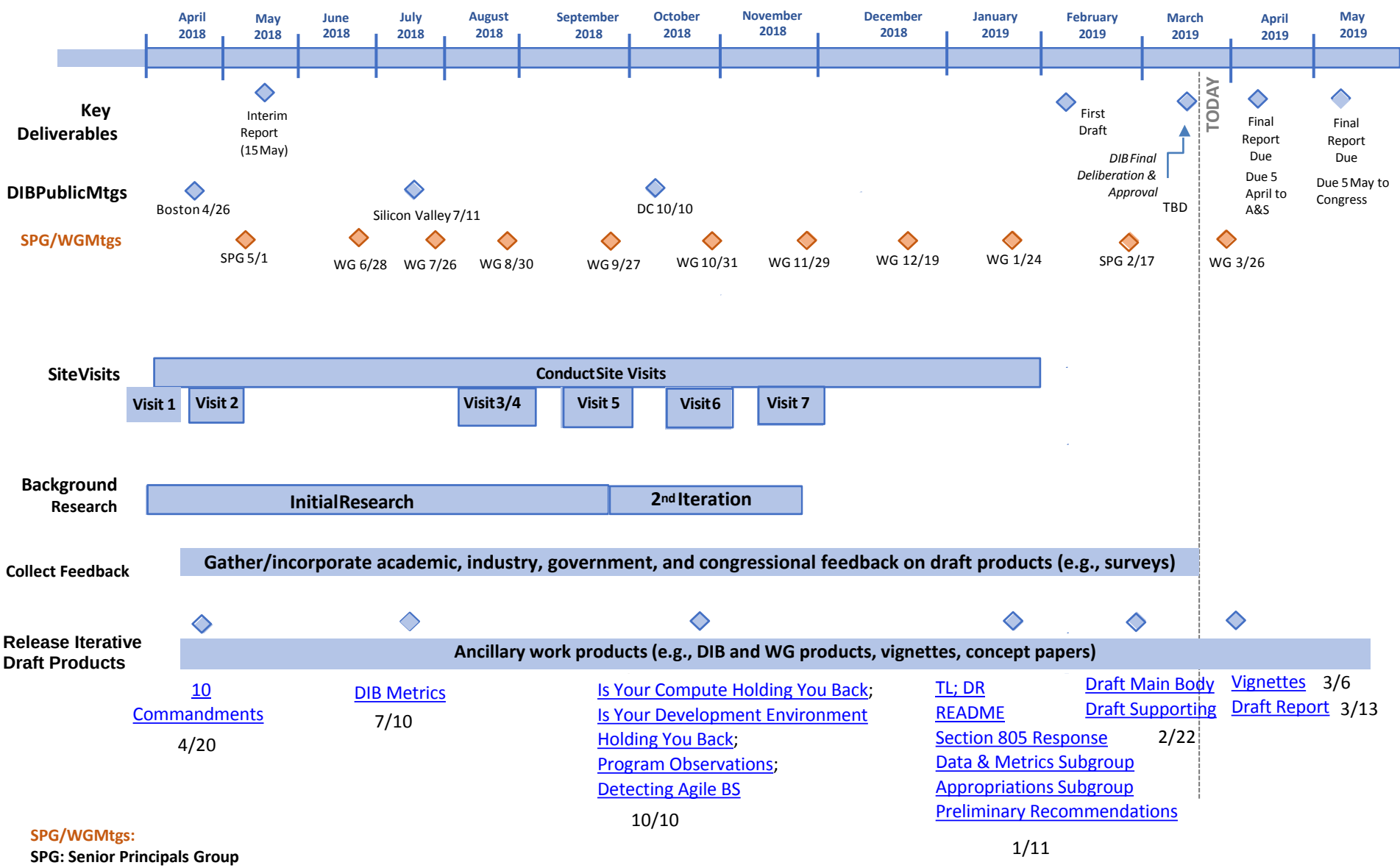
Mr. Gilman Louie
(Consultant)
Founder and Partner of Alsop-Louie Partners

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

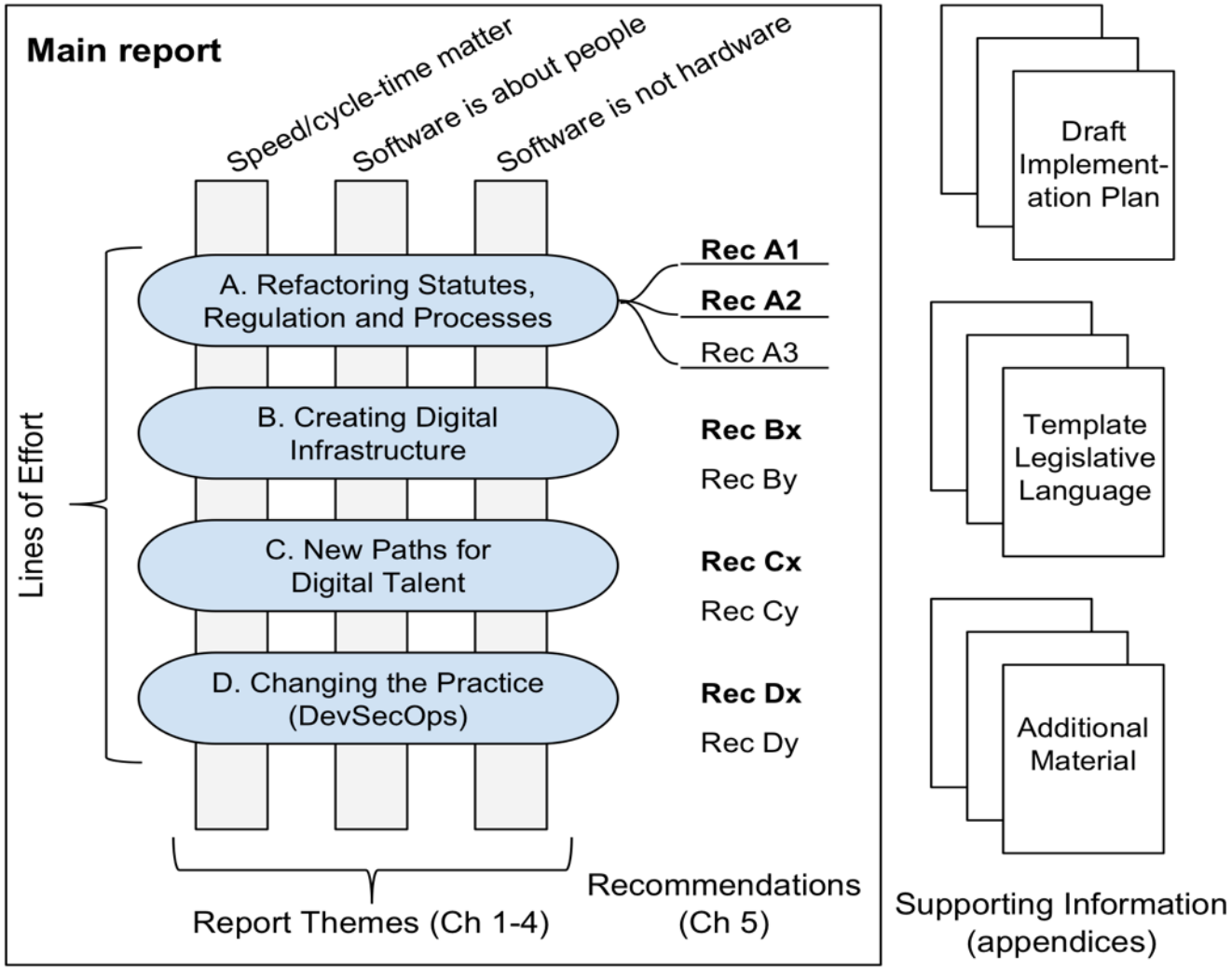
APPROACH



SPG/WG Mtgs:
SPG: Senior Principals Group
WG: Working Group

UNCLASSIFIED

DOCUMENT REVIEW



PATH TO ACTION

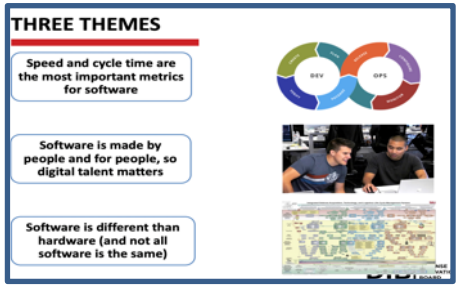
3 Themes → describe software capabilities world which must be improved by focusing on…

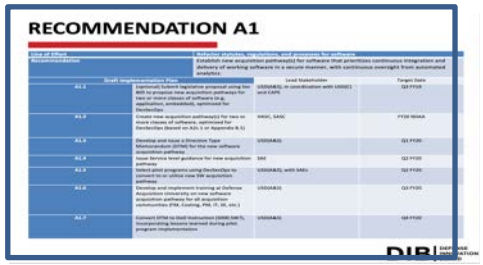
4 Lines of Effort → describe the main areas that must be change, by acting on…

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

Initial Implementation Plans comprised of…

Actions



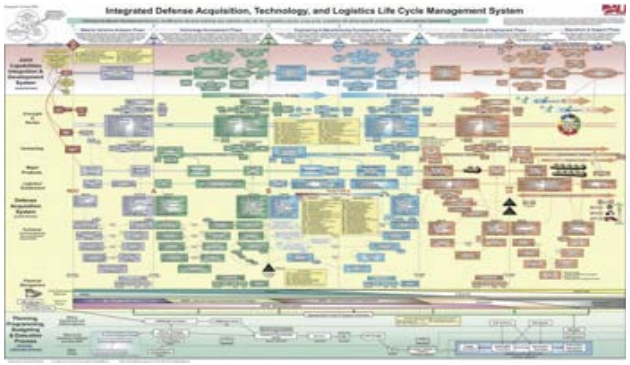


THREE THEMES

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)



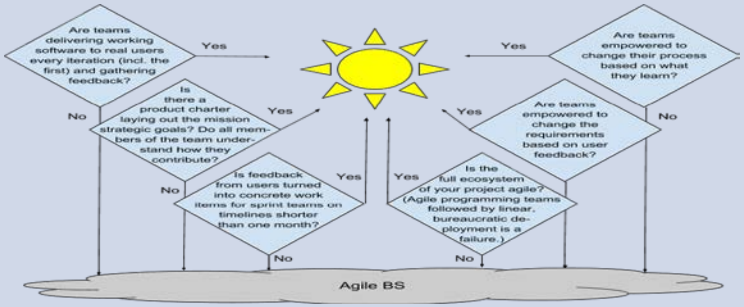
FOUR LINES OF EFFORT

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	Lead Stakeholder	Target Date
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.		
	Draft Implementation Plan	Lead Stakeholder	Target Date
A2.1	(optional) Submit legislative proposal using Sec 805 to create a new appropriations category for software and software-intensive programs	USD(A&S), with USD(C) and CAPE	Q3 FY19 for FY20 NDAA
A2.2	Create new appropriation category for software-intensive programs, with appropriate reporting and oversight for software (based on Action A2.1 or Appendix B.1)	HAC-D, SAC-D, with OSD, HASC, SASC	FY20 NDAA, FY20 budget
A2.3	Select initial programs using DevSecOps to convert to or use new SW Appropriation in FY20	USD(A&S), with Service Acquisition Executives	Q4 FY19
A2.4	Define budget exhibits for new software appropriation (replacement for P- and R-forms; see Appendix C)	USD(A&S), with USD(C), CAPE, HAC-D, SAC-D	Q4 FY19
A2.5	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	FASAB, with USD(A&S) and USD(C)	End FY20
A2.6	Make necessary modifications in supporting PPB&E systems to allow use and tracking of new software appropriation	USD(C) and CAPE	Q1 FY21
A2.7	Ensure programs using new software appropriation submit budget exhibits in the approved format	SAE with USD(C), CAPE	FY 22 POM

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	Lead Stakeholder	Target Date
B1.1	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	DoD CIO, USD(C) and Services (SAE and Service CIO)	Q3 FY19
B1.2	Designate organization responsible for creating and maintaining digital infrastructure for DoD agencies and organizations, including joint digital infrastructure available to the Services	USD(A&S), with CIO, CMO	Q3 FY19
B1.3	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])	USD(A&S), SAE with CAPE, USD(C)	FY20 budget
B1.4	Define baseline digital infrastructure systems and implement procurement and deployment processes and capability	Responsible organizations from B1.1, B1.2	Q2 FY20
B1.5	Implement digital infrastructure and provide access to ongoing and new programs	Responsible organizations from B1.1, B1.2	Q3 FY20
B1.6	Identify acquisition programs to transition to digital infrastructure	SAE	Q2 FY20
B1.7	Transition programs to digital infrastructure	SAE, PEO, PM	Q4 FY20

RECOMMENDATION B2

Line of Effort	Create and maintain cross-program/cross-service digital infrastructure		
Recommendation	Create, implement, support, and use fully automatable approaches to testing and evaluation (T&E), including security, that allow high confidence distribution of software to the field on an iterative basis.		
	Draft Implementation Plan	Lead Stakeholders	Target Date
B2.1	Establish procedures for fully automated testing on digital infrastructure (Rec B1), updating DoDI 5129.47 and Service equivalents	USD(A&S), DOT&E, with Service Testers	Q1 FY20
B2.2	Establish processes for automated security testing, including zero-trust assumptions, automated penetration testing, and red teams for vulnerability scanning	USD(A&S), DOT&E, with Service Testers	Q1 FY20
B2.3	Identify initial programs to use tools and workflows	SAE	Q1 FY20
B2.4	Implement minimum viable product (MVP) tools and workflows on digital infrastructure (Rec B1)	SAE, DOT&E, with PMOs	Q2 FY20
B2.5	Migrate initial programs to digital infrastructure using automated T&E	PEO, with Responsible Organizations	Q3 FY20
B2.6	Use tools and workflows, identify lessons learned and improvements (using DevSecOps iterative approach)	Service Testers, with PEO/PM	Q4 FY20
B2.7	Modify tools and workflows, document procedures	Responsible Organizations, Service Testers	Q4 FY20

RECOMMENDATION B3

Line of Effort	Create and maintain cross-program/cross-service digital infrastructure		
Recommendation	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.		
	Draft Implementation Plan	Lead Stakeholder	Target Date
B3.1	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	DoD CIO, with Service CIOs	Q3 FY19
B3.2	Establish DoD-wide repository for ATO artifacts with tools and access rules that enable Services to identify existing ATOs and utilize them when possible	DoD CIO, with Service CIOs, DISA	Q4 FY19
B3.3	Implement procedures and access controls so that Authorizing Officials have visibility over other programs that are using compatible ATOs	DoD CIO, with Service CIOs, DISA	Q2 FY20
B3.4	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	DoD CIO, with FedRAMP	Q4 FY20

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

RECOMMENDATION C2

Line of Effort	Create new paths for digital talent (especially internal talent)		
Recommendation	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.		
Draft Implementation Plan		Lead Stakeholders	Target Date
C2.1	Leverage existing training venues to add content about modern software development practices	USD(A&S), SAEs with DAU	Q4 FY19
C2.2	Create and provide training opportunities via boot camps and rotations for acquisition professionals to obtain hands-on experience in DevSecOps programs	A&S with SAEs, USD(P&R)	FY20 (MVP) FY21 (scale)
C2.3	Develop additional training opportunities for key leaders about modern software development practices	USD(A&S), SAE	Q2 FY20
C2.4	Create software continuing education programs and requirements for CIOs, SAEs, PEOs and PMs modeled after MCLE (Minimum Continuing Legal Education) for lawyers	A&S, DAU	Q3 FY20

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.

Draft Implementation Plan		Lead Stakeholders	Target Date
D1.1	Work with industry to modernize policies for software code ownership, licensing, and purchase. See 2018 Army IP directive as an example	USD(A&S)	Q3 FY19
D1.2	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)	USD(A&S)	Q3 FY20
D1.3	Modify DoDI 5000.02 and DoDI 5000.75 to make access to code and development environments the default	USD(A&S)	Q3 FY20
D1.4	Develop a comprehensive source code management plan for DoD including the safe and secure storage, access control, testing and field of use rights	USD(A&S), with CIO	Q4 FY20

RECOMMENDATION D2

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

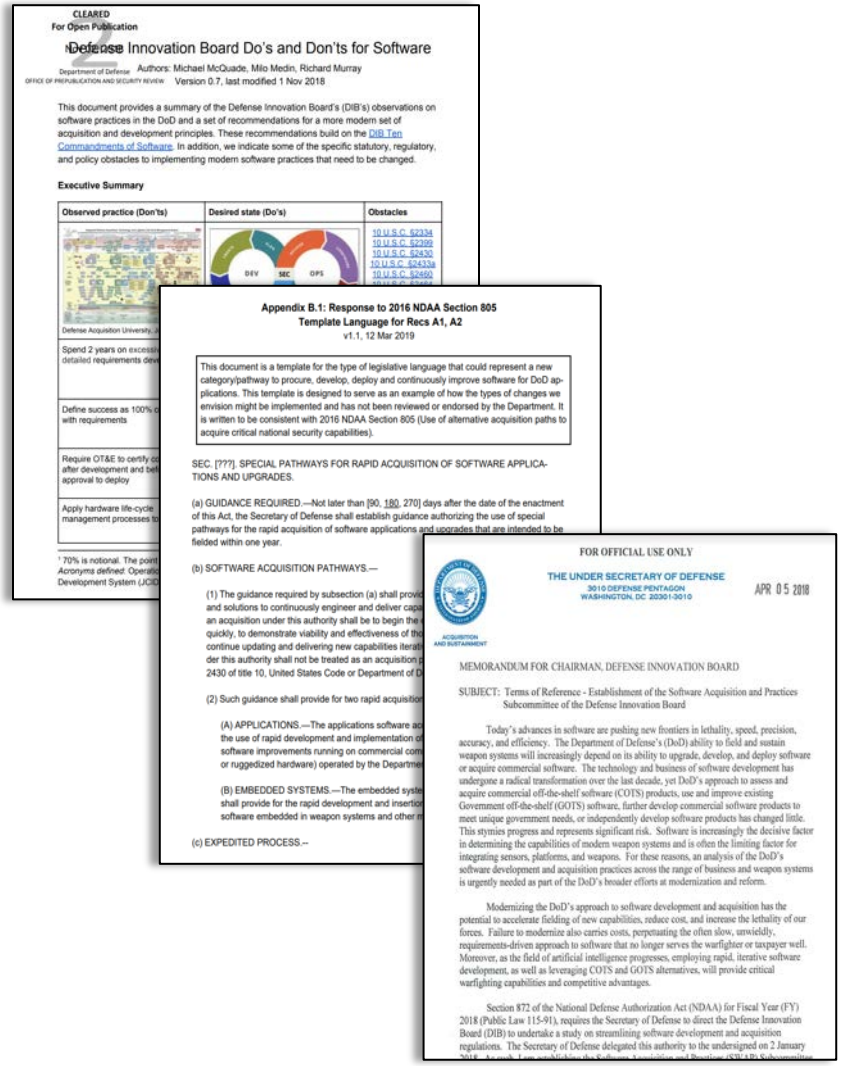
RECOMMENDATION D3

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		Lead Stakeholders	Target Date
D3.1	Modify requirements guidance by memo to shift from a list of requirements for software to a list of desired features and required interfaces/characteristics	USD(A&S), CMO	Q4 FY19
D3.2	Update CJCSI 3170.01H (JCIDS requirements process) to reflect contents of guidance memos	Joint Staff	Q1 FY20
D3.3	Modify DoDI 5000.02 and DoDI 5000.75 (or integrate into new DoDI 5000.SW)	USD(A&S)	Q2 FY20
D3.4	Define and use new budget exhibits for software programs using evolving lists of features in place of requirements (see also Rec A2)	USD(A&S), with USD(C), CAPE, HAC-D, SAC-D	Q3 FY20

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



CLEARED
For Open Publication

Defense Innovation Board Do's and Don'ts for Software

Department of Defense
OFFICE OF PREPUBLICATION AND SECURITY REVIEW

Authors: Michael McQuade, Milo Medin, Richard Murray
Version 0.7, last modified 1 Nov 2018

This document provides a summary of the Defense Innovation Board's (DIB's) observations on software practices in the DoD and a set of recommendations for a more modern set of acquisition and development principles. These recommendations build on the DIB Ten Commandments of Software. In addition, we indicate some of the specific statutory, regulatory, and policy obstacles to implementing modern software practices that need to be changed.

Executive Summary

Observed practice (Don'ts)	Desired state (Do's)	Obstacles
Defense Acquisition University, J...	DEV SEC OPS	10 U.S.C. §2334 10 U.S.C. §2399 10 U.S.C. §2430 10 U.S.C. §2433a 10 U.S.C. §2460
Spend 2 years on excessi... detailed requirements deve...		
Define success as 100% c... with requirements		
Require OT&E to certify co... after development and bef... approval to deploy		
Apply hardware life-cycle management processes to...		

[1] 70% is notional. The point...
Acronyms defined: Operatio... Development System (JCID...

Appendix B.1: Response to 2016 NDAA Section 805
Template Language for Recs A1, A2
v1.1, 12 Mar 2019

This document is a template for the type of legislative language that could represent a new category/pathway to procure, develop, deploy and continuously improve software for DoD applications. This template is designed to serve as an example of how the types of changes we envision might be implemented and has not been reviewed or endorsed by the Department. It is written to be consistent with 2016 NDAA Section 805 (Use of alternative acquisition paths to acquire critical national security capabilities).

SEC. [???]. SPECIAL PATHWAYS FOR RAPID ACQUISITION OF SOFTWARE APPLICATIONS AND UPGRADES.

(a) GUIDANCE REQUIRED.—Not later than [90, 180, 270] days after the date of the enactment of this Act, the Secretary of Defense shall establish guidance authorizing the use of special pathways for the rapid acquisition of software applications and upgrades that are intended to be fielded within one year.

(b) SOFTWARE ACQUISITION PATHWAYS.—

(1) The guidance required by subsection (a) shall provid... and solutions to continuously engineer and deliver capa... an acquisition under this authority shall be to begin the ... quickly, to demonstrate viability and effectiveness of the... continue updating and delivering new capabilities iterati... der this authority shall not be treated as an acquisition p... 2430 of title 10, United States Code or Department of D...

(2) Such guidance shall provide for two rapid acquisition...

(A) APPLICATIONS.—The applications software ac... the use of rapid development and implementation of... software improvements running on commercial com... or ruggedized hardware) operated by the Departme...

(B) EMBEDDED SYSTEMS.—The embedded syste... shall provide for the rapid development and insertio... software embedded in weapon systems and other m...

(c) EXPEDITED PROCESS.--

FOR OFFICIAL USE ONLY

THE UNDER SECRETARY OF DEFENSE
3010 DEFENSE PENTAGON
WASHINGTON, DC 20301-3010

ACQUISITION AND SUSTAINMENT

APR 05 2018

MEMORANDUM FOR CHAIRMAN, DEFENSE INNOVATION BOARD

SUBJECT: Terms of Reference - Establishment of the Software Acquisition and Practices Subcommittee of the Defense Innovation Board

Today's advances in software are pushing new frontiers in lethality, speed, precision, accuracy, and efficiency. The Department of Defense's (DoD) ability to field and sustain weapon systems will increasingly depend on its ability to upgrade, develop, and deploy software or acquire commercial software. The technology and business of software development has undergone a radical transformation over the last decade, yet DoD's approach to assess and acquire commercial off-the-shelf software (COTS) products, use and improve existing Government off-the-shelf (GOTS) software, further develop commercial software products to meet unique government needs, or independently develop software products has changed little. This stymies progress and represents significant risk. Software is increasingly the decisive factor in determining the capabilities of modern weapon systems and is often the limiting factor for integrating sensors, platforms, and weapons. For these reasons, an analysis of the DoD's software development and acquisition practices across the range of business and weapon systems is urgently needed as part of the DoD's broader efforts at modernization and reform.

Modernizing the DoD's approach to software development and acquisition has the potential to accelerate fielding of new capabilities, reduce cost, and increase the lethality of our forces. Failure to modernize also carries costs, perpetuating the often slow, unwieldly, requirements-driven approach to software that no longer serves the warfighter or taxpayer well. Moreover, as the field of artificial intelligence progresses, employing rapid, iterative software development, as well as leveraging COTS and GOTS alternatives, will provide critical warfighting capabilities and competitive advantages.

Section 872 of the National Defense Authorization Act (NDAA) for Fiscal Year (FY) 2018 (Public Law 115-91), requires the Secretary of Defense to direct the Defense Innovation Board (DIB) to undertake a study on streamlining software development and acquisition regulations. The Secretary of Defense delegated this authority to the undersigned on 2 January

DIB Metrics for Software

#	Metric	Target value (by software type)[i]				Typical DoD values for SW
		COTS[ii] apps	Custom -ized SW[iii]	COTS HW/OS[iv]	Real-time HW/SW[v]	
1	Time from program launch to deployment of simplest useful functionality	<1 mo	<3 mo	<6 mo	<1 yr	3-5 yrs
2	Time to field high priority fcn (spec → ops) or fix newly found security hole (find → ops)[vi]	N/A <1 wk	<1 mo <1 wk	<3 mo <1 wk	<3 mo <1 wk	1-5 yrs 1-18 m
3	Time from code committed to code in use	<1 wk	<1 hr	<1 da	<1 mo	1-18 m
4	Time req'd for full regression test (automat'd) and cybersecurity audit/penetration testing[vii]	N/A <1 mo	<1 da <1 mo	<1 da <1 mo	<1 wk <3 mo	2 yrs 2 yrs
5	Time required to restore service after outage	<1 hr	<6 hr	<1 day	N/A	?
6	Automated test coverage of specs / code	N/A	>90%	>90%	100%	?
7	Number of bugs caught in testing vs field use	N/A	>75%	>75%	>90%	?
8	Change failure rate (rollback deployed code)	<1%	<5%	<10%	<1%	?
9	% code available to DoD for inspection/rebuild	N/A	100%	100%	100%	0%
10	Complexity metrics	#/type of specs structure of code #/type of platforms		# programmers #/skill level of teams #/type deployments		Partial/ manual tracking
11	Development plan/environment metrics					
12	"Nunn-McCurdy" threshold (for any metric)	1.1X	1.25X	1.5X	1.5X each effort	1.25X Total $

https://innovation.defense.gov/software

PUBLIC MEETING

DEFENSE INNOVATION BOARD

MARCH 21, 2019 | NATIONAL DEFENSE UNIVERSITY

#DefenseInnovationBoard

@InnovationBoard

innovation.defense.gov

osd.innovation@mail.mil

Generating a Cost Estimate for Agile Software Development

"Succeeding at an Impossible Task"

Matthew R. Kennedy, PhD

Email: MatthewRKennedyEmail@gmail.com

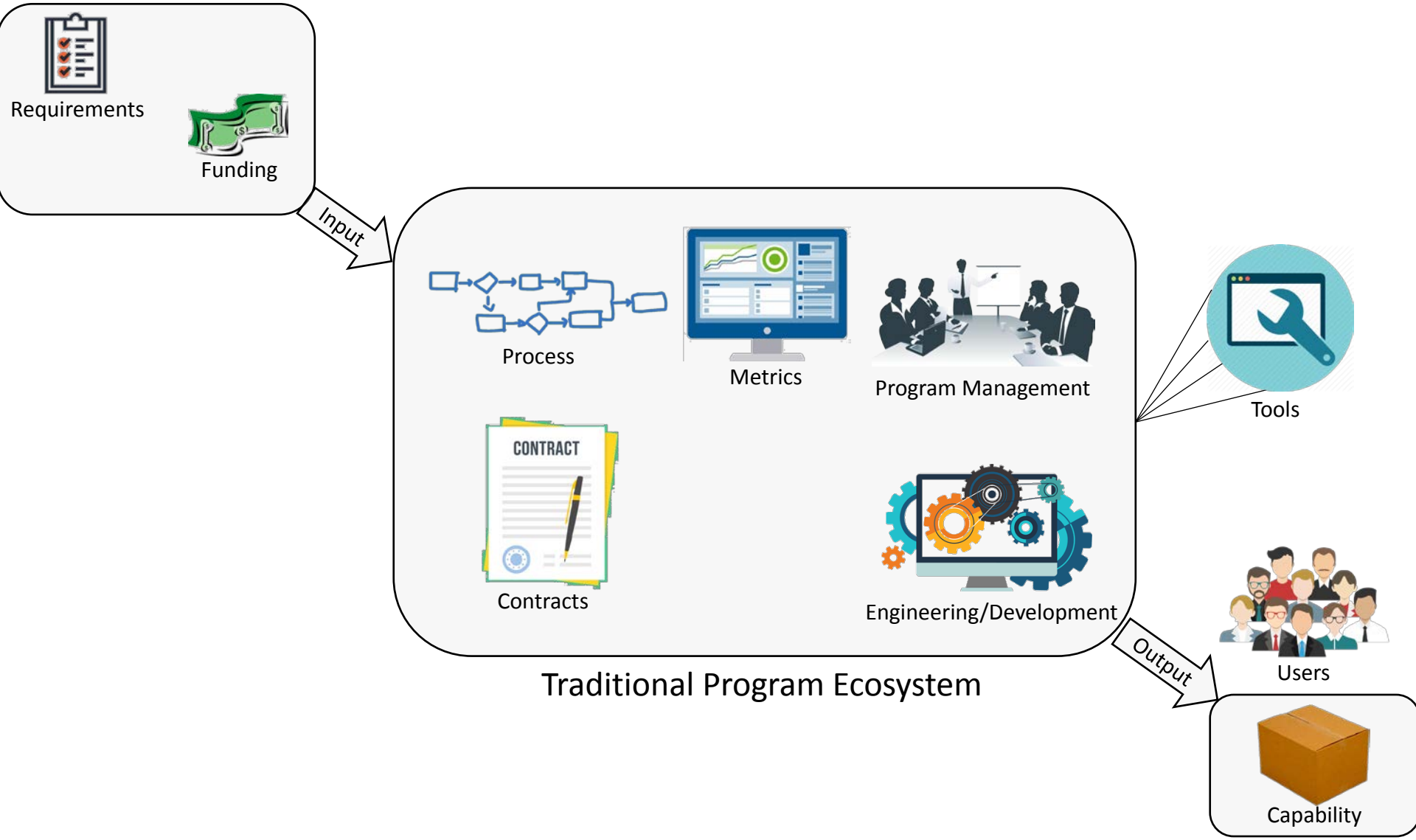
Disclaimer

- 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



Traditional Ecosystem (Input)

Requirements

Funding

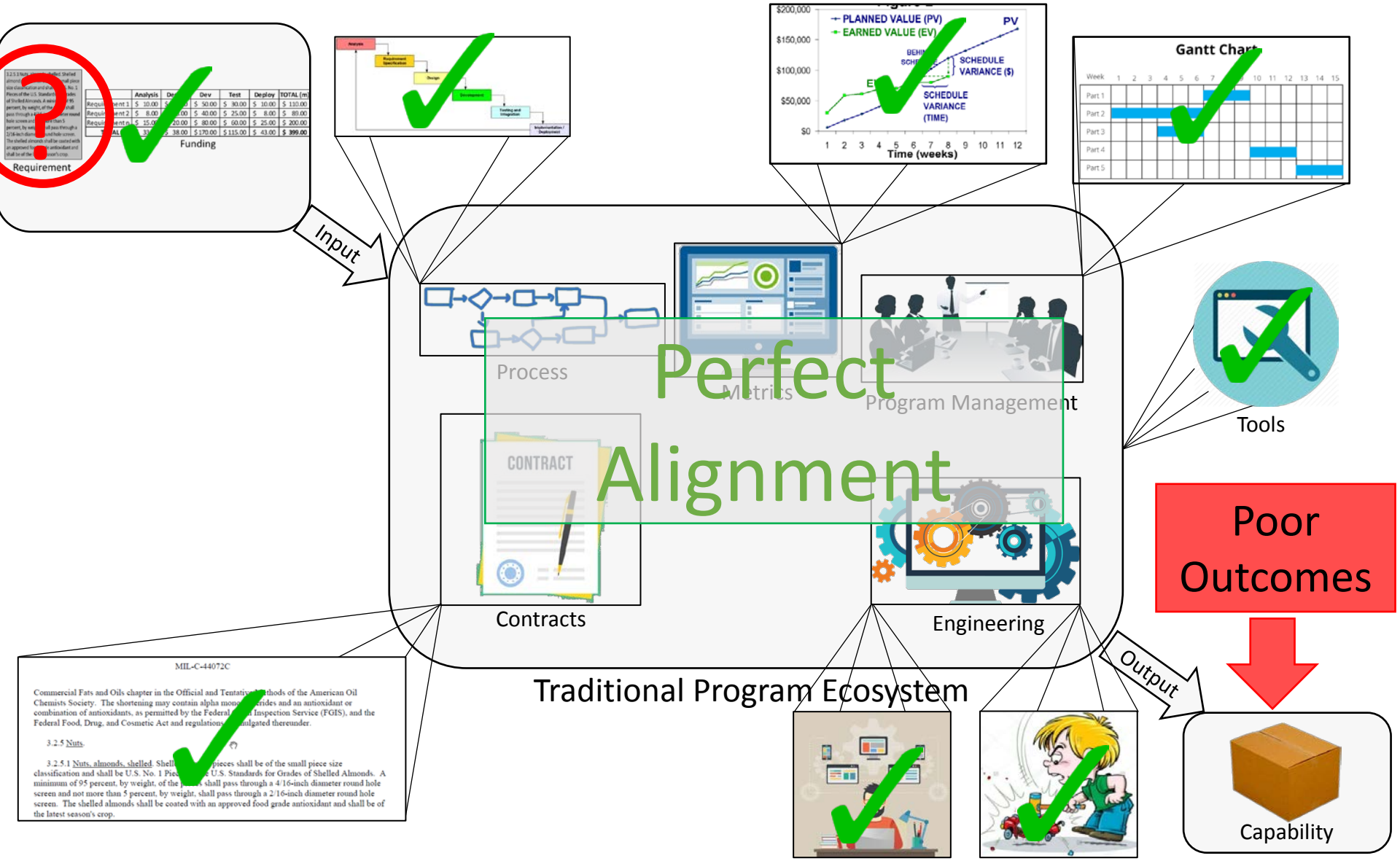
3.2.5.1 Nuts, almonds, shelled. Shelled almond pieces shall be of the small piece size classification and shall be U.S. No. 1 Pieces of the U.S. Standards for Grades of Shelled Almonds. A minimum of 95 percent, by weight, of the pieces shall pass through a 4/16-inch diameter round hole screen and not 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.

Requirement 1

	Analysis	Design	Dev	Test	Deploy	TOTAL (m)
Requirement 1	$ 10.00	$ 10.00	$ 50.00	$ 30.00	$ 10.00	$ 110.00
Requirement 2	$ 8.00	$ 8.00	$ 40.00	$ 25.00	$ 8.00	$ 89.00
Requirement n	$ 15.00	$ 20.00	$ 80.00	$ 60.00	$ 25.00	$ 200.00
TOTAL (m)	$ 33.00	$ 38.00	$ 170.00	$ 115.00	$ 43.00	**$ 399.00**

Cost Estimates

Traditional Ecosystem

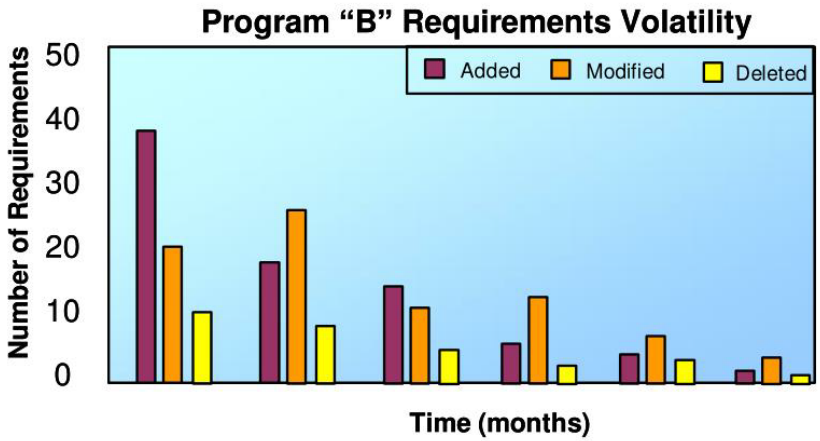


We Can't Gather All of the Requirements Up-Front

12 DIY
Home Improvement
Project Ideas

***The top ts are:**

1. La
2. In
3. C



Volatility metrics track the number and type of changes over time

Source: Hammer, T., Huffman, L., and Rosenberg, L. (1998).

18

*Standish report was published in 1994

We Don’t Know the Efficiency of the Workforce

Chick-fil-A®

Applebee's

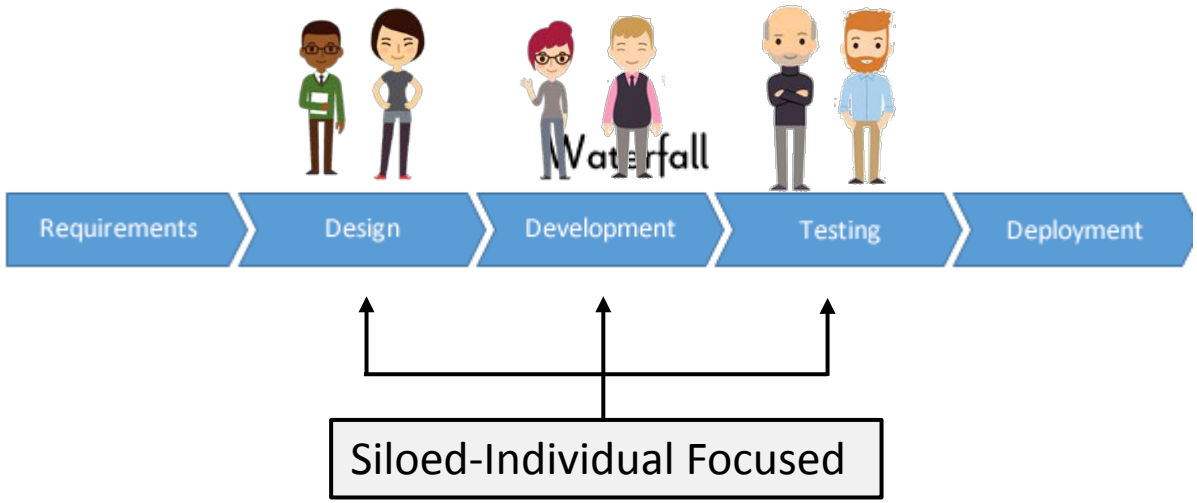
SUBWAY

FIVE GUYS®

2 (of many) Reasons Cost Estimation for Software Development is Challenging?

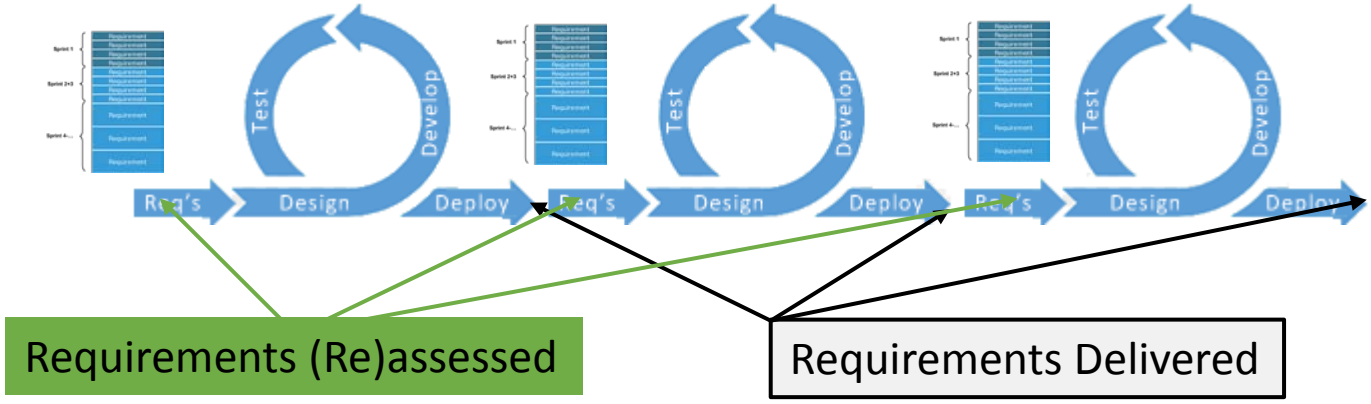
1. We can't gather all of the requirements up front

2. We don't know the efficiency of the workforce

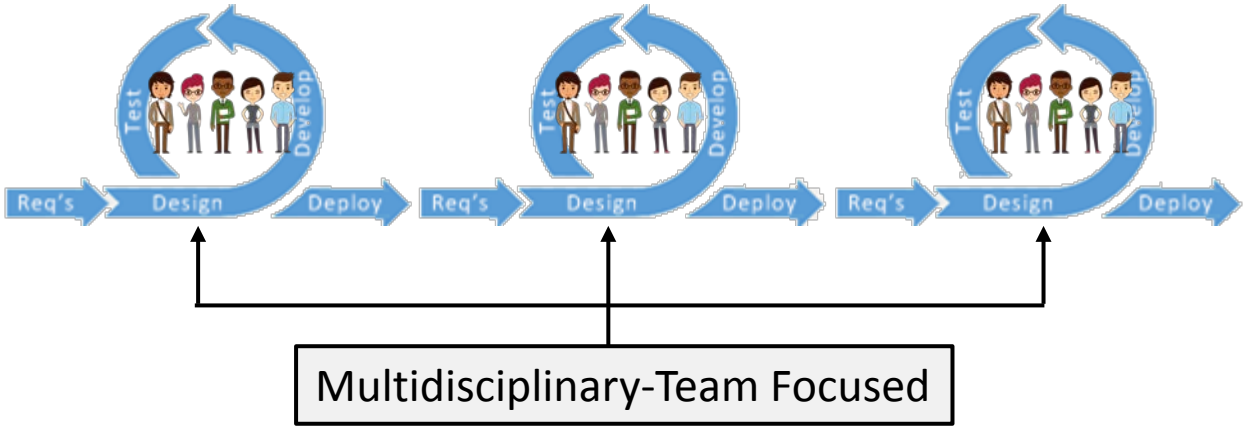


How Can Agile Development Help?

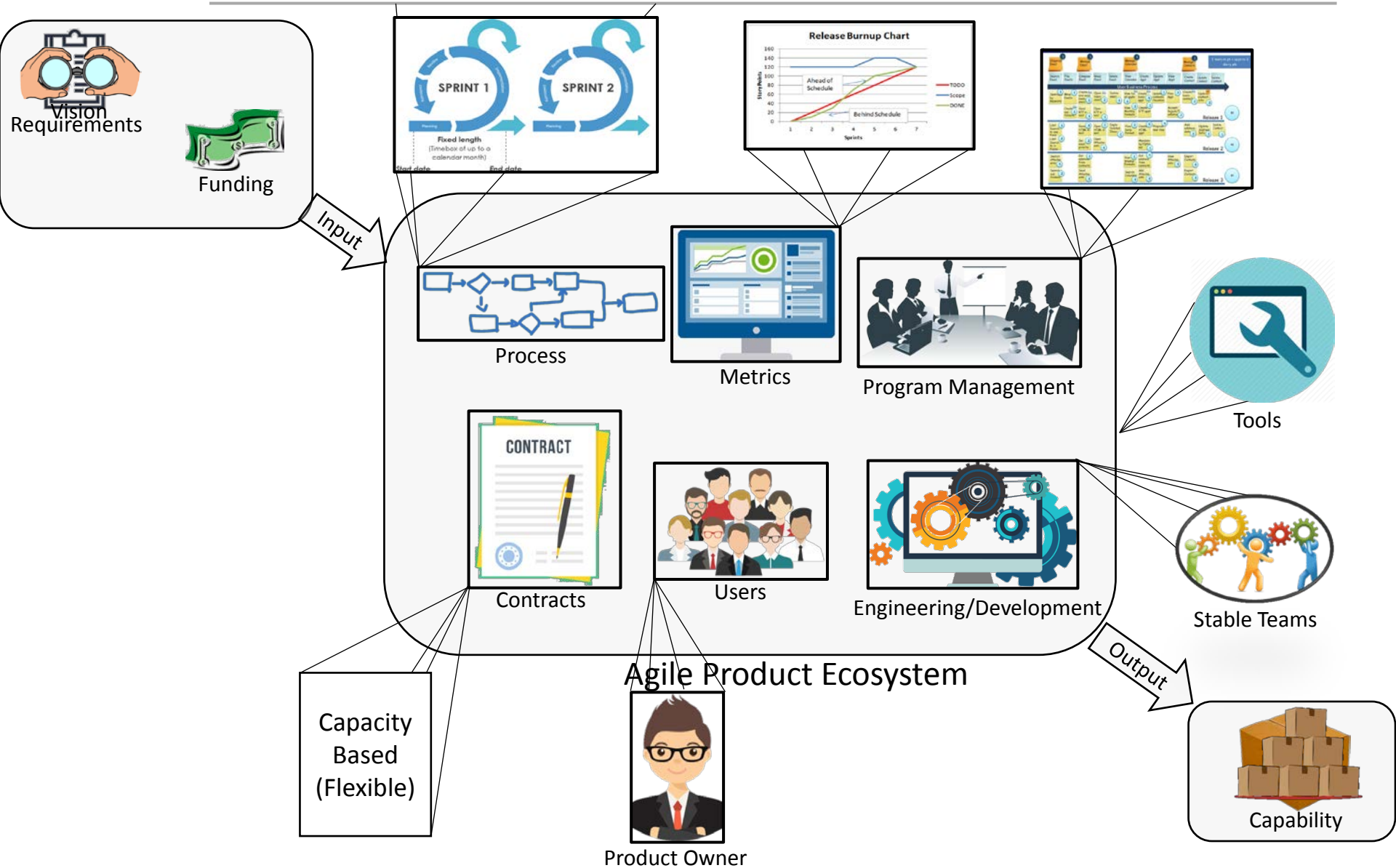
1) Requirements are continually (Re)assessed



2) <u>Stable</u> Teams

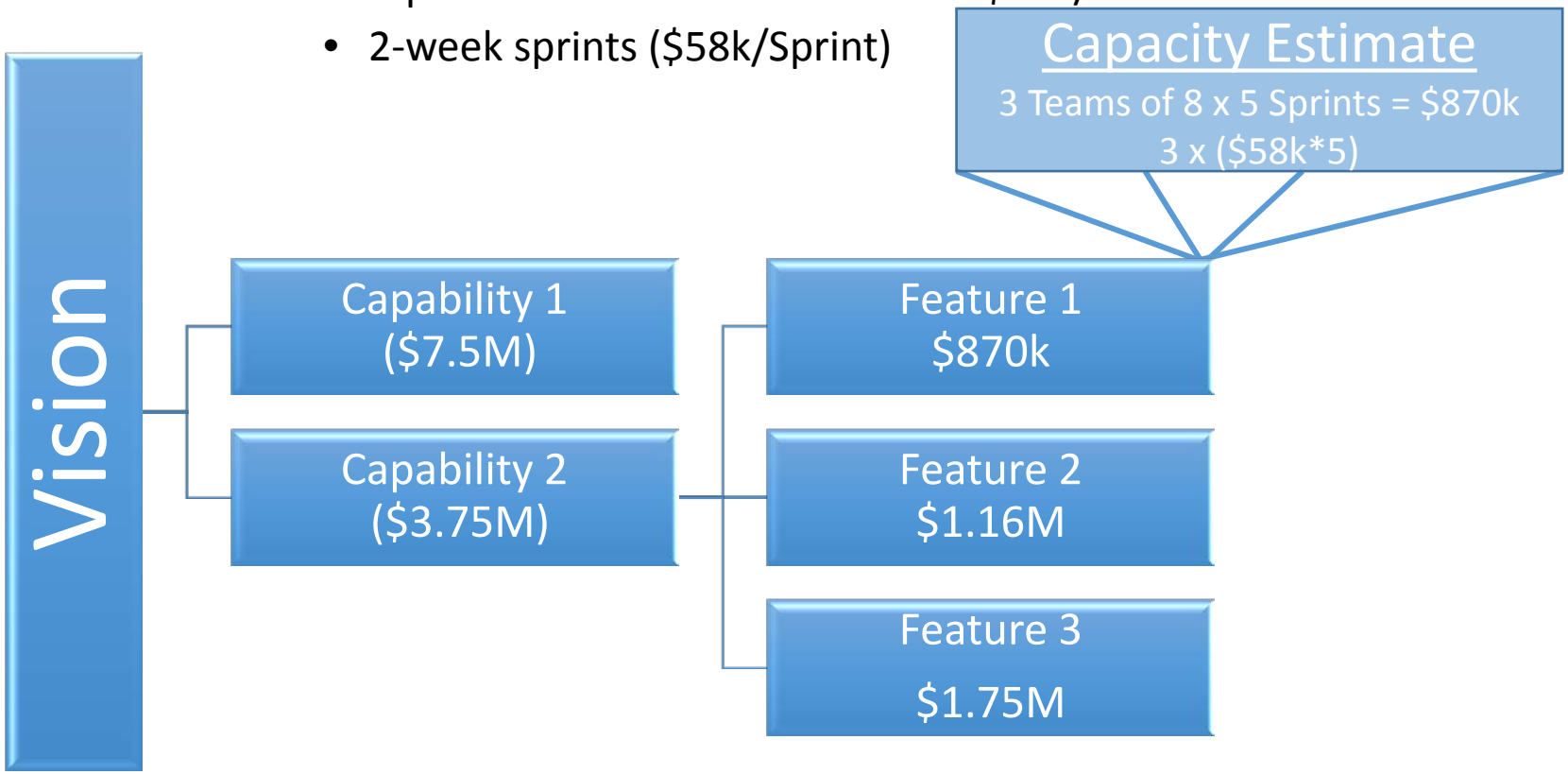


Agile Ecosystem



Capacity-Based Estimation

- #Team x (#Sprints x Burn-Rate of Team)
  - Estimation Data
    - 8-person team has a burn rate of $29k/Wk.
    - 2-week sprints ($58k/Sprint)



Tracking Project Progress (Feature 1)

- Refine estimates throughout development based on value delivered vs. estimate
  - Add Teams (Capacity)
  - (Re)move Teams (Capacity)
  - Do Not

Think Teams **NOT** Individuals

Initial Team Size	New Team Size	Iteration +1	Iteration +2	Iteration +3
6	7	–20%	–4%	+12%
6	7	0%	–6%	+15%
7	5	–12%	–8%	–8%
8	6	–20%	–20%	–16%
7	8	–15%		

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.)

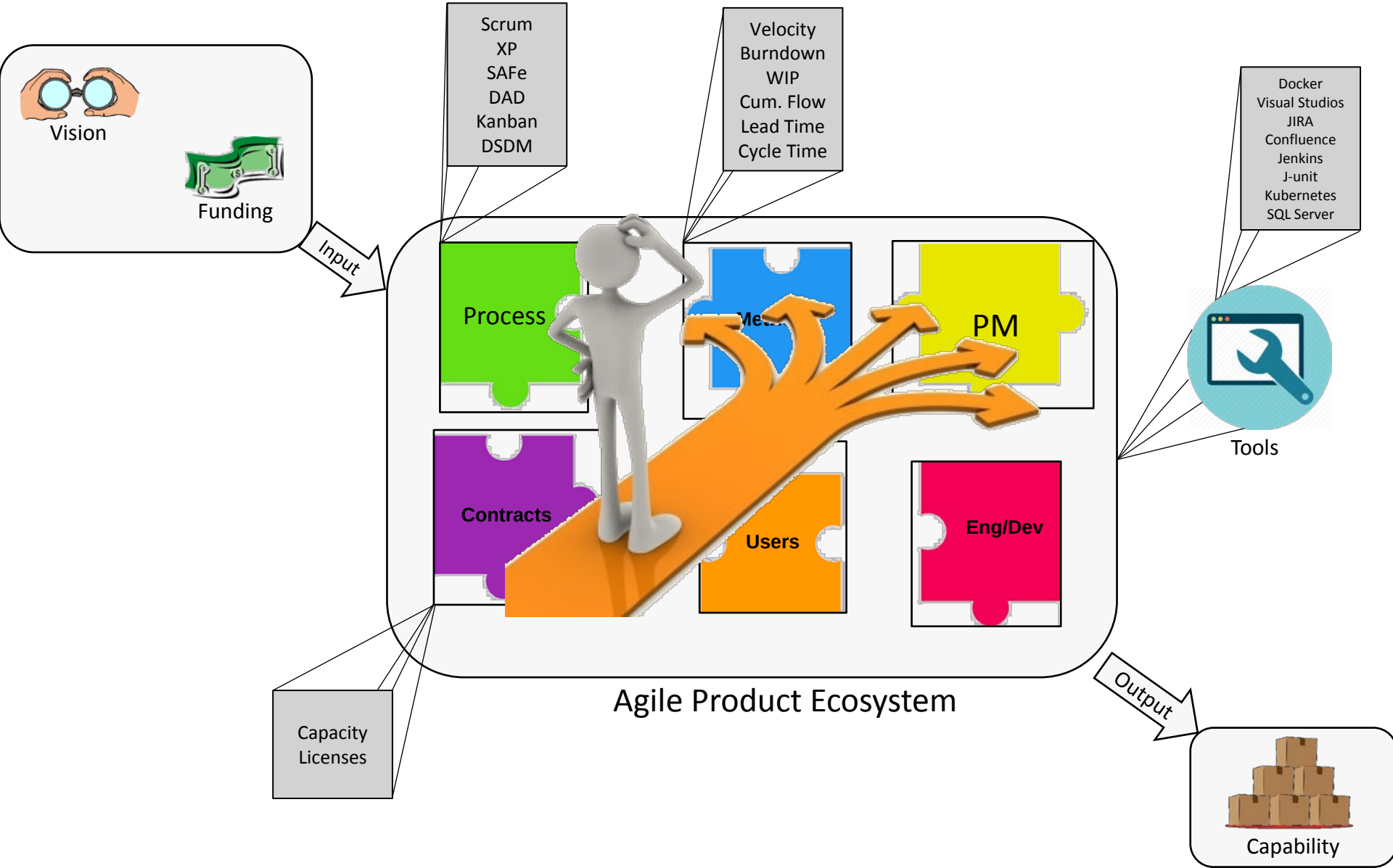
Matthew R. Kennedy, PhD

- 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

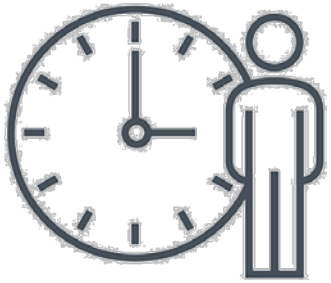
“New Start” Ecosystem



Comparison (Traditional vs. Agile)

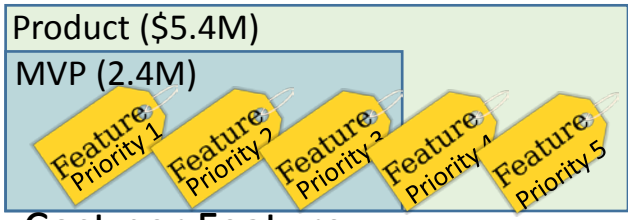
Individuals

Teams



Cost per Individual per Hour

Cost per Team per Sprint



Cost per Requirement per Phase

Cost per Feature

Adaptive Acquisition Framework

aka “The new 5000.02”

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

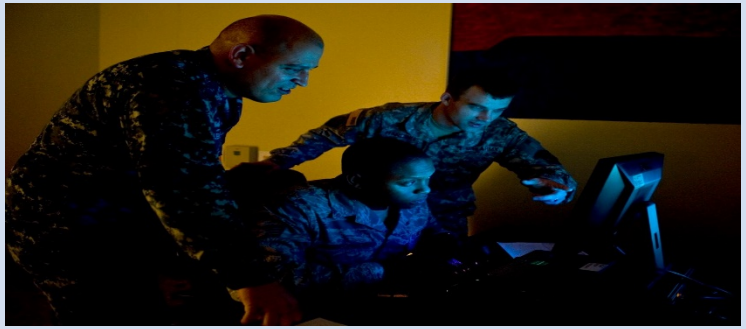
IT-CAST, 21 Aug 2019

FOUR LINES OF EFFORT

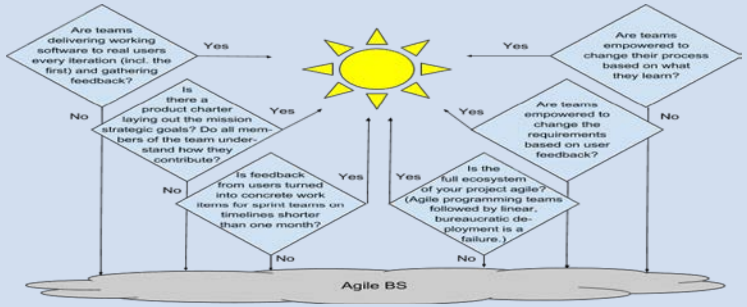
A. Refactor statutes, regulations, and processes for software

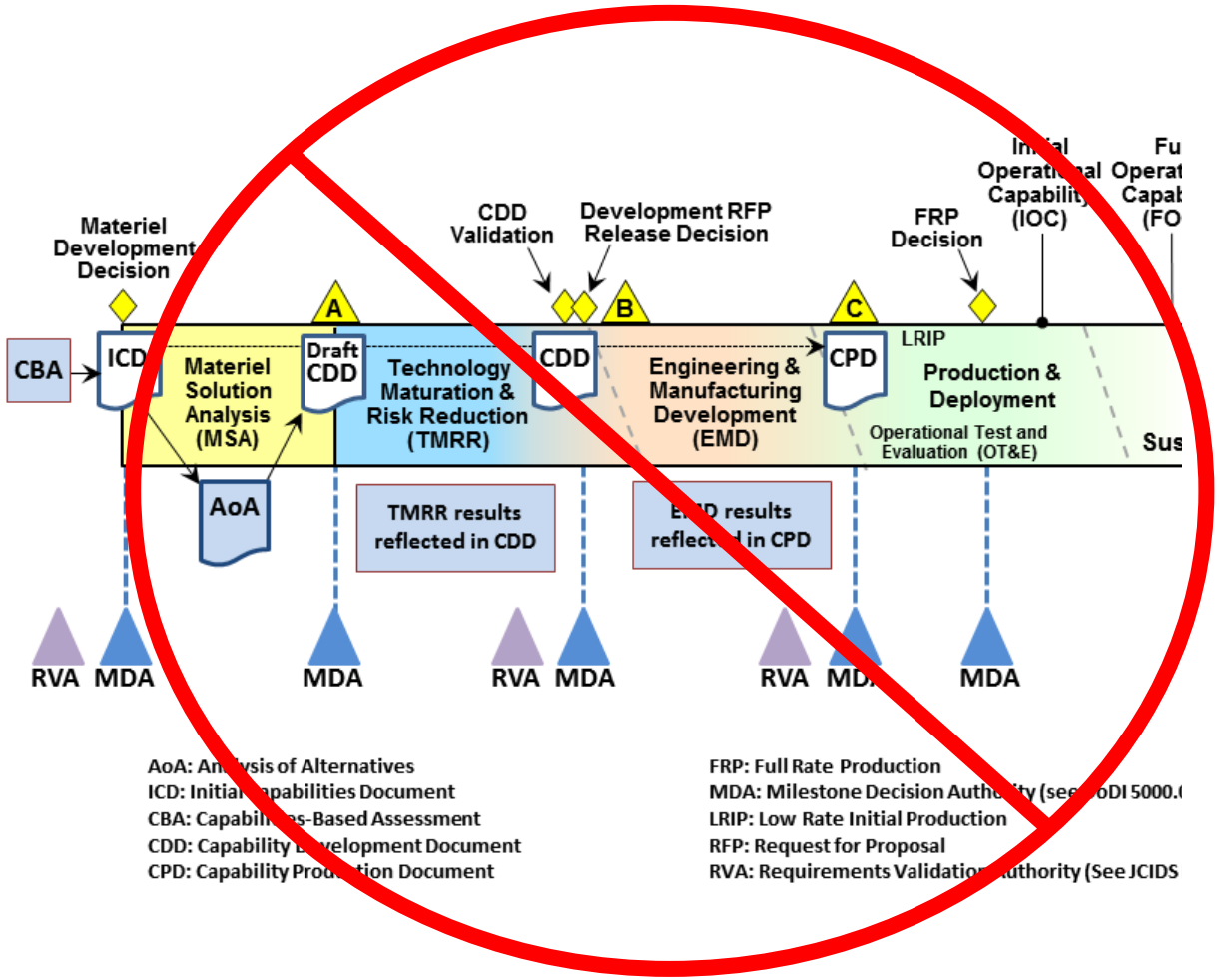
B. Create and maintain cross-program/ cross-service digital infrastructure

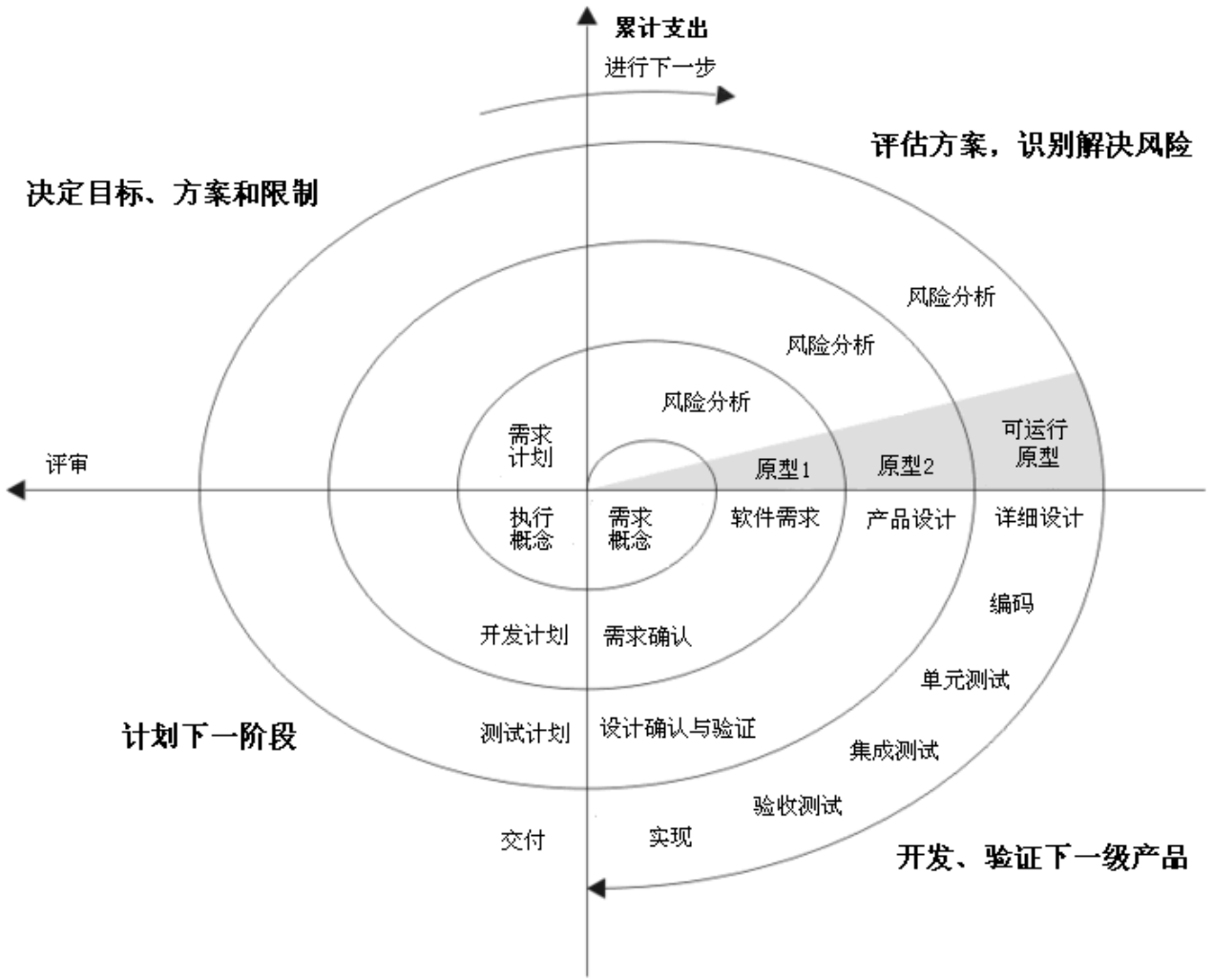
C. Create new paths for digital talent (especially internal talent)



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?

Integrated Defense Acquisition, Technology, and Logistics Life Cycle Management System

Following the Material Development Decision, the Milestone Decision Authority may authorize entry into the acquisition process at any point, consistent with phase-specific entrance criteria and statutory requirements

Material Solution Analysis Phase | Technology Development Phase | Engineering & Manufacturing Development Phase | Production & Deployment Phase | Operations & Support Phase

MDD | MS A | MS B | Post-CDR A | MS C | FRP DR

Joint Capabilities Integration & Development System (need-driven)

Oversight & Review

Contracting

Major Products

Logistics/ Sustainment

Defense Acquisition System (event-driven)

Technical

Financial Management

Planning, Programming, Budgeting & Execution Process (annual-calendar-driven)

Initiate Evolutionary Acquisition Strategy

Evolutionary Acquisition Strategy

Material Solution

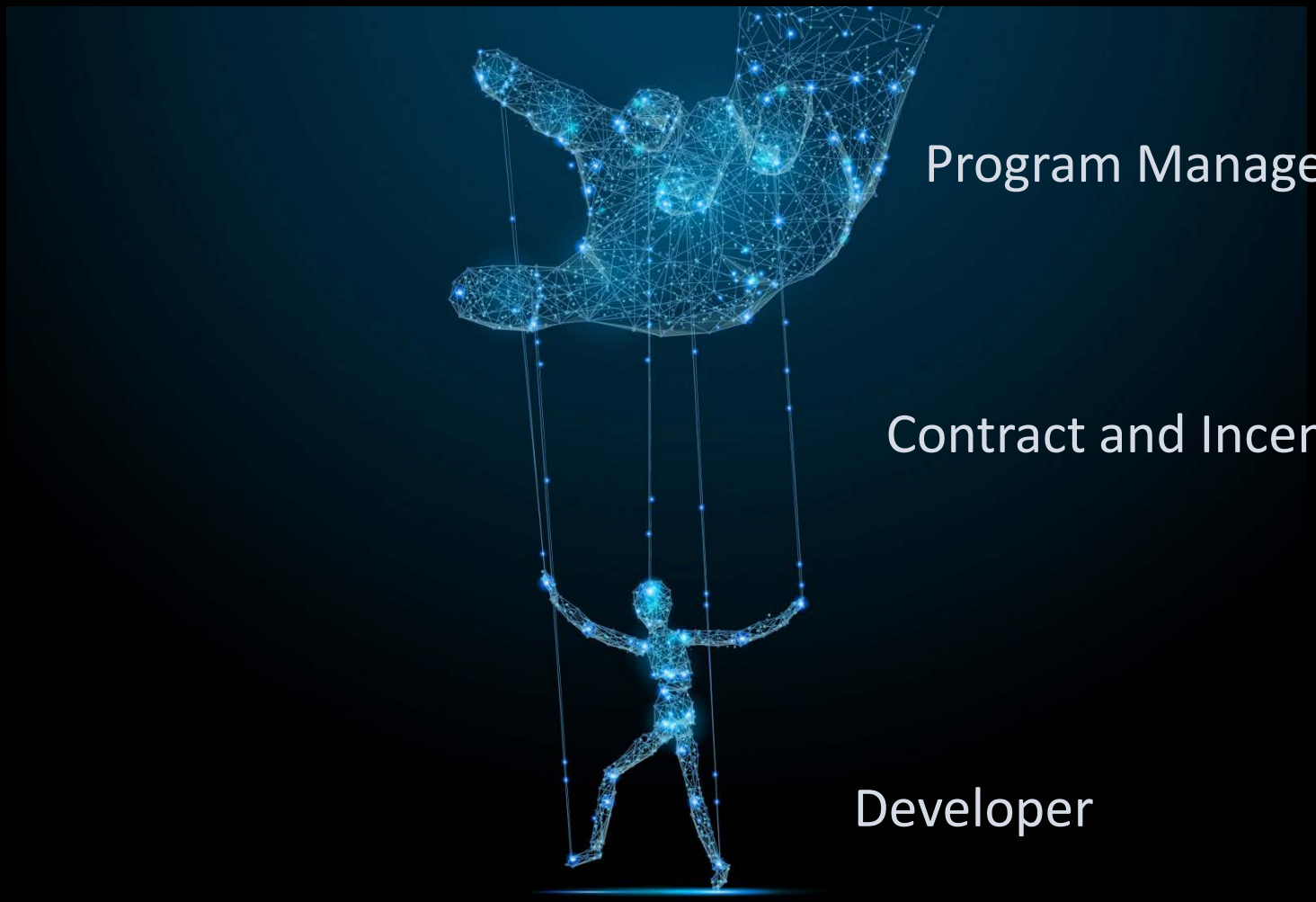
Prototypes

Production Representative Articles

Low-Rate Initial Production Systems

Full-Rate Production Systems

Appropriated Funds To Support Contracts

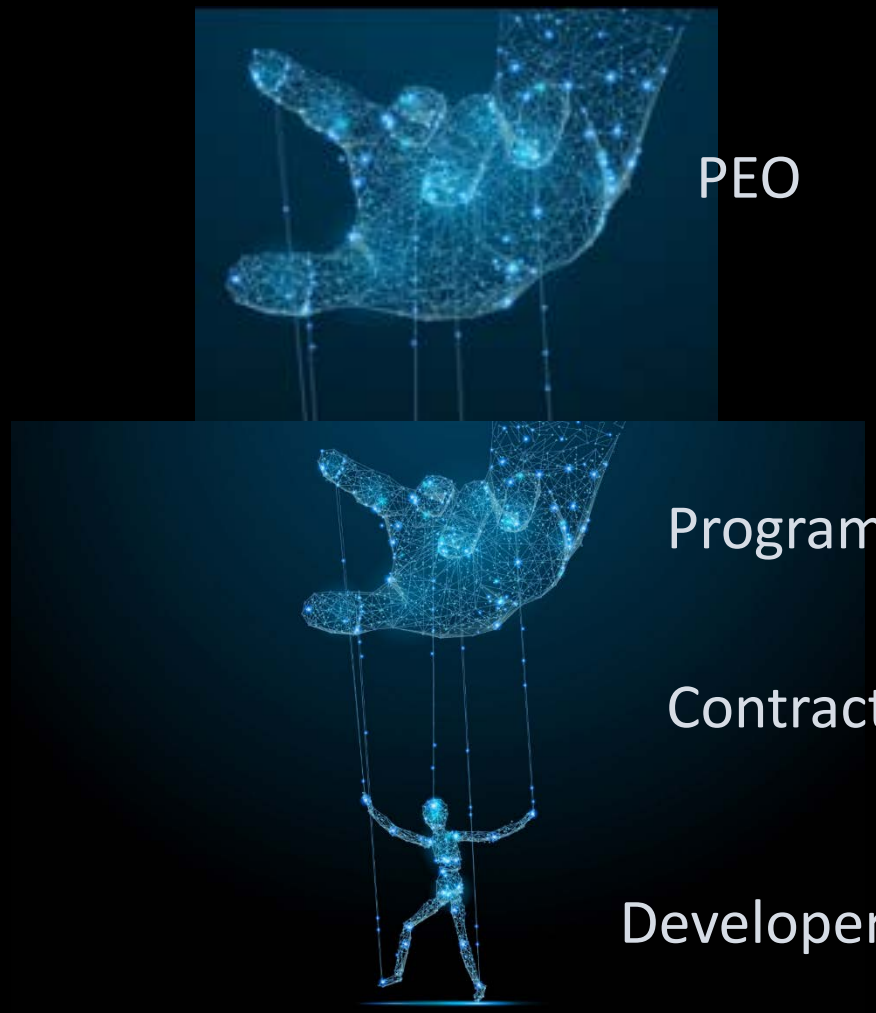


Program Manager

Contract and Incentives

Developer

Image source: https://psiloveyou.xyz/man-or-marionette-pinocchio-and-the-metamorphosis-of-manhood-f92ff2bf099c



PEO

Program Manager

Contract and Incentives

Developer

Image source: https://psiloveyou.xyz/man-or-marionette-pinocchio-and-the-metamorphosis-of-manhood-f92ff2bf099c

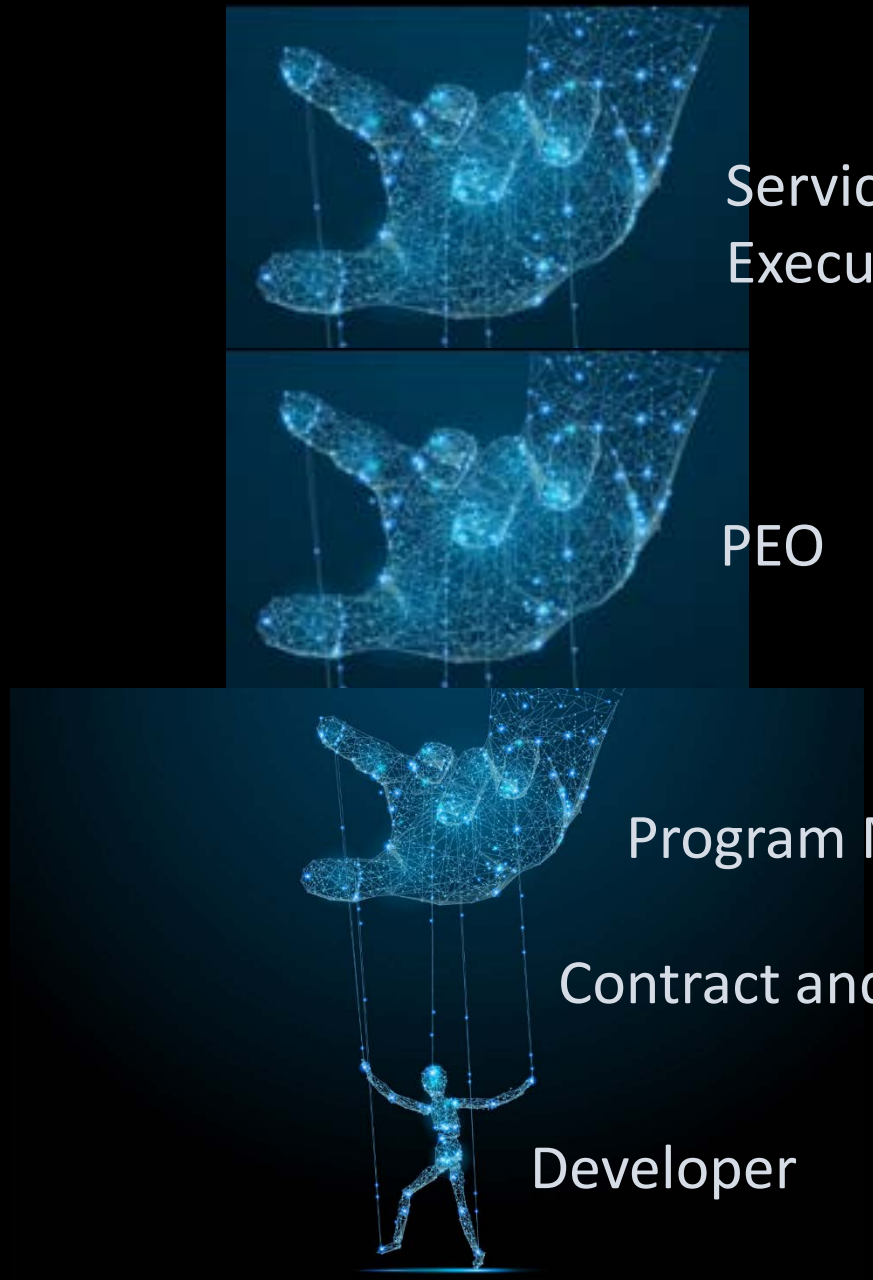
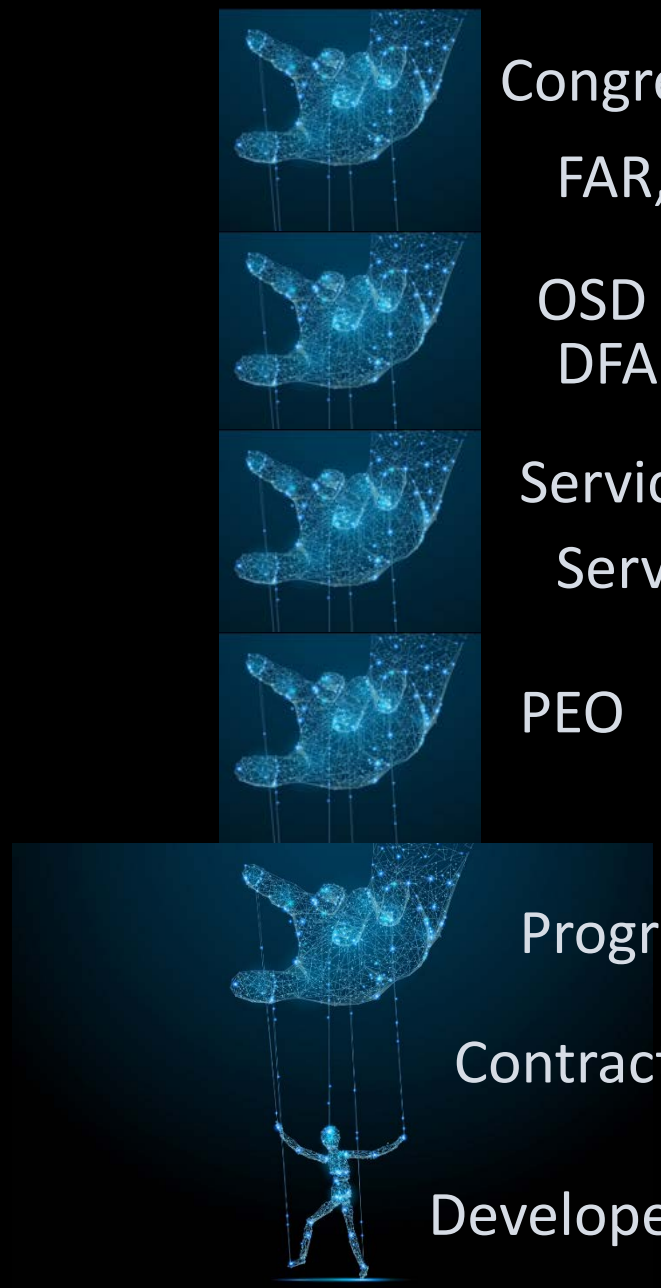


Image source: https://psiloveyou.xyz/man-or-marionette-pinocchio-and-the-metamorphosis-of-manhood-f92ff2bf099c



Congress

FAR, NDAA, Appropriations Bill, Statute

OSD

DFAR, 5000 series

Service Acquisition Executive

Service Acquisition Regulations

PEO

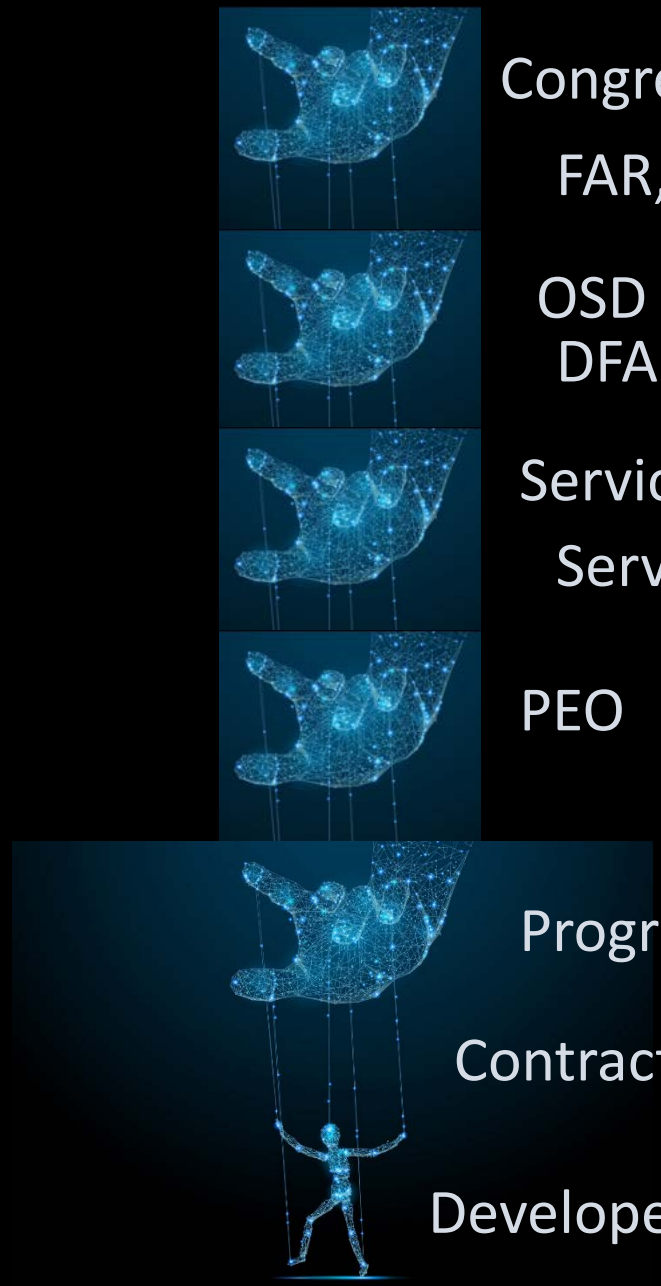
Program Manager

Contract and Incentives

Developer

Image source: https://psiloveyou.xyz/man-or-marionette-pinocchio-and-the-metamorphosis-of-manhood-f92ff2bf099c

Where is the Operational User?



Congress

FAR, NDAA, Appropriations Bill, Statute

OSD

DFAR, 5000 series

Service Acquisition Executive

Service Acquisition Regulations

PEO

Program Manager

Contract and Incentives

Developer

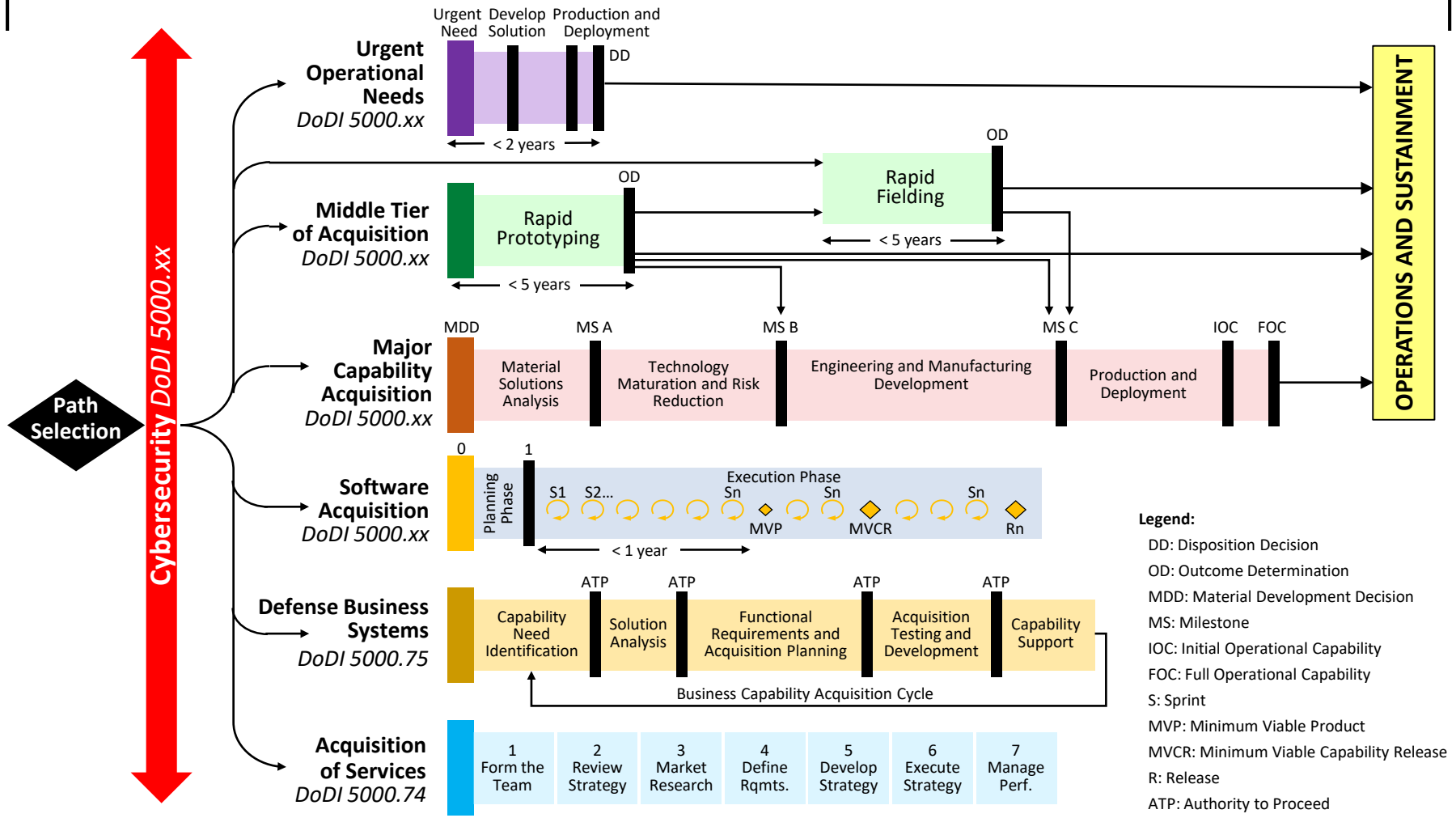
Image source: https://psiloveyou.xyz/man-or-marionette-pinocchio-and-the-metamorphosis-of-manhood-f92ff2bf099c

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

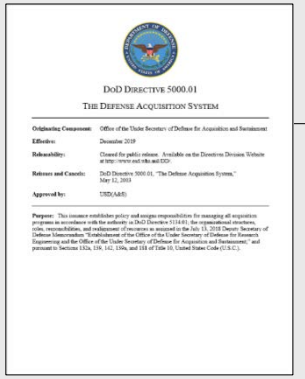
Current DoDI 5000.02

- **CORE A&S ACQUISITION POLICY** — A&S
  - Policy
  - Responsibilities
  - Procedures
  - Decision Points and Phases
- **FUNCTIONAL ENCLOSURES**

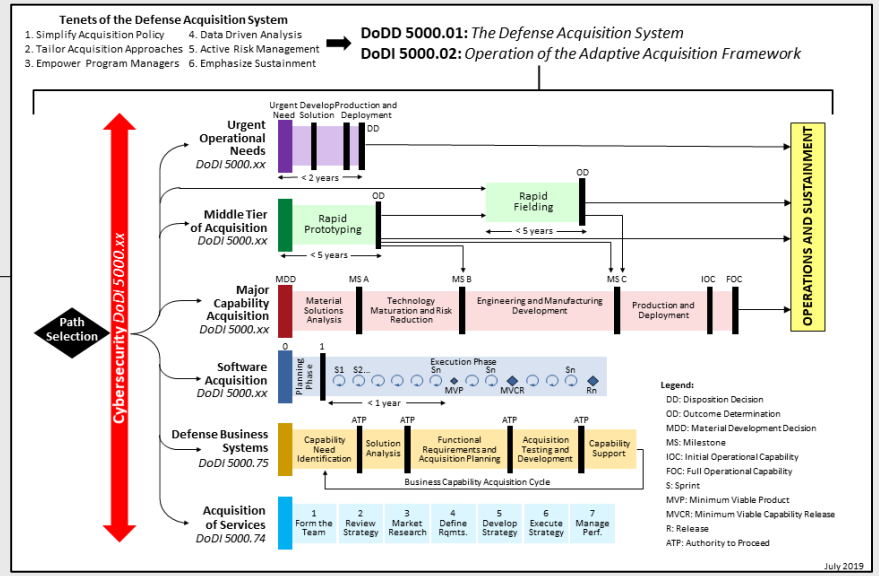
Acquisition Categories and Compliance Requirements	A&S
Program Management	A&S
Systems Engineering	R&E
Developmental T&E	R&E
Operational & Live Fire T&E	DOT&E
Life-Cycle Sustainment	A&S
Human Systems Integration	P&R
Affordability Analysis and Investment Constraints	A&S
Analysis of Alternatives	CAPE
Cost Estimating and Reporting	CAPE
Information Technology	CIO
Urgent Capability Acquisition	JRAC
Cybersecurity	R&E

Revised DoD Instruction 5000.02, Operation of the Adaptive Acquisition Framework

Revised DoD Directive 5000.01


DoD Directive 5000.01

The Defense Acquisition System

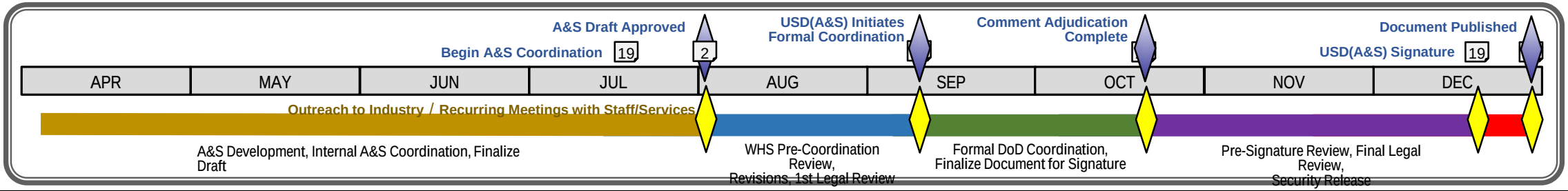


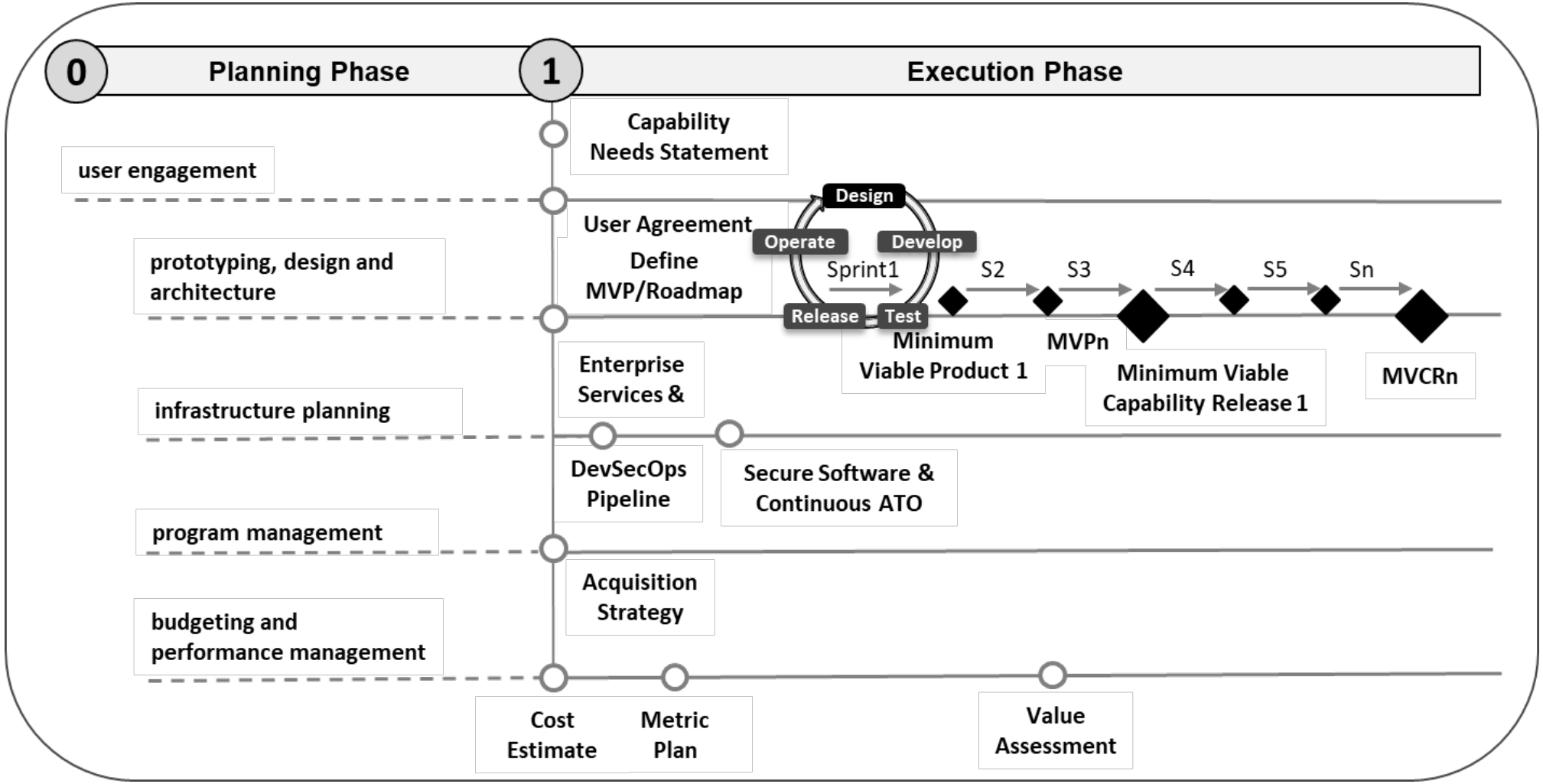
DAU Website

- DoD 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

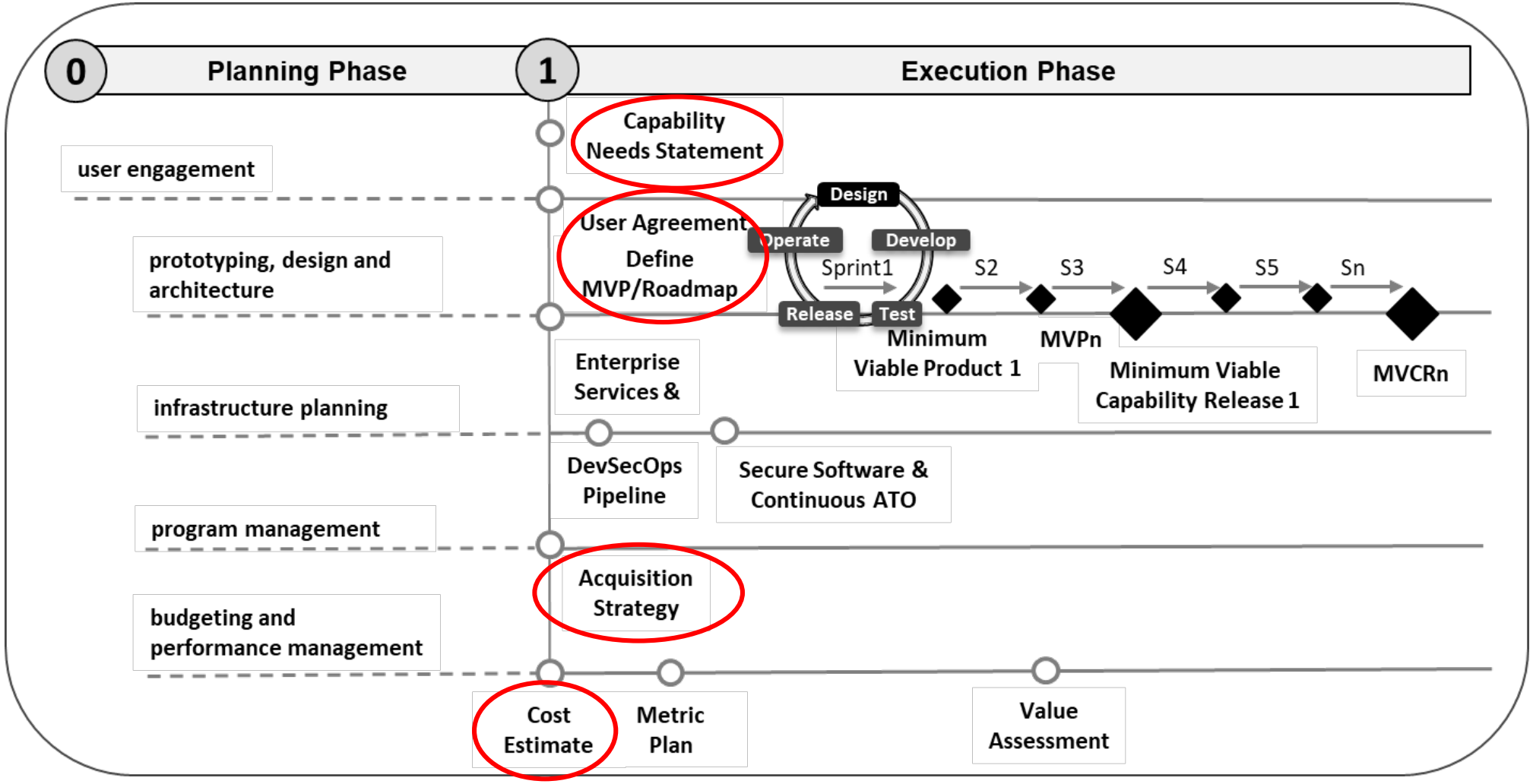
Separately Published Functional Policies

OT&E	DOT&E
DT&E	USD(R&E)
Systems Engineering	USD(R&E)
Human Systems Integration	USD(P&R)
AoAs	DCAPE
Cost Estimating	DCAPE
Information Technology	DoD CIO
Intelligence	USD(A&S)
Intellectual Property	USD(A&S)
Cybersecurity	USD(A&S) et.al.
Urgent	USD(A&S)

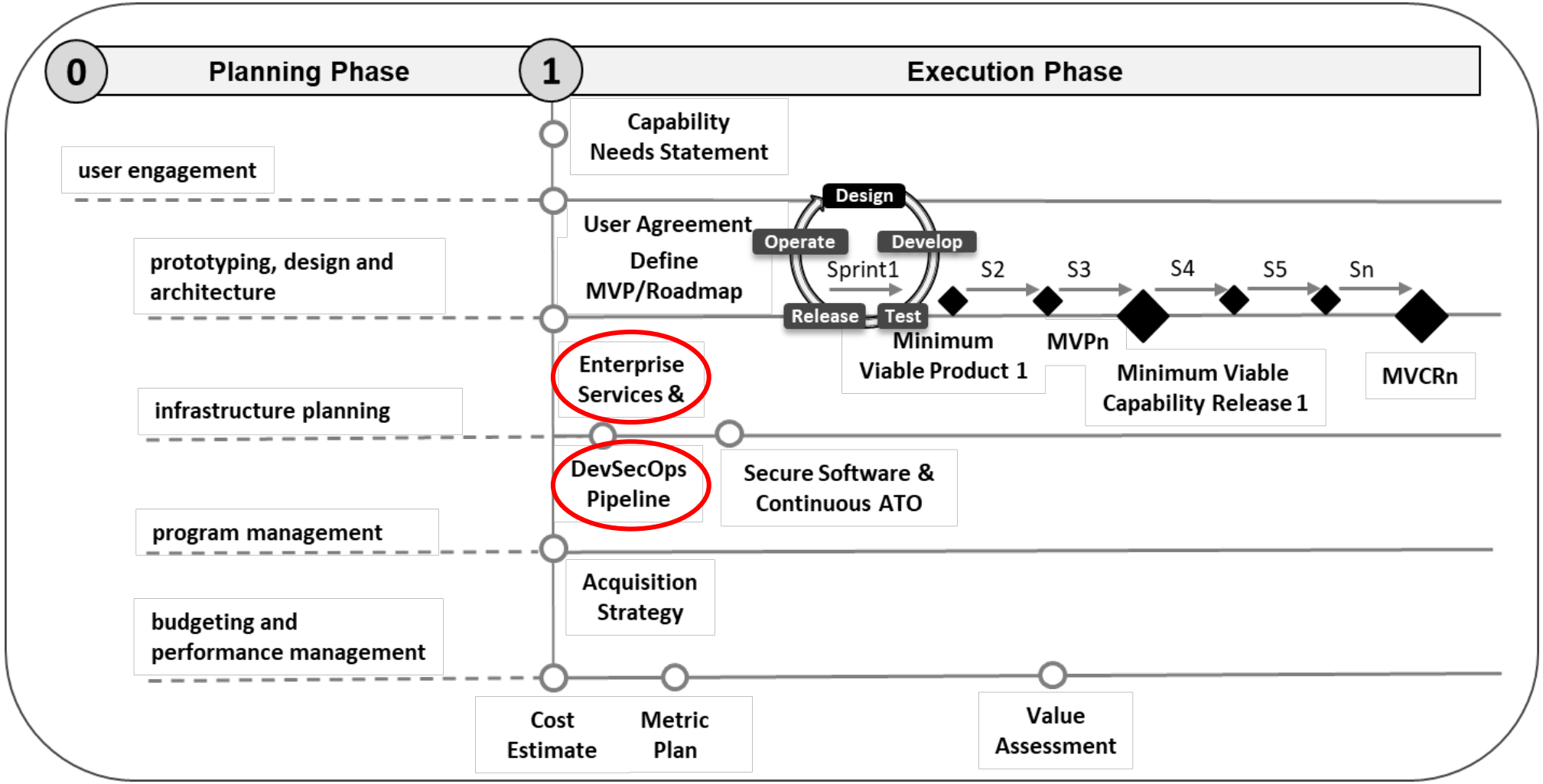




Software Acquisition Pathway – draft/pre-decisional



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 Assesment?

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

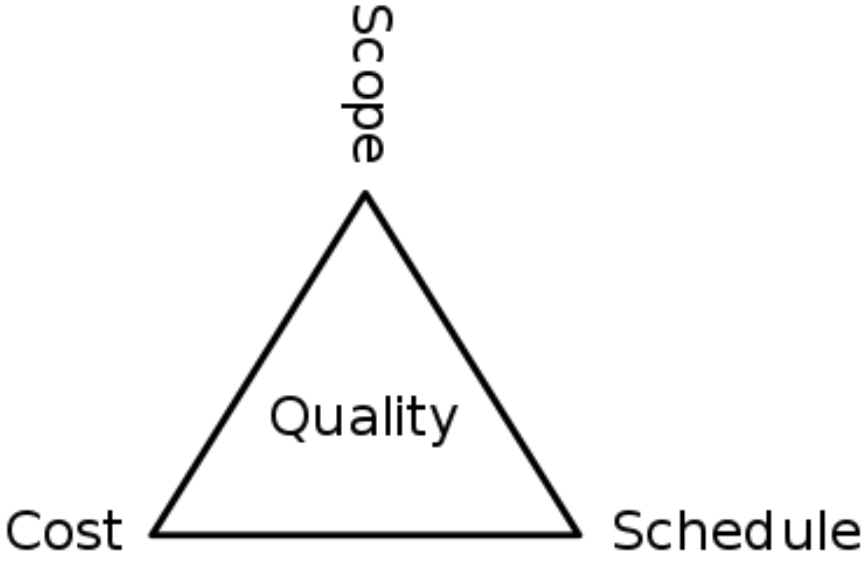
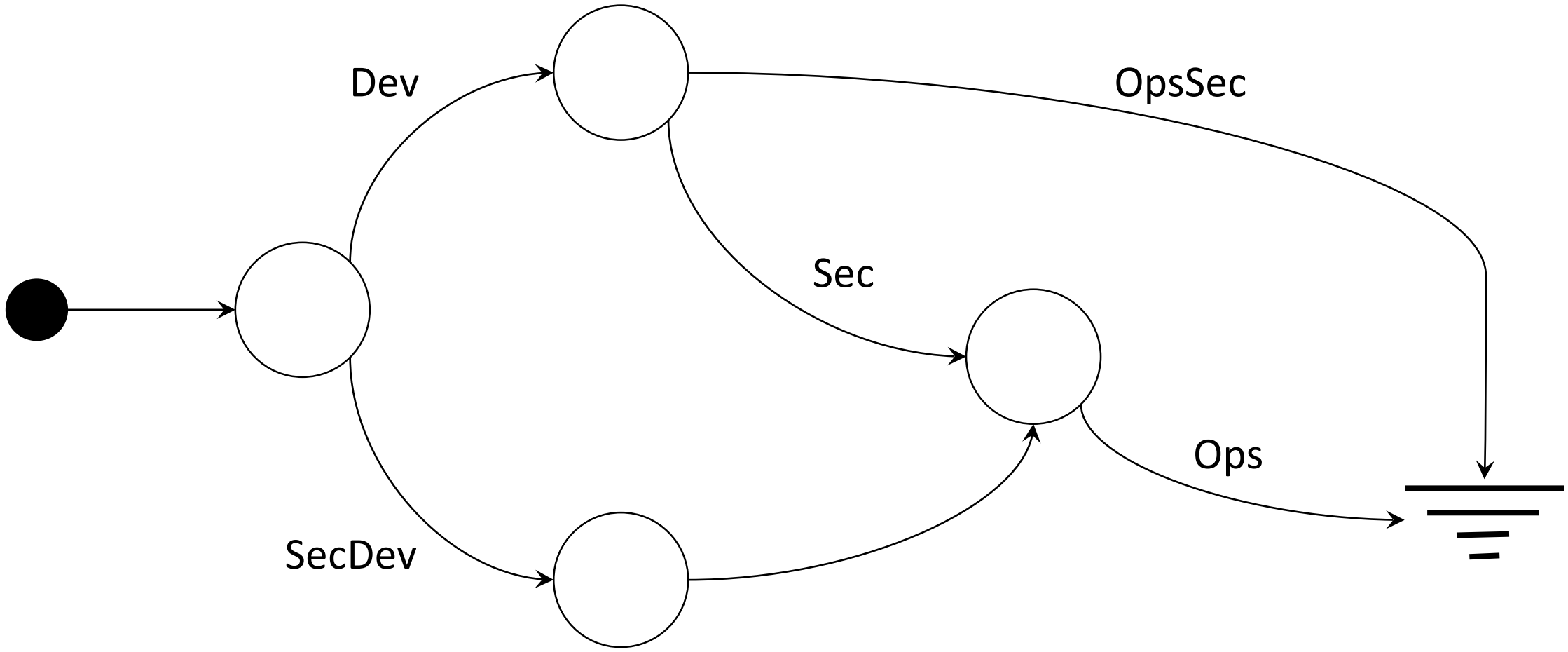


Image source: https://en.wikipedia.org/wiki/File:The-triad-constraints.svg

Enterprise DevSecOps



? [SecDevOps | DevSecOps | DevOpsSec] ?

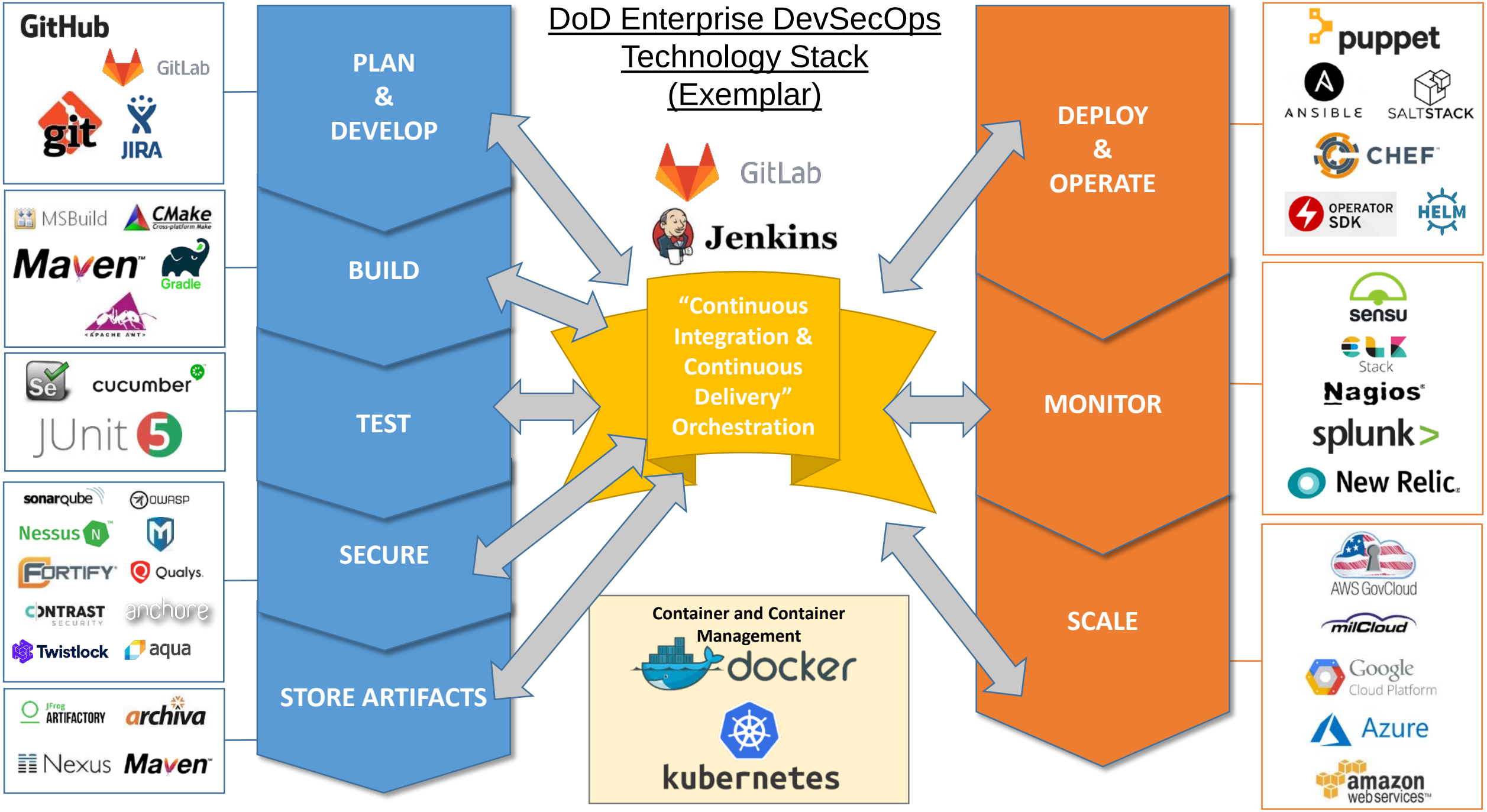
DevSecOps Model: Current

Planning	Development	Continuous Integration	Provision	Validation	Deployment	Operations
Customer Involvement	Architecture-First Approach	Automated Build	Base Image Provenance	Interface Validation	Deployment Orchestration	Continuous Monitoring
Issue Tracking	Version Control	Automated Unit Testing	Infrastructure Automation	Integration Testing	Small Releases	Practice Recovery
Threat Modeling	Coding Standards	Static Analysis	Instance Provisioning	Compliance / Accreditation	Canary Deployments	Upstream Feedback
Release Planning	Dependency Analysis	Code Quality Metrics	Credential Management	Chaos Engineering	Rolling Updates	
Sprint Planning	Observability	Release Packaging		Dynamic Analysis	Instant Rollback	
Sustainable Velocity	Test Driven Design			Vulnerability Scanning		
	Peer Review			Deployment Validation		

KEY
Existing
Added

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

DoD Enterprise DevSecOps Technology Stack (Exemplar)



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).

DoD Enterprise DevSecOps Technology

- 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.

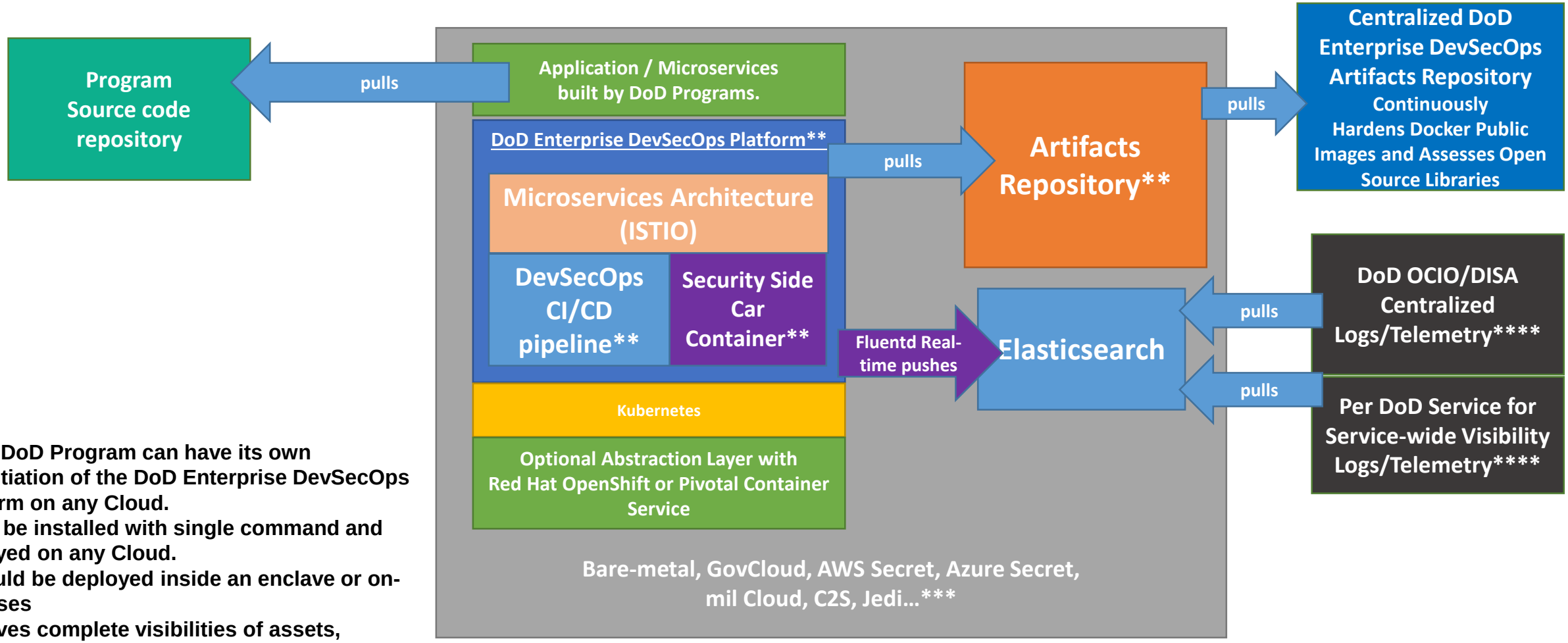
Cloud One: new Air Force Cloud Offering

- 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.

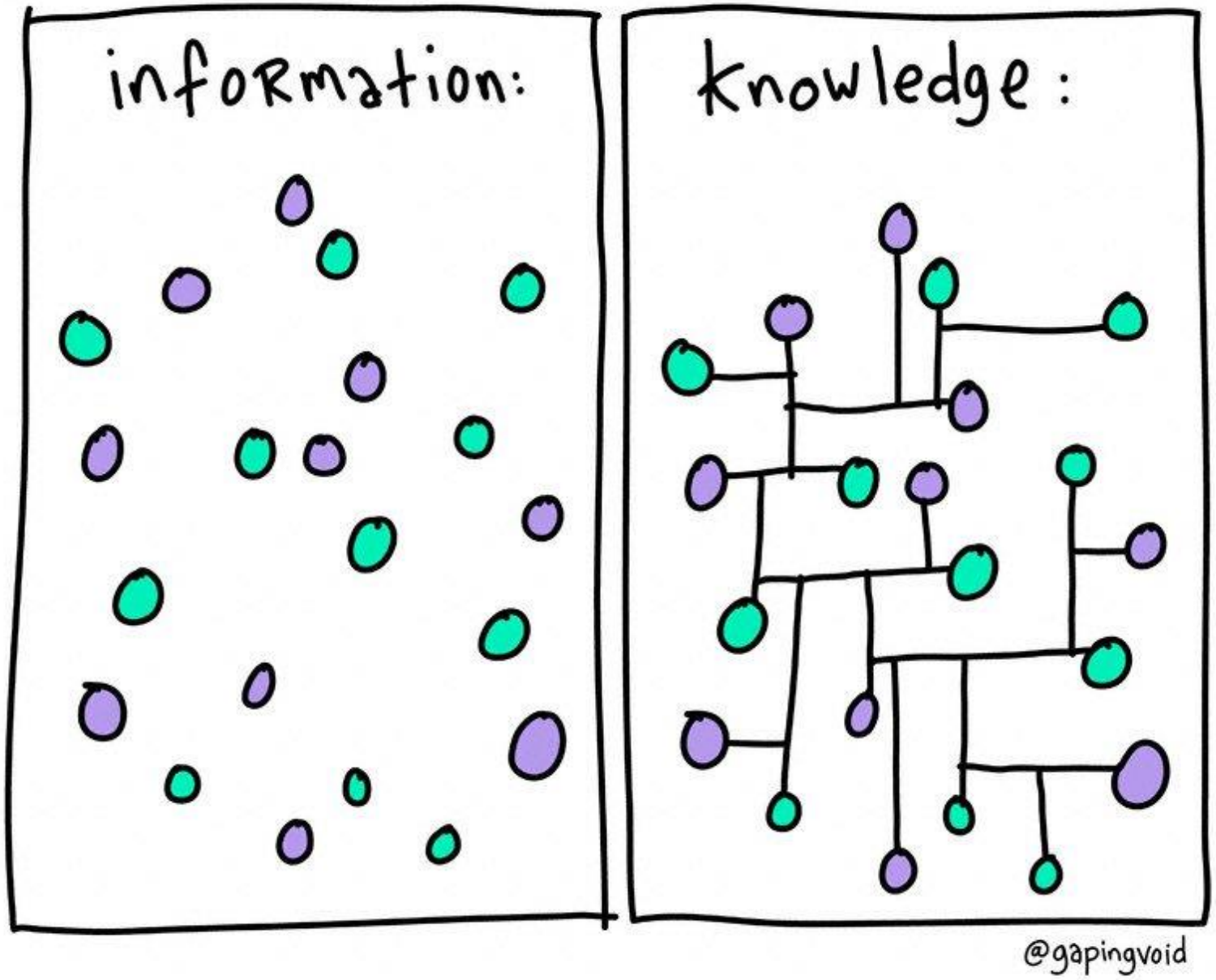
Questions and Feedback

A FOUNDATION FOR AGILE ACQUISITION DECISIONS

TORY CUFF

IT CAST 21 AUG 19

TESTED
TECH SOLUTIONS


information:
knowledge:
@gapingvoid

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.)

Hubbard, Douglas W. How to Measure Anything: Finding the Value of "Intangibles" in Business. III ed., Wiley, 2014.

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

A Strategy Driven Budget

https://comptroller.defense.gov/Portals/45/Documents/defbudget/fy2019/FY2019_Budget_Request.pdf

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

https://comptroller.defense.gov/Portals/45/Documents/defbudget/fy2019/FY2019_Budget_Request.pdf

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

Future Force Modernization Enterprise - Informs

Title 10 of the United States Code governs the organization, structure, and operation of the Armed Forces of the United States

SECDEF PRIORITIES
- Build a more lethal force
- Strengthen alliances and attract new partners
- Reform the DoD for greater performance and affordability

US Congressional Authorities

Art. I § 8: Financial dependency on Congress. Power implicit in the Constitutional power to make laws necessary and proper for executing its granted powers.

Procurement Laws: Title 5, 10, 31, 40, 41
T10 U.S.C. § 221: Future Years Defense Program
T10 U.S.C. § 2430: Major Def Acquisition Program
T10 U.S.C. § 2431a: Acquisition Strategy
T48 U.S.C. § 1 Federal Acquisition Regulation: Principal set of rules regarding govt. procurement in the US.

NDAA: Modification to multiyear procurement authority
NDAA 17: Secure U.S. Technological Superiority

https://dod.defense.gov/About/Office-of-the-Secretary-of-Defense/

Congressional Armed Service Committee

Senate Armed Services Committee
https://www.armed-services.senate.gov/hearings
- NDAA 2019; National Defense Strategy
- Accelerate New Technology to Meet Emerging Threats
- Posture of the Department of the Army
- Army Modernization
- Department of Defense Acquisition Reform Efforts
- All Arms Warfare in the 21st Century
- Reshaping the U.S. Military

House Armed Services Committee
https://armedservices.house.gov/hearings
- NDAA 2019; National Defense Strategy
- Readiness Subcommittee on Army Futures CMD: Army Futures Command: Will it help? ✓
- Military Technology Transfer: Threats, Impacts, and Solutions for the Department of Defense ✓
- Army FY 2019 Budget Request Readiness Posture
- Ground Force Modernization Budget Req for FY 2019
- Oversight and Reform of the DoD 4th Estate
- Promoting DOD's Culture of Innovation ✓
- A Review and Assessment of FY 2019 Budget Request for DoD Science and Technology Programs
- Assessing Military Service Acquisition Reform
- Readying the U.S. Military for Future Warfare
- The Army's Tactical Network Modernization Strategy

Additional Congressional Committees
Budget (H/S)
Oversight & Government Reform (H)
Science, Space, and Technology (H)
Technology Modernization Subcommittee (H)
Joint Select Committee on Budget and Appropriations Process Reform

US Congressional Budget

Appropriations (H/S)
- Budget Control Act
- President's Budget approved by Congress
- 2019 National Defense Authorization Act: $10.2 B
- DoD Appropriations Act, 2018
- Wasteful Defense Spending Reduction Act of 2018
- Military Infra Consolidation and Efficiency Act of 2017

FY2020 strategy-driven budget that integrates plans to transition from technology demonstrations to develpmt and fielding, with outcomes and timelines to dominate a new era of great power competition.

Independent Federal Advisory Committees to the SECDEF

Federal advisory committees have been called the "fifth arm of government" because of the significant role they play in advising federal agencies, the Congress, and the President on important national issues

- Bioengineering Sciences & Technologies Integrated Review Group
- Advisory Committee on Emerging Markets
- Visiting Committee on Advanced Tech
- Advisory Committee on Industrial Security and Industrial Base Policy
- Defense Trade Advisory Group
- Research, Eng, and Dev Advisory Comm
- Government-Industry Advisory Panel
- Threat Reduction Advisory Committee
- Basic Energy Sciences Advisory Comm
- US Army Science Board
- Advanced Scientific Computing Advisory Committee
- Biological and Environmental Research Advisory Committee
- National Geospatial Advisory Committee
- Advisory Committee on Public-Private Partnerships
- HS Science & Technology
- Industry Trade Advisory Committee on Intellectual Property Rights
- Industry Trade Advisory Committee on Small and Minority Business
- Information Sys Technical Advisory Comm
- Industry Trade Advisory Committee on Information and Comm Technologies, Services, and Electronic Commerce
- Renewable Energy and Energy Efficiency Advisory Committee
- National Medal of Technology and Innovation Nomination Evaluation Comm
- National Technical Information Service Advisory Board
- Advisory Committee on Emerging Markets
- Defense Digital Services
- Defense Science Board
- Defense Business Board

Secretary of Defense OSD

SECDEF: Jim Mattis
DEP SEC: Patrick M. Shanahan
Chief Mgmt. Officer: John H. Gibson II

USD for Acquisition, Technology and Logistics (AT&L)
Dissolved – NDAA 2017

New FY 2017 NDAA, PL 114-328
1 USD for Research & Engineering (R&E)
2 USD for Acquisition and Sustainmt (A&S) (USD(R&E)) & (USD(A&S))

Legacy R&D and Acquisition
USD for Research and Engineering
USD Comptroller/Chief Financial Officer
USD for Personnel and Readiness (P&R)
ASD for Acquisition (ASD(A))
ASD for Logistics and Materiel Readiness (L&MR)
ASD for Nuc & Chem & Biological Defense Programs
ASD for Research & Engineering ASD(R&E)
Dir. of OSD Cost Assmt and Program Eval (CAPE)
Director of Operational Test and Evaluation (DOT&E)
Deputy ASD Manufacturing and Industrial Base Policy

OUSD R&E — USD Michael D. Griffin, DUSD Dr. Lisa Porter
USD Dr. Ellen M. Lord, DUSD Alan R. Shaffer

Office of Force Transformation
In 2006, the Office of Force Transformation was disestablished, and its functions spread between the Office of the Under Secretary of Defense for Policy, and the Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics.

Defense Entrepreneur Forum's
A network of emerging defense leaders, civilian innovators, and social entrepreneurs who promote a culture of innovation and act upon transformational ideas that address national security challenges.

Close Combat Lethality Task Force (IN TF)
Close Combat Strategic Portfolio Review led by CAPE "to identify promising investment opportunities to improve close-combat effectiveness and survivability" to 2029. Six Topics: Sensing, Comms, Mnvr, Attack, Survival, Sustainability

Joint Artificial Intelligence Center (JAIC)
Reports to DoD CIO. Accelerate results by coordinating work on AI National Mission Initiatives, and increase collaboration with the private sector, academia and military allies on the developmnt of AI capabilities.

Joint Improvised Threat Defeat Organization (JIDO)
Enables the Department of Defense to counter improvised devices and emerging improvised threats in order to support Combatant Commands and deployed war fighters.

Federal Advisory Committees Panels and Boards

Defense Innovation Board
Provide SECDEF and senior leaders with independent advice and recommendations on innovative means to address future challenges through three focus areas: People and Culture, Technology and Capabilities, and Practices and Ops.

Defense Innovation University
Provides non-dilutive capital in exchange for commercial products that solve national defense problems. Contract with companies offering solutions in a variety of areas to solve a host of defense problems.

Defense Acquisition University
In-class corporate university for Defense Acquisition Workforce. Provide a global learning environment to develop qualified acquisition, requirements and contingency professionals to deliver effective and affordable warfighting capabilities.

Technology Innovation Program Advisory Board
NIST A body of experts in the field of technology innovation from high-tech companies, VCs, and univ. Accelerates inovation by offering competitive opportunities for cost-shared funding of high-risk research with potential of transformational results.

Proposal Review Panel for Civil, Mechanical, and Manufacturing Innovation
Advances future of manufacturing, the design of innovative materials and bdg technologies, infrastructure resilience and sustainability, and tools and systems for decision-making, robotics and controls.

Proposal Review Panel for Industrial Innovation and Partnerships
Invests in high-tech small businesses and collaborations between academia and industry to transform discoveries into innovative commercial technologies with societal benefits.

Emerging Tech & Research Advisory Committee
Representatives from industry, academic land U.S. Govt. representing diverse points of View from firms producing a range of goods, technologies, and software controlled for national security, non-proliferation, and foreign policy reasons.

National Advisory Council on Innovation and Entrepreneurship
EDA Identify and recommend solutions to critical issues driving the innovation economy; enable entrepreneurs and firms to successfully access and develop a skilled, globally competitive workforce.

Joint Staff J6 Joint Deployable Analysis Team
Analyzes current and future warfighter systms and procedures in field conditions to address Joint integration and interoperability issues to improve combat effectiveness and reduce fratricide and collateral damage.

Defense Systems Information Analysis Center
DSIAC A component of DoD's Information Analysis Center (IAC) enterprise. Generate, collect, analyze, synthesize, and disseminate Scientific and Technical Information (STI) to DoD and Federal government users and industry contractors.

Defense Innovation Marketplace
The Defense Innovation Marketplace was created to be a consolidated resource for both DoD and industry to help enable communication between industry and government.

Rapid Innovation Fund
Designed to transition small business technologies into defense acquisition programs. The program is designed to fund mature technology ideas.

Chief Technical Officer — Leading technology innovation at DoD to solve critical war-fighting problems, and to delivery technology solutions faster. Advance Technology & Innovation.

Report to ASD R&D

Defense Advanced Research Projects Agency (DARPA)
Responsible for the develpmt of emerging technologies for use by the military.
Adaptive Execution Office (AEO)
Defense Sciences Office (DSO)
Information Innovation Office (I2O)
Strategic Technology Office (STO)
Tactical Technology Office (TTO)
Biological Technologies Office (BTO)

BTO DSO I2O MTO STO TTO

Missile Defense Agency
Develop, test, and field integrated, layered, ballistic missile defense system (BMDS) to defend the United States, its deployed forces, allies, and friends against all ranges of enemy ballistic missiles in all phases of flight.
Functional & Executional staff components consisting of program managers, technology directors, engineering, contracting, sales, and advisors.

Strategic Intelligence Analysis Cell
New organization focused on understanding the capabilities and vulnerabilities of potential adversaries, assessing U.S. capabilities, tracking global technology trends, assessing emerging threats, and identifying potential opportunities that warrant action and merit investment.
Steer investments toward the most urgent technology needs and opportunities.

Director Chris Shank

Strategic Capabilities Office
Dep Sec Def Ash Carter est. as an office to find new applications for existing technologies. Rapidly develop a portfolio of capabilities to counter near-peer adversaries and improve the U.S. posture in engaging future threats.

Report to ASD Advanced Capabilities

Director Michael Brown

Defense Innovation Unit (DIU)
Accelerates adoption of leading commercial technology into the military. Implementing commercial solutions that can transform how the Defense Department operates.
Provides non-dilutive capital in exchange for commercial products that solve national defense problems.
Mountain View CA, Silicon Valley, CA; Austin TX; Boston MA; US Pentagon

THIRD OFFSET STRATEGY
Cyber and space capabilities, unmanned systems, directed energy, undersea warfare, hypersonics, and robotics, amongst others

Rapid Prototyping, Speed & Scale, Advance Technology, Accountability & Transparency, Speed of Relevance, Perpetual Motion, Unity of Command & Effort, Take Risks, Fail Early & Cheaply, Dominance to Lethality, Deliver Concepts & Requirements, Responsible Stewardship, FFME Relationships, Enhance Learning, Condense Dispersal of Units, Advance Innovation — *Directs*

FY 2017 NSS
FY 2017-18 NDAA

Defense Technical Information Center
Joint Reserve Dir. (JRD)

Director of Defense Research and Engineering for Research & Technology (ASD)
Sets strategic technical direction and investment strat to ensure tech battlefield dominance, integrate DOD's laboratory infrastructure, and provide stewardship of the tech community that conducts defense research.
- Asst. Director for Microelectronics (ME)
- Asst. Director for Cyber
- Asst. Director for Quantum Science (QS)
- Asst. Director for Directed Energy (DE)
- Asst. Director for Machine Learning & AI
- Dep Dir. for Research Technology & Laboratories
- Dep Dir. for Strategic Technology Protection & Exploitation
- Defense Microelectronic Activity (DMEA)

Director of Defense Research and Engineering for Advanced Capabilities (ASD)
Prototyping and experimentation designed to increase understanding of tech and its capabilities, drive down technical risk, and incorporate warfighter feedback.
- Asst. Director for Networked C3
- Asst. Director for Space
- Asst. Director for Autonomy
- Asst. Director for Hypersonics
- Dep Dir. for Mission Engineering & Integration
- Dep Dir. for Development Test & Evaluation
- Test Resource Mgmt. Center (TRMC)

- DASD Emerging Capability & Prototyping
- Dep Dir Prototyping & Experimentation
- DASD Systems Engineering

Critical Legacy Capability

MG Chris F. Bentley

Joint Capability Technology Demonstration (JCTD)
Conducts joint and combatant command warfighting needs through the execution and demonstration of prototypes within 2 to 4 years
- Rapid Reaction Technology Office
- Comparative Technology Office; - Initiatives & Analysis

Rapid Reaction Technology Office
Develops risk-reducing prototypes and demonstrations of land, sea, and air systems that address mission-focused combatant command, joint-Service, and interagency capability needs to counter emerging threats.

OUSD R&E — **Reorg in Summary**
Over the last several years, policymakers and others have expressed concern that the long-held technological edge of the U.S. military is eroding due, in part, to the proliferation of technologies outside the defense sector, organizational and cultural barriers to DOD effectively incorporating and exploiting commercial innovations, and insufficient engagement with leading-edge companies that have not historically been a part of the DOD innovation system. The establishment of the USD (R&E) as the fourth highest ranking DOD official—behind the Secretary, Deputy Secretary, and Chief Management Officer—was intended to promote faster innovation and to reduce risk-intolerance in the pursuit of new technologies.

DODD 5000.01 The Defense Acquisition System 2003
DODI 5000.02 Operation of the Defense Acquisition System 2015
AR 73-1 Test And Evaluation Policy 2018
AD 2017-22 Implementation of Acquisition Reform Initiatives 1 and 2 2017
AD 2017-29 Acquisition Reform Initiative #3: Improving The Integration and Synch of Science and Technology 2017
AD 2017-30 Acquisition Reform Initiative #4: Streamline Test and Evaluation and Minimizing Redundant Testing 2017
AD 2017-31 Acquisition Reform Initiative #5: Align Sust Policy to Foster Cost Efficiency and Improve Readiness 2017
AD 2017-32 Acquisition Reform Initiative #6: Streamlining The Contracting Process 2017
AD 2017-34 Acquisition Reform Initiative #7: Improving Cost Estimation and Resourcing 2017

Provides policy and governance, for the DoD and the national security innovation base, that enables the delivery and sustainment of critical capabilities to the U.S. Service Members and allies

OUSD A&S

Asst. Secretary of Defense for Acquisition
Provide best practices on acquisition programs IOT achieve affordable and capable warfare systems. Oversight of joint programs, industrial base and supply chain expertise, cross-service data analytics and metrics, and stewardship of acquisition workforce.
- DASD/Director Acquisition Enablers
- DASD Platform & Weapon Portfolio Management
- DASD Information & Integration Portfolio Management
- DASD Information & Integration Portfolio Mgmt.
- Director Defense Pricing and Contracting
- President Defense Acquisition University
- Director Def Contract Mgmt. Agency

Asst. Secretary of Defense for Sustainment
Focus on joint and cross-Service material readiness issues, support for the Services' up-front program logistics planning, and identifying best practices to drive down costs of weapon systems sustainment.
- DASD/Director Enterprise Solutions
- DASD Logistics
- DASD Project Support
- DASD Material Readiness
- DASD Energy
- DASD Environment
- DASD Infrastructure
- DASD Facilities Management
- Director Office of Economic Assistance
- Director Defense Logistics Agency

Asst. Secretary of Defense for Nuclear. Chemical, and Biological Defense Programs
Oversee and prescribe policy for nuclear forces modernization; arms control programs; and counter weapons of mass destruction (Counter WMD) programs.
- DASD Threat Reduction & Arms Control
- DASD Nuclear Matters
- DASD B&C Defense Programs
- Director Defense Threat Reduction Agency

Dep Asst. Secretary of Defense for Industrial Policy
Maintain an Industrial Base and economic analysis capability to identify the impact and effects of budget procurement and sustainment decisions; and assess posture of the IB to accommodate defense needs.
- Principle Director Industrial Assmts
- Director Global Markets & Investments
- Director Small Business Programs

Director Joint Rapid Acquisition Cell
Rapidly assesses requirements and solutions, collaborating with the Joint Staff J-8 and the respective Combatant Commanders (COCOM) and facilitates the transfer of funds to DoD components to resolve Immediate Warfighter Needs (IWN).

Director Special Programs
Army Special Programs Directorate (ASPD) is the Army Special Access Program Central Office (SAPCO). Assist OSD with governance, management, and oversight of sensitive DoD capabilities, information, technologies, and operations.

Senior Advisor International Cooperation
The roles and responsibilities assigned to IC will be further assessed pending determination of potential benefit of integrating with Defense Security Cooperation Agency.

- Director International Armaments Cooperation
- Director Planning, Programming, and Analysis

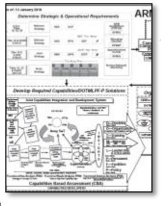
~~Director Strategy, Data & Design~~
Will oversee the reorganization of acquisition, technology, and logistics processes within the Office of the Secretary of Defense. "helping federal government organizations align technology with organizational objectives."

Defense Acquisition Board
Senior advisory board for the DoD Acquisition System and provides advice on critical acquisition decisions
Approves →

Major Defense Acquisition Program
A program that meets the ACAT I requirements and is classified as a MDAP by the Milestone Decision Authority

← *Sponsor*
Milestone Decision Authority
Initiates each increment of an evolutionary acquisition program as required by DoD Instruction 5000.02

Develop required capabilities and DOTMLPF-P solutions



The acquisition process encompasses the design, engineering, construction, testing, deployment, sustainment, and disposal of weapons or related items purchased from a contractor. As set forth by statute and regulation, from concept to deployment, a weapon system must go through a three-step process of identifying a required (needed) weapon system, establishing a budget, and acquiring the system. Each of these primary decision support systems/processes transform military forces to support the NDS.

1. The Joint Capabilities Integration and Development System (JCIDS) - for identifying requirements,
2. The Planning, Programming, Budgeting, and Execution System (PPBE) - for allocating resources and budgeting, and
3. The Defense Acquisition System (DAS) - for developing and/or buying the item.

The Defense Acquisition System, uses "milestones" to oversee and mange acquisition programs. At each milestone, a program must meet specific statutory and regulatory requirements before the program can proceed to the next phase of the acquisition process.
• Milestone A—initiates technology development,
• Milestone B—initiates engineering and manufacturing development, and
• Milestone C—initiates production and deployment.

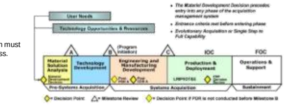
Army JCIDS produces an integrated set of DOTMLPF-P solution approaches that collectively provide the required capabilities (RC). As it is grounded in joint and Army concepts, the Army JCIDS provides traceability of all Army system and non-system solutions back to strategy. Three Outputs: Initial Capabilities Document (ICD): Defines the capability need Capability Development Document (CDD): Produced when the technology development phase is complete. The Capability Production Document (CPD): Engineering and Manufacturing Development Phase of the acquisition process

The DOD PPBE process serves as the primary resource management system for the department's military functions. Its purpose is to produce a plan, program, and defense budget in support of combatant commanders (CCDR).

The DOD Defense Acquisition System (DAS) establishes a management process to translate user needs and technological opportunities (developed or identified in the S&T program based on user needs) into reliable and sustainable systems that provide capability to the user. The basic policy is to ensure that acquisition of defense systems is conducted efficiently and effectively

The things that have to be done in defense acquisition never really change:
(1) Identify a need or desire for a new product.
(2) Reduce the technical risk to an acceptable level.
(3) Develop and test the product. (4) Field the product.
(5) Sustain and dispose of the product overtime.

JCIDS COBP



Requirements Pre Concept	Material Solution Analysis	R&E S&T Maturation & Risk Reduction	Engineering & Manufacturing Development	Production & Deployment	Sustainment & Disposal

Alternate Pathway(s)

Problem Identification & Definition — Wargames, Experimentation, TTXs, Hackathons, Tech Scouting, Challenge Competitions..
→ **Prototyping Limited Trials** — Prototyping, Testing & Evaluation, Refining, Developing CONOPS, Technology Transition..
→ **Decision Point**
- Scale Up
- Cancel
- Harvest Tech & Better Understand OP Needs
- Continue Limited Production

Acquisition
- Resourcing
- Contracting
- Costing
- Acquisition Logistics

DoD Budget Process
DoD Budget → OMB President's Budget → Congress → Budget Committees / Budget Resolutions → Armed Services Committees / Authorization Bills → Appropriations Committees / Appropriation Bills → OMB Apportionment Treasury Warrants → DoD

Categories of Funding
- Res, Dev, Test, & Evaluation (RDT&E)
- Procurement
- Other Procurement, Army (OPA) Funds
- Operation & Maintenance Army (O&MA)
- CDRs Emer Resp Program (CERP)
- Intelligence contingency funds (ICF)
- Logistics Civil Aug Program (LOGCAP)

CRS: Defense Primer USD R&E

Report to Congress Restructuring the DoD — AT&L Organization and CMO Organization

Bridging the Requirements Divide

Restructure DoD for future force modernization

Restructuring the USD for Acquisition, Technology and Logistics (AT&L) to maintain pace with the speed of the global technology market

2019 National Defense Authorization Act
The House bill contained a provision (sec. 219) that would direct the Secretary of Defense to develop a National Security Science and Technology Strategy to prioritize Department of Defense science and technology efforts and investments.

2018 National Security Strategy
To maintain our competitive advantage, the United States will prioritize emerging technologies critical to economic growth and security. We will nurture a healthy innovation economy that collaborates with allies and partners.

2017 Defense Posture Statement
The "Return to Great Power Competition" report. A commitment to the future to prepare the DoD for the extraordinary changes that are happening in our world and be ready for advances our adversaries are making.

2016 National Commission on the Future of the Army
The "End Strength" report. Army forces conduct their missions in a continuously changing strategic Environment. Recommends more funding is needed to meet the missions now and in the future.

2018 National Defense Strategy
The security environment is also affected by rapid technological advancements and the changing character of war. The drive to develop new technologies is relentless, expanding to more actors with lower barriers of entry, at accelerating speed.

2014 Reliance 21 Operating Plan
Technology Communities of Interests established in 2009 as a mechanism to encourage multi-agency coordination and collaboration in cross-cutting technology focus areas with broad multiple-Component investment.

2018 DIU China's Technology Transfer Strategy
The technologies where China is investing are the same ones where U.S. firms are investing and that will be foundational to future innovation. These are some of the same technologies of interest to build on the technological superiority of the U.S. military.

What are the National Technology Priorities?

The United States lacks a unified technology strategy synchronizing and educated work force, economic focus, and defense prioritization

Maintaining pace with the speed of the market

National Defense Authorization Act 2015-2019

Defense reform hearings concluded the US military was falling behind technologically and that the current acquisition structure and process were significant factors in the inability to access new sources of innovation. The NDAA disestablishes the Under Secretary of Defense for Acquisition, Technology, and Logistics (AT&L) and divides its existing duties among a new Under Secretary of Defense for Research and Engineering (USD(R&E)) and the renamed Under Secretary of Management and Support, a position mandated by the National Defense Authorization Act for Fiscal Year 2015 that took effect on February 1, 2017. (NDAA 17 Executive Summary).

National Security Strategy 2018
Priority Actions:
Understand Worldwide Science And Technology (S&T) Trends
Attract And Retain Inventors And Innovators
Leverage Private Capital and Expertise To Build and Innovate
Rapidly Field Inventions and Innovations

Defense Posture Statement 2017
The Land Investments:
Next-Generation Shoulder-launched Weapon
Replacement the Army Tactical Missile System (ATACMS)
Increased Firepower for Stryker AFVs
Develop Joint Light Tactical Vehicle to replace Humvees
Strengthen and improve the missile defense capabilities

2017 S&T Portfolio Realignment
The Army must immediately review the FY 2018 and 2019 investments to ensure the investments align with the new priorities - realigning what can be changed in the investment portfolio.. To better support the six modernization priorities.

National Commission on the Future of the Army 2016
It kind of began here for the Army. Military technology trends:

Precision Guidance	Cyber / Electro-Magnetic
Supercomputing and Big Data	Bio Technology
Robotics/Autonomy	Space Access
Nano Technology	

So Here are the Priorities for today

USD (R&E) Top 10 Technology Focus Areas

Hypersonics	Cybersecurity	Microelectronics
Directed Energy	Ai / Machine Learning	Autonomy
Commd, Control, Communications	Missile Defense	
Space Offense and Defense	Quantum Science & Computing	

USD (R&E) Dr. Michael Griffin derived list from the 2018 National Defense Strategy

Reliance 21 Operating Principles Communities of Interest

Advanced Electronics	Energy & Power Technology	✓ Biomedical
C4I	✓ Materials & Man Processes	✓ Cyber
✓ Electronic Warfare / Protection	✓ Autonomy	Ground & Sea Platforms
✓ Human Systems	✓ Counter WMD	Space
Air Platforms	✓ Engineered Resilient Systems	Weapons Technology
Counter IED	Sensors & Processing	✓ *Data to Decision*

✓ DoD S&T Priorities

China's Investment in Critical Technology

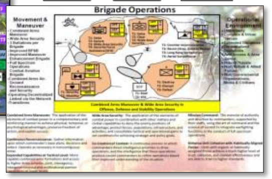
Artificial Intelligence Ai	Augmented / Virtual Reality
Robotics	Financial Technology (Fintech)

Globally Responsive, Forced Entry, EDREs, Analog, High Intensity Operations, Rapid Expeditionary Capability, Three Dimensional Hybrid Threat, Temp Windows of Dominance, Machine Gun Teams, Mass Combat Power, Fighting for Intelligence, Disintegrate

We still exist in a Brigade centric Land Force
Let's not forget that!

Primary Organizing Principle: Conduct Fire and Movement → Following Principles of: Find, Fix, Finish, Follow Thru → Executed thru Tactical Tasks: SBF, Breach, Assault, Follow Through

Future Force Modernization →



US Army Modernization Strategy
Addresses the challenges of the future operational environment and directly supports the 2018 National Defense Strategy's (NDS) line of effort, "Build a More Lethal Force."

Army Modernization Priorities

Long-Range Precision Fires	Army Network
Next Generation Combat Vehicles	Future Vertical Lift
Air and Missile Defense Capabilities	Soldier Lethality

Future Force Modernization Enterprise - Innovates

DEFENSEWERX
Cultivate a vibrant ecosystem by shaping successful collisions to innovate and commercialize technologies that benefit the warfighter.

AFWERX
Catalyst for agile Air Force engagement across industry, academia and non-traditional contributors to create transformative opportunities. Austin; DC; Las Vegas

SOFWERX
A very dynamic environment designed to create a high rate of return on collision for all participants. Through the use of our growing ecosystem, promotion of divergent thought, and neutral facilitation, our goal is to bring the right minds together to solve challenging problems.

MGMWERX
Augment Air University programs to enhance production of high-quality, innovative research and ideas that span issues of importance to the Air Force.
Montgomery, AL

National Security Innovation Network
Create new communities of innovators that solve national security problems.
Hacking 4 Defense; Techstars Air Force Accelerator Hanscom AFB & 20th Fighter Wing Boot Camp

Hacking 4 Defense
Students develop a deep understanding of the problems and needs of government sponsors in the Department of Defense and the Intelligence Community.
Stanford University

IN Q TEL
Technology vetted against the requirements of our partners' mission capability needs, then compares alternative approaches and rigorously validates technical claims.
Partners include the CIA, NSA, and DoD

Tech Link
Primary activity is marketing new inventions from the DoD and Department of VA and navigating businesses and entrepreneurs through the licensing process.
Montana State U. Office of Research & Economic Dev.

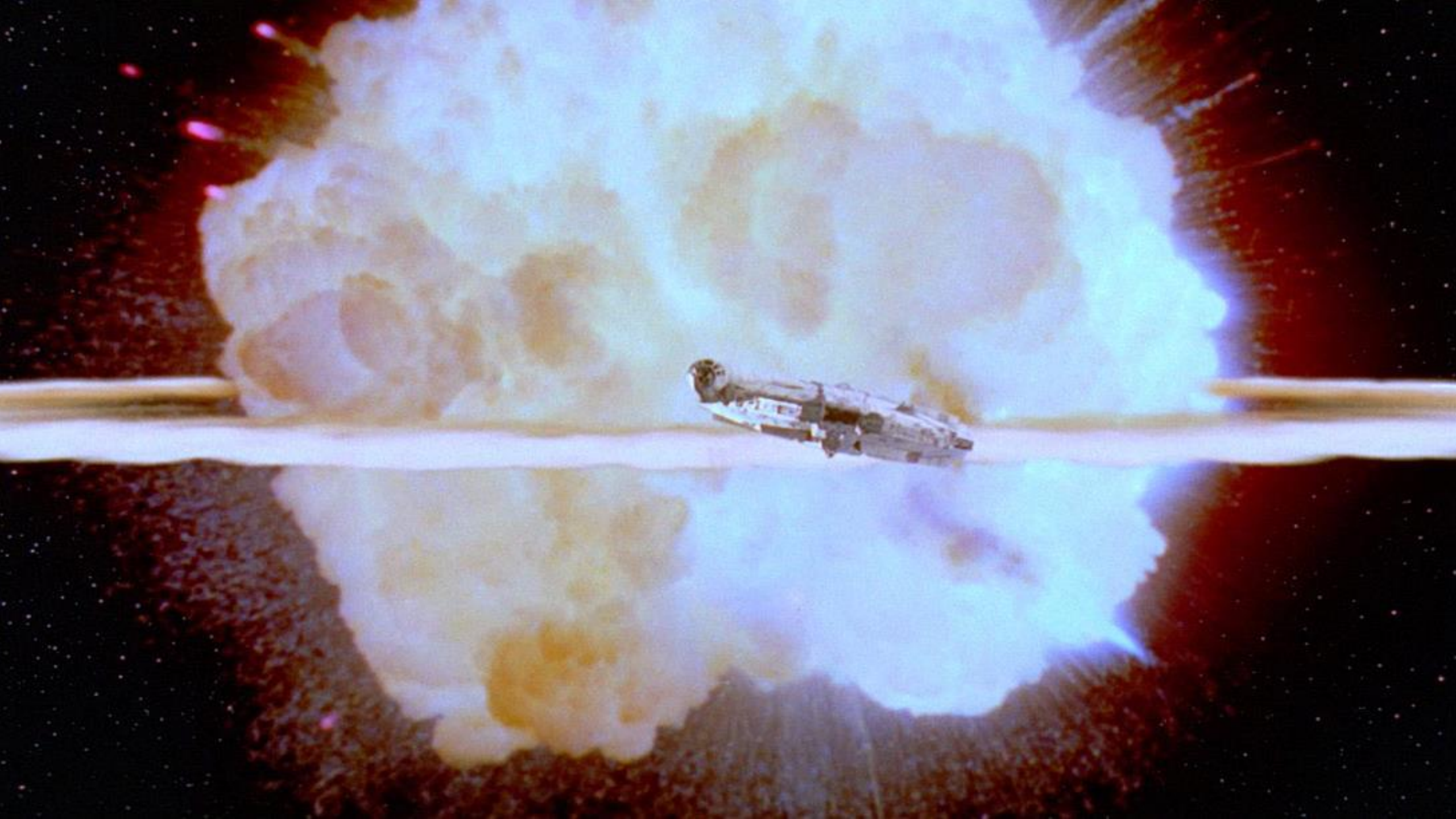
X Tech Search
The ASA(ALT) recognizes that the Army must enhance engagements with the entrepreneurial funded community small businesses, and other non-traditional defense partners.

Defense Innovation Unit (DIU)
Accelerates adoption of leading commercial technology into the military.
HACQER xView

National Defense Battlelab LLC. clarence@defensebattlelab.com

> 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.

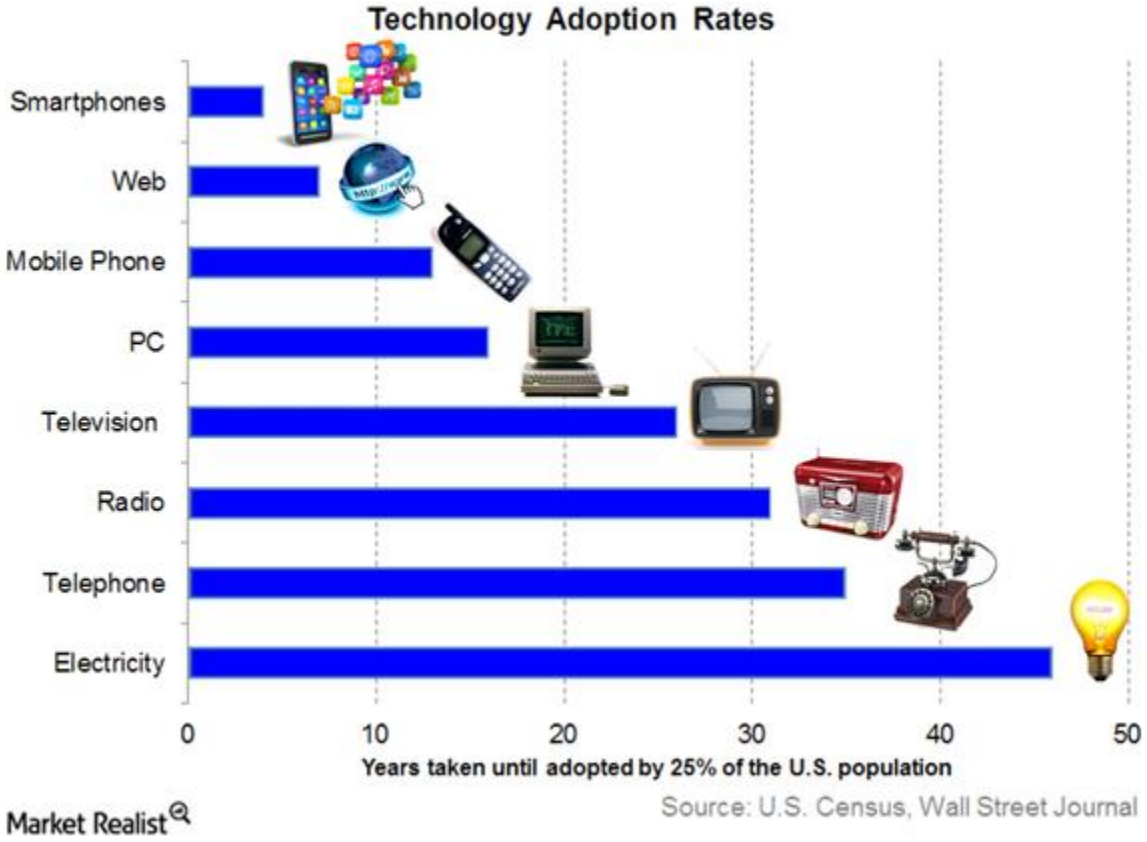
-GHCQ: Boiling Frogs? Technology organizations need to change to radically survive increasing technical and business disruptive



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)

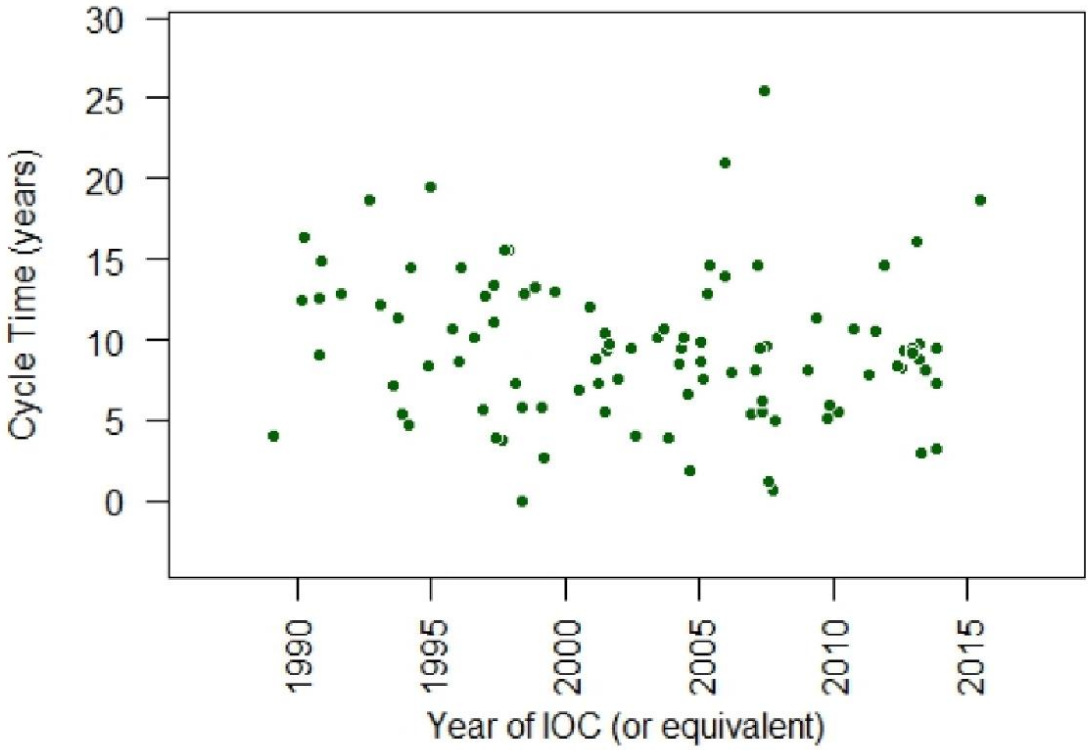


Figure 1. Program/Subprogram Cycle Time by IOC Year

Tate, David M. Acquisition Cycle Time: Defining the Problem (Revised). Institute for Defense Analysis, Oct 2016.

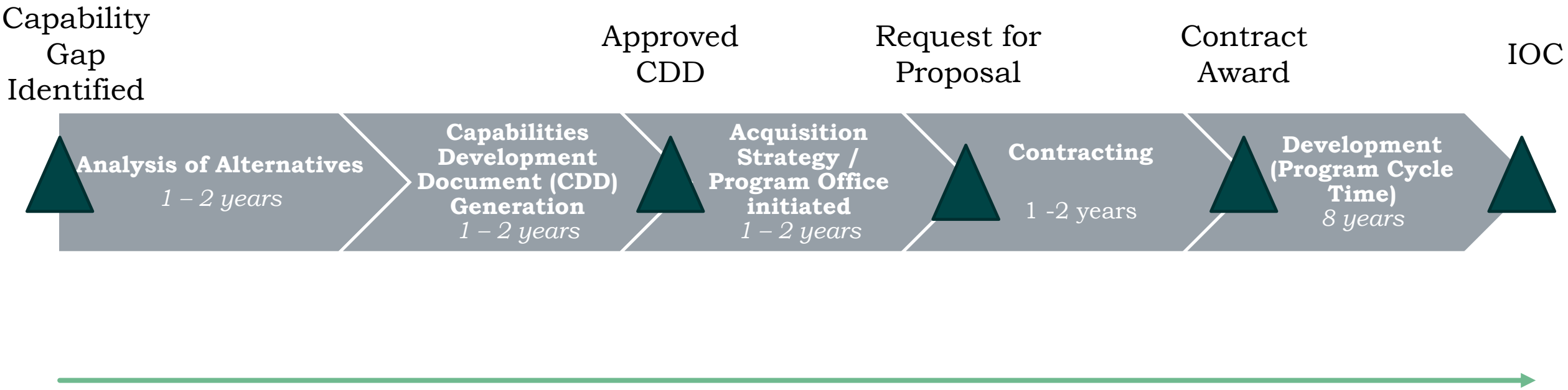
Shows cycle times for Major Defense Acquisition Program (MDAP) from initial development to Initial Operational Capability (or equivalent)

**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.

there is
nothing else.

@gapingvoid

gapingvoid
Culture Design Group

A PROPOSED PROCESS

> Without defined processes, you can't scale, you can't put metrics and instrumentation in place…you can't manage.

Rossman, John. *The Amazon Way: 14 Leadership Principles behind the World's Most Disruptive Company*. CreateSpace, 2014.

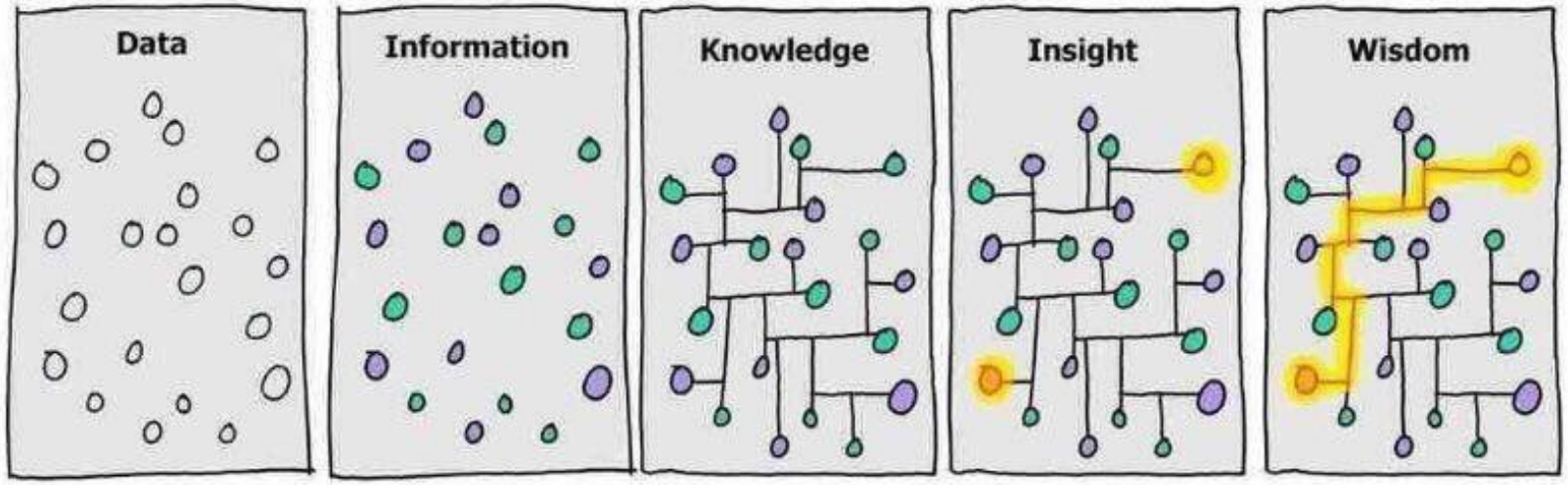
PROCESS

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

PROCESS

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



PROCESS

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

PROCESS

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

JSRC

I NEED A BUDGET ESTIMATE FOR MY PROJECT, BUT I DON'T HAVE A SCOPE OR A DESIGN FOR IT YET.
Dilbert.com DilbertCartoonist@gmail.com
OKAY, MY ESTIMATE IS $3,583,729.
12-7-09 ©2009 Scott Adams, Inc./Dist. by UFS, Inc.
YOU DON'T KNOW ANYTHING ABOUT MY PROJECT.
THAT MAKES TWO OF US.

PROCESS

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: 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)

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

2019 Software and Information Technology Cost Analysis Solutions Team

August 21, 2019

John Sautter

Cost Analyst

Approved for Public Release #19-1455. Distribution Unlimited

© 2019 Northrop Grumman Corporation

Table of Contents

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

© 2019 Northrop Grumman Corporation

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
- From Defense Innovation Board – Software Acquisition and Practices (SWAP Concept Paper Titled: Metrics for Software Development, Dated 03 May 2019.
  - 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.

© 2019 Northrop Grumman Corporation

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Alternate Sizing Choices

© 2019 Northrop Grumman Corporation

Alternate Sizing Choices - Agenda

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

© 2019 Northrop Grumman Corporation

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

Presented at the 2019 ICEAA Professional Development & Training Workshop - www.iceaaonline.com

Total Hours vs SW Changes -1

Model	Conditions	Obs	Adj R^2	SEE (Hrs)	PRED(30)
$THrs = 463 * (TSC)^{0.69}$	All data	329	0.36	48,385	17.3%
$THrs = 341 * (TSC)^{0.79}$	10% trimmed data	263	0.57	44,842	23.6%

Army Software Sustainment Cost Estimating Results

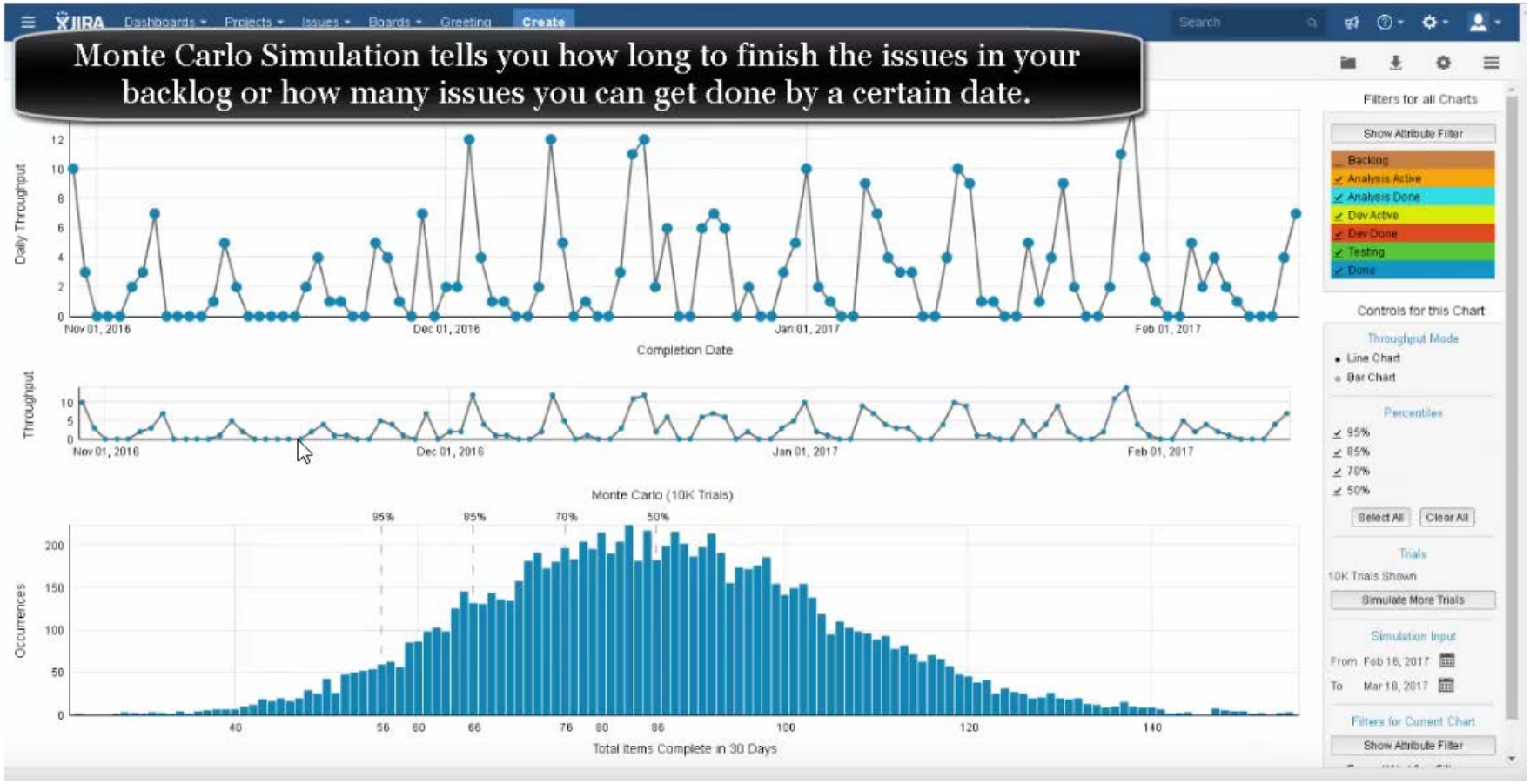
Cheryl Jones
James Doswell
Brad Clark
Robert Charette
Paul Janusz

- 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

© 2019 Northrop Grumman Corporation

Estimating with our Management Tools

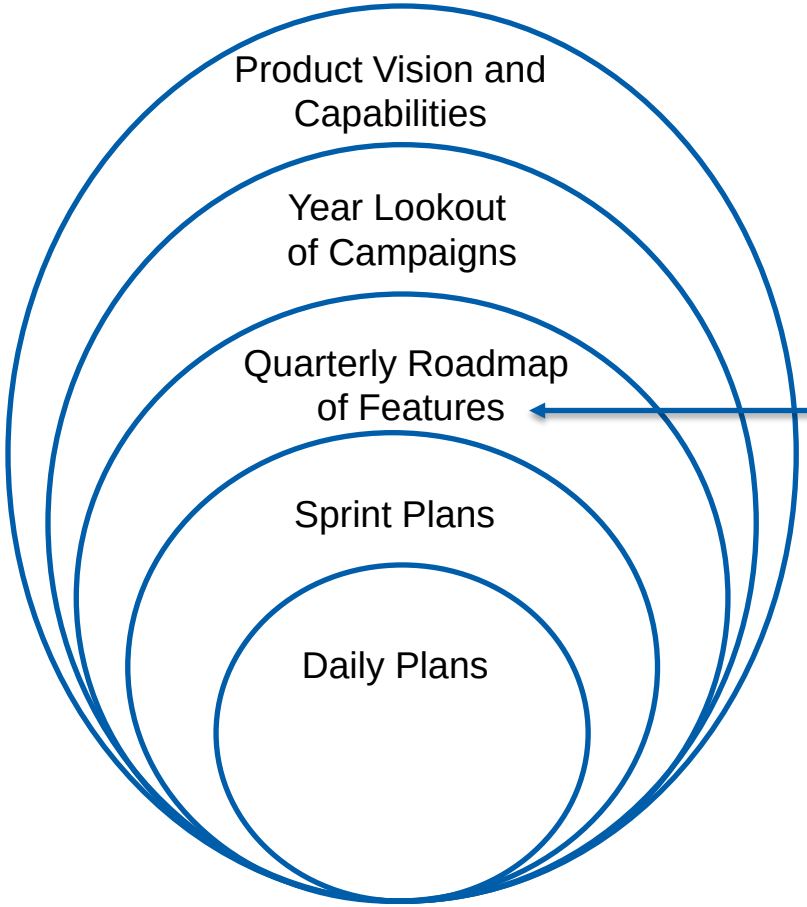


Reference: ActionableAgile for JIRA – AgileMetrics
https://marketplace.atlassian.com/apps/1216661/actionableagile-for-jira-agile-metrics?hosting=server&tab=overview

For Projects In-Flight – Story/Issue tracking can help in estimation and historicals

© 2019 Northrop Grumman Corporation

Planning with Spheres of Uncertainty



Aligning to Candidate Software Metrics

1. Usefulness metrics
   - 1.1. List of features implemented
   - 1.2. Prioritized list of features planned for future release
   - 1.3. Number of active users
   - 1.4. Level of satisfaction of the user base
   - 1.5. Ranked list of feature usage (most used to least used)
   - 1.6. Number and severity of defects reported during operations
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 cybersecurity audit/penetration tests

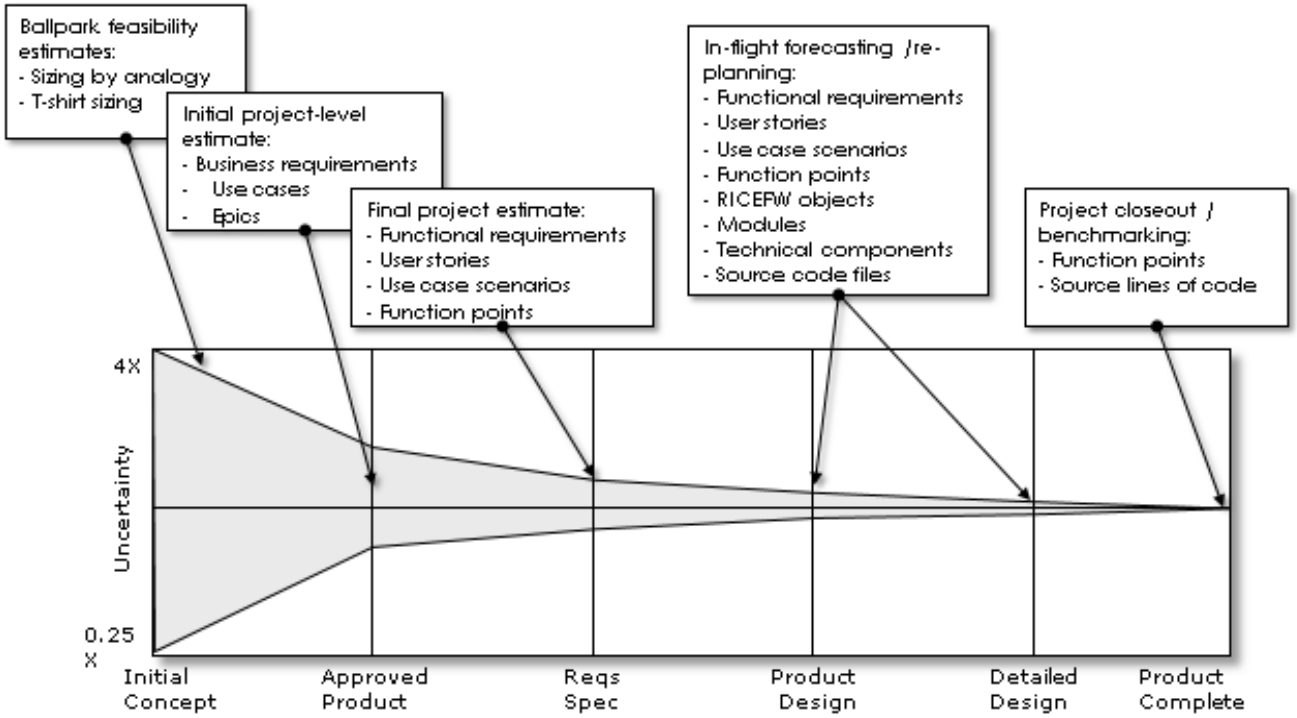
Cost Community Needs Size Drivers Early in Life Cycle

© 2019 Northrop Grumman Corporation

Sizing Methods Used at Various Stages of the Software Development Life Cycle

NORTHROP GRUMMAN

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."



QSM The Intelligence behind Successful Software Projects

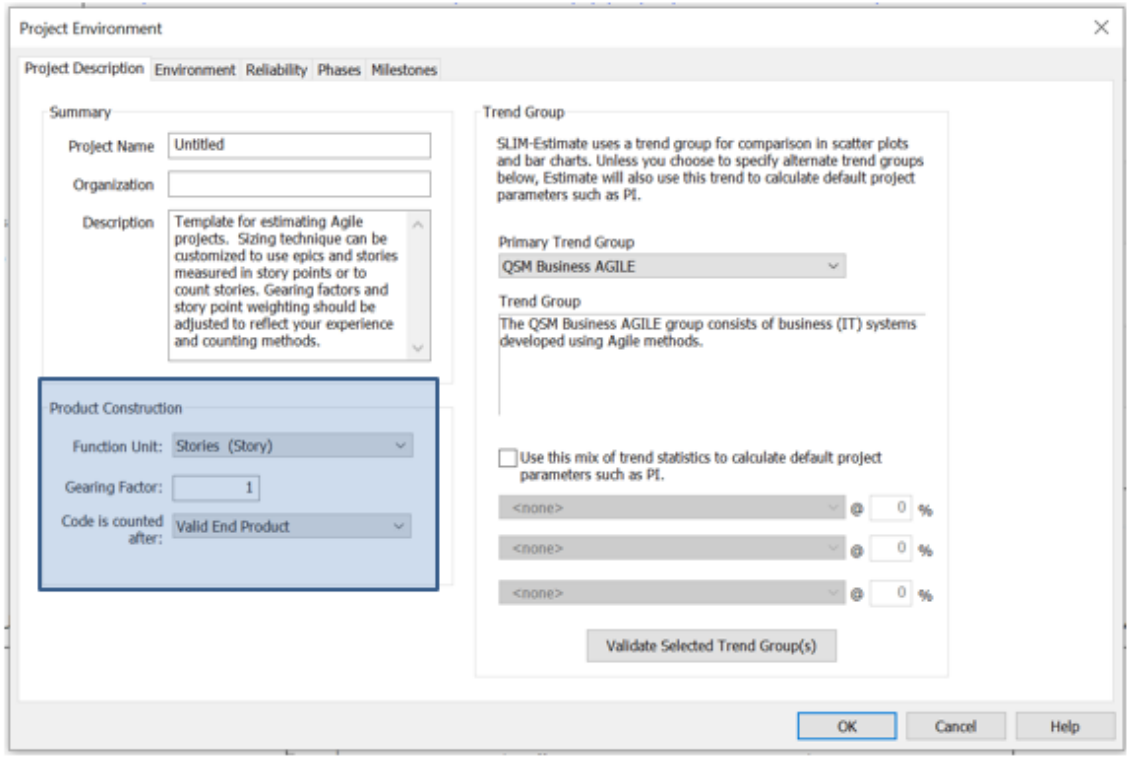
Quantitative Software Management (QSM) identifies some sizing Through the Cone of Uncertainty

See: https://www.qsm.com/infographic/software-sizing-matters

© 2019 Northrop Grumman Corporation

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



QSM The Intelligence behind Successful Software Projects

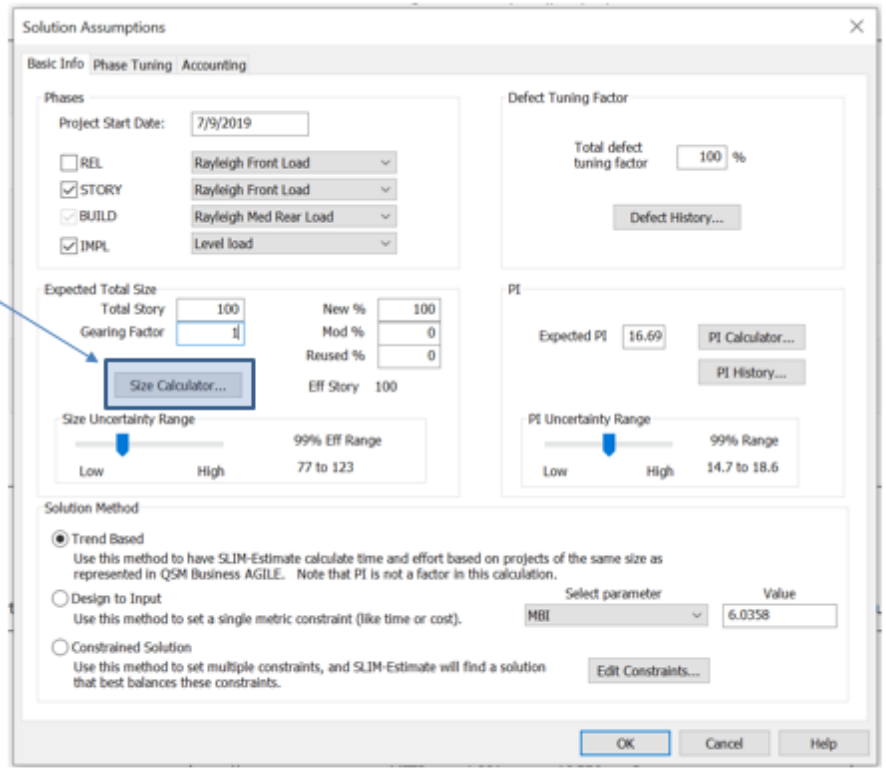
QSM SLIM – Parametric Suite

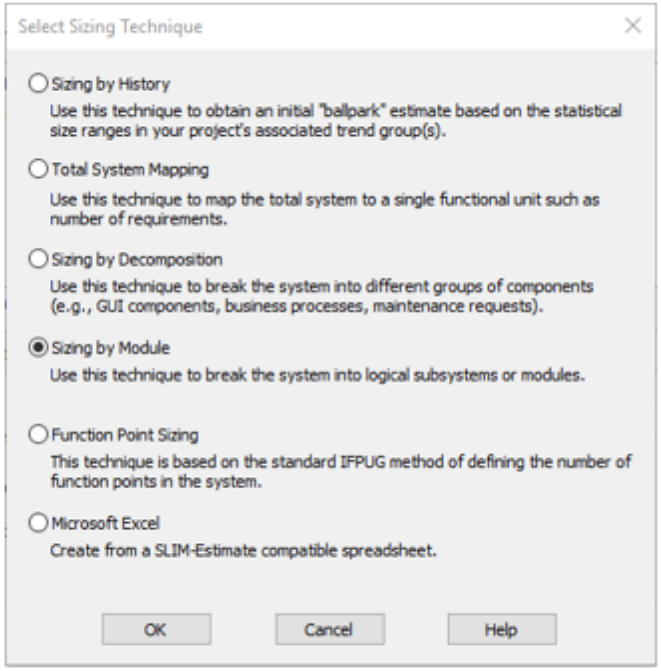
© 2019 Northrop Grumman Corporation

SLIM-Estimate Size Calculator Subsystem

NORTHROP GRUMMAN

The Size Calculator provides multiple sizing approaches that can be combined to refine size estimates and quantify sizing uncertainty



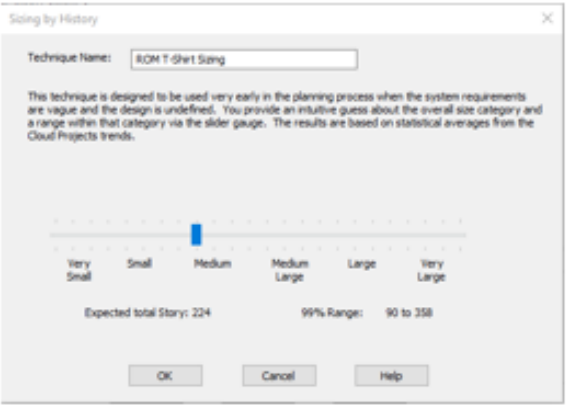


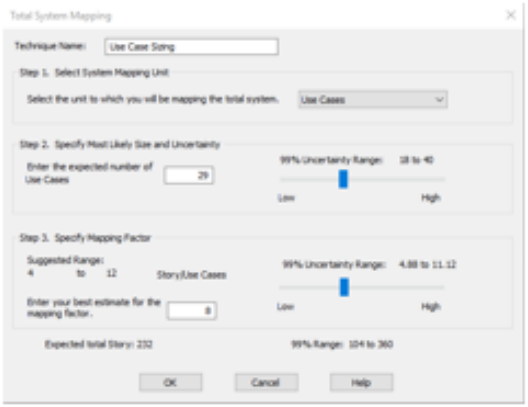
QSM The Intelligence behind Successful Software Projects

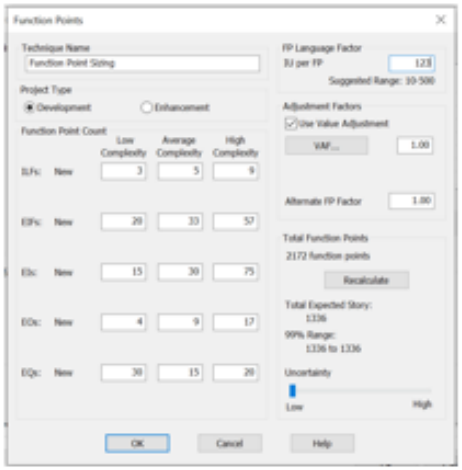
QSM SLIM – Parametric Suite

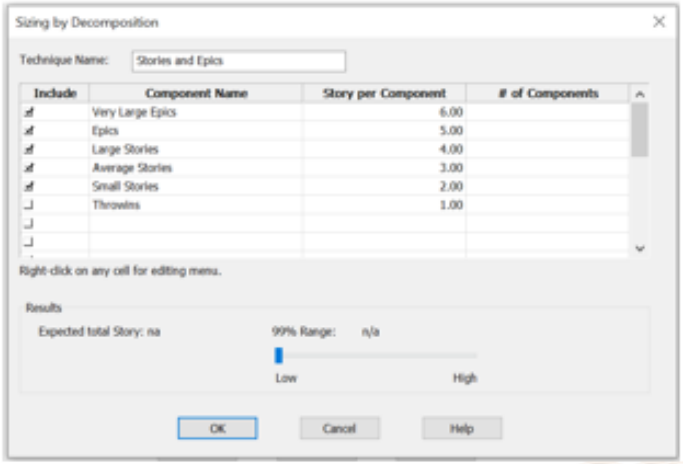
© 2019 Northrop Grumman Corporation

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 The Intelligence behind Successful Software Projects

QSM SLIM – Parametric Suite

© 2019 Northrop Grumman Corporation

TruePlanning Parametric Inputs

True Planning SW Size Inputs

Software Size		
Size Units	Source Lines of Code (SLOC)	
New Size	15,000	
New Size Non-executable	0.00%	%
Adapted Size	0	
Adapted Size Non-executable	0.00%	%
Percent of Design Adapted	0.00%	%
Percent of Code Adapted	0.00%	%
Percent of Test Adapted	0.00%	%
Design Repeat	0.00%	%
Reused Size	0	
Reused Size Non-executable	0.00%	%
Reuse Factor	1.00	
Deleted Size	0	
Removal Complexity	Remove functionality that is integrated through a single point	
Auto Generated Size	0	
Auto Gen Size Non-executable	0.00%	%
Auto Translated Size	0	
Auto Trans Size Non-executable	0.00%	%
Auto Translation Tool Efficiency	Nominal - 80%	

COSMIC Function Points
Functional Size
IFPUG Function Points
Source Lines of Code (SLOC)
Use Case Conversion Points (UCCP)

Calculators:

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

Types of "code":

- New
- Adapted/Modified
- Reused
- Deleted
- Auto Generated
- Auto Translated

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

© 2019 Northrop Grumman Corporation

TruePlanning Parametric Inputs

COSMIC Function Points

Tables and Calculators

New Size

This value describes the amount of entirely new functionality that does not reuse any existing artifacts. greater than zero for Reused Size or Adapted Size. Enter the value in single units of the selected sof

	Name	Entry	Exit	Read	Write
1	Feature 1	5	7	10	10
2	Featire 2	3	12	8	8
3	Feature 3	4	7	12	10
4	Feature 4	15	15	15	15
5	Feature 5	6	8	10	8
6	Feature 6	10	12	15	10

Use Case Conversion Points

Tables and Calculator

New Size

This value describes the amount of entirely new functionality that does not reuse any existing artifacts. When the value is be greater than zero for Reused Size or Adapted Size. Enter the value in single units of the selected software Size Units t

	Use Case Name	Number of Transactions	Simple Actors	Average Actors	Complex Actors	Data Volume Multiplier
1	Use Case 1	20	2	4	4	1.00
2	Use Case 2	20	2	4	4	1.00
3	Use Case 3	20	2	4	4	1.00
4	Use Case 4	20	2	4	4	1.00
5	Use Case 5	20	2	4	4	1.00

Fast Function Points

Tables and Calculators

New Size

This value describes the amount of entirely new functionality that does not reuse any existing artifacts. When the value is zero (0.00), the value must be greater than zero for Reused Size or Adapted Size. Enter the value in single units of the selected software Size Units to be created.

	Name	Create	Update	Delete	Read	Report	Data(Save)	Data(Read
1	Feature 1	5	5	16	14	14	0	
2	Feature 2	5	5	16	14	14	0	
3	Feature 3	5	5	16	14	14	0	
4	Feature 4	5	5	16	14	14	0	
5	Feature 5	5	5	16	14	14	0	

RICEFW Objects => Function Points

(Reports, Interfaces, Conversions, Enhancements, Forms, Workflows)

Tables and Calculators

New Size

This value describes the amount of entirely new functionality that does not reuse any existing artifacts. When the value is zero (0.00), the va Size or Adapted Size. Enter the value in single units of the selected software Size Units to be created.

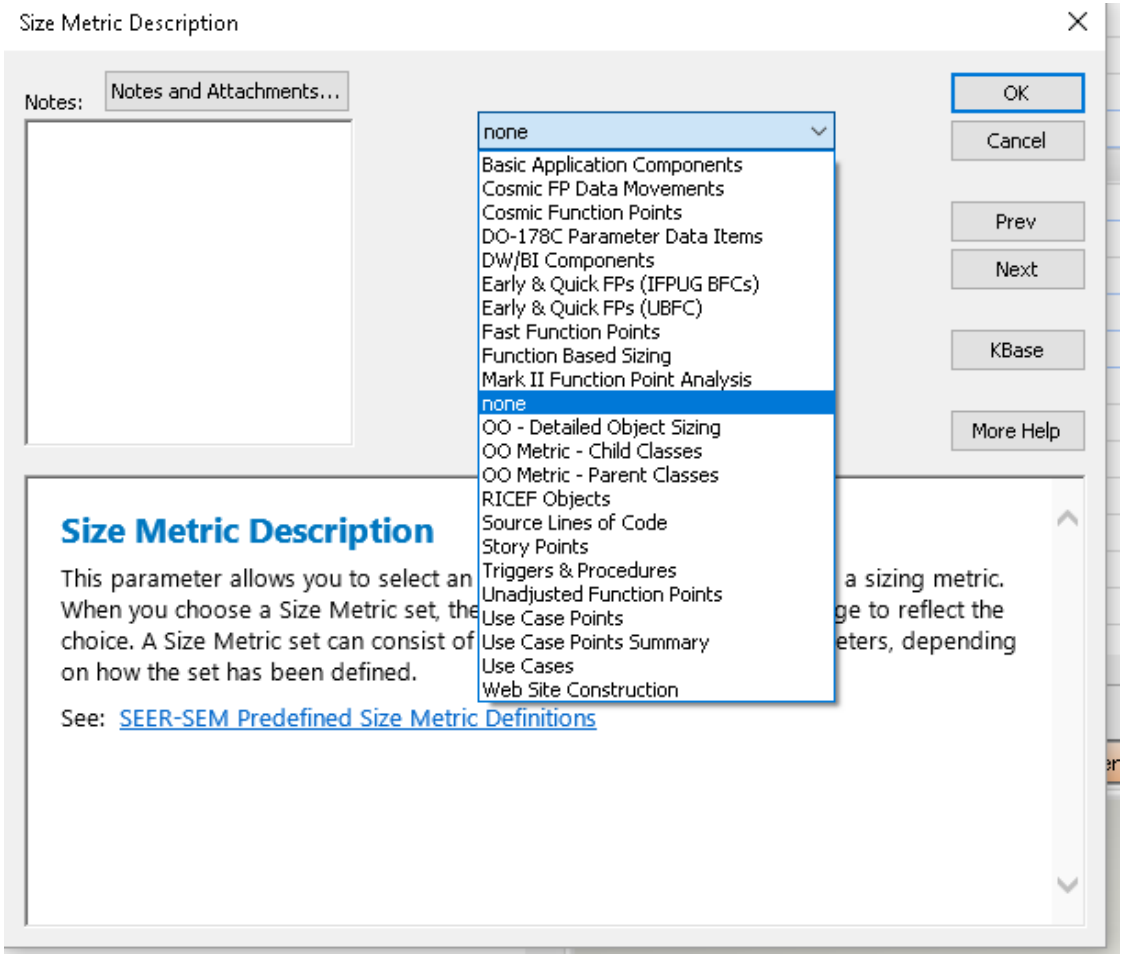
Section Name	Input Field	Units
Reports		
Reports - Low Complexity	5	
Reports - Average Complexity	5	
Reports - High Complexity	5	
Interfaces		
Interfaces - Low Complexity	4	
Interfaces - Average Complexity	6	
Interfaces - High Complexity	7	
Conversions		
Conversions - Low Complexity	9	
Conversions - Average Complexity	5	
Conversions - High Complexity	4	
Enhancements		
Enhancements - Low Complexity	5	
Enhancements - Average Complexity	10	
Enhancements - High Complexity	12	
Forms		
Forms - Low Complexity	4	
Forms - Average Complexity	4	
Forms - High Complexity	6	
Workflow		
Workflow - Low Complexity	3	
Workflow - Average Complexity	3	
Workflow - High Complexity	3	

Also provides the full complement of the IFPUG function points

International Function Point Users Group (IFPUG)

© 2019 Northrop Grumman Corporation

SEER For Software - Sizing



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.

© 2019 Northrop Grumman Corporation

SEER For Software - Sizing

EXTERNAL INPUTS (EI)	0	0	0
Input Screens	-	-	-
Interactive Inputs	-	-	-
Hardware Inputs	-	-	-
Batch Input Streams	-	-	-
EXTERNAL OUTPUTS (EO)	0	0	0
Screen Reports	-	-	-
Printed Reports	-	-	-
Media Outputs	-	-	-
Software Outputs	-	-	-
Hardware Outputs	-	-	-
EXTERNAL INQUIRIES (EQ)	0	0	0
Request/Response	-	-	-
Menus	-	-	-
Context Sensitive Help	-	-	-
Embedded Computer Inquiries	-	-	-
EXTERNAL INTERFACE FILES (EIF)	0	0	0
Reference Data	-	-	-
Fixed Messages	-	-	-
Shared Data Files	-	-	-
INTERNAL LOGICAL FILES (ILF)	0	0	0
Application Data Groups	-	-	-
Data Tables	-	-	-
Data Base Files	-	-	-
Internal Functions	0	0	0

Selected Definition

Description: Basic Application Components

File name: BasicAppComps.PRX

Input Name	Factor	FP or SLOC
Complex Reports	7	FP
Simple Screens	3	FP
Complex Screens	4	FP
Simple Forms	4	FP
Complex Forms	6	FP
Menus	3	FP
Simple Information Query	3	FP
Complex Information Query	4	FP
Simple Transactions	4	FP
Complex Transactions	6	FP
Simple Interface Files	5	FP
Complex Interface Files	10	FP
Simple Data Structures	7	FP

Description: Use Case Points

File name: Use Case Points.PRX

Input Name	Factor	FP or SLOC
Simple Actor (A1)	1.14774	FP
Average Actor (A2)	2.29548	FP
Complex Actor (A3)	3.44322	FP
Simple Use Case (U1)	5.73870	FP
Average Use Case (U2)	11.4774	FP
Complex Use Case (U3)	17.2161	FP

SEER for Software will translate into effective size. Historical ESLOC productivity can be used to validate a function based estimate.

© 2019 Northrop Grumman Corporation

Using Functional Requirements Counts

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

Project Type	Least (Hours)	Mean (Hours)	Most (Hours)
Enterprise Resource Planning (ERP)	125	192	325
Automated Information System (AIS)	50	141	375
Defense Software	75	154	300

  - 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)

© 2019 Northrop Grumman Corporation

Early and Quick Function Points

NORTHROP GRUMMAN

- Function based sizing early in life cycle
  - Consider using Early and Quick Function Points
  - <u>Develop an Excel lookup table for tagging functional requirements</u>
    - Include Simple Function Points

Reference: Early and Quick Function Point Manual Google "E&QFP"

Level 1 – Standard IFPUG Set

Level 1		Min	ML	Most
Basic Functional Component - Transactions	EIL	3	3	3
	EIA	4	4	4
	EIH	6	6	6
	EQL	3	3	3
	EQA	4	4	4
	EQH	6	6	6
	EOL	4	4	4
	EOA	5	5	5
	EOH	7	7	7
Basic Functional Component - Data	ILFL	7	7	7
	ILFM	10	10	10
	ILFH	15	15	15
	EIFL	5	5	5
	EIFM	7	7	7
	EIFH	10	10	10
Level 2		**Min**	**ML**	**Most**
Unclassified Elementary Process	GEI	4	4.2	4.4
	GEQ	3.7	3.9	4.1
	GEO	4.9	5.2	5.4
	UGO	4.1	4.6	5
	UGP	4	4.4	4.8
Unclassified Logical File	GILF	7.4	7.7	8.1
	FEIF	5.2	5.4	5.7
	UGDG	6.4	7.1	7.8

Use higher levels if the requirements are high level capabilities

Level 3		Min	ML	Most
Typical Process	TPS	14.1	16.5	19
	TPM	17.9	21.1	24.3
	TPL	22.3	26.3	30.2
General Process	GPS	26.4	35.2	44
	GPM	42.9	57.2	71.5
	GPL	59.4	79.2	98.9
General Data Group	GDGS	15	21.4	27.8
	GDGM	32.4	46.3	60.2
	GDGL	54.8	78.3	101.8
Level 4		**Min**	**ML**	**Most**
Macro Process	MPS	111.5	171.5	231.5
	MPM	185.8	285.9	385.9
	MPL	297.3	457.4	617.4
	None			
Simple File	SFILE	6	7	8
Simple EP	SEP	[illegible]		

Add Simple Functions to your lookup table

© 2019 Northrop Grumman Corporation

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

© 2019 Northrop Grumman Corporation

Simple Function Points (SiFP)

IFPUG Differences

- 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

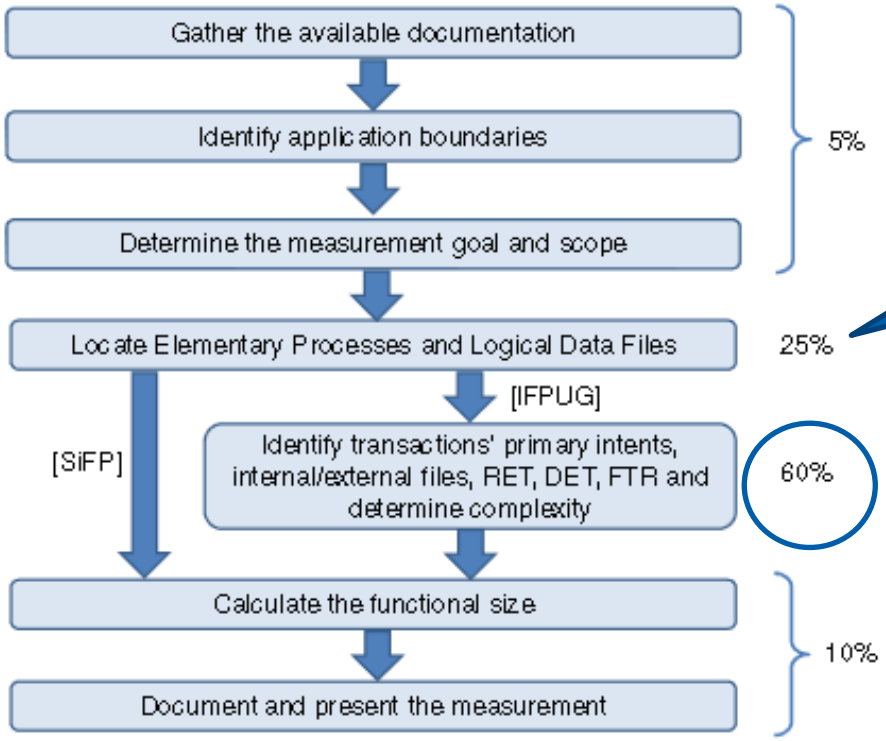
- In addition, analysis must consider distinguishing differences of ILF vs. EIF and the difference between an EO and an EQ.

© 2019 Northrop Grumman Corporation

Simple Function Points (SiFP)

Time Saved

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



- Also, no more EO vs. EQ arguments
- (ILFs vs. EIFs, no longer an issue)

Authors claim about 60% saving avoiding complexity determination

© 2019 Northrop Grumman Corporation

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

© 2019 Northrop Grumman Corporation

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/

© 2019 Northrop Grumman Corporation

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)

© 2019 Northrop Grumman Corporation

SNAP Framework Breakdown

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

If SNAP is used then must track hours per SNAP point separately from FP work

© 2019 Northrop Grumman Corporation

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Other Aspects

© 2019 Northrop Grumman Corporation

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

© 2019 Northrop Grumman Corporation

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Concluding remarks

© 2019 Northrop Grumman Corporation

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

© 2019 Northrop Grumman Corporation

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

© 2019 Northrop Grumman Corporation

Headquarters U.S. Air Force

Integrity - Service - Excellence

Tracking Software Development:

An example of feature-based estimating

Air Force Cost Analysis Agency

U.S. AIR FORCE

U.S. AIR FORCE

Overview

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.

Estimate Hierarchy

U.S. AIR FORCE

Users		Decisions Supported
AQ, OSD Equivalent	Strategic Long Term	Early Planning Estimates Competing Requirements & Priorities Broad Alternative Analysis (e.g. AoAs) Establish Program Baseline
CAPE, Services PEO, MAJCOMs	Operational Mid Term	Milestone Decisions POM/PB Budget Allocation
Developers, PMO	Tactical Near Term	Contract Award, Execution & Performance To Complete Estimating

Many levels of communication
Information and data needs vary across the levels

Estimate Hierarchy

U.S. AIR FORCE

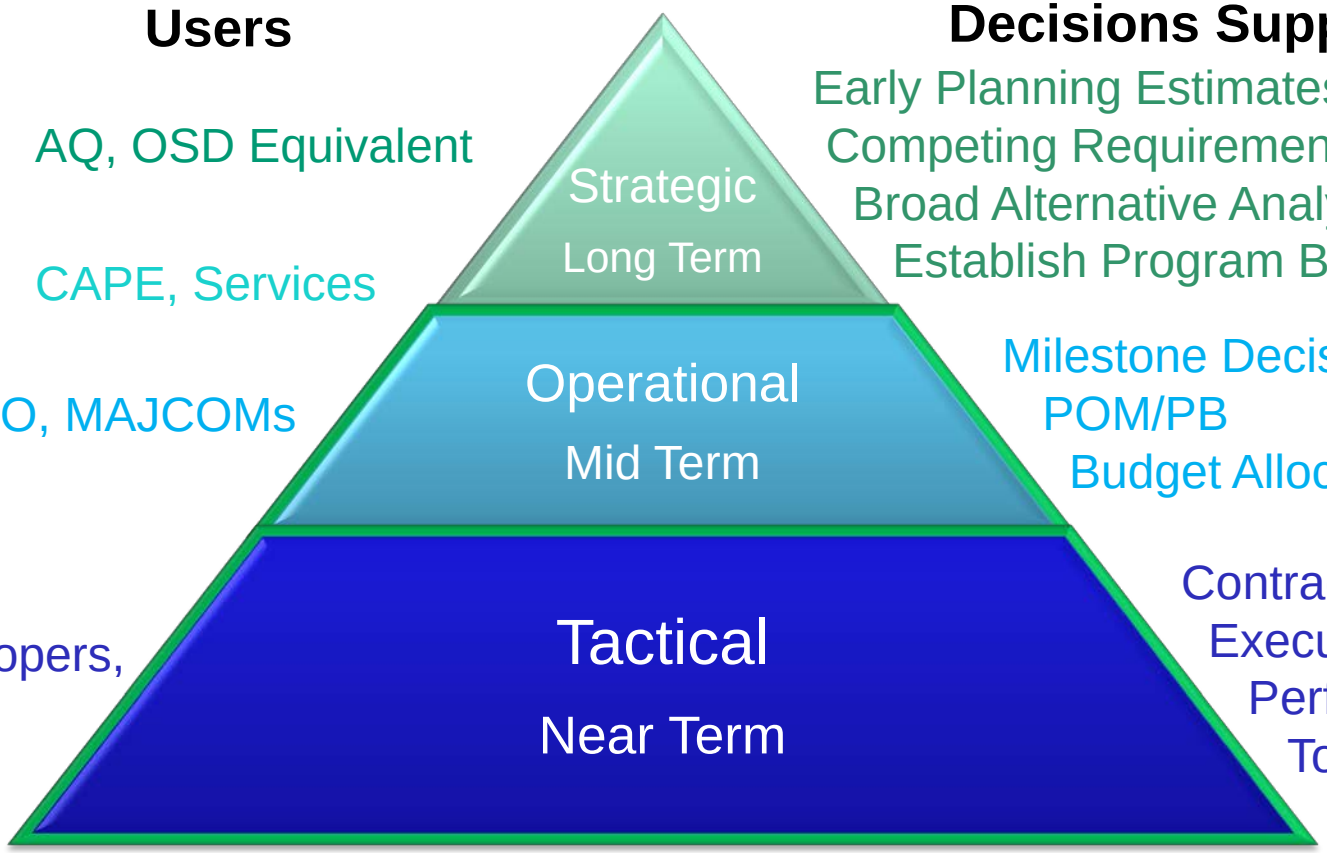
Users

AQ, OSD Equivalent

CAPE, Services

PEO, MAJCOMs

Developers, PMO



Decisions Supported

Early Planning Estimates
Competing Requirements & Priorities
Broad Alternative Analysis (e.g. AoAs)
Establish Program Baseline

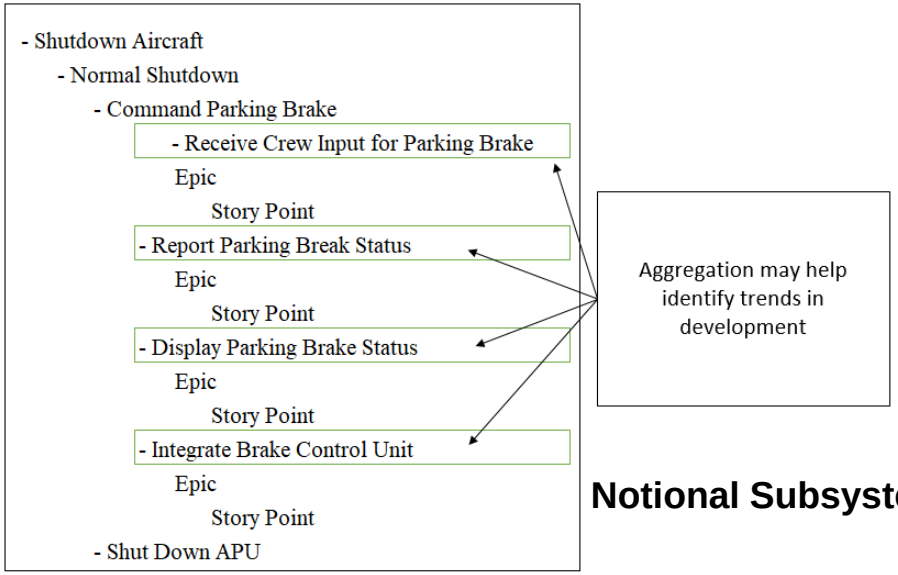
Milestone Decisions
POM/PB
Budget Allocation

Contract Award, Execution & Performance
To Complete Estimating

Many levels of communication
Information and data needs vary across the levels

Terminology

- **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**

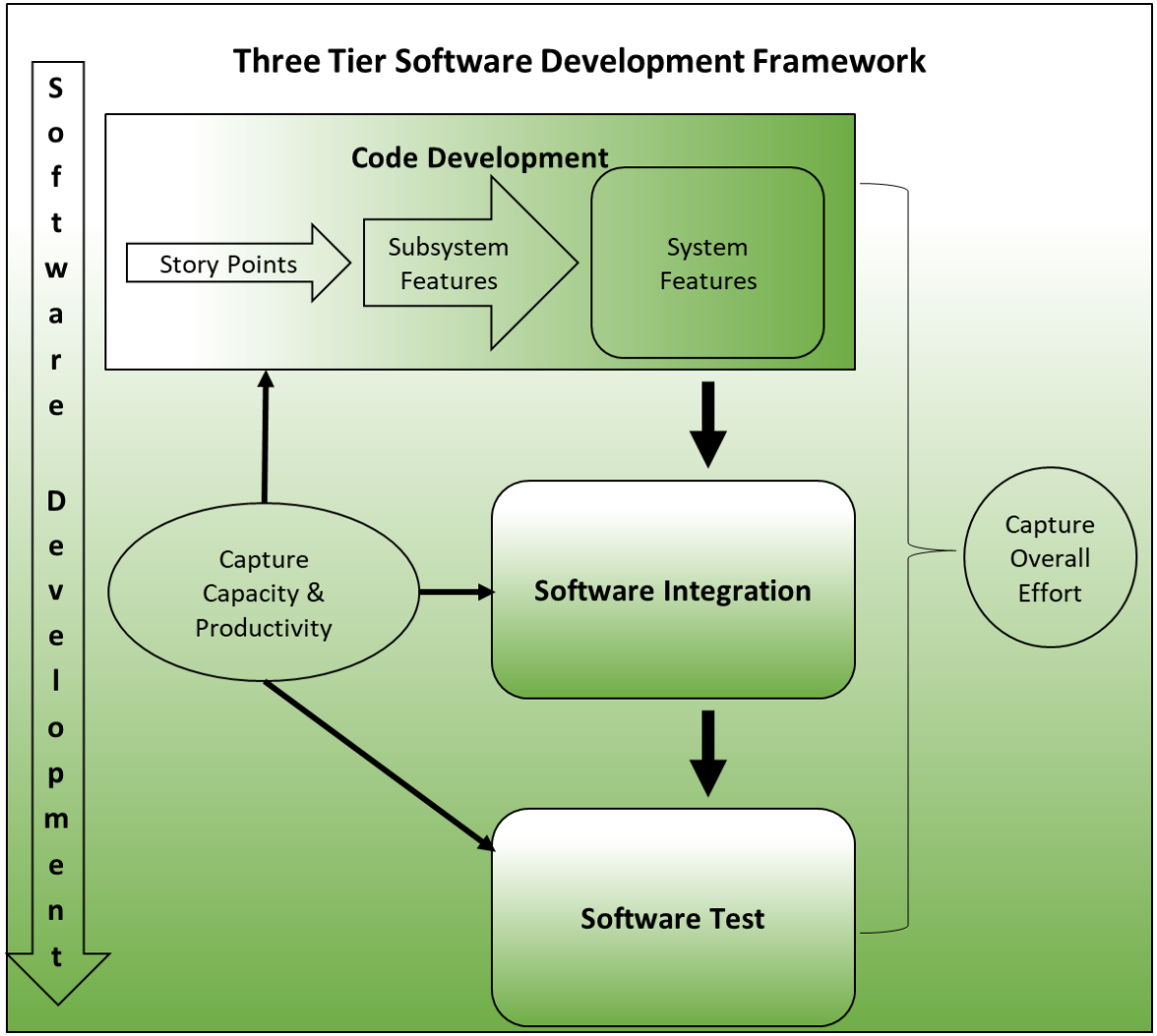


Notional Subsystem Feature Example

For presentation purposes: We will call them subsystem features (SSF)

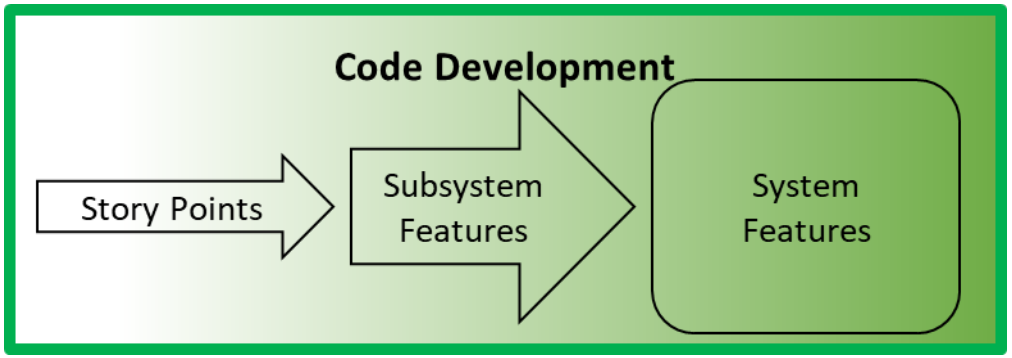
Approach

- **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**



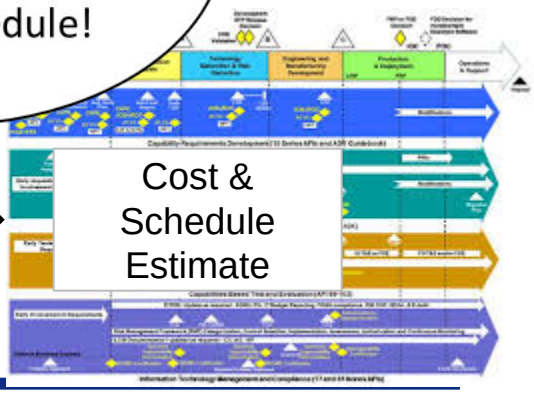
Approach

- **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

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	SSF 1
1.2.5.2.2	Communication/Idenfitfication Software Design	SSF 4
1.2.5.3	Navigation/Guidance	
1.2.5.4	Mission Computer/Processing	
1.2.5.4.1	Mission Management SW Design and Development	SSF 5 SSF 6 SSF 7
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	SSF 8 SSF 9 SSF 10
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	SSF 11 SSF 12 SSF 13
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 14 SSF 15

Features mapped to lowest possible WBS element

NOTIONAL EXAMPLE ONLY

U.S. AIR FORCE

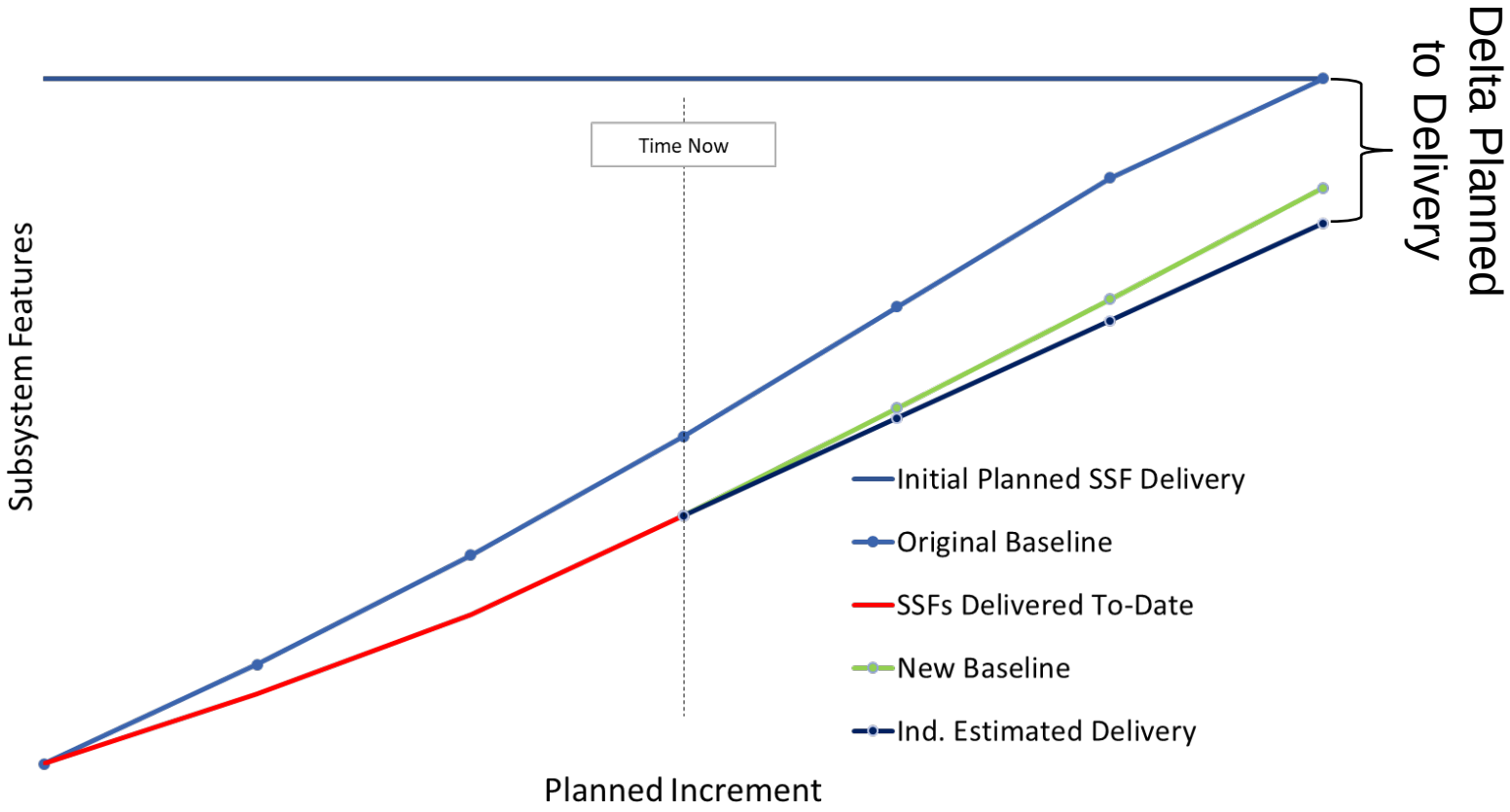
Tracking Hours

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	SSF 1
1.2.5.2.2	Communication/Idenfitfication Software Design	SSF 4
1.2.5.3	Navigation/Guidance	
1.2.5.4	Mission Computer/Processing	
1.2.5.4.1	Mission Management SW Design and Development	SSF 5 SSF 6 SSF 7
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	SSF 8 SSF 9 SSF 10
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	SSF 11 SSF 12 SSF 13
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 14 SSF 15

Hours tracked to the lowest possible WBS element

NOTIONAL EXAMPLE ONLY

Illustrated Example



U.S. AIR FORCE

Example

Prior to Execution:

Previous Increment's Statistics:	
Hrs/SSF:	14.0
Staffing:	40
Developer Planned Performance:	
Hrs/SSF:	13.4
Staffing:	50
Planned Delivery:	3450
Independent Estimate:	
Hrs/SSF:	14.0
Staffing:	45
Estimated Delivery:	2841

What Actually Happened:

Midway Through Increment

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
Hrs/SSF:	16.1	15.2	13.7			
Delivered SSF:	425	450	500			

Replan:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:	16.1	15.2	13.7	12.7	12.4	12.2	**13.7**
Delivered SSF:	425	450	500	540	550	560	**3025**

Estimated:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:				14.0	14.0	14.0	**14.0**
Delivered SSF:	425	450	500	489	489	489	**2841**

Replan comes to fruition. 3025 SSFs delivered.

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

U.S. AIR FORCE

Example

Prior to Execution:

Previous Increment's Statistics:

Hrs/SSF:	14.0
Staffing:	40

Developer Planned Performance:

Hrs/SSF:	13.4
Staffing:	50
Planned Delivery:	3450

Independent Estimate:

Hrs/SSF:	14.0
Staffing:	45
Estimated Delivery:	2841

What Actually Happened:

Midway Through Increment

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
Hrs/SSF:	16.1	15.2	13.7			
Delivered SSF:	425	450	500			

Replan:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:	16.1	15.2	13.7	12.7	12.4	12.2	**13.7**
Delivered SSF:	425	450	500	540	550	560	**3025**

Estimated:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:				14.0	14.0	14.0	**14.0**
Delivered SSF:	425	450	500	489	489	489	**2841**

Replan comes to fruition. 3025 SSFs delivered.

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

U.S. AIR FORCE

Example

Prior to Execution:

Previous Increment's Statistics:

Hrs/SSF:	14.0
Staffing:	40

Developer Planned Performance:

Hrs/SSF:	13.4
Staffing:	50
Planned Delivery:	3450

Independent Estimate:

Hrs/SSF:	14.0
Staffing:	45
Estimated Delivery:	2841

What Actually Happened:

Midway Through Increment

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
Hrs/SSF:	16.1	15.2	13.7			
Delivered SSF:	425	450	500			

Replan:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:	16.1	15.2	13.7	12.7	12.4	12.2	**13.7**
Delivered SSF:	425	450	500	540	550	560	**3025**

Estimated:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:				14.0	14.0	14.0	**14.0**
Delivered SSF:	425	450	500	489	489	489	**2841**

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

Prior to Execution:

Previous Increment's Statistics:	
Hrs/SSF:	14.0
Staffing:	40
Developer Planned Performance:	
Hrs/SSF:	13.4
Staffing:	50
Planned Delivery:	3450
Independent Estimate:	
Hrs/SSF:	14.0
Staffing:	45
Estimated Delivery:	2841

What Actually Happened:

Midway Through Increment

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
Hrs/SSF:	16.1	15.2	13.7			
Delivered SSF:	425	450	500			

Replan:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:	16.1	15.2	13.7	12.7	12.4	12.2	**13.7**
Delivered SSF:	425	450	500	540	550	560	**3025**

Estimated:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:				14.0	14.0	14.0	**14.0**
Delivered SSF:	425	450	500	489	489	489	**2841**

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

Prior to Execution:

Previous Increment's Statistics:

Hrs/SSF:	14.0
Staffing:	40

Developer Planned Performance:

Hrs/SSF:	13.4
Staffing:	50
Planned Delivery:	3450

Independent Estimate:

Hrs/SSF:	14.0
Staffing:	45
Estimated Delivery:	2841

What Actually Happened:

Midway Through Increment

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
Hrs/SSF:	16.1	15.2	13.7			
Delivered SSF:	425	450	500			

Replan:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:	16.1	15.2	13.7	12.7	12.4	12.2	**13.7**
Delivered SSF:	425	450	500	540	550	560	**3025**

Estimated:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:				14.0	14.0	14.0	**14.0**
Delivered SSF:	425	450	500	489	489	489	**2841**

Replan comes to fruition. 3025 SSFs delivered.

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

U.S. AIR FORCE

Example

Prior to Execution:

Previous Increment's Statistics:	
Hrs/SSF:	14.0
Staffing:	40

Developer Planned Performance:	
Hrs/SSF:	13.4
Staffing:	50
Planned Delivery:	3450

Independent Estimate:	
Hrs/SSF:	14.0
Staffing:	45
Estimated Delivery:	2841

What Actually Happened:

Midway Through Increment

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6
Hrs/SSF:	16.1	15.2	13.7			
Delivered SSF:	425	450	500			

Replan:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:	16.1	15.2	13.7	12.7	12.4	12.2	**13.7**
Delivered SSF:	425	450	500	540	550	560	**3025**

Estimated:

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	**Total**
Hrs/SSF:				14.0	14.0	14.0	**14.0**
Delivered SSF:	425	450	500	489	489	489	**2841**

Replan comes to fruition. 3025 SSFs delivered.

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

U.S. AIR FORCE

Challenges

- **Accumulating enough data for:**
  - **Normalization**
  - **Trend development**
  - **Early development planning**
- **Working across contractors:**
  - **Handling different software development methodologies**
  - **Availability & Fidelity of data**

U.S. AIR FORCE

QUESTIONS?

National Defense Industrial Association

Implementing Continuous Iterative Development in DoD Acquisition

Summary Progress and Status

IT CAST

August 2019

Summary Progress and Status

Key Initiatives	
Collect industry feedback on agile/CID info needs and measures (DSB, DIB, …) (usefulness, effectiveness)	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)
Provide industry recommendations to DoD for implementing DSB SW Task Report (and DIB SWAP) (NDIA/INCOSE/PSM Continuous Iterative Development & Sustainment WG)	Delivered 2 briefings 4/15/19 to DoD A&S, R&E (posted on NDIA SE Div website): • Detailed recommendations • Executive summary Briefed OUSD A&S and R&E 5/21/19.
DoD DIB SWAP study finalized and published 5/3/19. OUSD A&S brief to Congress Jun ’19.	https://innovation.defense.gov/software/ Likely mandated by GFY20 NDAA.
Develop consensus CID measurement framework (PSM)	Formed PSM/INCOSE/NDIA SME WG. Draft ICM table & indicator specs late Jul for review. Seeking additional reps for core team and reviewers. POC: Cheryl Jones, Geoff Draper, Larri Rosser
Provide industry feedback to DoD on draft SW acquisition policy (“SW 5000.02”)	Industry review in progress – seeking INCOSE input. Comments due to DoD early August. POC: G.Draper, C.Jones, G.Roedler, R.Yeman
Industry collaboration and outreach to further consensus on agile/CID development and measures.	• PSM CID WG. PSM User’s Group (Sep ’19). • NDIA SE Conference (Oct ’19)

https://www.ndia.org/divisions/systems-engineering/studies-and-publications

Studies and Publications

2019 Reports

- Implementing Continuous Iterative Development and Acquisition
  Defense Science Board (DSB) released a report in Feb-2018 containing seven recommendations regarding software design and acquisition. Section 868 of NDAA 2019 mandates implementation of these recommendations within 18 months. The Defense Innovation Board (DIB) Software Acquisition and Practices (SWAP) study group has also provided many insightful and largely compatible recommendations.

  NDIA, INCOSE and PSM support the DSB and DIB concepts and the opportunities they offer to DoD and the defense industry. In 2018 the NDIA Systems Engineering Division commissioned a working group to study the implementation of the DIB and SWAP recommendations. NDIA offers the working group's recommendations below to OUSD(A&S) and OUSD(R&E) representing a broad “industry perspective” on a path forward.

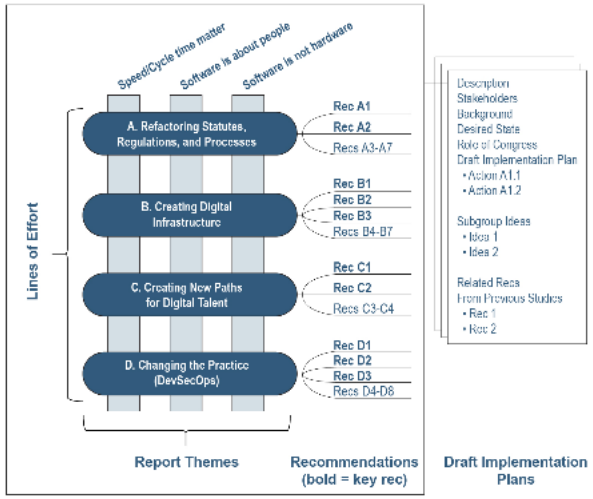
Executive Summary Brief (pdf)

Executive Summary Brief (ppt with links) view in presentation mode

Full Study Brief

Transmittal Letter to Dr. Boleng, OUSD (A&S)

DIB SWAP Study https://innovation.defense.gov/software/

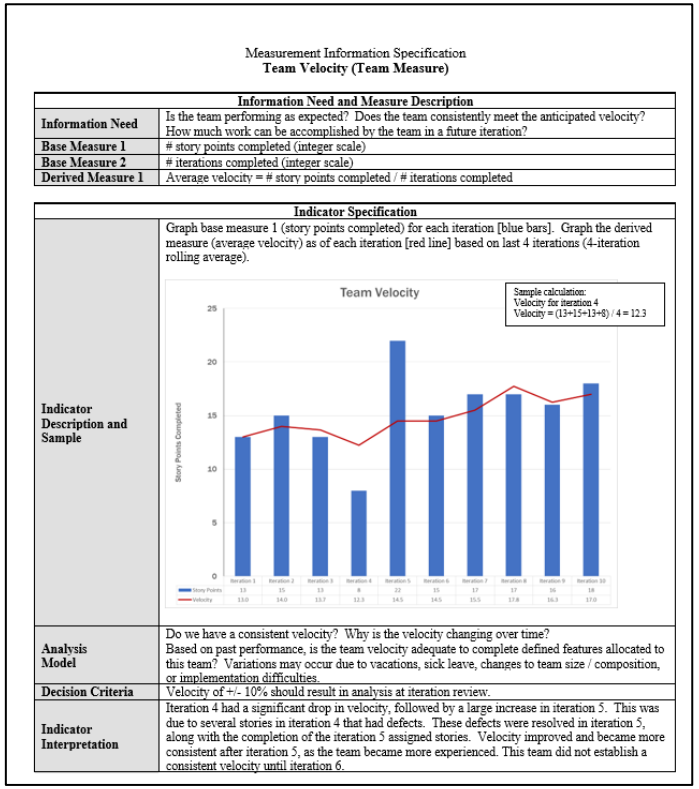


Developing a consensus measurement framework for agile development

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)**

Velocity	Acceleration	Burndown (Sprint, Release)	Committed vs. Completed
Defect Escapes	Defect Containment	Release Frequency	Automated Test Coverage



Measurement Information Specification
Team Velocity (Team Measure)

Information Need and Measure Description	
Information Need	Is the team performing as expected? Does the team consistently meet the anticipated velocity? How much work can be accomplished by the team in a future iteration?
Base Measure 1	# story points completed (integer scale)
Base Measure 2	# iterations completed (integer scale)
Derived Measure 1	Average velocity = # story points completed / # iterations completed

Indicator Specification	
Indicator Description and Sample	Graph base measure 1 (story points completed) for each iteration [blue bars]. Graph the derived measure (average velocity) as of each iteration [red line] based on last 4 iterations (4-iteration rolling average).
Analysis Model	Do we have a consistent velocity? Why is the velocity changing over time? Based on past performance, is the team velocity adequate to complete defined features allocated to this team? Variations may occur due to vacations, sick leave, changes to team size / composition, or implementation difficulties.
Decision Criteria	Velocity of +/- 10% should result in analysis at iteration review.
Indicator Interpretation	Iteration 4 had a significant drop in velocity, followed by a large increase in iteration 5. This was due to several stories in iteration 4 that had defects. These defects were resolved in iteration 5, along with the completion of the iteration 5 assigned stories. Velocity improved and became more consistent after iteration 5, as the team became more experienced. This team did not establish a consistent velocity until iteration 6.

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)

Information Categories	Measurable Concept	Team Information Need	Product Information Need	Enterprise Information Need	Potential Measures
Schedule and Progress	Work Unit Progress (team, product) Milestone Completion (enterprise)	Are story points delivered as committed?	Are features delivered as committed? Are we still on track to deliver all features per roadmap? (on plan)	Are capabilities delivered as committed?	(story points, features, capabilities) Burndown Committed vs. Completed Cumulative Flow Diagram (WIP)
	Work Unit Progress		Did we deliver expected capabilities / features? Is the roadmap still valid?	Is the user satisfied with the delivered products? Do they provide the desired functionality when needed?	Feature or Capability Implementation by priority
	Work Unit Progress		Is the integration and test progress proceeding as planned?		Test Progress (# test run and passed)
	Work Backlog		How much outstanding technical or mission debt exists?		Feature or Capability Backlog
Size and Stability	Functional Size and Stability Physical Size and Stability	How big is our system?	How big is our system?	How big is our system?	Stories produced (team) Features Capabilites Requirements SLOC
	Functional Size and Stability		How volatile are capabilities or features? Are we adding more features? What is the ability to accommodate changes in customer desirements?	How volatile are capabilities or requirements? What is the ability to accommodate changes in customer desirements?	Features Delivered Feature Volatility Capabilites Delivered Capability Volatility Backlog Volatility
	Functional Size and Stability	How much of the product is newly developed vs. reused from other sources?			Reuse of capability, features, stories, code

Backup –
Selected Excerpts from Briefings and Analyses

Interesting in getting involved more in consensus community solutions for any of these initiatives?

Let us know – help is welcome!

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)

Summary

The NDIA WG provides an industry perspective on picture of success, current state, obstacles and path forward for each DSB recommendation

DSB Recommendation	NDIA "Path Forward" recommendations	
#1 – Software Factory	14	Contracting, funding, incentives, methods, security, supply chain, and measures
#2 – Continuous Iterative Development	3	Pilots and continuous improvement
#3 – Risk Reduction & Metrics	10	Acquisition strategy, competitive prototyping, culture, workforce, IP, and measures
#4 – Legacy Systems	5	Assessments, supply chain, methods, tools, and modeling
#5 – Workforce Development	3	Competency models, workforce assessment, workforce management, and training
#6 – Sustainment	2	Contracting and industry-government transfer of sustainment responsibilities
#7 – Machine Learning	5	Risk, research, CONOPs, ML data, and Software Factory interactions

Details of each topic and recommendation are provided in the separate report.

DSB SW Task Force Recommendations

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**

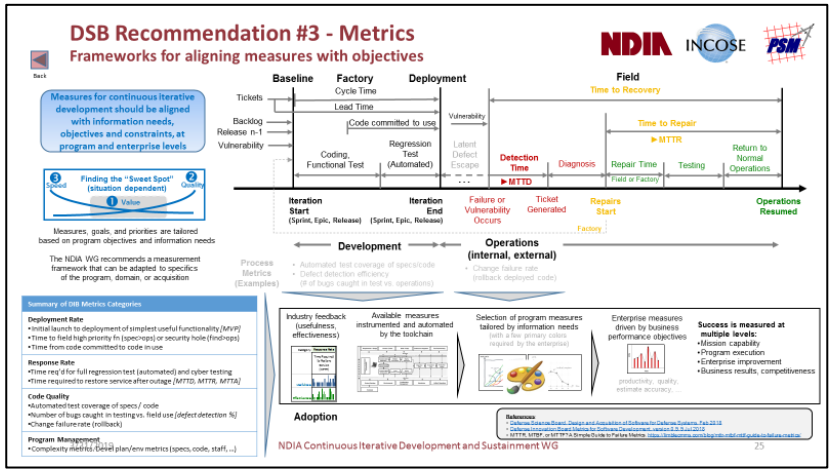
This briefing is an executive summary of those recommendations. Detailed report provided separately.

DSB #3b: Measures for CID

NDIA WG Recommendations

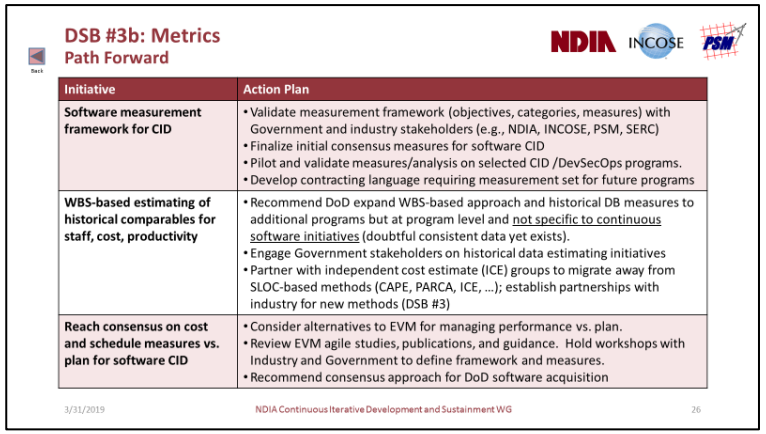
Picture of Success (end state)	
Consensus frameworks	• Objectives first - measures aligned and tailored from information needs, goals and constraints, at program and enterprise levels
Modernized measures	• Migration toward consensus alternatives to traditional waterfall and phase-based SW measures (LOC, 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

We are here



Measures for CID should be aligned with information needs and constraints, at program and enterprise levels

Recommendations for Path Forward:



DSB #3b: Metrics
Path Forward

Initiative	Action Plan
Software measurement framework for CID	• Validate measurement framework (objectives, categories, measures) with Government and industry stakeholders (e.g., NDIA, INCOSE, PSM, SERC) • Finalize initial consensus measures for software CID • Pilot and validate measures/analysis on selected CID /DevSecOps programs. • Develop contracting language requiring measurement set for future programs
WBS-based estimating of historical comparables for staff, cost, productivity	• Recommend DoD expand WBS-based approach and historical DB measures to additional programs but at program level and not specific to continuous software initiatives (doubtful consistent data yet exists). • Engage Government stakeholders on historical data estimating initiatives • Partner with independent cost estimate (ICE) groups to migrate away from SLOC-based methods (CAPE, PARCA, ICE, …); establish partnerships with industry for new methods (DSB #3)
Reach consensus on cost and schedule measures vs. plan for software CID	• Consider alternatives to EVM for managing performance vs. plan. • Review EVM agile studies, publications, and guidance. Hold workshops with Industry and Government to define framework and measures. • Recommend consensus approach for DoD software acquisition

3/31/2019 NDIA Continuous Iterative Development and Sustainment WG 26

Click thumbnails to zoom

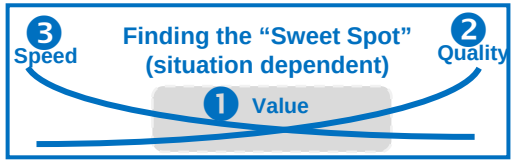
CID: Continuous Iterative Development
EVMS: Earned Value Management System
LOC: Lines of Code
WBS: Work Breakdown Structure

DSB Recommendation #3 - Metrics

Frameworks for aligning measures with objectives

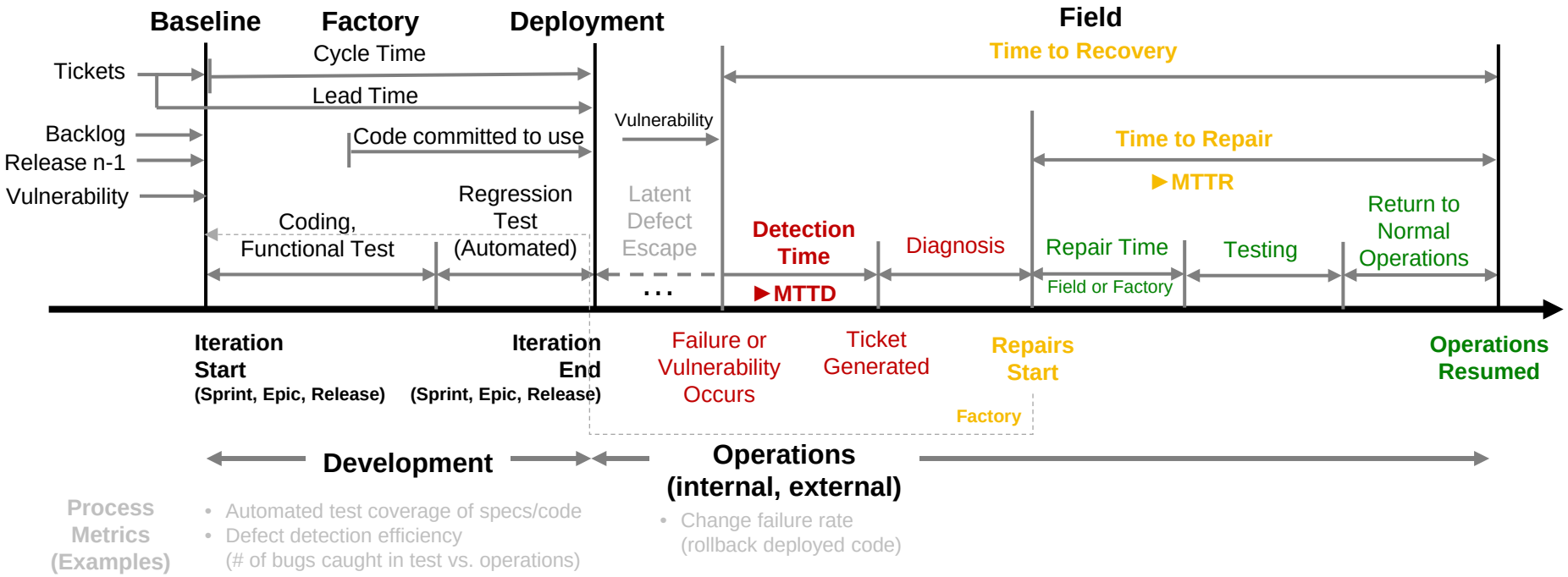
Back

Measures for continuous iterative development should be aligned with information needs, objectives and constraints, at program and enterprise levels



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

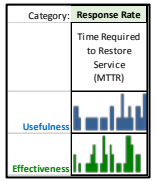


Process Metrics (Examples)

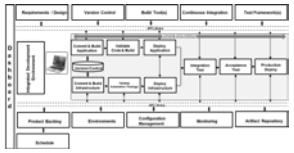
- Automated test coverage of specs/code
- Defect detection efficiency (# of bugs caught in test vs. operations)
- Change failure rate (rollback deployed code)

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)

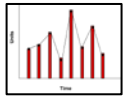


Available measures instrumented and automated by the toolchain



Selection of program measures tailored by information needs (with a few primary colors required by the enterprise)

Enterprise measures driven by business performance objectives



productivity, quality, estimate accuracy, …

Success is measured at multiple levels:

- Mission capability
- Program execution
- Enterprise improvement
- Business results, competitiveness

Adoption

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
- MTTR, MTBF, or MTTF? A Simple Guide to Failure Metrics. https://limblecmms.com/blog/mttr-mtbf-mttf-guide-to-failure-metrics/

Measurement Survey Integration and Analysis (PSM, SERC, INCOSE, NDIA)

Information Needs

Measurement Information Needs for Continuous Iterative SW Development
Interactive Survey of SMEs (Nov 2018 - Feb 2019)
Categories Derived from DIB Recommended Measures
(Importance: 1. VH; 2; H; 3: M; 4: L)

Category:	Deployment Rate					Response Rate				Code Quality				Program Management, Assessment, and Estimation							
Least Favorable (N): / Most Favorable (1):																					
Category Rank	3	2	1	4	5	1	2	4	3	4	1	2	3	4	8	7	1	6	5	2	3
Overall Rank	15	5	2	15	17	8	9	14	10	11	1	3	4	13	21	20	6	19	18	7	12
	Delivery Speed	Update Speed	Patch Speed	Lead Time	Cycle Time	Regression Test	Cyber Test	MTBF, MTTR, Ao	Fix Bugs	Test Efficiency	Baseline Quality	Deploy Quality	Service Delivery	Code Platform / Maint	New vs Reuse	Release Capab	SW Plan Resources	Reqts Stable	Breach Threshold	I&T on Plan	Capability on Plan
Sample Count	58	60	59	60	42	59	58	61	42	61	61	59	63	58	61	61	61	60	41	42	42
Mean	1.91	1.58	1.37	2.25	2.02	1.86	1.91	1.75	1.83	1.95	1.28	1.47	1.79	1.90	2.79	2.18	1.77	2.38	2.27	1.79	1.90
Median	2.00	2.00	1.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	1.00	1.00	2.00	2.00	3.00	2.00	2.00	2.00	2.00	2.00	2.00
Std Dev.S	0.76	0.65	0.67	0.84	0.92	0.75	0.76	0.85	0.76	0.83	0.61	0.68	0.86	0.79	0.99	0.96	0.72	0.78	0.81	0.81	0.66
Var.S	0.57	0.42	0.44	0.70	0.85	0.57	0.57	0.72	0.58	0.68	0.37	0.46	0.75	0.62	0.97	0.92	0.51	0.61	0.65	0.66	0.43
Very high	17	29	42	12	14	21	18	29	15	20	48	36	29	18	8	18	23	8	7	18	11
High	31	28	13	24	16	25	28	20	20	26	10	19	20	31	13	19	30	24	18	16	24
Medium	8	2	3	21	9	13	11	10	6	13	2	3	12	6	24	19	7	25	14	7	7
Low	2	1	1	3	3	0	1	2	1	2	1	1	2	3	16	5	1	3	2	1	0
Total	58	60	59	60	42	59	58	61	42	61	61	59	63	58	61	61	61	60	41	42	42
Weighted Score	1.91	1.58	1.37	2.25	2.02	1.86	1.91	1.75	1.83	1.95	1.28	1.47	1.79	1.90	2.79	2.18	1.77	2.38	2.27	1.79	1.90
4. How would you characterize your relative knowledge and experience with using measures for iterative SW development?	5. How quickly can we deliver initial capability for new products?	6. How quickly can we add and deliver high priority capabilities for an existing operational product?	7. How quickly can new security vulnerabilities be patched and deployed to fielded products?	8. What is the "lead time" duration from code committed to a repository to availability of tested functionality?	9. What is the cycle time to deliver a product?	10. How quickly can a full automated regression test be conducted to verify capability correctness?	11. How quickly can a complete cybersecurity test be successfully completed to ensure adequate resistance to vulnerabilities?	12. What is the reliability and availability of operational service capabilities?	13. How quickly can we address bug reports from the field?	14. The testing efficiency (e.g., defect density, coverage) appropriate relative to project plans? How long does it take to conduct testing within planned constraints?	15. Does new code functionality work as expected and not break previous functionality?	16. What is the quality of code deployed to the field?	17. Are product baseline updates reliable in the field such that operational service is not impacted?	18. Is the code and development platform well structured and maintainable?	19. How much of the code base is newly developed vs. reused from other sources?	20. How much capability has been delivered for each release?	21. Are sufficient resources available to execute the SW development plan? (staff, skills, tools, suppliers)	22. How stable are the mission capability requirements?	23. Are the appropriate thresholds established for SW measures that indicate serious breach relative to baseline plans such that re-evaluation is needed?	24. Is the integration and test progress proceeding as planned?	25. Are the capabilities and features being implemented and completed as planned?
Expertise	Delivery Speed	Update Speed	Patch Speed	Lead Time	Cycle Time	Regression Test	Cyber Test	MTBF, MTTR, Ao	Fix Bugs	Test Efficiency	Baseline Quality	Deploy Quality	Service Delivery	Code Platform / Maint	New vs Reuse	Release Capab	SW Plan Resources	Reqts Stable	Breach Threshold	I&T on Plan	Capability on Plan

Measurement Survey Integration and Analysis

Evaluation of DSB and DIB Measures – Usefulness and Effectiveness: VH=1; H=2; M=3; L=4

DSB Measures

Evaluation and Ranking of DSB Measures
Interactive Survey of SMEs at SERC Workshop, Nov 2019
(Usefulness: 1-4; Effectiveness in Use: 1-4)

Deployment Rate				Response Rate					
Useful	Effective	Useful	Effective	Useful	Effective	Useful	Effective	Useful	Effective
2		3		3		4		5	
5	4	7	7	8	6	14	9	11	14
Sprint Burndown		Epic and Release Burndown		Velocity		Cycle Time (Control Chart)		Cumulative Flow	
54	31	53	34	53	33	54	33	53	34
2.21	2.35	2.09	2.56	2.30	2.39	2.41	2.82	2.55	2.94
2.00	2.00	2.00	3.00	2.00	2.00	3.00	3.00	3.00	3.00
0.84	0.80	0.68	0.82	1.05	0.90	0.99	0.92	0.87	0.98
0.71	0.64	0.46	0.68	1.09	0.81	0.98	0.84	0.76	0.97
10	5	10	3	11	6	9	3	4	4
18	11	26	13	16	11	12	8	12	5
20	14	17	14	20	13	25	14	30	14
6	1	0	4	6	3	8	8	7	11
54	31	53	34	53	33	54	33	53	34
2.41	2.35	2.13	2.56	2.40	2.39	2.59	2.82	2.75	2.94
26. Is Sprint Burn a useful and effective measure?		27. Is Epic and Release Burndown a useful and effective measure?		28. Is Velocity a useful and effective measure?		29. Is Cycle Time (Control Chart) a useful and effective measure?		30. Is Cumulative Flow a useful and effective measure?	

DIB Measures

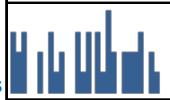
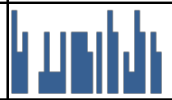
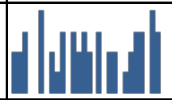
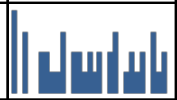
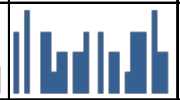
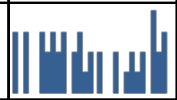
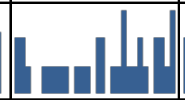
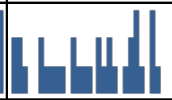
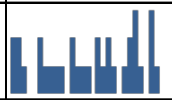
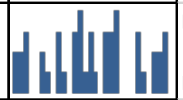
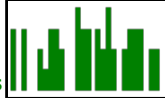
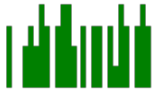
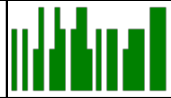
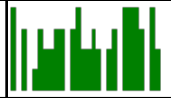
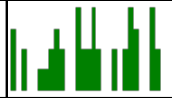
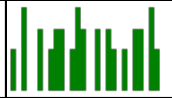
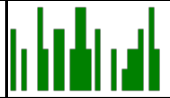
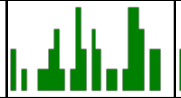
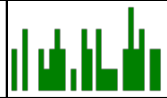
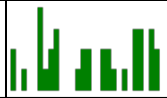
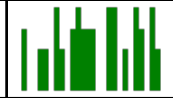
Evaluation and Ranking of DIB Measures
Interactive Survey of SMEs at SERC Workshop, Nov 2019
(Usefulness: 1-4; Effectiveness in Use: 1-4)

Deployment Rate								Response Rate				Code Quality Metrics					
Useful	Effective	Useful	Effective	Useful	Effective	Useful	Effective	Useful	Effective	Useful	Effective	Useful	Effective	Useful	Effective	Useful	Effective
5		4		1		6		2		1		2		1		3	
9	13	6	11	1	5	13	10	10	8	2	3	4	1	3	2	12	12
Time from Launch to MVP (initial lead time)		Time to Field High Priority Functions (incr lead time)		Time to Fix New Security Hole (patch cycle time)		Time from Code Commit to Use (factory cycle time)		Time for Automated Regression Test (reg test cycle time)		Time Required to Restore Service (MTTR)		Automated Test Coverage of Test Specs / Code		# of Bugs Caught in Test vs. Field (defect containment)		Change Failure Rate (rollback)	
52	31	54	35	56	31	53	30	53	31	53	33	55	33	55	30	53	30
2.30	2.90	2.11	2.89	1.72	2.61	2.73	2.83	2.48	2.87	1.70	2.45	1.83	2.24	1.80	2.20	2.55	2.93
2.00	3.00	2.00	3.00	1.00	2.00	3.00	3.00	3.00	3.00	2.00	2.00	2.00	2.00	2.00	2.00	3.00	3.00
0.77	0.98	0.87	0.83	0.88	1.05	0.80	0.91	0.76	0.88	0.74	1.18	0.57	0.97	0.87	1.00	0.87	1.05
0.59	0.96	0.75	0.69	0.78	1.11	0.64	0.83	0.57	0.78	0.55	1.38	0.32	0.94	0.75	0.99	0.76	1.10
7	3	11	1	25	4	2	3	4	1	21	9	9	8	23	8	5	3
24	7	27	11	18	13	17	6	21	11	22	9	33	13	25	12	20	8
19	11	13	14	12	5	24	14	24	10	8	6	9	8	5	6	24	7
2	10	3	9	1	9	10	7	4	9	2	9	4	4	2	4	4	12
52	31	54	35	56	31	53	30	53	31	53	33	55	33	55	30	53	30
2.31	2.90	2.15	2.89	1.80	2.61	2.79	2.83	2.53	2.87	1.83	2.45	2.15	2.24	1.75	2.20	2.51	2.93
31. Is 'Time from program launch to deployment of simplest useful functionality' a useful and effective measure?		32. Is 'Time to field high priority functions (spec > ops)' a useful and effective measure?		33. Is 'Time to fix newly found security hole (find > ops)' a useful and effective measure?		34. Is 'Time from code committed to code in use' a useful and effective measure?		35. Is 'Time required for full regression test (automated)' a useful and effective measure?		36. Is 'Time required to restore service after outage' a useful and effective measure?		37. Is 'Automated test coverage of test specs / code' a useful and effective measure?		38. Is 'Number of bugs caught in testing vs. field use' a useful and effective measure?		39. Is 'Change failure rate (rollback deployed code)' a useful and effective measure?	

Category:
Value
(N) Least Favorable
(1) Most Favorable
Category Rank
Overall Rank
Sample Count
Mean
Median
Std Dev.S
Var.S

Measurement Survey Integration and Analysis

Evaluation of DSB and DIB Measures – Usefulness and Effectiveness: VH=1; H=2; M=3; L=4

	Evaluation and Ranking of DSB Measures			Evaluation and Ranking of DIB Measures									
Category:	Response Rate			Deployment Rate				Response Rate		Code Quality Metrics			
Category Rank	3	4	5	5	4	1	6	2	1	2	1	3	
Overall Rank	8	14	11	9	6	1	13	10	2	4	3	12	
	Velocity	Cycle Time (Control Chart)	Cumulative Flow	Time from Launch to MVP (initial lead time)	Time to Field High Priority Functions (incr lead time)	Time to Fix New Security Hole (patch cycle time)	Time from Code Commit to Use (factory cycle time)	Time for Automate Regression Test (reg test cycle time)	Time Required to Restore Service (MTTR)	Automated Test Coverage of Test Specs / Code	# of Bugs Caught in Test vs. Field (defect containment)	Change Failure Rate (rollback)	
Usefulness													Least Favorable (N): Most Favorable (1):
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 Dev.S	1.05	0.99	0.87	0.77	0.87	0.88	0.80	0.76	0.74	0.57	0.87	0.87	
Var.S	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													Least Favorable (N): Most Favorable (1):
Sample Count	33	33	34	31	35	31	30	31	33	33	30	30	
Mean	2.39	2.82	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 Dev.S	0.90	0.92	0.98	0.98	0.83	1.05	0.91	0.88	1.18	0.97	1.00	1.05	
Var.S	0.81	0.84	0.97	0.96	0.69	1.11	0.83	0.78	1.38	0.94	0.99	1.10	

Metric Summary: Usefulness vs. Effectiveness (1-4)

DSB

All

Sprint Burndown

Useful \ Effective	1	2	3	4	
1	5	2	1	0	8
2	0	5	4	0	9
3	0	4	8	0	12
4	0	0	0	1	1
	5	11	13	1	30

Epic and Release Burndown

Useful \ Effective	1	2	3	4	
1	3	1	1	1	6
2	0	10	4	3	17
3	0	0	9	0	9
4	0	0	0	0	0
	3	11	14	4	32

Velocity

Useful \ Effective	1	2	3	4	
1	5	2	2	0	9
2	0	6	3	0	9
3	0	0	8	1	9
4	0	3	0	2	5
	5	11	13	3	32

Cycle Time (Control Chart)

Useful \ Effective	1	2	3	4	
1	3	4	0	1	8
2	0	4	1	2	7
3	0	0	11	3	14
4	0	0	1	2	3
	3	8	13	8	32

Cumulative Flow

Useful \ Effective	1	2	3	4	
1	2	1	0	1	4
2	2	3	4	1	10
3	0	0	10	4	14
4	0	0	0	4	4
	4	4	14	10	32

Industry

Sprint Burndown

Useful \ Effective	1	2	3	4	
1	5	1	1	0	7
2	0	2	1	0	3
3	0	0	1	0	1
4	0	0	0	0	0
	5	3	3	0	11

Epic and Release Burndown

Useful \ Effective	1	2	3	4	
1	3	1	1	0	5
2	0	4	2	0	6
3	0	0	0	0	0
4	0	0	0	0	0
	3	5	3	0	11

Velocity

Useful \ Effective	1	2	3	4	
1	5	0	1	0	6
2	0	1	2	0	3
3	0	0	1	0	1
4	0	0	0	0	0
	5	1	4	0	10

Cycle Time (Control Chart)

Useful \ Effective	1	2	3	4	
1	3	1	0	1	5
2	0	2	0	1	3
3	0	0	1	1	2
4	0	0	0	1	1
	3	3	1	4	11

Cumulative Flow

Useful \ Effective	1	2	3	4	
1	2	0	0	1	3
2	2	0	1	0	3
3	0	0	2	2	4
4	0	0	0	1	1
	4	0	3	4	11

DIB

All

Time from Launch to MVP (initial lead time)

Useful \ Effective	1	2	3	4	
1	3	1	0	0	4
2	0	6	5	4	15
3	0	0	6	4	10
4	0	0	0	2	2
	3	7	11	10	31

Time to Field High Priority Functions (incr lead time)

Useful \ Effective	1	2	3	4	
1	0	3	3	2	8
2	0	8	7	2	17
3	0	0	4	2	6
4	0	0	0	3	3
	0	11	14	9	34

Time to Fix New Security Hole (patch cycle time)

Useful \ Effective	1	2	3	4	
1	4	4	3	3	14
2	0	7	0	2	9
3	0	2	2	3	7
4	0	0	0	1	1
	4	13	5	9	31

Time from Code Commit to Use (factory cycle time)

Useful \ Effective	1	2	3	4	
1	0	2	0	0	2
2	0	3	5	1	9
3	3	1	9	1	14
4	0	0	0	5	5
	3	6	14	7	30

Time for Automated Regression Test (reg test cycle time)

Useful \ Effective	1	2	3	4	
1	1	1	0	1	3
2	0	9	3	1	13
3	0	1	7	5	13
4	0	0	0	2	2
	1	11	10	9	31

Time Required to Restore Service (MTTR)

Useful \ Effective	1	2	3	4	
1	9	2	2	2	15
2	0	7	3	6	16
3	0	0	1	0	1
4	0	0	0	1	1
	9	9	6	9	33

Automated Test Coverage of Test Specs / Code

Useful \ Effective	1	2	3	4	
1	6	2	0	1	9
2	2	9	7	3	21
3	0	2	1	0	3
4	0	0	0	0	0
	8	13	8	4	33

# of Bugs Caught in Test vs. Field (defect containment)

Useful \ Effective	1	2	3	4	
1	7	5	2	0	14
2	1	7	0	2	10
3	0	0	4	0	4
4	0	0	0	2	2
	8	12	6	4	30

Change Failure Rate (rollback)

Useful \ Effective	1	2	3	4	
1	0	2	3	0	5
2	0	5	0	1	6
3	3	0	4	8	15
4	0	0	0	3	3
	3	7	7	12	29

Industry

Time from Launch to MVP (initial lead time)

Useful \ Effective	1	2	3	4	
1	1	0	0	0	1
2	0	3	1	2	6
3	0	0	2	2	4
4	0	0	0	0	0
	1	3	3	4	11

Time to Field High Priority Functions (incr lead time)

Useful \ Effective	1	2	3	4	
1	0	0	2	1	3
2	0	3	2	1	6
3	0	0	0	1	1
4	0	0	0	1	1
	0	3	4	4	11

Time to Fix New Security Hole (patch cycle time)

Useful \ Effective	1	2	3	4	
1	2	0	3	1	6
2	0	1	0	1	2
3	0	2	0	0	2
4	0	0	0	0	0
	2	3	3	2	10

Time from Code Commit to Use (factory cycle time)

Useful \ Effective	1	2	3	4	
1	0	0	0	0	0
2	0	1	3	0	4
3	0	0	3	1	4
4	0	0	0	2	2
	0	1	6	3	10

Time for Automated Regression Test (reg test cycle time)

Useful \ Effective	1	2	3	4	
1	0	1	0	0	1
2	0	4	1	1	6
3	0	0	4	0	4
4	0	0	0	0	0
	0	5	5	1	11

Time Required to Restore Service (MTTR)

Useful \ Effective	1	2	3	4	
1	2	0	1	1	4
2	0	1	2	3	6
3	0	0	1	0	1
4	0	0	0	0	0
	2	1	4	4	11

Automated Test Coverage of Test Specs / Code

Useful \ Effective	1	2	3	4	
1	4	0	0	1	5
2	0	3	1	2	6
3	0	0	0	0	0
4	0	0	0	0	0
	4	3	1	3	11

# of Bugs Caught in Test vs. Field (defect containment)

Useful \ Effective	1	2	3	4	
1	2	0	1	0	3
2	1	1	0	1	3
3	0	0	2	0	2
4	0	0	0	1	1
	3	1	3	2	9

Change Failure Rate (rollback)

Useful \ Effective	1	2	3	4	
1	0	0	0	0	0
2	0	2	0	0	2
3	3	0	1	3	7
4	0	0	0	1	1
	3	2	1	4	10

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

Software Procurement & Project Management

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 analys of 25000 user stories across 70 projects

Get the requirements as good as you can as early as you can

Cognitively intensive

Changing midstream is disruptive

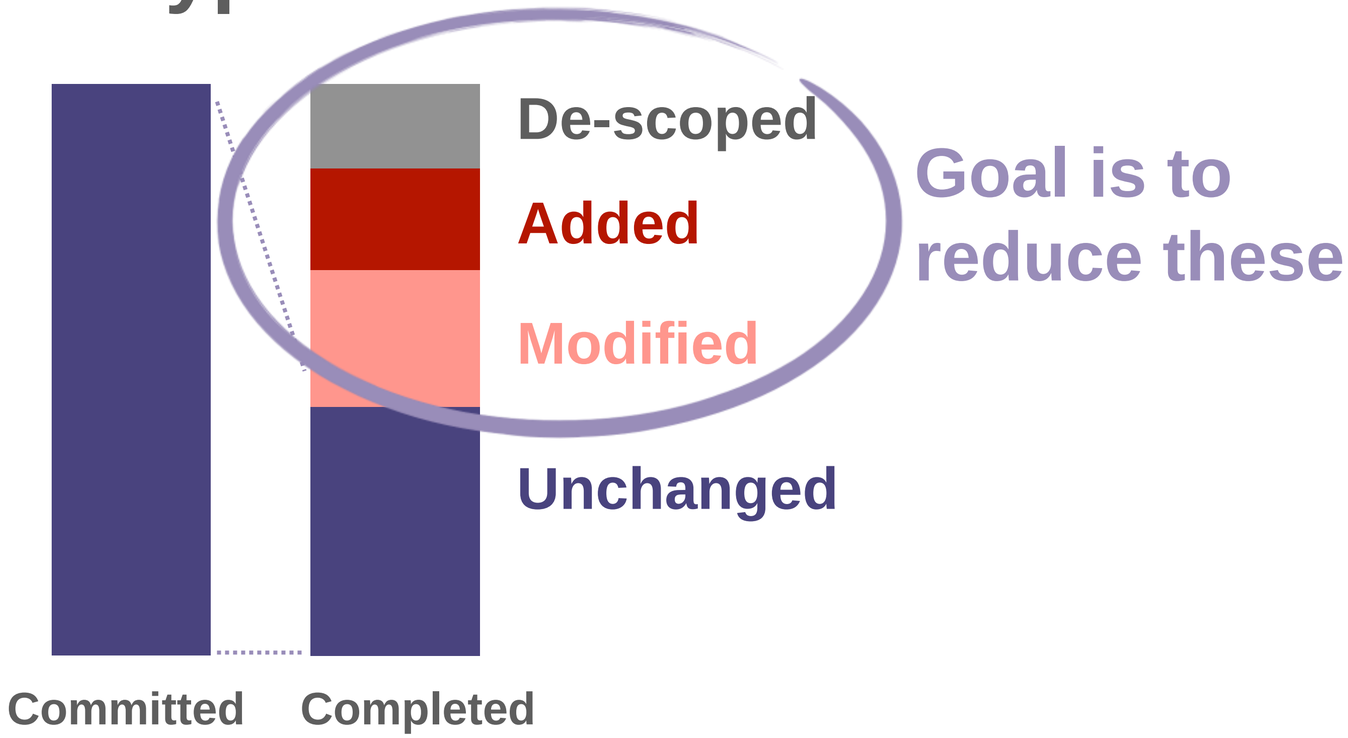
Evolve

Commonly

2%

Per month

Typical



Example User Story

Requirements in Agile

“User Stories” are the catalyst of the conversation.

Add Delivery Details **SP**

As a … Site visitor

I want … Add my delivery addresss

So that … I can receive my goods

Acceptance/Test Criteria …

I can click pencil to enter my zip code and full home address

Back

Who & what

Why

Given , when & then

COSMIC

I set out to automate the functional sizing of user stories

Discovering functional intent

Automated Functional Sizing

COSMIC

Delete Profile

As a …	Administrator
I want …	Delete a profile……
So that …	I can receive my goods

Deleting a profile

Estimated CFP: 6 CFP

As an **administrator** I want to delete a **profile** . Then the **system** should send a **confirmation email** to the **team** .

Edit | Functions 2 | Revisions 9 | Tests 14 | Related | Debug

Functional Steps	Interpretation	Data Movements (6 CFP)
delete **profile**	Delete **profile**	E request delete W delete **profile** X return error/confirmation
send **confirmation email**	Read **confirmation email**	E **confirmation email** id R read **confirmation email** from storage X display **confirmation email**

Who & what → Functions → Data movements

Analysing the text

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"

products

CSV
No setup

refine ScopeMaster

CSV

Benefits:

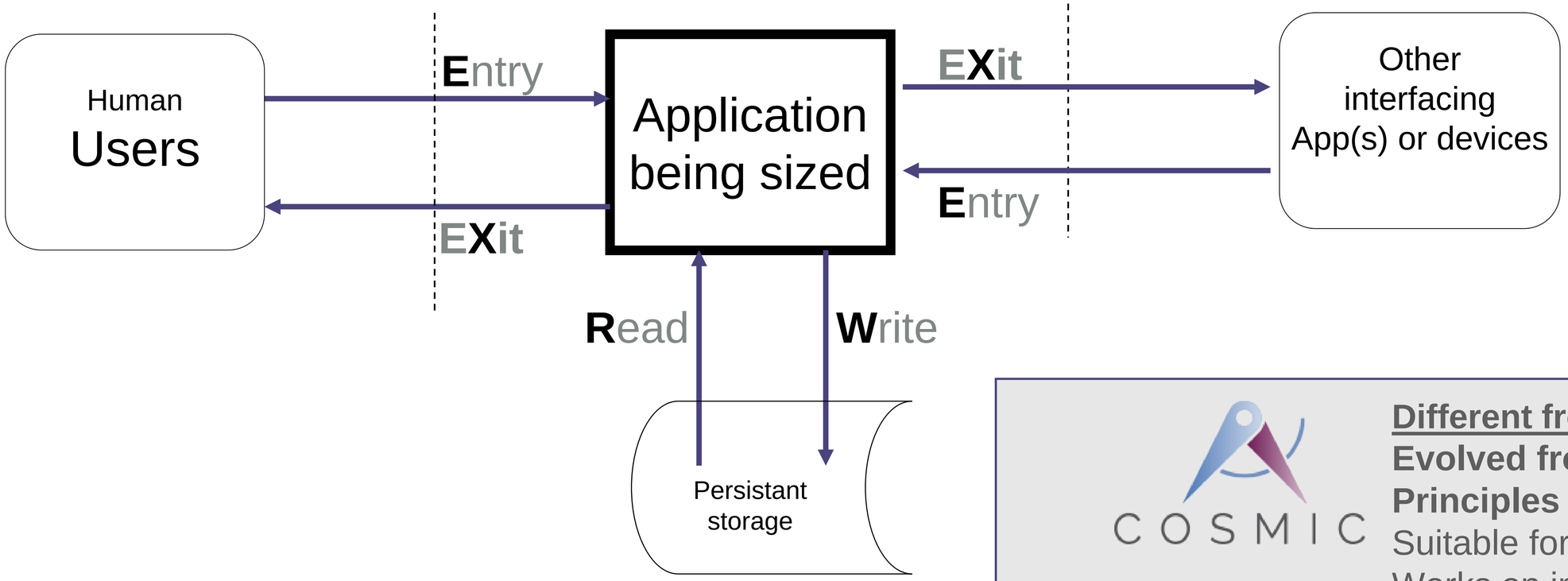
Sizing

- 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

COSMIC Functional Sizing - the successor to IFPUG



$$\sum E, X, R, W = CFP$$

https://cosmic-sizing.org

Different from IFPUG
Evolved from inc improvements
Principles not rules
Suitable for all S/W
Works on incomplete / **Agile**
Open source

2016 NIST - canonical reference reference for a FP
https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8101.pdf
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-307.pdf

Sizing software

Functional Size Metrics on Software Projects

Agile Story Points

SLOC

RICEFW

IFPUG FP

COSMIC Function Points

Very Flawed

- Not Valid
- Inconsistent
- Easy to game

Flawed

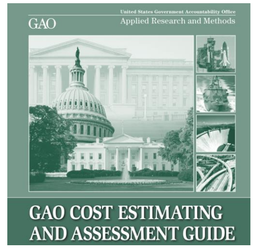
- Not Valid
- Inconsistent
- Easy to game

Good

- ISO Standard
- Consistent
- User stories insufficient
- Not ideal for embedded

Best

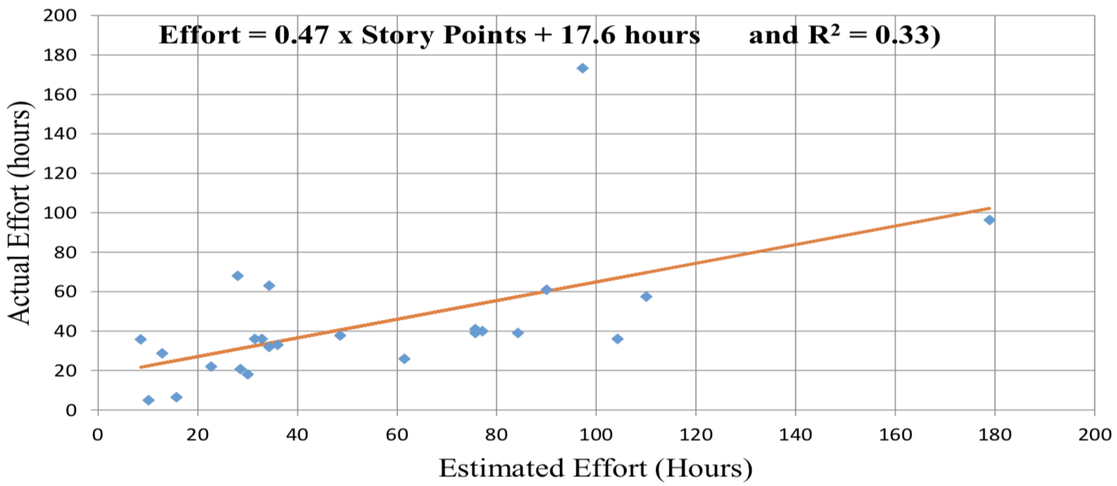
- ISO Standard
- Incomplete OK
- Principle-based
- Automated
- US. GAO Recommended



Case study to compare SP vs CFP

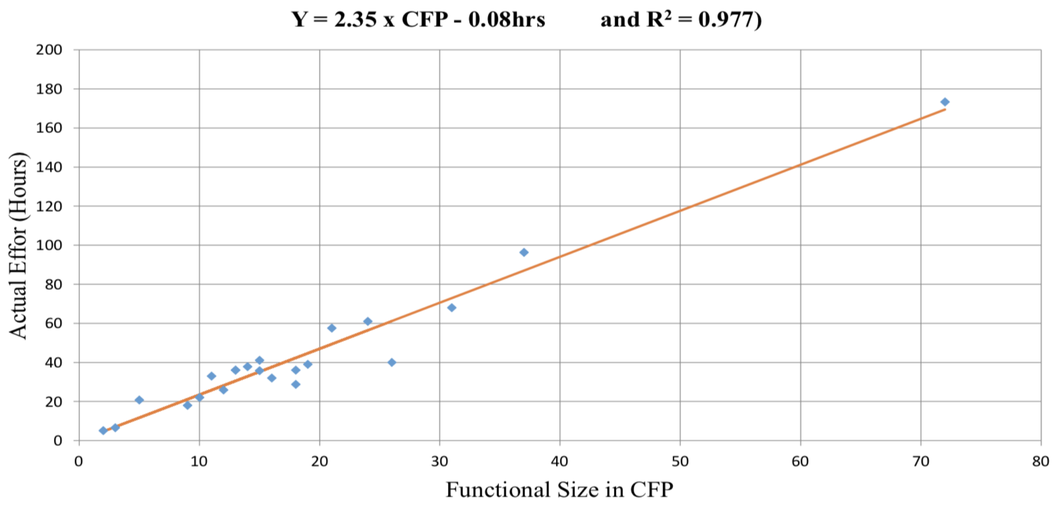
Story points vs actual effort

$R^2 = 0.33$



CFP vs actual effort

$R^2 = 0.97$



Conclusion:

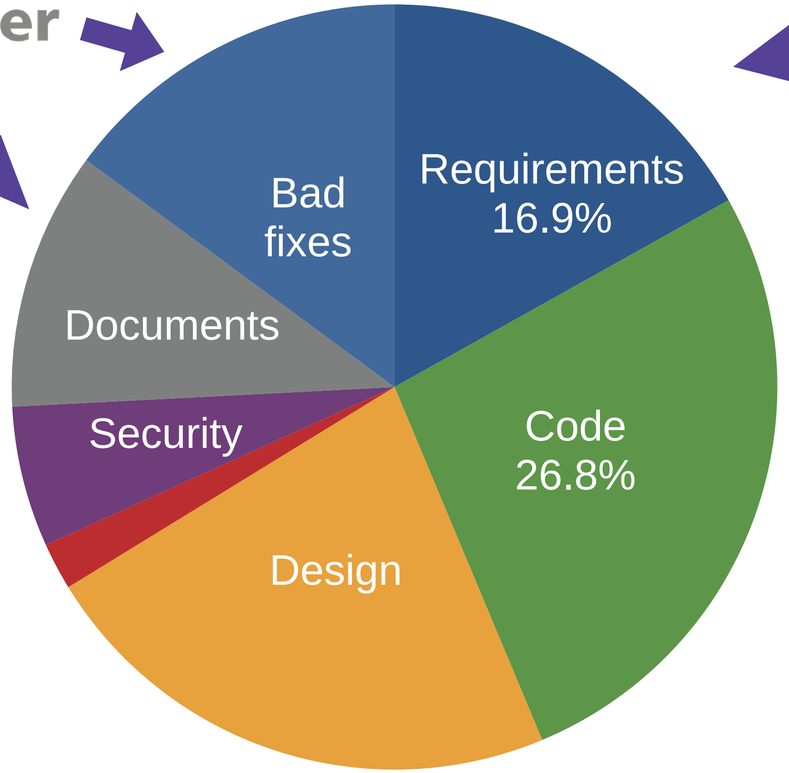
CFP is a better predictor of effort than story points.

C. Commeyne, A. Abran, R. Djouab. "Effort Estimation with Story Points and COSMIC Function Points - An Industry Case Study", Software Measurement News, Vol 21, No. 1, 2016 *

Typical Source of Defects on Software Projects

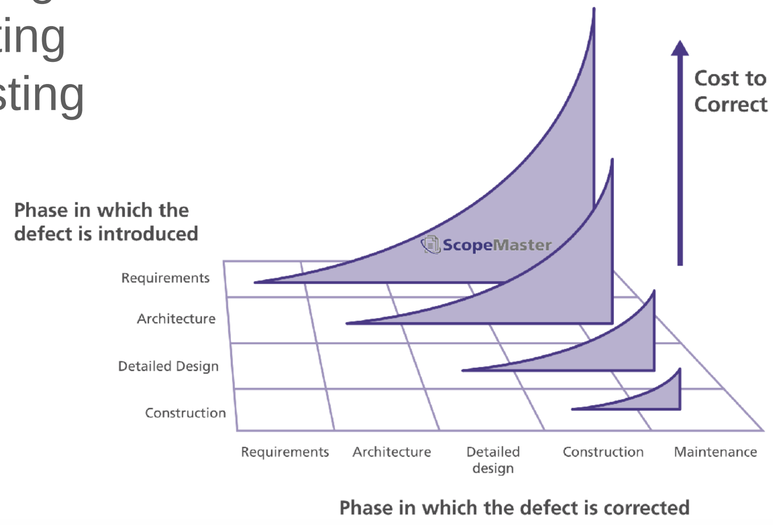
ScopeMaster
Helps

ScopeMaster
Targets

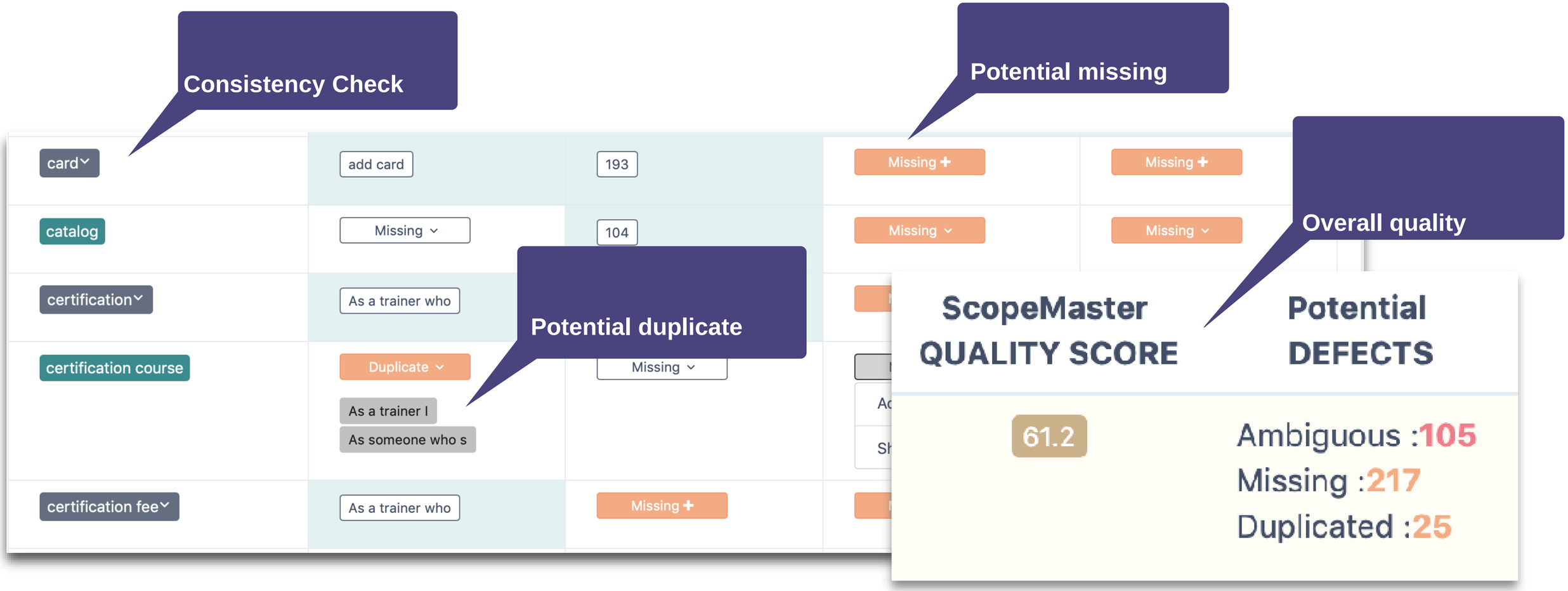


<u>Most Activity</u>
Unit testing
Systems testing
Functional testing
End to end testing
Acceptance testing

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

The key metrics

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

Benefits of just knowing the software / project size up front

Value of knowing the size

Given a typical Cost of $2,000 per CFP

	Indicative Benefits	
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:

Based on	Tool	Additional Benefits	Links
Pre-requirements	Capers Jones' Software Risk Master		Namcook Analytics LLC
High level written **requirements**	ScopeMaster	1.Requirements QA 2.Functional test generation	ScopeMaster Ltd
Matlab Simulink **Designs**	Proprietary - Renault		slides & video
Code	CAST Software Intelligence for Digital Leaders	1.Code quality analysis	Castsoftware

Summary

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

OFFICE OF THE DIRECTOR OF NATIONAL INTELLIGENCE

Estimating SW Costs from Requirements using Objective Function Points Research

Overall Classification

UNCLASSIFIED

Overview

- 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)?

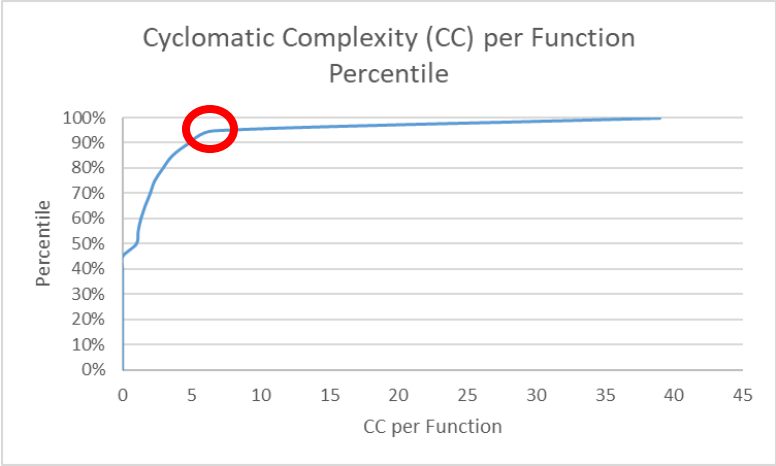
- Developing an Automated Objective Function Point Counter
  - Developing a standard automated approach to counting OFPs to avoid subjective estimates
    - » Standard Function Points (FP) require Function Point experts to derive
    - » Used UCC tool baseline since it already parses through most SW 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

Complexity Mapping to OFPs

- 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



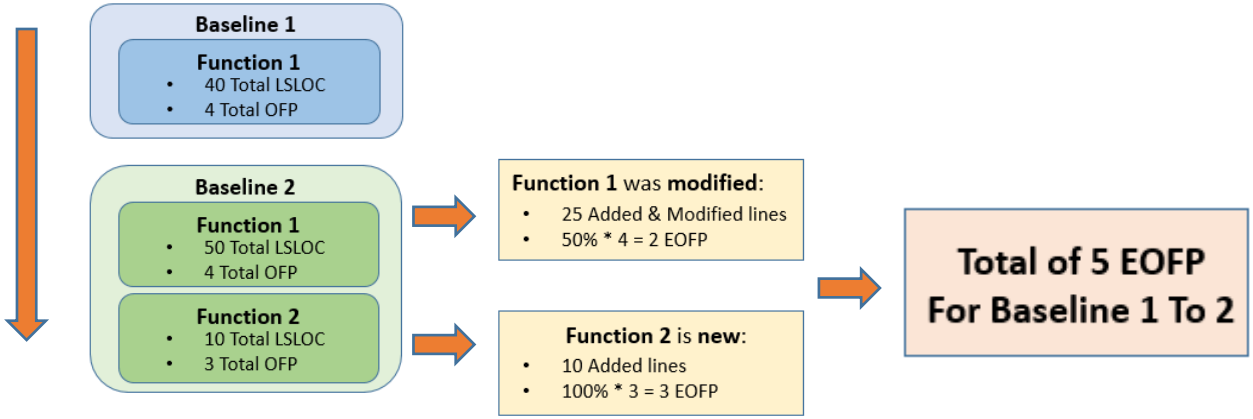
McCabe Complexity Values	McCabe Complexity Definitions	Traditional Function Point Mapping Definitions
1 – 10	• Structured and well written code • High Testability • Cost and Effort is Less	• External Input (EI): Functions that move data into the application without presenting data manipulation.
11 – 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)

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**



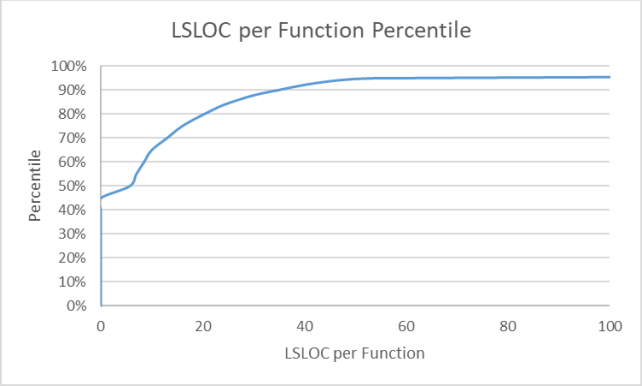
NASA's General Missions Analysis Tool (GMAT) Example

- Results from comparison between GMAT C++ code baselines 2017a and 2018a

	2017a	2018a
Physical Source Lines Of Code (PSLOC)	457,112	535,661
Logical Source Lines Of Code (LSLOC)	290,674	318,173
Delta LSLOC	27,499	
Modules (Classes)	3,104	3,191
OFPs	70,953	74,099
Delta OFPs	3,146	
DIFF Results		
NEW LSLOC	9.3%	
DELETED LSLOC	1.0%	
MODIFIED LSLOC	0.7%	
UNMODIFIED LSLOC	89.0%	
EOFPS	3,023.3	

- 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**

- 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**

Maintainability Index

85 and more:	**good maintainability**
65-85:	**moderate maintainability**
under 65:	**difficult to maintain**

bad pieces of code (big, uncommented, unstructured) the MI value can be even negative

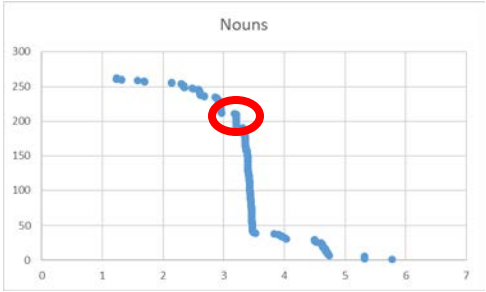
We are opening the door to new metrics to estimate effort

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)



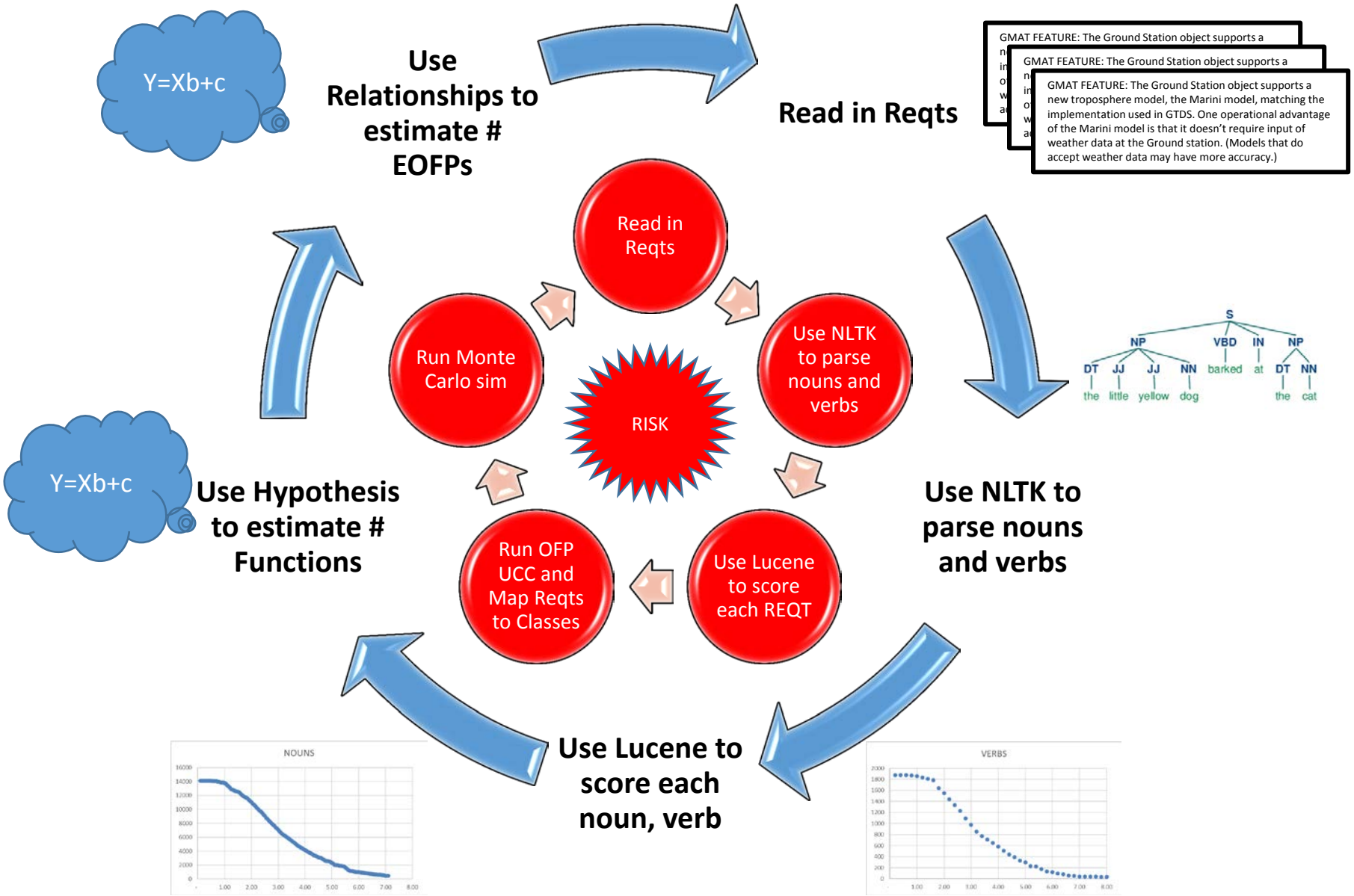
- # Functions currently do not trend to Unique Verbs
  - Verbs such as “Get or “Set” are used many times
  - Requires more research
- As previously noted, there are **6.98** Functions per Module (Class) in GMAT
  - Currently observing other programs in this range **(6 – 7)**
  - Multiplying the 211 Total 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**

Estimating Risk from Requirements

- 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



Overview of Accomplishments

- 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

BACKUP

BACKUP

GMAT Requirements for Unit Level Testing

Groundtrack Plot*	FRR-42.1.0	The system shall allow the user to choose among the following objects as the central body of a ground track plot:
	FRR-42.1.1	1) Default Celestial Body
	FRR-42.1.2	2) User-defined Body
	FRR-42.2.0	The Ground Track Plot shall draw the longitude and latitude time-history for the following object types:
	FRR-42.2.1	1) Spacecraft
	FRR-42.2.3	2) Groundstation
	FRR-42.3.0	The system shall display icons on the ground track to indicate the locations of the following object types:
	FRR-42.3.1	1) Spacecraft
	FRR-42.3.2	2) Groundstation
	FRR-42.4.0	The system shall display object labels next to the icons for the following object types:
	FRR-42.4.1	1) Spacecraft
	FRR-42.4.2	2) Groundstation
	FRR-42.5.0	The system shall allow the user to define the data plotting options for a ground track plot:
	FRR-42.5.1	1) The number of integration steps to skip between plot points
	FRR-42.5.2	2) The number of plot points to collect before updating a ground track plot
	FRR-42.5.3	3) The number of plot points to retain and redraw during propagation and animation.
	FRR-42.6.0	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:
	FRR-42.6.1	1) Show all iterations/perturbations
	FRR-42.6.2	2) Show current iteration/perturbation only
	FRR-42.6.3	3) Show solution only
	FRR-42.7.0	The system shall allow the user to specify a texture map using the following options
	FRR-42.7.1	1) Use default texture map for central body
	FRR-42.7.2	2) Use user-defined texture map.
	FRR-42.8	The system shall optionally display or not display a configured ground track plot
	FRR-42.9	The Ground Track Plot shall display the epoch in UTC Gregorian format
	FRR-42.10	The system shall allow the user to animate the Ground Track Plot after a run is complete
	FRR-42.11	The system shall display the latitude and longitude values when the cursor is placed over a GroundTrackPlot.

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**

Table 1 Function Complexity Matrix (Shared table between EI)

Files Type Referenced (FTR)	Data Elements Types (DETs)		
[redacted]	1-4	5-15	Greater than 15
Less than 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
Greater than 2	Average (4)	High (6)	High (6)

Table 2 Function Complexity Matrix (Shared table between EO , EQ)

Files Type Referenced (FTR)	Data Elements Types (DETs)		
[redacted]	1-5	6-19	Greater than 19
Less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 3	Average (5)	High (7)	High (7)

Table 3 Function Complexity Matrix (Shared table ILF)

Record Element Type (RET)	Data Elements Types (DETs)		
[redacted]	1-19	20-50	51 or More
1	Low (7)	Low (7)	Average (10)
2 to 5	Low (7)	Average (10)	High (15)
6 or More	Average (10)	High (15)	High (15)

Table 4 Function Complexity Matrix (Shared table EIF)

Files Type Referenced (FTR)	Data Elements Types (DETs)		
[redacted]	1-19	20 - 50	51 or More
1	Low (5)	Low (5)	Average (7)
2 to 5	Low (5)	Average (7)	High (10)
6 or More	Average (7)	High (10)	High (10)

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

McCabe Complexity Values	McCabe Complexity Definitions	Traditional Function Point Mapping Definitions
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)

More Complex

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:

	RET	FTR	DET	CC	OFP
FILE/Method1.1	2	1	10	10	3
FILE/Method1.2	3	2	20	20	7
FILE/Method2.1	5	3	35	45	10
TOTAL					20

- Where:
  - » RET: Record Element Type
  - » FTR: File Type Reference
  - » DET: Data Element Type
  - » CC: Cyclomatic Complexity
  - » OFP: Objective Function Points

McCabe Complexity Values	McCabe Complexity Definitions	Traditional Function Point Mapping Definitions
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)

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.

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.

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.

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

What We Do

Develop a pragmatic/defendable approach to estimate and measure the cost of software development – Simplified Function Point Analysis (SFPA)

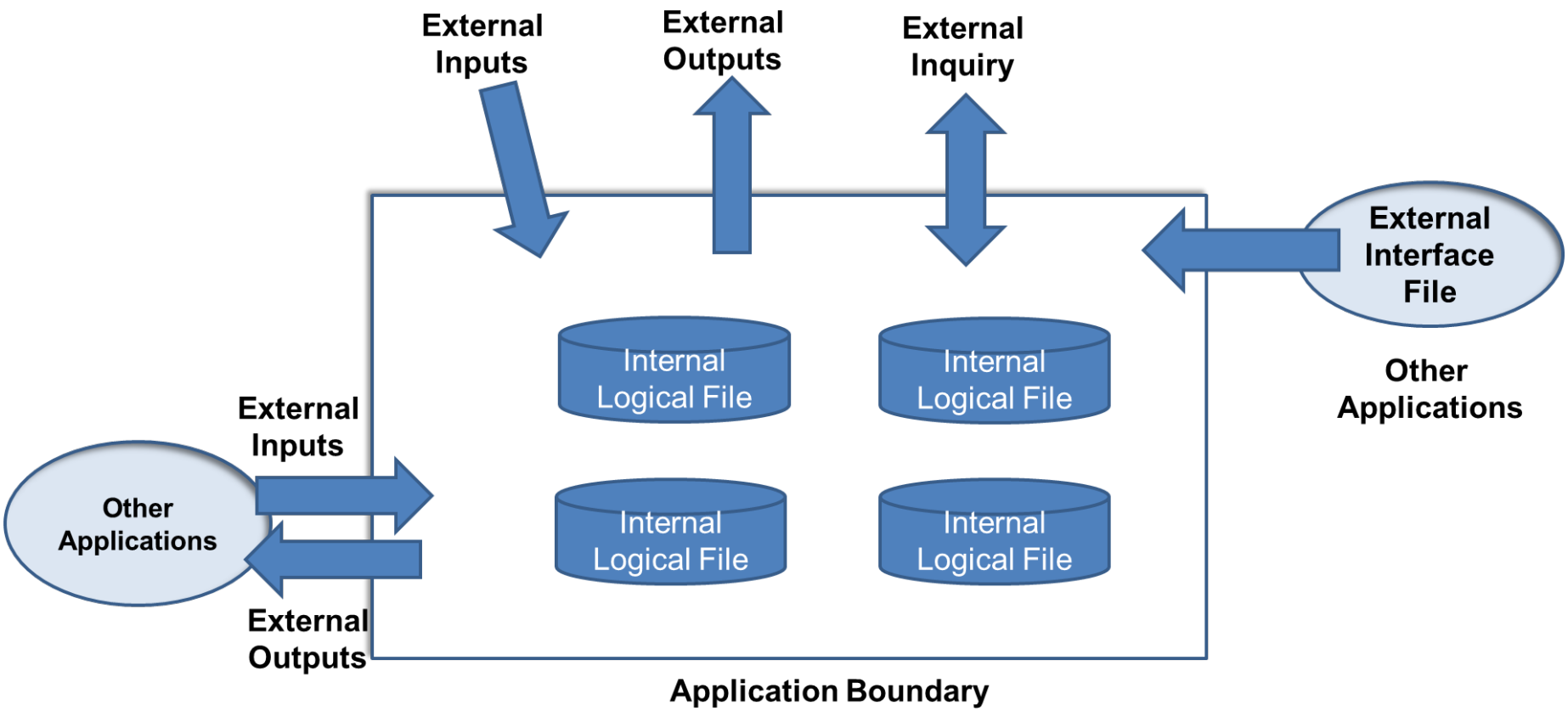
- Deliver SFPA training to the Cost Community as requested
- Standardize and expand the lexicon of verbs for the Pattern Matching Technique used in SFPA

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)

IFPUG Components	Low	Average	High
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6
Internal Logical Files	7	10	15
External Interface Files	5	7	10

SFPA Components	Weighting Factor
Elementary Processes (Create, Update, Delete, Report, Read)	4.6
Logical Data Groups (Saves)	7

* www.sifpa.org/en/

Simple Function Point Sizing

Count	Action	# Transactions	Create (EI)	Update (EI)	Delete (EI)	Report (EO)	Read (EQ)	Save	AFP
10	Approve	2		1			1	1	162
13	Create	2	1				1	1	211
1	Generate	1				1			5
3	Monitor	1					1		14
2	Provide	1				1			9
1	Providing	1				1			5
5	Request	1	1						23
2	Retrieve	1					1		9
15	Review	1					1		69
1	Search	1					1		5
8	Submit	2		1		1		1	130
2	Update	1		1				1	23
3	Validate	3		1		1	1	1	62
9	View	1					1		41

$$\boldsymbol{Size} = \sum (\mathbf{4.6}\ \boldsymbol{x\ Transations}) + (\mathbf{7.0}\ \boldsymbol{x\ Save})$$

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