Software Data Collection and Analysis for Proposal Evaluation

2018 ICEAA Annual Conference & Training Workshop - Software Estimating Track (SW06)

Kenneth Rhodes, Eric Hippensteel

INTRODUCTION

Software development is an area of chronic cost growth for DoD acquisition programs. Analysis of completed software development programs indicates that percent change in actual versus estimated development cost ranges from 8% to 231% and averages 98% (GAO, pg 21). A significant contributor to the growth has and continues to be unrealistic software sizing estimates, the source of which can be the Government or industry software development teams responsible for the work.

This paper discusses the use of a software development data collection form that Government proposal evaluators can use to assess the credibility of industry software development bids. The Office of the Under Secretary of Defense for Acquisition, Technology & Logistics (OUSD (AT&L)) Defense Procurement and Acquisition Policy (DPAP) recognizes software data collection and evaluation as a DoD acquisition best practice for its ability to validate contractor proposals efficiently. The form serves as a template to enable analysts to assess if the contractor's estimates are consistent with actual data and to gain a better understanding of the proposed effort. Software labor data included in this form enables the Government to assess the validity of proposed software development effort, productivity, sizing, and schedule. Analysis enabled by this data collection form supports fact finding during evaluation, enabling the Government to ask the right questions early in the evaluation process.

SOFTWARE DATA COLLECTION

There are numerous software data collection and evaluation tools, including Cost Analysis Requirements Description (CARD), Request for Proposal (RFP) data collection forms, and SRDRs, each of which have different objectives and provide value for specific efforts in a program's acquisition lifecycle. The data collection process this paper promotes is specifically for pre-solicitation activities (i.e. RFP development) and proposal evaluation. The proposal evaluation, or source selection in competitive contracting parlance, includes an "in-depth review of each proposal against the factors and sub-factors established in the solicitation, and assigns evaluation ratings" (DoD, pg 23). Requiring software development data from industry in

proposal submissions greatly enhances the depth of evaluation possible during source selection. The technique of requesting additional vendor data via a template (e.g., MS Excel table) included in the RFP enables the Government to collect well-defined data in a standardized format that supports data normalization, risk identification and value adjustments.

The CARD defines the features of a program that a cost analyst needs to develop a cost estimate. Prior to the release of an RFP, an Independent Government Estimate (IGE) or Independent Cost Estimate (ICE) may be required to assess the cost realism of the program compared to the budget. The organization developing either of these estimates needs specifics about the planned software development effort, including but not limited to information such as sizing, language and development process. There are two types of CARD Tables pertaining to software data collection: Software Development Tables and Software Maintenance Tables.

The Software Development Table "includes software development cost drivers" (CADE, pg 2). These tables identify software size in either a Source Lines of Code (SLOC)-based approach that includes metrics such as Government Furnished Code or Carryover Code or a Reports, Interfaces, Conversions, Enhancements, Forms and Workflows (RICE-FW) Object-based approach that includes metrics such as Reports or Workflows.

The Software Maintenance Table "includes software maintenance cost drivers" which are "necessary for robust software maintenance cost estimates" (CADE, pg 2). The maintenance data also includes SLOC-based or RICE-FW Object- based software sizing data and provides Product Quality data (e.g., number of software changes implemented and software change volatility), as well as Software Licensing data (e.g., quantity of licenses and total cost of all licenses).

DoD contractually requires prime contractors and major subcontractors to deliver Software Resources Data Reports (SRDR) for Major Defense Acquisition Programs (MDAPs) with software development costs greater than $20M during development contract execution. The intent of the SRDR is to "collect objective, measurable data commonly used by industry and DoD cost analysts" in order to "compile a repository of estimated and actual software product sizes, schedules, effort, and quality" (Software Resources Data Reporting, pg 2) that enables more accurate, robust cost estimates for future software development efforts. In order to accomplish this objective, the SRDR includes two key data reports: the Software Development Report (SDR) and the Software Maintenance Report (SMR). The SDR collects various software measures, including multiple variations of SLOC, as well as person-hours expended during the development phase. Similarly, the SMR collects software measures and person-hours expended during the maintenance phase.

Each of the software data collection and evaluation tools-- CARD, RFP data collection forms, and SRDR -- have different objectives and value to the acquisition process. Applying the tools throughout the lifecycle of a system (development, production and deployment, and operations and sustainment) contributes to more realistic software sizing and software

development effort estimates. Tailoring these tools to the specific circumstances of the program can support effective Program Manager (PM) decision-making that monitors and controls software growth during execution. Another tool, the "NCCA Software Program Definition Form" and its associated data field dictionary, "can be utilized to obtain information on the program being estimated" (Cummings, pg 11) as the program matures. The CARD is most valuable to the cost analyst developing the IGE for the PM and Milestone Decision Authority (MDA). The RFP data collection form allows the Cost/Price Analysis Team (C/PAT) analyst to perform data analysis and visualization for contracting officers and the Source Selection Authority (SSA). The SRDR shows requirements volatility and size growth by software build for the PM to assess program health or as historical data for cost analysts to estimates similar programs in the future.

SLOC SUMMARY TABLE EXAMPLE

Including a data collection form (MS Excel or other template) as a Section J Attachment in the RFP improves the evaluation of proposed software development efforts (process, productivity, and Basis of Estimate (BOE) traceability). The following structure provides an example of a software development data request form. For each RFP, the form is tailorable to program and source selection specific requirements.

Section 1: Work Breakdown Structure (WBS) & Computer Software Configuration Item (CSCI)

- Report the effort for all software and firmware efforts by WBS, traceable to the Cost and Software Data Report (CSDR) Plan, if applicable.
- Report CSCI effort at the Computer Software Component (CSC) level or Computer Software Unit (CSU) level in order to document historical SLOC planned for modification or reuse.

Section 2: Activity Description (Requirements and BOE Traceability)

- Delineate Prime / Subcontractor efforts.
- Define the technical capabilities of each CSCI and mapping to SOW matrix and requirements.
- Provide traceability to BOEs.

Section 3: Historic Effort Performance & Description

- Deliver SLOC (or other size measure) for historic efforts by mapping the completed functions to new capabilities, where reuse code is proposed.
- Document SLOC size by prescribed code counting method (require same method for all offerors to ensure consistent counting practices and type (Physical versus Logical)).

- Document historic effort by programming language, period of performance, contract number, total hours, Design, Code, Test and Integration (DCTI) hours, non-DCTI hours, End-to-End or DCTI productivity, and defect rate.

Section 4: Proposed Effort (Size)

- Document the amount and applicability of planned reused code.
- Proposed New, Modified, and Reuse Delivered SLOC.

Section 5: Proposed Effort (Effective/Equivalent SLOC)

- Document equivalent new SLOC (ESLOC) conversions, including supporting data and calculations.

Section 6: Proposed Hours (Productivity)

- Document productivity (ESLOC/Hour) metrics using historical performance with data and calculations for supporting adjustments, if required.

Note: Provide additional instructions and data field definitions as need to support accurate data collection. The more prescribed the form is, the less variation in offeror submissions.

IMPLEMENTATION

Utilizing an RFP Section J attachment for software data collection leverages the competitive RFP environment to collect information that may otherwise be difficult to obtain for software intensive weapon systems or Major Automated Information System (MAIS) development efforts. Typically, a proposal evaluation only considers data submitted as part of the proposal and thereby excludes use of an external estimating resource such as an SRDR database or SLOC-based effort estimating tools. Collecting software size and productivity information prior to and during sole source contract awards also provides leverage to support contract negotiations based on historical contractor performance and realistic effort estimation. Key to the data collection template for both competitive and sole source evaluations is identifying and defining the data fields for the software development effort. For example, a program with a high amount of anticipated reuse SLOC should identify the basis for any assumptions used to derive DCTI hours to reuse the SLOC (i.e. historical reuse size, current product line productivity, reuse definition and effort relationship). The evaluator is more capable of assessing the realism of the proposed effort with the additional proposal information.

In addition to defined data fields for software intensive weapon system and MAIS development efforts, the WBS within the data collection form should request data at the lowest reportable level, the Computer Software Unit (CSU). The CSU is the smallest subdivision of a Computer Software Configuration Item (CSCI) for the purposes of engineering management.

The WBS decomposes the effort and aligns with the program WBS referenced in other sections of the RFP. Implementing a WBS specific to system type and tailoring data fields, including multiple software development processes (agile, incremental, waterfall, etc.), and counting methods (SLOC (consistent counting methods), Function Points (size, not effort!) and Agile Metrics (User Stories, Features, etc.) ensures consistent depth of evaluation for all offerors' development approaches. The data field definitions will prescribe the units of measure, ensuring consistency across proposals. For SLOC measurement, the University of Southern California (USC) Unified Code Counting (UCC) ensures "consistency across independent organizations in the rules used to count software cost code is often difficult to achieve. To that end, the USC Center for Systems and Software Engineering (CSSE) has developed and released a code counting toolset called CodeCount to support sizing software code for historical data collection, cost estimation, and reporting purposes. This toolset is a collection of tools designed to automate the collection of source code sizing information" (UCC User Manual, 1). Employing the UCC for historical measurements / basis in the RFP data collection form provides normalized insights into the proposed software development effort.

Providing C/PAT analysts with the historical basis for proposal estimates and the data behind adjustment factors is a DPAP best practice; "employing a method to understand and define the efficient use of lines of software code (a Software Lines of Code Table was required as part of the cost proposal) provided helpful insight into the offerors' proposals" (DPAP, R242). The practice also benefits the Source Selection Evaluation Board (SSEB) that can trace requirements to CSCIs/CSUs and review the source/legacy SLOC and proposed hours. Analysts supporting C/PAT and SSEB evaluations can identify areas of strength (high reuse) and risk (low reuse, low productivity). Including a system-specific, i.e. tailorable, WBS with the data fields further decomposes the effort for realism adjustments and tradeoff analysis.

SOFTWARE DATA ANALYSIS

Throughout the proposal evaluation process, analysis of all proposal artifacts informs the validation of BOEs, risk identification and realism adjustments. The C/PAT analyst is able to develop discussion questions regarding discrepancies and weaknesses. For example, the C/PAT analyst may identify potential hours adjustments based on realistic productivity or reuse assumptions. Conversely, the C/PAT analyst may identify strengths, such as when high performing software development organizations include and justify variances from historical performance in the proposed effort estimation, e.g., risk adjustments due to increased complexity or staff scaling requirements. The SLOC summary table data informs C/PAT analysts understanding of the proposed effort, supports asking insightful questions, and facilitates synthesizing voluminous proposal data into dynamic views. The list that follows represents the types of questions that the software data collection form enables C/PAT analysts to explore when evaluating a proposal (i.e., software related BOEs) against reported historical performance:

- Is proposed reuse code less than the total DSLOC on the source program?
- How does the proposed software development productivity align with previous performance?
- Is it realistic to achieve greater productivity on a new, more complex system, than previous efforts (even with new software development methods such as agile)?
- Are the DCTI hours proportional to the non-DCTI hours?
- Is there effort proposed for all software development activities?
- Is the historical data decomposition, ideally at the CSU level, in alignment with the proposed effort?
- Are adjustment factors for productivity or reuse efforts justified by the historical data provided?
- Are the BOEs adjustment factors based on analogous data or industry standards from 'black box' software estimating tools?

The data collection form also facilitates evaluating each proposal against RFP technical requirements. For example, quantifying the effort to meet or exceed a capability by demonstrating past performance instead of qualifying engineering judgement. Customizing the form provides additional details for lower level efforts, such as software and firmware efforts for cyber security and cloud computing architecture. While the SSEB support will also find the WBS and hour estimates useful, the C/PAT analysis must "ensure that offerors' cost proposal and cost data is safeguarded and kept separate from the technical data." (NAVSEA Source Selection Guide, pg 14). The SSEB can communicate technical risks, and the resulting adjustments, to the C/PAT; however, the SSEB cannot review the resulting cost impact.

Performing data analysis on the additional proposal information also supports documentation of the evaluation results. For a source selection, the C/PAT must, "prepare a Cost/Price Report that documents the reasonableness or realism of proposed price and cost, and the basis of any determinations or adjustments made to proposed prices and costs. Ensure that no downward adjustments are made to proposed prices and costs." (NAVSEA Source Selection Guide, pg 14). Evaluation adjustments for subjective tradeoffs or Value Adjusted Total Evaluated Price (VATEP) tradeoffs feed into the traceability of BOEs and software reporting metrics in the software summary form, resulting in monetization of requirements. Performing a VATEP tradeoff "also provides the [Source Selection Team] the ability to assign a monetary value, or "monetize," the higher rated technical attributes, thus taking some of the subjectivity out of the best value evaluation" (DoD, pg 4). VATEP tradeoffs performed for superior characteristics, include adjustments based on SLOC counting basis, reuse basis and historical effort adjustment factors, historical (and analogous) software development productivity, hours traceability by CSCI, and labor rate / staff mix. Each of these inform contract award decisions and contribute to the Government achieving best value.

OUTCOMES

Evaluation of proposed software development effort via a customized data collection form informs discussion questions and VATEP tradeoffs for each offeror, as well as provides leverage for negotiations and facilitates documentation and findings. Performing analysis on the proposal in a traceable MS Excel file produces a more credible, defensible evaluation, which minimizes risk of protest. The software data collection form's BOE references, historic performance, and proposed software development hours traceability features increase the efficiency of proposal evaluation, shortening the time required to review seemingly overwhelming amounts of data. Documenting the evaluation results in a "written comparative analysis [to reported historical efforts] of proposals" (DoD, pg 34) and in a dynamic analysis tool to support targeted price negotiations are proven outcomes of the approach.

The SLOC summary table also provides rapid identification of proposal shortcomings. Past cost research shows that the actual sizing and development effort (in hours and dollars) is usually significantly greater than the initial estimates; however, if an offeror states in a proposal the ability to exceed past performance, then the evaluator needs a basis for determining the ensuing effort. Collecting historical performance data at the CSU-level and comparing it to the software development estimate allows for realistic evaluation of the proposal. Additional shortcomings typically found in proposals include, "new SLOC […] usually [being] under-estimated [and] re-usability of existing SLOC is usually over-estimated" (Jones-Hardin). Developing a data collection form in the RFP that traces the requirements to the proposal software development effort validates the contractor's reuse claims. Assessing the proposal hours against the complexity of the requirements enhances the understanding of the scope and identifies potential risks.

The risks identified in the source selection can carry through the contract award and into a program's Integrated Baseline Review (IBR) to support risk mitigation planning and successful program execution. The decomposition and evaluation of the proposal effort at the CSU level allows analysts, contracting officers, and PMs greater understanding of the details, complexity, and resource requirements. Major weapon systems and MAIS programs are "software development projects [that] are very large and complex. Breaking a software-intensive project into parts that are more manageable helps the people involved better understand the tasks and resources needed. By understanding the complexity of an effort, managers are better able to plan and manage the risk" (DAU, pg 10). Requiring additional software development information reporting in accordance with a program-specific WBS allows for effective proposal evaluation and facilitates risk identification for contract award and establishment of a risk-adjusted baseline for contract execution.

LIMITATIONS & FUTURE ENHANCEMENTS

There are limitations to the use of data collection forms, including data security and standardized scope reporting, that provide challenges to the offeror and proposal evaluator. Software intensive systems require data security and program protection throughout the lifecycle. For example, avoid running secure SLOC through unsecure code counting tools, which limits the data reporting capability of the offeror. Additionally, do not compromise secure code to vulnerabilities in order to collect metrics. Ideally, the software development metrics are captured during execution and readily available for the estimation of future efforts. In addition, code-counting tools run as standalone executables can negate data sharing permissions.

Despite counting standards for physical and logical SLOC across many languages, there are still differences in the size and quality of a software function. For example, "the UCC counts physical and logical SLOC and other metrics according to published counting standards which are developed […] so that the logic behind the metrics being produced is clear to all participants" (UCC User Manual, pg 16). However, if offeror incentives include limiting SLOC growth on new development, the offeror can count unnecessary functions/features as reuse with the intent of later deletion from the initial measureable code basis. Accounting for productivity for the same code base limits the use of inefficient code writing as a basis for software effort estimation.

For source selection, evaluations compare proposals with the RFP requirements. Value adjustments for areas where the offeror exceeds the requirement supports the total evaluated price. Cost realism also applies when price assessments are inconsistent with the technical proposal. Future efforts for the SLOC summary table include expanding the template to accommodate additional data fields, software development approaches and new reporting standards. Defining and requiring proposals to adhere to consistent software effort counting methods will increase the evaluation capability of analysts. The data collection form will continue to support validation of BOEs with adjustment factors and no supporting data or consistent scope definition. Also, the "developer's definition of software effort may not align with cost analyst's standardized definition of effort" (Gallo-Hardin-Wilke, pg 5), requiring a normalization of software development activities to compare proposed efforts and produce a total evaluation price for each offeror. Using a consistent definition for what DCTI and non-DCTI software development activities should be included in the effort.

CONCLUSION

Collection of historical software development and maintenance data contributes to more realistic software sizing and effort estimates at critical decision milestones throughout a program's lifecycle. During RFP development, proposal evaluation, and source selection discussions, the implementation and use of SLOC Summary Tables to collect software

labor/effort data provides the necessary information for the Government to assess the validity of proposed software development efforts. Including the SLOC Summary Table in the Section J Attachment of the RFP improves the information available to evaluate proposed software development efforts (process (agile, waterfall, etc.), productivity, and BOE traceability) and supports the analysis required to perform realism adjustments. Access to historical data provides additional leverage to the Government in negotiations. Software labor data requested in this form enables the Government to assess the validity of proposed software development effort, productivity, software sizing, staffing and schedule. Analysis of proposal data supports fact finding during evaluation, specifically asking the right questions and understanding the risks. C/PAT analysts use the data collection and evaluation approach to evaluate offeror estimates for consistency with historical data and to support value-based adjustments. The data collection form assesses consistency within the price and technical proposal volumes and supports cost realism adjustments for VATEP that feed trade-off decisions.

REFERENCES

CADE. “Guidelines for the Preparation and Maintenance of CARD Tables (v14).” Section d.

Center for Systems and Software Engineering. “UCC User Manual V.2011.10.” University of Southern California, Oct. 2011.

Cummings, Cheri, et al. “Software Development Estimating Handbook: Phase One.” Naval Center for Cost Analysis, Feb. 1998.

DAU. “DAU ACQ 101 Fundamentals of System Acquisition Management: Lesson 22 - Software Acquisition: Development and Management.” DAU.

DoD. “Defense Federal Acquisition Regulation Supplement Procedures, Guidance and Information, Subpart 215.3--Source Selection.” 31 Mar. 2016.

“DPAP Best Practices, Lessons Learned, and Recommendations.” 20 Dec. 2013.

Gallo, Michael, Paul Hardin, and John Wilke. “Simple Is Not Necessarily Better- Why Software Productivity Factors Can Lead to Bad Estimates.” 38th Annual DoD Cost Analysis Symposium. 38th Annual DoD Cost Analysis Symposium, Feb. 2005.

GAO-04-393. “Stronger Management Practices Are Needed to Improve DOD’s Software-Intensive Weapon Acquisitions.” Mar. 2004.

Jones, Robert R., and Paul Hardin. “Software Code Growth: A New Approach Based on Historical Analysis of Actuals.” 2007 ISPA-SCEA Joint Conference. 2007 ISPA-SCEA Joint Conference, June 2007, New Orleans, LA.

Naval Sea Systems Command. “NAVSEA Source Selection Guide.” Aug. 2012.

“Software Resources Data Reporting: Development and Maintenance Reports and Data Dictionary.” June 2016. DI-MGMT-82035