Improving Software Estimating Relations for Army Software Sustainment Data

Authors

Cheryl Jones, US Army AFC, CCDC-Armaments Center
James Doswell, Office of the Deputy Assistant Secretary of the Army for Cost and Economics
Paul Janusz, US Army AFC, CCDC-Armaments Center
Bradford Clark, Software Metrics Inc.
Christian Smart, Galorath Federal, Inc.
Kimberly Roye, Galorath Federal, Inc.

Abstract

The Army has conducted a study over the past seven years to improve the estimation accuracy of software sustainment systems cost using data from 192 Army software systems. The collected data for these software systems was reported as multi-year data. Annual data is more representative of how data is collected at the source. Annualized data has shown improvement in Cost Estimating Relationships (CER) results. Additionally, there has been a focus on analyzing the causal relationships in the data first before modeling. Before linear regression is employed to derive a CER, a causal analysis of the data should be performed to expose cause and effect relationships. The discovery of causal relationships provides a firm justification for derived CERs. The search for causal relationships has also provided additional insight into other cause and effect relationships in the data.

This is an update to the paper entitled "New Army Software Sustainment Cost Estimating Results", ICEAA, June 2019

1 Executive Summary – Accomplishments

Through the support of DASA-CE leadership, the software sustainment initiative has continued moving the U.S. Army from a position of making educated guesses on what was being spent on software sustainment to an empirical foundation of how much it costs to sustain software.

The initiative has proposed and used a software sustainment work breakdown structure (WBS) that is now being promulgated throughout Army sustainment organizations. The WBS has created standard definitions of the different classes of software sustainment activities that Army programs are performing, and allows these activities to be quantitatively measured. It also permits software sustainment funding streams to be associated with work performed down to the software sustainment release level.

The software sustainment data collected resulted in a large, comprehensive software sustainment database which has significantly enhanced the types and kinds of sustainment data available. The information in the database includes software release level data as well as management and process data on 192 Army systems in sustainment. The information in the database supports the detailed analysis of software sustainment cost, schedule and risk drivers, and provides insight into the state of software sustainment management and processes practices.

The initiative's results have been provided to the Army software sustainment community, DASA-CE cost analysts, Life Cycle Management Command (LCMC) and Program Executive Office (PEO) cost analysts, as well as cost analysts from the other services. The results establish a robust foundation for software sustainment fact-based decisions, including:

- Allocations of Costs by WBS Elements
- Cost & Schedule Estimating Relationships
- Cost Benchmarks

The results in this paper are from a continuation of refining cost estimating relationships (CERs). There were two approaches to improving CERs. One approach was to convert multi-year software sustainment release data into fiscal year increments. The CERs from the transformed data was analyzed for accuracy.

The second approach used causal analysis to identify relationships within the data. This work, in conjunction with the Software Engineering Institute, either verified or refuted previously assumed cause-effect relationships in the data thereby simplifying (by eliminating extraneous variable) or verified (by identifying causal relationships) the selection of variables to use in CERs. As a result, the CERs presented in previous papers has been reduced and refined.

The initiative has firmly established a robust foundation for making software sustainment fact-based decisions and has created a valuable Army data repository asset for the future.

2 Key Infrastructure Components

This initiative determined that there were three key components required to establish a sound foundation for generating cost estimating relationships:

- An Army software sustainment Work Breakdown Structure (WBS)
- A repository of software sustainment execution data
- Context information that enables the accurate interpretation, comparison and contrasting of the collected cost and technical software sustainment data

These components are discussed next.

2.1 Work Breakdown Structure

A viable software sustainment Work Breakdown Structure (WBS) is the cornerstone for credible cost estimates. Without a WBS, it is difficult to compare similar systems to one another for cost estimation purposes. The WBS separates cost into different elements which clarifies what is included or excluded in the cost data.

The current Army software sustainment WBS is shown in Figure 1. The elements of the WBS are based on actual practices across all Army systems. It defines the cost elements that comprise Army software sustainment. The WBS was designed to be tailored for specific system and organizational instantiations and can be adjusted to account for variations in domain driven technical characteristics and installed cost accounting systems.

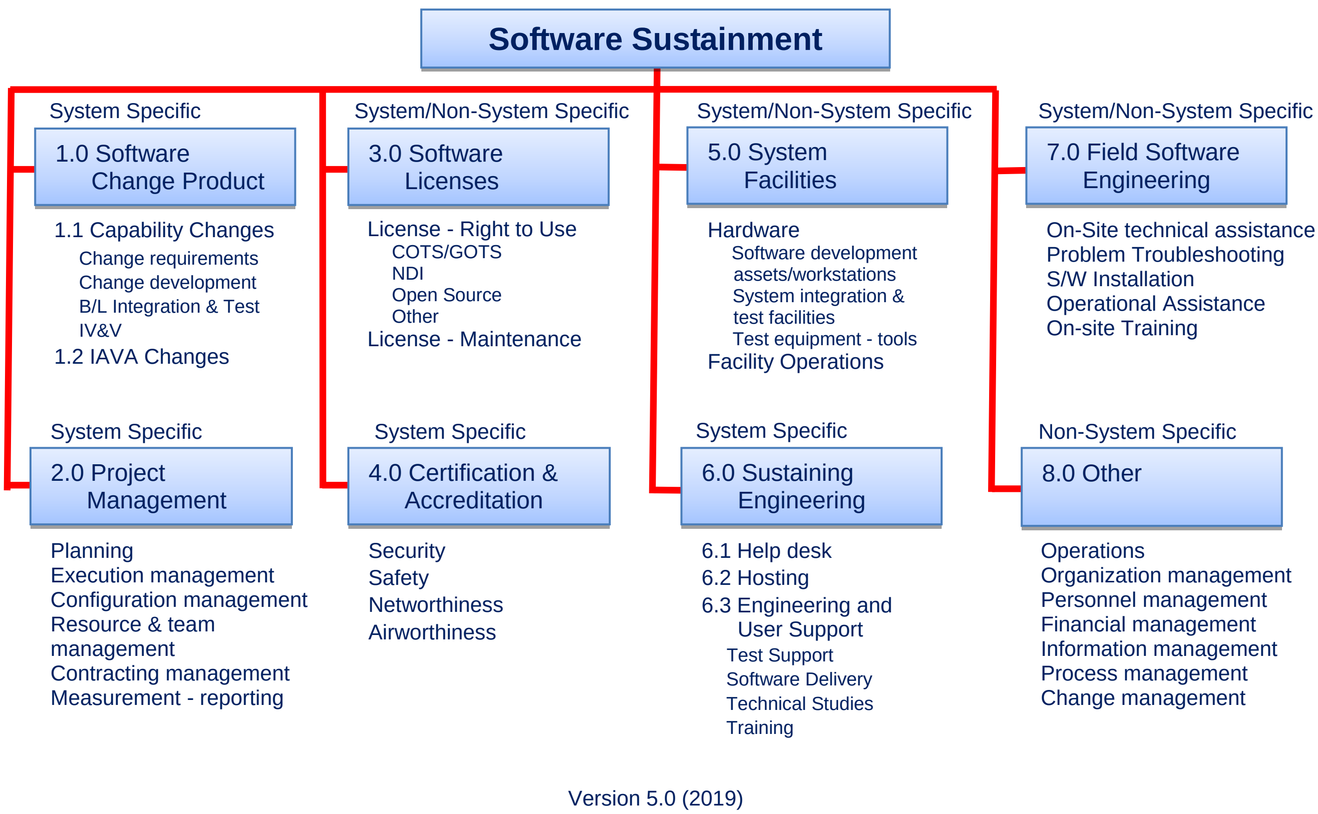


Figure 1. Army Software Sustainment Work Breakdown Structure

2.2 Software Sustainment Data Questionnaire and Repository

An Army Software Sustainment Data Questionnaire was developed to collect data and the data is stored in the data repository. There are three general categories of data are collected using the questionnaire.

1. System characterization data that describes the technical and programmatic characteristics of the operational system and the system sustainment strategy. This data includes sustainment activity, release and change profiles, domain and mission characteristics, program technical and management risks, etc. This data provides information on how the software baselines are maintained; and supports the normalization of diverse program data sets. Note that this contextual data is not directly mapped to the WBS.
2. System specific effort and cost data at the total system level, and for each of the WBS elements, including both government and contractor costs. Only costs that are attributable to a specific system are included. The software license WBS also has a detailed breakout that describes each license used by the system, including costs for each license (if purchased by the system), and other details.
3. Release level data for capability-releases and IAVA-only releases, WBS 1.0. This data includes system software sustainment cost and technical data mapped to specific output products and activities. This data includes release characterization data, release effort and cost, schedule information, output products (software size), software changes, etc.

For all categories of data, the actual execution cost and effort data is obviously preferred, but Full-Time Equivalent (FTE) or planning data was collected if actual data was unavailable. The actual/estimated data was tracked in the repository for discrimination in data analysis.

2.3 The Importance of Understanding the Context

There exists too much variability across the program products and activities for a single cost model to be correct in all instances. Every Army system is unique in some way. Therefore, the collection of system "context" data is required to enable the accurate interpretation, comparison and contrasting of the collected cost and technical software sustainment data.

Integral to the data analysis are the definitions of distinct software systems based on sustainment organization, commodities, application super domains, maintenance change types, sustainment phase, and number of software variants, platform variants, users and licenses.

Data analysis revealed that Army software sustainment activities are not "monolithic." That is to say that there is no single model or cost estimating relationship (CER) that can be defined to address the multitude of variables across Army software sustainment activities that will yield a valid cost estimate. All of the different products and activities that are being costed differently have to be taken into account, and their results integrated into a composite, context driven estimate.

2.4 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, a composite of one or more changes, is the primary SWS change product. A release can be either a formal release or an engineering release. SWS may include software enhancements, software maintenance, and/or cybersecurity updates.

Software maintenance (SWM) includes defect repair, rehosting, adaptations, updates, and reconfiguration of the software. SWM is a type of change performed on the software.

SWS may be funded by multiple funding sources. Costs include both fixed and variable costs accrued at both the system and organizational levels for both organic (government) and contractor resources.

3 Sustainment Data Characterization

A two-phase data collection activity was conducted for the initiative. During the first phase data was collected for five programs from each Army software sustainment activity, including PEOs and Life Cycle Management Centers (LCMCs), for a total of 56 systems. This first phase established an understanding of the software sustainment activities and data environment across the Army, which drove the refinement of the data collection questionnaire.

The second data collection phase collected software sustainment data across the remaining Army programs along with updates to some of the first phase systems. This included 136 additional systems and allowed analysis using a more complete data set. Both weapons and non-weapons systems comprise the dataset.

3.1 Data Overview

The amount of data collected resulted in over 411,000 repository data fields based on 192 Systems, 1,040 Releases and 3,434 software licenses, Figure 2.

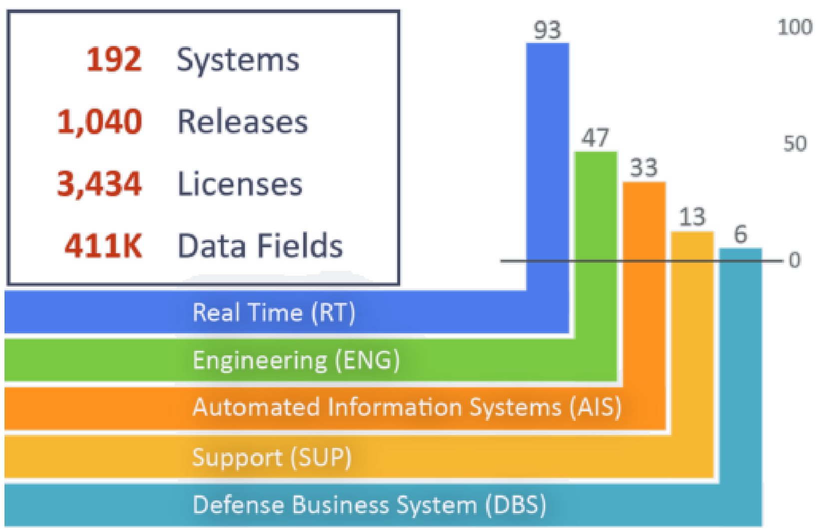


Figure 2. Data Demographics

When systems are divided into application super domains, there were 93 Real-Time Systems (RT), 47 Engineering Systems (ENG), 33 Automation Information Systems (AIS), 13 Support Systems (SUP), and 6 Defense Business Systems (DBS).

3.2 System Age

The data contains systems that vary in years in sustainment up to 40 years and are in one of two phases (see Figure 3). Post-Deployment Software Support (PDSS) characterizes systems whose hardware component are still in production; however, the software components require sustainment activities to support fielded systems. PDSS systems are managed under the Program Executive Office (PEO), and are funded with RDT&E or Production funding. Post-Production Software Support (PPSS) systems are operationally sustained via a Life Cycle Management Center (LCMC) and are funded with O&M funds.

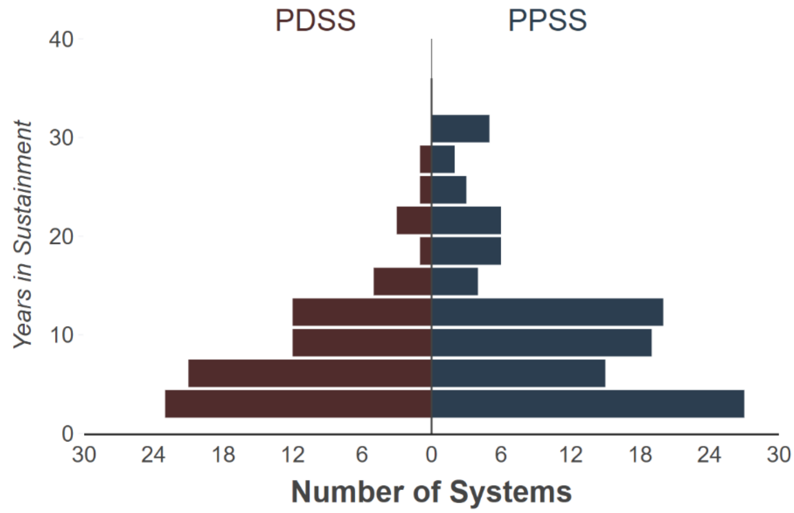


Figure 3. Age of Systems

3.3 Software Release Characterization

Figure 4 shows that releases are divided into capability releases (718) and IAVA-only releases (322). Capability releases modify software while IAVA releases scan the software for vulnerabilities. Of the capability releases, there were 318 primarily maintenance releases, 170 primarily enhancement releases, 195 hybrid releases that were a mix of maintenance and capability enhancements, and 16 releases classified as “Other.” IAVA-only releases were all classified as Cyber releases, and a few capability releases contained only IAVA updates, for a total of 341. Different types of releases were each analyzed for CERs.

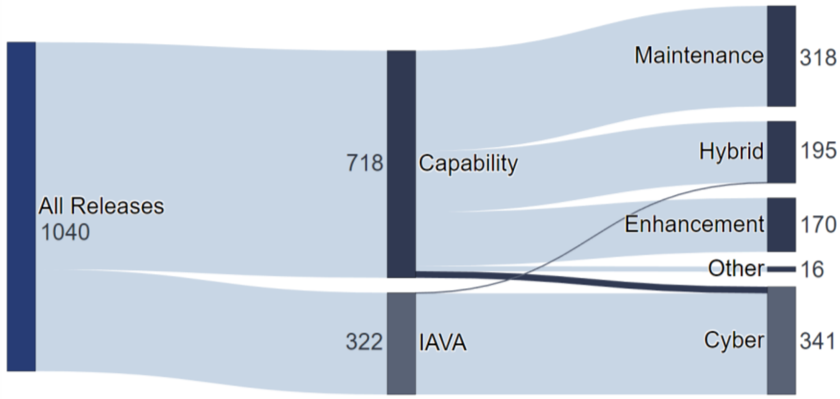


Figure 4. Releases by Change Type

3.4 Release Size Measures

Systems were asked to report the size measures that were used within their program. Figure 5 shows the size measures reported. Software Changes (SC) was the most common size measure with data provided for 571 releases. SCs are enhancements or maintenance changes to the software. The second most common size measure reported was a count of the number of IAVAs, with data from 420 releases. Some systems reported the number of requirements implemented in the release, for 224 releases. Source lines of code (SLOC) counts were reported for 152 releases. A subset of those releases broke down the code counts for new, modified, reused, and autogenerated code. Other size measures included Function Points and RICE-FW objects reported in 39 releases. Story-Points were reported for 11 Agile releases.

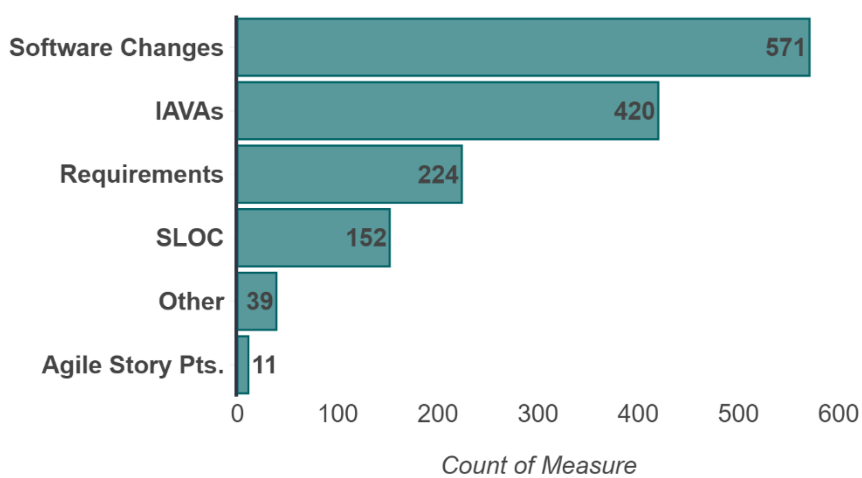


Figure 5. Releases by Size Measure

4 Cost Estimation Relationships for Capability Releases

The analysis in this paper examines the effort to maintain software in the WBS Element 1.0, Software Change Product. The analysis consisted of the derivation CERs focused on capability release data. A capability release changes the software to improve its capability or repair a problem. Software changes were treated as the independent variable to estimate the dependent variable, effort hours. These derived CERs are utilized later in the acquisition lifecycle (post Milestone C) when expected release size, anticipated software change counts, is known.

The data for this analysis differs from previous analyses presented in earlier papers by using *annualized* data. Figure 6 shows the schedule data for WBS 1.0 based on 614 observations and measured in months. Some of the durations were less than three months indicating an emergency or patch release. Other release durations spanned years, indicating major or multiple rolled-up releases.

Because software maintenance data is typically tracked annually, the effort was made to transform multi-year data into equivalent annual data. This consisted of scanning capability releases for multi-year data and converting all size, effort and cost measures into fiscal years. The annual data is also more consistent with the other WBS cost data which was reported by FY.

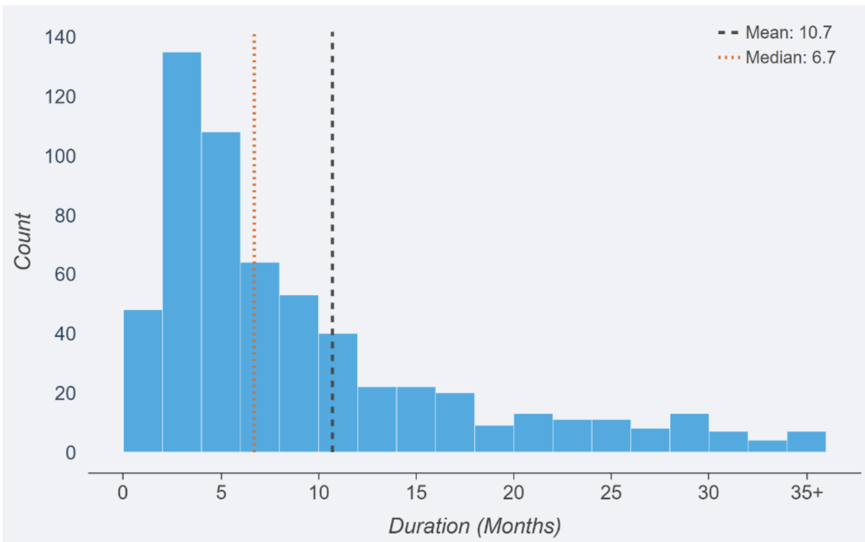


Figure 6. Release Duration Distribution

Future data collection efforts are focusing on collecting annual data. Annualizing the data in the current repository will enable combining the old and new data.

4.1 Ground Rules/Assumptions

The following ground rules or assumptions apply to each CER:

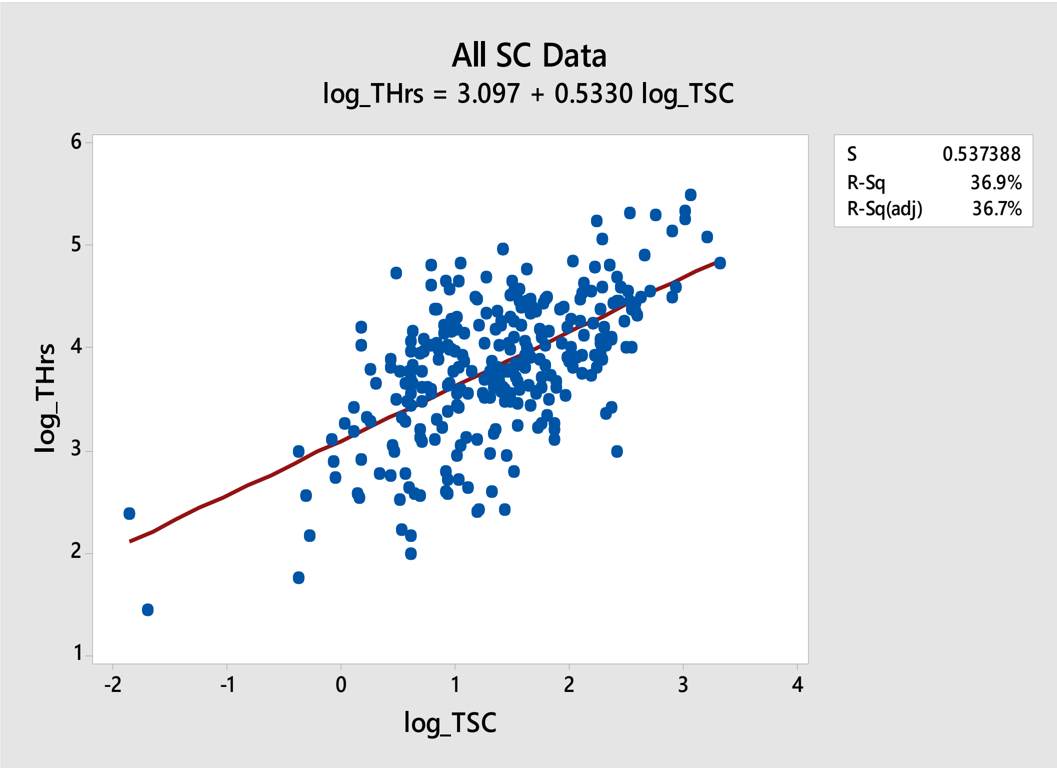
- The CER applies to WBS 1.0, Software Change Product, only for capability releases.
- Defense Business Systems were not included in this analysis.
- Data that was not within 50% of the reported annual labor hours per person-year and annual burden labor rate were labeled outliers and not used in this analysis.
- Software size is expressed as total Software Changes (TSC), Requirements, or Equivalent Source Lines of Code (ESLOC).
- Due to the non-normal distribution of the raw data, both dependent and independent variables were transformed using $\log_{10}$. Zero values were represented as 0.1.
- All categorical variables (super domain, commodities ACAT levels, etc.) were represented as dummy variable (0,1).
- Ordinary Least Squares (OLS) regression was used to derive the CERs. The Minitab

statistics tool and Python statistics libraries were used for OLS analysis. Non-linear search optimization methods will eventually be used to refine the CERs presented here but are not included in this paper.

- Adjusted R^2, the Coefficient of Determination, was used to compare model fits to the data. Adjusted R^2 was used because models with different independent variable are compared. R^2 ranges between 0.0 and 1.0 where 0.0 means no fit and 1.0 means a perfect fit.

4.2 Methodology

After initially reviewing the annualized data for CERs, the data was shown to have a high amount of variability. Figure 7 shows a log-scale scatter plot of 306 observations for all



Variable	Coef.	Std Err	T-Value	P-Value
Constant	3.0967	0.063	49.45	0.000
Log(SC)	0.5330	0.0399	13.34	0.000

Figure 7. All CER Data Scatter Plot

annualized capability release data that had the independent variable, total software changes (TSC), and the dependent variable, total release hours (THrs). The plot shows a regression model of THrs = 1,249 * $TSC^{0.53}$ with a large amount of variation and an R^2 of 0.37.

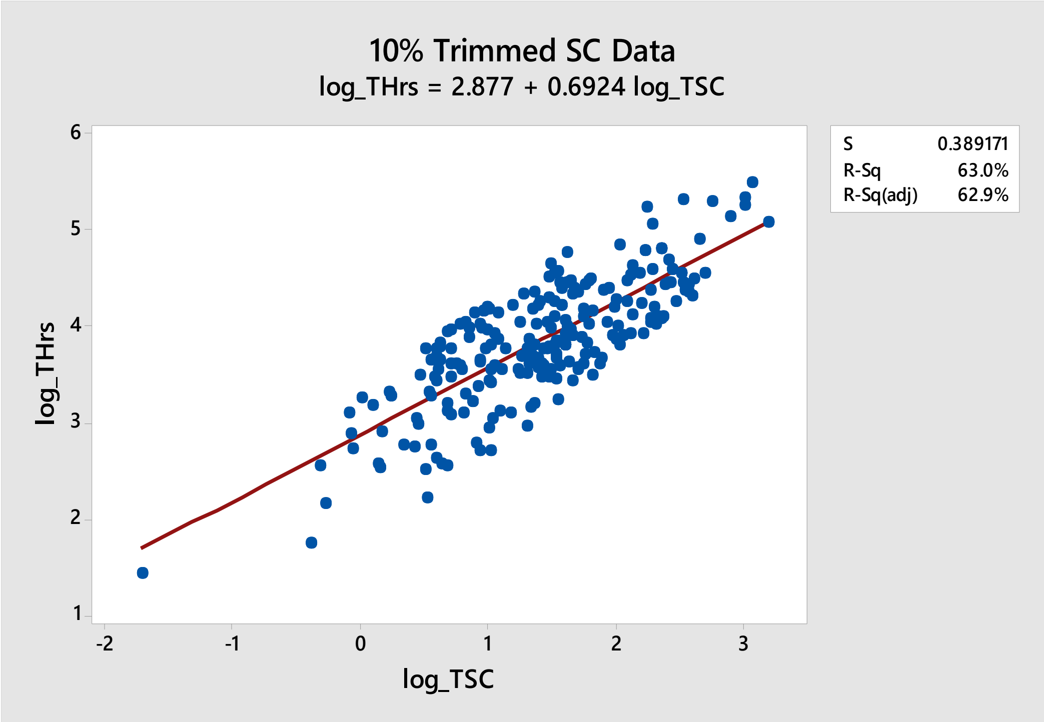
Due to the poor results, the data was trimmed and segmented into groups to tighten variability using two strategies:

1. The upper and lower 10% of the data was trimmed from the dataset. Trimming was based on

unit cost (total release hours / #software changes). While the data had been scrubbed for hours and cost outliers, some of the unit costs were extremely low and some were extremely high.

2. Meta-data was used to derive multiple categories, each of which was analyzed for CERs using the trimmed data.

The first strategy trimmed the upper and lower10% of the data based on unit cost (Hours/SC). Figure 8 shows the scatter plot and trendline on 244 observations. The regression model is THrs = 754 * $TSC^{0.69}$ with an R^2 of 0.63.



Variable	Coef.	Std Err	T-Value	P-Value
Constant	2.8774	0.0538	53.45	0.000
Log(SC)	0.6924	0.0341	20.32	0.000

Figure 8. Trimmed Data CER

The second strategy used categorical data to create similar data groups. These groups are thought to share the same product and environmental attributes thus reducing variation. The groups were:

- Super Domains
  - Real-Time (RT)
  - Engineering (ENG)
  - Automated Information Systems (AIS)
  - Support (SUP)
- Acquisition Category (ACAT) Level
- Commodities (13)

4.3 CER Results

Ordinary Least Squares (OLS) regression was used to find CERs in the capability release data. The dependent variable was total release hours (THrs). The most common independent variable was the total software changes (TSC). However, total system requirements (TReqts) and total requirements implemented (TReqts_Imp) were also used.

Table 1 shows the CER models, conditions for the OLS regression, the number of observations (Obs), and the R^2. For comparison, previous model fit statistics are shown. These are models that used the data as reported, i.e., non-annualized data. Models with an R^2 of 0.75 or above are highlighted in green. Models highlighted in red had P-Values greater than 0.1..

The strongest CERs in Table 1 have software changes (SC) as the common independent variable along with total system requirements (Req_T).

Table 1. Cost Estimating Relationships with Annualized Data-1

Annualized Data Model		Conditions	Obs	Adj R^2	Prev Obs	Prev R^2
THrs = 1,249 * $(TSC)^{0.53}$		All data	306	0.37	329	0.36
THrs = 754 * $(TSC)^{0.69}$		10% trimmed data	244	0.63	263	0.57
AIS	THrs = 459 * $(TSC)^{0.69}$	10% trimmed & Super Domains (Categorical)	244	0.63*	263	0.62
ENG	THrs = 703 * $(TSC)^{0.69}$					
RT	THrs = 869 * $(TSC)^{0.69}$					
SUP	THrs = 1,208 * $(TSC)^{0.69}$					
Aviation	THrs = 656 * $TSC^{0.71}$	10% trimmed & Commodities (Categorical)	244	0.70*	263	0.68*
Business	THrs = 348 * $TSC^{0.71}$					
C5ISR	THrs = 704 * $TSC^{0.71}$					
ChemBio	THrs = 174 * $TSC^{0.71}$					
Comms	THrs = 650 * $TSC^{0.71}$					
Fire	THrs = 724 * $TSC^{0.71}$					
Intel	THrs = 781 * $TSC^{0.71}$					
Missiles	THrs = 1,460 * $TSC^{0.71}$					
MissionCmd	THrs = 1,600 * $TSC^{0.71}$					
Network	THrs = 805 * $TSC^{0.71}$					
SATCOM	THrs = 1,484 * $TSC^{0.71}$					
Simulation	THrs = 368 * $TSC^{0.71}$					
Vehicles	THrs = 411 * $TSC^{0.71}$					
THrs = 787 * $SC\_Total^{0.90}$ / $Req\_T^{0.17}$		10% trimmed	32	**0.81**	32	**0.84**
THrs = 808 * $SC\_Total^{0.81}$ / $Req\_Imp^{0.11}$		10% trimmed	104	0.70	65	0.74

***** *High P-Values for one or more coefficients*

The decision was made to conduct further analysis within each categorical group. While dummy variables are a viable approach to including categorical data in a regression model, the only variation in the CER is the intercept, the constant. The exponent is constant across all members in the category.

The data was segmented into separate groups and subgroups to eliminate the need for dummy variables. This allows each member of a group to have its own constant (intercept) and exponent (slope). The disadvantage of this approach is a smaller number of data observations.

Table 2 shows the results of the data subgrouping. The categories chosen for subgrouping were:

- Acquisition Category Levels (ACAT)
  - ACAT I (59 observations)
  - ACAT II (34 observations)
  - ACAT III+ (151 observations; includes ACAT IV and non-Programs of Record)
- Super Domains (RT, ENG, AIS, SUP)

The data was segmented into two tiers: ACAT groups followed by super domain within each ACAT group. All data groups used the 10% upper and lower trimmed data.

Table 2. Cost Estimating Relationships with Annualized Data-2

Annualized Data Model		Conditions	Obs	Adj R^2
ACAT I ACAT II ACAT III+	THrs = 769 * SC_Total$^{0.68}$ THrs = 1,124 * SC_Total$^{0.68}$ THrs = 713 * SC_Total$^{0.68}$	10% trimmed & ACAT Levels (Categorical)	244	0.64*
THrs = 483 * SC_Total$^{0.80}$		10% trimmed, ACAT I	59	**0.79**
AIS ENG RT	THrs = 573 * SC_Total$^{0.79}$ THrs = 409 * SC_Total$^{0.79}$ THrs = 577 * SC_Total$^{0.79}$	10% trimmed, ACAT I & Super Domains	59	0.79*
THrs = 359 * SC_Total$^{0.90}$		10% trimmed, ACAT I & RT	35	**0.85**
THrs = 1,563 * SC_Total$^{0.40}$		10% trimmed, ACAT I & ENG	19	0.63
THrs = 2,805 * SC_Total$^{0.43}$		10% trimmed, ACAT I & AIS	5	**0.87**
THrs = 1,265 * SC_Total$^{0.65}$		10% trimmed, ACAT II	34	0.65
AIS ENG RT	THrs = 3,428 * SC_Total$^{0.62}$ THrs = 1,648 * SC_Total$^{0.62}$ THrs = 1,125 * SC_Total$^{0.62}$	10% trimmed, ACAT II & Super Domains	34	0.68*
THrs = 2,742 * SC_Total$^{0.0.34}$		10% trimmed, ACAT II & RT	18	0.05
THrs = 1,545 * SC_Total$^{0.64}$		10% trimmed, ACAT II & ENG	10	**0.84**
THrs = 794 * SC_Total$^{0.91}$		10% trimmed, ACAT II & AIS	4	0.36
THrs = 851 * SC_Total$^{0.64}$		10% trimmed, ACAT III+	151	0.55
AIS ENG RT SUP	THrs = 479 * SC_Total$^{0.65}$ THrs = 951 * SC_Total$^{0.65}$ THrs = 851 * SC_Total$^{0.65}$ THrs = 706 * SC_Total$^{0.65}$	10% trimmed, ACAT III+ & Super Domains	151	0.55
THrs = 1,076 * SC_Total$^{0.58}$		10% trimmed, ACAT III+ & RT	60	0.49
THrs = 583 * SC_Total$^{0.87}$		10% trimmed, ACAT III+ & ENG	49	0.58

Annualized Data Model	Conditions	Obs	Adj R^2
THrs = 386 * $SC\_Total^{0.73}$	10% trimmed, ACAT III+ & AIS	28	0.59
THrs = 2,673 * $SC\_Total^{0.34}$	10% trimmed, ACAT III+ & SUP	17	0.33

***** *High P-Values for one or more coefficients*

The approach of using multiple-tiered segmentation groups (two tiers in this analysis) provides better insight into the data. Each CER was allowed to have its own constant, exponent and model fit. It can be observed which members of the lower tier impact the model fit for the tier above.

For instance, it can be seen in Table 2 that the ACAT I tier (the topmost tier) has an Adj-R^2 of 0.79 using the super domains as categorical variables. When the data is segmented into sub-tiers, it can be seen that the Adj-R^2 for ACAT I-RT and ACAT I-AIS super domains are strong but the ENG super domain is weak. This provides the insight to go back into the data and investigate if there are common factors in the context data that explain the ENG CER's poor performance.

4.4 Conclusions

The CER analysis of the annualized data is not much different in performance than the multi-year release data as collected. Using CERs based on annualized data makes cost estimation easier going forward because funding requests are done by fiscal year. Since future data is being collected annually, using the converted annualized data will make it possible to combine the new and old datasets.

Segmenting categorical data into tiers generally shows more accuracy for members in the lower tier. It also highlights poor performing members in the lower tier that need further investigation.

5 Causal Analysis

The Army DASA-CE software sustainment initiative collaborated with the Software Engineering Institute (SEI), Pittsburgh, PA, to investigate cause and effect relationships in collected maintenance data. There are a large number of factors in the software maintenance data and it is challenging to identify which ones are useful for grouping data or valid for CER inputs. These factors are:

- Unit Cost: Hours per Software Change
- Commodities (13)
- Duration Type (Release Structure: Cyclic, Sequential, Sequential with Overlap, and Concurrent releases)
- Maintenance Phase
- Number of Inter-Services Partners
- Acquisition Category (ACAT) Level
- Super Domains (RT, ENG, SUP, AIS)
- Sustainment Phase (MS-C LRP, MS-C FRP, O&S)/Time in Phase
- Number of Appropriations
- Number of Software variants
- Software Baseline size in Source Lines of Code
- Number of Hardware Platforms
- Number of Hardware Platform variants

- Number of Users
- Number of Licenses

It is well known that *correlations* among factors does not necessarily mean *causation*. Because of this, regression models are often the wrong tool to use for causal search, i.e., identifying which factors affect the outcome. These models may use predictor variables that are influenced by variables outside the model. The model may have a good fit to the data but will not be accurate making estimates.

Causal analysis searches for cause and effect across all factors and shows valid and confounding relationships. Figure 9 shows an example of a causal graph. The variable of interest is Y and the candidate predictor variables are X_1, X_2, X_3, X_4, X_5, and X_6. The edges in the graph indicate a causal relationship. The arrows indicate the direction of the cause and effect relationship.

It is observed in Figure 9 that there are four possible predictor variables used individually or together to predict Y. However, there are confounding relationships that exist that need to be addressed. Since X_2 influences both X_1 and Y and X_1 influences Y, a change in X_2 will produce a change in both X_1 and Y. Both X_1 or X_2 should be tested and the lowest performing predictor should be dropped from consideration. X_6 influences Y as does X_5 but X_6 also influences X_5. A change in X_6 will produce a change in both X_5 and Y. Again, test X_5 and X_6 and drop the lowest

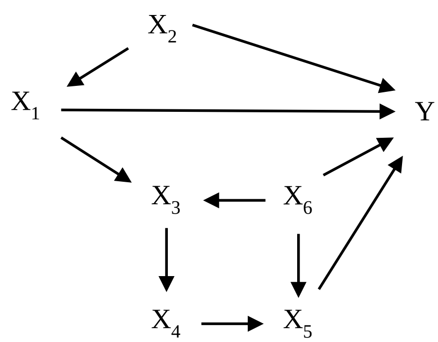


Figure 9. Causal Graph Example

performing predictor. From these simple observations, the number of predictor variables has been reduced from four to two with the assurance that confounding relationships have been eliminated.

As demonstrated with the example, it is evident that causal analysis should proceed regression analysis. This would apply not only to software CERs but all CERs.

Working with SEI experts on causal analysis, work was done to find validated cause and effect relationships using both discrete (categorical & ordinal) and continuous data. Their analysis showed that many of the factors discuss earlier showed no relationship with the variable of interest, total release hours. This saved a lot of CER analysis time. An example of their results for the Real-Time super domain are presented in this section.

5.1 Methodology

As a result of causal analysis, the data was segmented into two tiers: the first tier was data segmented by super domain and the second tier segmented the data by ACAT level within each super domain. Unit cost expressed as total release hours per software change was used as the variable of interest in the analysis. This derived measure ensured that software changes and hours are always considered.

Table 3 shows a summary of the influencing factors within each super domain – ACAT level group. Different factors appear in different groups meaning different predictors are used in CERs for that group. Unexpectedly, the most common factor across all groups is the count of Inter-Service Partners, i.e., the number other DoD services that participate in a system's sustainment.

This paper only shows the causal relationship graphs and CERs for the Real-Time super domain and the three ACAT level releases. Graphs and CERs were also derived for the Engineering and Automated Information Systems super domains and ACAT levels. The intent is to show the utility of the concept of using causal analysis to drive the formulation of CERs.

Table 3. Causal Analysis Influential Factors

Super Domain	ACAT I	ACAT II	ACA III+
RT	Phase Inter-Service Partner Count HW Platforms	Service Partner Count	Hardware Variants Maintenance Phase Software Baseline
ENG	(None)*	Number of Appropriations Hardware Variants Maintenance Phase Inter-Service Partner Count Software Baseline	Duration Type Hardware Variants Maintenance Phase
AIS	Inter-Service Partner Count*	(No info)	Duration Type Inter-Service Partner Count

** There were very few cases in two of three ACAT I datasets making casual effects harder to analyze.*

5.2 CER Results

Using causal analysis graphs, Ordinary Least Squares (OLS) regression was used to find CERs using different independent variables with total software changes (TSC) being the common independent variable. The dependent variable is total release hours (THrs).

The variable of interest in each causal graph is the unit cost, hours per software change (Hr_SC). The directed edges in the graph indicate the direction of the causal relationships. Other relationships displayed in the graph are of interest for different questions such as which data does not contribute to CER formulation and can be eliminated from data collection.

The graph in Figure 10 shows causal relationships for the Real-Time super domain and ACAT I releases. The factors of interest are Software baseline line (measured in source lines of code), Inter-service partner counts, and Maintenance Phase are possible predictors of Hr_SC.

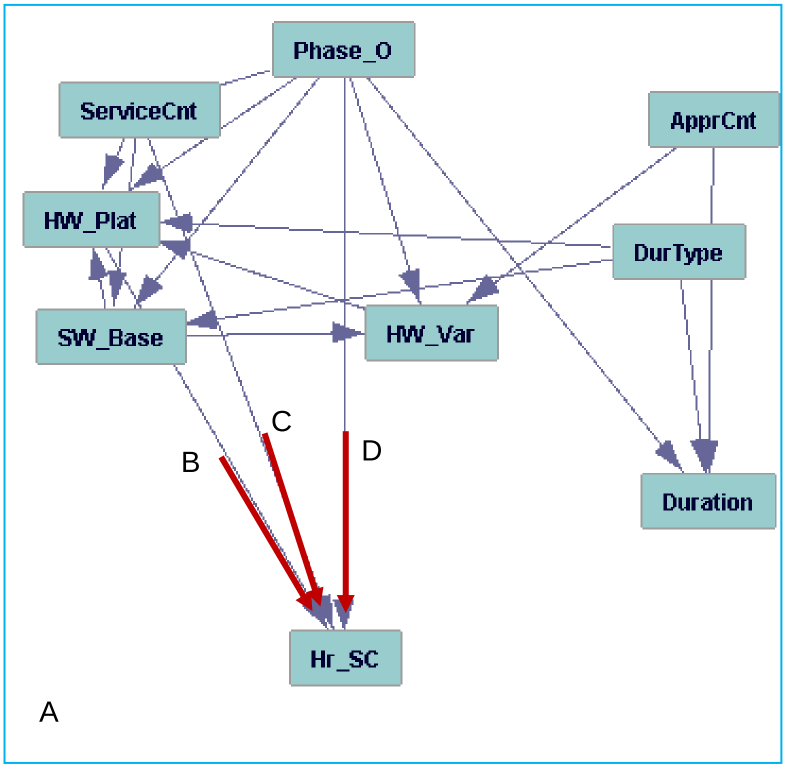


Figure 10. Real-Time ACAT I Causal Graph

Table 4 shows the CER models, conditions for the OLS regression, the number of observations (Obs), and the Adj-R^2. Based on the graph in Figure 10, CERs with different predictors were examined. These are models use 10% trimmed annualized data. Models with an R^2 of 0.75 or above are highlighted in green. CER predictors highlighted in red had a P-value greater than 0.1.

The first row of Table 4 provides a reference to compare to other CERs in the table.

Table 4. Real-Time ACAT I CER Results

Data Model		Conditions	Obs	Adj R^2
ACAT I ACAT II ACAT III+	$THrs = 544 * (TSC)^{0.74}$ $THrs = 544 * (TSC)^{0.74}$ $THrs = 544 * (TSC)^{0.74}$	10% | RT | ACAT Levels	68	0.71*
A: $THrs = 577 * (TSC)^{0.81}$		10% | RT | ACAT I	23	0.74
B: $THrs = 164 * (TSC)^{1.0} * ServCnt^{1.10}$		10% | RT | ACAT I | ServCnt	23	**0.84**
C: $THrs = 91 * (TSC)^{0.94} * HW\_Plat^{0.18}$		10% | RT | ACAT I | HW_Plat	22	**0.80**
MS C D: MS C - FRP O&S	$THrs = 260 * (TSC)^{0.74}$ $THrs = 394 * (TSC)^{0.74}$	10% | RT | ACAT I | Phases (Ordinal)	23	0.71

Data Model	Conditions	Obs	Adj R^2
$THrs = 787 * (TSC)^{0.74}$			
$THrs = 161 * (TSC)^{1.00} * ServCnt^{1.05} * HW\_Plat^{0.0011}$	10% | RT | ServiceCnt | HW_Plat	22	0.84*

***** *High P-Values for one or more coefficients*

The causal graph in Figure 11 shows the relationships for the Real-Time super domain and ACAT II releases. There is only one relationship for HR_SC, Inter-service partner counts.

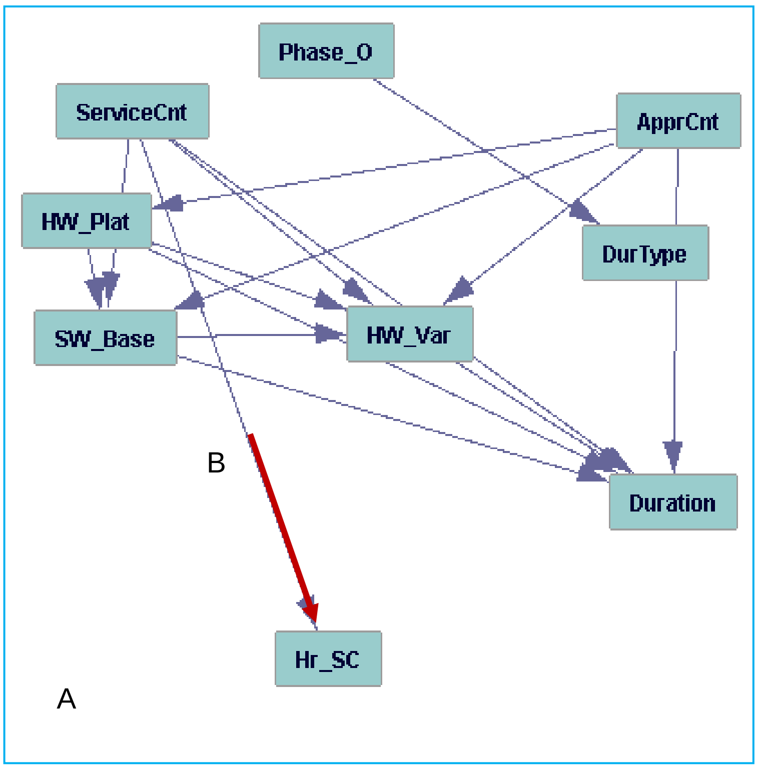


Figure 11. Real-Time ACAT II Causal Graph

Table 5 shows the CERs, conditions for the OLS regression, the number of observations (Obs), and the Adj-R^2. Models with an R^2 of 0.75 or above are highlighted in green. CER predictors highlighted in red had a P-value greater than 0.1. The first row of Table 5 provides a reference to compare to other CERs in the table.

Table 5.. Real-Time ACAT II CER Results

Data Model		Conditions	Obs	Adj R^2
ACAT I ACAT II ACAT III+	$THrs = 544 * (TSC)^{0.74}$ $THrs = 544 * (TSC)^{0.74}$ $THrs = 544 * (TSC)^{0.74}$	10% | RT | ACAT Levels	68	0.71*
A: $THrs = 308 * (TSC)^{0.95}$		10% | RT | ACAT II	23	**0.75**
B: $THrs = 287 * (TSC)^{0.94} * ServCnt^{1.86}$		10% | RT | ACAT II | ServiceCnt	23	**0.75**

***** *High P-Values for one or more coefficients*

The causal graph in Figure 12 shows relationships for the Real-Time super domain and ACAT III+. The ACAT category is a collection of lower-level programs and non-programs of record. There are three factor that appear to influence the unit cost, hours per software change: Software Baseline (measured in source lines of code), Maintenance Phase, and the number of Hardware Variants.

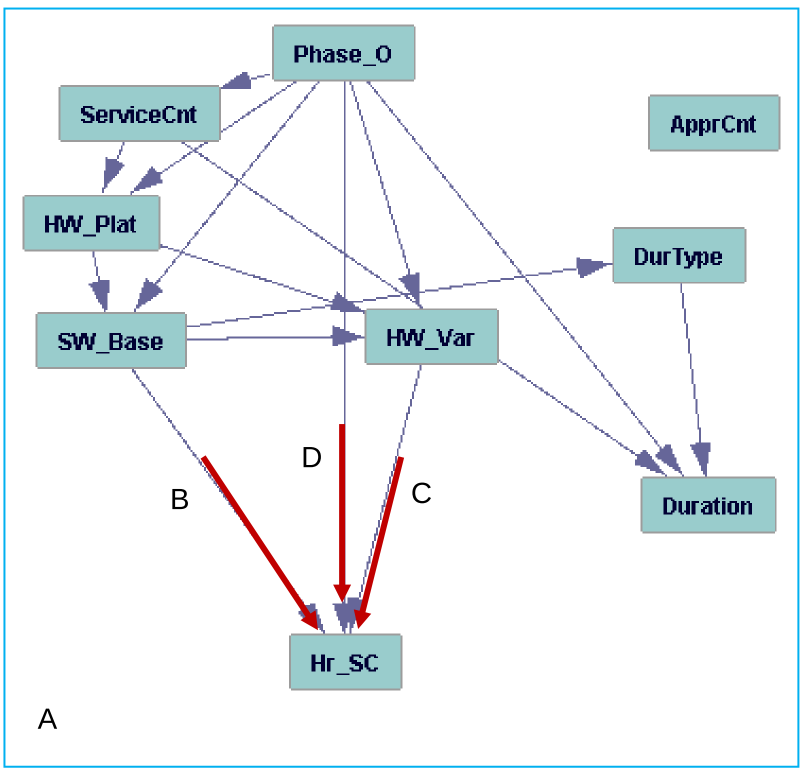


Figure 12. Real-Time ACAT III+ Causal Graph

Table 6 shows the results of CER analysis. The first row is provided for comparison to the models in the table. None of the causal-directed models outperforms the model in the first row which is interesting. Models that use Maintenance Phase as a predictor variable appear to have low significance for the MS-C LRP model. This finding provides insight into what group of data needs further investigation into the context data for those releases. CER predictors highlighted in red had a P-value greater than 0.1.

Table 6. Real-Time and ACAT III+ CER Results

Data Model		Conditions	Obs	Adj R^2
ACAT I ACAT II ACAT III+	THrs = 544 * $(TSC)^{0.74}$ THrs = 544 * $(TSC)^{0.74}$ THrs = 544 * $(TSC)^{0.74}$	10% | RT | ACAT Levels	68	0.71*
A: THrs = 467 * $(TSC)^{0.99}$		10% | RT | ACAT III+	22	0.61
B: THrs = 280 * $(TSC)^{1.02}$ * $SW\_Base^{0.46}$		10% | RT | ACAT III+ | SW_Base	22	0.65
C: THrs = 252 * $(TSC)^{1.01}$ * $HW\_Var^{0.74}$		10% | RT | ACAT III+ | HW_Var	22	0.64*
MS C - FRP D: MS C - LRP O&S	THrs = 272 * $(TSC)^{0.92}$ THrs = 465 * $(TSC)^{0.92}$ THrs = 776 * $(TSC)^{0.92}$	10% | RT | ACAT III+ | Phases (Ordinal)	22	0.67*
MS C - FRP MS C - LRP O&S	THrs = 138 * $(TSC)^{1.07}$ * $SW\_Base^{0.51}$ THrs = 70 * $(TSC)^{1.07}$ * $SW\_Base^{0.51}$ THrs = 328 * $(TSC)^{1.07}$ * $SW\_Base^{0.51}$	10% | RT | ACAT III+ | SW_Base | Phases (Ordinal)	22	0.67*

***** *High P-Values for one or more coefficients*

Using the Real-Time super domain and ACAT levels, causal graphs were used to direct CER analysis. This approach made the search for CERs more efficient and often produced better results than seen with the annualized data CERs. This approach also highlighted data groups that need further investigation into their poor CER performance.

5.3 Conclusions

Causal relationship analysis provided insight into which independent variables should be examined for predicting total release hours. This saves a lot of random analysis time. The discovered relationships also suggested other relationships that could answer different information needs such as which data does not contribute to CER formulation and can be eliminated from data collection

Segmenting data as suggested by causal analysis generally shows more CER accuracy in each segment versus trying to find a one-size-fits-all CER. It also highlights poor performing members in the segment that need further investigation.

6 Conclusions and Next Steps

The Army DASA-CE software sustainment initiative has worked over the past seven years of moving the U.S. Army from a position of making educated guesses on what was being spent on software sustainment and what it was being used for, to being able to provide deep insights from an Army-wide perspective into how software sustainment is being performed, how much it costs, and what software is being delivered to the warfighter.

There now exists an Army software sustainment WBS that creates standard definitions of the different types of software sustainment. There is also an Army Software Sustainment Data repository which contains system context-information, annual cost and effort data, software release data, and data on software licenses. The information in the repository supports the detailed analysis of software sustainment cost, schedule and risk drivers, and provides insight into the state of software sustainment management and processes practices.

The annualized data analysis presented in this paper shows that it produces CERs generally as accurate as multi-year release data. Future analysis will be based on annualized data. Newly collected annual data will be compatible with the converted multi-year release data.

The causal relationship analysis approach done in conjunction with the Software Engineering Institute has shown to be a key step in the formulation of CERs. Without this analysis, much time would have been wasted in searching for viable CERs. Additionally, the causal analysis confirmed cause and effect between independent and dependent variables avoiding the trap of spurious correlations being used for CERs.

The next steps are to continue collecting software maintenance release data annually. The causal relationships will be updated using both new and old data and the CERs will be revised based on the discovered relationships.

Acronyms

ACAT	Acquisition Category
Adj R^2	Adjusted-R^2
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	Defense Cost and Resource Center
CER	Cost Estimating Relationship
Chem/Bio	Chemical/Biological
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
FSE	Field Software Engineering
FTE	Full Time Equivalent
FY	Fiscal Year
HW_Plat	Hardware platforms
HW_Var	Hardware variants
IAVA	Information Assurance Vulnerability Alert
IAVM	Information Assurance Vulnerability Management
ICEAA	International Cost Estimating and Analysis Association
LCMC	Life Cycle Management Centers
LOE	Level of Effort
Maint%	Percent of the release that is Maintenance changes
MS C – FRP	Milestone C Full Rate Production
MS C – LRP	Milestone C Low Rate Production
O&S	Operations and Sustainment
Obs	The number of observations
ODC	Other than Direct Costs
OLS	Ordinary Least Squares statistical regression
OMA	Operations and Maintenance Army funding
OPA	Other Program Army funding

OSD	Office of the Secretary of Defense
OSMIS	Operation/Sustainment Management Information System
PDSS	Post-Deployment Software Support
PEO	Program Executive Office
PM	Person-Months of effort
POM	Program Objective Memorandum
POR	Program of Record
PPSS	Post-Production Software Support
PTR	Problem Trouble Report
RDT&E	Research, Development, Testing, and Evaluation
RMF	Risk Management Framework
RT	Real-Time super domain
SC	Software Changes
SEC	Software Engineering Center
SEI	Software Engineering Institute
ServCnt	Inter-Service Partners count
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
SWM	Software Maintenance
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

References

[1] Ferguson, J. "Crouching Dragon, Hidden Software: Software in DoD weapon systems," *IEEE Software*, July/August 2001, Vol. 18 No.4:105-107.

[2] Crawford, Bruce T., "Military Software Is 'Next Great Frontier'," National Defense, 21 April 2017.

[3] Jones et al., "Investigation into the Ratio of Operating and Support Costs to Life-Cycle Costs for DoD Weapon Systems," *Defense ARJ*, January 2014, Vol. 21 No. 1: 442–464.

[4] Defense Science Board. Design and Acquisition of Software for Defense Systems, Department of Defense, February 2018.

[5] McGarry, John et al. "Software Maintenance, Sustaining Engineering, Software Maintenance, Sustaining Engineering, and Operational Support: Key Factors That Impact Program and System Performance," Systems and Software Technology Conference, 25 April 2012.

[6] Judy, James et al. "Software Maintenance, Sustaining Engineering, Software Maintenance, Sustaining Engineering, and Operational Support: Estimating Software Maintenance Costs for U S Army Weapons Estimating Software Maintenance Costs for U.S. Army Weapons Systems," PSM User' Group Meetings and Workshops, 28 July 2012.

[7] Judy, James. "Software Maintenance and Sustaining Engineering Cost Estimation," PSM User' Group Meetings and Workshops, 28 July 2012.

[8] McGarry, John and Robert Charette, "Workshop: Life Cycle Maintenance Cost Estimation Model," PSM User' Group Meetings and Workshops, 31 July 2012.

[9] Judy, James et al., "Estimating Software Maintenance Costs for U.S. Army Systems," COCOMO Forum 2013, 16 October 2013.

[10] Judy, James et al., "Software Maintenance Cost Estimating Relationships – One Size Does Not Fit All," 28th International Forum on COCOMO and Systems/Software Cost Modeling, 23 October 2013.

[11] Judy, James et al., "How Much Does Software Maintenance Cost?," ICEAA Workshop 2015, 9 June 2015.

[12] McGarry, John and Robert Charette, "Army Software Maintenance - Addressing the Critical Issues," 17th Annual PSM Users' Group Workshop, 22 February 2016.

[13] Jones, Cheryl et al., "Workshop: Transitioning Defense Software Maintenance/ Sustainment to DevOps," 17th Annual PSM Users' Group Workshop, 23 February 2016.

[14] Government Accountability Office. *GAO Cost Estimating and Assessment Guide, Best Practices for Developing and Managing Capital Program Costs*, GAO-09-3SP, March 2009.

[15] Judy, James et al. "Estimating System Software Sustainment Costs: Generating Critical Decision Information to Inform the Army Software Enterprise", ICEAA, 5 June 2017

[16] Boehm et. Al., Software Cost Estimation with COCOMO II, Prentice Hall PTR, Upper

Saddle River, NJ, 2000.