Dynamic Software Effort Estimation – How SWEET It Is!

ICEAA Workshop
Pittsburgh, PA

May 17th, 2022

Technomics
Better Decisions Faster

Agenda

- Problem Statement
- Analytical Approach
- Prototype Overview
- Example Analysis
- Conclusion
- Q&A

Research Team Introduction

Will Gellatly

Senior Analyst

William Gellatly has provided IT estimation support and analysis for multiple government agencies including DHS, USPS and the U.S. Census Bureau. William holds a M.S. in Information Technology and Management from the University of North Carolina at Greensboro.

Lindsey Jones

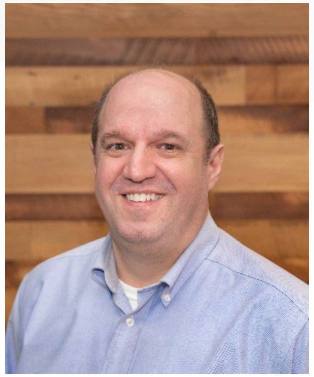
Lead Analyst

Lindsey has over 6 years of experience supporting Navy programs with Cost and Software Data Report (CSDR) planning, validation, and policy; Earned Value Management (EVM) analysis; and other program office support. Lindsey is a Certified Cost Estimator/Analyst (CCEA) and holds B.S. degrees in Statistics and Chemical Engineering with minors in Mathematics and Chemistry from Virginia Tech.

Peter Braxton

SME

Peter Braxton is a Subject Matter Expert at Technomics, Inc. He has over 20 years of experience performing cost and risk analysis and delivering associated training for a broad spectrum of federal government clients. The inaugural VP for Professional Development and a multiple ICEAA Educator of the Year winner, he has shown a long-standing commitment to knowledge sharing within the community. He holds an AB in Mathematics from Princeton and an MS in Operations Research from the College of William and Mary. He is a semi-retired game show contestant and avid cruciverbalist.



Dave Brown

SME

Dave Brown is a SME, with expertise in software and IT estimating. He has 30 years of experience providing cost estimating and analysis to NGA, DoD and DHS clients. Throughout his 11-year career at Technomics, Dave completed life cycle cost estimates, independent cost assessments, cost research, program management support, modeling and simulation, data analysis, and database development.

Alex Wekluk

Senior Analyst

Alex Wekluk is a Senior Analyst supporting NGA CAPE. He has sixteen years' experience performing cost, risk, and technical analyses for the DoD and the IC. He secured a USPTO patent for Marine Corps weapon design work and earned the IC Meritorious Unit Citation for exemplary performance identifying cost-reduction measures. He holds a BS in Mechanical Engineering from Virginia Tech and a MA in Economics from George Mason University.

Problem Statement: The Need for Better Software Estimates

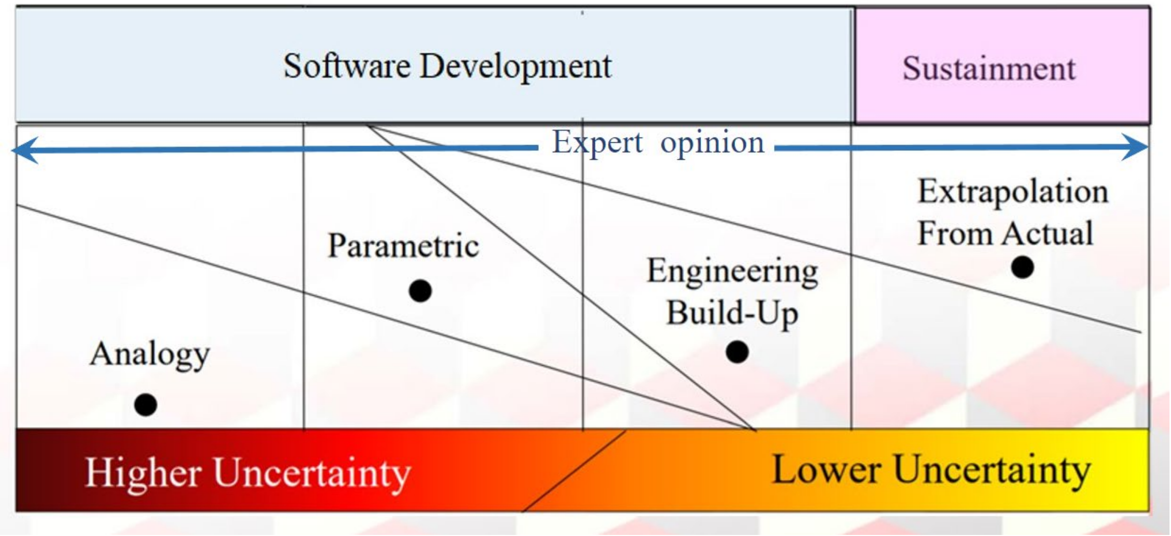
- Software / IT is notoriously prone to cost and schedule overruns:

	Olympics	Software/ IT	Dams	NASA/ DoD	Rail	Bridges/ Tunnels	Roads
Average Cost Growth	156%	43-56%	24-96%	52%	45%	34%	20%
Frequency of Occurrence	10/10	8/10	8/10	8/10	9/10	9/10	9/10
Frequency of Doubling	1 in 2	1 in 4	1 in 5	1 in 6	1 in 12	1 in 12	1 in 50
Average Schedule Delay	0%	63-84%	27-44%	27-52%	45%	23%	38%
Frequency of Schedule Delay	0/10	9/10	7/10	9/10	8/10	7/10	7/10

- Estimation of software development effort can be improved by:
  - Techniques that can be used early in the program life cycle
  - Accommodation for agile software development
  - Data-driven approaches that can use a variety of inputs such as user count, project duration, and development team capability
  - Use of historical benchmarking data on actual software development projects
- Existing commercial tools tend to be "black box," in that they do not expose the underlying equations or data

Problem Statement: Which Estimating Method is Best?

- The best approach to software development estimates depends on when the estimate is needed, and **data available**.



- Estimates generated late in the life cycle are most defensible with *Extrapolation from Actuals*.
  - Requires actual cost history
  - Suggested ICEAA 2022 presentation/paper: ***Are We Agile Enough to Estimate Agile Software Development Costs?*** *(Kosmakos / Brown)*
- *Mid-cycle estimates, where a sizing metric can be obtained can best use a parametric method*
  - Requires a standard sizing metric such as function points
  - This is the problem we are addressing in today's presentation.
- *Early-cycle estimates, where a sizing metric is not available need another method such as analogy, or T-shirt sizing*
  - Suggested ICEAA 2022 presentation/paper*:* ***Uncertainty of Expert Judgment in Agile Software Sizing***. (Braxton)

What is ISBSG?

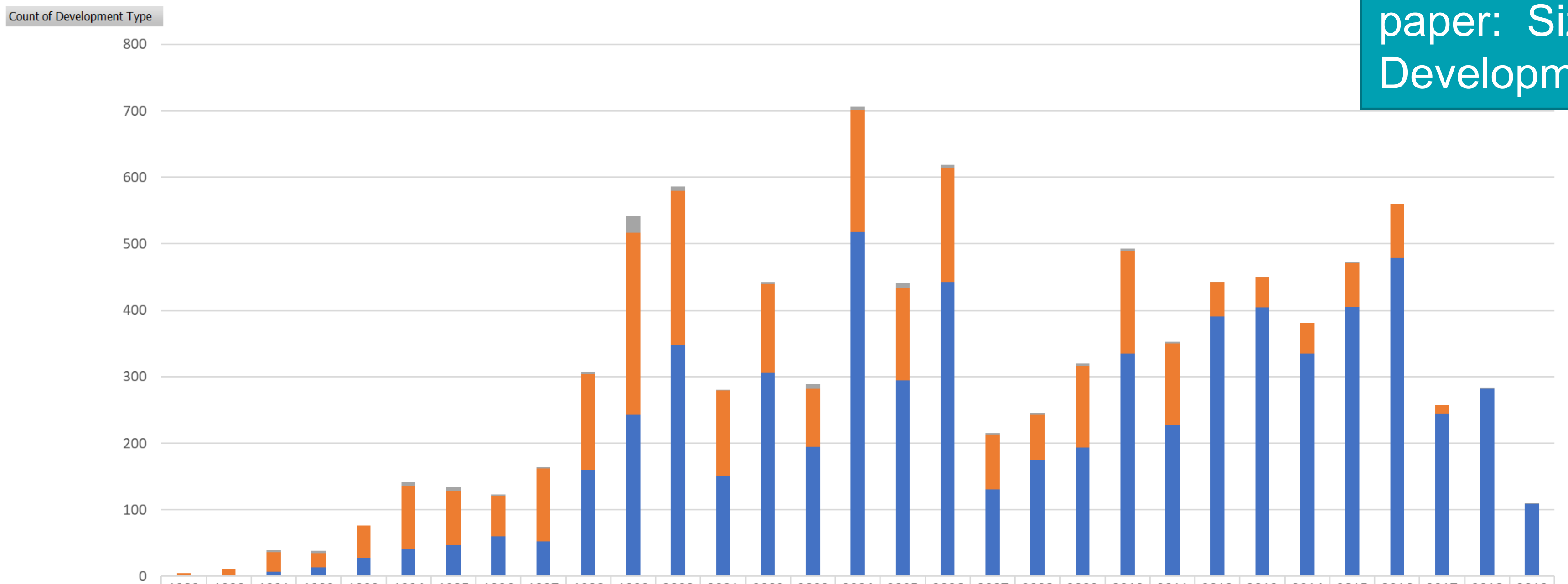
- International Software Benchmarking Standards Group (ISBSG) is a database containing software project data across the industry, https://www.isbsg.org/
  - Data submitted by IT and metrics organizations
  - Majority of the projects are IT Systems
    - 20+ Industry Sectors
  - 10,600 observations (projects)
    - Projects from 1989 to 2020
    - 252 fields of variables. Quantitative: 105, Qualitative: 147
  - U.S. and International data

- Why ISBSG?
  - Largest industry accepted benchmarking data for software metrics available to the public
  - A unique opportunity for cost estimators—much more data than we normally have!

ISBSG Data Overview

- Data Points per Fiscal Year
  - By Decade
    - 1990s: 1,576
    - 2000s: 4,145
    - 2010s: 3,804*
  - By Development Type
    - Enhancement: 6,620
    - New Development: 2,806
    - Re-development: 104

Additional break-outs in paper: Sizing Methods, Development Methodologies



	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004
Re-development			3	4		6	5	2	2	3	25	6	1	2	6	6
New Development	5	9	29	21	48	95	82	61	109	144	274	232	128	134	88	183
Enhancement		2	7	13	28	41	47	60	53	160	243	348	151	306	195	518

	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Re-development	8	5	2	3	4	3	3	1	1		1			1	1
New Development	139	172	82	68	122	155	123	51	45	47	66	81	13		
Enhancement	294	442	131	175	194	335	227	391	404	334	405	479	245	283	109

Analytical Approach – Dependent Variables

- Normalized Work Effort: Software Development Hours
  - Preferred approach
  - Significantly more data points than “Total Project Cost” (8,657 vs. 1,826)
  - Allows use of project specific labor rates
- Total Project Cost
  - Introduces exchange rate risk
    - 19 different currencies included in the data-set
    - US Dollar = 741 data points
    - Other currencies = 1,085 data points
- Used actuals, not estimates

Additional analysis in paper: Productivity, Size Growth

Analytical Approach – Independent Variables

- Evaluated potential independent variables
  - Driver of dependent variable (effort)
  - Larger dataset (n) is preferred
  - Reasonable to expect a cost analyst to know pre-software development
  - Tested 26 independent variable Spearman correlations to dependent variable

Ranked Correlation Summary

Effort to:	Coefficient	P-Value	Determination	n	Notes:
COSMIC Exit	0.715	0.0000000	0.511	258	COSMIC Only (Low n)
COSMIC Read	0.713	0.0000000	0.508	258	
COSMIC Write	0.711	0.0000000	0.506	258	
Lines of Code	0.692	0.0000000	0.478	704	Difficult to Estimate
COSMIC Entry	0.673	0.0000000	0.453	258	COSMIC Only (Low n)
Adjusted Function Points	0.620	0.0000000	0.385	6,660	
Average Team Size	0.558	0.0000000	0.312	1,274	Function of Effort and Elapsed Time
Project Elapsed Time	0.554	0.0000000	0.307	8,446	
Added count	0.547	0.0000000	0.299	2,291	
Input count	0.498	0.0000000	0.248	1,794	
Enquiry count	0.475	0.0000000	0.226	1,776	
Changed count	0.450	0.0000000	0.203	2,114	
Output count	0.417	0.0000000	0.174	1,794	
File count	0.394	0.0000000	0.156	1,794	
User Base - Distinct Users	0.327	0.0000000	0.107	672	
User Base - Concurrent Users	0.270	0.0000000	0.073	797	
Interface count	0.252	0.0000000	0.064	1,776	
Deleted count	0.183	0.0205630	0.033	2,114	Statistically Insignificant

Analytical Approach – EERs

- Effort Estimating Relationships (EERs) developed based on ISBSG data
- The team analyzed 30 potential input variables, >100 EERs, and both linear and log-linear (i.e., power) relationships
  - Examples shown in table to the right
- Down-selection of independent variables based on:
  - Inputs that didn't show statistical significance
  - Inputs that don't follow a logical relationship (for example, relationship showing negative correlation known to be positive)
  - Inputs for which there was insufficient data
  - Inputs that showed high multicollinearity
- Consideration was given to whether inputs will be known to the estimator, especially early in the program life cycle. For example, *Project Elapsed Time* or *Team Size* might be unknown
  - Solved by providing multiple EERs that accommodate various combinations of input variables

	Adjusted Function Points	Function Point Categories	COSMIC Data	User Base - Concurrent Users	IT experience (Weighted Average)	Average Team Size	Software Process CMM	Agile/Non-Agile
1	X							
2	X							X
3	X						X	
4	X						X	X
5	X			X				
6	X			X				X
7	X			X			X	
8	X			X			X	X
9	X			X	X			
10	X			X	X			X
11	X			X	X		X	
12	X			X	X		X	X
13	X			X		X		
14	X			X		X		X
15	X			X		X	X	
16	X			X		X	X	X
17	X			X	X	X		
18	X			X	X	X		X
19	X			X	X	X	X	
20	X			X	X	X	X	X
21		X						
22		X						X
23		X					X	
24		X					X	X
25		X		X				
26		X		X				X
27		*X*		*X*			*X*	
28		*X*		*X*			*X*	*X*
29		X		X	X			
30		X		X	X			X

Larger sample set is included in the paper

What is SWEET?

- Software Effort Estimating Tool (SWEET) is a prototype tool that estimates the effort required to develop software with whatever input parameters are known

- Model features:
  - Simple-to-use interface
  - Runs as a stand-alone Excel spreadsheet without the need for Macros, VBA code, or 3rd party software
  - Life Cycle aware: model applies tailored method based on the inputs available at different stages of the program lifecycle
  - Clear Box allows the estimator to see underlying equations AND data
  - EER customization using only estimator-selected source data
  - Compatible with methods developed using function point sizing

SWEET Interface

Sizing

Adjusted Function Points	
Adjusted Function Points	100

Input count	Output count	Enquiry count	File count	Interface count

COSMIC Entry	COSMIC Exit	COSMIC Read	COSMIC Write

Other Inputs (Quantitiative)

User Base - Distinct Users	
User Base - Concurrent Users	1000
IT experience (Weighted Average)	
Project Elapsed Time	
Max Team Size	
Average Team Size	

Other Inputs (Non-Quantitiative)

Software Process CMM	Level 3+
Agile Development	Yes

Automatically filters data-set based on inputs

EER	Remove	ISBSG Project ID	Data Quality Rating	UFP rating	Year of Project	Industry Sector	Language Type	Primary Programming Language	Count Approach	Adjusted Function Points	Normalized Work Effort
EER		10200	A	A	2009	lectronics & Computer	3GL	Java	IFPUG 4+	90	1112
EER		10248	A	A	2012	Communication	3GL	Java	IFPUG 4+	293	5829
EER		10283	B	B	2008	Financial	3GL	PL/I	IFPUG 4+	224	12548
EER		10388	A	A	2009	Service Industry	4GL	ASP.Net	IFPUG 4+	139	5062
EER		10426	A	A	2008	Service Industry	3GL	C#	IFPUG 4+	101	2006
EER		10481	B	B	2004	Service Industry	3GL	PL/I	IFPUG 4+	304	5377
EER		10486	A	A	2007	Government	3GL	JavaScript	IFPUG 4+	132	1485
EER		10676	A	A	2007	Government	4GL	ABAP	IFPUG 4+	249	2204
EER		10759	A	B	2004	Government	3GL	C#	IFPUG 4+	263	911
EER		10898	A	A	2007	Financial	3GL	C#	IFPUG 4+	1153	16532
EER		11083	A	B	2005	Government	4GL	ABAP	IFPUG 4+	228	2877

Removes outliers from calculation with simple input

- As the Estimator enters input data and selects or removes data points, SWEET will automatically pick the best EER, run the statistical regression, and display the results.
  - All calculations are closed end formulas, using Excel's OFFSET and LINEST functions.
  - This allows for real-time calculation, without the need for Solver or any VBA code.

SWEET Outputs

- Quick view on "Inputs" tab provides dynamic, high-level effort estimate as inputs are changed
  - Estimated effort
  - R^2
  - Degrees of freedom
- Detailed EER stats provide real-time equation information
  - Data fields are added and removed as "Inputs" tab is updated
  - Effort equation is developed
  - Stats are modified
  - Estimate is provided
- Based on the input variables used and underlying data selected by the estimator, SWEET has a nearly infinite number of potential EERs under the hood

Results (Quick View)

Estimated Effort (Hours)	2,316.0
R^2	0.530148012
Degrees of Freedom	278

Detailed EER Stats

	LN(IT experience (Weighted	LN(Adjusted Function Points)	Intercept
Coefficient	-0.157	0.672	4.779
SE	0.120	0.059	0.464
R^2 / SEE	0.338	0.848	
F / dF	71.937	282	
SSR / SSE	103.513	202.890	
T-stat	-1.304	11.289	10.294
P-value	0.1932	0.0000	0.0000

Log Form — Normalized Effort (hrs) = e^(4.78) * Adjusted Function Points^0.67* IT experience (Weighted Average)^-0.16

	LN(Project Elapsed Time)	LN(IT experience (Weighted	LN(Adjusted Function Points)	Intercept
Coefficient	0.798	-0.339	0.504	4.372
SE	0.076	0.104	0.053	0.395
R^2 / SEE	0.530	0.718		
F / dF	104.559	278		
SSR / SSE	161.613	143.232		
T-stat	10.502	-3.261	9.489	11.056
P-value	0.0000	0.0012	0.0000	0.0000

Initial Estimate - Function Points Only

Inputs:

Type of Analysis	LOG				
Output	Normalized Work Effort	*Full life-cycle effort for all teams reported in hours.*			
Sizing	Adjusted Function Points				
	Adjusted Function Points	3500			
	Input count	Output count	Enquiry count	File count	Interface count
	COSMIC Entry	COSMIC Exit	COSMIC Read	COSMIC Write	
Inputs (Quantitiative)	User Base - Distinct Users				
	User Base - Concurrent Users				
	IT experience (Weighted Average)				
	Project Elapsed Time				
	Max Team Size				
	Average Team Size				
Inputs (Non-Quantitiative)	Software Process CMM				
	Agile Development	Yes			
Analysis Inputs	Significance (Alpha)	0.05			

Results:

Results (Detailed Stats)

	1	2	3		
	2	1	Intercept		
	Agile (0 - Not Agile, 1 - Agile)	LN(Adjusted Function Points)	Intercept		
Coefficient	-0.440	0.660	4.143		
SE	0.177	0.010	0.052		
R^2 / SEE	0.393	1.102			
F / dF	2111.487	6523			
SSR / SSE	5125.815	7917.571			
T-stat	-2.488	64.937	79.338		
P-value	0.0129	0.0000	0.0000		

EERT Calculation

Estimated Effort (Hours)	8,855

EER

Log Form	Normalized Effort (hrs) = e^(4.14) * Adjusted Function Points^0.66* (e^(-0.44))^Agile Development
Linear Form	

Updated Estimate – Multiple Inputs

Inputs:

Type of Analysis	LOG				
Output	Normalized Work Effort	*Full life-cycle effort for all teams reported in hours.*			
Sizing	Adjusted Function Points				
	Adjusted Function Points	3500			
	Input count	Output count	Enquiry count	File count	Interface count
	COSMIC Entry	COSMIC Exit	COSMIC Read	COSMIC Write	
Inputs (Quantitiative)	User Base - Distinct Users	150			
	User Base - Concurrent Users				
	IT experience (Weighted Average)	5			
	Project Elapsed Time				
	Max Team Size				
	Average Team Size				
Inputs (Non-Quantitiative)	Software Process CMM				
	Agile Development	Yes			
Analysis Inputs	Significance (Alpha)	0.05			

Results:

Results (Detailed Stats)

	1	2	3	4	5	
	4	3	2	1	Intercept	
	Agile (0 - Not Agile, 1 - Agile)	LN(IT experience (Weighted Average))	LN(User Base - Distinct Users)	LN(Adjusted Function Points)	Intercept	
Coefficient	-0.585	-0.529	0.108	0.574	5.507	
SE	0.570	0.277	0.039	0.144	1.121	
R^2 / SEE	0.341	1.024				
F / dF	7.638	59				
SSR / SSE	32.057	61.906				
T-stat	-1.026	-1.908	2.786	3.996	4.914	
P-value	0.3089	0.0612	0.0072	0.0002	0.0000	

EERT Calculation

Estimated Effort (Hours)	10,869

EER

Log Form	Normalized Effort (hrs) = e^(5.51) * Adjusted Function Points^0.57* User Base - Distinct Users^0.11* IT experience (Weighted Average)^-0.53* (e^(-0.58))^Agile Development
Linear Form	

SWEET Benefits

Why SWEET is better than other available options and research?

- Life Cycle Aware EERs
  - Instead of publishing a single EER, SWEET has multiple EER's, each with its own set of input variables. One EER may be appropriate for the middle to end of the life cycle, when more is known about the program. SWEET also has EERs that will work even if a program is early in the life cycle and not much is known.
- Clear Box
  - SWEET allows estimators to see the underlying equation(s) AND the underlying data. Any tool / estimate is only as good as the data that goes into it.
- Customizable EERs
  - In addition to seeing the underlying data, the estimator has the option to select the data that is most analogous, and de-select any other data. SWEET then re-builds the coefficients of the EER using the data selected by the estimator.
  - Estimators can create their own EER without needing any statistical skills or tools. Estimators need to assess how analogous the source data is for their estimate.

Conclusions

- ISBSG data is a valuable resource for software cost estimation
- SWEET can dynamically create EERs, based on variables and data selected by the estimator
  - Is appropriate for a different parts of the life cycle
  - Is superior to a static EER
  - Doesn't require deep statistical knowledge
  - Saves time
- Software estimates are most defensible when
  - Underlying EERs are shown
  - Underlying data is shown
  - Estimators have the option to keep or omit data, based on comparability to the target system


Technomics
Better Decisions Faster