Capability-Based Software Cost Estimation (CaBSCE)

Modernizing Software Cost Estimates for Agile and DevSecOps

Introduction

CaBSCE introduces a novel method for cost estimation that aligns with modern software practices, where flexibility and speed are paramount. We at the SEI are designing CaBSCE specifically for early lifecycle estimates, and our objectives are to

- align with Agile and DevSecOps practices by eliminating dependencies on detailed requirements and specifications
- eliminate the reliance on source lines of code (SLOC)
- produce defensible, flexible, and objective cost estimates

Methods

We will analyze the wealth of software cost and scheduling data that the Department of Defense (DoD) collects to identify software components with similar functions and features. By identifying these functions, CaBSCE can provide actual effort ranges for similar functions.

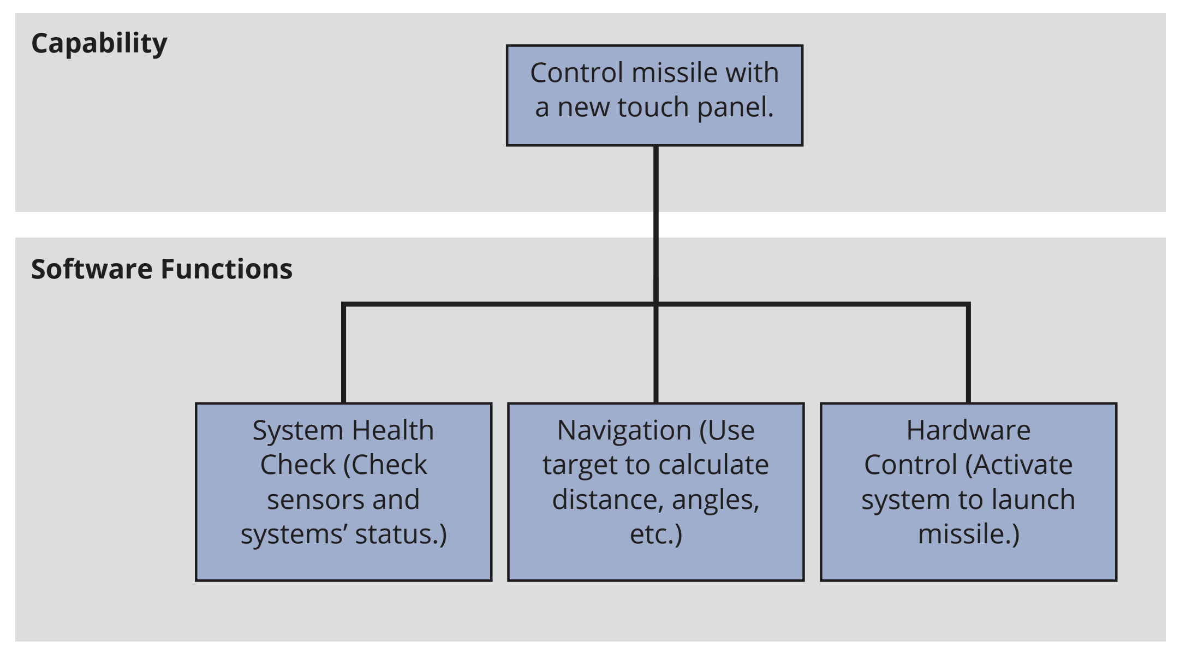
The data points that underlie these effort ranges may represent different architectural decisions, many organizations, and varying related requirements. This enables CaBSCE to account for uncertainties and be generalizable.

Initial Work

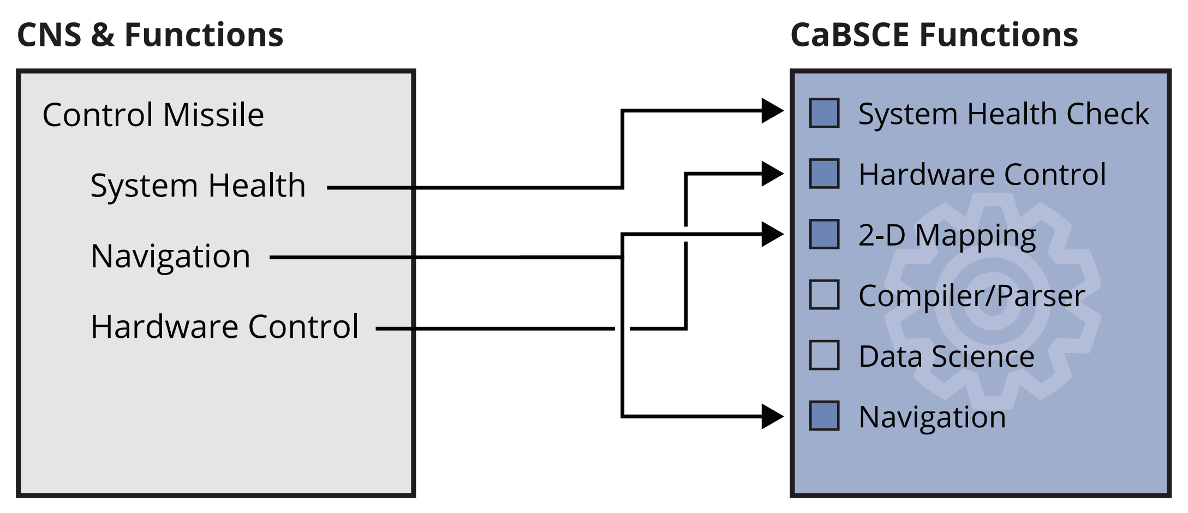
We classified application types from a commercial software development dataset (International Software Benchmarking Standards Group [ISBSG]) with development complexity ratings and a list of implemented functions.

Software cost estimation is as easy as 1, 2, 3:

1. Identify major software functions from a capability needs statement (CNS).


Capability
Control missile with a new touch panel.
Software Functions
System Health Check (Check sensors and systems' status.)
Navigation (Use target to calculate distance, angles, etc.)
Hardware Control (Activate system to launch missile.)

2. Map major software functions to CaBSCE's software functions.

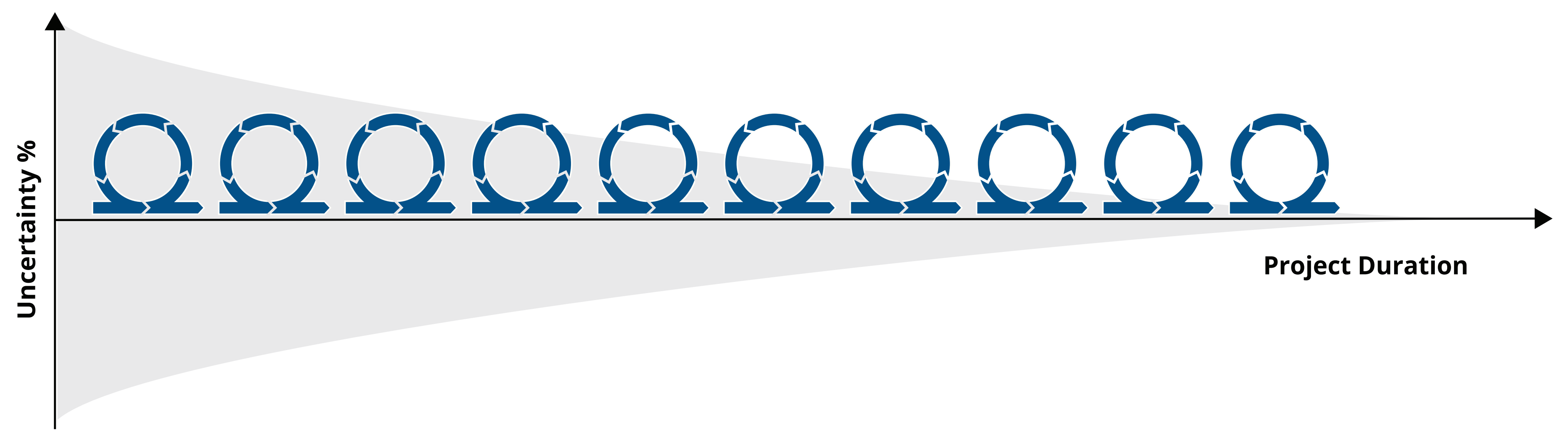


3. Get corresponding effort ranges; sum for the total program software effort.

WBS	Line Item	Min Effort	Max Effort
1	Control Missile		
1.1	Software	12410	25820
1.1.1	System Health	250	1500
1.1.2	Navigation	7296	14592
1.1.3	Hardware Control	486	9728

CaBSCE provides feasible effort ranges rather than a point estimate. Point estimates hide high risk and high uncertainty.

How CaBSCE Aligns with Agile and DevSecOps



Know high-level capabilities

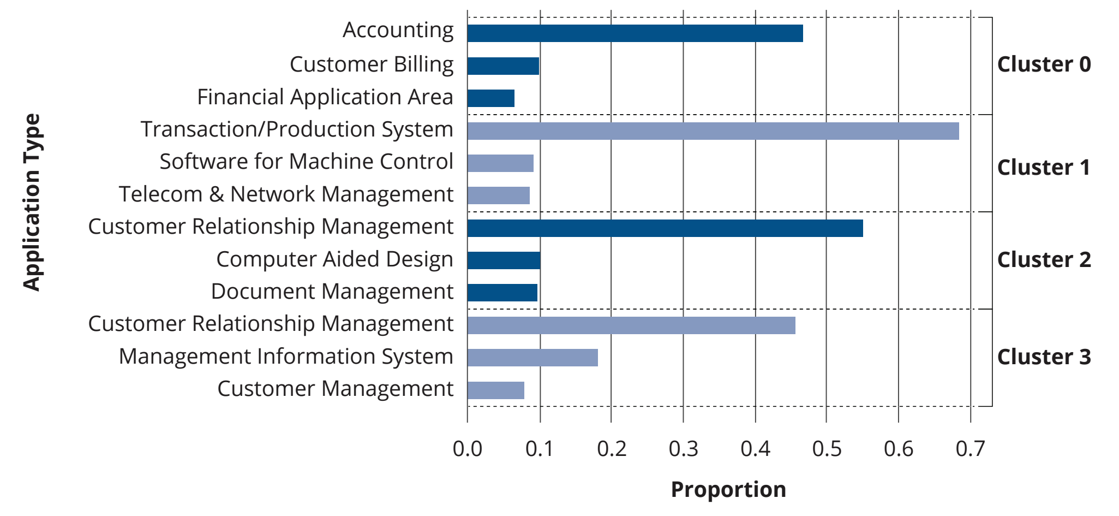
Estimate effort ranges indicative of uncertainties

Uncertainties clarified through iterations

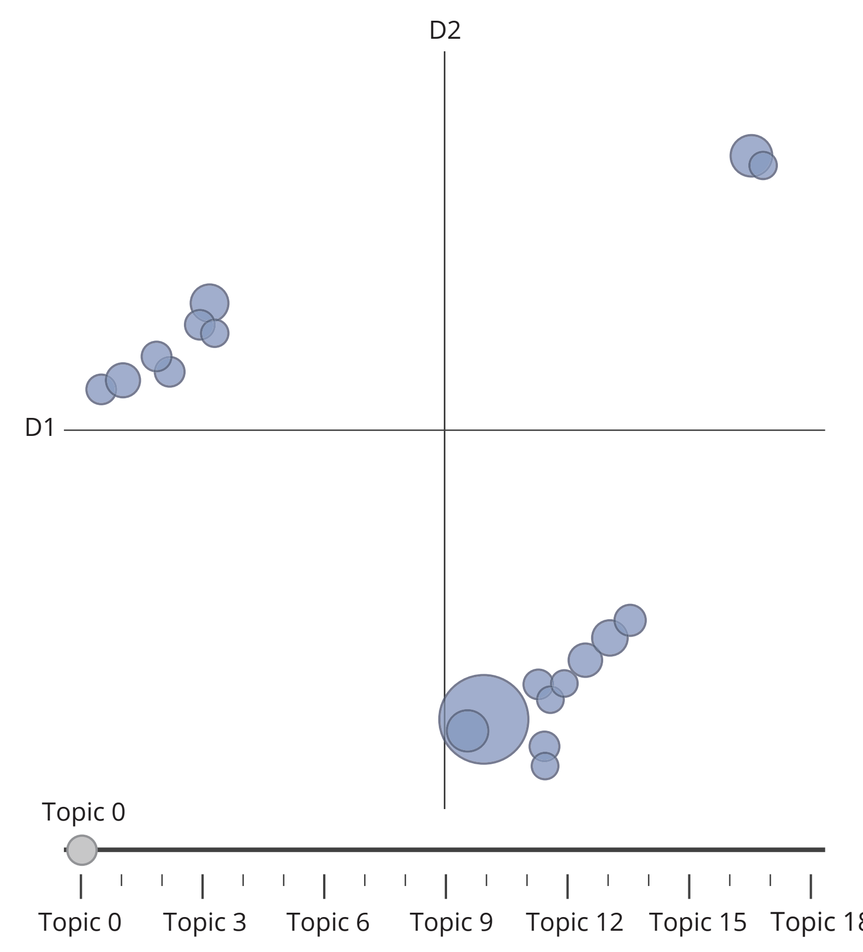
More detailed and accurate estimates

Interim Results

We used a development complexity definition to classify the implemented functions. K-means clustering on the complexity ratings returned groups of application types such as those below, where the figure shows the top three application types in each group:



We also listed the software functions implemented and used BERTopic (NLP [natural language processing]) to identify similar application types based on the listed functions:



NLP topic modeling of the list of software functions returned more consistent results.

Future Steps

We will identify software functions implemented and run NLP topic modeling with DoD software development data.

Carnegie Mellon University
Software Engineering Institute

Anandi Hira | avhira@sei.cmu.edu
Bill Nichols, Chris Miller, Julie Cohen, Nick Testa, Shannon Gallagher, and Jeff Mellon