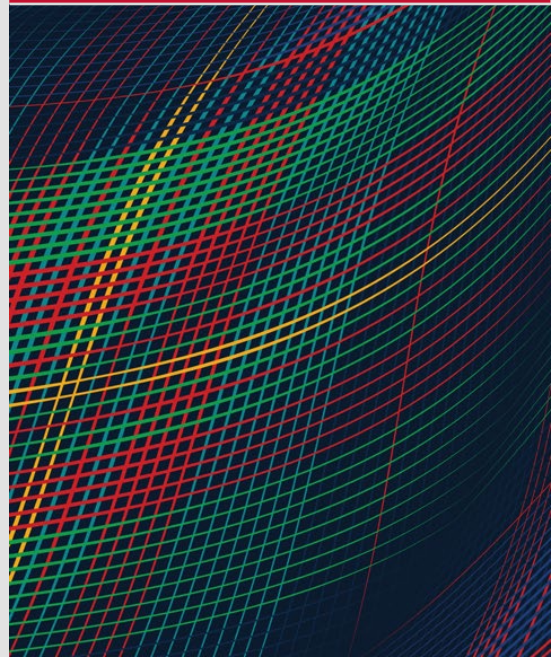
Carnegie Mellon University
Software Engineering Institute

Bridging the Gap Between Traditional Sizing Measures and Agile/DevSecOps Measures

JOINT IT & SOFTWARE COST FORUM
SEPTEMBER 18, 2024

Anandi Hira and Chris Miller



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.

DM24-1137

CMU SEI is a DoD R&D Federally Funded Research and Development Center

Established in 1984 at Carnegie Mellon University (CMU)

Charged to improve the state of the practice of software engineering, cybersecurity, and AI.

Collaborates with CMU and broadly in academia, government, and industry

Capable of conducting both fundamental research and classified work

~610 staff members

Offices in Pittsburgh and DC, with locations near customer facilities in MA, TX, and CA

Agenda

Carnegie Mellon University Software Engineering Institute

What Are the Gaps?

Bridging the Gap

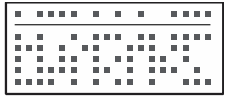
CaBSCE: Capability-Based Software Cost Estimation

Closing the Gap

How to Participate

Advancement Gap: Advancements in Development Outpace Estimation

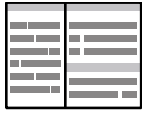
Software Development Advancements



Push Cards and Batch Processing

Text Code Editor

Visual Basic with some GUI Interfacing

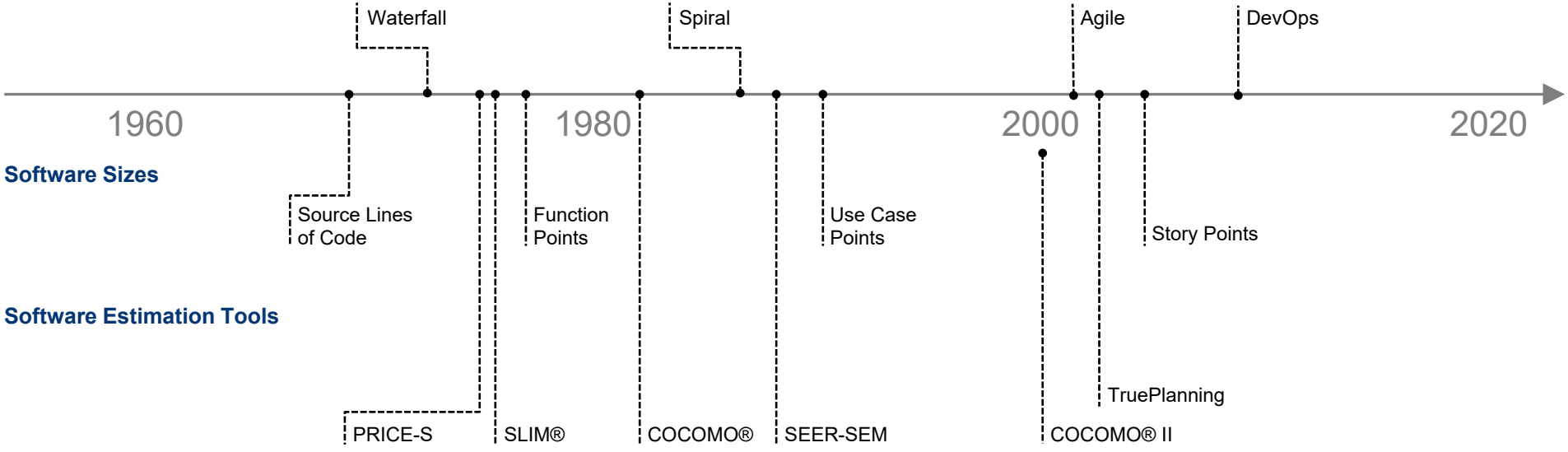


IDEs Such as Eclipse

Web Development and Version Control Tools (Git and Github)

Software Development Tool Stacks

Lifecycle Models



Software Sizes

Source Lines of Code

Function Points

Use Case Points

Story Points

Software Estimation Tools

PRICE-S

SLIM®

COCOMO®

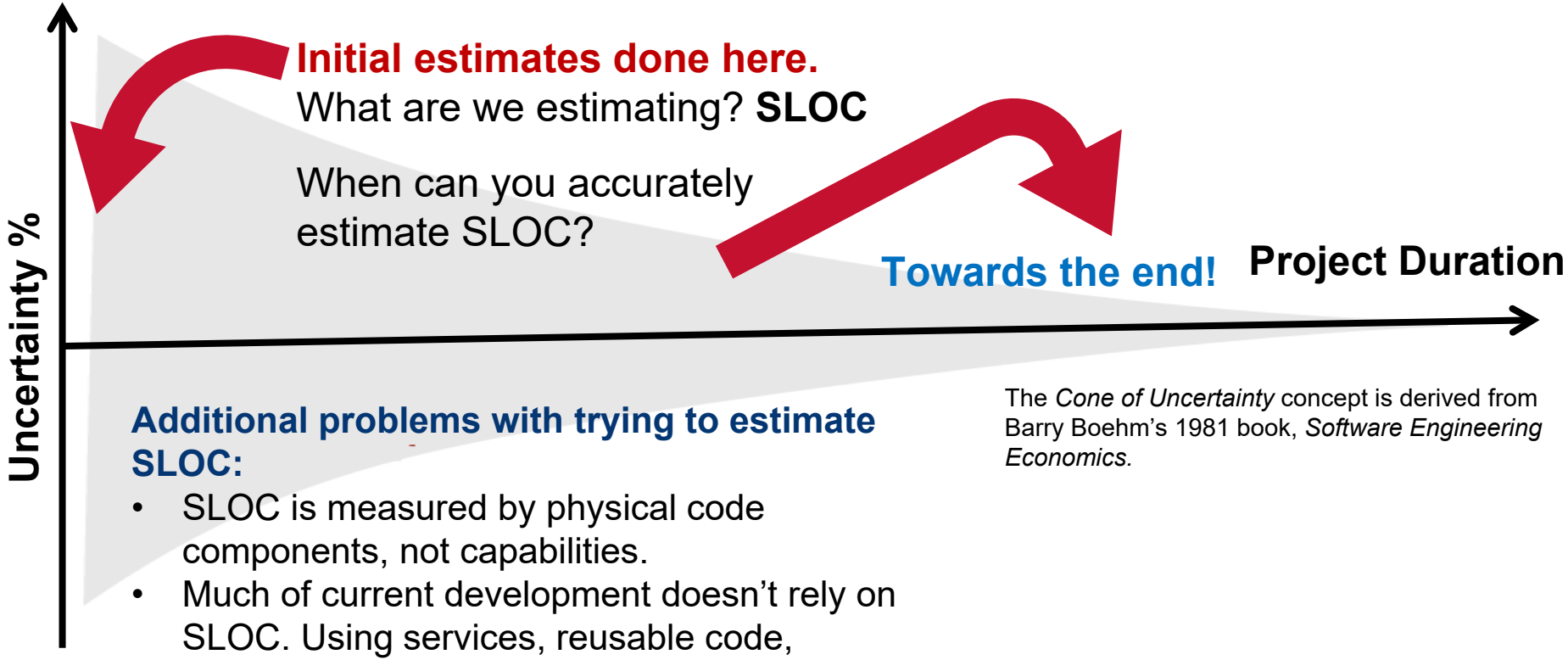
SEER-SEM

COCOMO® II

TruePlanning

Estimation Gap: Problem with Source Lines of Code (SLOC)

Carnegie Mellon University
Software Engineering Institute

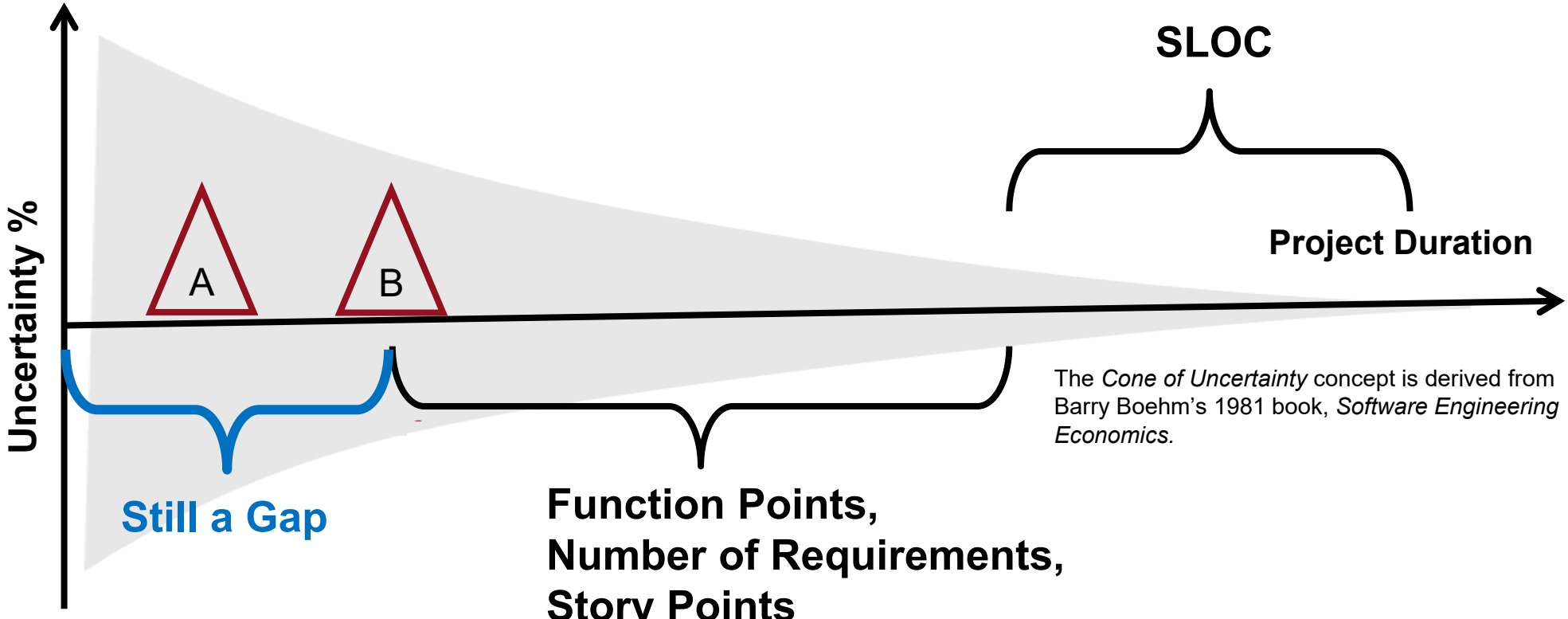


Additional problems with trying to estimate SLOC:

- SLOC is measured by physical code components, not capabilities.
- Much of current development doesn't rely on SLOC. Using services, reusable code, COTS/GOTS,

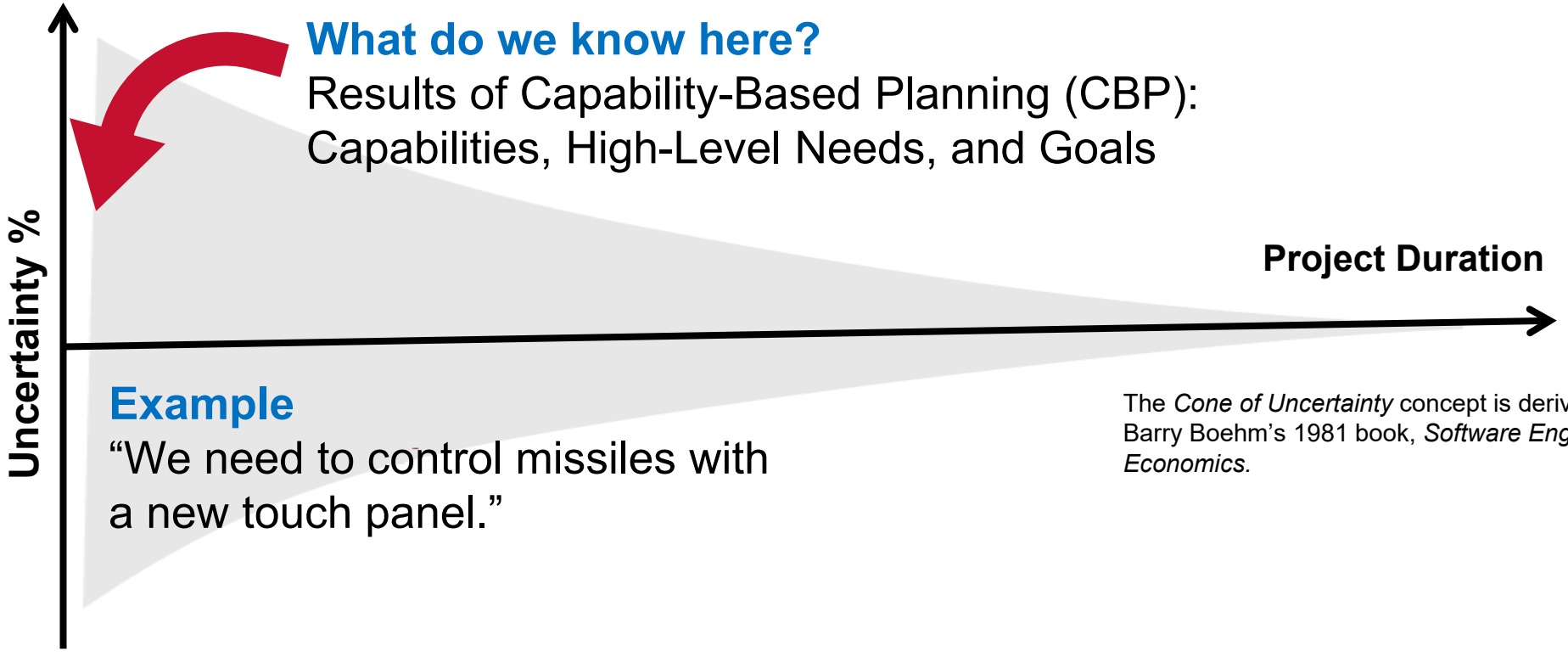
The *Cone of Uncertainty* concept is derived from Barry Boehm's 1981 book, *Software Engineering Economics.*

What About Other Size Metrics?



The *Cone of Uncertainty* concept is derived from Barry Boehm's 1981 book, *Software Engineering Economics*.

Bridge the Gap: Starting with a Clean Slate



What do we know here?

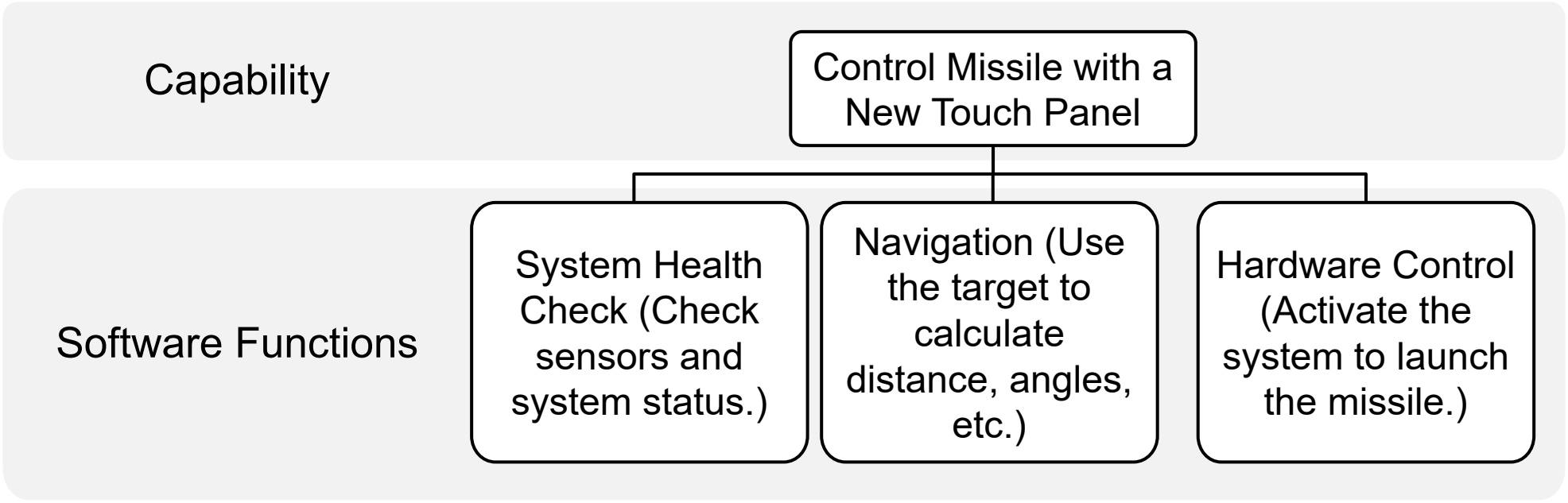
Results of Capability-Based Planning (CBP): Capabilities, High-Level Needs, and Goals

Example

"We need to control missiles with a new touch panel."

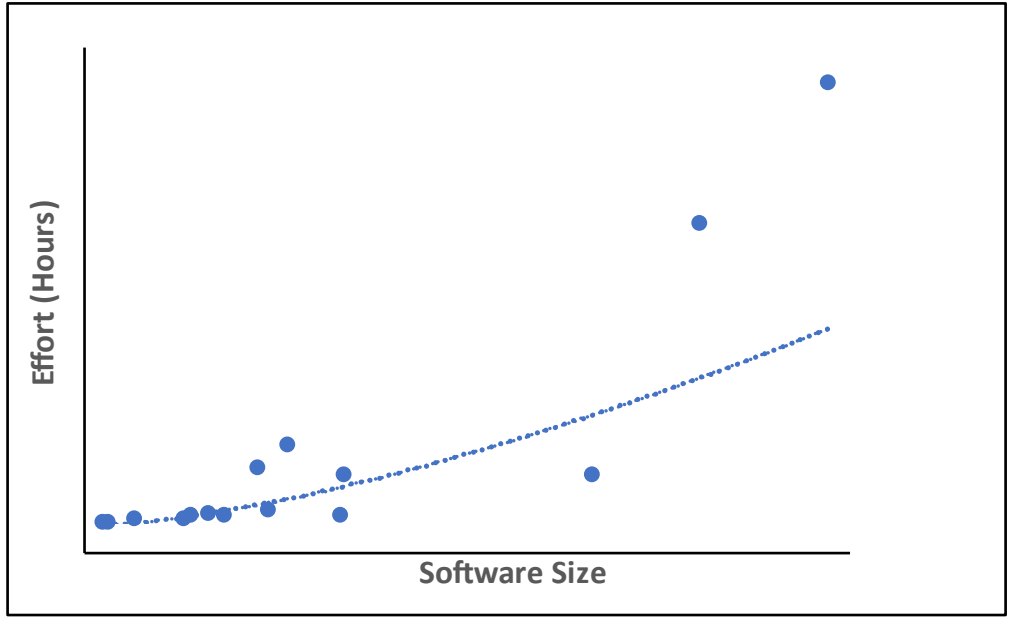
The *Cone of Uncertainty* concept is derived from Barry Boehm's 1981 book, *Software Engineering Economics*.

Capabilities and Software Functions



The Role of Size Measures

- Size metrics provide an objective measure that correlates to software development costs/effort.
- Software size measures are most useful if they can be estimated when they are needed (i.e., early in the lifecycle, intermittently throughout the lifecycle).
- Size measures should allow estimators to perform cost analyses (e.g., compare alternatives).



Identifying a Comprehensive Set of Software Functions

Carnegie Mellon University Software Engineering Institute

GPS/Navigation

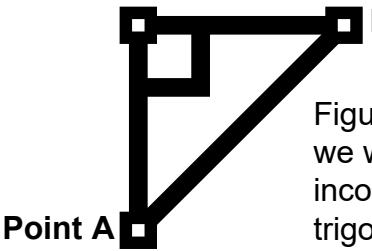
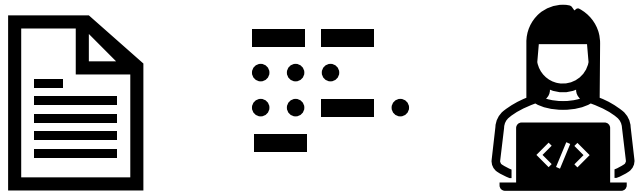


Figure out where we are and where we want to go. We will need to incorporate some math, specifically trigonometry, to determine how to get there.

vs.

Compilers/Parsers



Parse the text and code for prespecified patterns.

Redefining software size means demystifying the software sizing process while providing an objective, flexible, and defensible estimation method.

Bridging the Gap Between Traditional Sizing Measures and Agile/DevSecOps Measures

CaBSCE: Capability-Based Software Cost Estimation

Carnegie Mellon University
Software Engineering Institute

Research Objectives

A new software sizing method that has capability clusters/groups based on similar software functions

- based on information known at program initiation (e.g., capabilities)
- accounts for uncertainties (e.g., implementation variations)

A cost model based on capability clusters and group sizing

- effort ranges of similar capabilities
- evidence based but also flexible

Note: The size is not a number; it is a type of software capability.

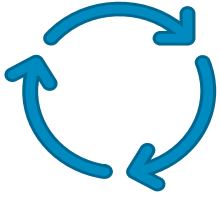
Identifying Capability Clusters

Carnegie Mellon University Software Engineering Institute

Software Development Data

ID	Descriptive Variables	Effort
1234	ABC 123	#
2234	XYZ acd	#
1345	mn ABC	#
2346	ABC yyy	#
3456	dco XYZ	#
4456	XYZ qrs	#

Iterate for improvements.



Study descriptive variables to identify clusters of similar software functions:

1. Use complexity descriptions to classify data points.
2. Identify keywords for the functionality implemented.
3. Define groups of similar software functions using clustering and natural language processing (NLP).

Capability Clusters

ID	Descriptive Variables	Effort
1234	ABC 123	#
2234	XYZ acd	#
1345	mn ABC	#
2346	ABC yyy	#
3456	dco XYZ	#
4456	XYZ qrs	#

Orange and purple represent two clusters of similar software functions.

Developing a Corresponding Cost Model

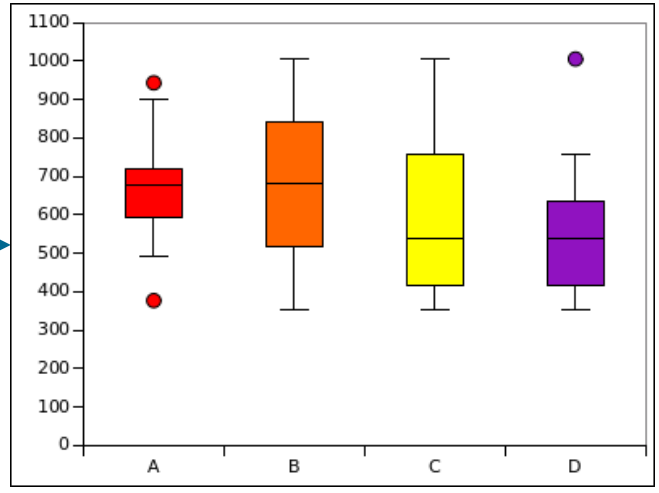
Capability Clusters

ID	Descriptive Variables	Effort
1234	ABC 123	#
2234	XYZ acd	#
1345	mn ABC	#
2346	ABC yyy	#
3456	dco XYZ	#
4456	XYZ qrs	#

Get effort ranges per capability cluster.

Minimum, Maximum, Mean, Median, Variance, Standard Deviation

Cost Model



Orange = Cluster B
Purple = Cluster D

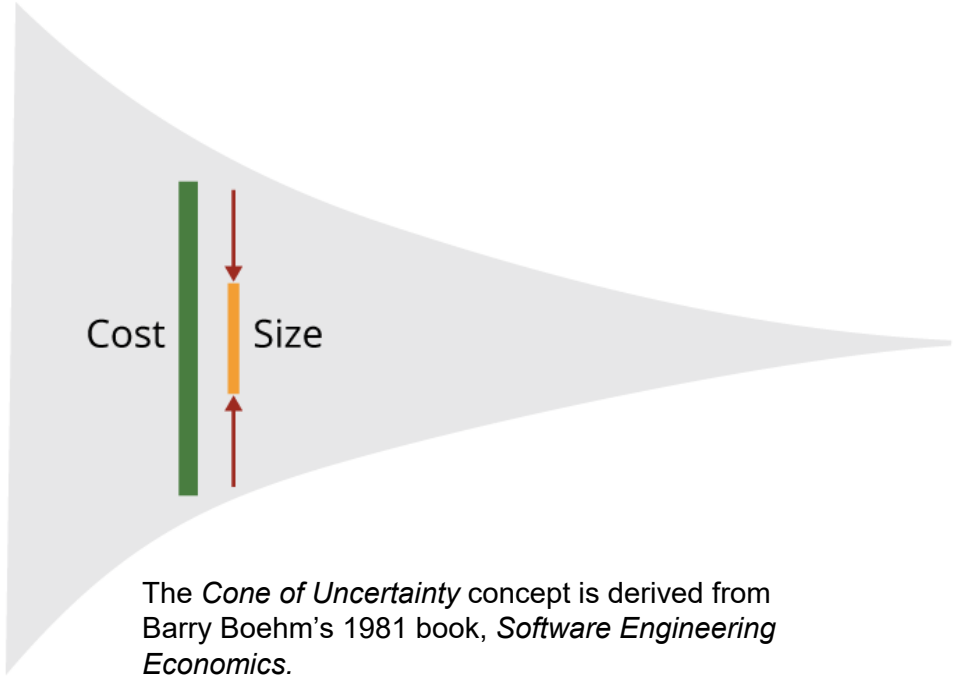
Relevance and Impact of CaBSCE

Overcomes Weaknesses of Other Size Metrics

- Use early in the lifecycle.
- Use for evidence-based estimates.
- Reduce sizing errors.

Clarify Uncertainties

- Current reports on scope/cost growth also include sizing errors.
- With size constant, changes can be tracked due to
  - implementation uncertainty
  - scope growth

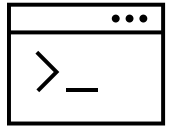


The *Cone of Uncertainty* concept is derived from Barry Boehm's 1981 book, *Software Engineering Economics.*

Closing the Gap

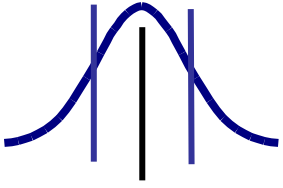
The “just trust us” approach will likely not work.

- SLOC cannot be suddenly dropped; the transition should happen as alternative estimation solutions “prove” to be better predictors.



Maintain existing estimation approaches while piloting new alternatives.

- DoD policy and statutes require SLOC (e.g., Software Resources Data Report [SRDR]).

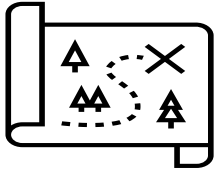


Validate improvements.

- Calculate inherent errors between estimates and actuals for all estimation methods.
- Derive prediction intervals (PRED) to determine which approach provides the best estimation capability.

Develop a transition roadmap.

- Expect DoD-wide adoption to be gradual; require objective evidence.



How to Participate

Collaborate on CaBSCE

1. Pilot: Become an early adopter to contribute to and improve the state of the practice.
2. Review intermediate results for their usability and applicability.
3. Share organizational data.
4. Execute an NDA with SEI for data submitted to government/DoD.

For more details, read our information sheet by scanning the QR code on the left or by sending email to info@sei.cmu.edu.

Join our team!

We are looking to grow our team and add a member who has these skills:

- software estimation, process modeling, and software measurement
- able to serve as an analytical, customer-focused expert who works with customers
- familiar with parametric cost models.

If this opportunity sounds interesting, read more about it or scan the QR code on the left.

Contact Us

Anandi Hira
Researcher

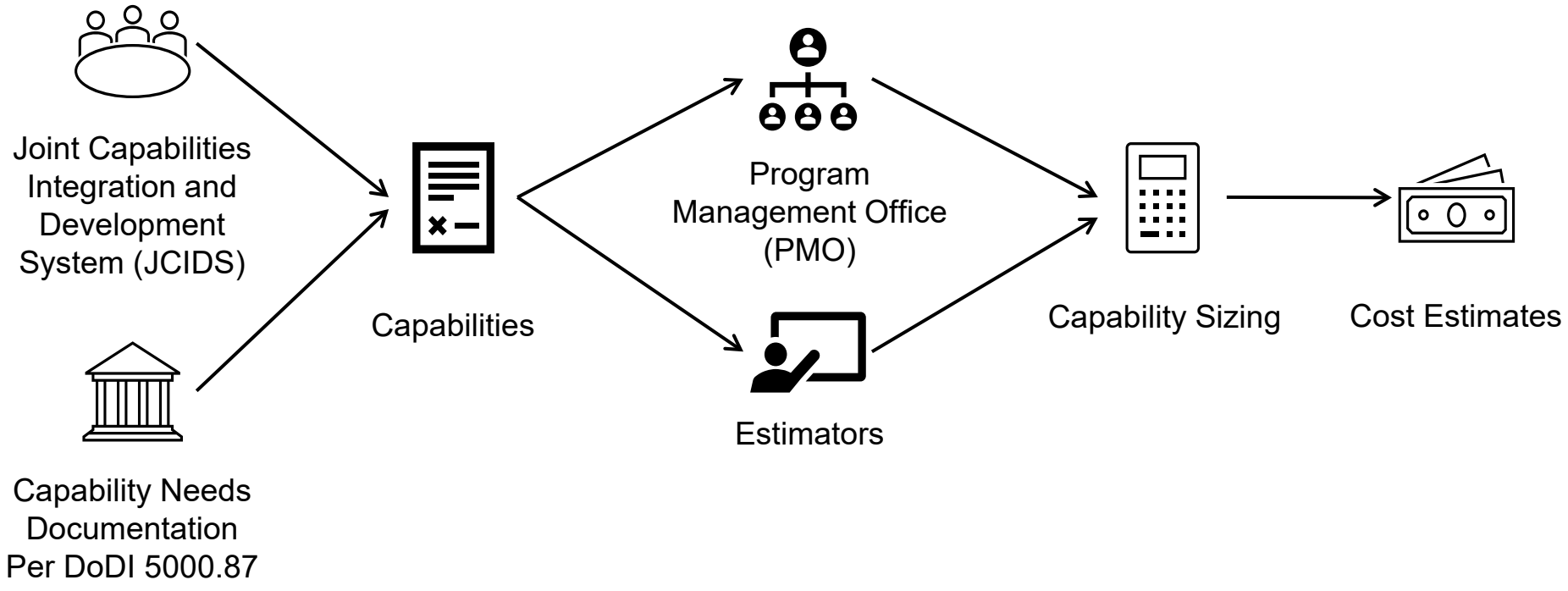
Chris Miller
Senior Engineer

Email: info@sei.cmu.edu

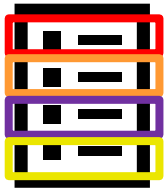
Backup

Carnegie Mellon University
Software Engineering Institute

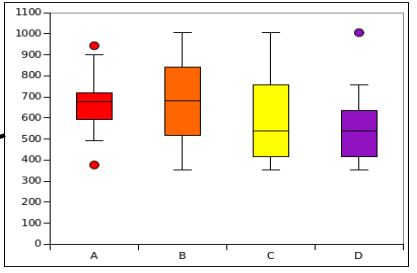
CaBSCE Use Case: DoD Estimation Process



CaBSCE Use Case: Estimators' Steps



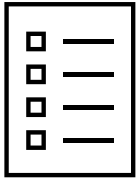
Potential Capability Taxonomy



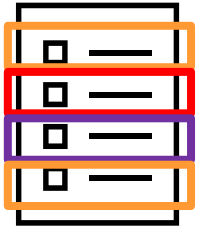
Potential Cost Model

Estimators

Map to capability clusters.

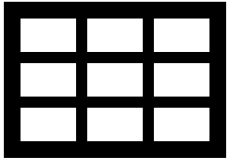


CNS or SW-ICD



CNS or SW-ICD

Use capability clusters and effort ranges to develop an estimate.



Software Cost/Effort Estimate