WHY CARE ABOUT CEBOK-S IF WE DON'T BUILD SOFTWARE? (SWR11)

CAROL DEKKERS
QUALITY PLUS TECHNOLOGIES, INC.

WHO AM I?

CAROL DEKKERS, SCEC, PMP, CFPS (FELLOW), P.ENG. CSM

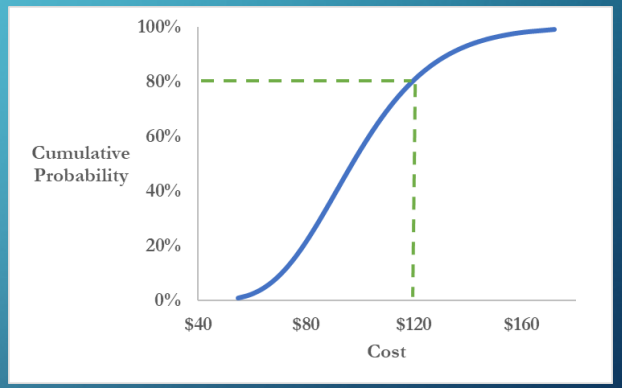
Lead Author of CEBOK-S	ICEAA 2022 Educator of the year 2023 Global Leader in Consulting 2023 IFPUG Honorary Fellow SCEC, CFPS (Fellow), PMP, CSM
Industry leadership	U.S. expert and project editor for ISO/IEC JTC1 SC7 SW Engineering standards ICEAA Board & Software SIG IFPUG Past President & Industry Standards Chair Past PMI Metrics SIG, ASQ SD Inst SW Excellence
Other	Mechanical engineer, author, speaker, consultant Mother of 2 & yaya of 1, FL resident Love of tennis, travel, craft beverages and sunsets

GOALS (TALKING POINTS)

- What's different about software cost estimating?
- Why CEBoK-S now ?
- How does CEBoK-S fit with CEBoK ?
- Coverage and content of CEBoK-S
- Next steps SCEC
- Questions

SOFTWARE VS HARDWARE DEVELOPMENT SAME OR DIFFERENT ?

- Software and hardware development have major differences…
  - Product life cycles
  - Cost drivers (and dis/economies of scale)
  - Behaviors
  - No two software development projects are identical
- Software development methodologies, tools, and technologies are constantly changing -- estimating cost and schedule is non-trivial
- It is critical that cost analysts understand the unique challenges of estimating software and how to apply basic cost estimating techniques to each software estimating situation



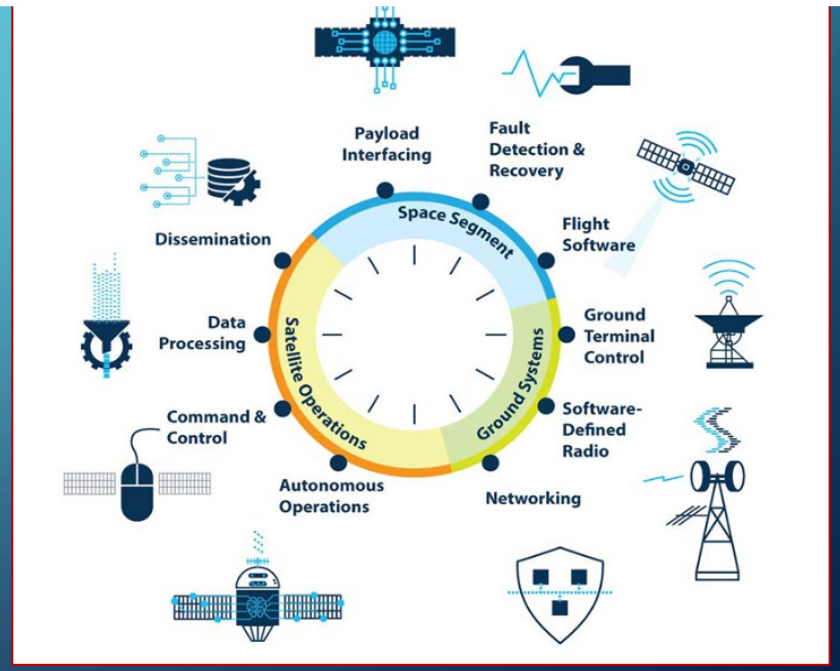
WHY IS SOFTWARE COST ESTIMATING IMPORTANT?

SOFTWARE COST = DOMINANT COST CENTERS

F-35 Joint Strike Fighter

- Compared to F-16, the F-35 has:
  - 177x computer code, and
  - 300x software development effort costs
  - 90% of the functionality is delivered by software $1T USD investment (over 50 years)[2]
- 9 countries (23 bases):
  - UK 2015: "one of the most delayed and problematic fighter programs in history…limitations in the flight control software"[3]
  - 500th F-35 delivered in Mar 2020

Satellite programs dominated by complex application software



CEBoK-S

WHY IS SOFTWARE COST ESTIMATING IMPORTANT ?

SOFTWARE IS EMBEDDED IN EVERYTHING…

Future layered in-vehicle and back-end architecture

Existing layer Modified layer New layer

Cloud platform	Combine in-vehicle data with environmental data
Connectivity (back-haul)	
User interface/user experience/ human-machine interface	
Applications	Significant increase in number of applications
Artificial intelligence/ advanced analytics	Analyze data for real-time decisions and autonomous driving
Middleware layer/ operating system	Abstract applications from hardware
Electronic/electrical hardware[1]	
Sensors / Actuators / Power components	Closely controlled add-on app and modules due to safety considerations
Vehicle	

[1]Including operating system in status quo.

Future factors for brand differentiation:

- **Infotainment features** requiring "plug and play" capabilities
- **Autonomous capabilities** including sensor-fusion algorithms as a complement to hardware
- **Safety features** based on "fail-operational" behavior
- Software will move further down the stack to hardware (smart sensors)
- Stacks become horizontally integrated
- New layers will be added to the stack

McKinsey&Company

WHY IS SOFTWARE COST ESTIMATING IMPORTANT?

COST & SCHED GROWTH = LEGACY OF DISASTER

	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

1 **COMMON**

Multiple Industries Experience Significant Cost and Schedule Growth – Has Been a Problem for a Long Time

2 **FREQUENT**

70-80% of Projects Experience Cost and Schedule Growth

3 **HIGH**

Cost: 50% or More on Average (Mean)
Schedule: 30% or More on Average (Mean)

4 **EXTREME (FOR COST)**

Cost Growth in Excess of 100% Is a Common in Most Projects (1 in 6)

CEBoK-S

Source: Christian B. Smart, Solving for Risk Management: Understanding the Critical Role of Uncertainty in Project Management

WHY IS SOFTWARE COST ESTIMATING IMPORTANT ?

SOFTWARE PROJECTS = MAJOR OVERRUNS

IT Comparison

IT organizations waste an average of $101 million for every $1 billion spent on projects and programs due to poor project performance, compared to the global average of $114 million for every $1 billion spent.*

A number of the study results for IT organizations outperform the global total in the 2020 *Pulse of the Profession®*. The table below identifies in red the areas that have a meaningful difference from the global total.

PROJECT OUTCOMES	GLOBAL TOTAL	IT
Percentage of projects meeting goals and business intent	69%	71%
Percentage of projects completed on time	53%	59%
Percentage of projects completed within budget	59%	64%
Percentage of projects with scope creep	35%	33%
Percentage of projects deemed failures	13%	14%
Percentage of project budget lost if a project fails	37%	35%

Source" PMI Pulse of the Profession® 2020
Research Highlights by Region and Industry

EXAMPLES OF CHALLENGED PROGRAMS
(WITH SOFTWARE COMPONENTS)

Phoenix federal pay system (COTS solution)

$310M CDN budget 2009-16

May 2018: 600K pay backlog

2019: $2.6B CDN to stabilize data (pre-replacement)

Denver airport baggage system

$560M USD & over budget,

16 month delay,

system finally scrapped

(software scope creep & poor project communication)

Healthcare.gov (Agile)

$860M USD spent,

$150M overrun

"failed agile project"

36 states, 10x users, abandoned

Miller Coors (ERP package)

Corporate merger, with 7 instances of SAP software

Failed integration

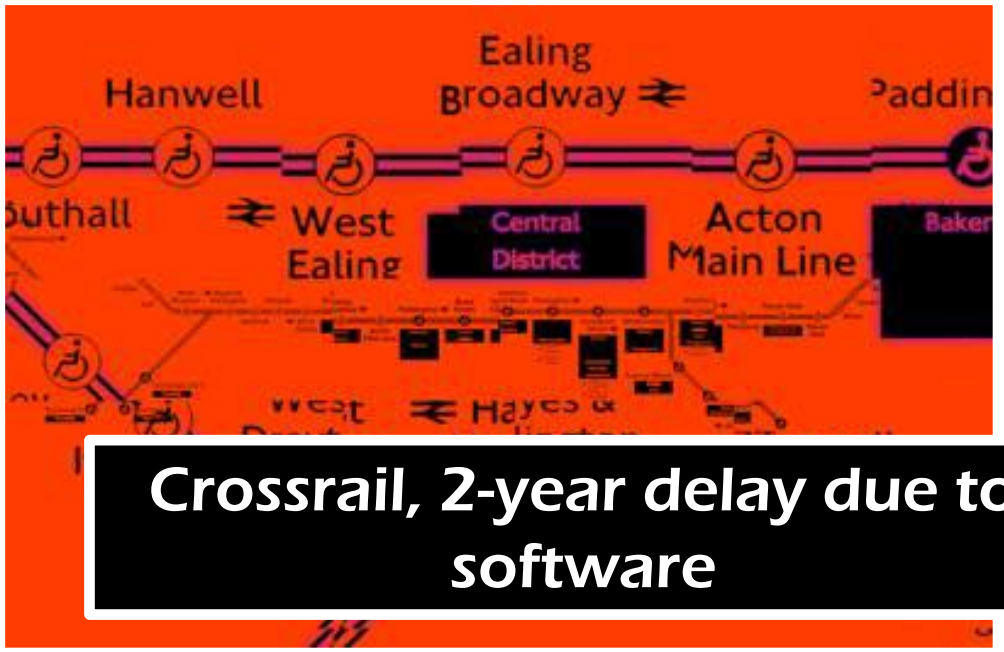
$100 M USD lawsuit/counter

Finally settled out of court

CEBoK-S

WHY IS SOFTWARE COST ESTIMATING IMPORTANT ?

SOFTWARE CAN IMPACT PROGRAM DELIVERY



Crossrail, 2-year delay due to software

Netherlands toll road system delayed opening due to software

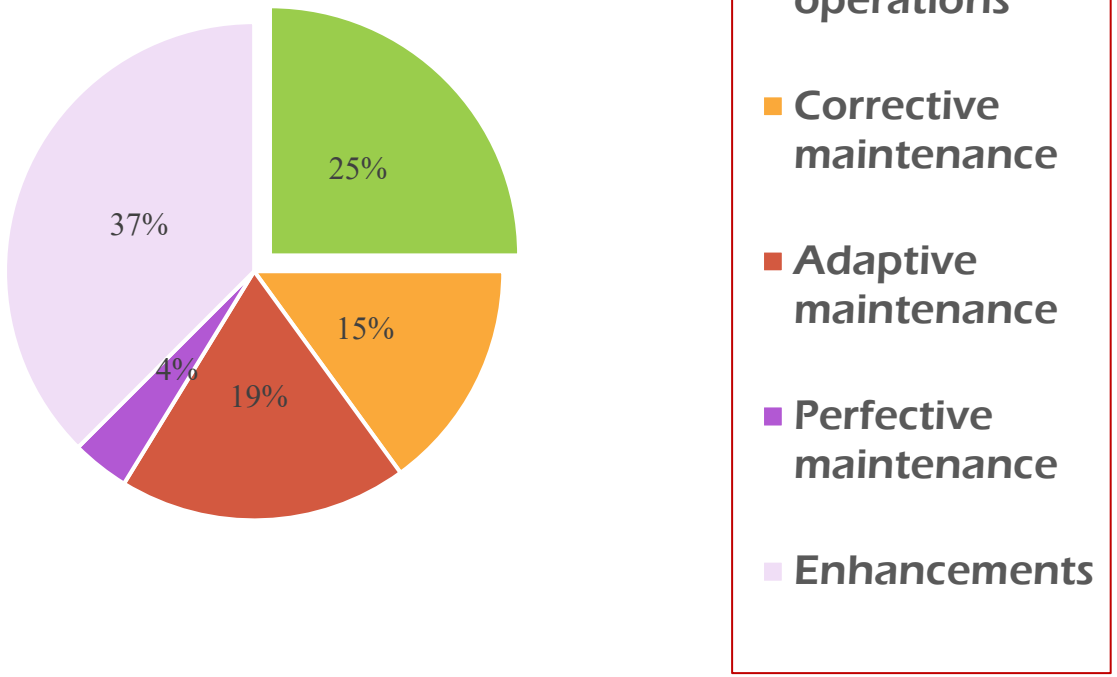
Eurlings tunnels program overspend > €47 M:

- Unproven new technology
- Tunnel security software overlooked

WHY IS SOFTWARE COST ESTIMATING IMPORTANT?
SOFTWARE SUSTAINMENT

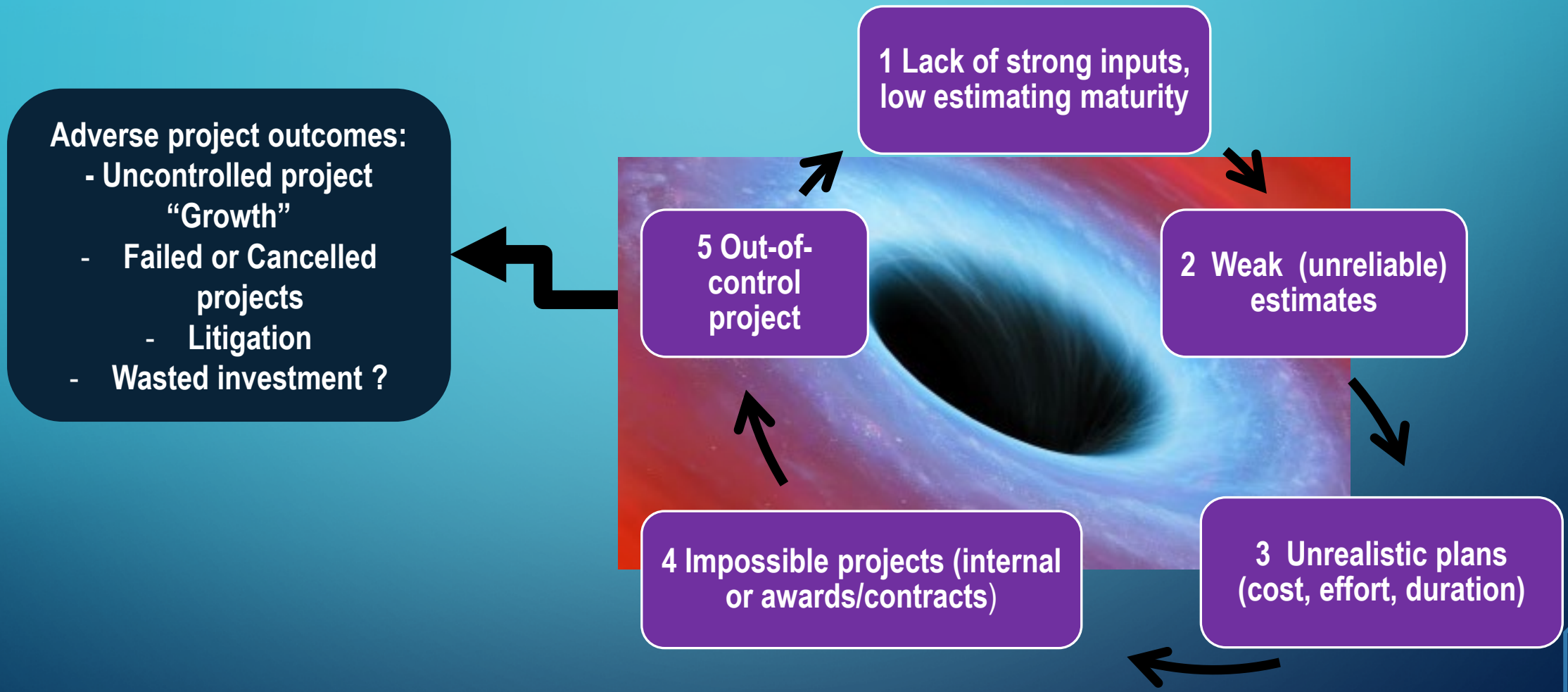
Sustainment costs are 75 % of Total Cost of Ownership (TCO) of Software

Software sustainment cost is derived from the changes made to software after it has been delivered to the end user. Software does not "wear out" but it will become less useful as it gets older, plus there WILL always be issues within the software itself.

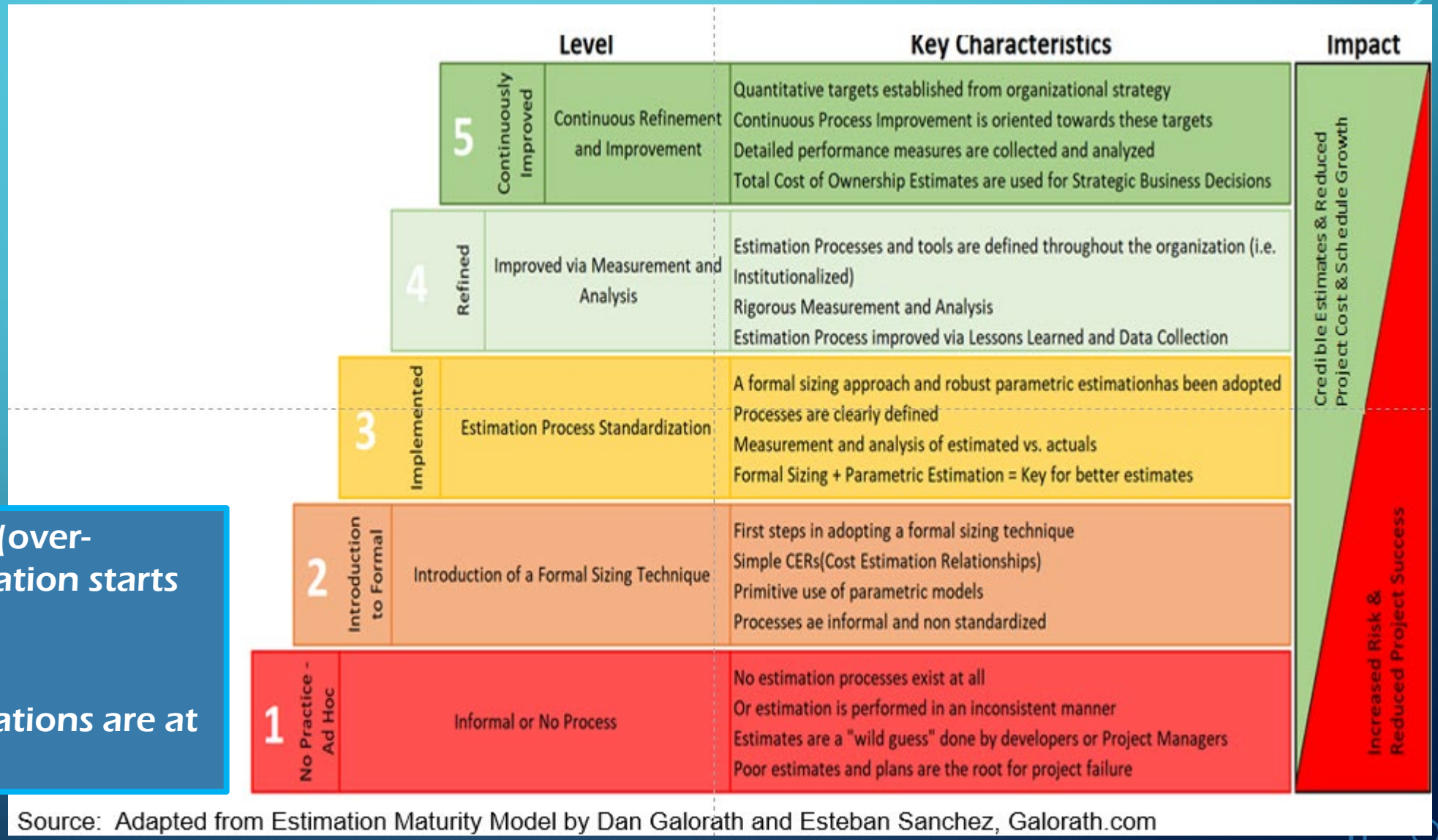


CEBoK-S

WHY IS SOFTWARE COST ESTIMATING IMPORTANT
BREAK THE CYCLE WITH KNOWLEDGE



SOFTWARE ESTIMATING MATURITY



Source: Adapted from Estimation Maturity Model by Dan Galorath and Esteban Sanchez, Galorath.com

- **Estimation Bias (over-optimism) mitigation starts at Level 2**
- **95% of organizations are at Level 1**

INTRODUCING THE SOFTWARE COST ESTIMATION BODY OF KNOWLEDGE CEBOK-S

LESSON 0: INTRODUCTION TO CURRICULUM

LESSON 1: IMPORTANCE AND MOTIVATION FOR CEBOK-S

LESSON 2: SOFTWARE DEVELOPMENT PARADIGMS

LESSON 3: FIVE-STEP ESTIMATING PROCESS

LESSON 4: ESTIMATING CUSTOM SOFTWARE DEVELOPMENT

LESSON 5: SOFTWARE SUSTAINMENT

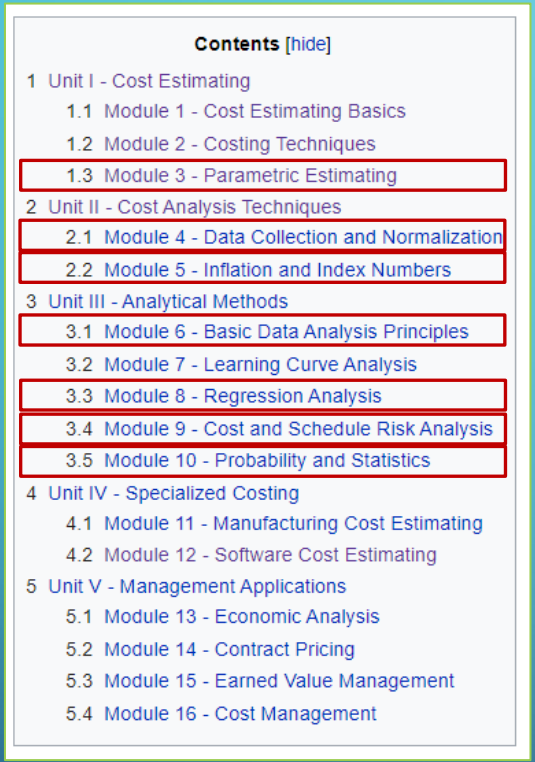
LESSON 6: ESTIMATING PROCURED SOFTWARE SOLUTIONS

LESSON X: SOFTWARE SIZE

LESSON Y: PRODUCTIVITY

LESSON Z: COMMERCIAL ESTIMATING MODELS

CEBOK-S PRE-REQUISITE KNOWLEDGE ICEAA CEBOK



Contents [hide]

1 Unit I - Cost Estimating
- 1.1 Module 1 - Cost Estimating Basics
- 1.2 Module 2 - Costing Techniques
- 1.3 Module 3 - Parametric Estimating

2 Unit II - Cost Analysis Techniques
- 2.1 Module 4 - Data Collection and Normalization
- 2.2 Module 5 - Inflation and Index Numbers

3 Unit III - Analytical Methods
- 3.1 Module 6 - Basic Data Analysis Principles
- 3.2 Module 7 - Learning Curve Analysis
- 3.3 Module 8 - Regression Analysis
- 3.4 Module 9 - Cost and Schedule Risk Analysis
- 3.5 Module 10 - Probability and Statistics

4 Unit IV - Specialized Costing
- 4.1 Module 11 - Manufacturing Cost Estimating
- 4.2 Module 12 - Software Cost Estimating

5 Unit V - Management Applications
- 5.1 Module 13 - Economic Analysis
- 5.2 Module 14 - Contract Pricing
- 5.3 Module 15 - Earned Value Management
- 5.4 Module 16 - Cost Management

- **Basic knowledge of cost estimating content as highlighted**
- **Available to ICEAA members https://wikidev.iceaaonline.com/wiki/Main_Page**

DIFFERENT TYPES OF SOFTWARE → DIFFERENT ESTIMATING TECHNIQUES

Custom-developed software

- **Software development efforts → turn software requirements into working software**
- **Includes custom enhancement and replacement of legacy applications**
- **Examples**
  - Standalone automated information system (AIS), business information systems
  - Real-time and/or embedded software (e.g., in aircraft or hardware components)
  - Logistics, command and control, health care, student loans, government, security, telecom, data warehouse, sales, fulfillment, banking…

DIFFERENT TYPES OF SOFTWARE → DIFFERENT ESTIMATING TECHNIQUES

Procured (packaged) software:

- **Commercial/Government-off-the-shelf (COTS/GOTS)**
  - Packaged software with user-level and enterprise licenses
  - Software as a Service (SaaS)
  - Service oriented architecture (SOA)
  - Information System/Business System (Enterprise Resource Planning - ERP systems)
  - Enterprise Data Warehouse (EDW) / Data Mart / Operational Data Stores (ODS)

Many projects are hybrid combinations (across a spectrum) ranging from fully custom developed to fully procured.

CEBOK-S COVERAGE: DEVELOPMENT + PROCUREMENT + SUSTAINMENT

Software-intensive program[1]

Investment:

- Program/project management
- Systems engineering
- BPR/ Change management
- System Development
- System Procurement
  - Hardware (make and/or buy)
  - Software (buy)
- System level integration & test
- System deployment/implementation

Operations & support (O&S)[2]

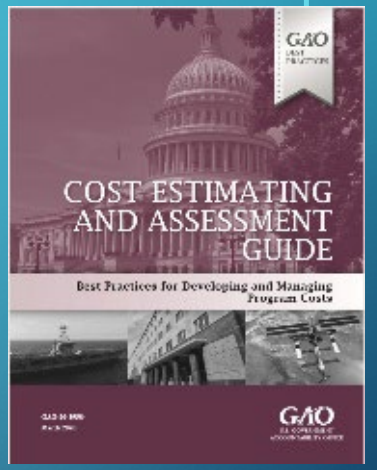
- Help desk/service desk support
- Technology refresh/upgrade
- System sustainment

Software life cycle (example)

- Investment
  - Plan (sourcing, business case, governance)
  - Develop and/or procure
    - **Software development** — Requirements → implementation — **SCEBoK Lessons 2,4**
    - **Software procurement** — **SCEBoK Lesson 6**
  - System Integration
  - Deployment
- Operations & Support (O&S) – Software Sustainment — **SCEBoK Lesson 5**
  - Software Product Changes (Maintenance)
  - Recurring Software Licenses
  - Help Desk, Hosting, Facilities, etc.
- End of life

TYPES OF SOFTWARE ESTIMATES/2

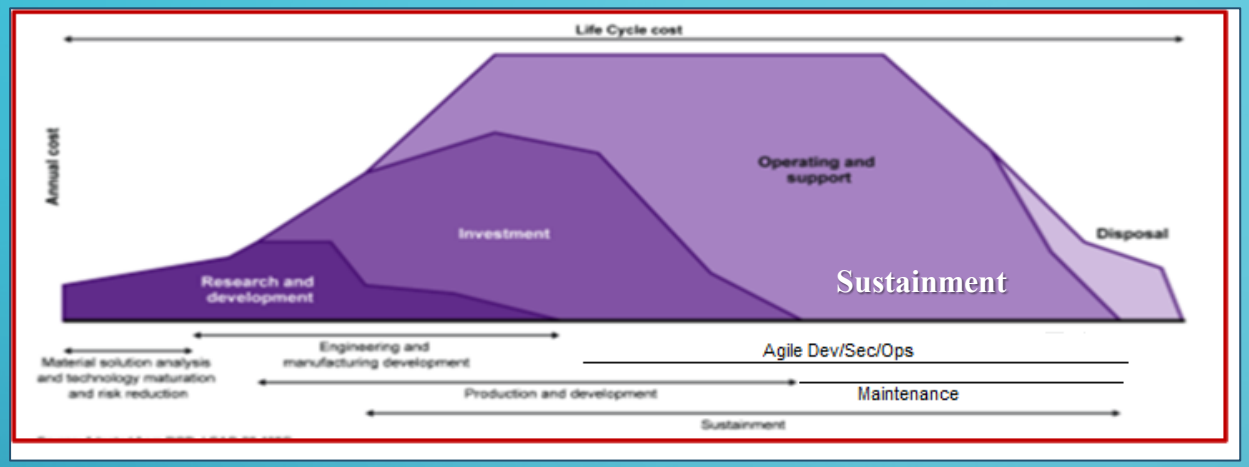
- **Software cost estimates may cover:**
  - the entire **software life cycle** (R&D through to disposal)
  - **software development** (requirements through first installation)
  - **software procurement**
  - only a portion of the life-cycle (e.g. the testing phase, or the coding phase)
  - multiple projects and products, (depending on the purpose of the estimate)
- **Software life-cycle is typically organized around two major phases: Investment and Operations & Support (O&S)**



Notional Software Life-cycle Model[1]

1. Notional Software Life Cycle Model adapted from the GAO Cost Estimating and Assessment Guide Mar 2020

TYPES OF SOFTWARE ESTIMATES/3



Generic Product Life Cycle Costs [1]

- SW life cycle cost estimate (LCCE) → R&D + Investment + Sustainment
- SW development (SD) estimate → Subset of Investment costs
- SW procurement estimate → Subset of Investment costs
- SW sustainment estimate → Subset of System O&S costs

1 Source: GAO Cost Estimating and Assessment Guide Mar 2020

TYPES OF SOFTWARE ESTIMATES

- **Software Life Cycle Cost estimate (LCCE)**
  - Also called “Cradle to grave”
  - Includes all software life cycle activities
  - SCEBoK provides guidance on preparing reliable estimates for:
    - software development;
    - software procurement;
    - software sustainment
  - **CEBoK-S does not provide all components for a software LCCE**
  - Contributes to software-intensive program estimate

ROLE OF SOFTWARE ESTIMATING

Software life cycle focus: Investment

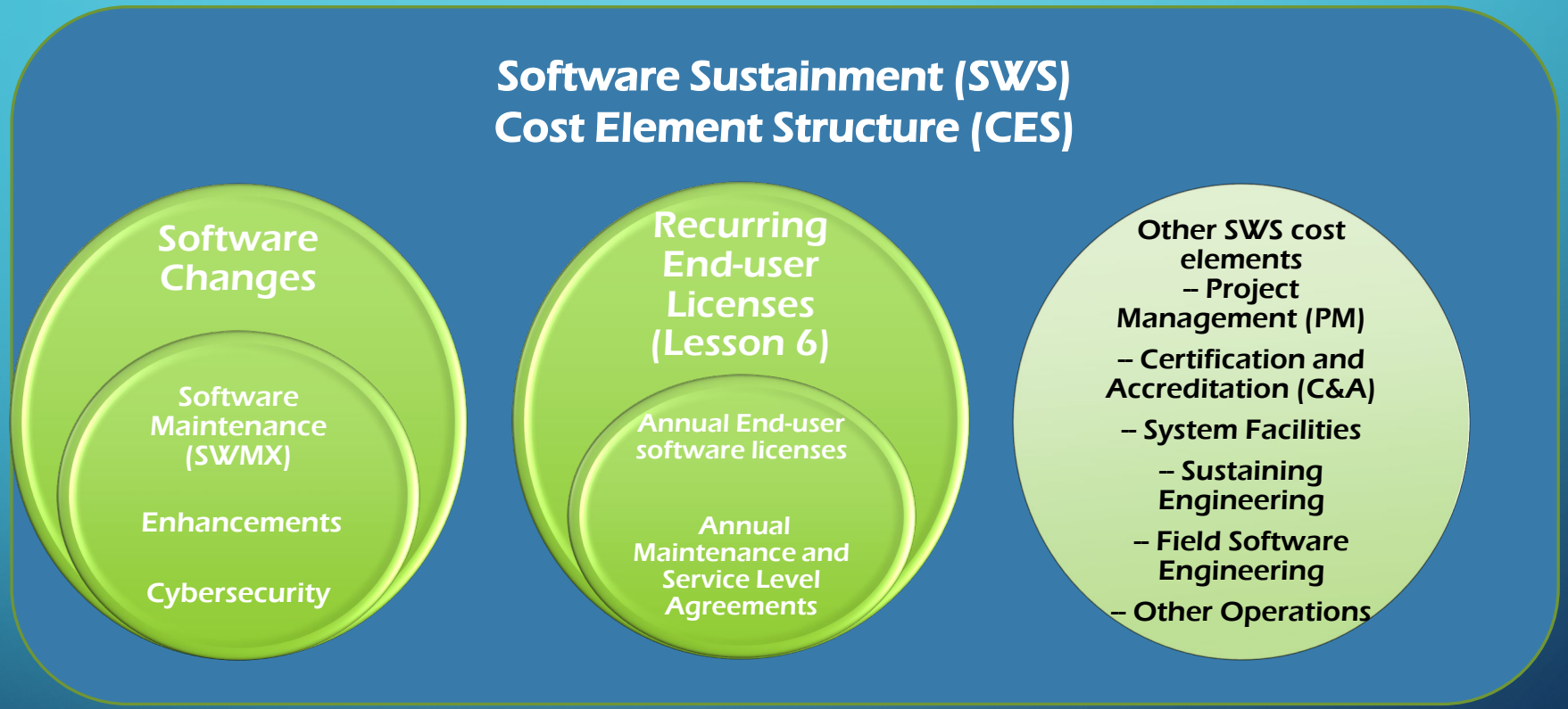
Milestone Event	Purpose of Software Cost & Schedule Estimate	Typical Questions Addressed in the Estimate
R&D feasibility study	Provide insights to support trade-off studies and Analysis of Alternatives (AoA); establish preliminary budget	What is the cost and performance trade off for the potential solutions? Is the program affordable to acquire/buy/develop and operate in the long term?
Development RFP Release or Internal Project Budget Established	Provide analysis to support approval of the release of the Request for Proposal (RFP) for the Software Development or Internal Project Funding. An independent cost estimate can be used to compare against the proposal responses	What is the likely cost and schedule of the software development effort? Why are the bid responses so varied and which ones have reasonable software estimates?
Contract award or Kickoff internal project with budget and schedule	Provide a reasonable cost and schedule estimate for executing development of the chosen solution; support establishment of the original software life cycle baseline for the program	Are the development and ongoing operations costs of the chosen solution consistent with budget and funding plans? Are there alternative procurement strategies that result in a more affordable and efficient program?

ROLE OF SOFTWARE ESTIMATING

Software life cycle focus: Sustainment

ACQUISITION EVENT	PURPOSE OF SOFTWARE COST & SCHEDULE ESTIMATE	TYPICAL QUESTIONS ADDRESSED IN THE ESTIMATE
During software development when there are scope changes	Update software estimate to reflect revised scope Compare actual performance of the projects to anticipated	What effect on size, effort, cost and schedule did the scope change(s) make? How well is the project performing? Is the project on budget and on schedule? What is the progress?
Initial deployment / installation	Update O&S cost estimates based on the system's current design characteristics, the latest deployment schedule, and the current training support plans	What technologies or strategies can be pursued to lower the ongoing sustainment and support costs?
Post-deployment	Continue to track sustainment costs and update sustainment cost estimates yearly throughout the program's life-cycle	Are preliminary information and assumptions on software sustainment costs still relevant and accurate? What are the variances from past estimates and what are the reasons for variances?

SOFTWARE SUSTAINMENT (SWS) COST ELEMENT STRUCTURE (CES)[1]



Adapted from the SWS WBS v5.0 from the New Army Software Sustainment Cost Estimating Results DASA-CE by Cheryl Jones, James Doswell, et al, presented at ICEAA May 2019

CEBoK-S

NEXT STEPS CEBOK-S CERTIFICATION

WWW.ICEAAONLINE.COM

- SCEC Certification (2-year experience requirement)
- Base costs →(payable to ICEAA):
  - $595 materials ($745 non-members)
  - $350 SCEC exam ($515 non-members)
  - $150 ICEAA 1-year membership
  - SCEC Exam → 60 Multiple Choice exam
- Licensed training providers (trainer must be SCEC)
  - Logapps
  - Quality Plus Technologies, Inc.
  - Others

CEBoK Module 12 Software Cost Estimating Agenda/Outline

1. Cost drivers for software development
2. Measures of Size
3. Measures of Productivity
4. The Constructive Cost Model II (COCOMO II) Basic Estimating Equation
5. The CEBoK-S Basic Estimating Equation
6. “Backfiring” and Associated Considerations
7. The Analogy Technique, with a borrowed “productivity factor”
8. Some data sources
9. Software Sustainment
10. Additional topics (time-permitting)

A FINAL NOTE

'The software industry has the worst metrics and measurement practices of any industry in human history -- Capers Jones

Capers Jones, *Quantifying Software – Global and Industry Perspectives*, 2018

CEBoK-S will help create realistic data-based estimates. Over time, this will lead to more successful projects… and (hopefully) better metrics. -- Carol Dekkers

Carol Dekkers

+1 813-816-1329

caroldekkers@gmail.com

www.qualityplustech.com

CEBoK-S

www.iceaaonline.com