Software Cost Estimating: Friend or Foe (to Agilists)?

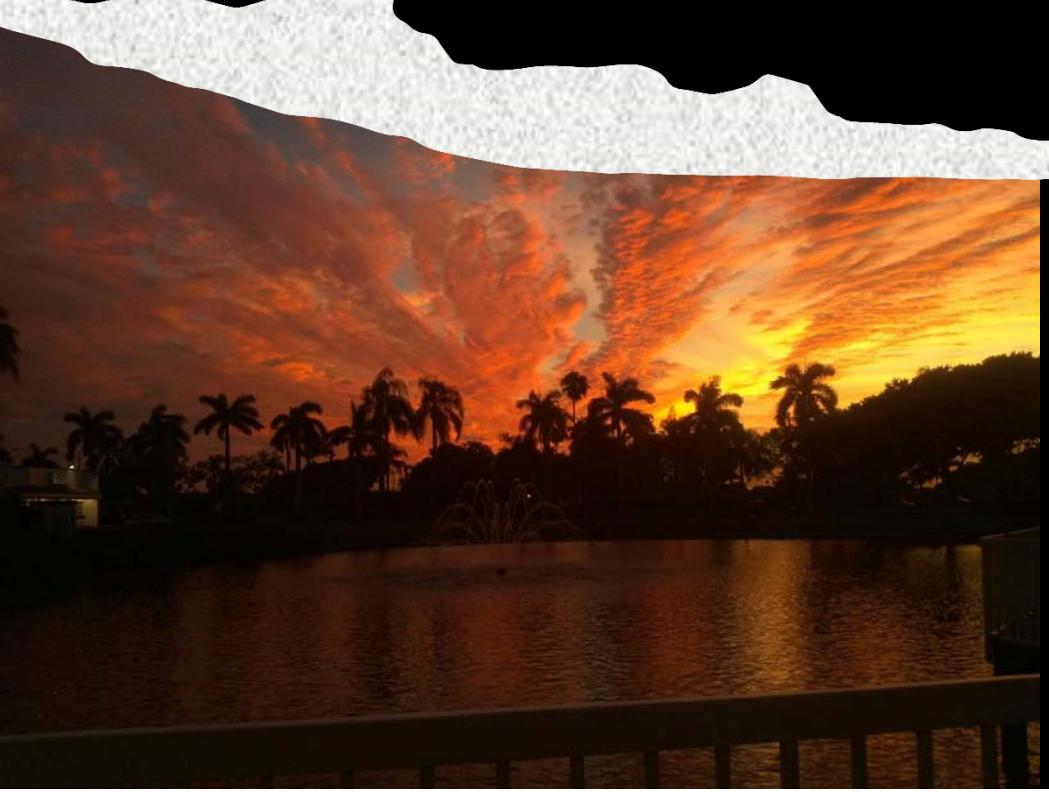
Carol Dekkers, PMP, CFPS (Fellow), P.Eng.
Quality Plus Technologies, Inc.
Dekkers@qualityplustech.com

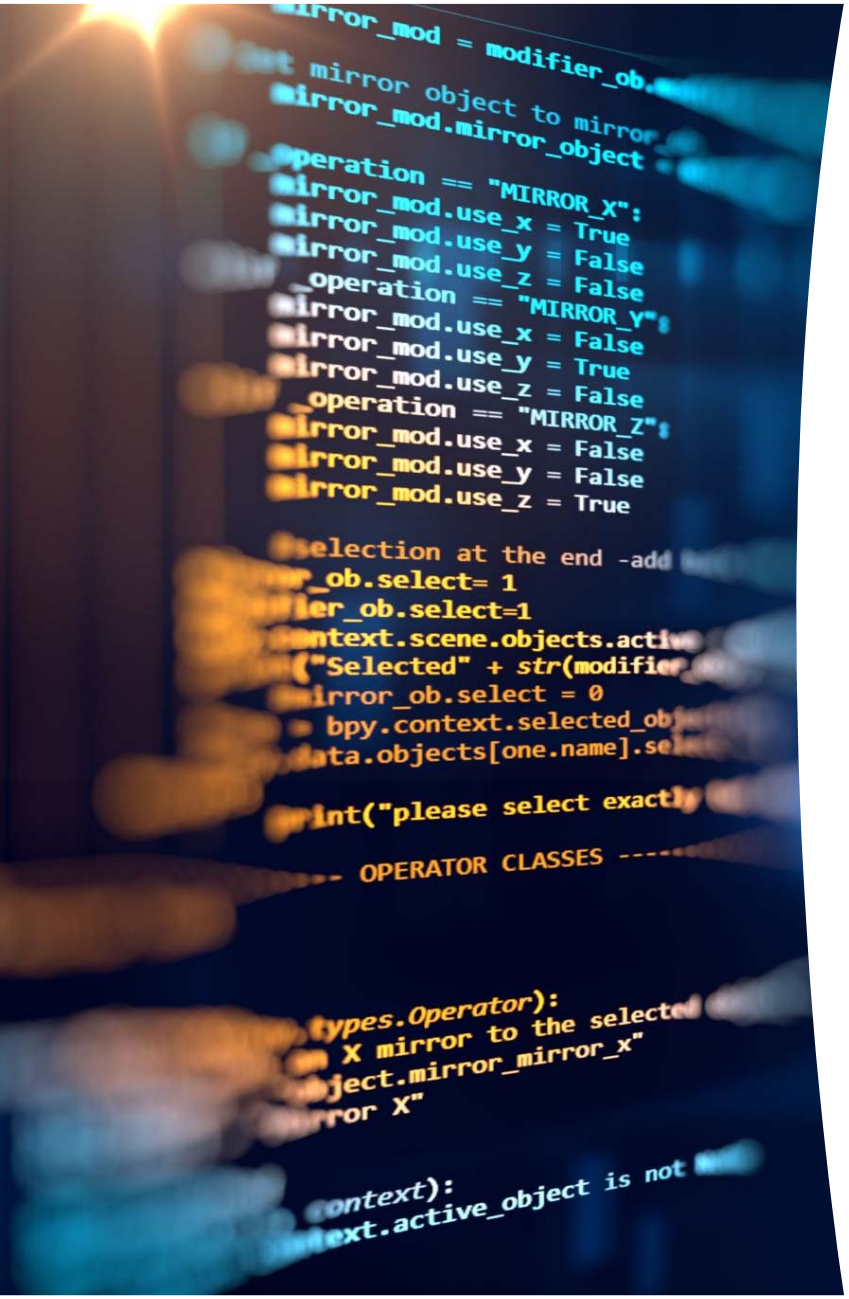
WHO AM I?

Carol Dekkers, PMP, CFPS (Fellow), CSM, P.Eng.

Lead author of the International Cost Estimating and Analysis Association (ICEAA)
Software Cost Estimating Body of Knowledge (CEBOK-S)

ICEAA 2022 Educator of the Year
Past-president, International Function Point Users Group (IFPUG)
ISO project editor for software engineering standards since 1994
Founder, Quality Plus Technologies, Inc.
Global consultant, published author, thought leader, speaker

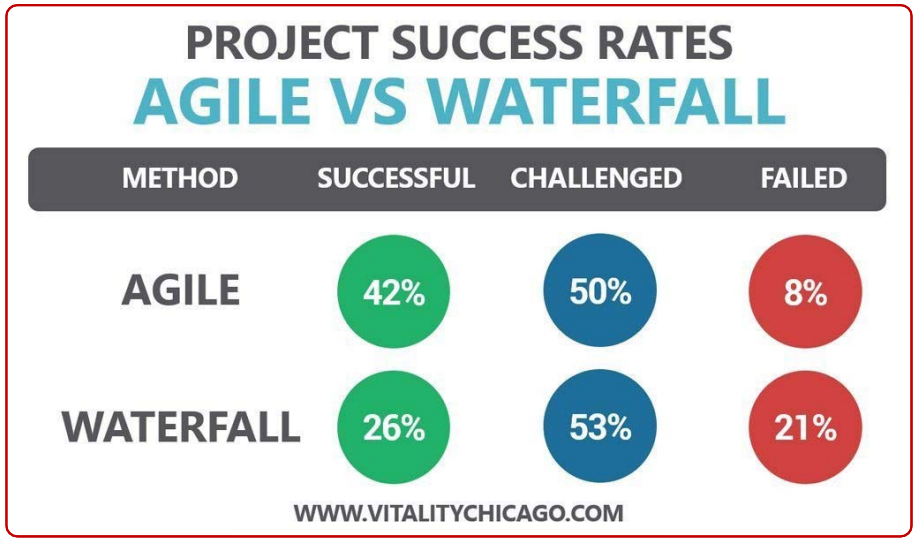




What is the current state of software development?

Software Development Projects: Status Quo

Software Project Success[1]



Software Project Growth[2]

	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 Occurrence in Most Projects (1 in 6)

1. Standish Group *CHAOS report* (2015). <Note: similar percentages in 2021>
2. Dr Christian Smart, *Solving for Risk Management: Understanding the Critical Role of Uncertainty in Project Management* (2021)

Top 10 Reasons for Project Challenges & Failures

Cause	Customer	Supplier	Comment / Solution
1. Poor user input	X	X	Training, time
2. Stakeholder conflicts	X	?	Project Management
3. Vague requirements	?	?	Terminology
4. Poor cost and schedule estimation	**?**	**X**	**Estimates overly-optimistic, risk (avoidance)**
5. Skills that do not match the job	X	X	Training
6. Hidden costs of going "Lean and Mean"	**X**	**X**	**Unrealistic goals and resources**
7. Failure to plan	?	?	Structure, PM
8. Communication breakdowns	X	X	Blame (He said, she said)
9. Poor architecture		X	Planning
10. Late "failure" warning signals		X	Measurement

Source: Loren May, CrossTalk
http://info.psu.edu.sa/psu/cis/biq/se501/a/a1/MajorCausesofSoftwareProjectFailures.pdf

What is the Impact of Unrealistic* Estimates?

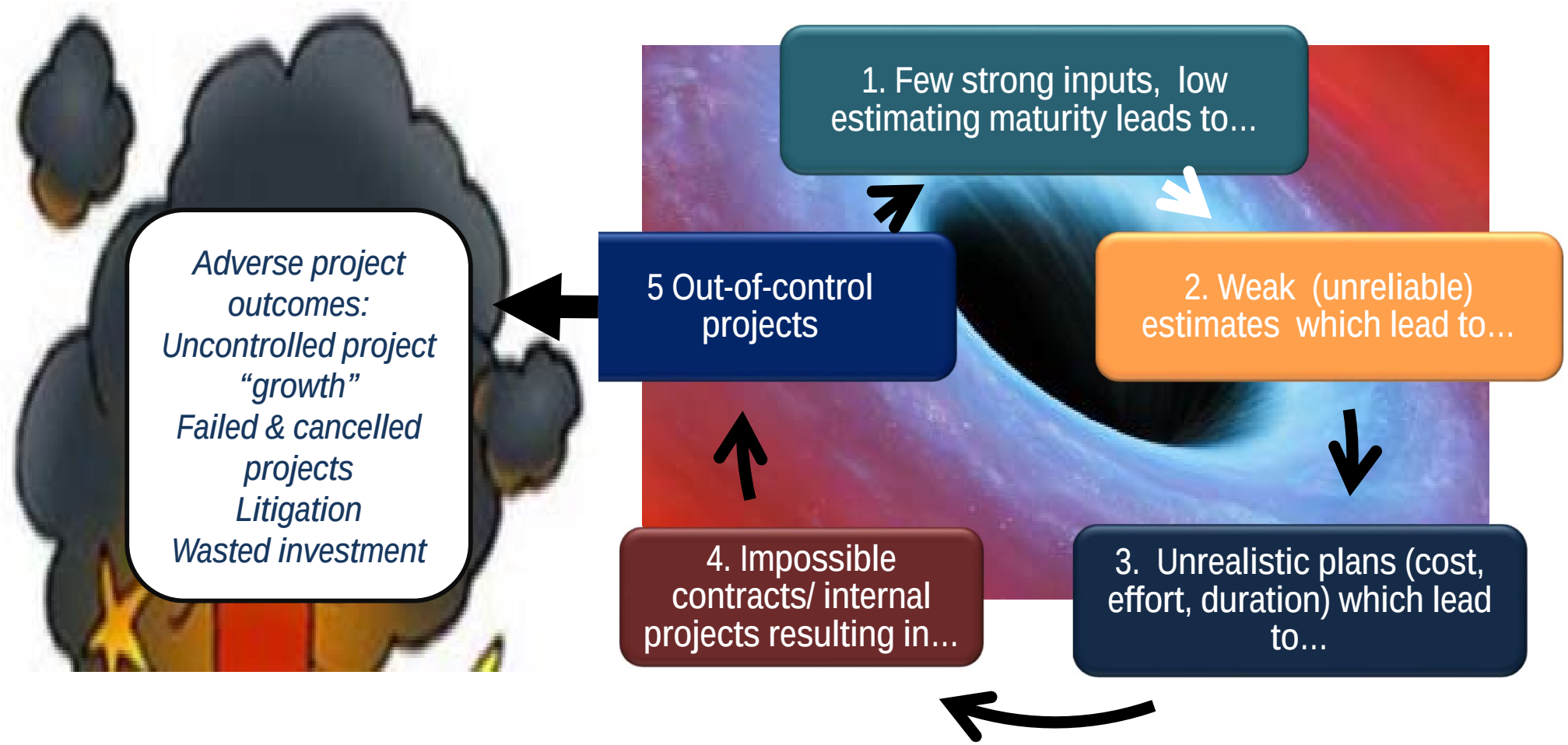
Standish Group on U.S. government / business:

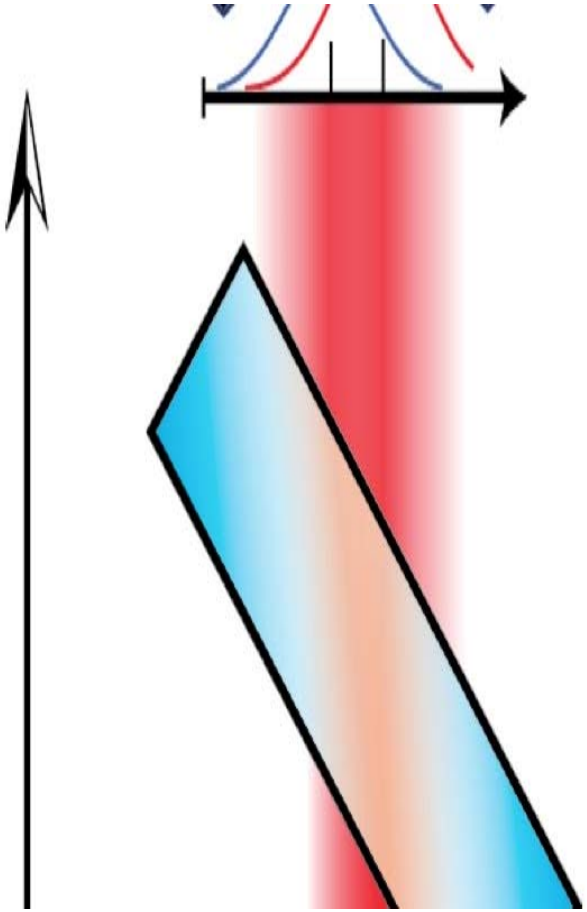
~ $81 B USD = canceled software projects

~ $59 B USD = budget overruns

*** Unrealistic estimates → Overly-optimistic (Cost unreasonably low, and duration too short) and Overlooks (avoids) risks**

Unrealistic estimates lead to poor project outcomes





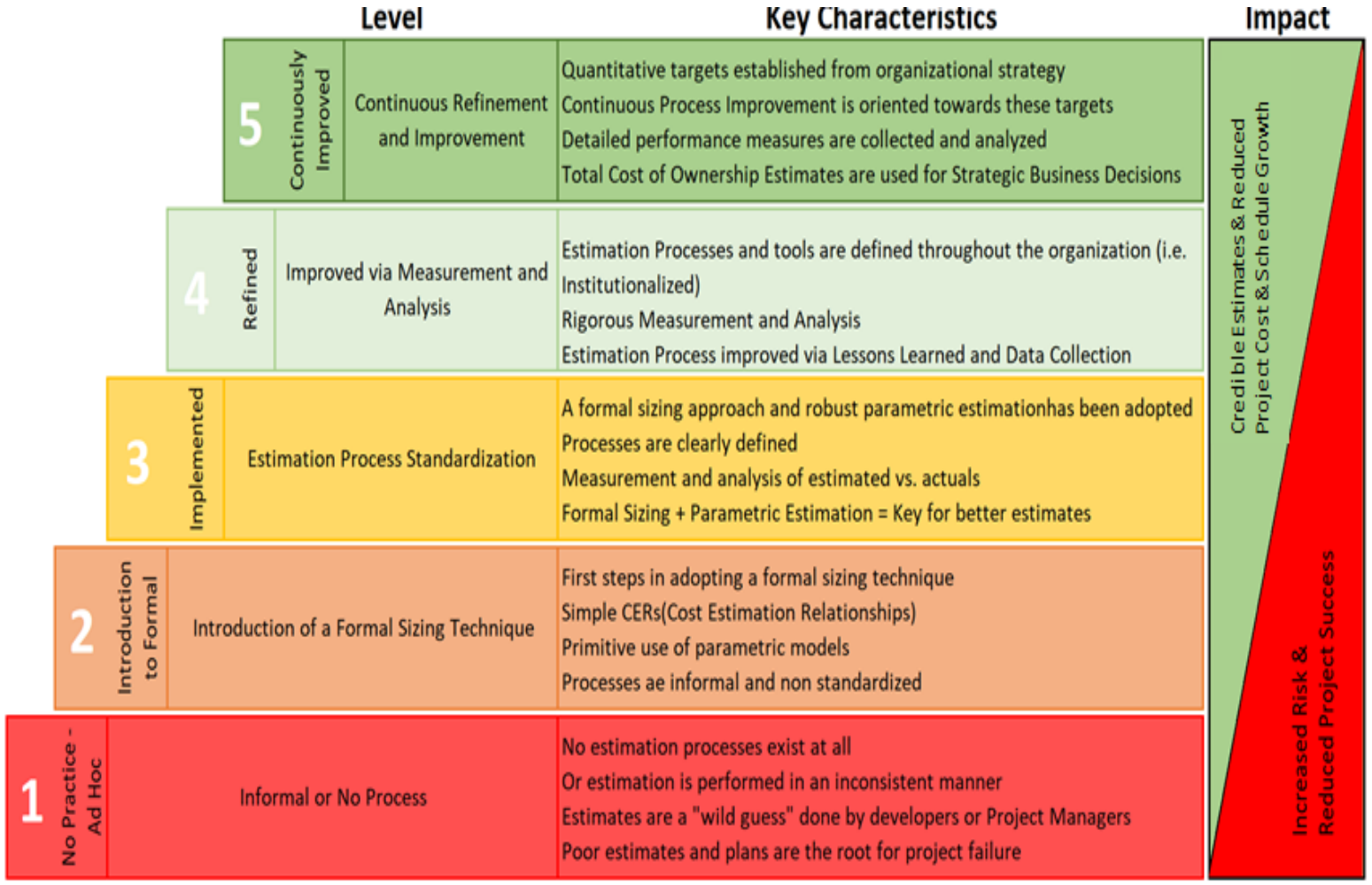
Okay, but isn't agile development (by its nature) unpredictable? Why bother estimating... ?

10 Lessons Learned about Software Cost Estimating (in Agile)

1. Historical data based estimates are more reliable than theory (or guesses)

- **Data analysis** is important (data must be similar, relevant, comparable)
- **Data normalization** is critical (units of measure, scope, who, what, OT)
- **Historical data is valuable** → tells a factual story (CER, SER)

2. Cost estimating Maturity Model

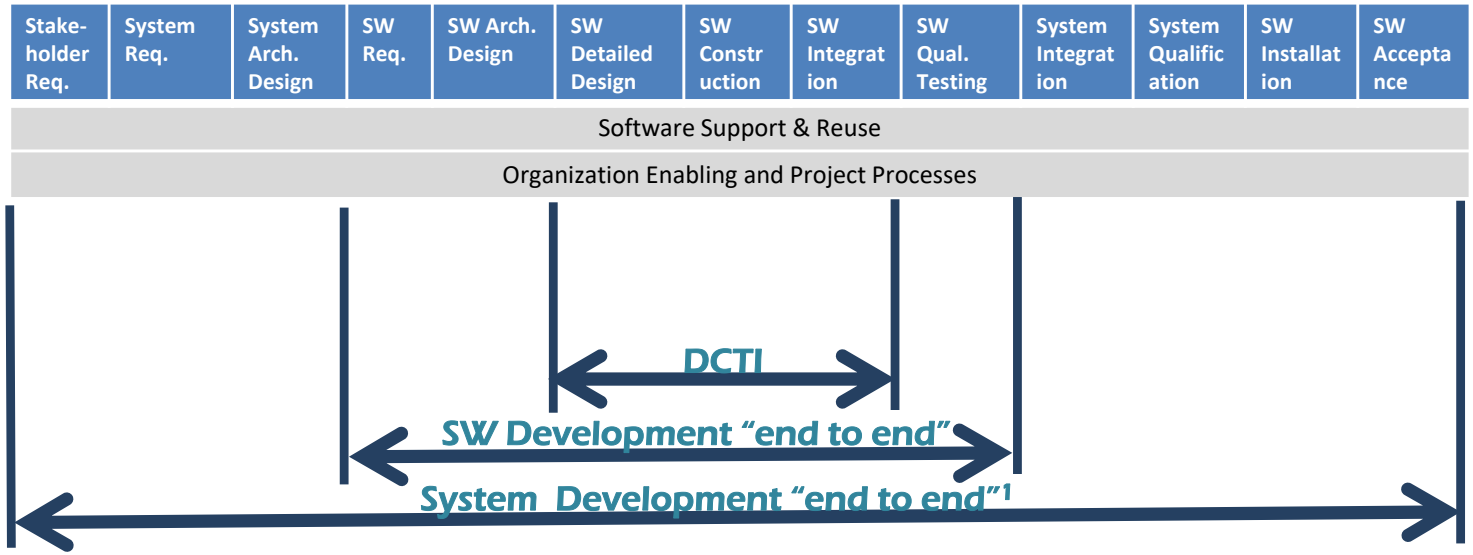


Many companies do not follow formal estimating practices (Level 1)

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

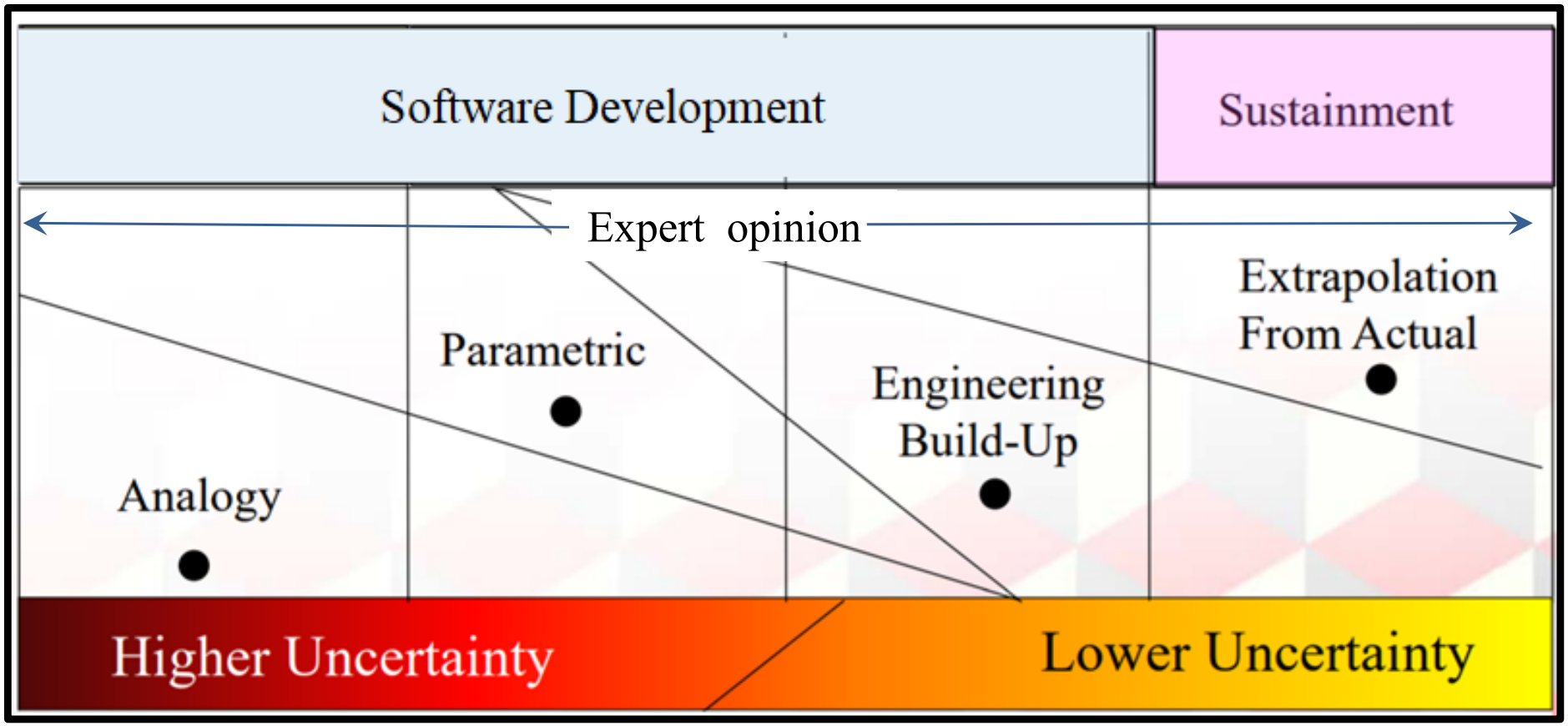
3. Estimation Scope is critical: Range of software activities included

- Design, Code, Test, Integration (DCTI) factors cover the "core" parts of the process; other activities must be estimated separately or significant omissions will occur in the estimate
- Software "end to end" productivity attempts to cover all "software-specific" activities; higher level systems engineering activities must be estimated separately where relevant



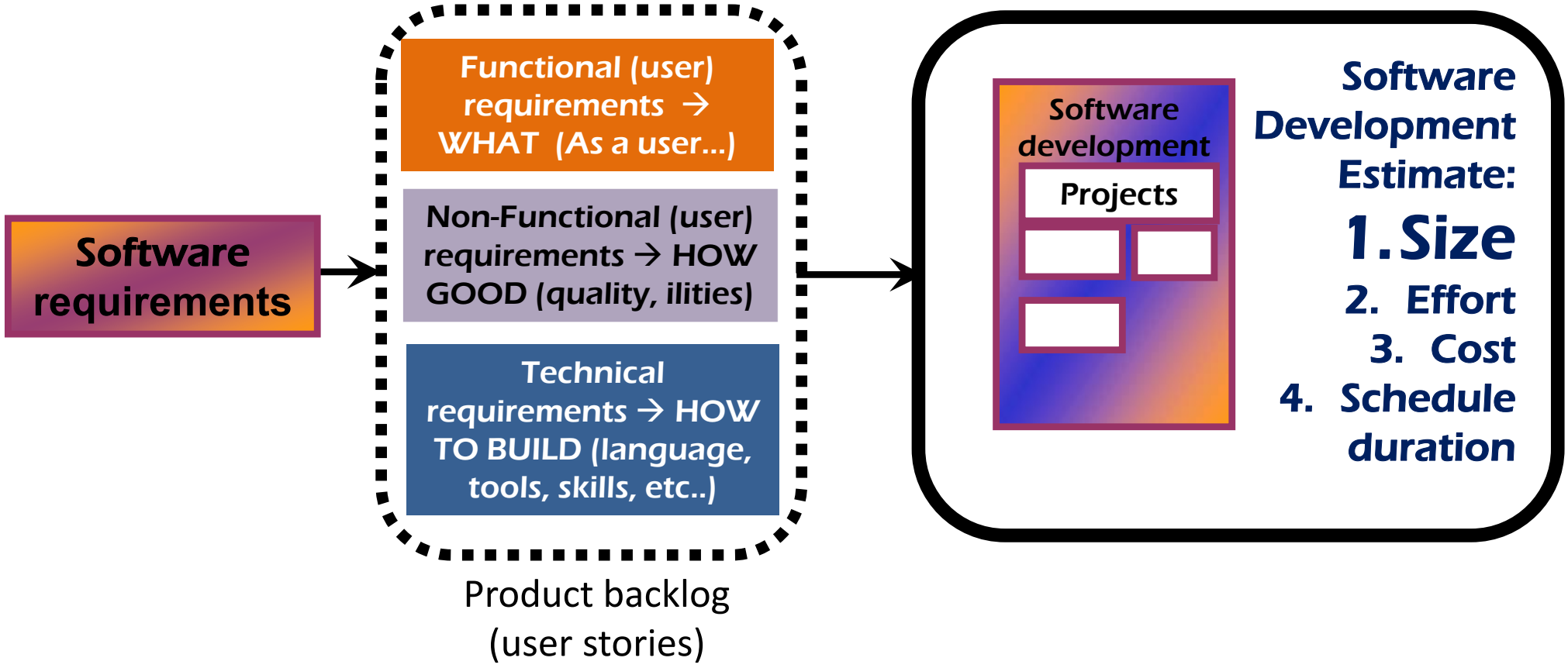
Source: ICEAA CEBOK-S Draft (2021)

4. There is no "one size fits all" estimating approach



Source: ICEAA CEBOK-S Draft (2021)

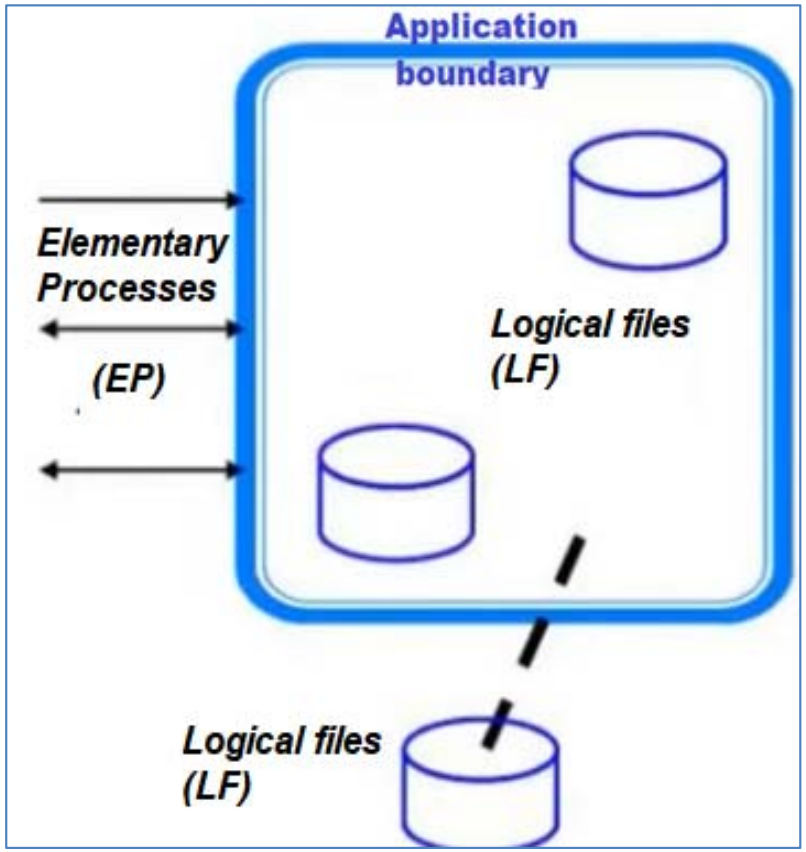
5. Quantifying software size is fundamental to a good estimate



Source: Quality Plus Technologies, Inc.

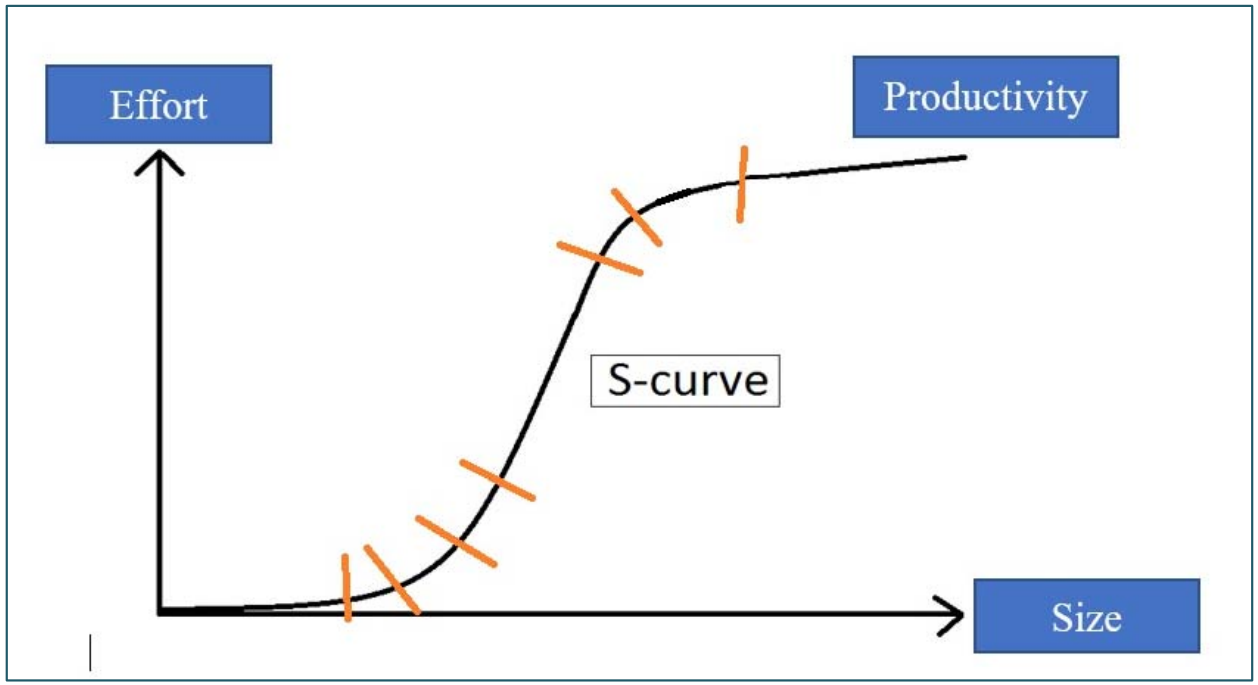
6. Estimate the product backlog using functional size (Simple FP or FP)

- Simple Function Points is standardized (International Function Point Users Group SFP v2.1 method)
- Size based on number of Logical Files (LF = 7 SFP each) and Elementary Processes (EP = 4.6 SFP each)
- Examples:
  - CRUD + entity = 4 EP + 1 LF = 25.4 SFP
  - 10 unique reports = 10 EP = 46 SFP
  - Data sent from other system = 1 or more EP
  - Data sent to other system = 1 or more EP



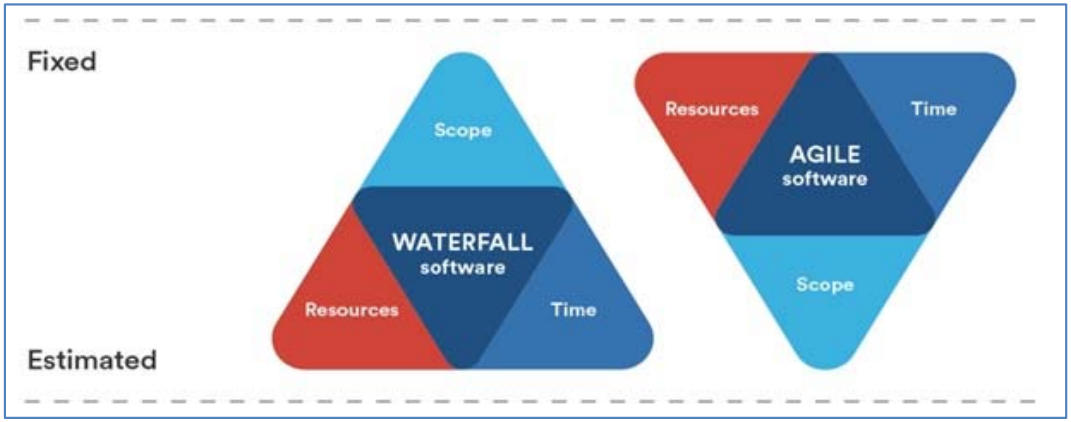
7. Software development is subject to Diseconomies of Scale (Exp > 1)

Effort = Size^Exp * 1/Productivity



16

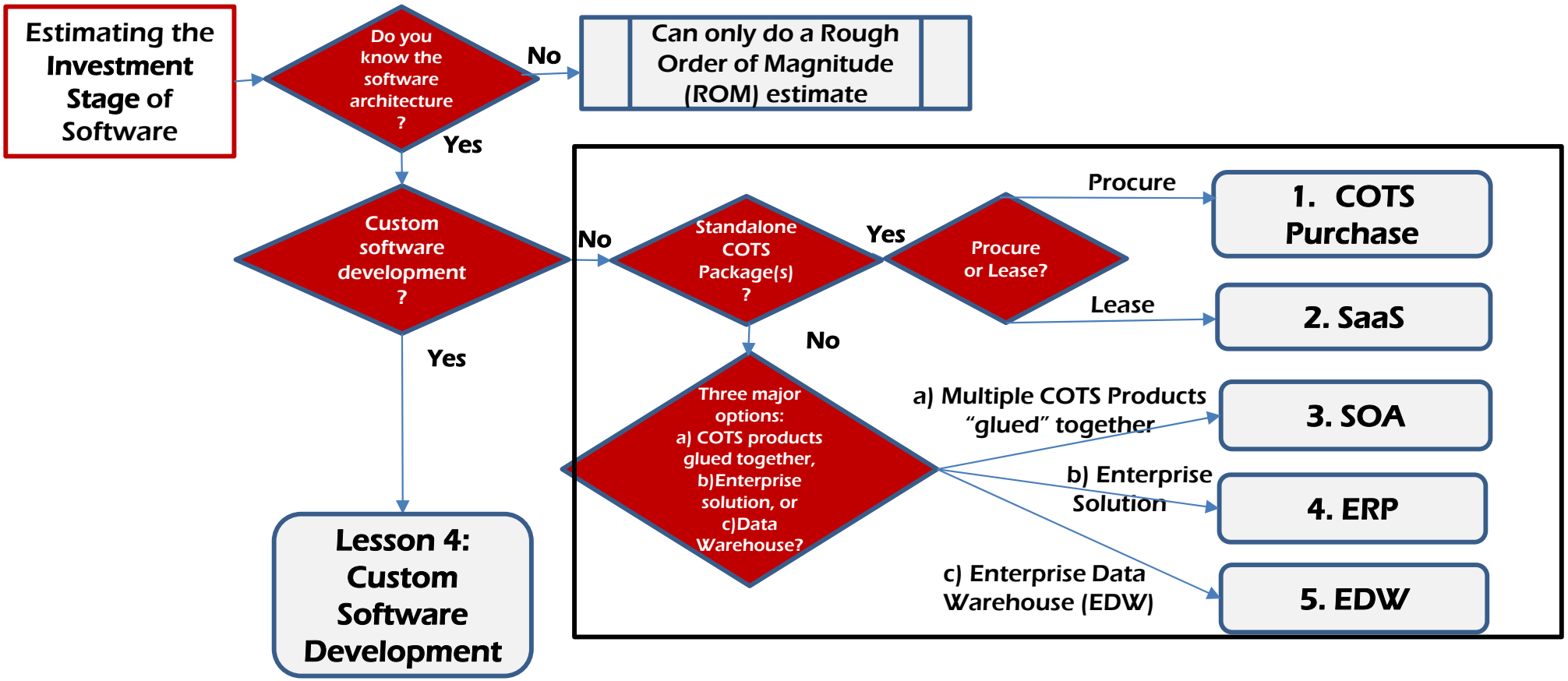
8. Agile versus Waterfall: different cost (estimating) considerations



Adapted from
https://www.process.st/waterfall-vs-agile/

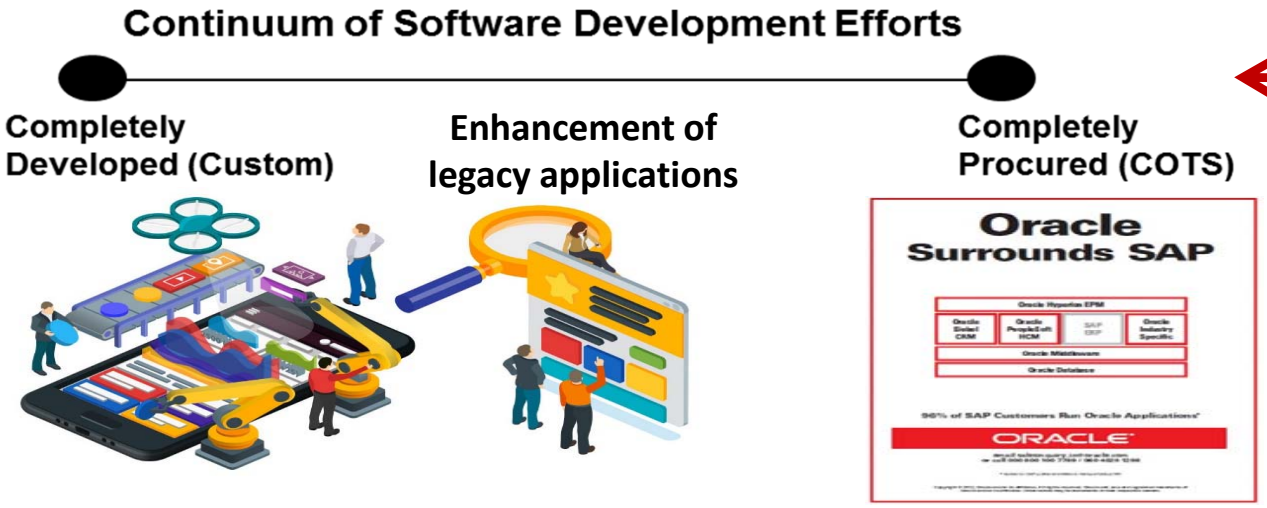
Topic	Agile	Predictive / waterfall	Hybrid-agile (Partly predictive/partly agile) TYPICALLY GOV"T
Fixed variables	Cost & schedule	Scope (requirements)	Initial high-level scope fixed, but is flexible to prioritization and change during development
Estimated	Scope (features)	Cost & schedule	Cost & schedule initially estimated, but becomes fixed during development
Driver	Change-driven	Plan-driven	Hybrid
Development risks[1]	Cost & schedule mostly fixed. Delivered size may fall short	Scope fixed → cost & schedule overruns	Cost & schedule mostly fixed during build. Delivered scope may fall short

9. Hybrid software solutions are Non-trivial: Development vs Procurement



Source: ICEAA CEBOK-S Draft (2021)

10. Continuum of Development vs Procurement



Software development paradigms

Configuration, licensing, integration, customization

All software efforts (with or without system activities) lie on a continuum of completely developed (custom) to completely procured[1]

Procured software involves effort to configure, integrate, and customize (see Lesson 6)

1 Commercial Off-the-Shelf (COTS) and Information System (IS)/Business Systems software (packages) are covered in Lesson 6
This slide is used here to depict where software development paradigms fit in the context of software development

Agile and Cost Estimation

- Poor cost and schedule estimates can doom a project or initiative from the “get go”
- ***Software*** cost and schedule estimates are especially challenging to cost estimators, **and** the team (TLA’s, types of requirements)
- Communication between the two disciplines (cost estimating and development) can lead to far better outcomes
- Software cost estimation is a professional endeavor (ICEAA CEBOK-S)

Draft: Software Cost Estimation Body of Knowledge CEBOK-S

ICEAA International Cost Estimating and Analysis Association

LESSON 0: INTRODUCTION TO CURRICULUM

LESSON 1: IMPORTANCE AND MOTIVATION FOR CEBOK-S

LESSON 2: SOFTWARE DEVELOPMENT PARADIGMS

LESSON 3: CEBOK-S 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

Builds on Pre-requisite Knowledge: ICEAA CEBOK Modules

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

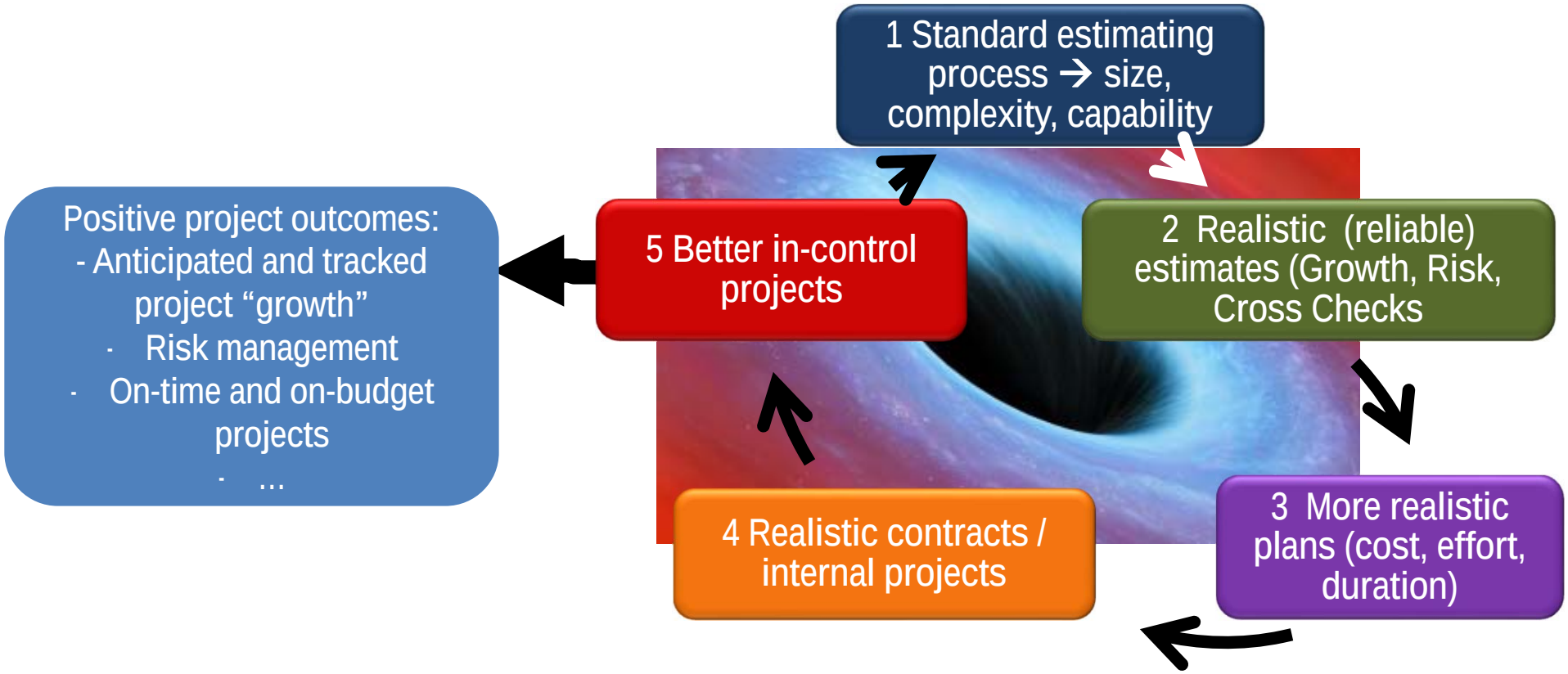
3 Unit III - Analytical Methods
- 3.1 Module 6 - Basic Data Analysis Principles
- 3.2 Module 7 - Learning Curve Analysis
- 3.3 Module 8 - Regression Analysis
- 3.4 Module 9 - Cost and Schedule Risk Analysis
- 3.5 Module 10 - Probability and Statistics

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**

Data-based, realistic agile development estimates → positive project outcomes



A Final Note…

'The software industry has the worst metrics and measurement practices of any industry in human history' – Capers Jones (2018)[1]

"Size- and data-based software estimates are the key to better project outcomes, and in time, better metrics." – Carol Dekkers, Dec 2021

1. Source: Capers Jones, **Quantifying Software – Global and Industry Perspectives**, 2018

Carol Dekkers

813-816-1329

caroldekkers@gmail.com

www.qualityplustech.com

CEBOK-S:

www.iceaaonline.com