

Simple is Not Necessarily Better: Why Software Productivity Factors Can Lead to Bad Estimates

2007 ISPA-SCEA Joint Conference
New Orleans, LA

Michael A. Gallo
Paul Hardin

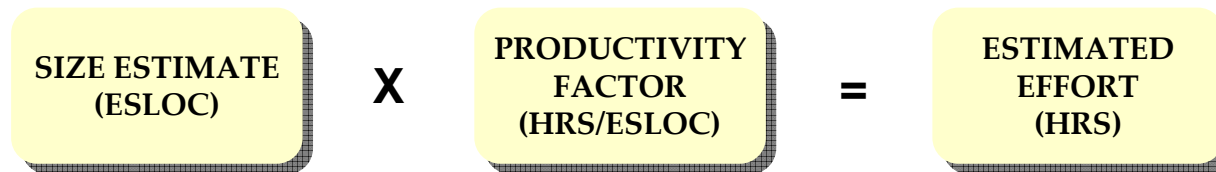
201 12th Street South
Suite 612, West Tower
Arlington, VA 22202
Voice (703) 412-0602

5290 Overpass Road
Suite 206
Santa Barbara, CA 93111
Voice (805) 964-9894

The problem...

- The use of simple productivity factors to estimate software development cost injects unnecessary risk into the program.

SOFTWARE COST ESTIMATING USING THE PRODUCTIVITY METHOD



- ESLOC (Equivalent New Source Lines of Code) is a weighted sum of New, Modified, Reused, etc. The weights used for modified and reused, etc are typically less than 1 which implies that the cost of this code is less than the cost of new code.
- In almost all cases, $ESLOC \leq Delivered\ SLOC\ (DSLOC)$

- Based on either an analogy to a similar completed project development or based on an average of productivities of several analogous projects
- Frequently reflects only 'core' software development activities (Design, Code, Unit Test)
- Method for computing ESLOC may differ from method used to compute estimated ESLOC

- Estimated effort will reflect the set of activities included in the productivity factor
- Additional activities (e.g. Requirements, System Integration & Test, etc) are estimated as a factor of the 'core' software development estimate

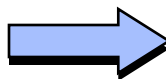
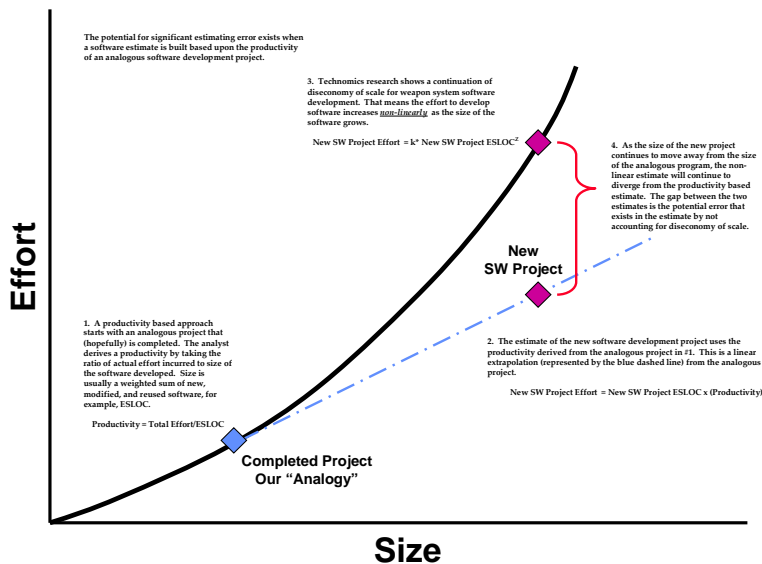
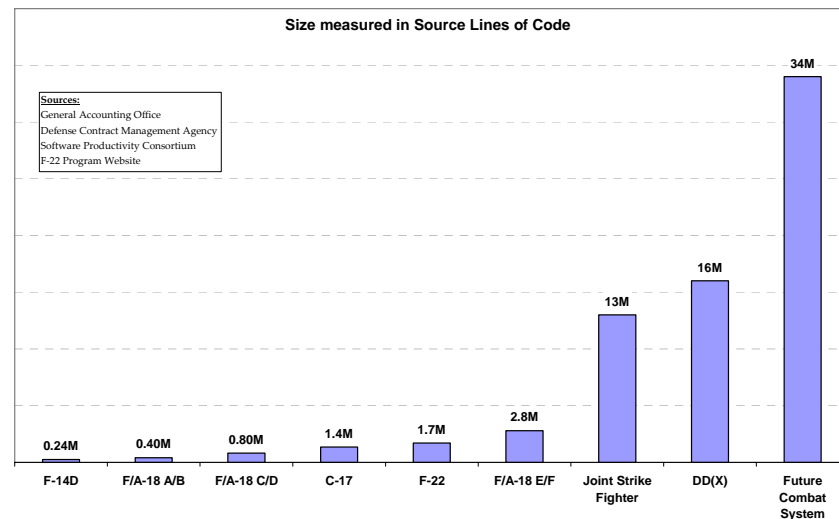
Risks Associated With Productivity Factors

1. Linear extrapolation fails to account for diseconomies of scale
2. Error can be exacerbated when the estimate is treated discretely rather than as a whole system
3. May neglect to properly count ESLOC for incremental developments
4. Leads to erroneous use of adjustment factors to account for missing software development activities

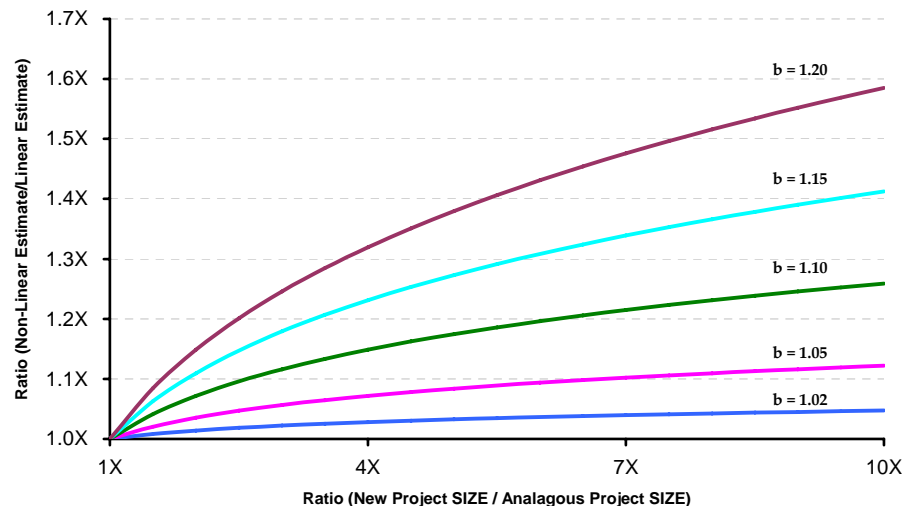
Risk #1: Linear Extrapolation

- Diseconomies of scale remains prevalent in DoD software projects
 - Effort = $a * \text{Size}^b$; $b > 1$
- Productivity methods use a linear relationship
 - Effort = $a * \text{Size}^1$
- Impact: a potentially large underestimation of effort when size of project is substantially larger than its analogy

Trend in DoD Weapon System Software Size



Potential Error Arising from the Use of a Linearly Extrapolated Productivity Analogy to Estimate SW Development Effort



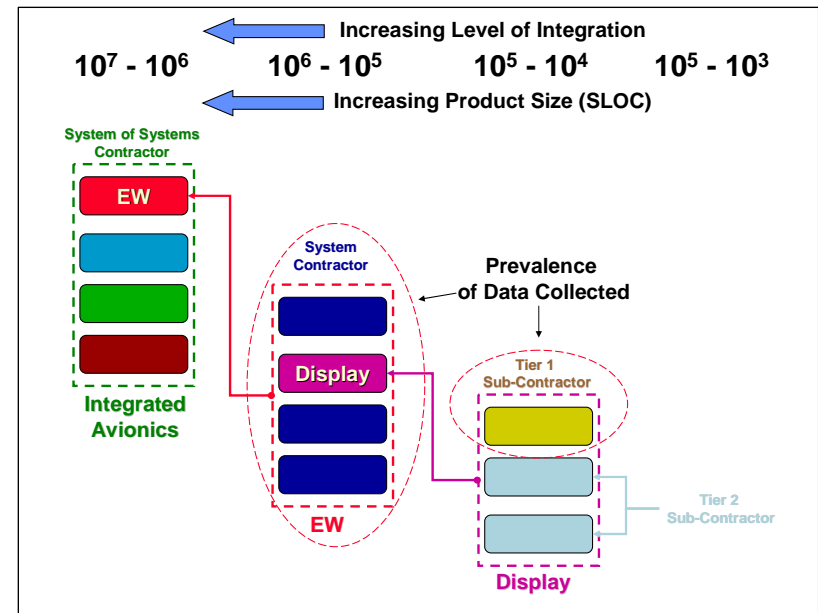
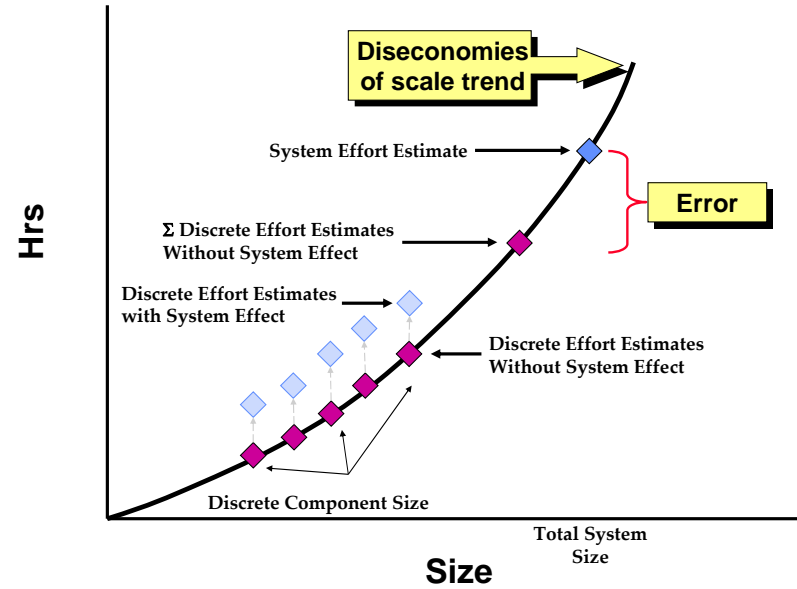
Simple productivity factors fail to reflect diseconomies of scale

Risk #2: Ignoring System Effects

- The whole (i.e. system) is greater than the sum of its parts (i.e. components and sub-systems)

Existing DoD databases have perpetuated this problem

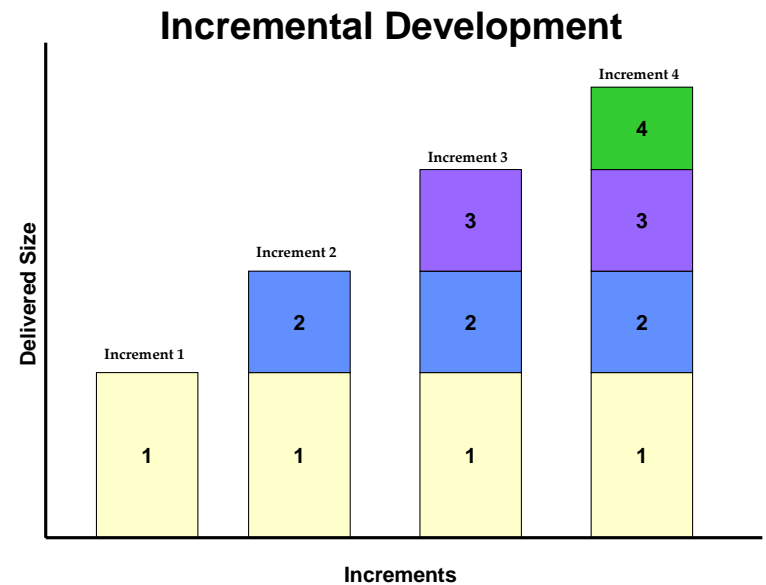
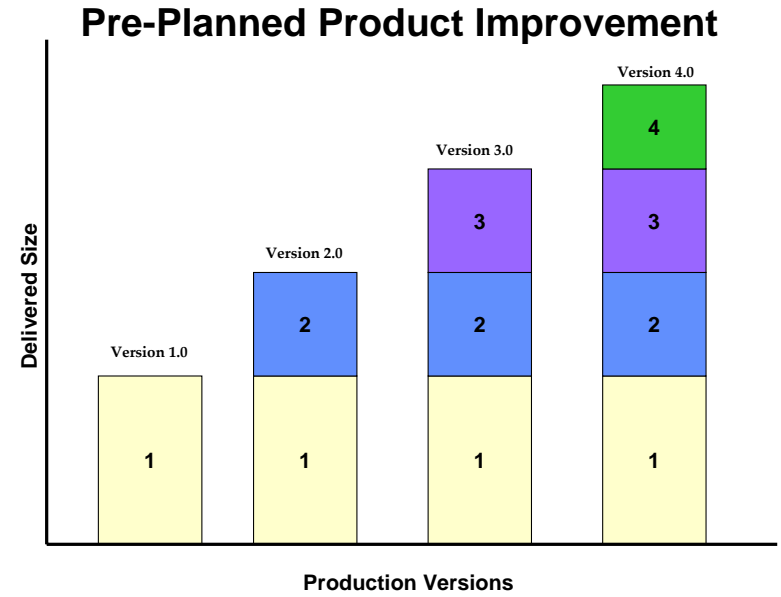
- Systems are broken up and sanitized to protect proprietary data
- Little if any insight into the number of development organizations by system
- Few (if any) total system data points



Current DoD data and analyses miss the 'big picture'

Risk #3: Incremental Development

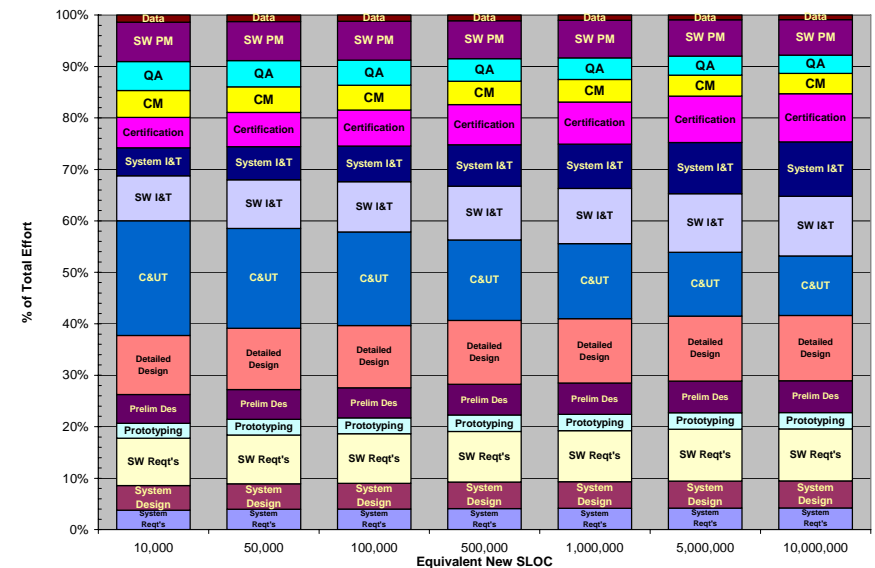
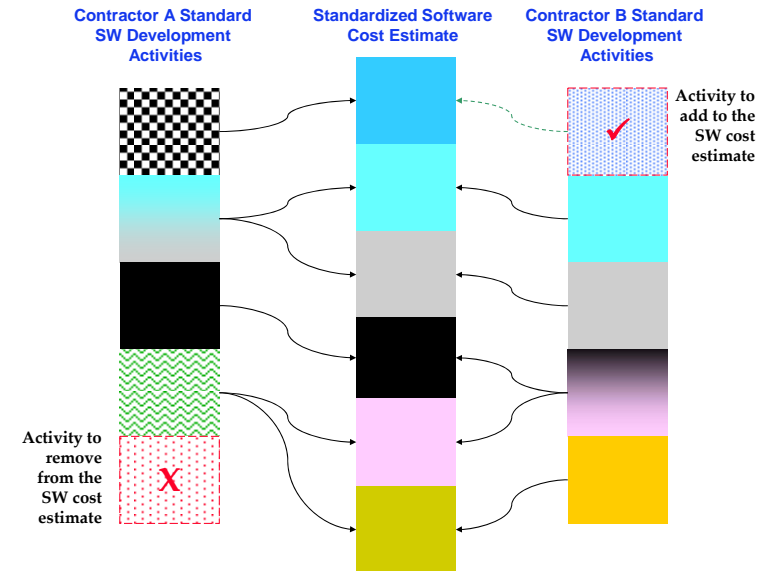
- Incremental development looks suspiciously like pre-planned product improvement
 - Successive deliveries of software are built upon pre-existing base of software
 - Cost to build, integrate and test the latest product build is a function of the product's cumulative size
- Estimates should incorporate pre-existing base in the ESLOC computation



ESLOC computations must include cumulative reuse

Risk #4: Flawed Adjustment Factors

- Developer's definition of software effort may not align with cost analyst's standardized definition of effort
- It is common practice to apply simple (fixed) factors to add or remove software development activities
- However, distribution of activities changes as the size of the software changes – *Result: too much (or too little) addition or removal of effort*



Application of fixed factors increases estimating error

How to Mitigate These Risks

Problem	Our Approach
Linear Extrapolation	Use proper parametric estimating relationships
System Effect	Derive equations at the system level; specify equations below system level
Incremental Sizing	Include 'Carryover' in ESLOC computation
Flawed Adjustment Factors	Use parametric estimating relationships to add/remove activities

A new Technomics model (VERA) implements these approaches

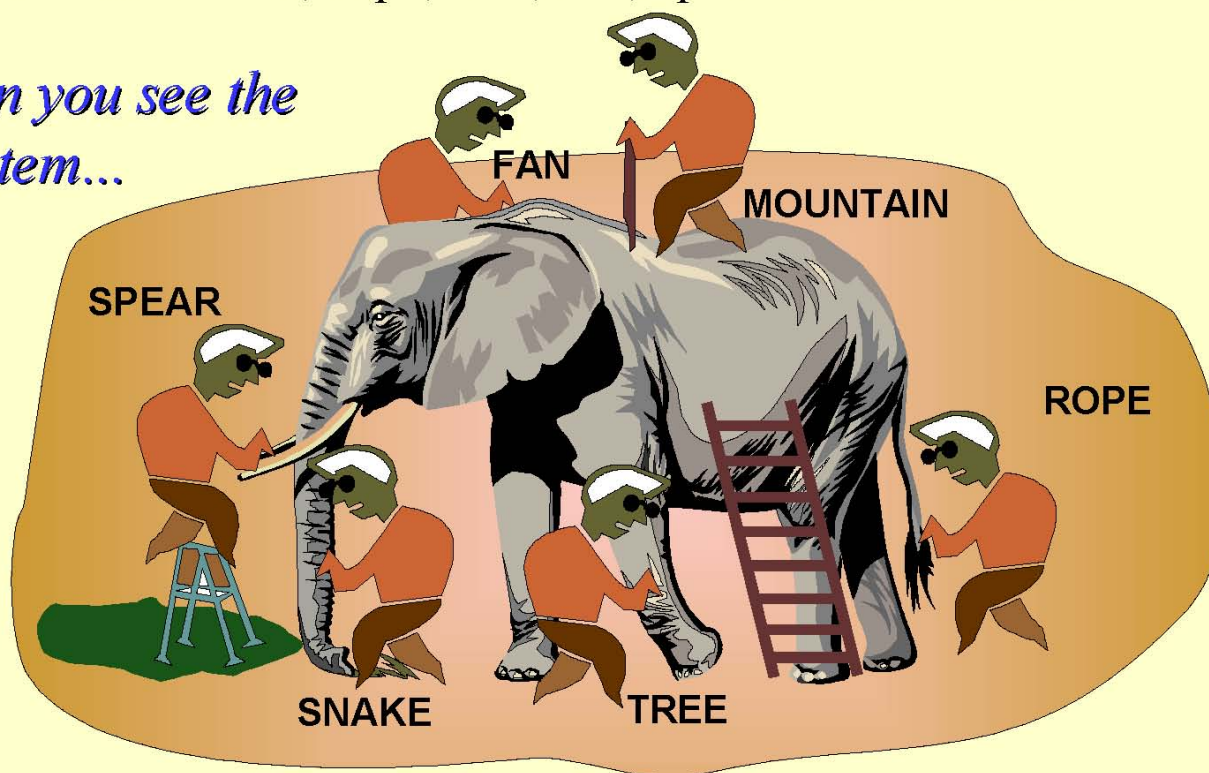
Final Thought

Source: "Engineering of Systems for Navy Interoperability", Version 5.8 dated April 2003 by Eric Honour, Honourcode, Inc.
<http://jax.acqconf.com/Presentations/Power%20Points/SE-Interop.pdf>

System Thinking

The 6 blind wise men think the elephant is a set of components consisting of a mountain, rope, tree, fan, spear and snake . . .

But when you see the total system...



... AN ELEPHANT IS AN ELEPHANT

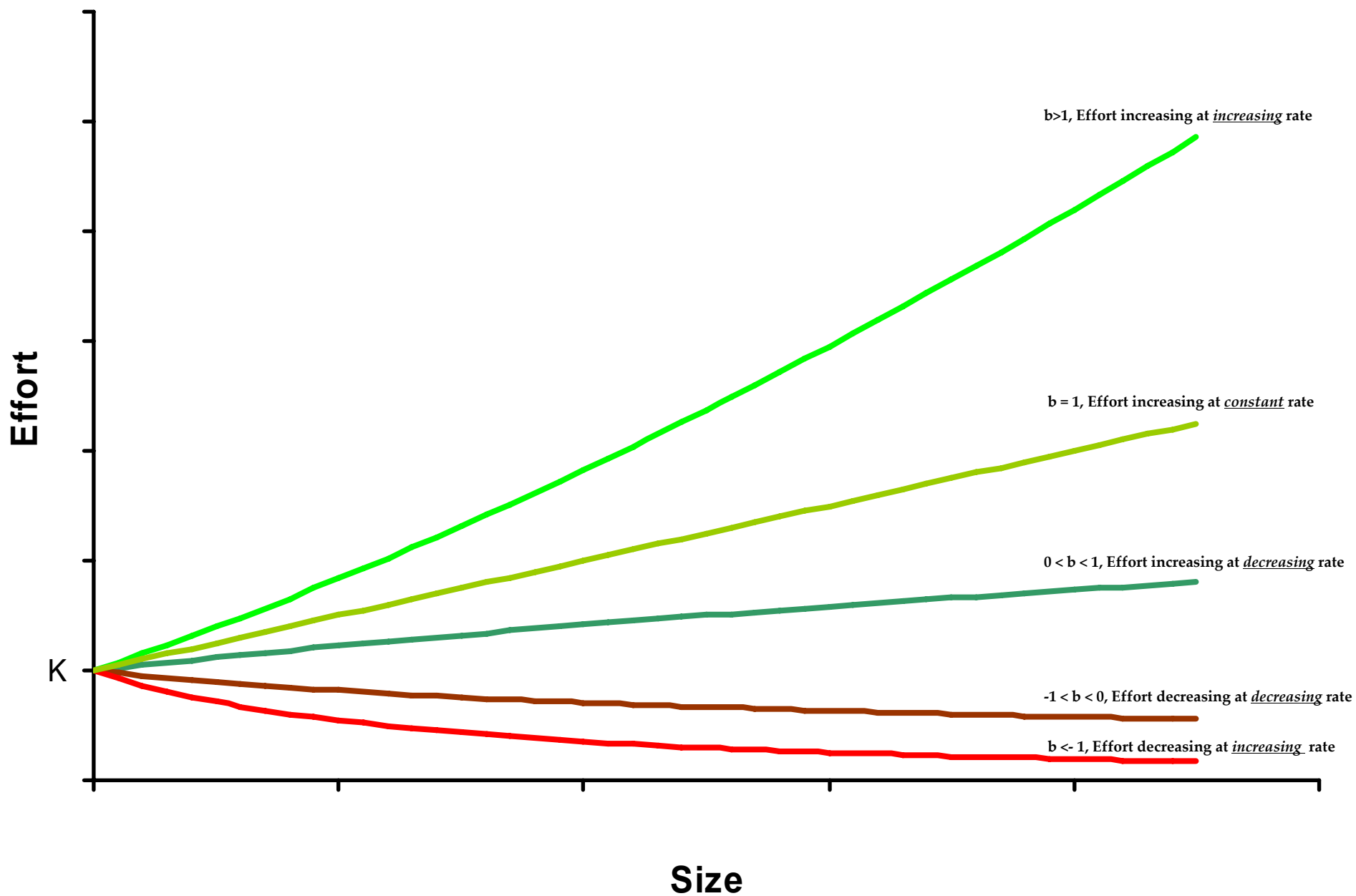
Engineering of Systems for Navy Interoperability

Technomics POCs

- Mike Gallo, Senior Cost Analyst
Phone (571) 366-1405
E-mail: mgallo@technomics.net
- Paul Hardin, Senior Cost Analyst
Phone: (571) 366-1407
E-mail: phardin@technomics.net
- Fax: (703) 412-0600
Web Site: www.technomics.net

Hyperlink Graphics Slides

Graph of Effort = $K * SIZE^b$



Effort

The potential for significant estimating error exists when a software estimate is built based upon the productivity of an analogous software development project.

3. Technomics research shows a continuation of diseconomy of scale for weapon system software development. That means the effort to develop software increases *non-linearly* as the size of the software grows.

$$\text{New SW Project Effort} = k * \text{New SW Project ESLOC}^z$$

4. As the size of the new project continues to move away from the size of the analogous program, the non-linear estimate will continue to diverge from the productivity based estimate. The gap between the two estimates is the potential error that exists in the estimate by not accounting for diseconomy of scale.

1. A productivity based approach starts with an analogous project that (hopefully) is completed. The analyst derives a productivity by taking the ratio of actual effort incurred to size of the software developed. Size is usually a weighted sum of new, modified, and reused software, for example, ESLOC.

$$\text{Productivity} = \text{Total Effort} / \text{ESLOC}$$

New SW Project

2. The estimate of the new software development project uses the productivity derived from the analogous project in #1. This is a linear extrapolation (represented by the blue dashed line) from the analogous project.

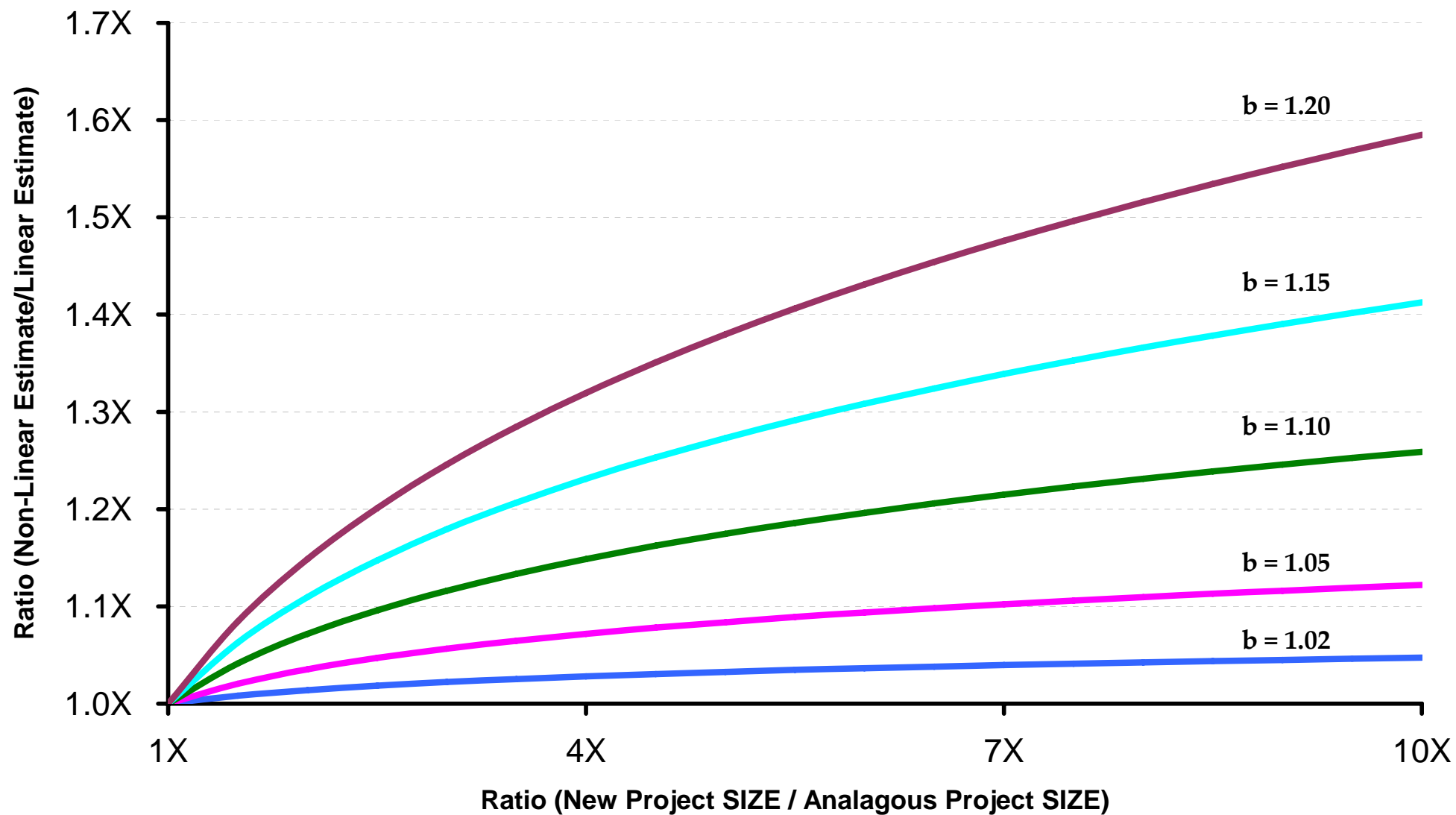
$$\text{New SW Project Effort} = \text{New SW Project ESLOC} \times (\text{Productivity})$$

Completed Project
Our "Analogy"

Size



Potential Error Arising from the Use of a Linearly Extrapolated Productivity Analogy to Estimate SW Development Effort

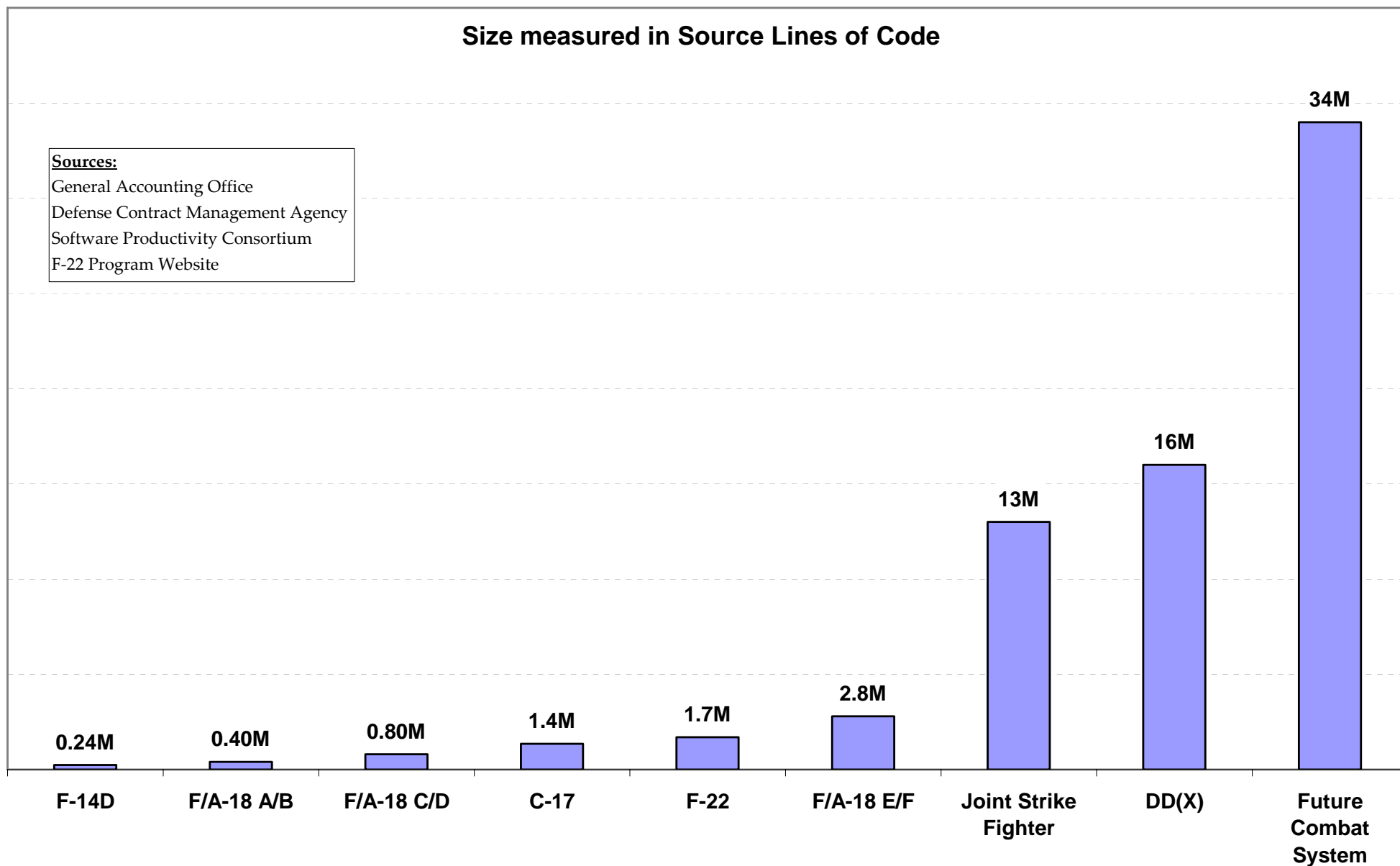


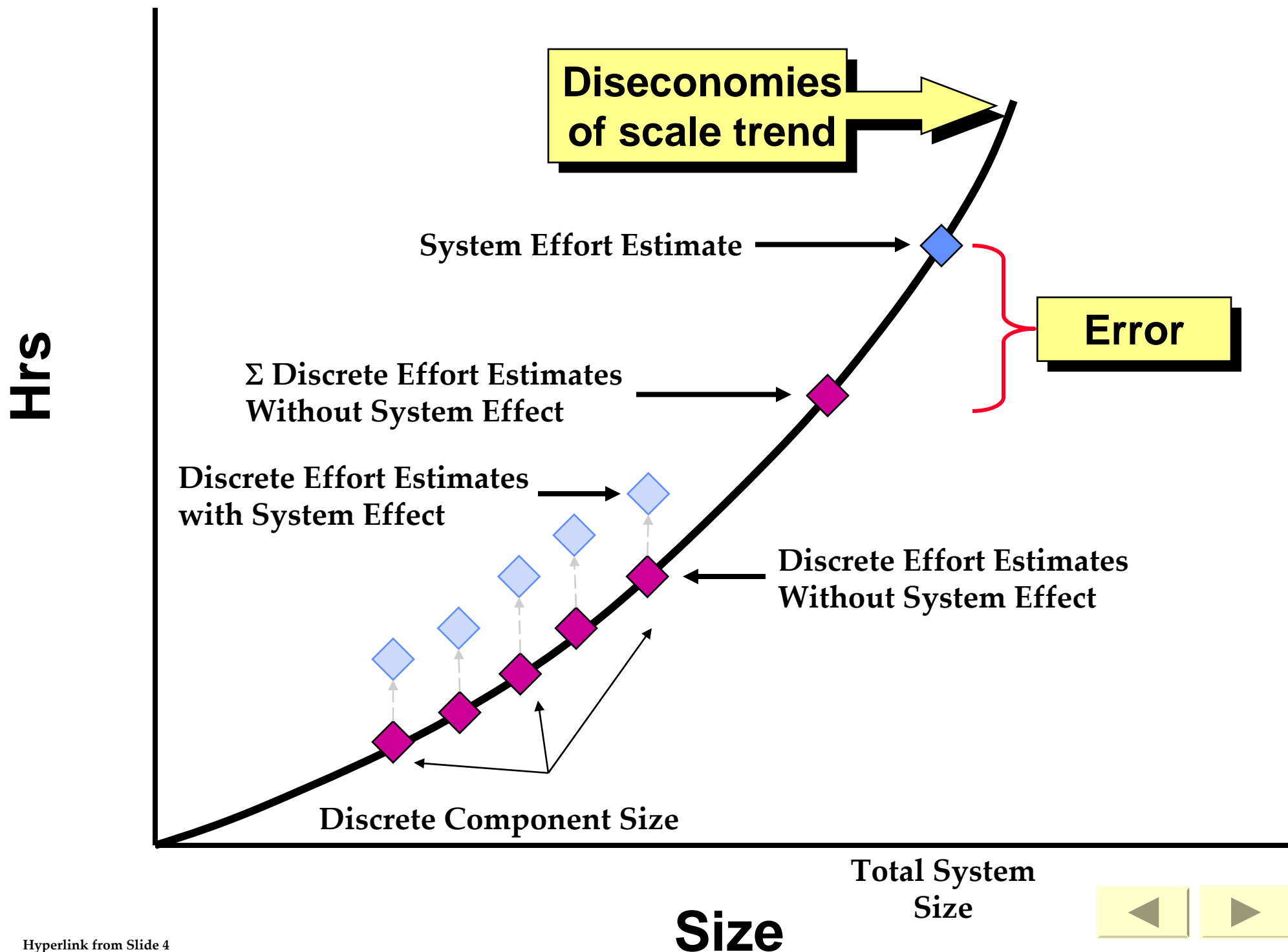
Trend in DoD Weapon System Software Size

Size measured in Source Lines of Code

Sources:

- General Accounting Office
- Defense Contract Management Agency
- Software Productivity Consortium
- F-22 Program Website





← Increasing Level of Integration

$10^7 - 10^6$

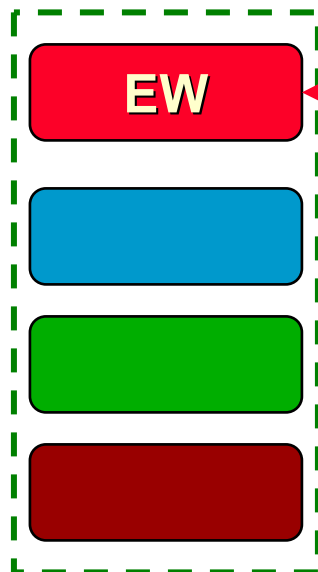
$10^6 - 10^5$

$10^5 - 10^4$

$10^5 - 10^3$

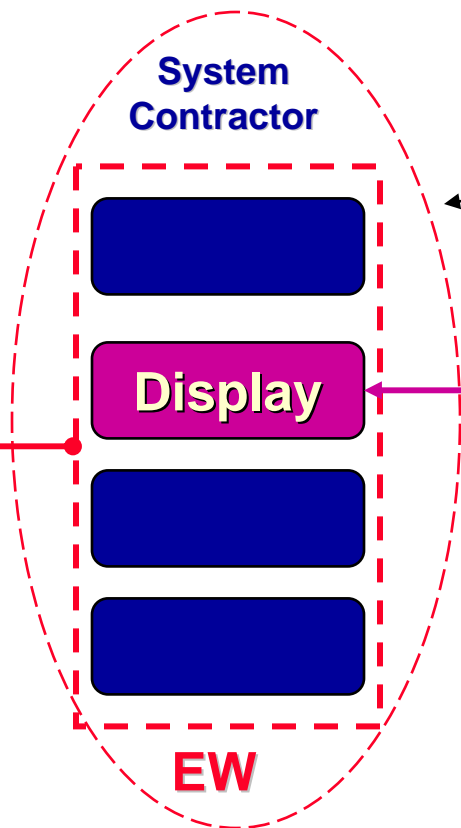
← Increasing Product Size (SLOC)

System of Systems Contractor

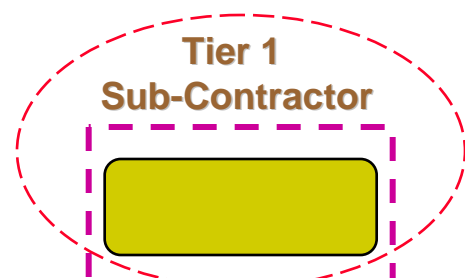


Integrated Avionics

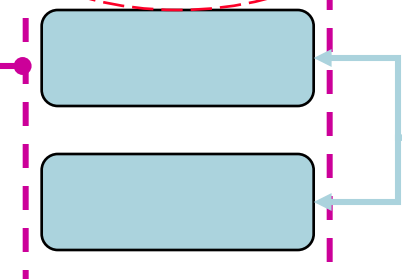
System Contractor



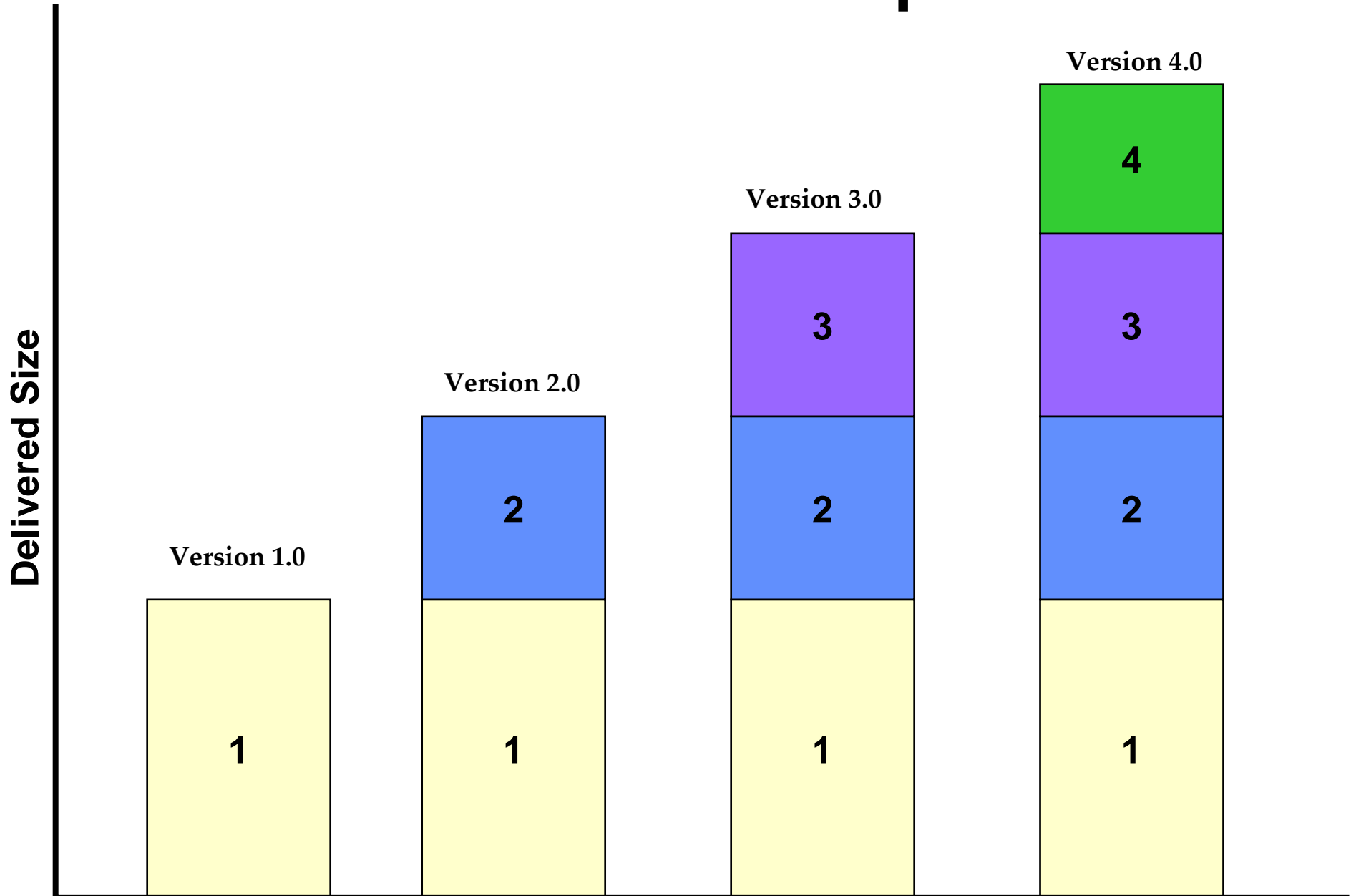
Prevalence of Data Collected



Tier 2 Sub-Contractor



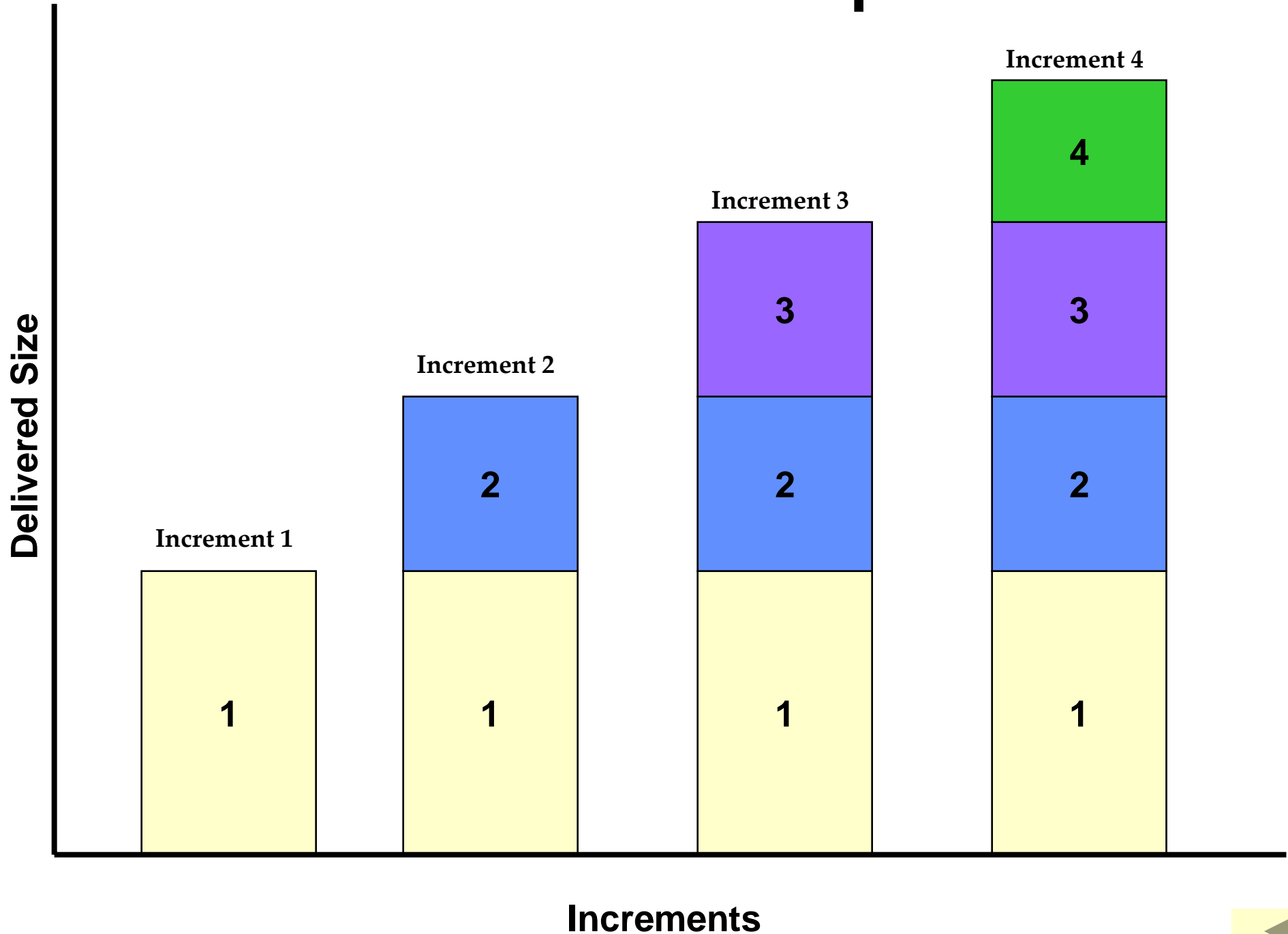
Pre-Planned Product Improvement



Production Versions



Incremental Development



Contractor A Standard

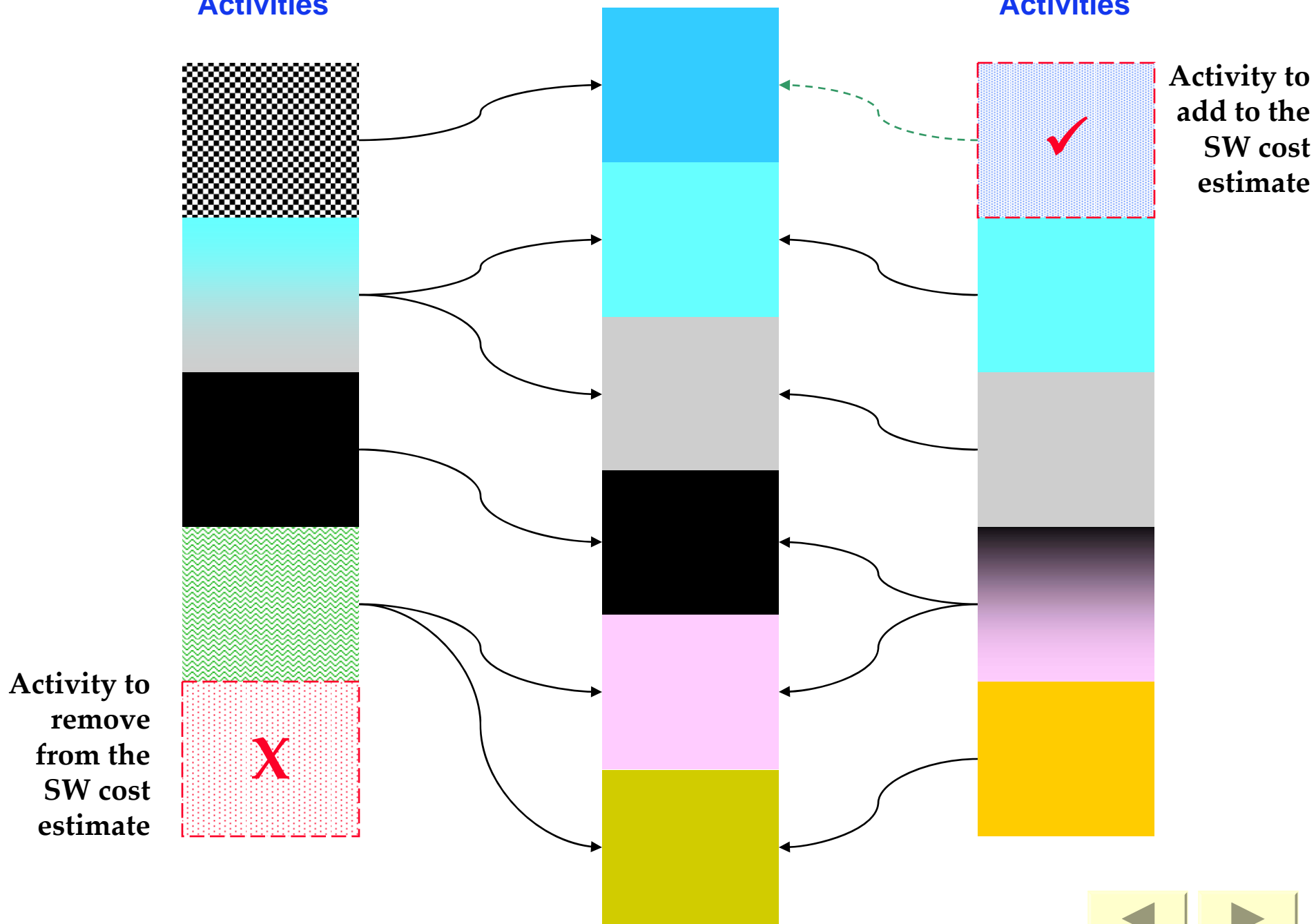
Standardized Software

Contractor B Standard

**SW Development
Activities**

Cost Estimate

**SW Development
Activities**



Activity to
remove
from the
SW cost
estimate

Activity to
add to the
SW cost
estimate



