



# **COTS Empirical Data: A Top-10 List And Its Implications**

**Barry Boehm, USC and CeBASE\*  
Opening Address, GSAW 2002  
March 13, 2002**

**(boehm@sunset.usc.edu)**

**(<http://www.cebase.org>)**

**\*Center for Empirically-Based Software Engineering**



# Outline

- **COTS Software: A Behavioral Definition**
- **COTS Empirical Hypotheses: Top-10 List**
  - **COTS Best Practice Implications**
- **COTS-Based Systems (CBS) Challenges**

# COTS Software: A Behavioral Definition

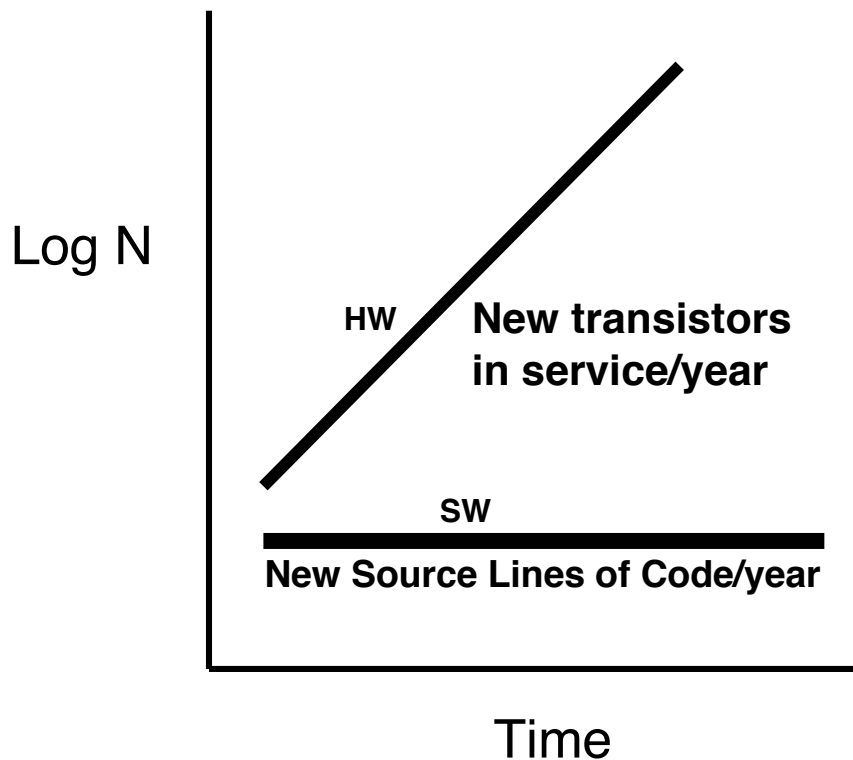
- **Commercially sold and supported**
  - **Avoids expensive development and support**
- **No access to source code**
  - **Special case: commercial open-source**
- **Vendor controls evolution**
  - **Vendor motivation to add, evolve features**
- **No vendor guarantees**
  - **Of dependability, interoperability, continuity, ...**

# COTS Top-10 Empirical Hypotheses-I

- Basili & Boehm, *Computer*, May 2001, pp. 91-93

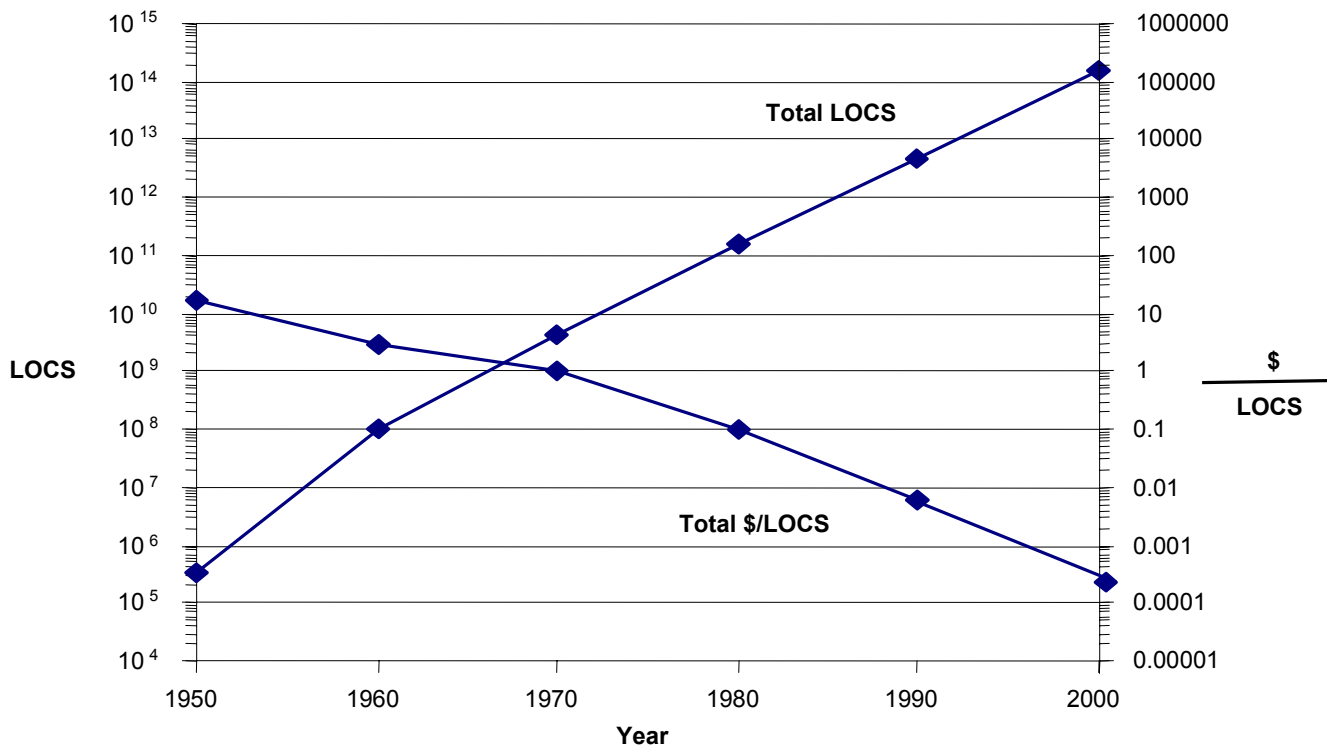
- **Over 99% of all executing computer instructions come from COTS**
  - Each has passed a market test for value
- **More than half of large-COTS-product features go unused**
- **New COTS release frequency averages 8-9 months**
  - Average of 3 releases before becoming unsupported
- **CBS life-cycle effort can scale as high as  $N^2$** 
  - $N = \#$  of independent COTS products in a system
- **CBS post-deployment costs exceed development costs**
  - Except for short-lifetime systems

# Usual Hardware-Software Trend Comparison



- Different counting rules
  - Try counting software as Lines of Code in Service (LOCS)
- $$= \sum (\#platforms) * (\#object\ LOCS/platform)$$

# Lines of Code in Service: U.S. Dept. of Defense



# 1. DoD LOCS: % COTS, 2000

M = Million, T = Trillion

Platform	# P'forms (M)	<u>LOCS</u> P'forms (M)	LOCS (T)	% COTS	Non-COTS LOCS (T)
Mainframe	.015	200	3	95	.15
Mini	.10	100	10	99	.10
Micro	2	100	200	99.5	1.00
Combat	2	2	4	80	.80
<b>Total</b>			<b>217</b>		<b>2.05</b>

(<1%)

**• COTS often an economic necessity**



## 2. Use of COTS Features: Microsoft Word and Power Point

- K. Torii Lab, Japan: ISERN 2000 Workshop

- **Individuals:** 12-16% of features used
- **10-Person Group:** 26-29% of features used

- **Extra features cause extra work**
- **Consider build vs. buy for small # features**

# 3. COTS Release Frequency: Survey Data

- Ron Kohl surveys: GSAW 1999-2001\*

GSAW Survey	Release Frequency (months)
1999	6.3
2000	8.5
2001	8.75

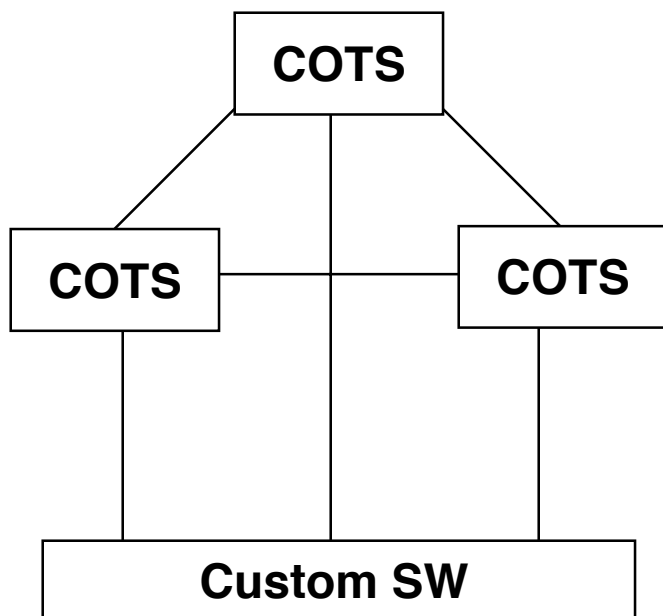
- Adaptive maintenance often biggest CBS life cycle cost
- Average of 3 releases before becoming unsupported

\* Ground System Architecture Workshop: 2002- Aerospace Corp., LA, March 13-15

# 4. CBS Effort Scaling As High As $N^2$

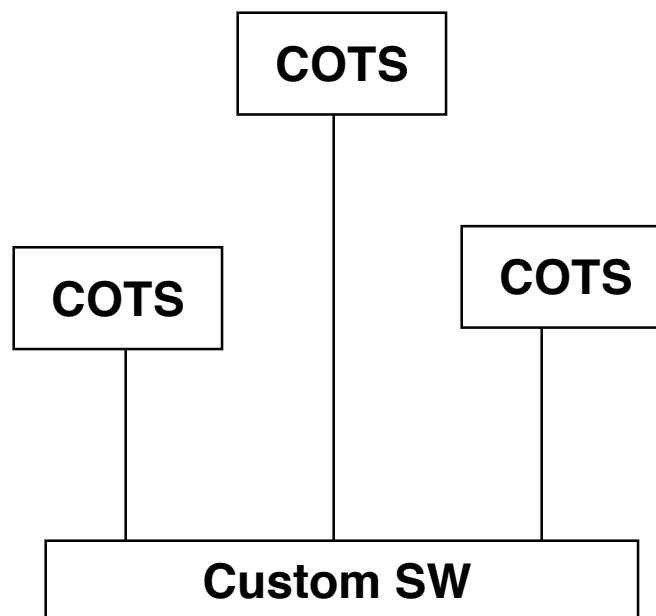
$N$  = number of independent COTS products

**Strong COTS Coupling**



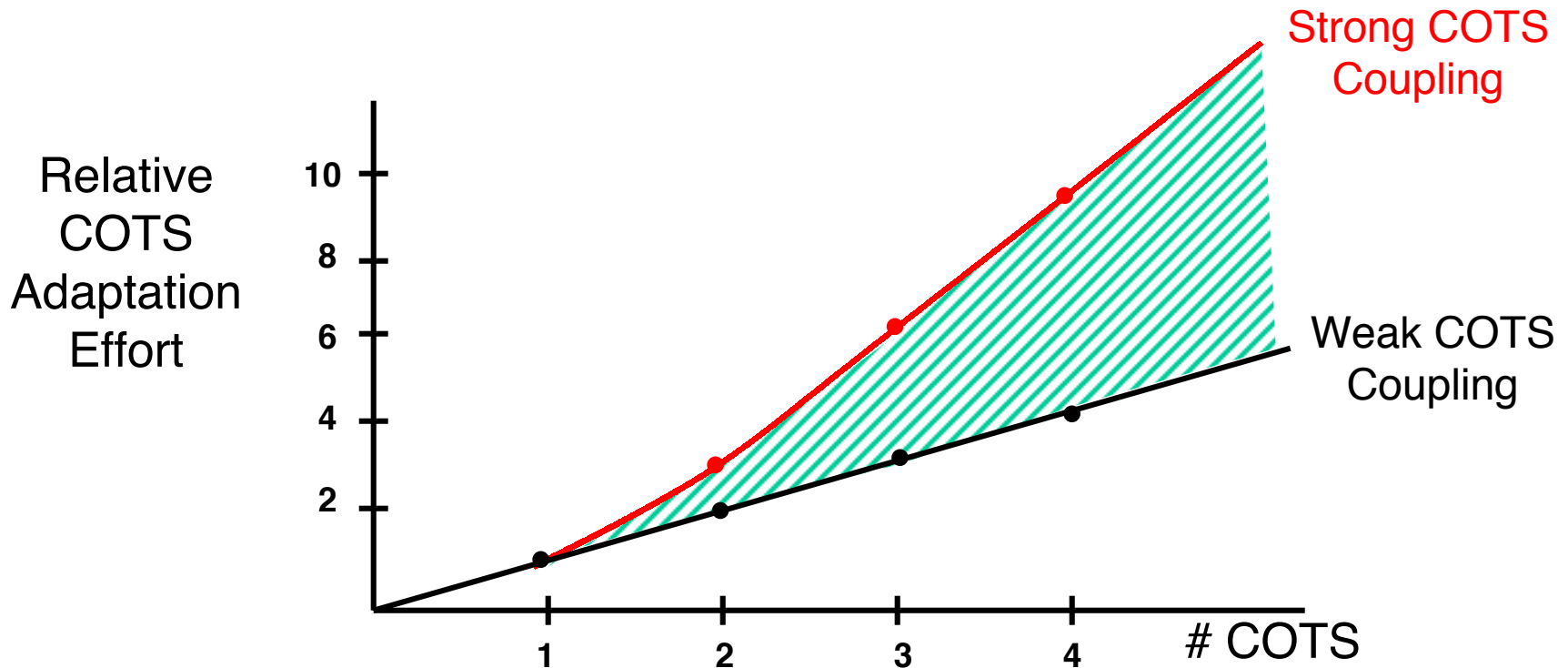
Effort  $\sim N + N(N-1)/2 =$   
 $N(N+1)/2$  (here = 6)

**Weak COTS Coupling**



Effort  $\sim N$  (here = 3)

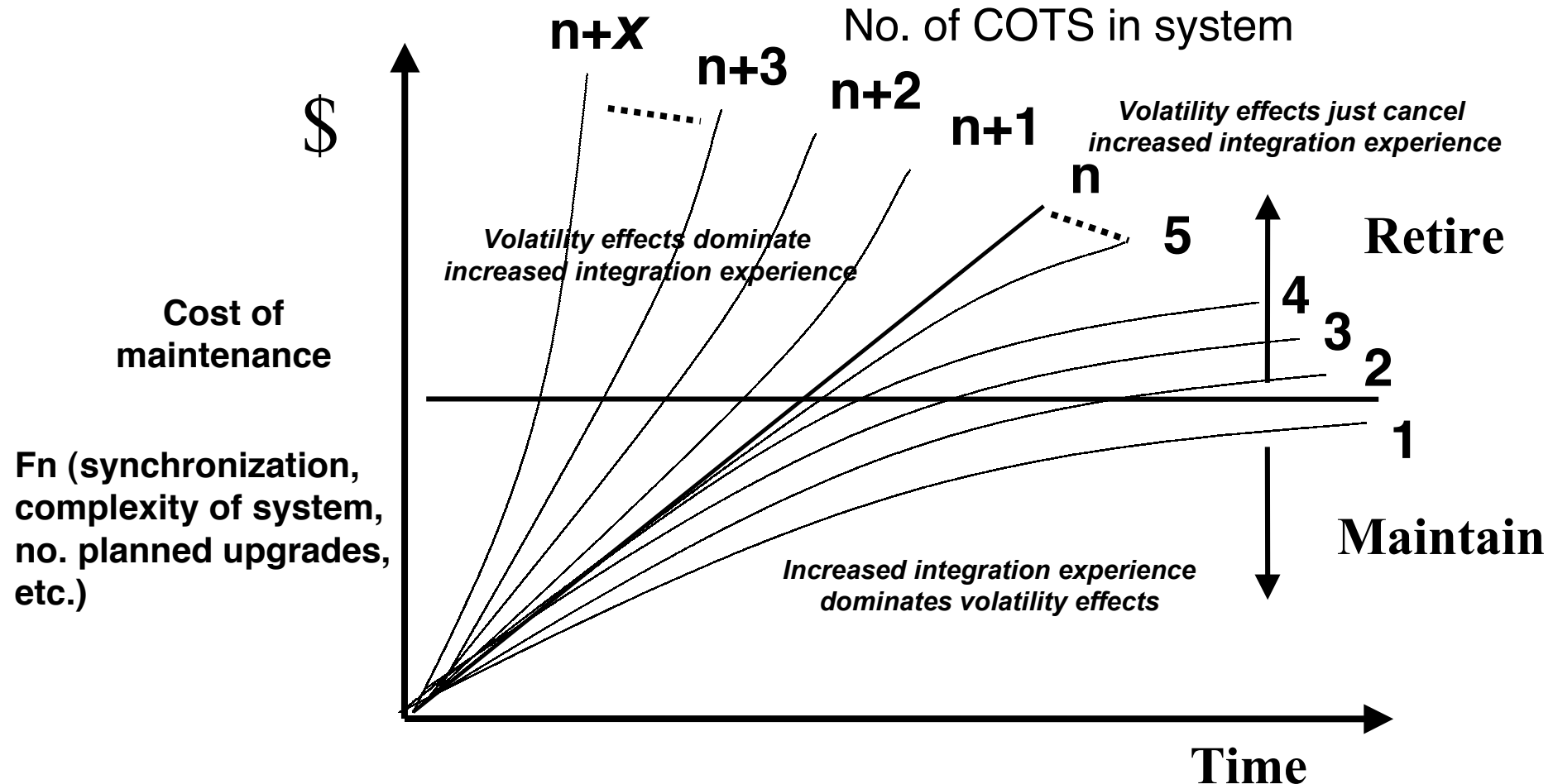
## 4. CBS Effort Scaling - II



- **Reduce # COTS or Weaken COTS coupling**
  - **COTS choices, wrappers, domain architectures, open standards, COTS refresh synchronization**

# 5. CBS Life Cycle Costs: The COTS-LIMO Model

- Chris Abts, ESCOM 2001

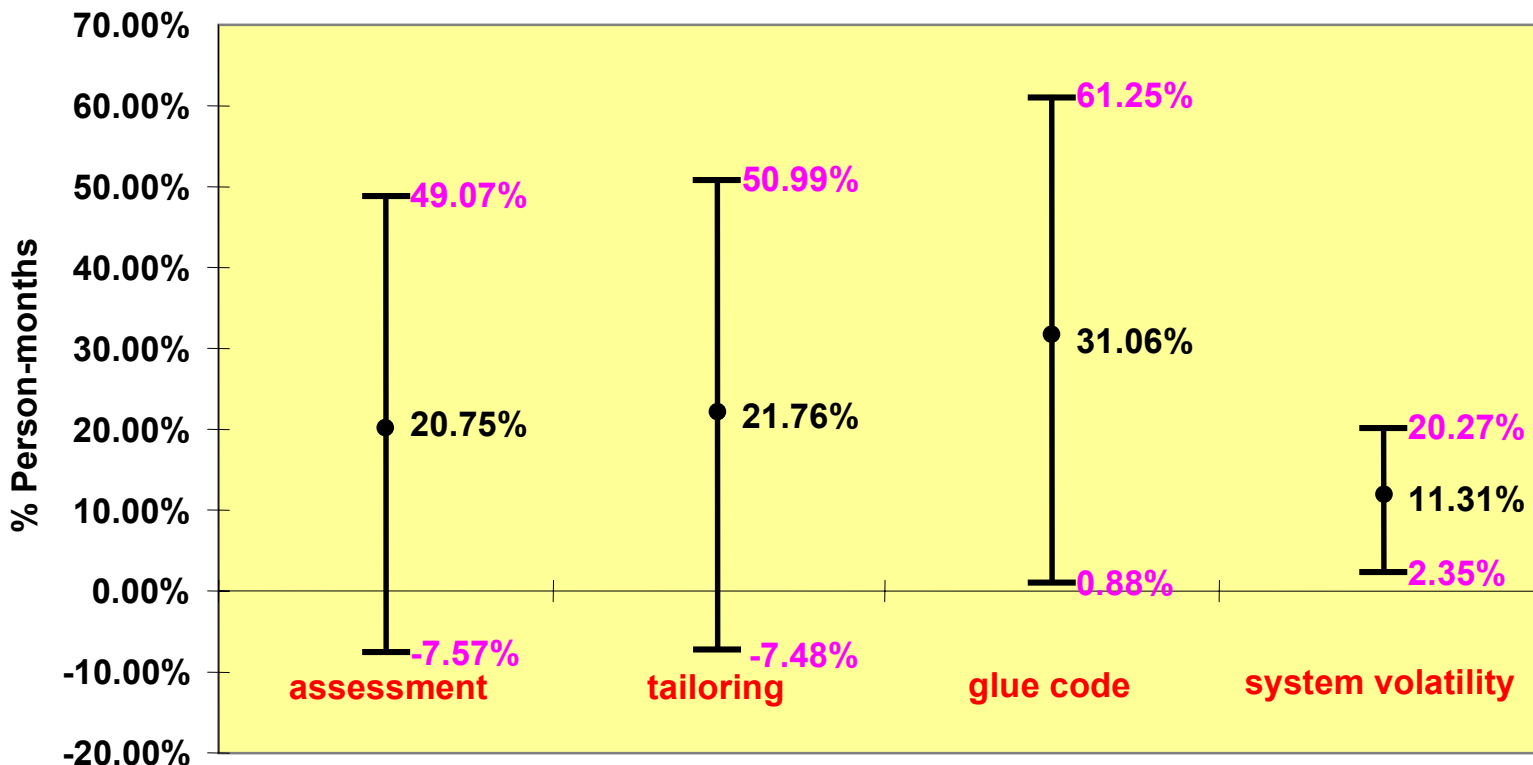


# COTS Top-10 Empirical Hypotheses-II

- **Less than half of CBS development effort comes from glue code**
  - **But glue code costs 3x more per instruction**
- **Non-development CBS costs are significant**
  - **Worth addressing, but not overaddressing**
- **COTS assessment and tailoring costs vary by COTS product class**
- **Personnel factors are the leading CBS effort drivers**
  - **Different skill factors are needed for CBS and custom software**
- **CBS project failure rates are higher than for custom software**
  - **But CBS benefit rates are higher also**

# 6. COCOTS Effort Distribution: 20 Projects

- Mean % of Total COTS Effort by Activity (+/- 1 SD)



- Glue code generally less than half of total
- No standard CBS effort distribution profile

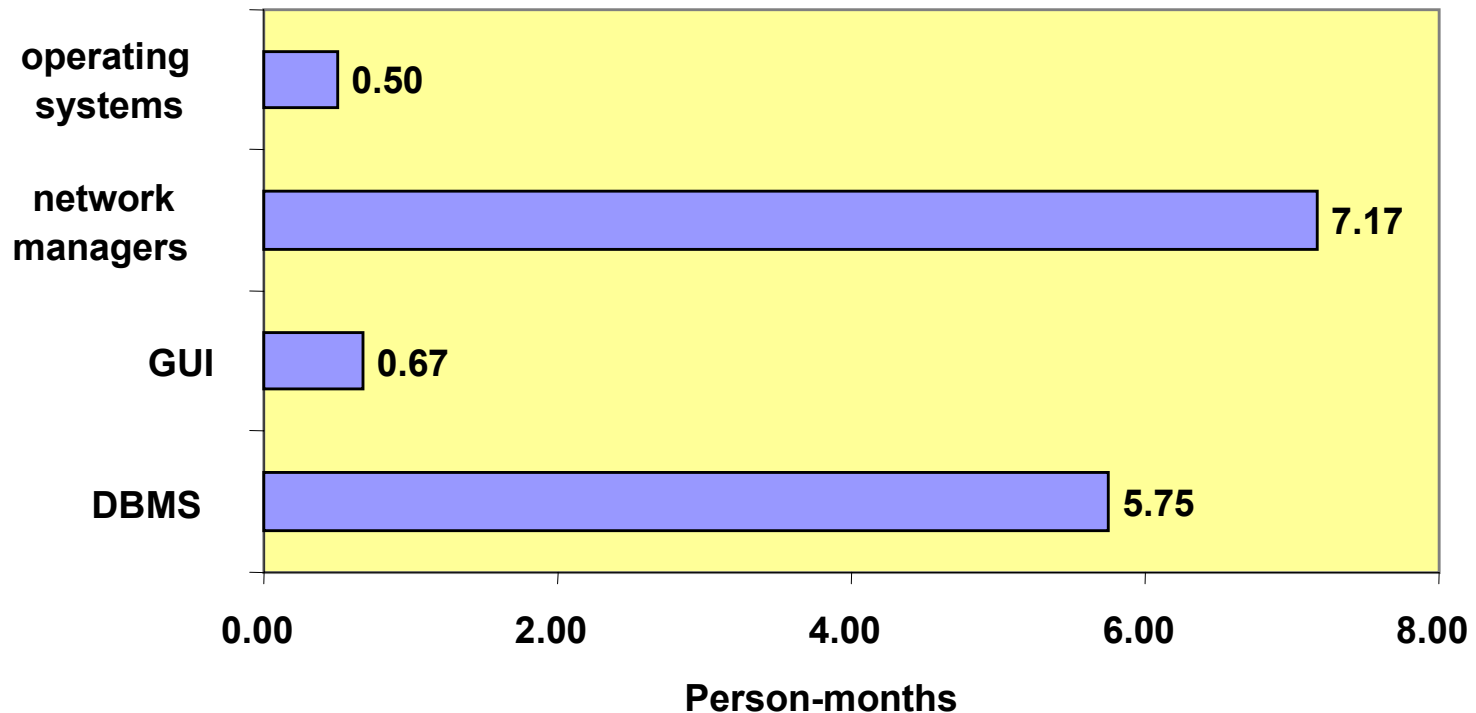
## 7. Non-Development CBS Costs Are Significant

- **Include licenses, training, vendor relations**
- **Benefits of addressing them**
  - **Volume discounts: \$1-10M/year savings**
  - **Unified vendor interface**
- **Costs of overaddressing them**
  - **Go with immature open-source freeware**
  - **Using COTS license cost as COTS assessment cost limit**

• **Use risk to determine how much COTS assessment is enough**

# 8. Median Assessment Effort by COTS Class

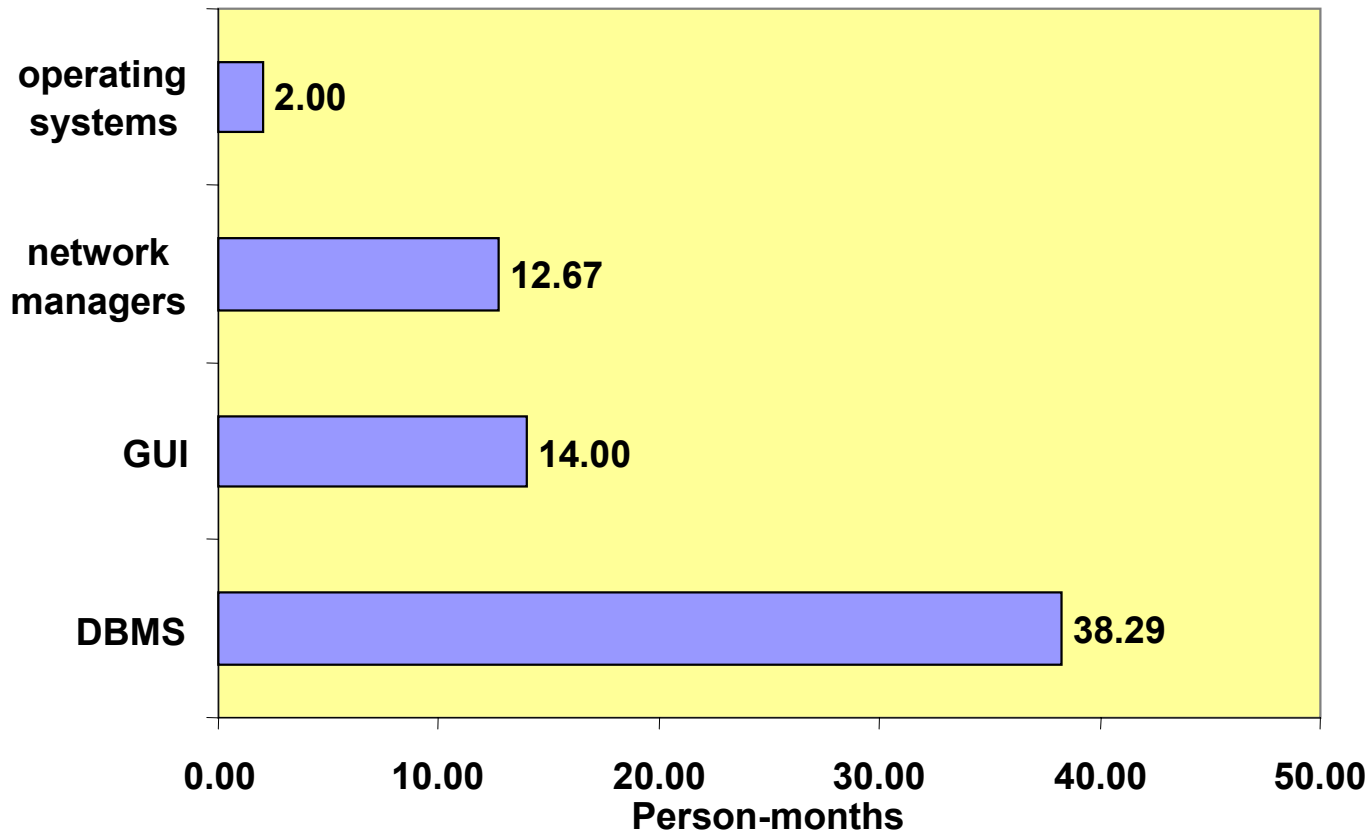
– COCOTS database: 20 projects



- **DBMS, network COTS most likely sources of assessment effort**
- **Again, no standard CBS effort distribution**

# 8. Median Tailoring Effort by COTS Class

– COCOTS database: 20 projects



- **DBMS the most likely source of COTS tailoring effort**
- **Again, no standard CBS effort distribution**

# 9. Personnel Factors Are Leading CBS Effort Drivers: Different Skill Factors Needed

## Custom Development

- Rqts./implementation assessment
- Algorithms; data & control structures
- Code reading and writing
  
- Adaptation to rqts. changes
- White-box/black-box testing

## CBS Development

- Rqts./ COTS assessment
- COTS mismatches; connector structures
- COTS mismatches; assessment, tailoring, glue code development; coding avoidance
- Adaptation to rqts., COTS changes
- Black-box testing

**• Rethink personnel recruiting and evaluation criteria**

## 10. CBS Projects Tend to Fail Hard

### - Major Sources of CBS Project Failure

- CBS skill mismatches
- CBS inexperience
- CBS optimism
- Weak life cycle planning
- CBS product mismatches
- Hasty COTS assessment
- Changing vendor priorities
- New COTS market entries

• These are major topics for COTS risk assessment

# CBS Challenges

- **Process specifics**
  - **Milestones; role of “requirements”**
  - **Multi-COTS refresh frequency and process**
- **Life cycle management**
  - **Progress metrics; development/support roles**
- **Cost-effective COTS assessment and test aids**
  - **COTS mismatch detection and redressal**
- **Increasing CBS controllability**
  - **Technology watch; wrappers; vendor win-win**
- **Better empirical data and organized experience**