University of Southern California
Center for Software Engineering

Costing the Development of Secure Systems

Presented by: Barry Boehm
Contributions by: Edward Colbert, Yue Chen, Danni Wu, Barry Boehm, & Donald Reifer
{ecolbert, yuec, boehm}@cse.usc.edu, danwu@usc.edu, d.reifer@ieee.org

8th Annual Practical Software & Systems Measurement Users' Group Conference

 1 27 July 2004

University of Southern California
Center for Software Engineering

Goal Of Presentation

- Review
  - Model for costing development of **secure systems**
  - Extensions to COCOMO II for development of **secure software systems**

 2 27 July 2004

University of Southern California
Center for Software Engineering

Outline

- Background
- Approach
- Develop Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
- Supplement

 3 27 July 2004

University of Southern California
Center for Software Engineering

Why Extend COCOMO II for Security

- Military projects have considered security in developing software since early 1980s
- Until recently commercial projects often gave it little weight
- Threat to business-critical systems & private information has grown
  - Security can no longer be ignored

 4 27 July 2004

University of Southern California
Center for Software Engineering

Why Extend COCOMO II for Security? (cont.)

- Few cost models (including COCOMO II) include security factors
  - Based 1980s military perspective (*Orange Book)*
  - Developing secure systems has changed dramatically (*Common Criteria*)
- Project cost agreed to be high; but wide variation in amount of added cost estimated by different models
  - [Bisignani and Reed 1988] estimates factor of 8 cost increase for very highly secure software
  - 1990's Softcost-R model estimates factor of 3.43 [Reifer 2002]
  - [Hall and Chapman 2002] report reduced overall system costs

 5 27 July 2004

University of Southern California
Center for Software Engineering

Why Extend COCOMO II for Security? (cont.)

- U.S. Information Security Management Act (FISMA) mandates Agencies to make security's role explicit in IT investments & capital planning
- In 2002, FAA asked USC-CSE's help

 6 27 July 2004

University of Southern California
Center for Software Engineering

Why Cost Model for Secure System? (From FAA Project Workshop, July 2003)

- Concerns about information–security cost model
  - "Pure" software models
    - Estimates software development cost
    - Excludes
      - Additional cost critical to information security
        - » e.g., training, licenses
      - Life-cycle Operations & Maintenance costs for information security
        - » Development group responsible for first 2 years O&M

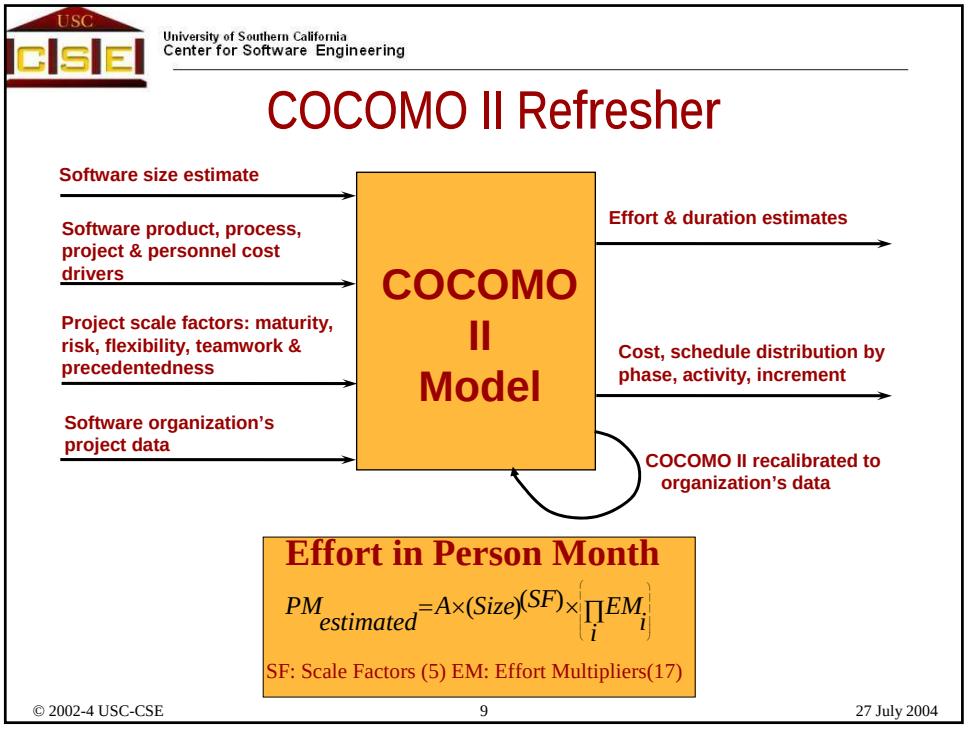
 7 27 July 2004

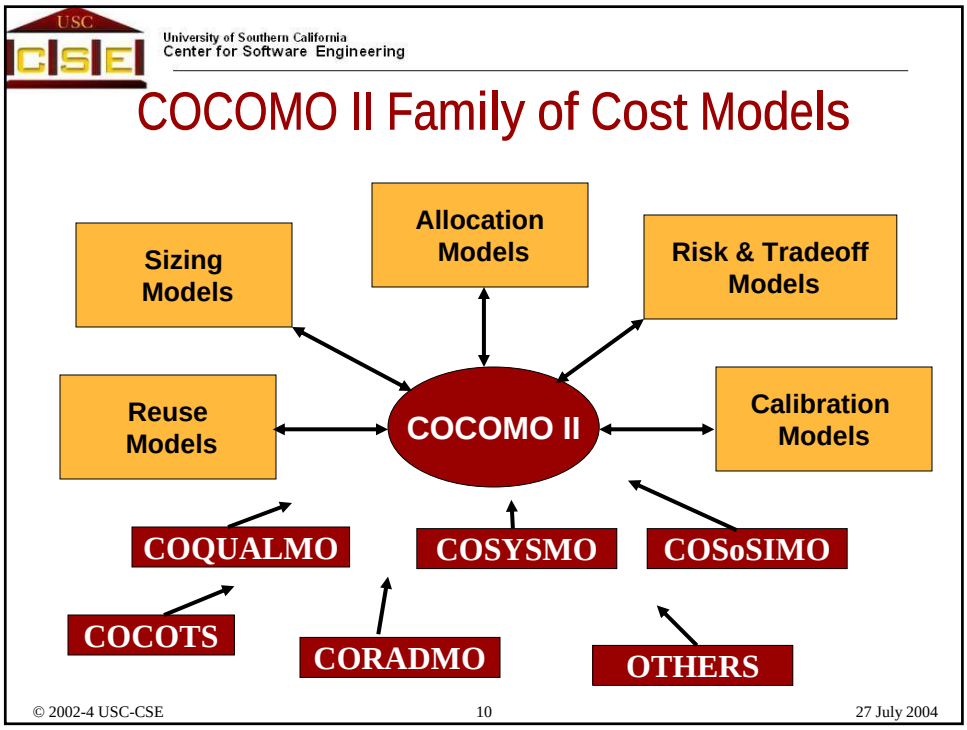
University of Southern California
Center for Software Engineering

Outline

- Background
- Approach
- Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
- Supplement

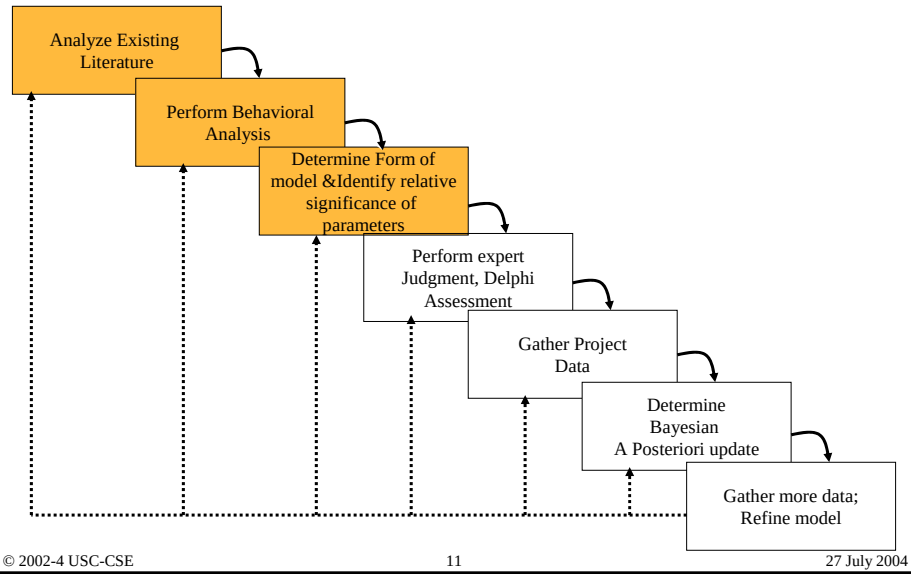
 8 27 July 2004


University of Southern California
Center for Software Engineering
COCOMO II Refresher
Software size estimate
Software product, process, project & personnel cost drivers
Project scale factors: maturity, risk, flexibility, teamwork & precedentedness
Software organization's project data
COCOMO II Model
Effort & duration estimates
Cost, schedule distribution by phase, activity, increment
COCOMO II recalibrated to organization's data
Effort in Person Month
$PM_{estimated} = A \times (Size)^{(SF)} \times \left\{ \prod_i EM_i \right\}$
SF: Scale Factors (5) EM: Effort Multipliers(17)
© 2002-4 USC-CSE
9
27 July 2004


University of Southern California
Center for Software Engineering
COCOMO II Family of Cost Models
Sizing Models
Allocation Models
Risk & Tradeoff Models
Reuse Models
COCOMO II
Calibration Models
COQUALMO
COSYSMO
COSoSIMO
COCOTS
CORADMO
OTHERS
© 2002-4 USC-CSE
10
27 July 2004

University of Southern California
Center for Software Engineering

COCOMO II Modeling Methodology



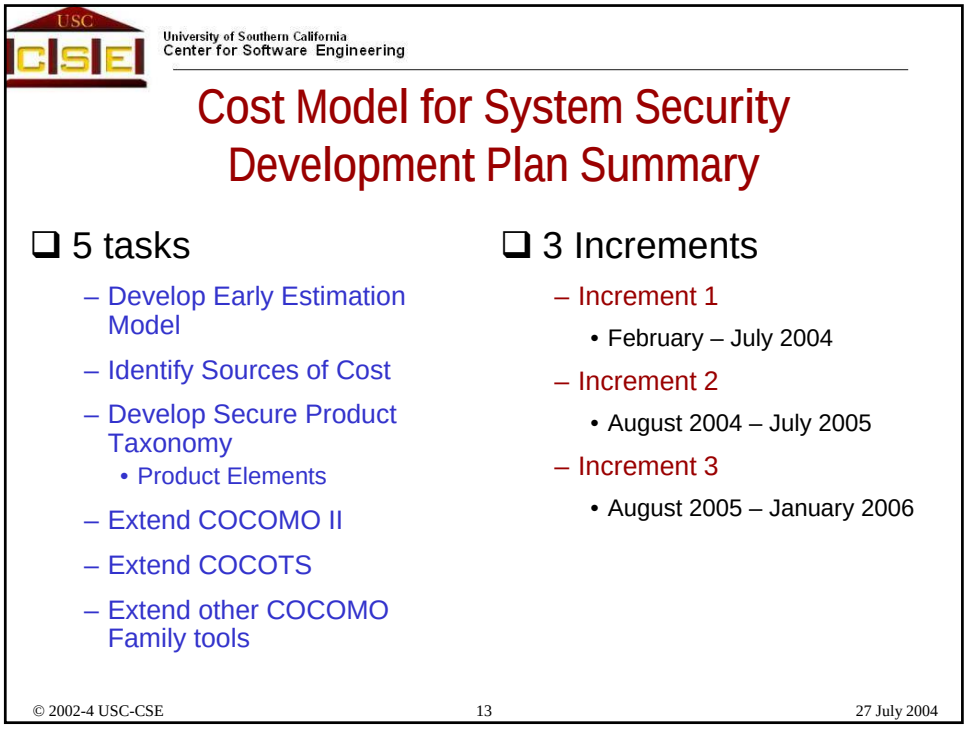
 11 27 July 2004

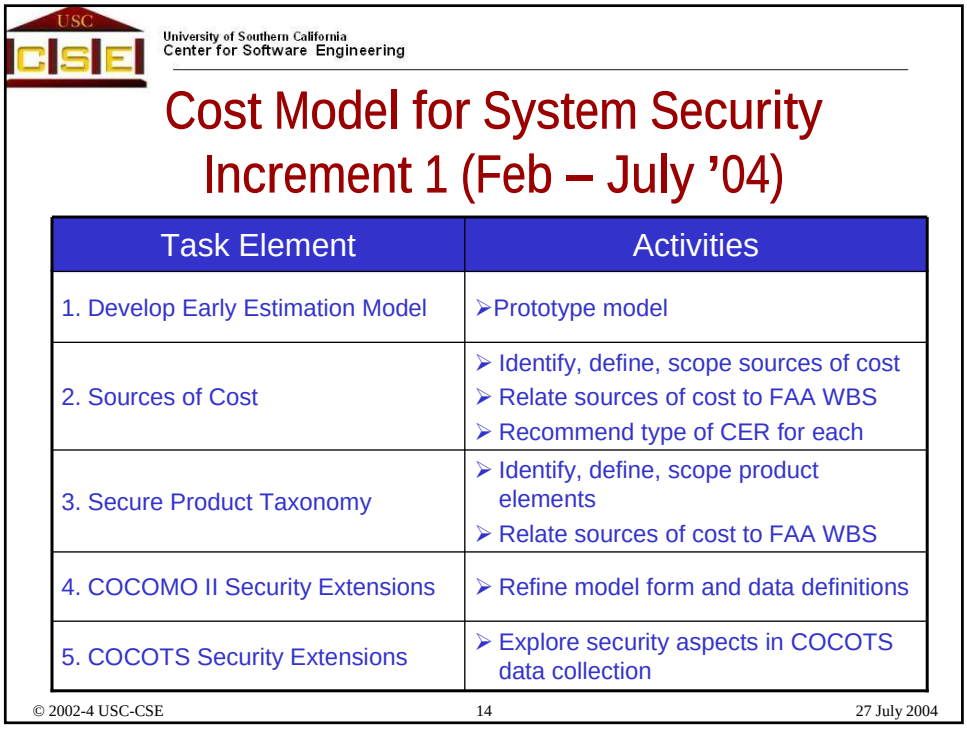
University of Southern California
Center for Software Engineering

Cost Model for Secure System Approach

- Develop cost model incrementally
  - Identify major sources of cost
  - Describe conditions under which each source of cost will drive decisions at “top level”
  - Identify Cost Estimation Relations (CER)
    - Ways to estimate costs
    - Define “early” techniques
    - Refine/replace as get more data

 12 27 July 2004


University of Southern California
Center for Software Engineering
Cost Model for System Security Development Plan Summary
5 tasks
– Develop Early Estimation Model
– Identify Sources of Cost
– Develop Secure Product Taxonomy
• Product Elements
– Extend COCOMO II
– Extend COCOTS
– Extend other COCOMO Family tools
3 Increments
– Increment 1
• February – July 2004
– Increment 2
• August 2004 – July 2005
– Increment 3
• August 2005 – January 2006
© 2002-4 USC-CSE
13
27 July 2004


University of Southern California
Center for Software Engineering
Cost Model for System Security Increment 1 (Feb – July '04)
Task Element
Activities
1. Develop Early Estimation Model
Prototype model
2. Sources of Cost
Identify, define, scope sources of cost
Relate sources of cost to FAA WBS
Recommend type of CER for each
3. Secure Product Taxonomy
Identify, define, scope product elements
Relate sources of cost to FAA WBS
4. COCOMO II Security Extensions
Refine model form and data definitions
5. COCOTS Security Extensions
Explore security aspects in COCOTS data collection
© 2002-4 USC-CSE
14
27 July 2004

University of Southern California
Center for Software Engineering

Outline

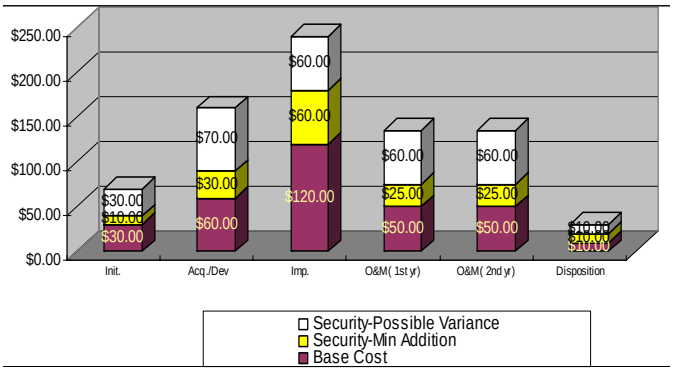
- Background & Results of Previous Workshops
- Approach
- Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
- Supplement

 15 27 July 2004

University of Southern California
Center for Software Engineering

Draft Model of Cost Distribution System Purchase

- Distribution depends on type system acquired
- # systems
  - affects
    - Installation
    - O&M
    - Disposition
  - After 1st, costs are less
- Example: $1M System
  - **Numbers for show only**

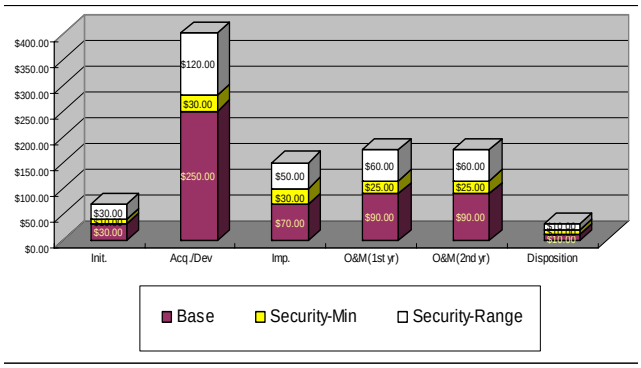


 16 27 July 2004

University of Southern California
Center for Software Engineering

Draft Model of Cost Distribution System Development

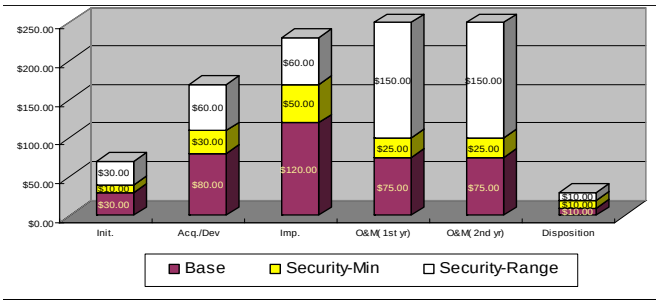
- Distribution depends on type system acquired
- # systems
  - affects
    - Installation
    - O&M
    - Disposition
  - After 1st, costs are less
- Example: $1M System
  - **Numbers for show only**



University of Southern California
Center for Software Engineering

Draft Model of Cost Distribution System Development with COTS

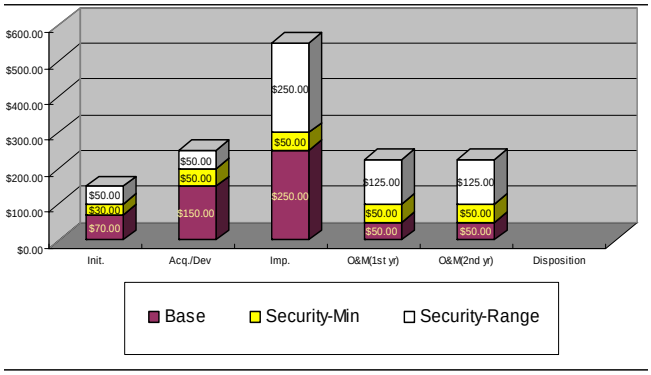
- Distribution depends on type system acquired
- # systems
  - affects
    - Installation
    - O&M
    - Disposition
  - After 1st, costs are less
- Example: $1M System
  - **Numbers for show only**



University of Southern California
Center for Software Engineering

Draft Model of Cost Distribution Services

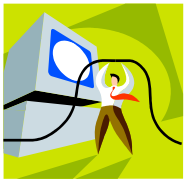
- Distribution depends on type system acquired
- # systems
  - Affects
    - Implementation
    - O&M
    - Disposition
  - After 1st, costs are less
- Example: $1M System
  - **Numbers for show only**



University of Southern California
Center for Software Engineering

Cost Model for Secure System Approach

- Identify major sources of cost
  - To
    - Develop
    - Own
  - Including
    - Facilities
    - Equipment
    - People
    - Acquired Systems
    - Services



University of Southern California
Center for Software Engineering

How to Estimate Costs?

- Costing Approaches
  - Activity Models
  - Unit Costing
  - Analogy Base
  - Parametric
- For each source of cost, identify appropriate means
  - *Cost Estimation Relation (CER)*

 21 27 July 2004

University of Southern California
Center for Software Engineering

Cost Estimation Relations (CER) Example

Sample Activity	Preparation for Training	Classroom Training	Periodic Training on new procedures	Software Development
CER	*Activity–based*	*Unit costing*	*Analogy-based*	*Parametric*
Rule	*10-20 hours for each Class Hour*	N trainers total M trainees	*It cost us $XXX last year,…*	*COCOMO II*

 22 27 July 2004

University of Southern California
Center for Software Engineering

Outline

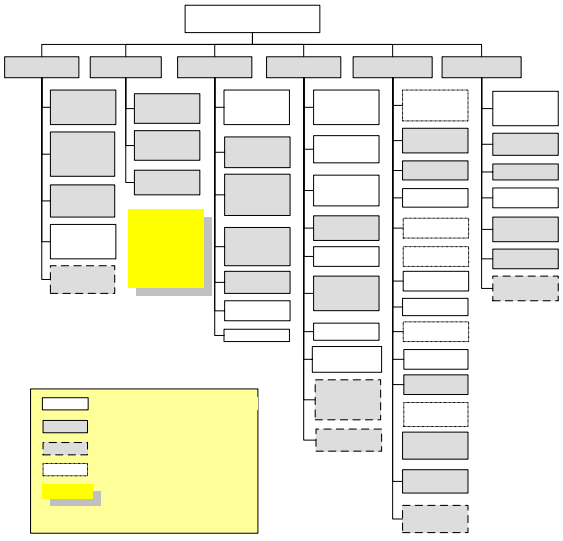
- Background & Results of Previous Workshops
- Approach
- Develop Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
- Supplement

 23 27 July 2004

University of Southern California
Center for Software Engineering

FAA Acquisition & Standard WBS

- Analyzed FAA WBS to identify where security will affect activities
  - Looking at how to cost each activity affected by security
- Expanding study to Standard WBS



 24 27 July 2004

University of Southern California
Center for Software Engineering

Outline

- ❑ Background & Results of Previous Workshops
- ❑ Approach
- ❑ Develop Early Estimation Model
- ❑ Identify Sources of Cost
- ❑ Develop Secure Product Taxonomy
- ❑ Extend COCOMO II
- ❑ Next Steps & Summary
- ❑ Supplement

For this presentation, we're skipping this section because of time constraints. See Supplement

 25 27 July 2004

University of Southern California
Center for Software Engineering

Effect Of Security On COCOMO II

- ❑ Security functional requirements add to project's KSLOC's
  - Many systems add separate *trusted software* component(s)
    e.g. Security Manager or Authorization– & Access-Administrator
  - Core Application (*untrusted software*) adds code to do support authorization & access checks

$$PM_{total} = PM_{trusted} + PM_{application}$$

- ❑ Security assurance requirements increase effort to produce code
  - A few functional requirements also increase effort

 26 27 July 2004

University of Southern California
Center for Software Engineering

Modification & Use of COCOMO II

- Add 1 driver (SECU) that addresses security
  - Based on 3 factors
    - Design & Development for Security
    - Operational Security
    - Physical Security (Development Constraints)
  - Derived from *Common Criteria* & industry best practices
- Constrain existing COCOMO II drivers
  - Security overlaps with some existing drivers
    - e.g. Reliability (RELY), Documentation (DOC)
  - Don't want to double count effect

 27 27 July 2004

University of Southern California
Center for Software Engineering

Modification & Use of COCOMO II (cont.)

- Compute effort for trusted & non-trust software separately
  - Trusted software is typically developed at higher assurance levels
- New Driver Ratings
  - Nominal
  - High
  - Very High
  - Extremely High
  - *Sky High*
  - *Stratospheric*

New COCOMO Levels

 28 27 July 2004

University of Southern California
Center for Software Engineering

Example of COCOMO Security Extension

Assume:

Application	1000 KSLOC	SECU (App)	= Nominal
Trusted SW	10 KSLOC	SECU (trusted)	= High
SF =	1		

All Multipliers (except Security) = 1 (Nominal)

SECU Values

- Nominal = 1
- High = 2
- VH = 3
- EH = 4
- SH = 6
- Strat = 7

Effort in Person Month

$$PM_{estimated} = A \times (Size)^{(SF)} \times \left(\prod_i EM_i \right)$$

SF: Scale Factors (5) EM: Effort Multipliers(17)

Then

PM (total) = 2.94 * (PM (Trusted) + PM (App))
= 2.94 * (10 * SECU(Trusted)) + (1000 * SECU(App))
= 2.94 * ((10 * 2) + (1000 * 1))
= 2.94 * 1020

 29 27 July 2004

University of Southern California
Center for Software Engineering

Outline

- ❑ Background & Results of Previous Workshops
- ❑ Approach
- ❑ Develop Early Estimation Model
- ❑ Identify Sources of Cost
- ❑ Develop Secure Product Taxonomy
- ❑ Extend COCOMO II
- ❑ Next Steps & Summary
- ❑ Supplement

 30 27 July 2004

University of Southern California
Center for Software Engineering

Summary (cont.)

- Proposed extensions to COCOMO for development of Secure Systems
  - Based on Common Criteria & DITSCAP
  - 1 Driver: SECU
    - 3 Factors:
      - Development for Security
      - Operational Security
      - Physical Security (Development Constraints)
  - Affects on other COCOMO II Drivers
  - Affects on size
  - Affects on project risk

 31 27 July 2004

University of Southern California
Center for Software Engineering

Summary

- Development of cost model for secure systems
  - Full life–cycle
    - Not just development
    - Including
      - 2 years Operation & Maintenance
      - Disposition
- Hopefully stimulated your interest & motivated you to participate by
  - Attending Workshops
  - Participate in Delphi
  - Sharing project data

 32 27 July 2004

University of Southern California
Center for Software Engineering

Next Steps

- Refine Behavior Analysis
  - Commercial
  - Security community
- Refine models
- Conduct Delphi
- Collect & analyze data
- Analyze security affects on other models
- Write Ph.D. Thesis (theses?)

 33 27 July 2004

University of Southern California
Center for Software Engineering

Next *Costing Secure Systems* Workshop

- Date: 28 October 2004
- Location: USC in LA
- Website: cse.usc.edu
  - Click on Events
  - Click on **19th International Forum on COCOMO and Software Cost Modeling**
- Questions:
  - Ed Colbert
  - Phone: (213) 821-1200
  - E-mail: ecolbert@usc.edu

 34 27 July 2004

University of Southern California
Center for Software Engineering

References

- Bisignani, M. and Reed, T. (1988). "Software Security Costing Issues", *COCOMO Users' Group Meeting Proceedings*. Los Angeles: USC Center for Software Engineering.
- Boehm, B. W., Abts, C., et al. (2000). *Software Cost Estimation with COCOMO II*, Prentice–Hall: Englewood Cliffs, NJ
- Hall, A. and Chapman, R. (2002). "Correctness by Constructuon: Developing a Commercial Secure System", (January/February): pp. 18-25.
- ISO JTC 1/SC 27 (1999a). *Evaluation Criteria for IT Security (Common Criteria), Part 1: Introduction and general model*, Standard No. ISO/IEC 15408-1, International Organization for Standardization (ISO), Vol. 1, No. 3, http://www.commoncriteria.org/, http://www.iso.ch.
- ISO JTC 1/SC 27 (1999b). *Evaluation Criteria for IT Security (Common Criteria), Part 2: Security functional requirements*, Standard No. ISO/IEC 15408-2, International Organization for Standardization (ISO), Vol. 2, No. 3, http://www.commoncriteria.org/, http://www.iso.ch.
- ISO JTC 1/SC 27 (1999c). *Evaluation Criteria for IT Security (Common Criteria), Part 3: Security assurance requirements*, Standard No. ISO/IEC 15408-3, International Organization for Standardization (ISO), Vol. 3, No. 3, http://www.commoncriteria.org/, http://www.iso.ch.
- National Computer Security Center (1985). *Trusted Computer System Evaluation Criteria*, National Computer Security Center: Washington, D.C.
- Reifer, D. (2002). *Security: A Rating Concept for COCOMO II*. Reifer Consultants, Inc.

 35 27 July 2004

University of Southern California
Center for Software Engineering

Outline

- Background & Results of Previous Workshops
- Approach
- Develop Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
- Supplement

 36 27 July 2004

University of Southern California
Center for Software Engineering

Cost Model for System Security Increment 2 (Aug '04 – July '05)

Task Element	Activities
1. Develop Early Estimation Model	➢ Experimental use & refinement
2. Sources of Cost	➢ Prioritize sources of cost needing CER's ➢ Refine, prototype, experiment with top-priority CER's ➢ Relate to scope of COCOMO II security extensions
3. Secure Product Taxonomy	➢ Experimental use, feedback, and refinement
4. COCOMO II Security Extensions	➢ Refine, scope, form, definitions based on results of Tasks 1-3 ➢ Experimentally apply to pilot projects, obtain usage feedback
5. COCOTS Security Extensions	➢ Develop initial scope, form, definitions based on results of Tasks 1-4

 37 27 July 2004

University of Southern California
Center for Software Engineering

Cost Model for System Security Increment 3 (Apr '05 – Sep '06)

Task Element	Activities
1. Develop Early Estimation Model	➢ Evolution; integration with other models
2. Sources of Cost	➢ Refine sources of cost, CER's based on usage feedback ➢ Integrate with other models ➢ Address lower-priority CER's as appropriate
3. Secure Product Taxonomy	➢Monitor evolution
4. COCOMO II Security Extensions	➢ Baseline model definitions ➢ Collect project data ➢ Develop initially calibrated model; experiment and refine
5. COCOTS Security Extensions	➢Experimentally apply to pilot projects ➢Refine, baseline based on usage feedback

 38 27 July 2004

University of Southern California
Center for Software Engineering

Secure Product Taxonomy

- Analyzing
  - Product security objectives relative to security functional requirements (SFR's)
  - SFR's to
    - Typical trusted Software Size Range
    - Effort to produce

 39 27 July 2004

University of Southern California
Center for Software Engineering

Security Taxonomy

- Information Security
  - Resources
  - Services
  - Data
    - e.g., files
- Physical Security
  - Environment
- Person Security
  - Certification
  - Personal security experience

 40 27 July 2004

University of Southern California
Center for Software Engineering

Information Security

- Security Objectives
  - Preventive
  - Detective
  - Corrective
- Solutions
  - Develop from scratch
  - Use Commercial Product
  - Sign Security Service Contract

 41 27 July 2004

University of Southern California
Center for Software Engineering

Preventive Security Objectives

- PSO1: Confidentiality
  - Only authorized users can access data
    - Enforcing access control
- PSO2: Authentication
  - Validate user using password or other authentication techniques
- PSO3: Integrity
  - Protecting data from malicious modification
- PSO4: Availability
  - Information can be accessed when user needs it
- PSO5: Non-Repudiation
  - Sender cannot deny having sent information
  - Receiver cannot deny having received the information

 42 27 July 2004

University of Southern California
Center for Software Engineering

Confidentiality

- Threats
  - Malicious (unauthorized) users can
    - Steal and compromise critical information
    - Break secrecy of information and privacy of users
- Approach of treating threats
  - Check user's capabilities
    - use access control list (ACL)
  - Encrypt messages
    - E.g., use public-private key
- Commercial Product
  - Keynote, WebID, Passport Authorization, WebSpeed, JAAS, Distribution 2000, etc

 43 27 July 2004

University of Southern California
Center for Software Engineering

Partial Map Security Objectives to Common Criteria

Common Criteria		Security Objectives								
Class	Family	Authentication	Identity Management	Confidentiality	Integrity	Availability	Non-repudiation	Accountability	Recoverability	Intrusion Detection and Response
Security Audit (FAU)	ARP									X
	GEN							X		
	SAA									X
	SAR							X		
	SEL							X		
	STG							X		
Communication (FCO)	NRO						X			
	NRR						X			
Cryptographic Support (FCS)	CKM	X		X	X		X			
	COP	X		X	X		X			
User Data Protection (FDP)	ACC			X						
	ACF		X							
	DAU	X	X							
	ETC		X							
	IFC			X						
	IFF			X						
	ITC		X							
	ITT			X	X					
	RIP			X	X					
	ROL								X	
	SDI				X					
	UCT			X						
	UIT				X					
Identification & authentication (FIA)	AFL	X								
	ATD		X							
	SOS		X							

 44 27 July 2004

University of Southern California
Center for Software Engineering

Partial Map Common Criteria to COCOMO

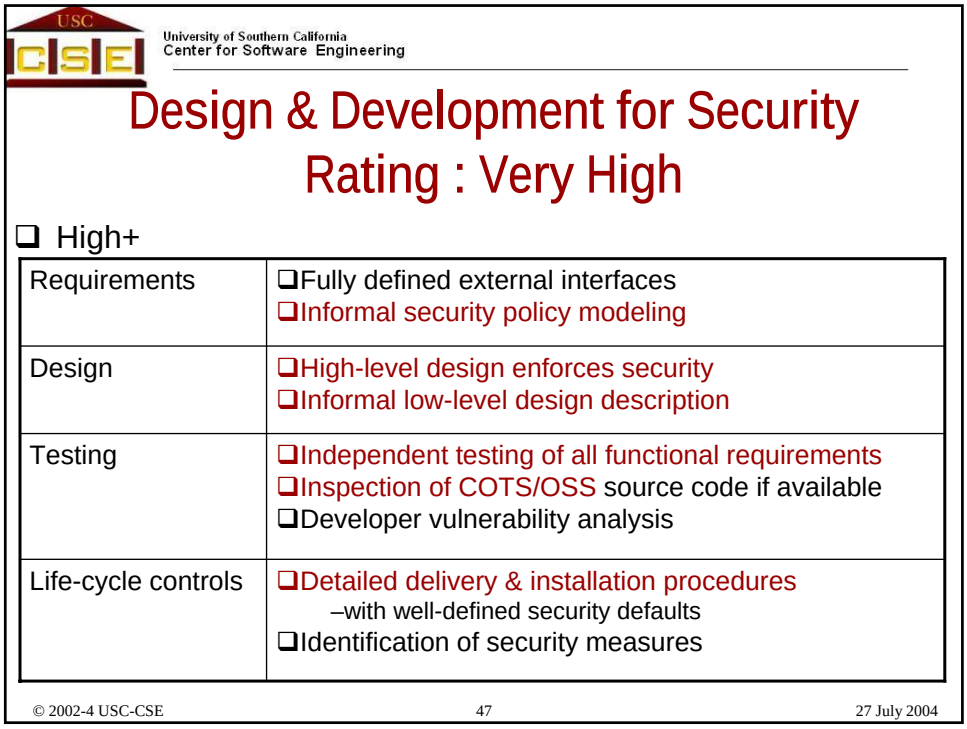
Common Criteria SFR's		COCOMO Drivers: Scalar PREC	FLEX	RESL	TEAM	PMAT	Multiplicative RELY	DATA	CPLX	RUSE	DOCU	TIME	STOR	PVOL	ACAP	PCAP	PCON	APEX	PLEX	LTEX	TOOL	SITE	SCED	SECU	Existing Drivers Affected	Total Drivers Affected
FAU	ARP																								0	0
	GEN												I												1	1
	SAA																								0	0
	SAR																								0	0
	SEL																								0	0
	STG												D												1	1
FCO	NRO																					D			1	1
	NRR																					D			1	1
FCS	CKM																								0	0
	COP											D													1	1
FDP	ACC																								0	0
	ACF																								0	0
	DAU																								0	0
...																										

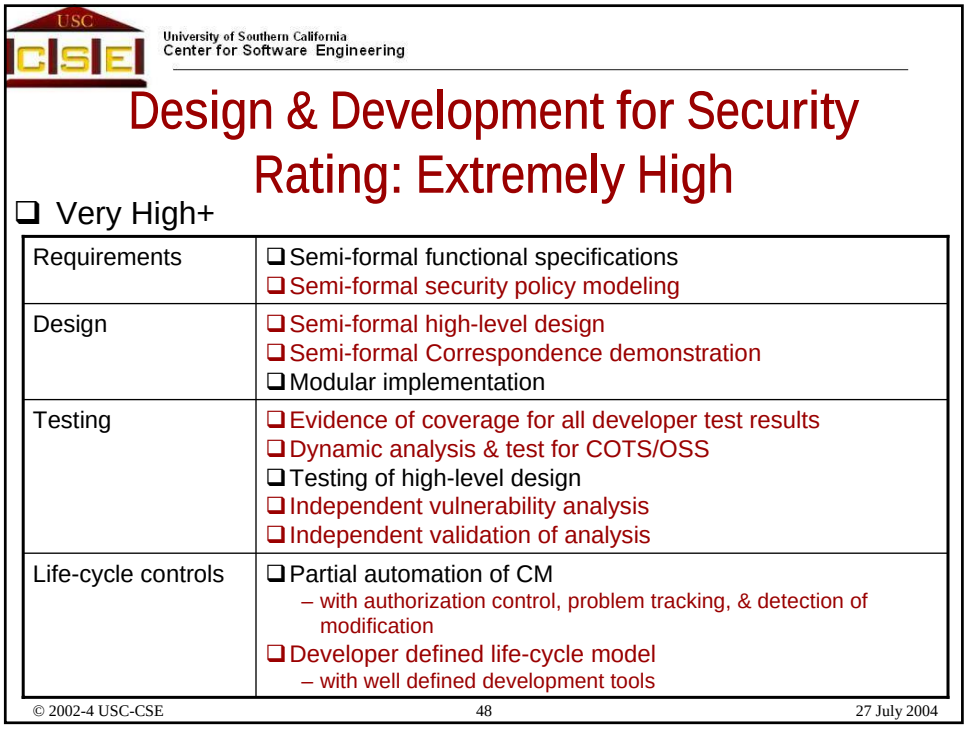
University of Southern California
Center for Software Engineering

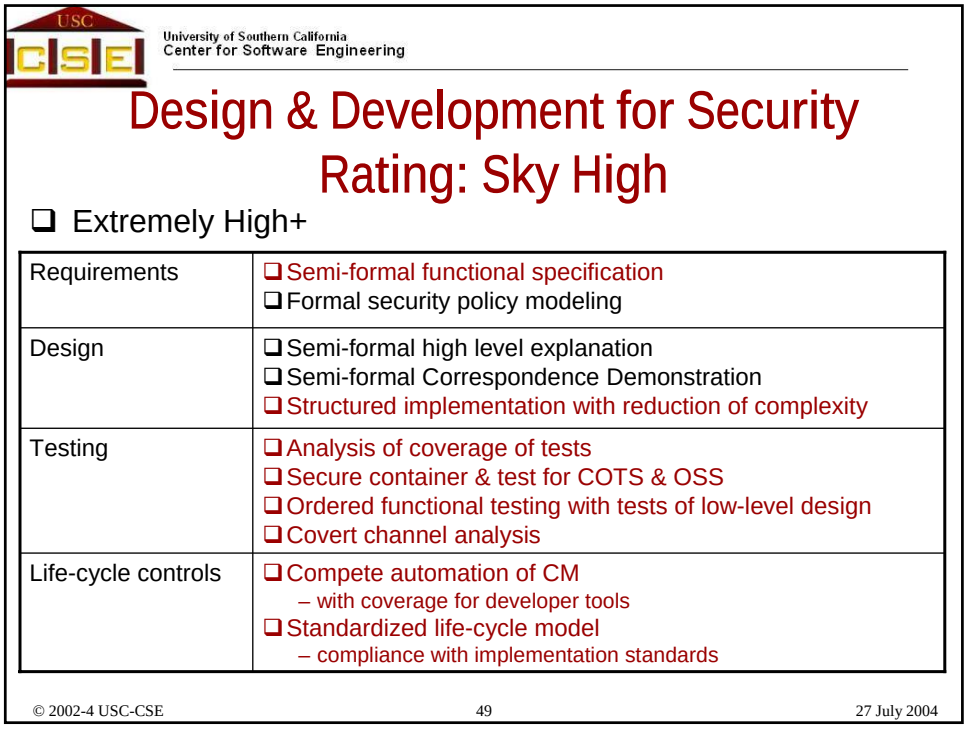
Design & Development for Security Rating : Nominal & High

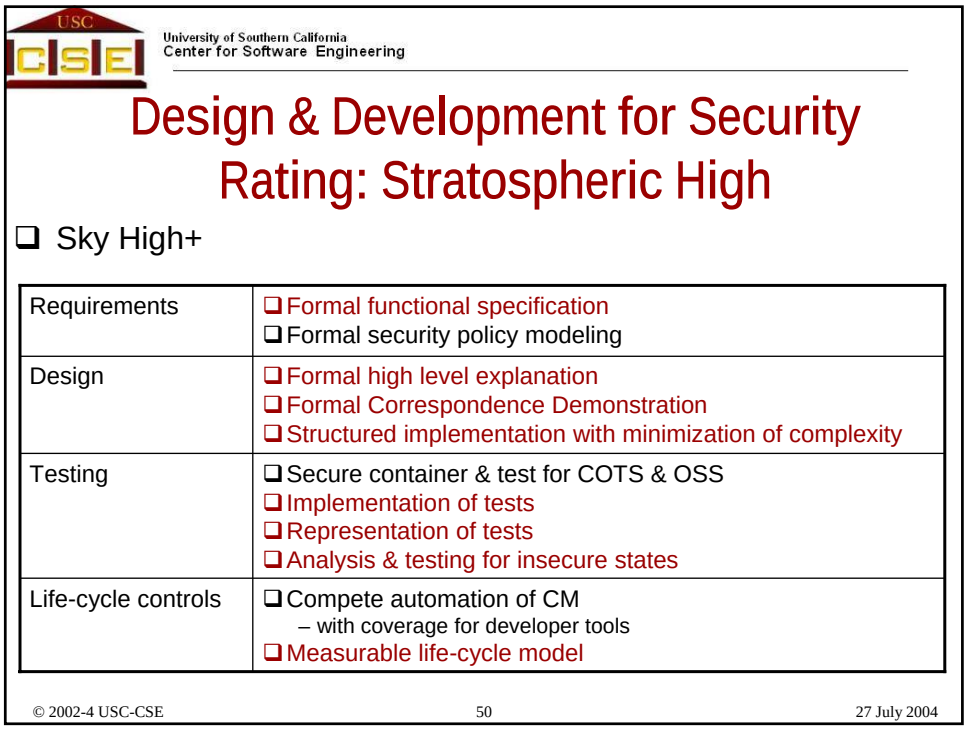
- Nominal
  - No security requirements
  - No protection other than provided by execution environment
- High

Requirements	❑Informal security requirements formulated for system
Design	❑Analysis of security functions using –Informal functional & interface specification –Descriptive high-level design –Informal demonstration of corresponding pairs
Testing	❑Developer tests implementation of requirements –Black box testing
Life-cycle controls	❑Simple Configuration Management with version numbers


University of Southern California
Center for Software Engineering
Design & Development for Security Rating : Very High
❑ High+
Requirements
❑Fully defined external interfaces
❑Informal security policy modeling
Design
❑High-level design enforces security
❑Informal low-level design description
Testing
❑Independent testing of all functional requirements
❑Inspection of COTS/OSS source code if available
❑Developer vulnerability analysis
Life-cycle controls
❑Detailed delivery & installation procedures
–with well-defined security defaults
❑Identification of security measures
© 2002-4 USC-CSE
47
27 July 2004


University of Southern California
Center for Software Engineering
Design & Development for Security Rating: Extremely High
❑ Very High+
Requirements
❑ Semi-formal functional specifications
❑ Semi-formal security policy modeling
Design
❑ Semi-formal high-level design
❑ Semi-formal Correspondence demonstration
❑ Modular implementation
Testing
❑ Evidence of coverage for all developer test results
❑ Dynamic analysis & test for COTS/OSS
❑ Testing of high-level design
❑ Independent vulnerability analysis
❑ Independent validation of analysis
Life-cycle controls
❑ Partial automation of CM
– with authorization control, problem tracking, & detection of modification
❑ Developer defined life-cycle model
– with well defined development tools
© 2002-4 USC-CSE
48
27 July 2004


USC
CSE
University of Southern California
Center for Software Engineering
Design & Development for Security Rating: Sky High
Extremely High+
Requirements
Semi-formal functional specification
Formal security policy modeling
Design
Semi-formal high level explanation
Semi-formal Correspondence Demonstration
Structured implementation with reduction of complexity
Testing
Analysis of coverage of tests
Secure container & test for COTS & OSS
Ordered functional testing with tests of low-level design
Covert channel analysis
Life-cycle controls
Compete automation of CM
– with coverage for developer tools
Standardized life-cycle model
– compliance with implementation standards
© 2002-4 USC-CSE
49
27 July 2004


USC
CSE
University of Southern California
Center for Software Engineering
Design & Development for Security Rating: Stratospheric High
Sky High+
Requirements
Formal functional specification
Formal security policy modeling
Design
Formal high level explanation
Formal Correspondence Demonstration
Structured implementation with minimization of complexity
Testing
Secure container & test for COTS & OSS
Implementation of tests
Representation of tests
Analysis & testing for insecure states
Life-cycle controls
Compete automation of CM
– with coverage for developer tools
Measurable life-cycle model
© 2002-4 USC-CSE
50
27 July 2004

University of Southern California
Center for Software Engineering

Applying Secure Product Taxonomy

- Application
  - Determine new product's security objectives
  - Map objectives to SFR's
  - Determine typical size range
  - Determine typical effort
- Issues:
  - Need to look at impact of re-use
  - Need to determine effect of SFR's on COCOMO drivers
    - Highly accurate estimation will be difficult to achieve
      - Gathering data to validate expert opinion will relatively large number of project
    - May still be useful for early estimation

 51 27 July 2004

University of Southern California
Center for Software Engineering

Formula for Cost of System & of Security

$$E_{total} = E_{Initial/Mission\ Analysis} + E_{Investment\ Analysis} + E_{System\ Engineering} + E_{Dev\ \&\ Imp} + E_{Sys\ of\ Sys\ Integration} + E_{Install/Deployment} + E_{O\&M} + E_{Disposal}$$

$$E_{Dev\ \&\ Imp} = E_{Design\ \&\ Build\ HW} + E_{Design\ \&\ Build\ SW} + E_{Purchased\ Services} + E_{COTS\text{-}Sys} + E_{Env\text{-}Mods\text{-}design} + E_{Bus\text{-}Proc\text{-}Re\text{-}engineering}$$

$$E_{total}\ (Security) = E_{total}\ (with\ security) - E_{total}\ (without\ security)$$

 52 27 July 2004

University of Southern California
Center for Software Engineering

Formula Elements & COCOMO Family

Formula Elements	COCOMO Family Member
$E_{\text{System Engineering}}$	COSYSMO (new)
$E_{\text{design \& build SW}}$	COCOMO-II
	COCOTS
$E_{\text{Sys of Sys Integration}}$	COSoSIMO (new)

 53 27 July 2004

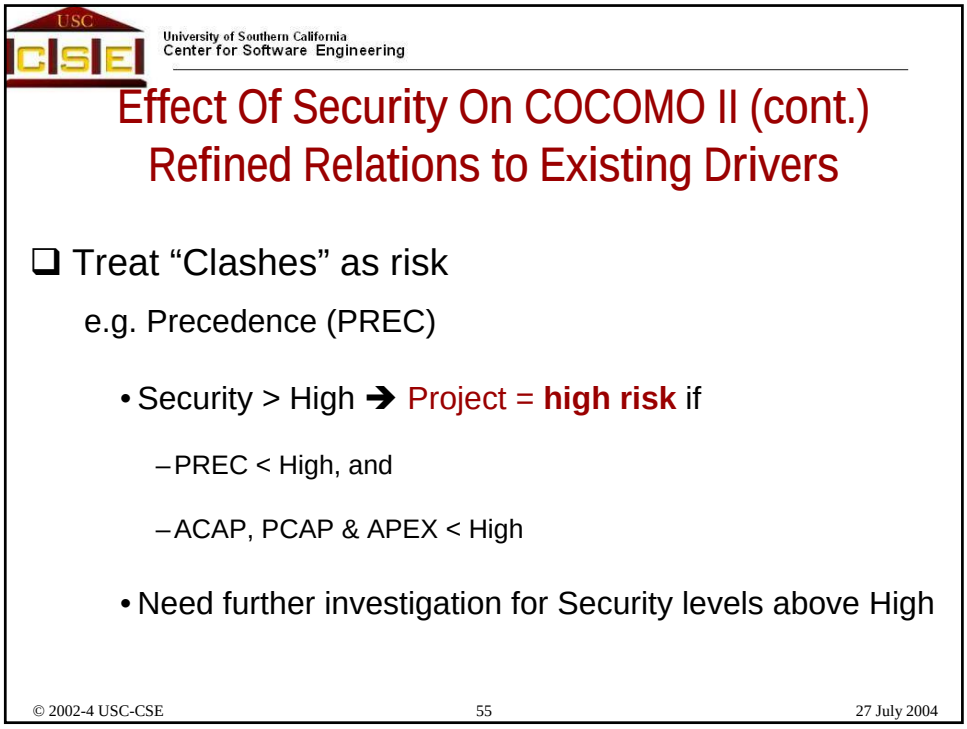
University of Southern California
Center for Software Engineering

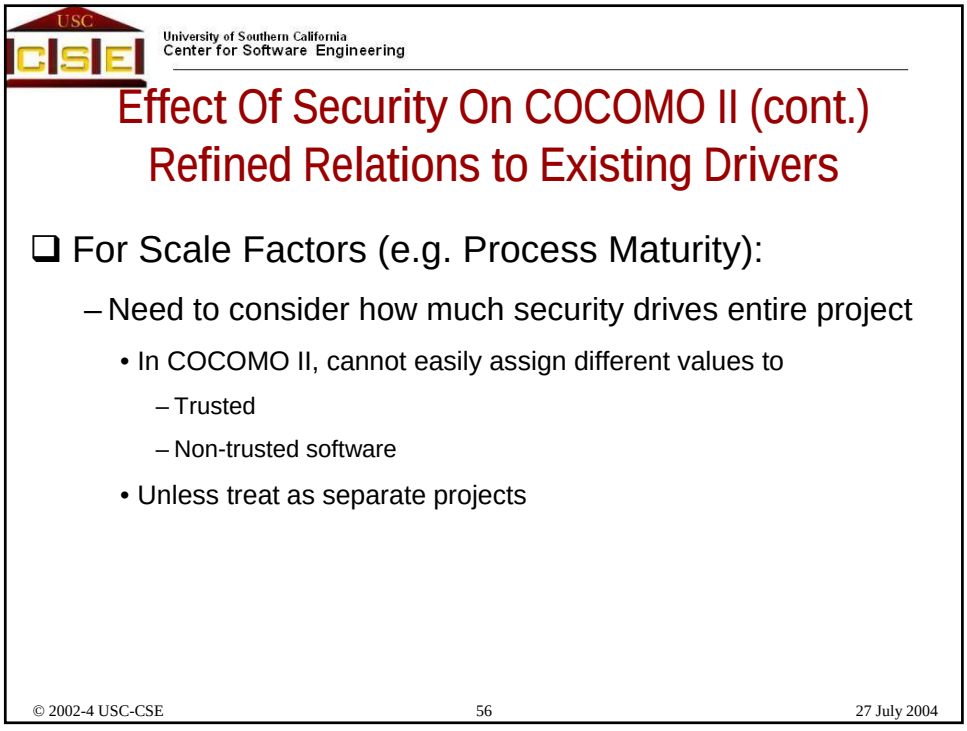
Effect Of Security On COCOMO II (cont.) Relations to Existing Drivers Pre-Workshop

If Security driver rating is >= high & following drivers must be > Nominal

- **PREC** Precedence (team done similar systems)
- **PMAT** Process Maturity
- **TEAM** Team Cohesion
- **RELY** Required software reliability
- **CPLX** Product complexity
- **DOCU** Documentation match to life-cycle needs
- **SITE** Multi-site development
- **TOOL** Use of software tools
- **ACAP** Analyst Capability
- **PCAP** Programmer Capability
- etc.

 54 27 July 2004


University of Southern California
Center for Software Engineering
Effect Of Security On COCOMO II (cont.)
Refined Relations to Existing Drivers
❑ Treat "Clashes" as risk
e.g. Precedence (PREC)
• Security > High ➔ Project = high risk if
– PREC < High, and
– ACAP, PCAP & APEX < High
• Need further investigation for Security levels above High
© 2002-4 USC-CSE
55
27 July 2004


University of Southern California
Center for Software Engineering
Effect Of Security On COCOMO II (cont.)
Refined Relations to Existing Drivers
❑ For Scale Factors (e.g. Process Maturity):
– Need to consider how much security drives entire project
• In COCOMO II, cannot easily assign different values to
– Trusted
– Non-trusted software
• Unless treat as separate projects
© 2002-4 USC-CSE
56
27 July 2004