



Extending COCOMO II to Estimate the Cost of Developing Secure Software

Edward Colbert, Murali Gangadharan

Donald Reifer and Barry Boehm

{ecolbert, murali, boehm}@cse.usc.edu d.reifer@ieee.org

Ground System Architectures Workshops

GSAW2003

March 4 - 6, 2003

Outline

- Why Extend COCOMO II for Security?**
- The COCOMO II Family of Models
- New Security Cost Driver and its Factors
- COCOMO II Cost Driver Values
- Next Steps and Summary

Why Extend COCOMO II for Security



Why Extend COCOMO II for Security

- ❑ Military projects have considered security in developing software since the 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
- ❑ Few cost models (including COCOMO II) include security factors
 - Based 1980s military perspective (Orange Book)
 - Developing secure systems has changed dramatically

Adding Security to Cost Models

- Several approaches for addressing security
 - Modernize existing models
 - Primarily by updating definitions of cost drivers
 - Add new cost drivers to cost models
 - Develop separate security model

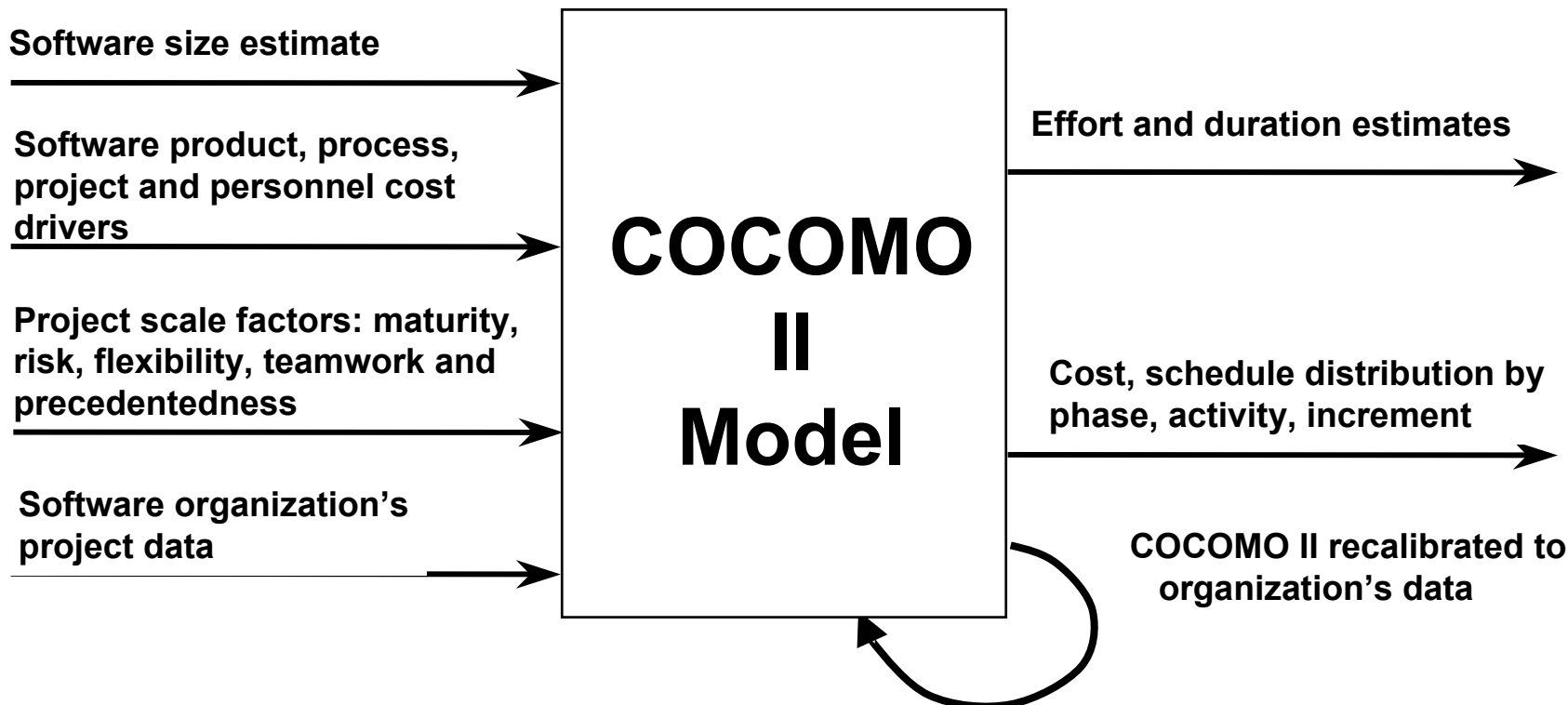
Adding Security to Cost Models

- USC has taken intermediate approach
 - Add factor that addresses security from 3 viewpoints
 - Development
 - Operational
 - Physical
 - Include factors as appropriate to all COCOMO II family cost models
 - Address both commercial & military projects regardless of
 - Size
 - Domain
 - Level of maturity

Outline

- Why Extend COCOMO for Security?
- **The COCOMO II Family of Models**
- New Security Cost Driver and its Factors
- COCOMO II Cost Driver Values
- Next Steps and Summary

COCOMO II Refresher



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)

Outline

- Why Extend COCOMO for Security?
- The COCOMO II Family of Models
- New Security Cost Driver and its Factors**
- COCOMO II Cost Driver Values
- Next Steps and Summary

COCOMO II Security Driver (SECU)

□ Viewpoints

- Physical Security
- Operational Security
- Development for Security

□ Security strategies embraced

- Ad hoc Defense (Low)
- Passive Defense (Nominal)
- Active Defense (High)
- Layered Defense (Very High)
- Defense in Depth (Extremely High)

Security Factors

□ Development for Security

- Effect of processes for development & validation when security a factor

□ Operational Security

- Effect of security policies, processes, tools and facilities that:
 - Permit identification of security events
 - Define subsequent actions to identify key elements
 - Report pertinent information to appropriate individual, group, or process

□ Development Constraints

- Constraints placed on development when protecting software facilities:
 - From outside perimeter to inside office space
 - Includes all of information system resources

Development for Security

Rating : Low & Nominal

Low

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

Nominal

Requirements	<input type="checkbox"/> Informal security requirements formulated for system
Design	<input type="checkbox"/> Analysis of security functions using <ul style="list-style-type: none"> – Informal functional & interface specification – Descriptive high-level design – Demonstration of corresponding pairs
Testing	<input type="checkbox"/> Developer tests implementation of requirements <ul style="list-style-type: none"> – Black box testing
Life-cycle controls	<input type="checkbox"/> Simple Configuration Management (CM) with version numbers



Development for Security

Rating : High

Nominal +

Requirements	<input type="checkbox"/> Fully defined external interfaces <input type="checkbox"/> Informal security policy modeling
Design	<input type="checkbox"/> Security enforcing high-level design <input type="checkbox"/> Informal low-level design description
Testing	<input type="checkbox"/> Independent testing of all functional requirements <input type="checkbox"/> Inspection of COTS source code if available
Life-cycle controls	Detailed delivery & installation procedures <input type="checkbox"/> Identification of security measures for life-cycle



Development for Security

Rating: Extremely High

Very High +

Requirements	<ul style="list-style-type: none"><input type="checkbox"/> Fully defined external interfaces<input type="checkbox"/> Informal security policy modeling
Design	<ul style="list-style-type: none"><input type="checkbox"/> Semi-formal high level explanation<input type="checkbox"/> Structured implementation with reduction of complexity<input type="checkbox"/> Secure container for COTS and Open-source
Testing	<ul style="list-style-type: none"><input type="checkbox"/> Analysis of coverage of tests<input type="checkbox"/> Ordered functional testing with tests of low-level design<input type="checkbox"/> Covert channel analysis
Life-cycle controls	<ul style="list-style-type: none"><input type="checkbox"/> Compete automation of CM<ul style="list-style-type: none">– with coverage for developer tools<input type="checkbox"/> Standardized life-cycle model<ul style="list-style-type: none">– with compliance to implementation standards

Operational Security

Rating: Low & Nominal

Low

- No organization-wide security policies
- Ad-hoc security practices
- Optional firewall & virus protection

Nominal

Administration	<input type="checkbox"/> Basic Security policies <ul style="list-style-type: none"> – inc. <ul style="list-style-type: none"> • Password and Virus Protection policy • Network access and system use policy <input type="checkbox"/> Guidelines for administrators & users
Protection	<input type="checkbox"/> Reasonable practices for <ul style="list-style-type: none"> – Checksum Verification – Software firewall(s) – Operating system logging
Authentication	<input type="checkbox"/> Simple password-based authentication schemes

Operational Security

Rating: Very High

High +

Administration	<input type="checkbox"/> Comprehensive Security policies – inc. <ul style="list-style-type: none"> • Business continuity plans • Disaster recovery plans <input type="checkbox"/> Incident response teams handle security breaches
Protection	<input type="checkbox"/> A layered defense strategy is implemented to protect the system with reasonable practices for <ul style="list-style-type: none"> – Proxy servers – Layered system monitoring with Intrusion Detection Systems
Authentication	<input type="checkbox"/> Digital certificates & signatures used for <ul style="list-style-type: none"> – Authentication – Message integrity – Non-repudiation

Development Constraints Rating Description

Nominal

- None

High

- All source materials are locked up when not in active use

Very High

- High +
 - Audited security markings in code

Extremely High

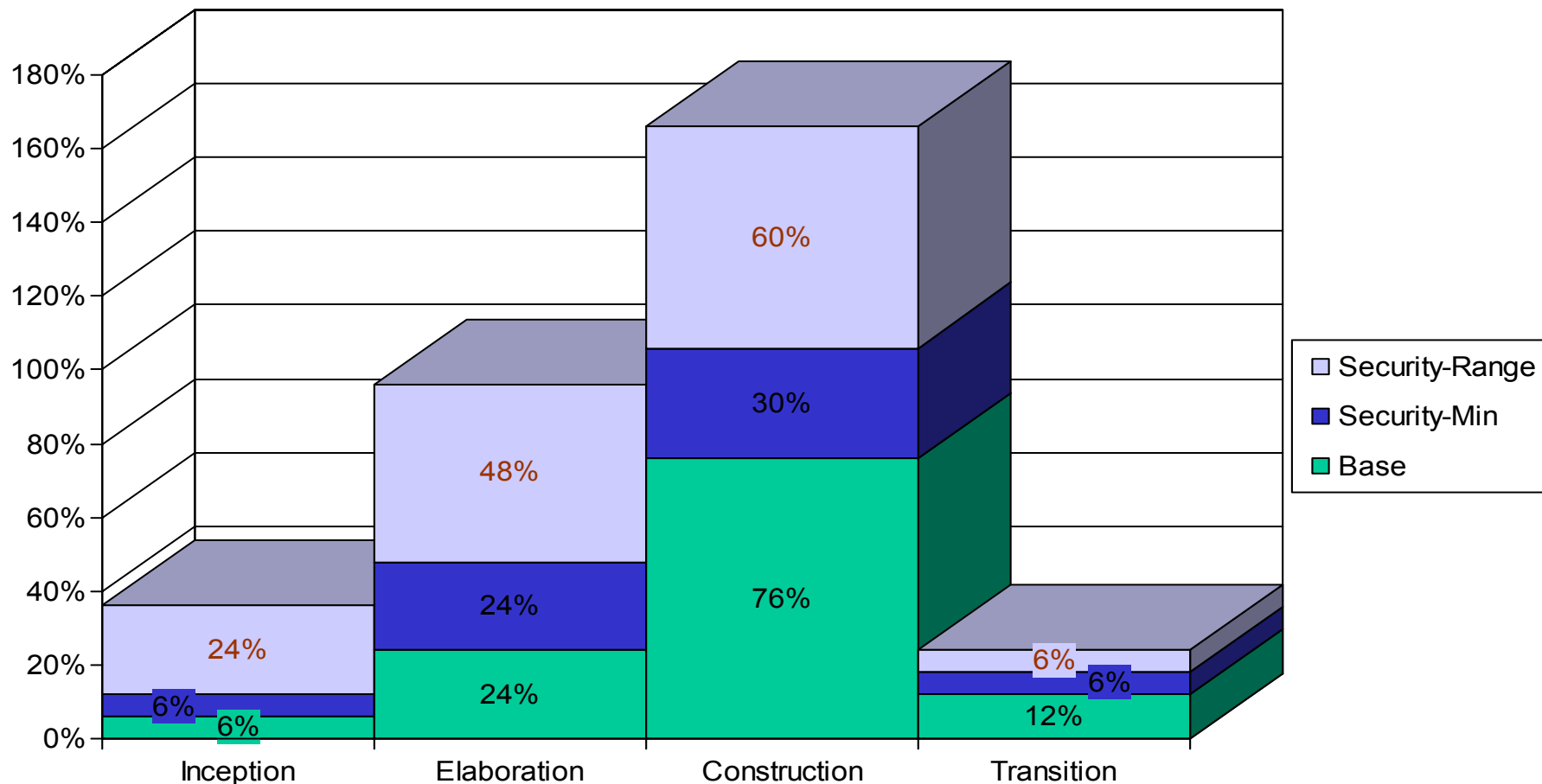
- Very High+
 - Multi-compartment developer communication constraints

Outline

- Why Extend COCOMO for Security?
- The COCOMO II Family of Models
- New Security Cost Driver and its Factors
- **COCOMO II Cost Driver Values**
- Next Steps and Summary

Draft Baseline Security Effort Distribution

☐ With wide standard deviations



Outline

- Why Extend COCOMO for Security?
- The COCOMO II Family of Models
- New Security Cost Driver and its Factors
- COCOMO II Cost Driver Values
- **Next Steps and Summary**

Next Steps

- ❑ Reach consensus on cost drivers
 - FAA Workshop on security-May 03
 - Delphi run & calibration of factors in works
- ❑ Initiate efforts to statistically validate accuracy of model
 - Survey available 160+ COCOMO project data
 - Perform initial calibration
- ❑ Create enhanced COCOMO data collection forms
 - gather security related efforts
- ❑ Compare actual project data to expert opinions
 - Calibrate the model by weighting
 - Actual data
 - Expert opinions using Bayesian statistical techniques

Summary

- ❑ Proposed extensions to COCOMO for development of Secure Systems
 - Based on *Common Criteria*
 - 1 Driver: SECU
 - 3 Factors:
 - Development for Security
 - Operational Security
 - Physical Security
 - Affects on other COCOMO II Drivers
 - RELY, CPLX, DOCU, SITE, TOOL
 - Affects on Size
 - Affects on project risk

- ❑ Hopefully stimulated your interest and motivated you to participate by sharing project data