Costing Secure Systems

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

18th Annual COCOMO II & Software Costing Forum
2003
Goal Of Presentation

- Review proposed
  - New draft model for costing of system security
  - Extensions to COCOMO II for development of secure software systems
  - FAA model

- Kick-off
  - Behavior Analysis
  - Delphi Process
  - Data Search (Collection)
Outline

- Why Costing Model for Secure Systems?
- Approach
- Develop Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
Why Costing Model for Secure Systems

"Amateur testing of our systems do not show us in any way our flaws."

STEPHEN MCHALE, Transportation Security Administration official
Barriers to Effective Computer Security

USA TODAY Snapshots

Cyberattacks become more advanced
What are the barriers to effective computer security in your organization?

- Increasing sophistication of threats: 46%
- Capital expense: 42%
- Lack of time: 41%
- Pace of change or lack of qualified staff: 25%
- Complexity of technology: 24%

Note: More than one answer allowed


By Darryl Haralson and Sam Ward, USA TODAY
Why Extend COCOMO II for Security
“No Time To Relax”, InfoWeek, 2003 July 28

- 76K security incidents reported in first ½ of 2003
  - 82K for all of 2002
  - 80% blended threats
  - According to CERT

- But, business-technology execs are betting that
  - Have built fundamental security infrastructure
  - Just need to execute & keep pace with threats

- 815 professionals responded
  - 58% ranked security as high priority
    - Down from 72% in 2001 survey
  - 39% plan to increase spending on security
    - Down from 49% in 2002
  - 45% of companies few victim of worms or Viruses
    - Down from 70% in 2002
  - 19% hit by denial of service attacks
Why Extend COCOMO II for Security

- **2002 Computer Crime & Security Survey**
  - Conducted by Computer Security Institute (CSI) & San Francisco office of the Federal Bureau of Investigation (FBI)
  - Computer security practitioners in U.S. corporation, government agencies, financial institutions, medical institutions, & university

- **90% of respondents detected breaches last year**
- **80% acknowledged financial lost**
- **Losses = $456B for 44% respondents (223) who were willing to quantify their losses**
  - $20B reported by 20% of respondents in 1997
- **Internet was the main source of attack (74%)**
  - Vast majority used firewalls (90%), antivirus software (90%), access control (82%), physical security (84%), and intrusion detection (60%)
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
Concerns about information–security cost model
» from FAA project workshop, July 2003

— “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
Concerns about information–security cost model
   » from FAA project workshop, July 2003

– Cost estimation relations (CER’s) affected by
  • Threats
  • Environments
  • Solutions

– Different stakeholders have different
  • Problem space
  • Solution space

– Very little calibration data available
Outline

- Why Costing Model for Secure Systems?
- Approach
  - Develop Early Estimation Model
  - Identify Sources of Cost
  - Develop Secure Product Taxonomy
  - Extend COCOMO II
- Next Steps & Summary
COCOMO II Modeling Methodology

1. Analyze Existing Literature
2. Perform Behavioral Analysis
3. Determine Form of model & Identify relative significance of parameters
4. Gather Project Data
5. Determine Bayesian A Posteriori update
6. Gather more data; Refine model
Identify major sources of cost

- To
  - Develop
  - Own

- Including
  - Facilities
  - Equipment
  - People
  - Acquired Systems
  - Services
Develop cost model incrementally

- 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
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
    - October 2003-March 2004
  - Increment 2
    - April 2004-March 2005
  - Increment 3
    - April 2005-September 2006
## Cost Model for System Security Increment 1 (Oct ’03 – Mar ’04)

<table>
<thead>
<tr>
<th>Task Element</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Develop Early Estimation Model</td>
<td>➢ Prototype model</td>
</tr>
<tr>
<td>2. Sources of Cost</td>
<td>➢ Identify, define, scope sources of cost</td>
</tr>
<tr>
<td></td>
<td>➢ Relate sources of cost to FAA WBS</td>
</tr>
<tr>
<td></td>
<td>➢ Recommend type of CER for each</td>
</tr>
<tr>
<td>3. Secure Product Taxonomy</td>
<td>➢ Identify, define, scope product elements</td>
</tr>
<tr>
<td></td>
<td>➢ Relate sources of cost to FAA WBS</td>
</tr>
<tr>
<td>4. COCOMO II Security Extensions</td>
<td>➢ Refine model form and data definitions</td>
</tr>
<tr>
<td>5. COCOTS Security Extensions</td>
<td>➢ Explore security aspects in COCOTS data collection</td>
</tr>
</tbody>
</table>
# Cost Model for System Security Increment 2 (Apr ’04 – Mar ’05)

<table>
<thead>
<tr>
<th>Task Element</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Develop Early Estimation Model</td>
<td>➢ Experimental use &amp; refinement</td>
</tr>
<tr>
<td>2. Sources of Cost</td>
<td>➢ Prioritize sources of cost needing CER’s</td>
</tr>
<tr>
<td></td>
<td>➢ Refine, prototype, experiment with top-priority CER’s</td>
</tr>
<tr>
<td></td>
<td>➢ Relate to scope of COCOMO II security extensions</td>
</tr>
<tr>
<td>3. Secure Product Taxonomy</td>
<td>➢ Experimental use, feedback, and refinement</td>
</tr>
<tr>
<td>4. COCOMO II Security Extensions</td>
<td>➢ Refine, scope, form, definitions based on results of Tasks 1-3</td>
</tr>
<tr>
<td></td>
<td>➢ Experimentally apply to pilot projects, obtain usage feedback</td>
</tr>
<tr>
<td>5. COCOT Security Extensions</td>
<td>➢ Develop initial scope, form, definitions based on results of Tasks 1-4</td>
</tr>
</tbody>
</table>
# Cost Model for System Security Increment 3 (Apr ’05 – Sep ’06)

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

- Why Costing Model for Secure Systems?
- Approach
- Develop Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
Draft Security Effort Distribution

- With large Standard Deviation

* Installation costs are highly dependent on # systems
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)
# Cost Estimation Relations (CER) Example

<table>
<thead>
<tr>
<th>Sample Activity</th>
<th>Preparation for Training</th>
<th>Classroom Training</th>
<th>Periodic Training on new procedures</th>
<th>Software Development</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CER</strong></td>
<td>Activity–based</td>
<td>Unit costing</td>
<td>Analogy-based</td>
<td>Parametric</td>
</tr>
<tr>
<td><strong>Rule</strong></td>
<td>10-20 hours for each Class Hour</td>
<td>N trainers total M trainees</td>
<td>It cost us $XXX last year,…</td>
<td>COCOMO II</td>
</tr>
</tbody>
</table>

It cost us $XXX last year,…
Outline

- Why Costing Model for Secure Systems?
- Approach
- Develop Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
Analyzed to identify where security will affect activities
Intend to expand study to more general WBS
Security Acquisition Types

Different types of acquisition have different WBS, thus have different cost estimation.

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1</td>
<td>Simple purchase of commercial systems</td>
</tr>
<tr>
<td>Type 2</td>
<td>Fully development programs</td>
</tr>
<tr>
<td>Type 3</td>
<td>Programs involving both development and COTS.</td>
</tr>
<tr>
<td>Type 4</td>
<td>Leased services (w/ &amp; w/o levels of security, e.g. FTI).</td>
</tr>
</tbody>
</table>
Sources of Cost

- Facilities
- Equipment
- Personnel
  - Hiring
  - Training
  - Team Cohesion
  - Management
    - Including identifying security polices et.c
  - Security Certification
Sources of Cost (cont.)

- Acquired System
  - License/Purchase
  - Construction
    - Including requirement, development, test, etc
  - Operation
    - Including audit, log, recovery&response, redundancy, etc
  - Maintenance
    - Including HW/SW maintenance, technology refresh, etc
  - Disposition
  - Documentation

- Services
  - Purchase
  - Technology Refresh
### Example of How WBS Affects Acquisition Cost

<table>
<thead>
<tr>
<th>Life Cycle Phase</th>
<th>Work Breakdown Structure</th>
<th>Acquisition Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Type 1</td>
</tr>
<tr>
<td>3.0 Solution Development</td>
<td>3.1 Program management (business-side)</td>
<td>F, P</td>
</tr>
<tr>
<td></td>
<td>3.6 Data and Documentation</td>
<td>F, P, A</td>
</tr>
<tr>
<td>4.0 Implementation</td>
<td>4.1 Program management (business-side)</td>
<td>F, P</td>
</tr>
<tr>
<td></td>
<td>4.2 Engineering, Planning, and Design</td>
<td>F, P, A</td>
</tr>
</tbody>
</table>

**Cost Factor Types:**
- F: Facilities
- P: Personal
- A: Acquisition
- E: Equipment
- S: Services
Outline

- Why Costing Model for Secure Systems?
- Approach
- Develop Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
Security Taxonomy

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

- Security Objectives
  - Preventive
  - Detective
  - Corrective

- Solutions
  - Develop from scratch
  - Use Commercial Product
  - Sign Security Service Contract
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
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
Authentication

- Threats
  - Malicious users can
    - Disguise themselves to other authenticated users
    - Break into machine/system and misrepresent other people’s information

- Approach of treating threats
  - Password
  - Trusted servers distribute certification
    - User can be identified by his certification

- Commercial Product
  - Kerberos, WebID, SecurID, WebSpeed, CloneSafe SAC, NMAS
    (Novell Modular Authentication Services), LDAP Adapter, JAAS, etc
Integrity

- Threats
  - Malicious users may delete or modify data
    - Causes data inconsistency inside the system

- Approach of treating threats
  - Check checksum
  - Check sequences numbers
  - Text check
    - Manual
    - Automatic

- Commercial Product
  - Tripwire, PGP, IMS Data Management Tools, TrueSource
Availability

- Threats
  - Malicious users (Penetrator) may
    - Destroy data
    - Eliminate chance to recovery
    - Consume available resource
      - DoS (Deny of Service Attack)

- Approach of treating threats
  - Provide redundancy and recovery mechanism
    - E.g, making several backups, having redundant servers, giving mirror storage, using more than one name to identify the file, etc

- Commercial Product
  - BIG-IP, AXS-600L, StoneBeat FullCluster, SonicWALL SSL-R
Non-repudiation

 Threats

- Malicious users may
  - Deny having sent vile information
  - Deny having received message to gain extra profit

 Approach of treating threats

- Digital signature
Detective Security Objectives

☐ DSO: Intrusion Detection

– Detect abnormal behaviors in system
– Detect attacks
– Intrusion Detection System
  • System living on host or network to detect and response intrusion
  • Commercial Product
    – NFR, SecureNet, SecureHost, IBM Tivoli Intrusion Manager
Corrective Security Objectives

- **CSO1: Response**
  - Automatic response
    - Active response
    - Passive response
  - Manual response
    - Computer Emergency Quick-Response Team

- **CSO2: Recovery**
  - Use simulated or back-up data to recover corrupted data
    - E.g., rollback
Security Functional Requirements in CC

- **FAU**: Security Audit
- **FCO**: Communication
- **FCS**: Cryptographic support
- **FDP**: User data protection
- **FIA**: Identification and authentication
- **FMT**: Security management
- **FPR**: Privacy
- **FPT**: Protection of the TSF (TOE Security Functions)
- **FRU**: Resource utilization
- **FTA**: TOE (Target of Evaluation) access
- **FTP**: Trusted path/channels
# Map Security Objectives with Common Criteria

<table>
<thead>
<tr>
<th>Security Objective</th>
<th>Common Criteria Security Functional Requirement Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FAU</td>
</tr>
<tr>
<td>Preventive</td>
<td></td>
</tr>
<tr>
<td>PSO1</td>
<td></td>
</tr>
<tr>
<td>PSO2</td>
<td></td>
</tr>
<tr>
<td>PSO3</td>
<td></td>
</tr>
<tr>
<td>PSO4</td>
<td></td>
</tr>
<tr>
<td>PSO5</td>
<td></td>
</tr>
<tr>
<td>Detective</td>
<td></td>
</tr>
<tr>
<td>DSO1</td>
<td></td>
</tr>
<tr>
<td>Corrective</td>
<td></td>
</tr>
<tr>
<td>CSO1</td>
<td></td>
</tr>
<tr>
<td>CSO2</td>
<td></td>
</tr>
</tbody>
</table>

**Common Criteria Security Functional Requirement Classes**

- FAU: False Assumption Uncovered
- FCO: False Claim Objective
- FCS: False Claim Security
- FDP: False Data Protection
- FIA: False Information Assumption
- FMT: False Misuse Tolerance
- FPR: False Privacy Risk
- FPT: False Privacy Tolerance
- FRU: False Requirement Uncovered
- FTA: False Trust Assumption
- FTP: False Trust Port

© 2002-3 USC-CSE
Outline

- Why Costing Model for Secure Systems?
- Approach
- Develop Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
USC has taken intermediate approach

- Add factor that addresses security from 3 viewpoints
  - Development
  - Operational
  - Physical (Development Constraints)

- Include factors as appropriate to all COCOMO II family cost models

- Address both commercial & military projects regardless of
  - Size
  - Domain
  - Level of maturity
COCOMO II Security Driver (SECU)

- **Viewpoints**
  - Design & Development for Security
  - Operational Security
  - Physical Security (Development Constraints)

- **Driver Ratings**
  - Nominal
  - High
  - Very High
  - Extremely High
  - *Sky High*
  - *Stratospheric*  
    
    New COCOMO Levels
Security Factors

- **Design & Development for Security**
  - Effect of design methodologies & processes for development & validation when security is factor

- **Operational Security**
  - Effect of security policies, processes, tools & 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
Design & Development for Security Rating: Nominal & High

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

- High

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Informal security requirements formulated for system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Analysis of security functions using</td>
</tr>
<tr>
<td></td>
<td>- Informal functional &amp; interface specification</td>
</tr>
<tr>
<td></td>
<td>- Descriptive high-level design</td>
</tr>
<tr>
<td></td>
<td>- Informal demonstration of corresponding pairs</td>
</tr>
<tr>
<td>Testing</td>
<td>Developer tests implementation of requirements</td>
</tr>
<tr>
<td></td>
<td>- Black box testing</td>
</tr>
<tr>
<td>Life-cycle controls</td>
<td>Simple Configuration Management with version numbers</td>
</tr>
</tbody>
</table>
# Design & Development for Security Rating: Very High

## Very High

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Fully defined external interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Informal security policy modeling</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design</th>
<th>High-level design enforces security</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Informal low-level design description</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Testing</th>
<th>Independent testing of all functional requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inspection of COTS/OSS source code if available</td>
</tr>
<tr>
<td></td>
<td>Developer vulnerability analysis</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Life-cycle controls</th>
<th>Detailed delivery &amp; installation procedures</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>– with well-defined security defaults</td>
</tr>
<tr>
<td></td>
<td>Identification of security measures</td>
</tr>
</tbody>
</table>
## Design & Development for Security Rating: Extremely High

### Extremely High

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Semi-formal functional specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Semi-formal security policy modeling</td>
</tr>
<tr>
<td>Design</td>
<td>Semi-formal high-level design</td>
</tr>
<tr>
<td></td>
<td>Semi-formal Correspondence demonstration</td>
</tr>
<tr>
<td></td>
<td>Modular implementation</td>
</tr>
<tr>
<td>Testing</td>
<td>Evidence of coverage for all developer test results</td>
</tr>
<tr>
<td></td>
<td>Dynamic analysis &amp; test for COTS/OSS</td>
</tr>
<tr>
<td></td>
<td>Testing of high-level design</td>
</tr>
<tr>
<td></td>
<td>Independent vulnerability analysis</td>
</tr>
<tr>
<td></td>
<td>Independent validation of analysis</td>
</tr>
<tr>
<td>Life-cycle controls</td>
<td>Partial automation of CM</td>
</tr>
<tr>
<td></td>
<td>– with authorization control, problem tracking, &amp; detection of modification</td>
</tr>
<tr>
<td></td>
<td>Developer defined life-cycle model</td>
</tr>
<tr>
<td></td>
<td>– with well defined development tools</td>
</tr>
</tbody>
</table>
Design & Development for Security Rating: Sky 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 |
Design & Development for Security Rating: Stratospheric High

- Stratospheric

| 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  

Existing COCOMO Drivers Affected By Security

- Proposed security cost driver (SECU) constrain existing COCOMO II cost drivers

<table>
<thead>
<tr>
<th>RELY</th>
<th>Required software reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPLX</td>
<td>Product complexity</td>
</tr>
<tr>
<td>DOCU</td>
<td>Documentation match to life-cycle needs</td>
</tr>
<tr>
<td>SITE</td>
<td>Multi-site development</td>
</tr>
<tr>
<td>TOOL</td>
<td>Use of software tools</td>
</tr>
</tbody>
</table>

- Security functions add to project’s size
- Increases project risk
Projects risk is increased when

- Security driver rating is >= high &
- Rating of following drivers are <= low or nominal
  - RELY Required software reliability
  - CPLX Product complexity
  - DOCU Documentation match to life-cycle needs
  - SITE Multi-site development
  - TOOL Use of software tools
Outline

- Why Costing Model for Secure Systems?
- Approach
- Develop Early Estimation Model
- Identify Sources of Cost
- Develop Secure Product Taxonomy
- Extend COCOMO II
- Next Steps & Summary
Next Steps

- **Cost Model for System Security**
  - Refine Early Estimation Model
  - Prototype

- **Sources of Cost for Development Activities**
  - Refine
    - Sources of cost
    - Relation to FAA WBS & Win-Win Spiral WBS
    - CER definitions for each cost

- **Secure Product Taxonomy**
  - Refine
    - Set & definition product elements
    - Relation to FAA WBS & Win-Win Spiral WBS

- **COCOTXS**
  - Explore security aspects in COCOTXS data collection
Next Steps (cont.)

- COCOMO II
  - Reach consensus on cost drivers
    - Delphi run & calibration of factors in works
    - Work on FAA projects
  - 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

- Expanding effort to
  - Development of cost model for secure systems
    - Not just software
  - Full life-cycle
    - Not just development
    - Including
      - 2 years Operation & Maintenance
      - Disposition
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

Hopefully stimulated your interest & motivated you to participate by sharing project data