Extending COCOMO II to Estimate the Cost of Developing Secure Software

Murali Gangadharan
Edward Colbert
{murali, ecolbert}@cse.usc.edu

CSE Annual Research Review 2003
Outline

- Why Extend COCOMO II for Security?
- The COCOMO II Modeling Methodology
- New Security Cost Driver and its Factors
- COCOMO II Cost Driver Values
- Analysis of Security Impact On Other Drivers
- Next Steps and Summary
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 (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
Outline

- Why Extend COCOMO for Security?
- The COCOMO II Modeling Methodology
- New Security Cost Driver and its Factors
- COCOMO II Cost Driver Values
- Analysis of Security Impact On Other Drivers
- Next Steps and Summary
COCOMO II Modeling Methodology

1. Analyze Existing Literature
2. Perform Behavioral Analysis
3. Determine Form of model & Identify relative significance of parameters
4. Perform expert Judgment, Delphi Assessment
5. Gather Project Data
6. Gather more data; Refine model
7. Determine Bayesian A Posteriori update
8. Refine model
Outline

- Why Extend COCOMO for Security?
- The COCOMO II Modeling Methodology
- New Security Cost Driver and its Factors
- COCOMO II Cost Driver Values
- Analysis of Security Impact On Other Drivers
- Next Steps and Summary
COCOMO II Security Driver (SECU)

- Viewpoints
  - Development for Security
  - Operational Security
  - Physical Security (Development Constraints)

- 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**

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Informal security requirements formulated for system</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Design</strong></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>- Demonstration of corresponding pairs</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td>Developer tests implementation of requirements</td>
</tr>
<tr>
<td></td>
<td>- Black box testing</td>
</tr>
<tr>
<td><strong>Life-cycle controls</strong></td>
<td>Simple Configuration Management (CM) with version numbers</td>
</tr>
</tbody>
</table>
Development for Security
Rating : High

- Nominal +

<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>
<tr>
<td>Design</td>
<td>Security enforcing high-level design</td>
</tr>
<tr>
<td></td>
<td>Informal low-level design description</td>
</tr>
<tr>
<td>Testing</td>
<td>Independent testing of all functional requirements</td>
</tr>
<tr>
<td></td>
<td>Inspection of COTS source code if available</td>
</tr>
<tr>
<td>Life-cycle controls</td>
<td>Detailed delivery &amp; installation procedures</td>
</tr>
<tr>
<td></td>
<td>Identification of security measures for life-cycle</td>
</tr>
</tbody>
</table>
## Development for Security

**Rating: Very High**

- **High+**

### Requirements
- Semi-formal functional specifications
- Formal security policy modeling

### Design
- Semi-formal high-level design
- Modular implementation
- Wrapper & dynamic analysis for COTS & Open-source

### Testing
- Evidence of coverage for all developer test results
- 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
## Development for Security

**Rating: Extremely High**

- **Very High** +

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

<table>
<thead>
<tr>
<th>Design</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-formal high level explanation</td>
<td></td>
</tr>
<tr>
<td>Structured implementation with reduction of complexity</td>
<td></td>
</tr>
<tr>
<td>Secure container for COTS and Open-source</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Testing</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis of coverage of tests</td>
<td></td>
</tr>
<tr>
<td>Ordered functional testing with tests of low-level design</td>
<td></td>
</tr>
<tr>
<td>Covert channel analysis</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Life-cycle controls</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Compete automation of CM</td>
<td></td>
</tr>
<tr>
<td>– with coverage for developer tools</td>
<td></td>
</tr>
<tr>
<td>Standardized life-cycle model</td>
<td></td>
</tr>
<tr>
<td>– with compliance to implementation standards</td>
<td></td>
</tr>
</tbody>
</table>
Operational Security  
Rating: Low & Nominal  

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

- Nominal  
  
| Administration | Basic Security policies  
|                |  
|                |  
|                |  
|                |  
|                |  
| Protection     | Reasonable practices for  
|                |  
|                |  
| Authentication | Simple password-based authentication schemes  

- Administration  
  - Basic Security policies  
    - inc.  
      - Password and Virus Protection policy  
      - Network access and system use policy  
  - Guidelines for administrators & users  

- Protection  
  - Reasonable practices for  
    - Checksum Verification  
    - Software firewall(s)  
    - Operating system logging  

- Authentication  
  - Simple password-based authentication schemes
## Operational Security

**Rating:** High

- **Nominal +**

<table>
<thead>
<tr>
<th>Administration</th>
<th>Security policies compliant</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- inc.</td>
</tr>
<tr>
<td></td>
<td>- Incident Response policy</td>
</tr>
<tr>
<td></td>
<td>- Data classification policy</td>
</tr>
<tr>
<td></td>
<td>Documentation &amp; logging of all security incidents</td>
</tr>
<tr>
<td></td>
<td>- Audit trails</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protection</th>
<th>Active defense strategy is implemented to protect the system with reasonable practices for</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Hardware firewalls</td>
</tr>
<tr>
<td></td>
<td>- Virtual Public Networks</td>
</tr>
<tr>
<td></td>
<td>- Active system &amp; network monitoring</td>
</tr>
</tbody>
</table>

| Authentication          | Two factor authentication using passwords & soft-tokens                                   |
## Operational Security Rating: Very High

### Administration

- **Comprehensive Security policies**
  - inc.
    - Business continuity plans
    - Disaster recovery plans
- **Incident response teams handle security breaches**

### Protection

- **A layered defense strategy is implemented to protect the system with reasonable practices for**
  - Proxy servers
  - Layered system monitoring with Intrusion Detection Systems

### Authentication

- **Digital certificates & signatures used for**
  - Authentication
  - Message integrity
  - Non-repudiation
## Operational Security

**Rating**: Extremely High

### Very High +

| Administration | A Formal model of the security policy is created  
|                | Hardware write-protected audit trails to capture forensic evidence |
| Protection     | The defense-in-depth strategy for protection is implemented with reasonable practices for  
|                | – Honey pots  
|                | – Protection against insider and outsider attacks  
|                | Independent vulnerability analysis and penetration tests |
| Authentication | Multi-factor authentication with biometrics |
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
<table>
<thead>
<tr>
<th>Rating</th>
<th>Protect with</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High (VH)</td>
<td>Agents used to dynamically search for potential security breaches as connectivity is established and maintained across system in real-time. All of “High” plus top level anti-tamper and data integrity analysis.</td>
<td>1.45</td>
</tr>
<tr>
<td>High (H)</td>
<td>Hardened wrappers and bindings used to isolate access to operating system and middleware. All of “Nominal” plus lots of vendor liaison to ensure that adequate security protection is built-in the product.</td>
<td>1.29</td>
</tr>
<tr>
<td>Nominal (N)</td>
<td>COTS wrapped and layered for “plug-and-play” to minimize potential threats. COTS scanned both statically and dynamically for information munitions. Vendor error reports checked to ensure that there are no open security holes and all patches have been installed.</td>
<td>1.15</td>
</tr>
<tr>
<td>Low (L)</td>
<td>No protection. COTS scanned statically for viruses, worms, Trojan horses and other information munitions.</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Outline

- Why Extend COCOMO for Security?
- The COCOMO II Modeling Methodology
- New Security Cost Driver and its Factors
- COCOMO II Cost Driver Values
- Analysis of Security Impact On Other Drivers
- Next Steps and Summary
Draft Security Effort Distribution

- **Inception**
  - Security-Range: 6%
  - Security-Min: 6%
  - Base: 24%

- **Elaboration**
  - Security-Range: 24%
  - Security-Min: 24%
  - Base: 48%

- **Construction**
  - Security-Range: 60%
  - Security-Min: 30%
  - Base: 60%

- **Transition**
  - Security-Range: 12%
  - Security-Min: 6%
  - Base: 6%
The values for proposed security cost driver (SECU)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Rating</td>
<td>0.94</td>
<td>1.02</td>
<td>1.27</td>
<td>1.43</td>
<td>1.75</td>
</tr>
<tr>
<td>Range of Rating</td>
<td>0.91 to 1.0</td>
<td>1.0 to 1.05</td>
<td>1.1 to 1.4</td>
<td>1.2 to 1.6</td>
<td>1.4 to 2.0</td>
</tr>
</tbody>
</table>
Outline

- Why Extend COCOMO for Security?
- The COCOMO II Family of Models
- New Security Cost Driver and its Factors
- COCOMO II Cost Driver Values
- Analysis of Security Impact On Other Drivers
- Next Steps and Summary
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 $\geq$ high

- Rating of following drivers are $\leq$ low
  
  - ACAP : Analyst Capability
  - PCAP : Programmer Capability
  - APEX : Application Experience
  - PLEX : Platform Experience
  - LTEX : Language Experience
Outline

- Why Extend COCOMO for Security?
- The COCOMO II Modeling Methodology
- New Security Cost Driver and its Factors
- COCOMO II Cost Driver Values
- Analysis of Security Impact On Other Drivers
- 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 and DITSCAP
  - 1 Driver: SECU
  - 3 Factors:
    • Development for Security
    • Operational Security
    • Physical Security (Development Constraints)
  - 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