

## The Systems *Security* Engineering Capability Maturity Model

<http://www.sse-cmm.org/>

1

## What is security engineering?

- Security engineering, or aspects thereof, attempts to:
  - establish a balanced set of security needs
  - transform security needs into security guidance
  - establish confidence in the correctness and effectiveness of security mechanisms
  - judge that operational impacts due to residual security vulnerabilities are tolerable
  - integrate all aspects into a combined understanding of the trustworthiness of a system

2

## Current state of affairs

- Security products come to market through:
  - lengthy and expensive evaluation or
  - no evaluation
- Results:
  - technology growth more rapid than its assimilation
  - unsubstantiated security claims

3

## What is needed?

- **Continuity** (of use of previous knowledge)
- **repeatability** (of successful previous efforts)
- **efficiency** (of developers and evaluators)
- **assurance** (of security needs being addressed)

4

## One Potential Solution

- Can knowing something about the organization or individual provide a solution?
- Examples:
  - ISO 9000
  - Certification of Information System Security Professionals (CISSP)
  - Capability Maturity Model (CMM)
  - Past Performance

5

## Why was SSE-CMM developed?

### *Objective*

- advance security engineering as a defined, mature, and measurable discipline

### *Project Goal*

- Develop a mechanism to enable:
  - selection of appropriately qualified sec. eng. providers
  - focused investments in sec. engineering practices
  - capability-based assurance

### *Why the CMM approach?*

- accepted way of improving process capability
- increasing use in acquisition as indicator of process capability

6

## Why are Maturity Levels important?

### Maturity Levels (as defined in Capability Maturity Models)

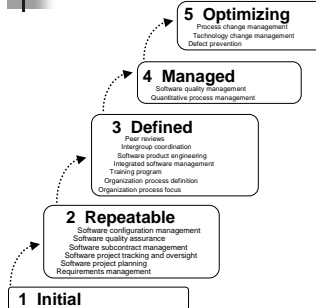
- define ordinal scale for measuring / evaluating process capability
- define incremental steps for improving process capability

*Maturity Levels Discriminate Process Capability*

7

## How do CMMs define Maturity?

### Staged Capability Maturity Model



8

## How do CMMs define Maturity?

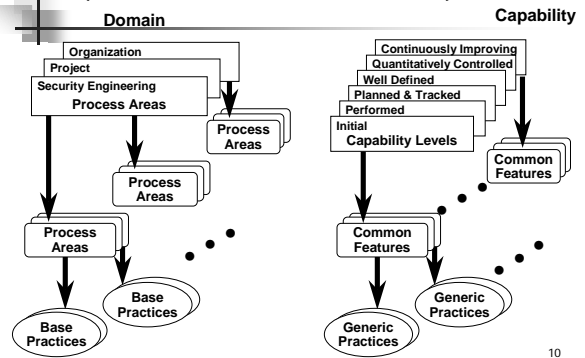
### Two aspects (dimensions):

- the domain
  - process areas
  - base practices
- the organization
  - institutionalization of process areas
  - implementation of process areas

9

## SSE-CMM Model Architecture

(based on SE-CMM Architecture)



10

## SSE-CMM Architecture

(Capability Aspect)



Implementation or institutionalization practices that enhance the capability to perform any process

Set of practices that address the same aspect of process management or institutionalization

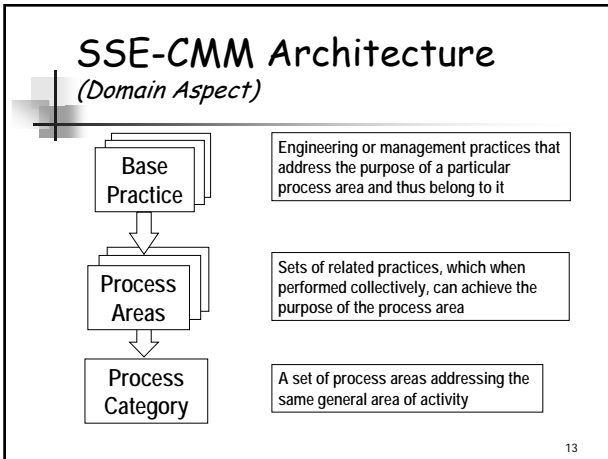
A set of common features that work together to provide a major enhancement in the capability to perform a process

11

## Capability Levels and Common Features

- |  |   |
|--|---|
| <p><b>0 INITIAL</b></p> <p><b>1 PERFORMED INFORMALLY</b></p> <ul style="list-style-type: none"> <li>Base practices performed</li> <li>SW process ad-hoc</li> <li>Success depends on individual effort</li> </ul> <p><b>2 PLANNED &amp; TRACKED</b></p> <ul style="list-style-type: none"> <li>Planning performance</li> <li>Disciplined performance</li> <li>Verifying performance</li> <li>Tracking performance</li> </ul> <p><b>3 WELL-DEFINED</b></p> <ul style="list-style-type: none"> <li>Defining a standard process</li> <li>Perform the defined process</li> <li><i>Coordinate practices</i></li> </ul> | <p><b>4 QUANTITATIVELY CONTROLLED</b></p> <ul style="list-style-type: none"> <li>Establishing measurable quality goals</li> <li>Objectively managing performance</li> </ul> <p><b>5 CONTINUOUSLY IMPROVING</b></p> <ul style="list-style-type: none"> <li>Improving organizational capability</li> <li>Improving process effectiveness</li> </ul> |
|--|---|
- Note: Capability Levels and Common Features are taken from the SE-CMM; Italics indicate SSE-CMM additional Common Feature

12



- ### Security Engineering Process Areas
- Administer System Security Controls
  - Assess Operational Security Risk
  - Attack Security
  - Build Assurance Argument
  - Coordinate Security
  - Determine Security Vulnerabilities
  - Monitor System Security Posture
  - Provide Security Input
  - Specify Security Needs
  - Verify and Validate Security
- 14

- ### Administer System Security Controls
- **Goals:**
    - Security controls are properly configured and used
  - **Base Practices:**
    - Establish security responsibilities
    - Manage security configuration
    - Manage security awareness, training, and education programs
    - Manage security services and control mechanisms
- 15

- ### Assess Operational Security Risk
- **Goals:**
    - An understanding of the security risk associated with operating the system within a defined environment is reached
  - **Base Practices:**
    - Select risk analysis method
    - Prioritize operational capabilities and assets
    - Identify threats
    - Assess operational impacts
- 16

- ### Attack Security
- **Goals:**
    - System vulnerabilities are identified and their potential for exploitation is determined.
  - **Base Practices:**
    - Scope attack
    - Develop attack scenarios
    - Perform attacks
    - Synthesize attack results
- 17

- ### Build Assurance Argument
- **Goals:**
    - The work products and processes clearly provide the evidence that the customer's security needs have been met.
  - **Base Practices:**
    - Identify assurance objectives
    - Define assurance strategy
    - Control assurance evidence
    - Analyze evidence
    - Provide assurance argument
- 18

## Coordinate Security

- **Goals:**
  - All members of the project team are aware of and involved with security engineering activities to the extent necessary to perform their functions.
  - Decisions and recommendations related to security are communicated and coordinated.
- **Base Practices:**
  - Define coordination objectives
  - Identify coordination mechanisms
  - Facilitate coordination
  - Coordinate security decisions and recommendations

19

## Determine Security Vulnerabilities

- **Goals:**
  - An understanding of system security vulnerabilities is reached.
- **Base Practices:**
  - Select vulnerability analysis method
  - Analyze system assets
  - Identify threats
  - Identify vulnerabilities
  - Synthesize system vulnerability

20

## Monitor System Security Posture

- **Goals:**
  - Both internal and external security related events are detected and tracked.
  - Incidents are responded to in accordance with policy.
  - Changes to the operational security posture are identified and handled in accordance with security objectives.
- **Base Practices:**
  - Analyze event records
  - Monitor changes
  - Identify security incidents
  - Monitor security safeguards

21

## Provide Security Input

- **Goals:**
  - All system issues are reviewed for security implications and are resolved in accordance with security goals.
  - All members of the project team have an understanding of security so they can perform their functions.
  - The solution reflects the security input provided.
- **Base Practices:**
  - Understand security input needs
  - Determine constraints and considerations
  - Identify security alternatives
  - Analyze security of engineering alternatives
  - Provide security engineering guidance
  - Provide operational security guidance

22

## Specify Security Needs

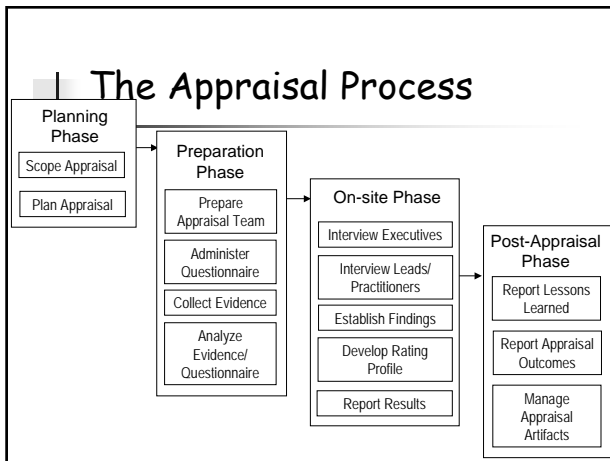
- **Goals:**
  - A common understanding of security needs is reached between all applicable parties, including the customer.
- **Base Practices:**
  - Gain an understanding of customer security needs
  - Identify applicable laws, policies, standards, and constraints
  - Identify system security context
  - Capture security view of system operation
  - Capture security high-level goals
  - Define security related requirements
  - Obtain agreement on security

23

## Verify and Validate Security

- **Goals:**
  - Solutions meet security requirements
  - Solutions meet the customer's operational security needs.
- **Base Practices:**
  - Identify verification and validation targets
  - Define verification and validation approach
  - Perform verification
  - Perform validation
  - Provide verification and validation results

24



## Use by Security Evaluation Organizations

- **Alternative to extensive evaluation/re-evaluation**
  - confidence in integration of security engineering with other disciplines
  - confidence in end results
- **Issues**
  - Does not guarantee good results
  - need to ensure uniform appraisals
  - need good understanding of model and its use
  - Does not eliminate need for testing/evaluation
  - how does it actually contribute to assurance

26

## Use by Acquirers

- Standard RFP language and bidder evaluation
- Understanding programmatic risks
- Avoid protests (uniform assessments)
- Greater level of confidence in end results
- **Issues**
  - Does not guarantee good results
  - need to ensure uniform appraisals
  - need good understanding of model and its use

27