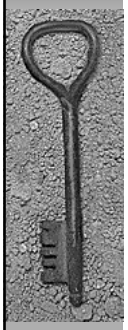
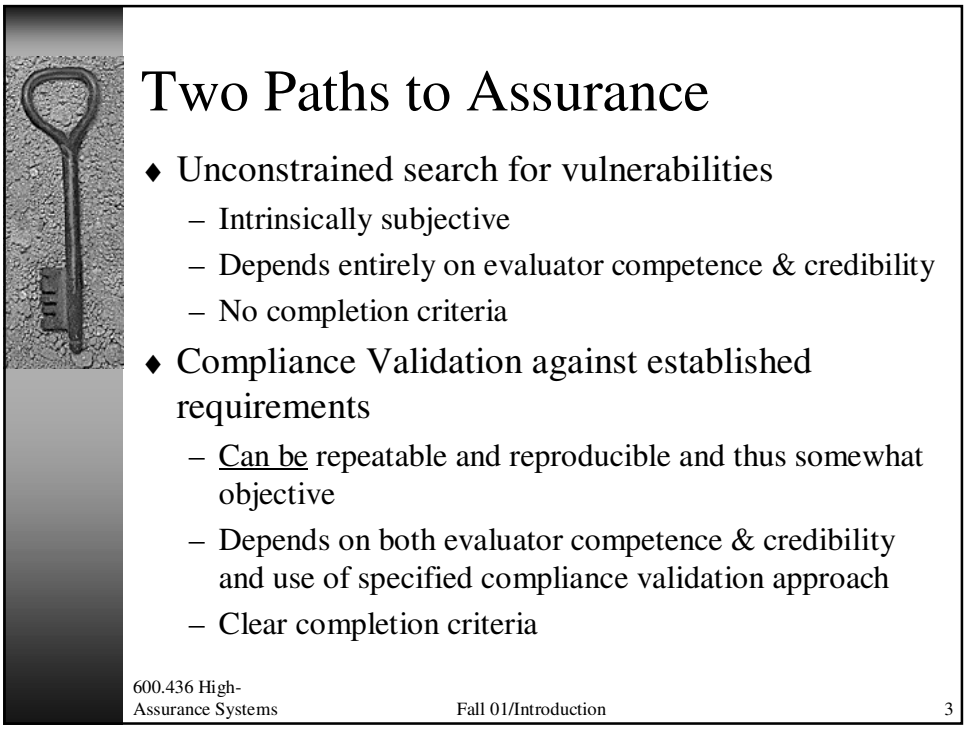
Security Assurance: Purpose, Product, and Proof

- The User's Role in Security Assurance (Purpose)
- The Developer's Role in Security Assurance (Product)
- The Evaluator's Role in Security Assurance (Proof)
- Matching Purpose to Product to Proof



From First Week… Informal Definition: "Assurance"

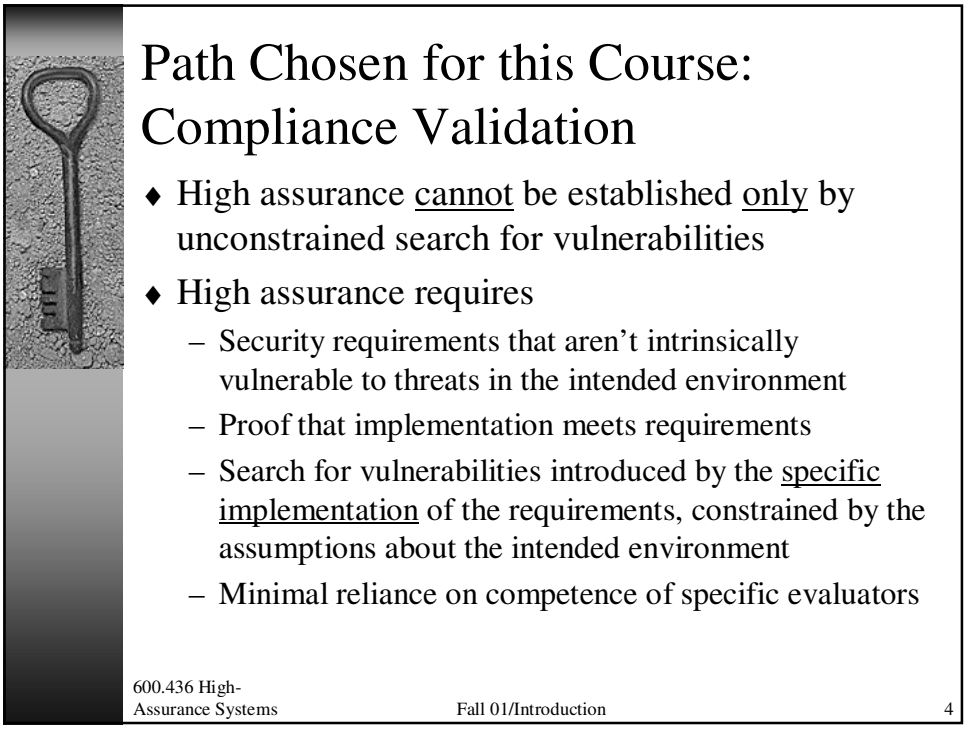
- Assurance is the process by which one obtains confidence in the security that a software system will enforce when operated correctly.
- This includes the policies enforced, the degree of confidence in the enforcement, and an assessment of the *appropriateness* of those policies for the context in which the system will be used.



Two Paths to Assurance

- Unconstrained search for vulnerabilities
  - Intrinsically subjective
  - Depends entirely on evaluator competence & credibility
  - No completion criteria
- Compliance Validation against established requirements
  - Can be repeatable and reproducible and thus somewhat objective
  - Depends on both evaluator competence & credibility and use of specified compliance validation approach
  - Clear completion criteria

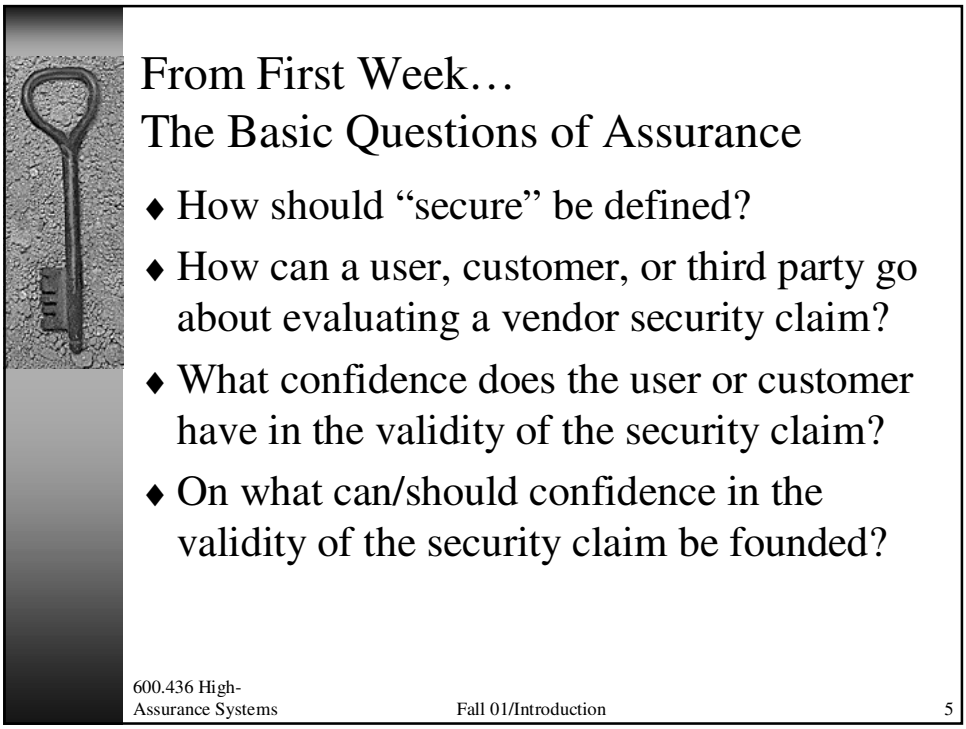
600.436 High-Assurance Systems Fall 01/Introduction 3



Path Chosen for this Course: Compliance Validation

- High assurance cannot be established only by unconstrained search for vulnerabilities
- High assurance requires
  - Security requirements that aren't intrinsically vulnerable to threats in the intended environment
  - Proof that implementation meets requirements
  - Search for vulnerabilities introduced by the specific implementation of the requirements, constrained by the assumptions about the intended environment
  - Minimal reliance on competence of specific evaluators

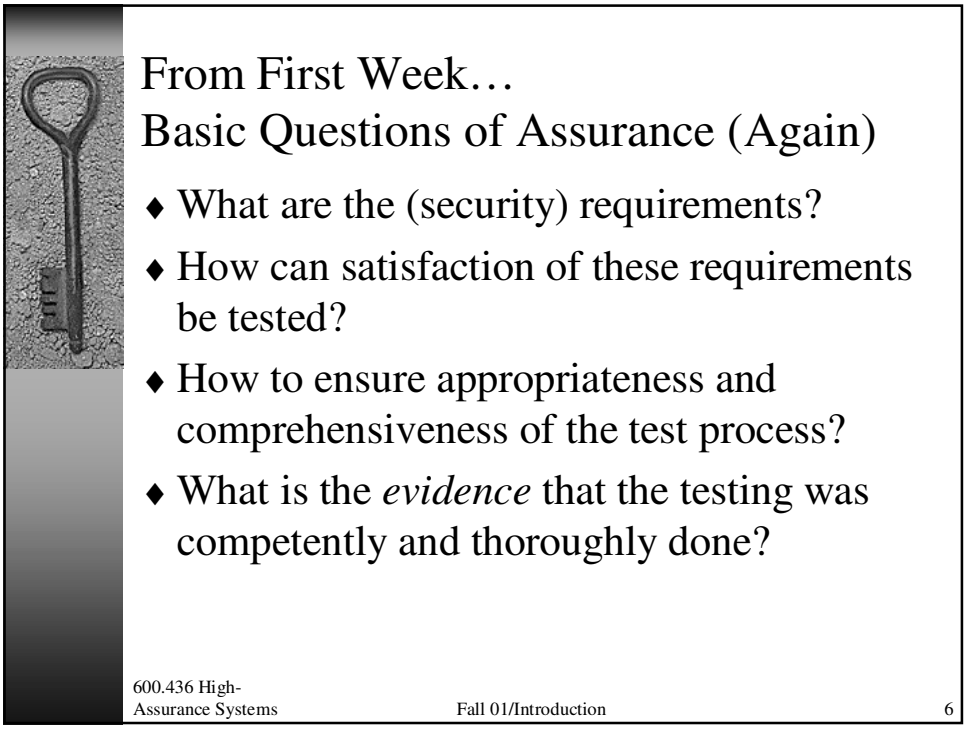
600.436 High-Assurance Systems Fall 01/Introduction 4



From First Week… The Basic Questions of Assurance

- How should "secure" be defined?
- How can a user, customer, or third party go about evaluating a vendor security claim?
- What confidence does the user or customer have in the validity of the security claim?
- On what can/should confidence in the validity of the security claim be founded?

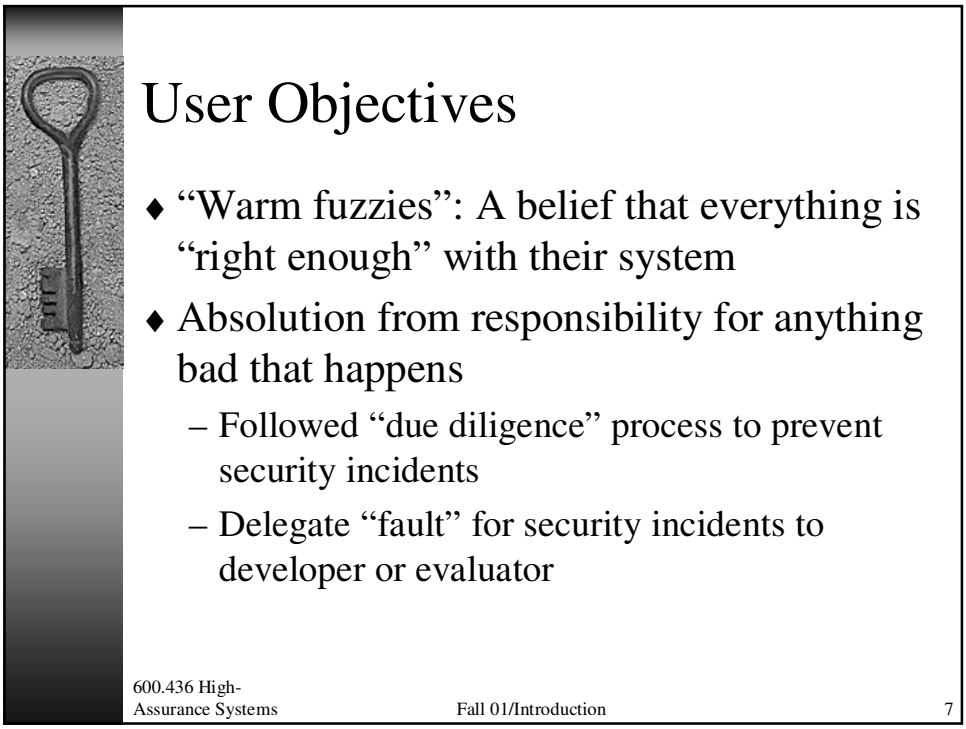
600.436 High-Assurance Systems Fall 01/Introduction 5



From First Week… Basic Questions of Assurance (Again)

- What are the (security) requirements?
- How can satisfaction of these requirements be tested?
- How to ensure appropriateness and comprehensiveness of the test process?
- What is the *evidence* that the testing was competently and thoroughly done?

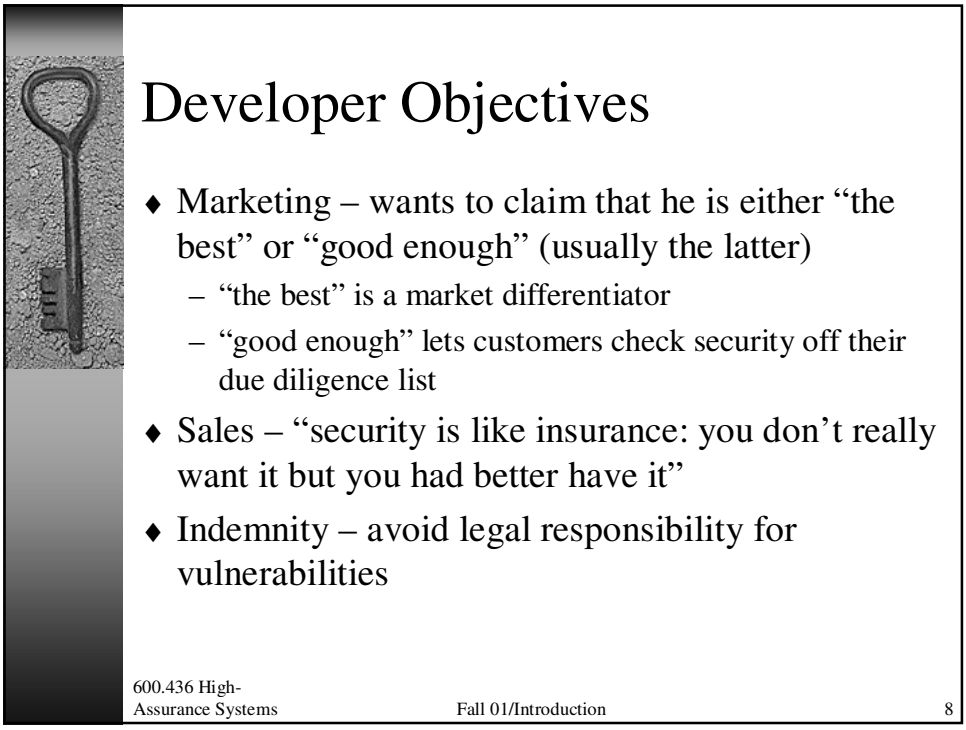
600.436 High-Assurance Systems Fall 01/Introduction 6



User Objectives

- "Warm fuzzies": A belief that everything is "right enough" with their system
- Absolution from responsibility for anything bad that happens
  - Followed "due diligence" process to prevent security incidents
  - Delegate "fault" for security incidents to developer or evaluator

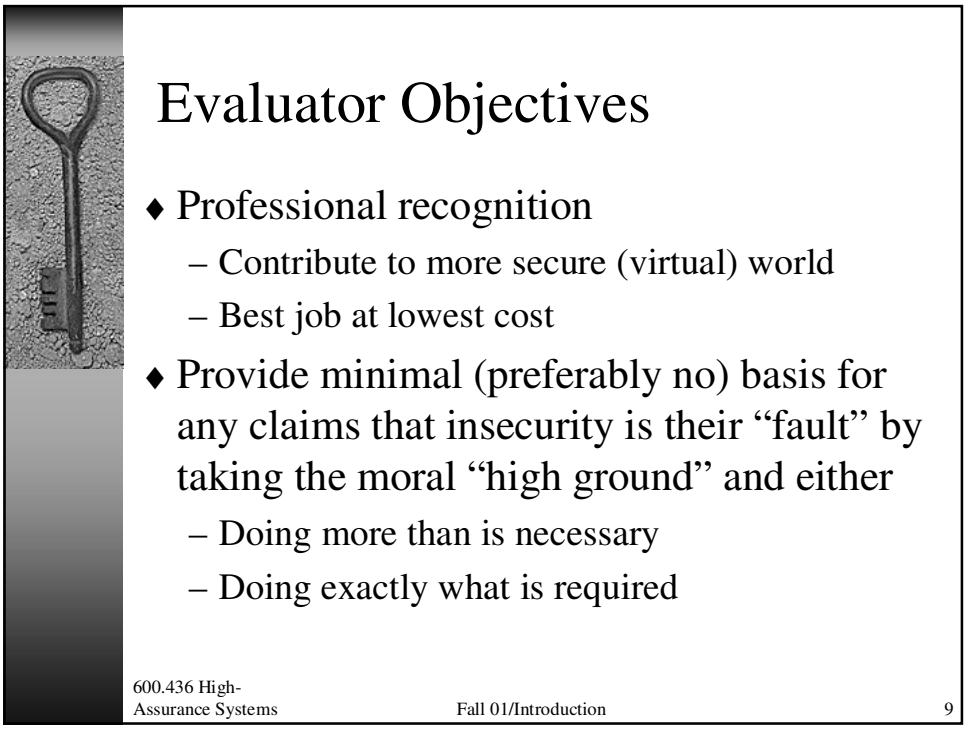
600.436 High-Assurance Systems Fall 01/Introduction 7



Developer Objectives

- Marketing – wants to claim that he is either "the best" or "good enough" (usually the latter)
  - "the best" is a market differentiator
  - "good enough" lets customers check security off their due diligence list
- Sales – "security is like insurance: you don't really want it but you had better have it"
- Indemnity – avoid legal responsibility for vulnerabilities

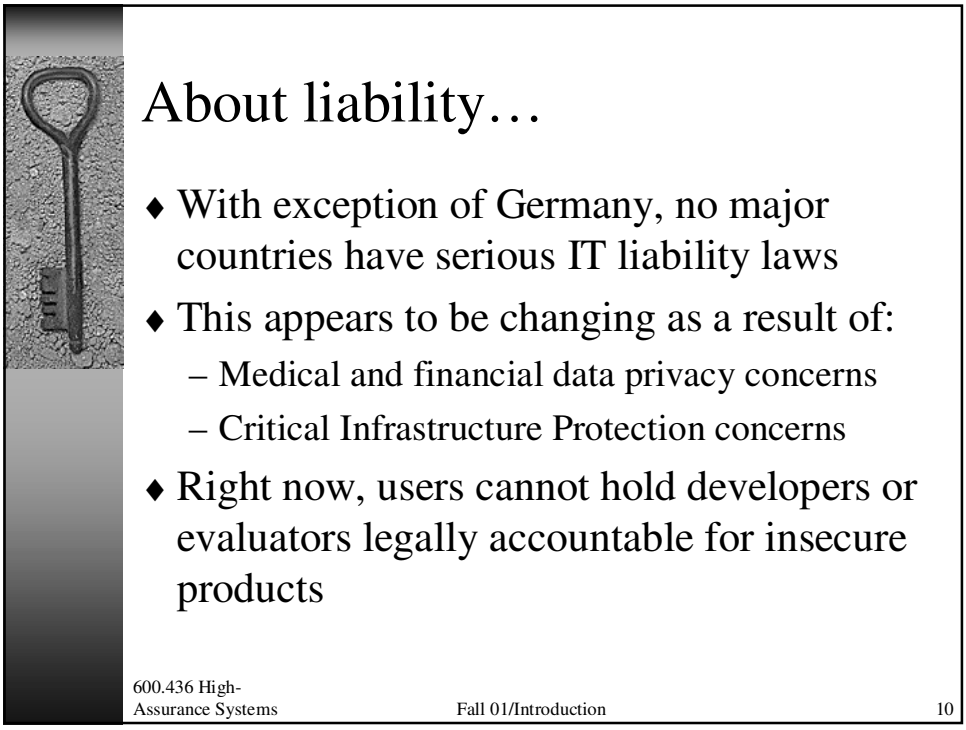
600.436 High-Assurance Systems Fall 01/Introduction 8



Evaluator Objectives

- Professional recognition
  - Contribute to more secure (virtual) world
  - Best job at lowest cost
- Provide minimal (preferably no) basis for any claims that insecurity is their "fault" by taking the moral "high ground" and either
  - Doing more than is necessary
  - Doing exactly what is required

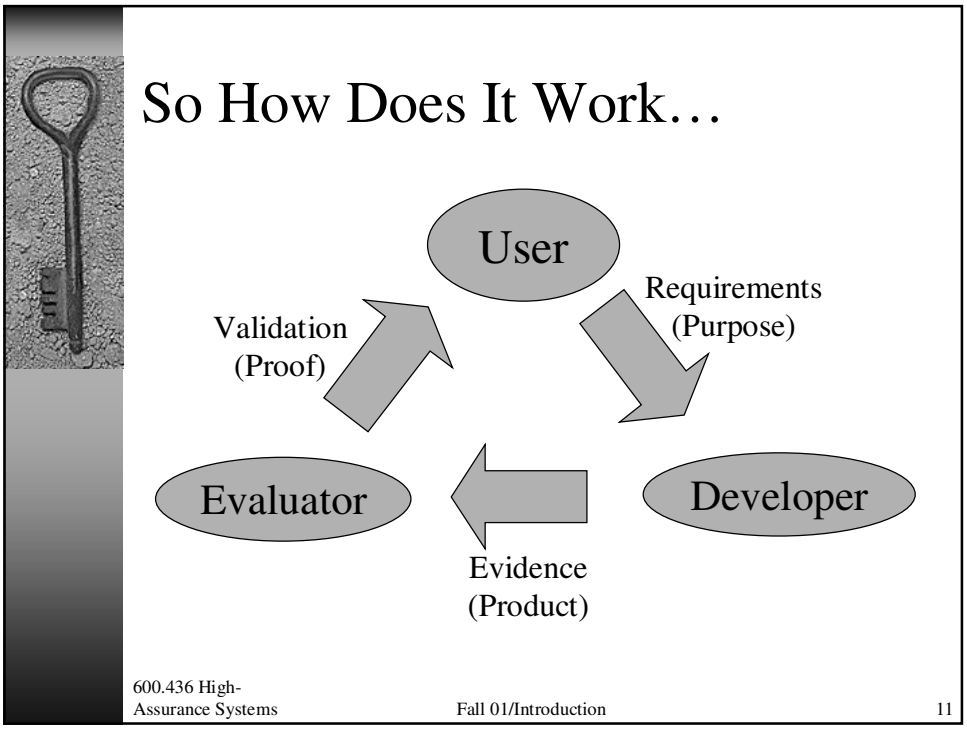
600.436 High-Assurance Systems Fall 01/Introduction 9

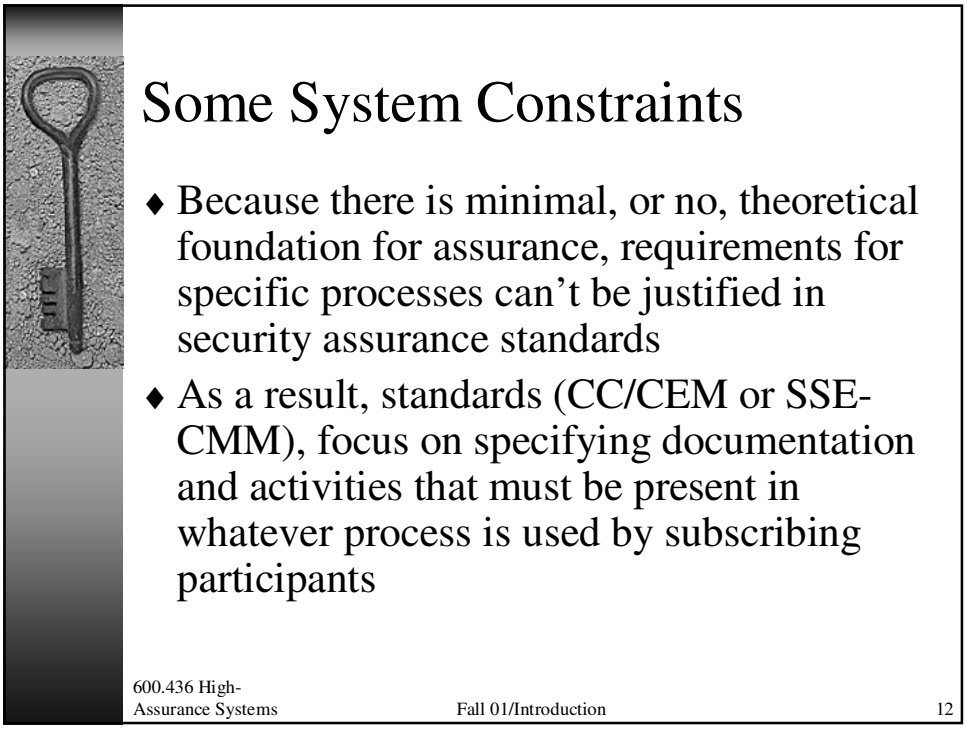


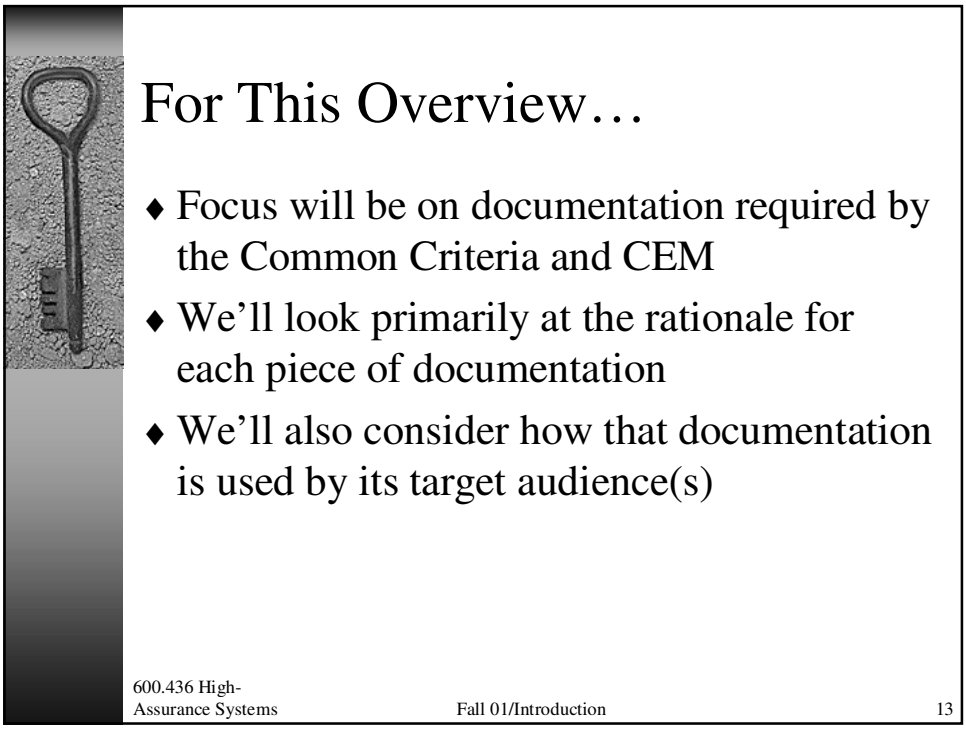
About liability…

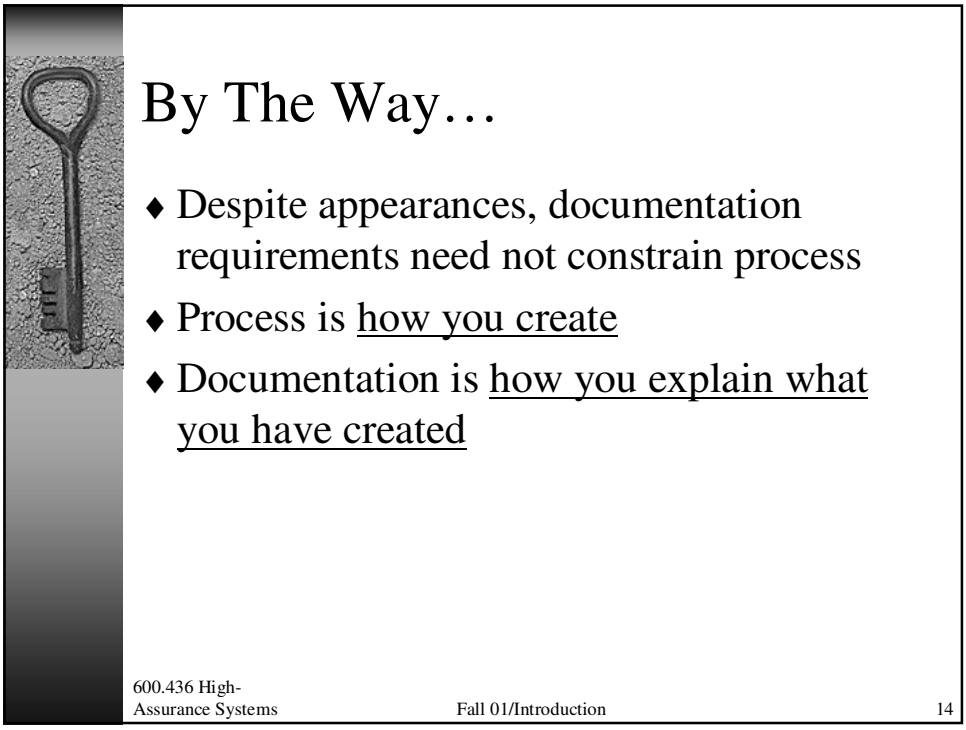
- With exception of Germany, no major countries have serious IT liability laws
- This appears to be changing as a result of:
  - Medical and financial data privacy concerns
  - Critical Infrastructure Protection concerns
- Right now, users cannot hold developers or evaluators legally accountable for insecure products

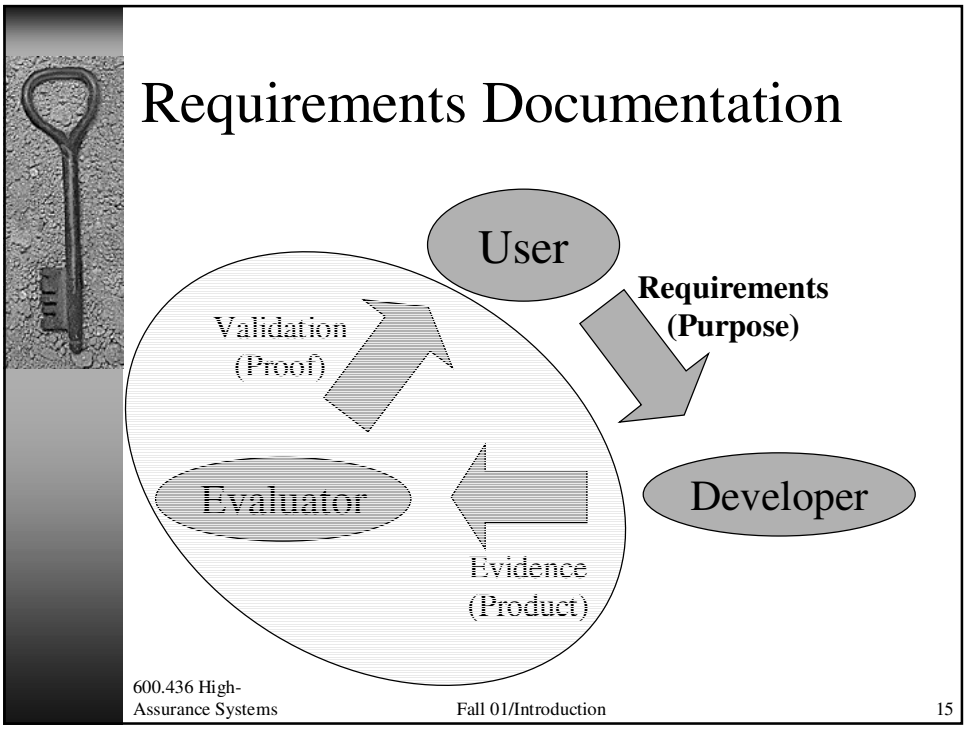
600.436 High-Assurance Systems Fall 01/Introduction 10

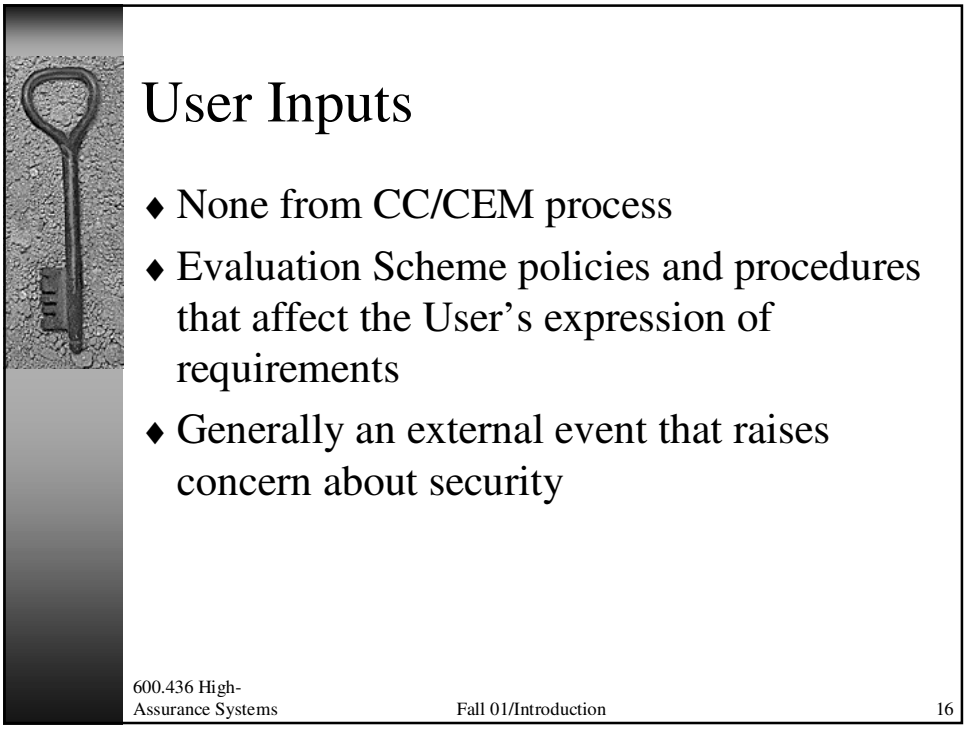

So How Does It Work…
User
Requirements
(Purpose)
Validation
(Proof)
Evaluator
Developer
Evidence
(Product)
600.436 High-
Assurance Systems
Fall 01/Introduction
11

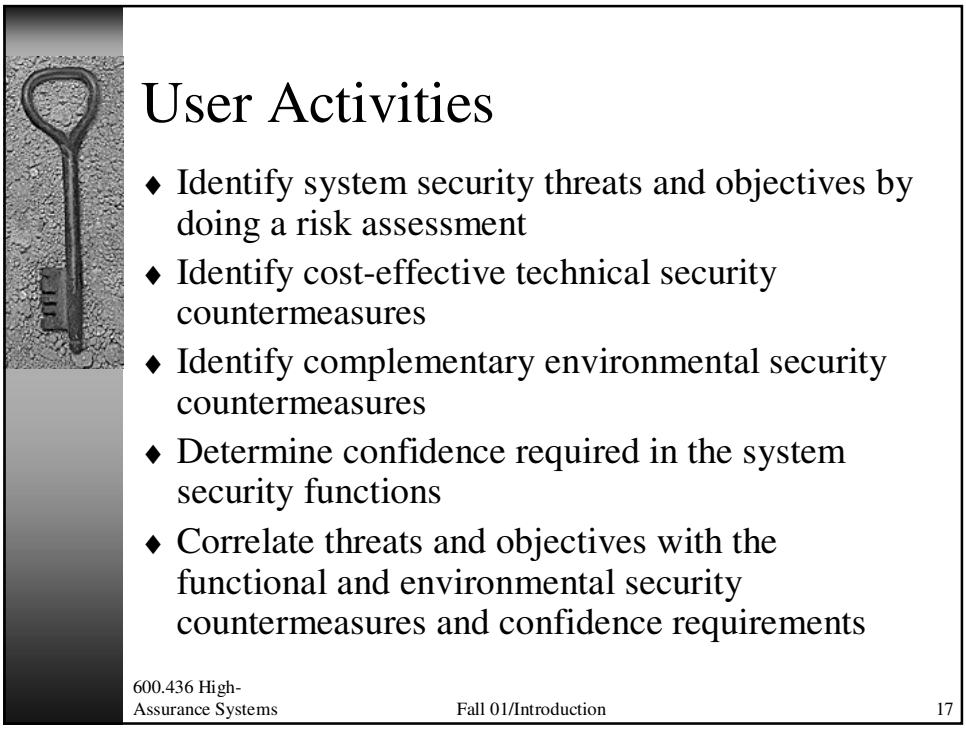

Some System Constraints
♦ Because there is minimal, or no, theoretical foundation for assurance, requirements for specific processes can't be justified in security assurance standards
♦ As a result, standards (CC/CEM or SSE-CMM), focus on specifying documentation and activities that must be present in whatever process is used by subscribing participants
600.436 High-
Assurance Systems
Fall 01/Introduction
12

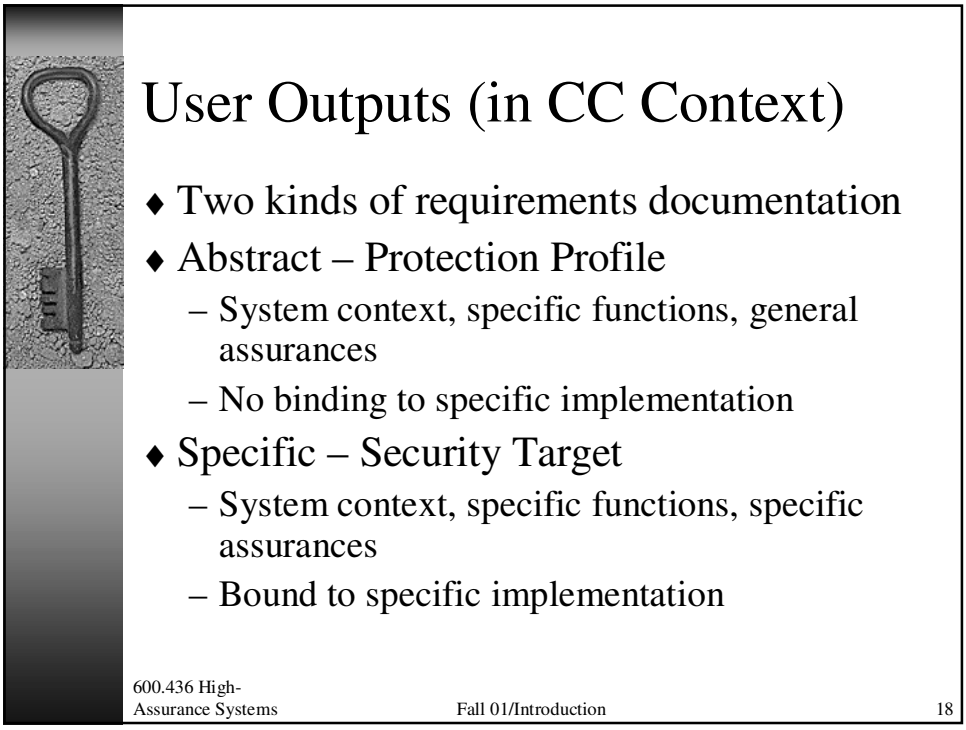

For This Overview…
♦ Focus will be on documentation required by the Common Criteria and CEM
♦ We'll look primarily at the rationale for each piece of documentation
♦ We'll also consider how that documentation is used by its target audience(s)
600.436 High-Assurance Systems
Fall 01/Introduction
13

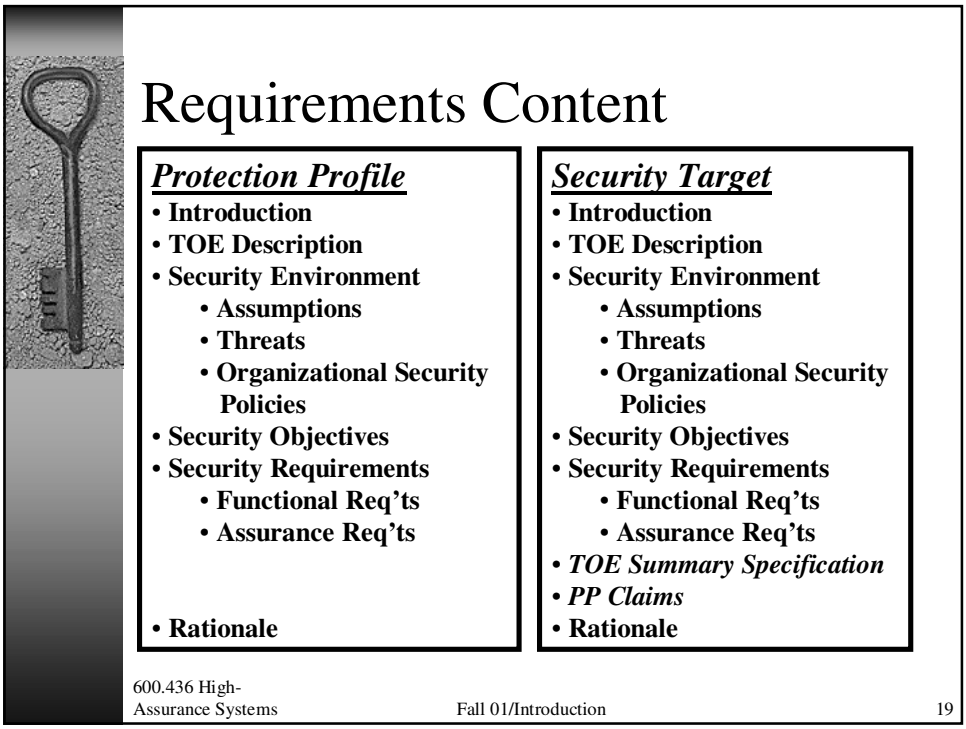

By The Way…
♦ Despite appearances, documentation requirements need not constrain process
♦ Process is how you create
♦ Documentation is how you explain what you have created
600.436 High-Assurance Systems
Fall 01/Introduction
14


Requirements Documentation
User
Requirements
(Purpose)
Validation
(Proof)
Evaluator
Developer
Evidence
(Product)
600.436 High-
Assurance Systems
Fall 01/Introduction
15


User Inputs
♦ None from CC/CEM process
♦ Evaluation Scheme policies and procedures that affect the User's expression of requirements
♦ Generally an external event that raises concern about security
600.436 High-
Assurance Systems
Fall 01/Introduction
16

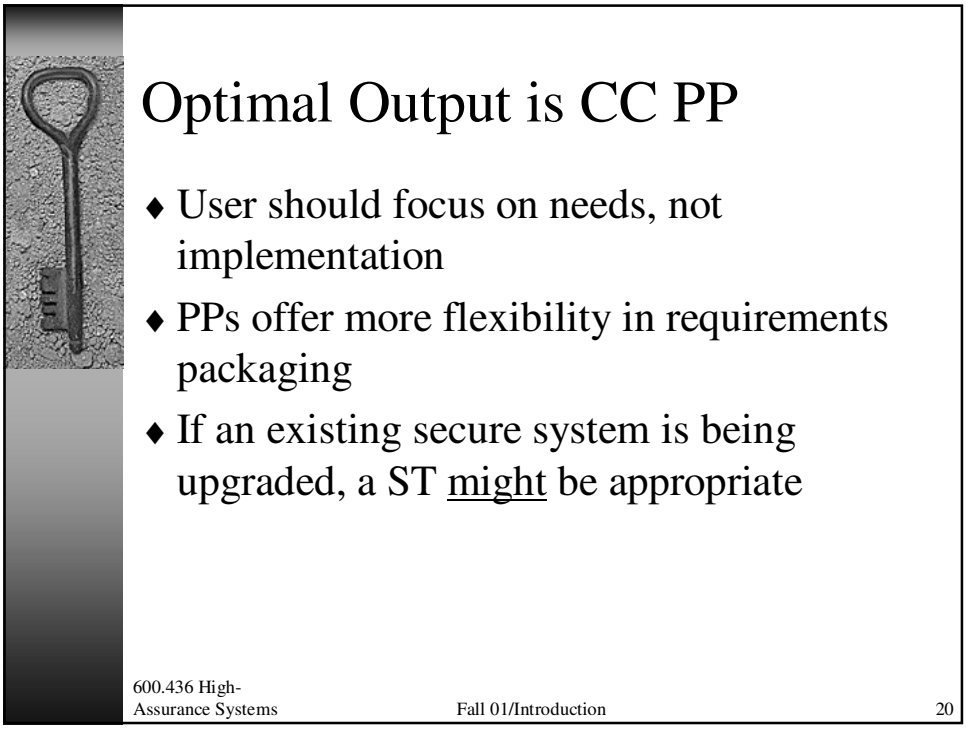

User Activities
♦ Identify system security threats and objectives by doing a risk assessment
♦ Identify cost-effective technical security countermeasures
♦ Identify complementary environmental security countermeasures
♦ Determine confidence required in the system security functions
♦ Correlate threats and objectives with the functional and environmental security countermeasures and confidence requirements
600.436 High-Assurance Systems
Fall 01/Introduction
17


User Outputs (in CC Context)
♦ Two kinds of requirements documentation
♦ Abstract – Protection Profile
– System context, specific functions, general assurances
– No binding to specific implementation
♦ Specific – Security Target
– System context, specific functions, specific assurances
– Bound to specific implementation
600.436 High-Assurance Systems
Fall 01/Introduction
18



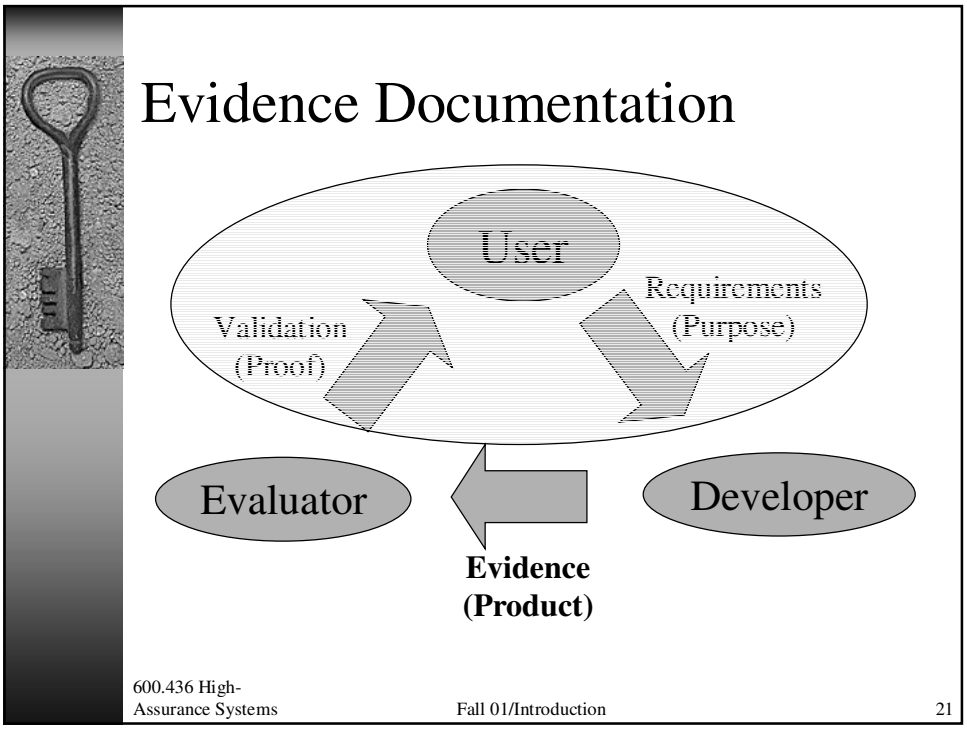
Requirements Content

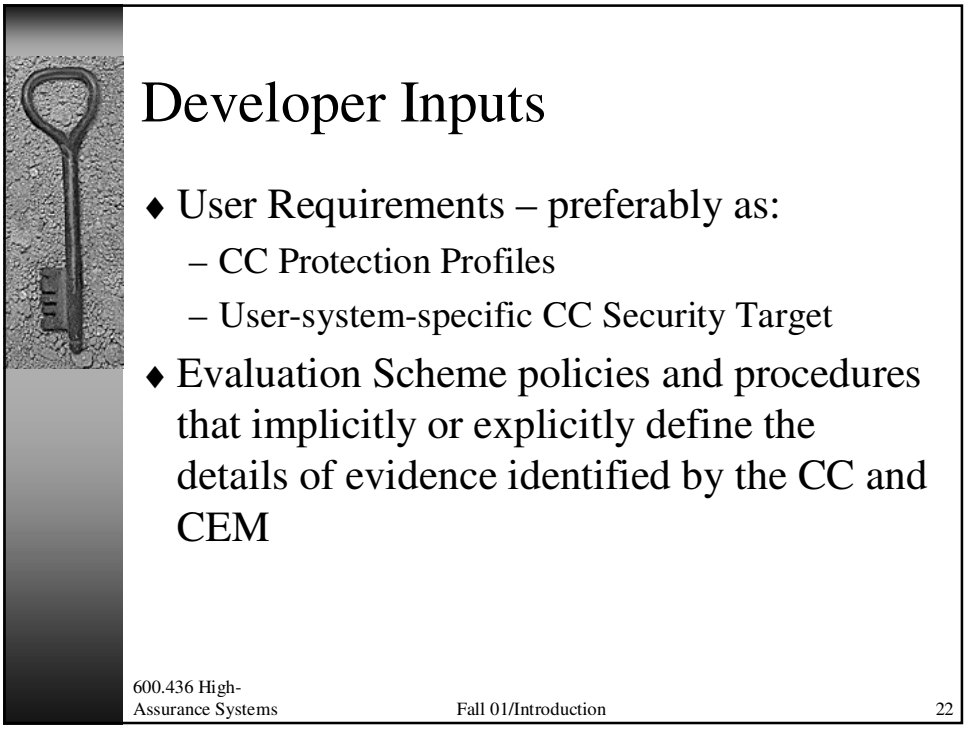
Protection Profile	***Security Target***
• **Introduction**	• **Introduction**
• **TOE Description**	• **TOE Description**
• **Security Environment**	• **Security Environment**
– **Assumptions**	– **Assumptions**
– **Threats**	– **Threats**
– **Organizational Security Policies**	– **Organizational Security Policies**
• **Security Objectives**	• **Security Objectives**
• **Security Requirements**	• **Security Requirements**
– **Functional Req'ts**	– **Functional Req'ts**
– **Assurance Req'ts**	– **Assurance Req'ts**
	• ***TOE Summary Specification***
	• ***PP Claims***
• **Rationale**	• **Rationale**

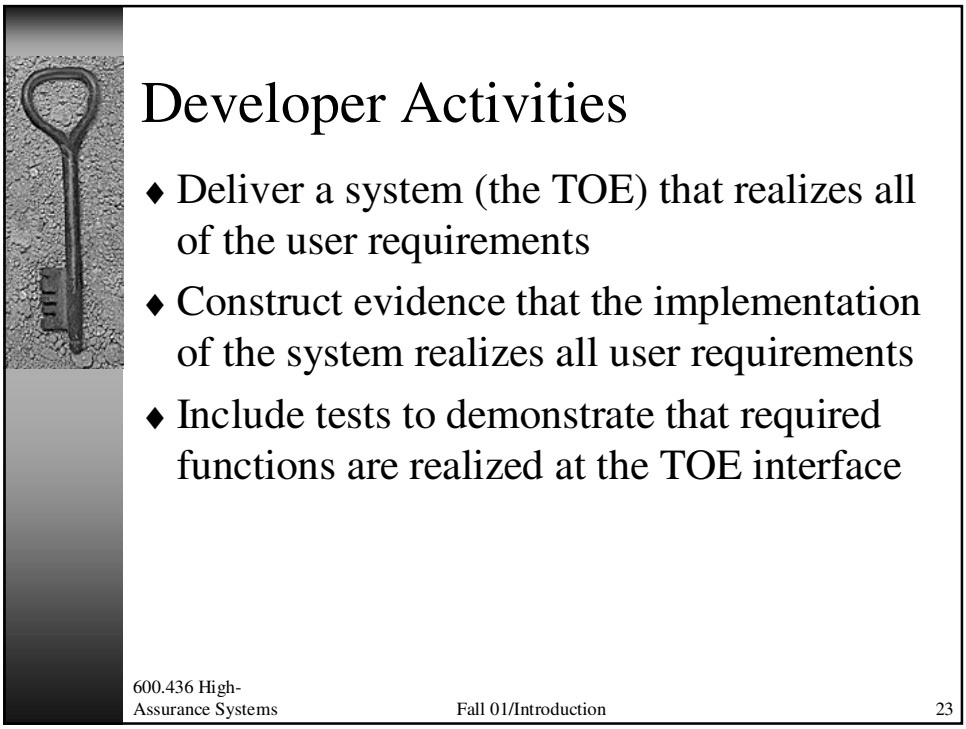


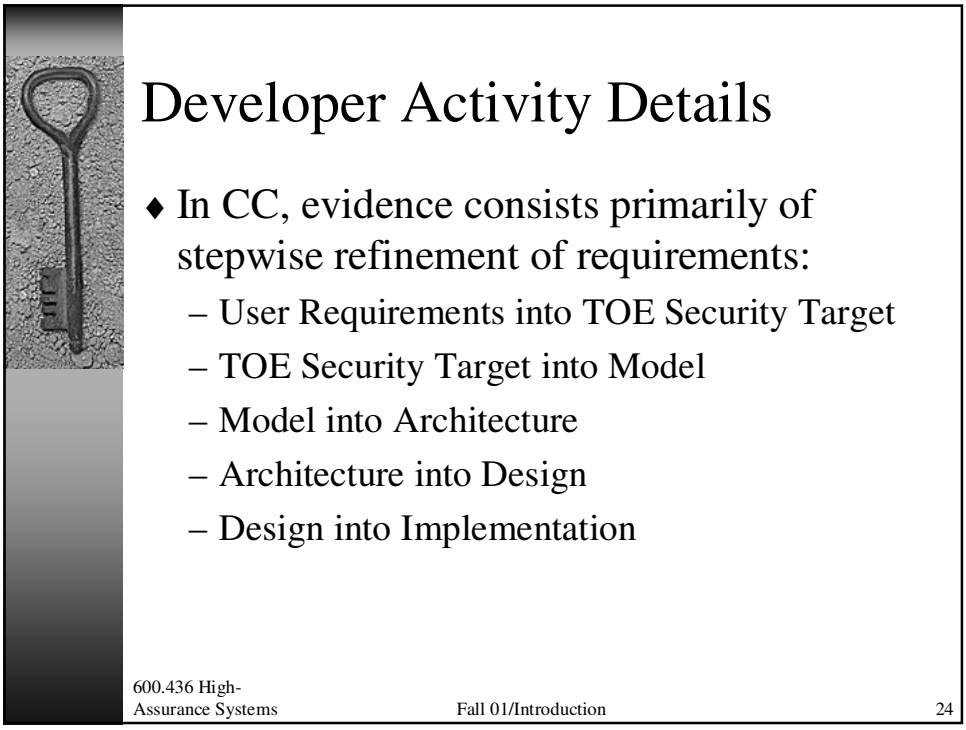
Optimal Output is CC PP

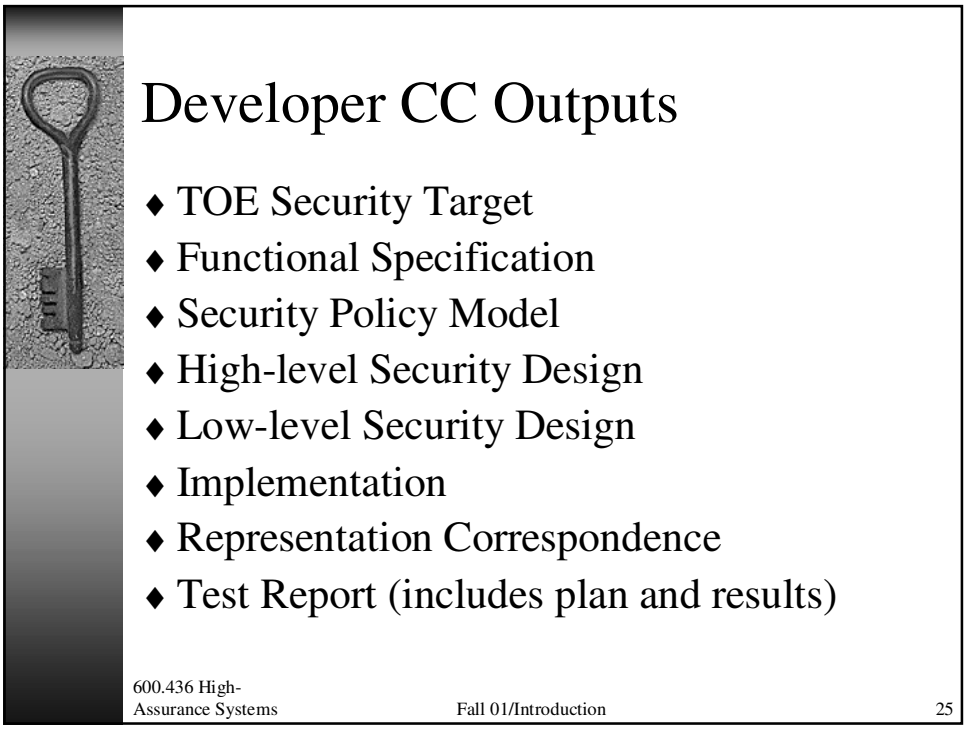
- User should focus on needs, not implementation
- PPs offer more flexibility in requirements packaging
- If an existing secure system is being upgraded, a ST might be appropriate

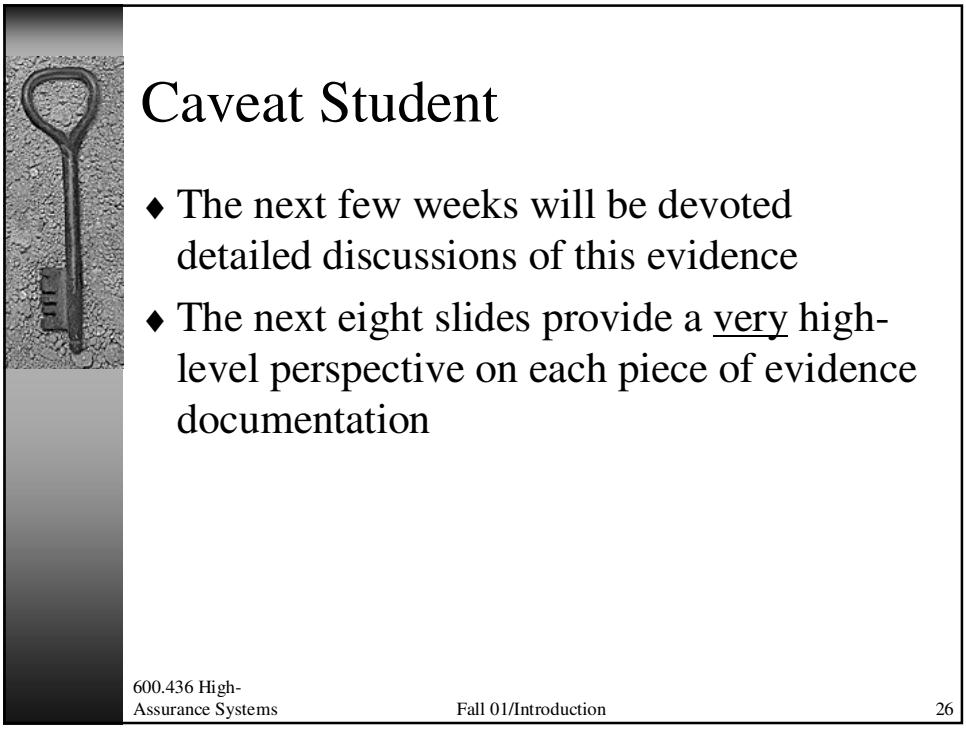

Evidence Documentation
User
Requirements
(Purpose)
Validation
(Proof)
Evaluator
Developer
Evidence
(Product)
600.436 High-
Assurance Systems
Fall 01/Introduction
21

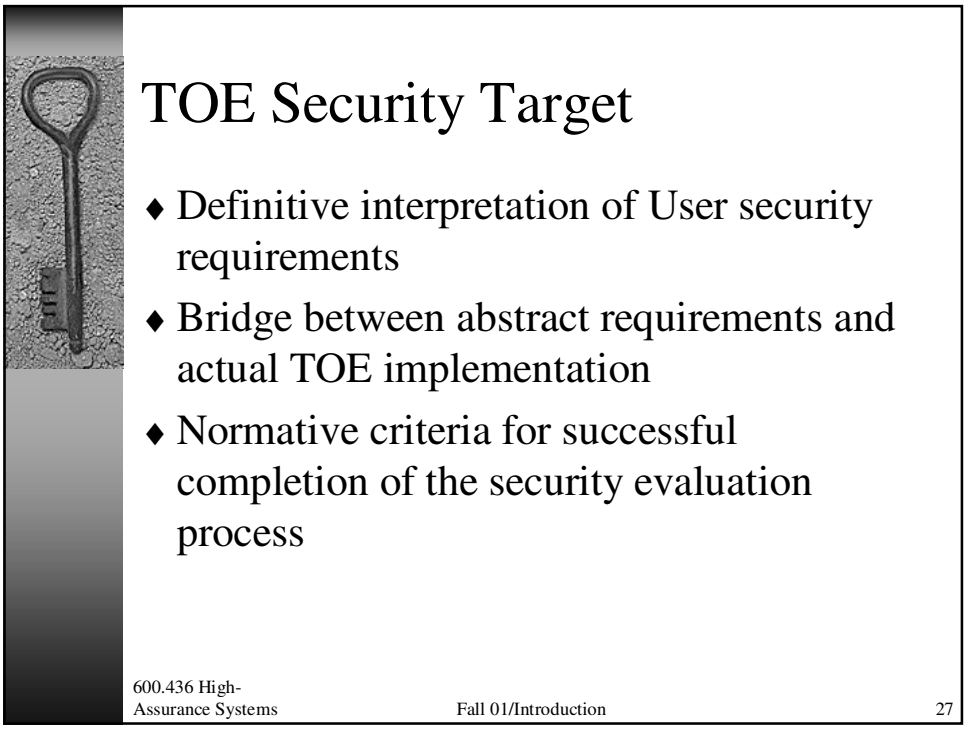

Developer Inputs
♦ User Requirements – preferably as:
– CC Protection Profiles
– User-system-specific CC Security Target
♦ Evaluation Scheme policies and procedures that implicitly or explicitly define the details of evidence identified by the CC and CEM
600.436 High-
Assurance Systems
Fall 01/Introduction
22


Developer Activities
♦ Deliver a system (the TOE) that realizes all of the user requirements
♦ Construct evidence that the implementation of the system realizes all user requirements
♦ Include tests to demonstrate that required functions are realized at the TOE interface
600.436 High-Assurance Systems
Fall 01/Introduction
23


Developer Activity Details
♦ In CC, evidence consists primarily of stepwise refinement of requirements:
– User Requirements into TOE Security Target
– TOE Security Target into Model
– Model into Architecture
– Architecture into Design
– Design into Implementation
600.436 High-Assurance Systems
Fall 01/Introduction
24


Developer CC Outputs
♦ TOE Security Target
♦ Functional Specification
♦ Security Policy Model
♦ High-level Security Design
♦ Low-level Security Design
♦ Implementation
♦ Representation Correspondence
♦ Test Report (includes plan and results)
600.436 High-Assurance Systems
Fall 01/Introduction
25

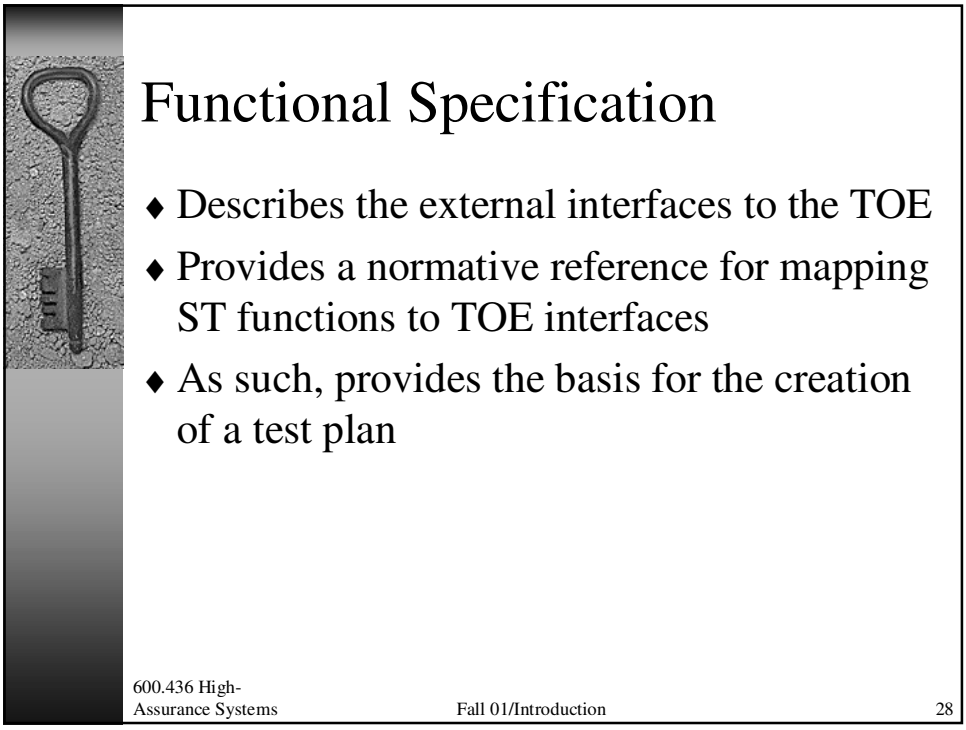

Caveat Student
♦ The next few weeks will be devoted detailed discussions of this evidence
♦ The next eight slides provide a very high-level perspective on each piece of evidence documentation
600.436 High-Assurance Systems
Fall 01/Introduction
26



TOE Security Target

- Definitive interpretation of User security requirements
- Bridge between abstract requirements and actual TOE implementation
- Normative criteria for successful completion of the security evaluation process

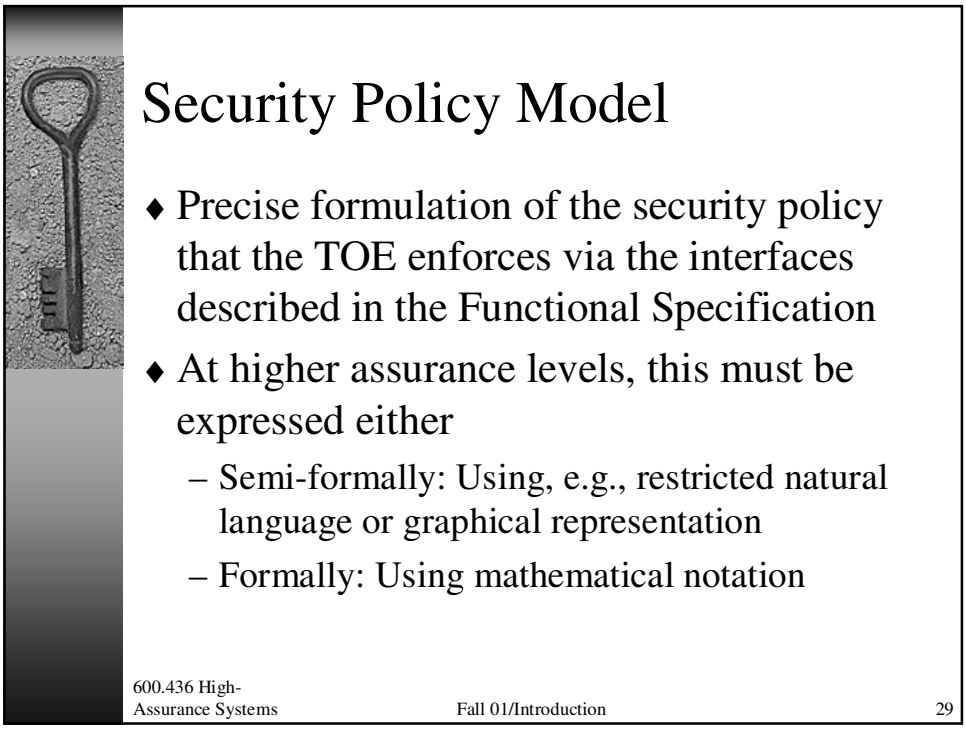
600.436 High-Assurance Systems Fall 01/Introduction 27

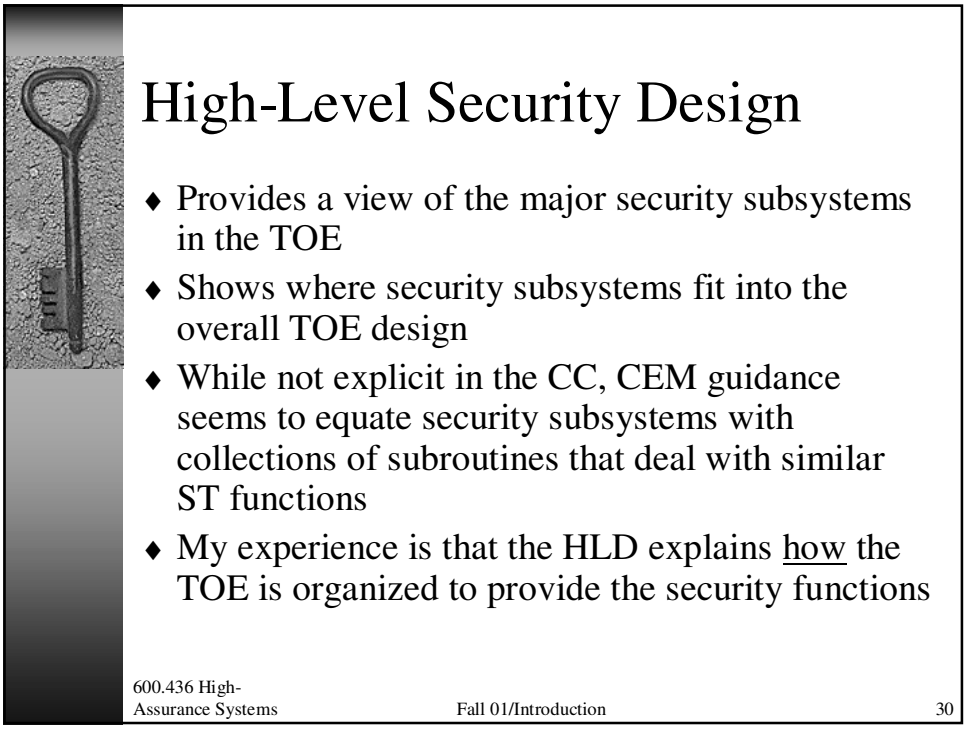


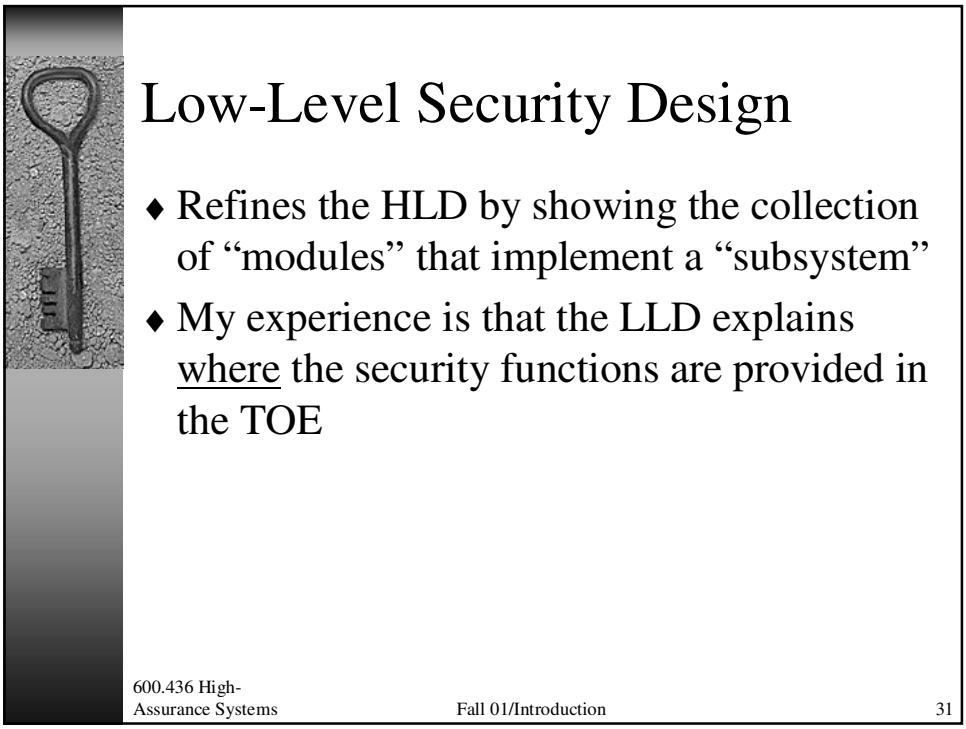
Functional Specification

- Describes the external interfaces to the TOE
- Provides a normative reference for mapping ST functions to TOE interfaces
- As such, provides the basis for the creation of a test plan

600.436 High-Assurance Systems Fall 01/Introduction 28


Security Policy Model
♦ Precise formulation of the security policy that the TOE enforces via the interfaces described in the Functional Specification
♦ At higher assurance levels, this must be expressed either
– Semi-formally: Using, e.g., restricted natural language or graphical representation
– Formally: Using mathematical notation
600.436 High-Assurance Systems
Fall 01/Introduction
29

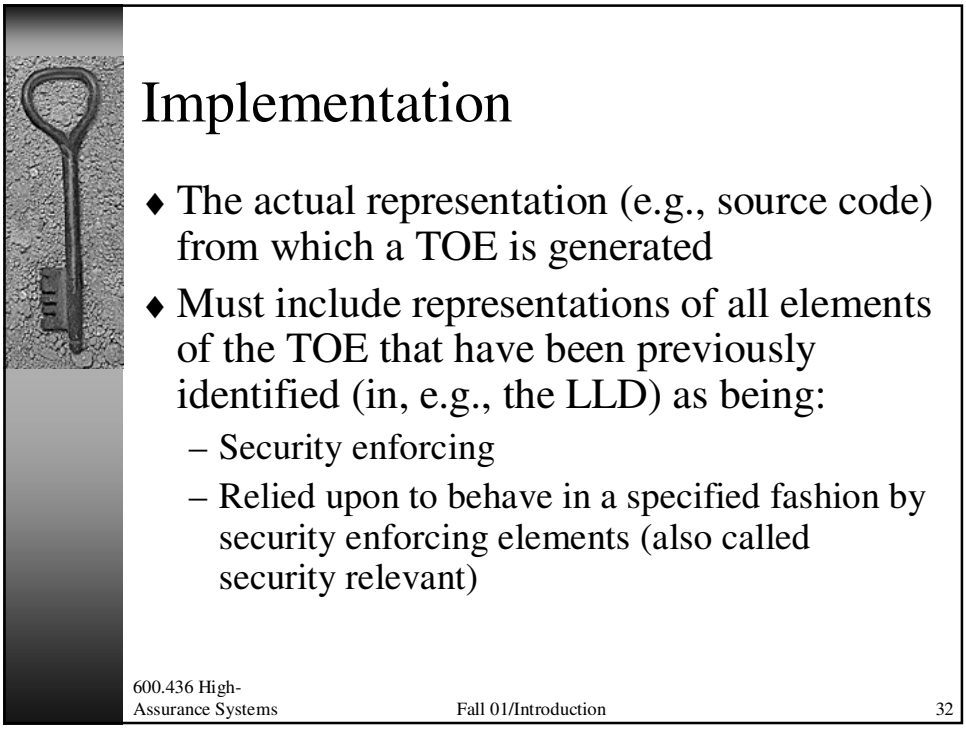

High-Level Security Design
♦ Provides a view of the major security subsystems in the TOE
♦ Shows where security subsystems fit into the overall TOE design
♦ While not explicit in the CC, CEM guidance seems to equate security subsystems with collections of subroutines that deal with similar ST functions
♦ My experience is that the HLD explains how the TOE is organized to provide the security functions
600.436 High-Assurance Systems
Fall 01/Introduction
30



Low-Level Security Design

- Refines the HLD by showing the collection of "modules" that implement a "subsystem"
- My experience is that the LLD explains where the security functions are provided in the TOE

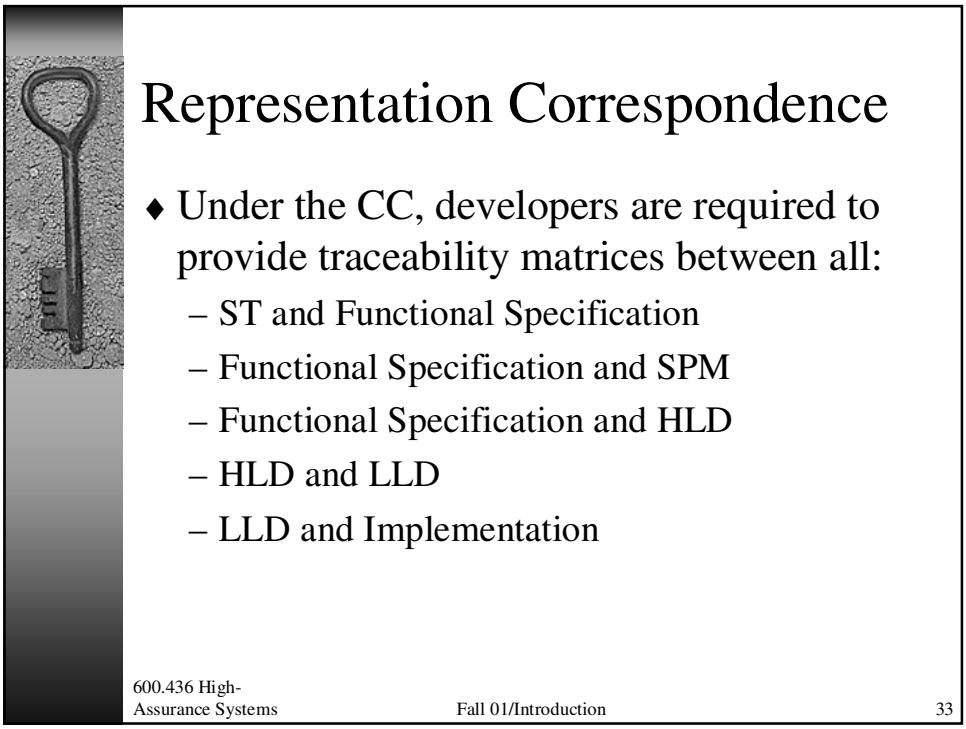
600.436 High-Assurance Systems Fall 01/Introduction 31



Implementation

- The actual representation (e.g., source code) from which a TOE is generated
- Must include representations of all elements of the TOE that have been previously identified (in, e.g., the LLD) as being:
  - Security enforcing
  - Relied upon to behave in a specified fashion by security enforcing elements (also called security relevant)

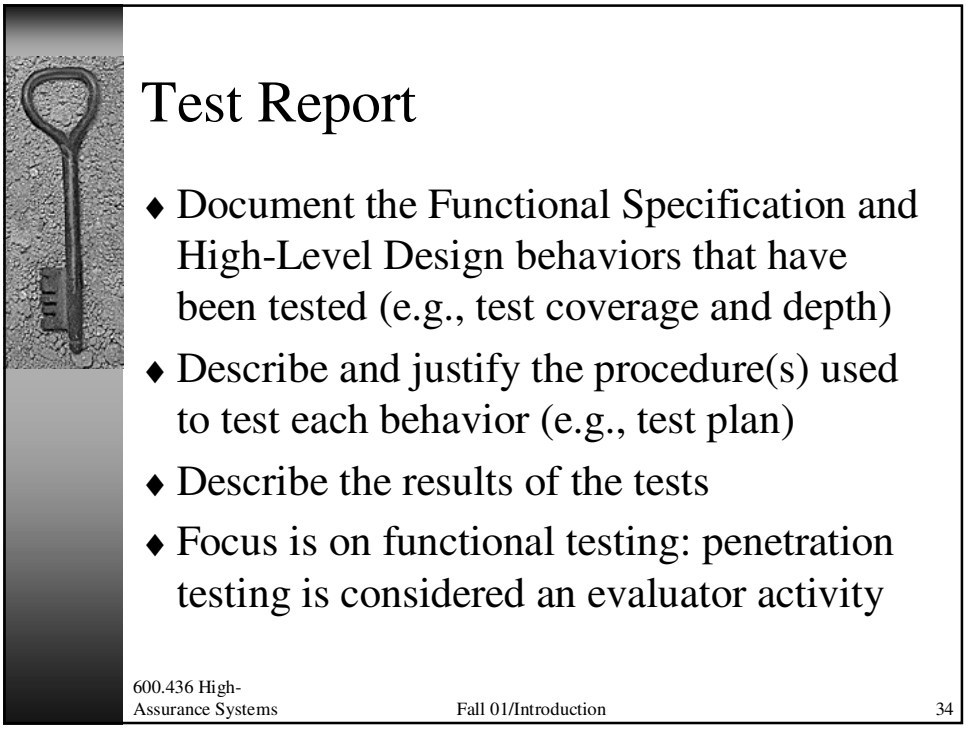
600.436 High-Assurance Systems Fall 01/Introduction 32



Representation Correspondence

- Under the CC, developers are required to provide traceability matrices between all:
  - ST and Functional Specification
  - Functional Specification and SPM
  - Functional Specification and HLD
  - HLD and LLD
  - LLD and Implementation

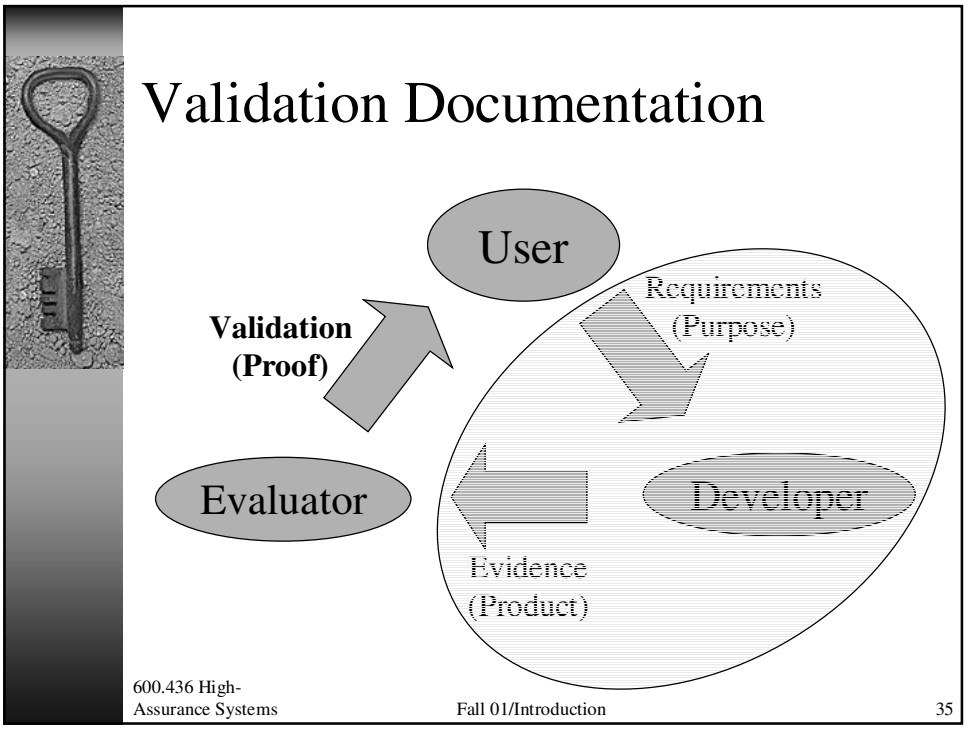
600.436 High-Assurance Systems Fall 01/Introduction 33

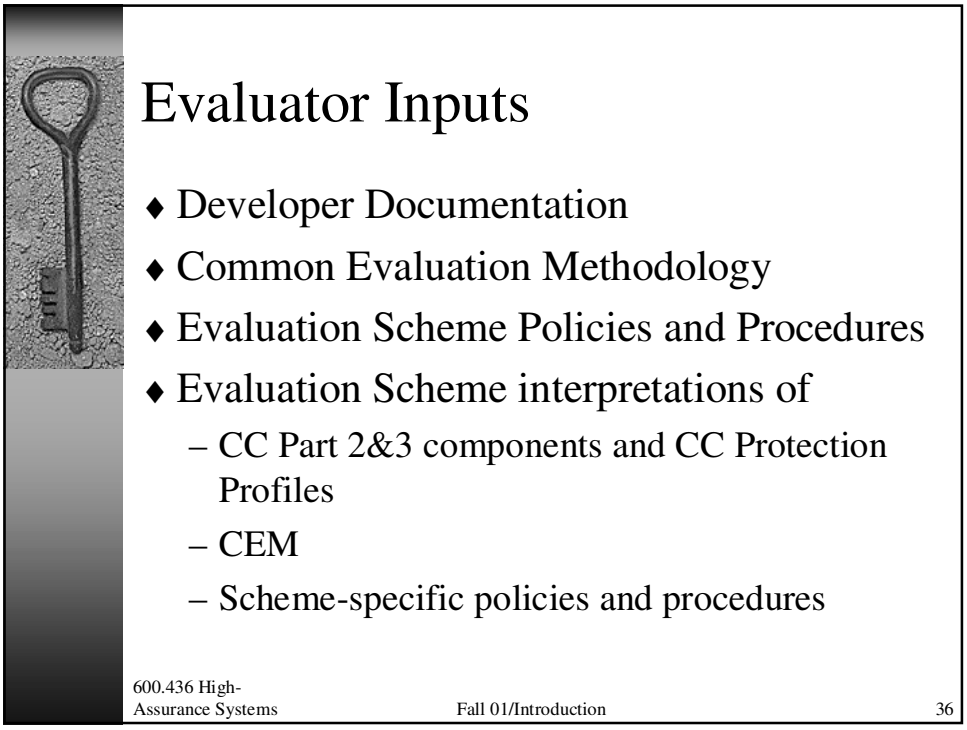


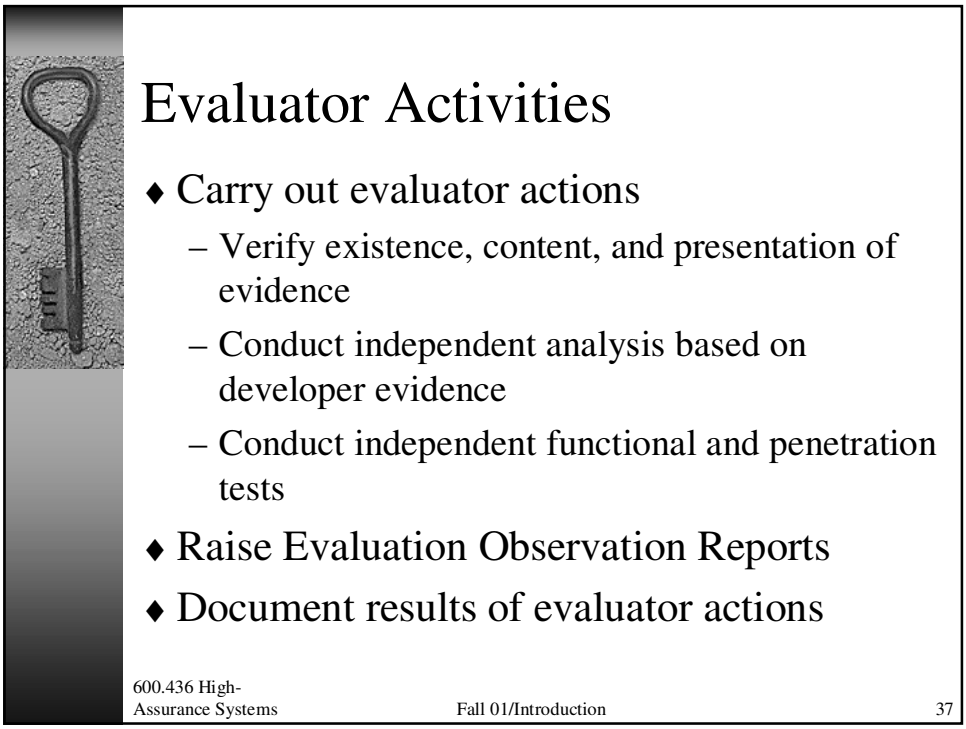
Test Report

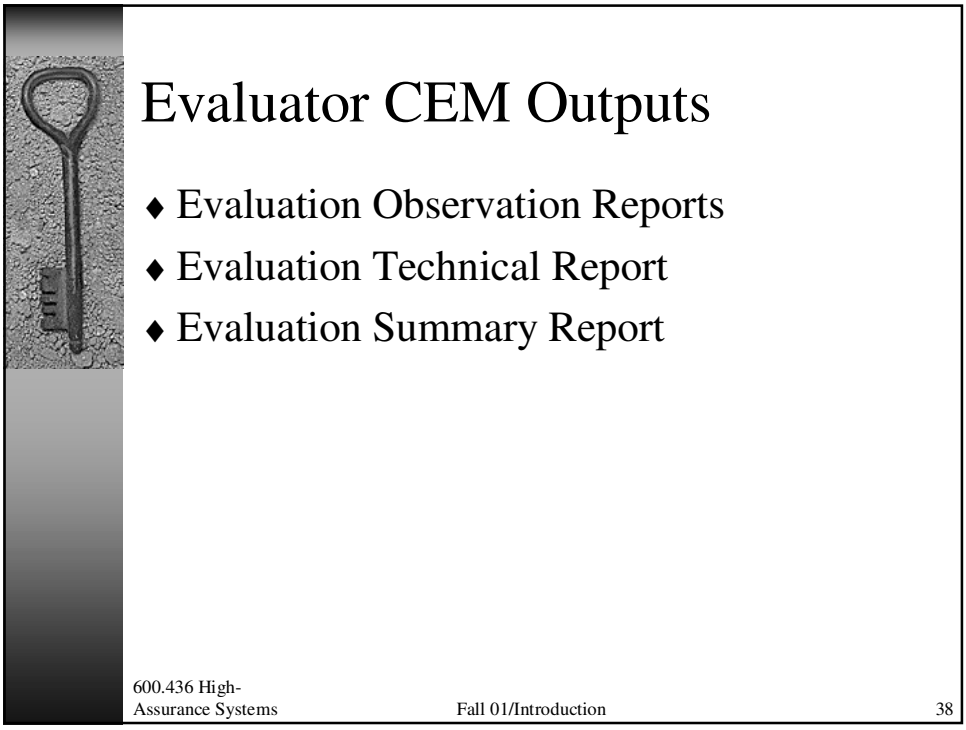
- Document the Functional Specification and High-Level Design behaviors that have been tested (e.g., test coverage and depth)
- Describe and justify the procedure(s) used to test each behavior (e.g., test plan)
- Describe the results of the tests
- Focus is on functional testing: penetration testing is considered an evaluator activity

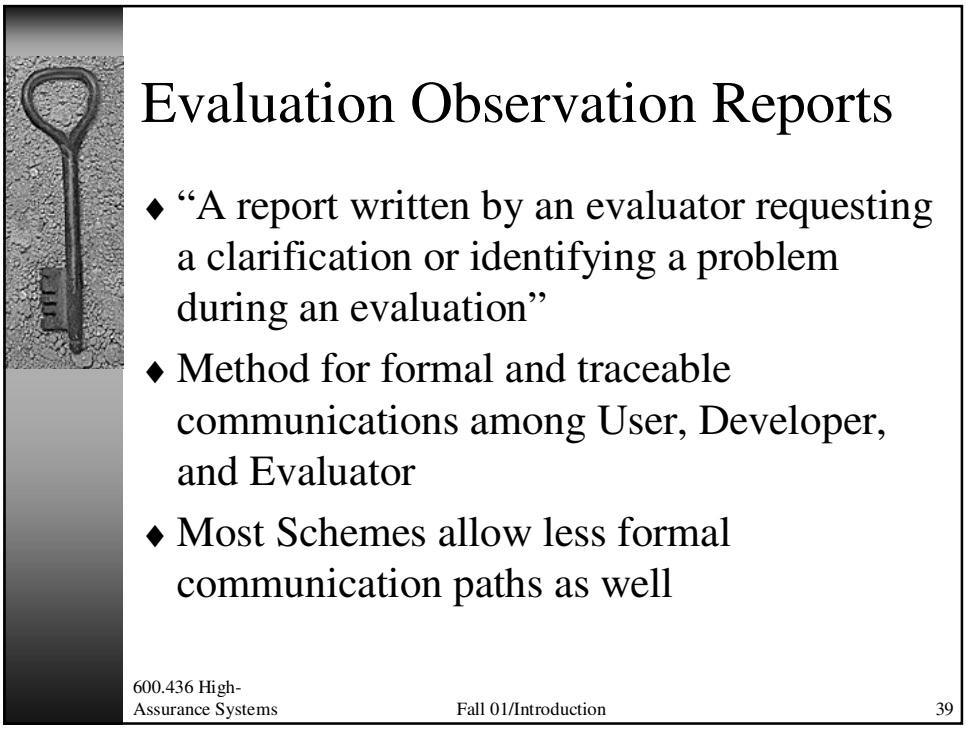
600.436 High-Assurance Systems Fall 01/Introduction 34


Validation Documentation
User
Requirements
(Purpose)
Validation
(Proof)
Evaluator
Developer
Evidence
(Product)
600.436 High-
Assurance Systems
Fall 01/Introduction
35


Evaluator Inputs
♦ Developer Documentation
♦ Common Evaluation Methodology
♦ Evaluation Scheme Policies and Procedures
♦ Evaluation Scheme interpretations of
– CC Part 2&3 components and CC Protection Profiles
– CEM
– Scheme-specific policies and procedures
600.436 High-
Assurance Systems
Fall 01/Introduction
36


Evaluator Activities
♦ Carry out evaluator actions
– Verify existence, content, and presentation of evidence
– Conduct independent analysis based on developer evidence
– Conduct independent functional and penetration tests
♦ Raise Evaluation Observation Reports
♦ Document results of evaluator actions
600.436 High-Assurance Systems
Fall 01/Introduction
37

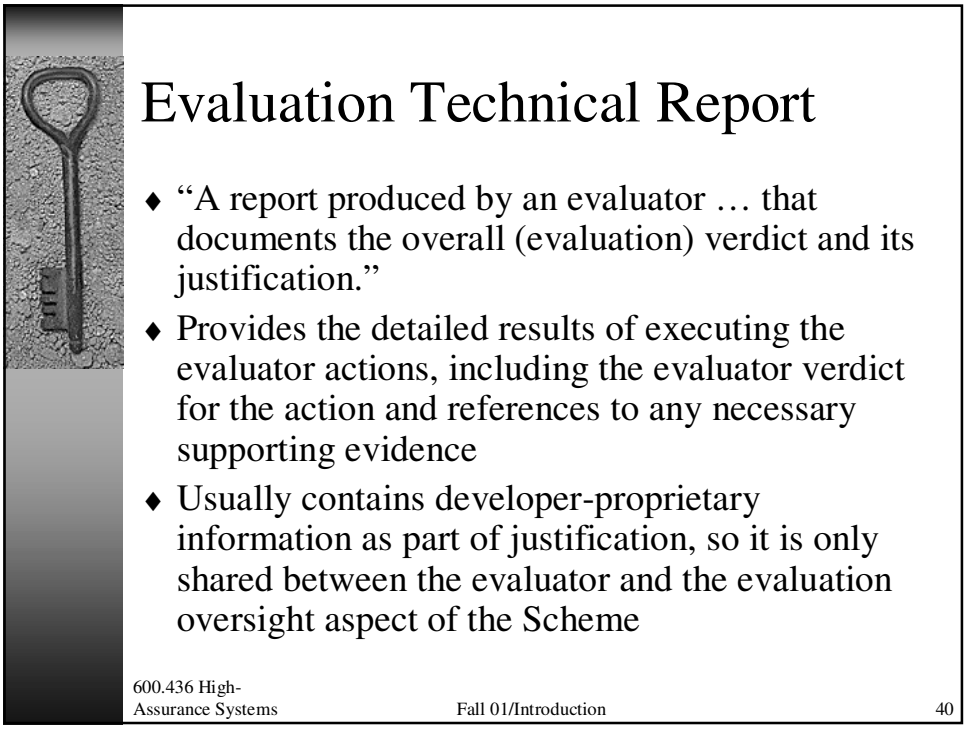

Evaluator CEM Outputs
♦ Evaluation Observation Reports
♦ Evaluation Technical Report
♦ Evaluation Summary Report
600.436 High-Assurance Systems
Fall 01/Introduction
38



Evaluation Observation Reports

- "A report written by an evaluator requesting a clarification or identifying a problem during an evaluation"
- Method for formal and traceable communications among User, Developer, and Evaluator
- Most Schemes allow less formal communication paths as well

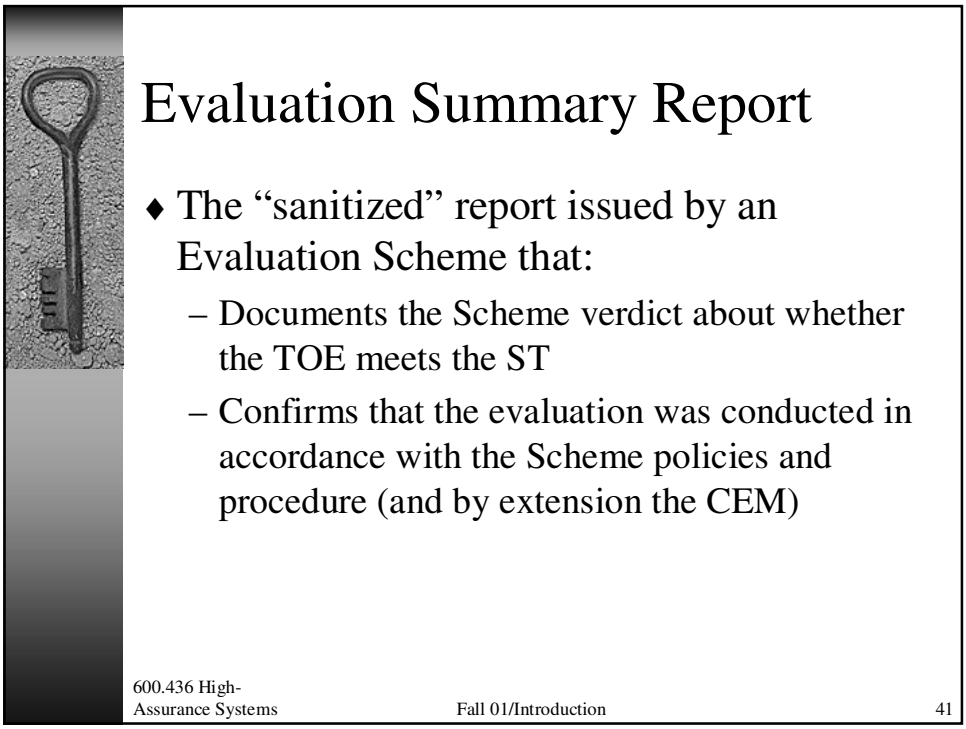
600.436 High-Assurance Systems Fall 01/Introduction 39



Evaluation Technical Report

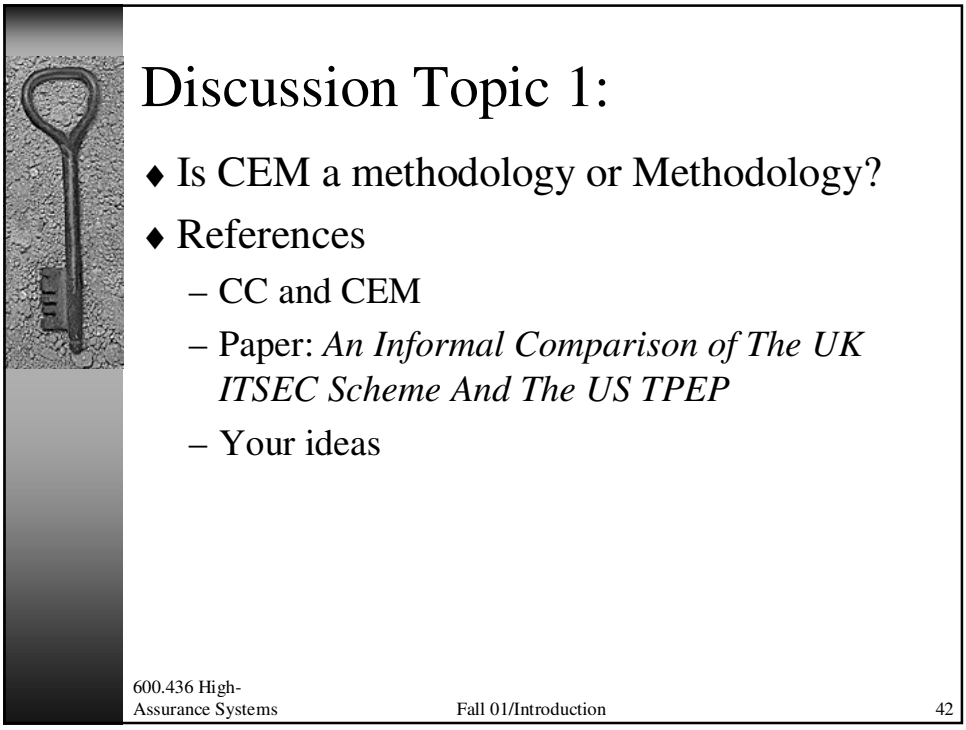
- "A report produced by an evaluator … that documents the overall (evaluation) verdict and its justification."
- Provides the detailed results of executing the evaluator actions, including the evaluator verdict for the action and references to any necessary supporting evidence
- Usually contains developer-proprietary information as part of justification, so it is only shared between the evaluator and the evaluation oversight aspect of the Scheme

600.436 High-Assurance Systems Fall 01/Introduction 40



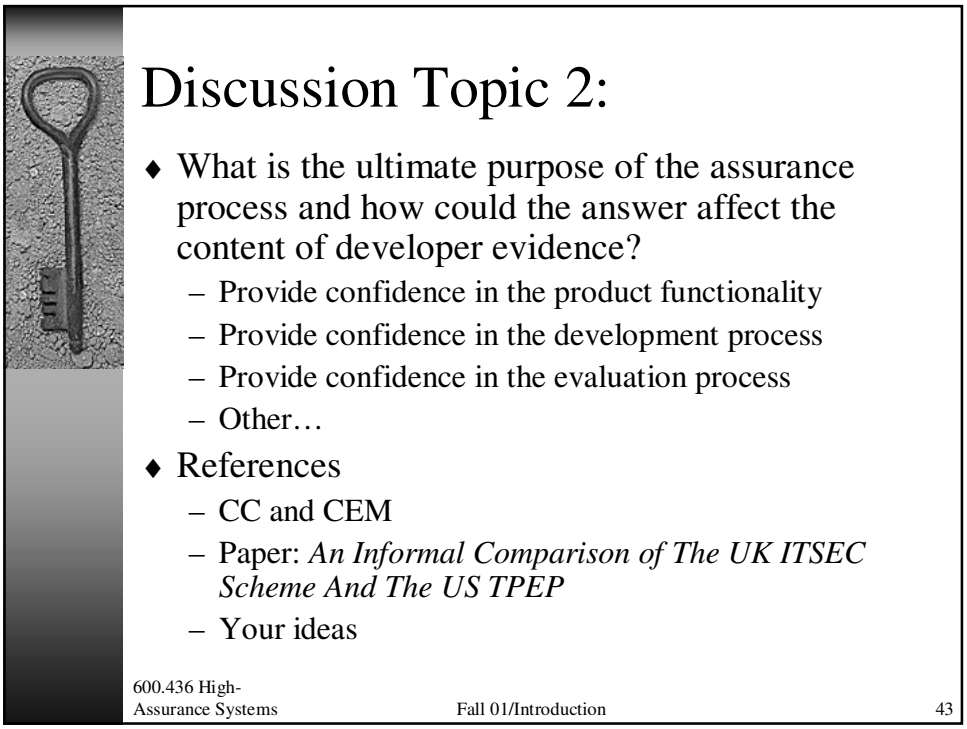
Evaluation Summary Report

- The "sanitized" report issued by an Evaluation Scheme that:
  - Documents the Scheme verdict about whether the TOE meets the ST
  - Confirms that the evaluation was conducted in accordance with the Scheme policies and procedure (and by extension the CEM)



Discussion Topic 1:

- Is CEM a methodology or Methodology?
- References
  - CC and CEM
  - Paper: *An Informal Comparison of The UK ITSEC Scheme And The US TPEP*
  - Your ideas


Discussion Topic 2:
♦ What is the ultimate purpose of the assurance process and how could the answer affect the content of developer evidence?
– Provide confidence in the product functionality
– Provide confidence in the development process
– Provide confidence in the evaluation process
– Other…
♦ References
– CC and CEM
– Paper: An Informal Comparison of The UK ITSEC Scheme And The US TPEP
– Your ideas
600.436 High-Assurance Systems
Fall 01/Introduction
43