



# Measuring and Assessing Software Trustworthiness: Approaches and Challenges

Elizabeth Fong

National Institute of Standards and Technology

CASCON2010 Workshops

Software Certification Consortium: Certification Methods for Safety-Critical Software

3<sup>rd</sup> November 2010

Toronto, Canada

# Presentation Outline

- **Overview of certification process model**
- **Selected Projects within Information Technology Laboratory, NIST as related to assessing trustworthiness of software**
  - **Software Metrics and Tool Evaluation (SAMATE)**
  - **Software Labels**
  - **Structured Assurance Case Methodology**
- **Challenges**

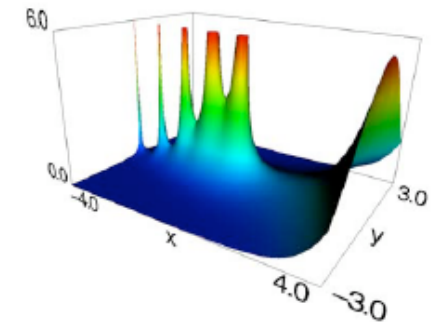
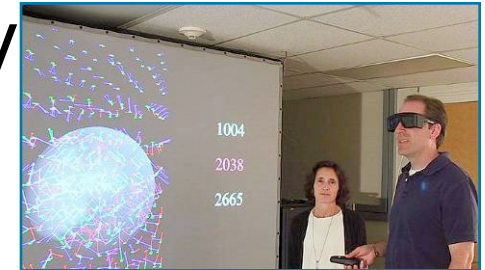
# National Institute of Standards and Technology Information Technology Laboratory

To promote US innovation and industrial competitiveness by advancing

*measurement science,  
standards, and  
technology*

through research and development in

*information technology,  
mathematics, and  
statistics*



# Certification Process Components

## Certification

qualified bodies to do the testing and certification  
control board - advisory and arbiter

## Validation

Process - policy and procedures for  
testing

## Conformance Testing

Test assertions

Test suite

(test software, test scripts, test  
criteria)

## Standard

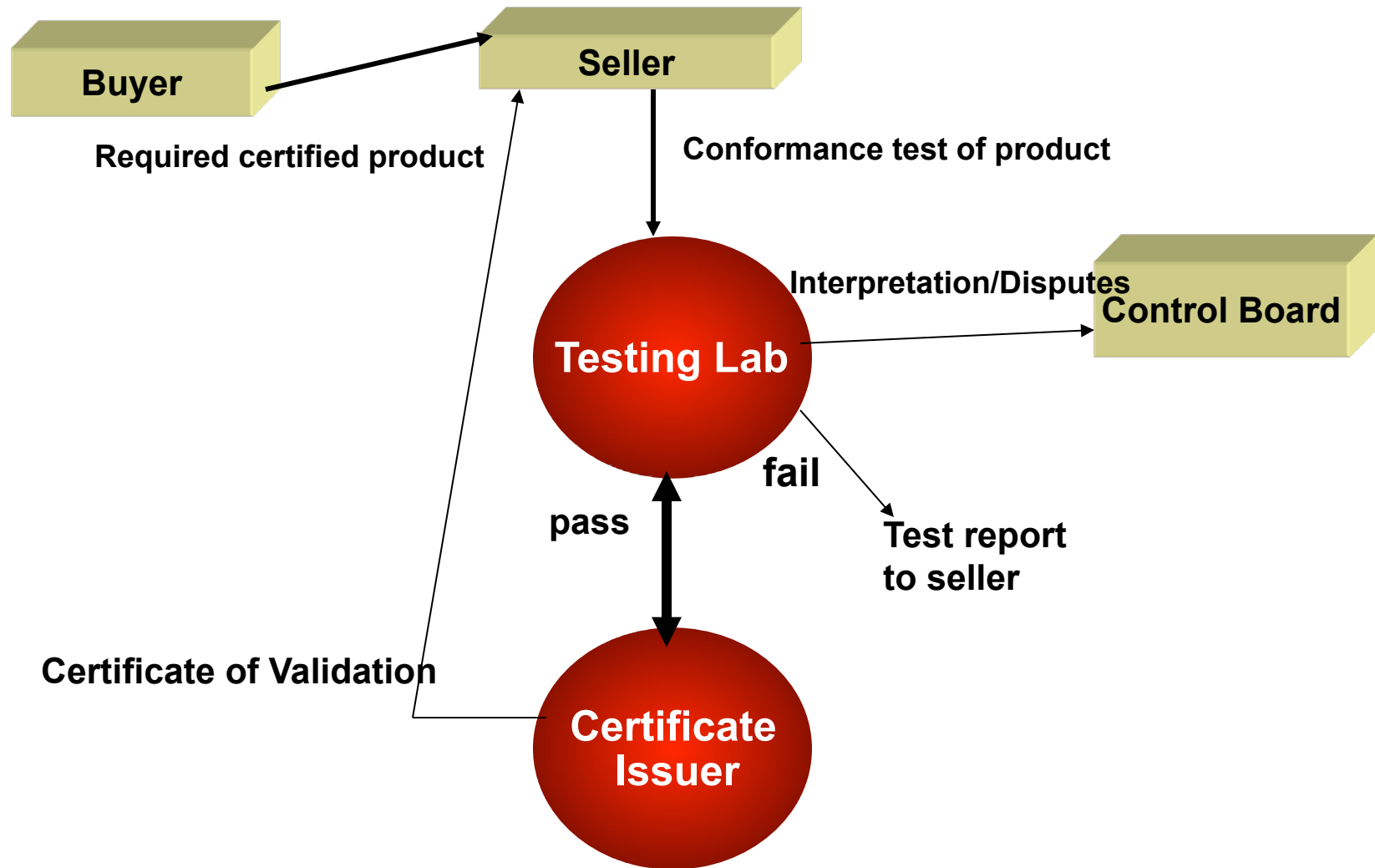
Conformance clause

# Certification Regimes

- Self certification
- Vendor declared rating system
- 3<sup>rd</sup> party certification

# Validation & Certification Process

## Flow



# SAMATE Overview

- **Research and Measurement Opportunities**
  - Test and measure the effectiveness of SwA tools
  - Standard definitions for software weaknesses
  - Identify gaps in tools and methods and needed research
- **Major subtasks**
  - Static source code security analyzers
  - Static Analysis Tool Exposition (SATE)
  - Studies on tool contribution to assurance
  - Web application scanners
  - Vote tools evaluation methodology
  - Software labels

<http://samate.nist.gov/>

# **Software labels**

**Investigation into the feasibility of software  
fact labels**

**And**

**What information can help the software  
buyer decide which product is better or  
more secure**



# Audiences and Scope

- Small businesses (dentist, drycleaners, accountant, plumber, restaurant)
- Integrators (mission critical systems)
- Naïve home users (my brother)

# Software Rating Systems

- No standards
- Common criteria
- Due diligence questionnaires
- Rating for security and quality by some software assurance tools
  - OWASP application security verification standards
  - Veracode security rating system
  - Coverity software integrity rating

# Software Label

- Software Facts should be:
  - Voluntary
  - Absolutely simple to produce
  - In a standard format for other claims
- What could be easily supplied?
  - Source available? Yes/No/Escrowed
  - Default installation is secure?
  - Accessed: network, disk, ...
  - What configuration files? (registry, ...)
  - Certificates (e.g., "No Severe weaknesses found by CodeChecker ver. 3.2")
- Cautions
  - A label can give false confidence.
  - A label shut out better software.
  - Labeling diverts effort from real improvements.

**Security Facts**  
Type: Web Application Oct 25, 2010

**OWASP Top 10 2010**

A1-Injection	●
A2-Cross Site Scripting (XSS)	●
A3-Authentication	○
A4-Object References	●
A5-Cross Site Request Forgery	●
A6-Security Configuration	●
A7-Cryptographic Storage	●
A8-URL Access Control	●
A9-Transport Layer Protection	●
A10-Redirects and Forwards	○

**Custom Code Modules**

Name	Language	Size (LOC)
Reports	Java	13000
Mailer	Java	136000
UI	JSP	215185
Engine	Java	512013
Database	SQL	65000

**Libraries**

Name	Language	IT
Struts 2.1.0	Java	●
Log4j 1.9.1	Java	○
DOM 1.2	Java	○

**Platform Components**

Name	Language	IT
WebSphere	Java	●

**Interfaces and Connections**

Name	Protocol	D	S
MPayment	SOAP	●	●
DB2	JDBC	●	○
FileNet	FTP	●	○

**Sensitive Data**

Name	C	I	A
Medical Imagery	●	●	○
Statements	●	●	○

**Application Security Program**

Key Practice Area	M
M1-Strategy and Metrics	●
M2-Policy and Compliance	●
M3-Education and Guidance	○
M4-Threat Assessment	○
M5-Security Requirements	○
M6-Secure Architecture	○
M7-Design Analysis	○
M8-Code Review	○
M9-Security Testing	○
M10-Vulnerability Mgmt	○
M11-Environment Hardening	○
M12-Operational Enablement	○

Security Contact: security@owasp.org

# **Trustworthy Software**

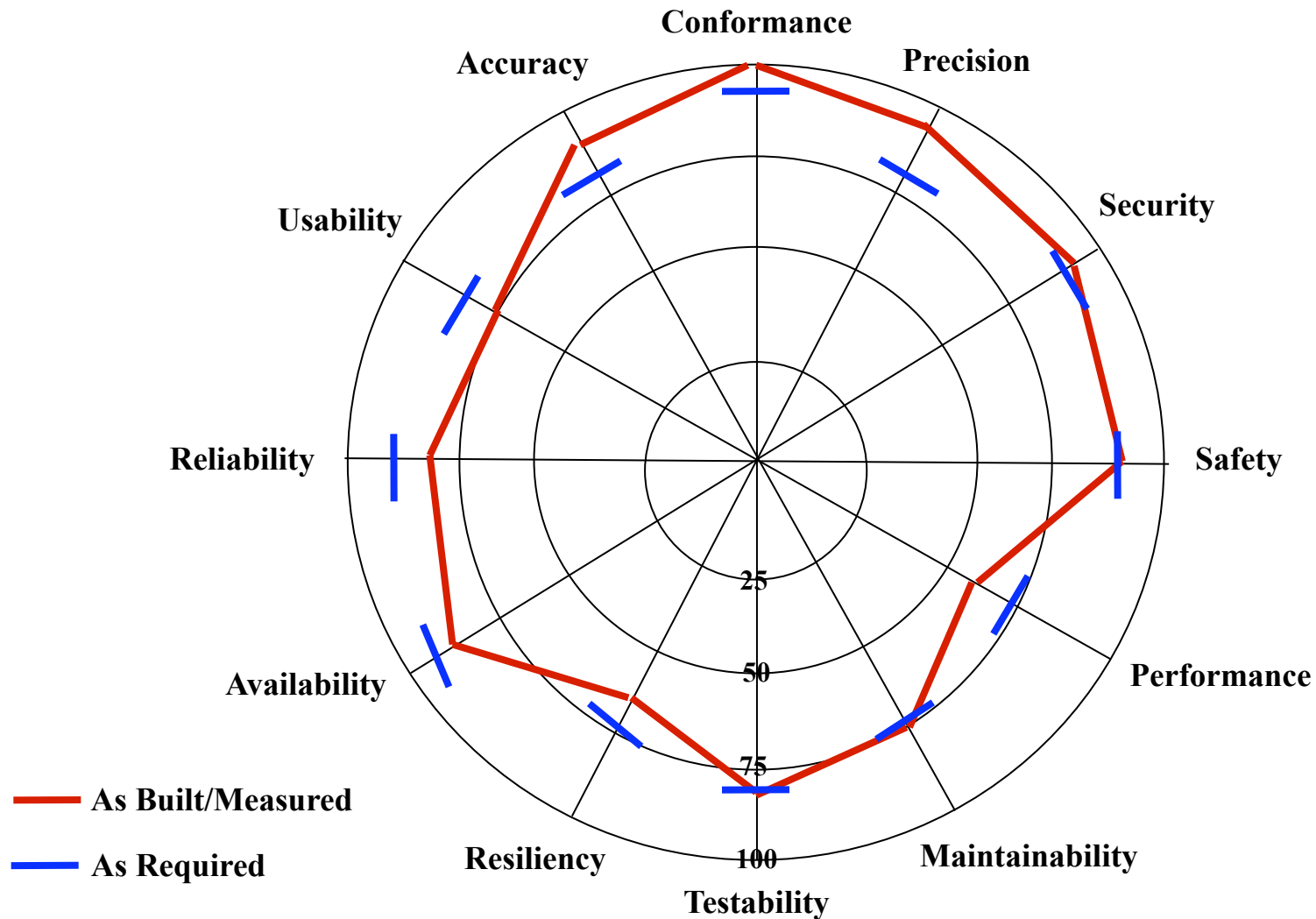
**Software system that performs as intended for a specific purpose, when needed, with operational resiliency, and without unwanted side-effects, behaviors, or exploitable vulnerabilities.**

# **Is trustworthiness of software measurable?**

## **Assumptions:**

- Software trustworthiness is composed of many attributes, i.e., security, reliability, availability, etc.**
- Each attribute can be individually measured under constraints/conditions e.g. operational context.**

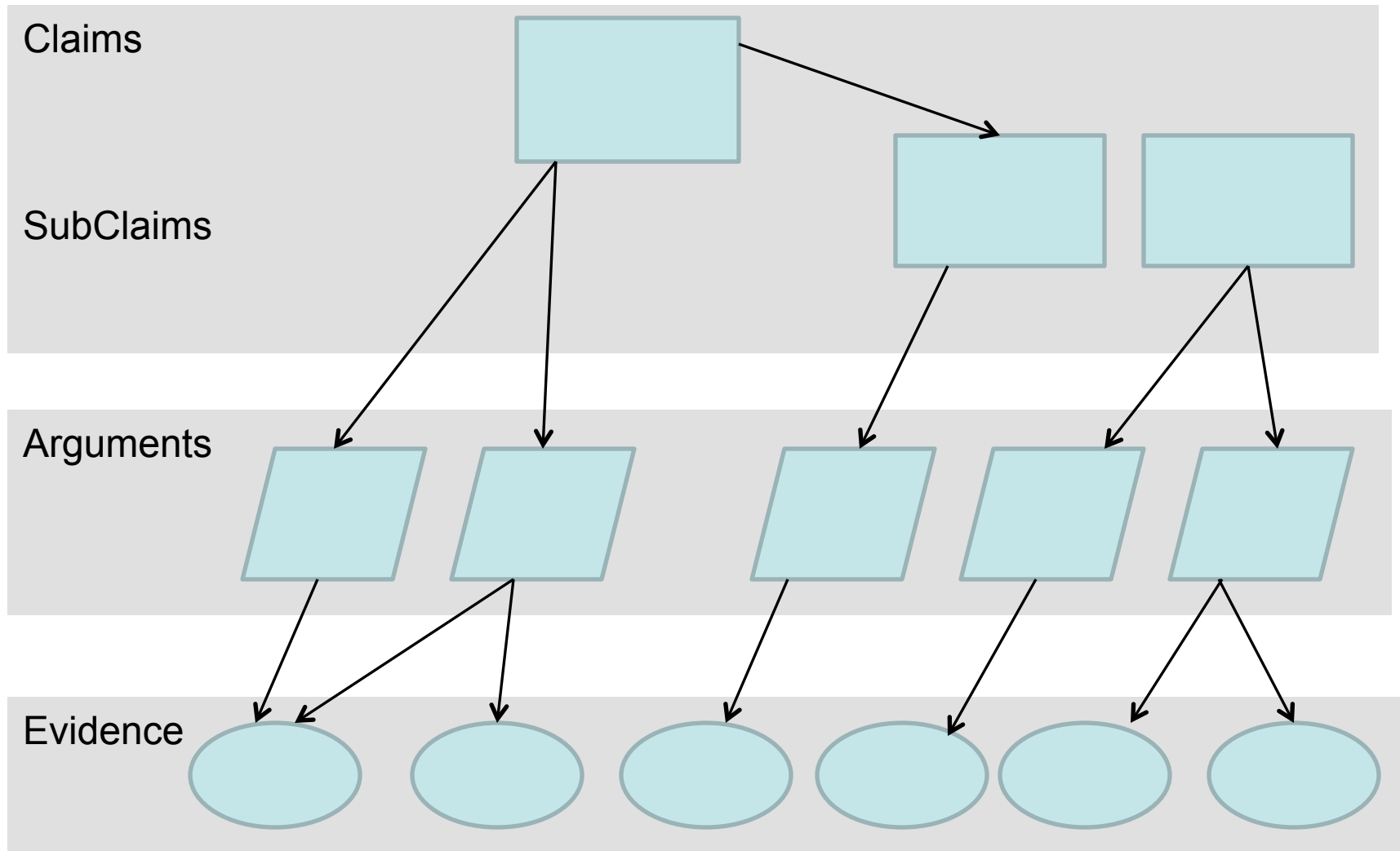
# Concept of Trustworthy Software Index



# **Measuring and Assessing Software for Trustworthiness**

- Develop argumentation models of trustworthiness based upon assurance claims, arguments and evidence (an “assurance case model”)**
- Conduct research into the composability models of trustworthy software characteristics (safety, dependability, security, reliability, etc.)**

# Assurance Case Structure





# **Use Structured Assurance Case Methodology as Certification Process**

- **Provides a product-focused perspective into the attributes of the software system**
  - **Not a compliance-driven view (i.e. not a “checklist”, such as Common Criteria specifications)**
  - **Graphical notation improves analysis**
  - **Argumentation and evidence helps explain why you believe a product’s attribute is assured**
  - **Provides traceability between assurance claims and evidence**

# Challenges

- How to provide a “measure” of trust to software based on uncertainty analysis
- Software supply chain risk management