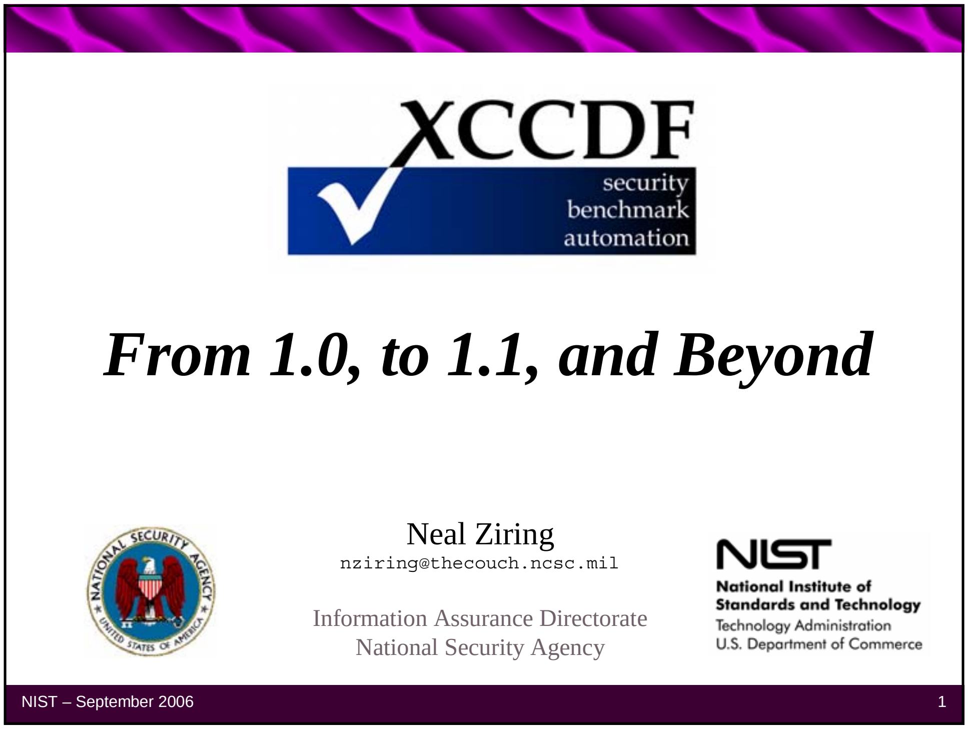


XCCDF
security
benchmark
automation

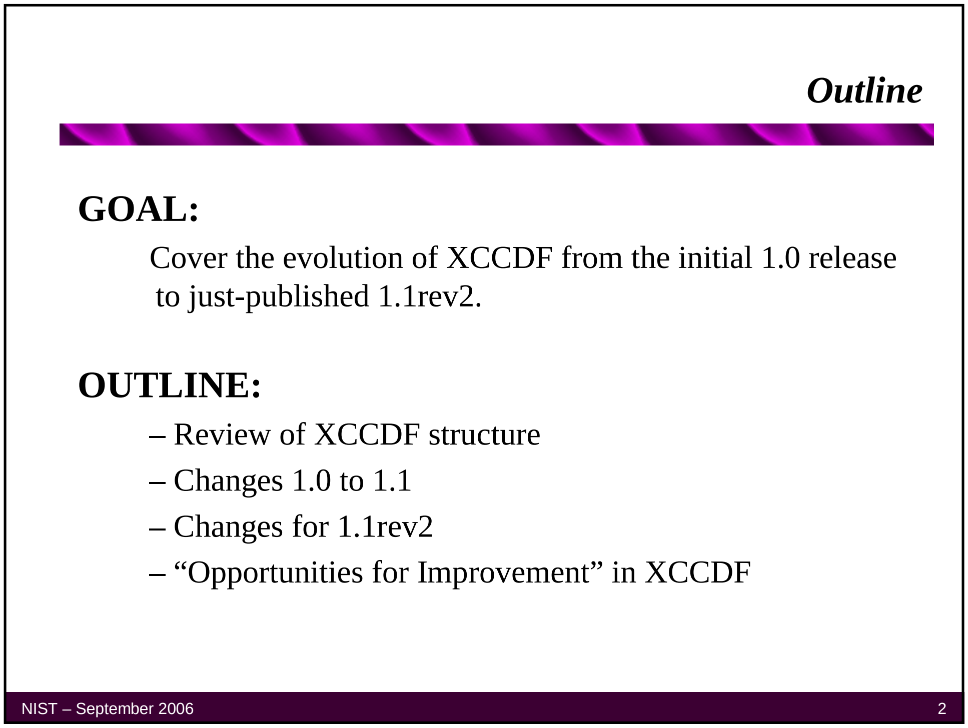
From 1.0, to 1.1, and Beyond

Neal Ziring
`nziring@thecouch.ncsc.mil`

Information Assurance Directorate
National Security Agency

National Institute of Standards and Technology
Technology Administration
U.S. Department of Commerce

NIST – September 2006 1



Outline

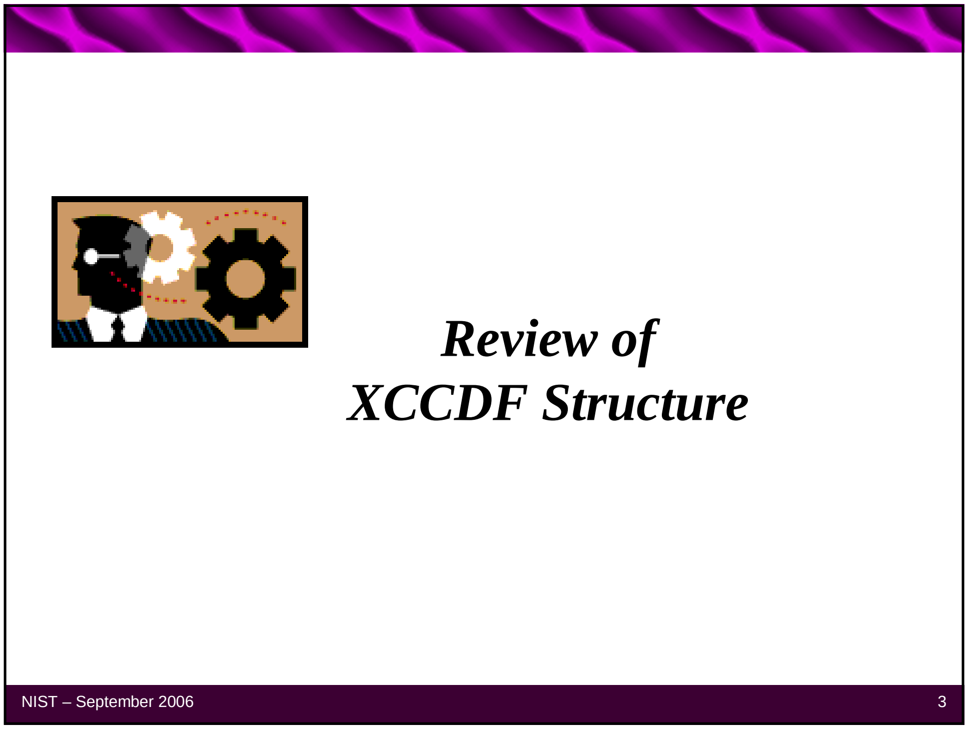
GOAL:

Cover the evolution of XCCDF from the initial 1.0 release to just-published 1.1rev2.

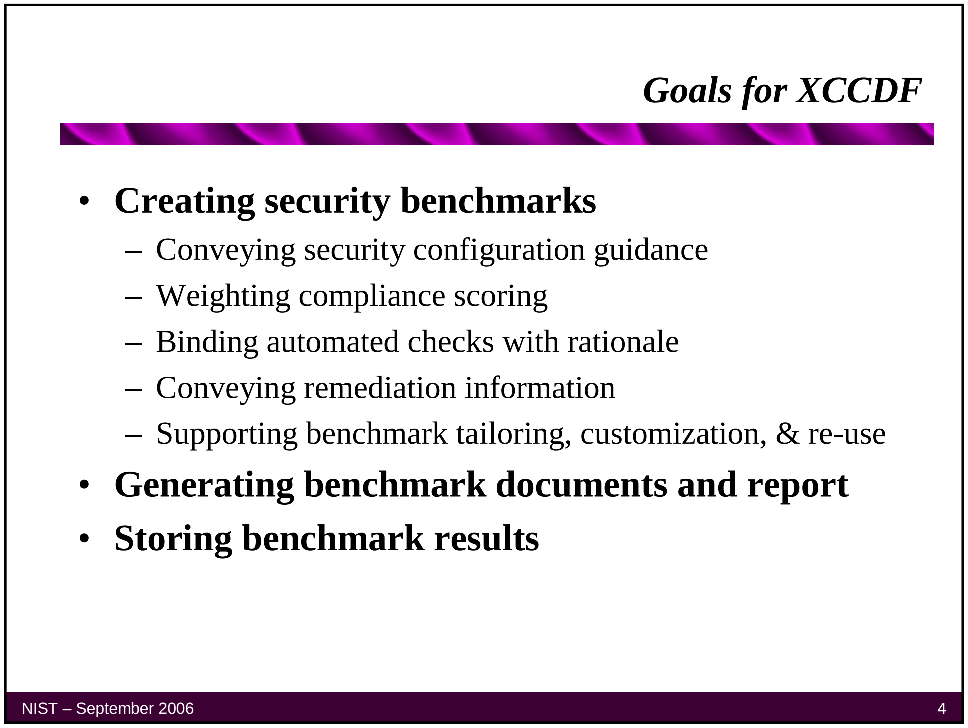
OUTLINE:

- Review of XCCDF structure
- Changes 1.0 to 1.1
- Changes for 1.1rev2
- “Opportunities for Improvement” in XCCDF

NIST – September 2006 2

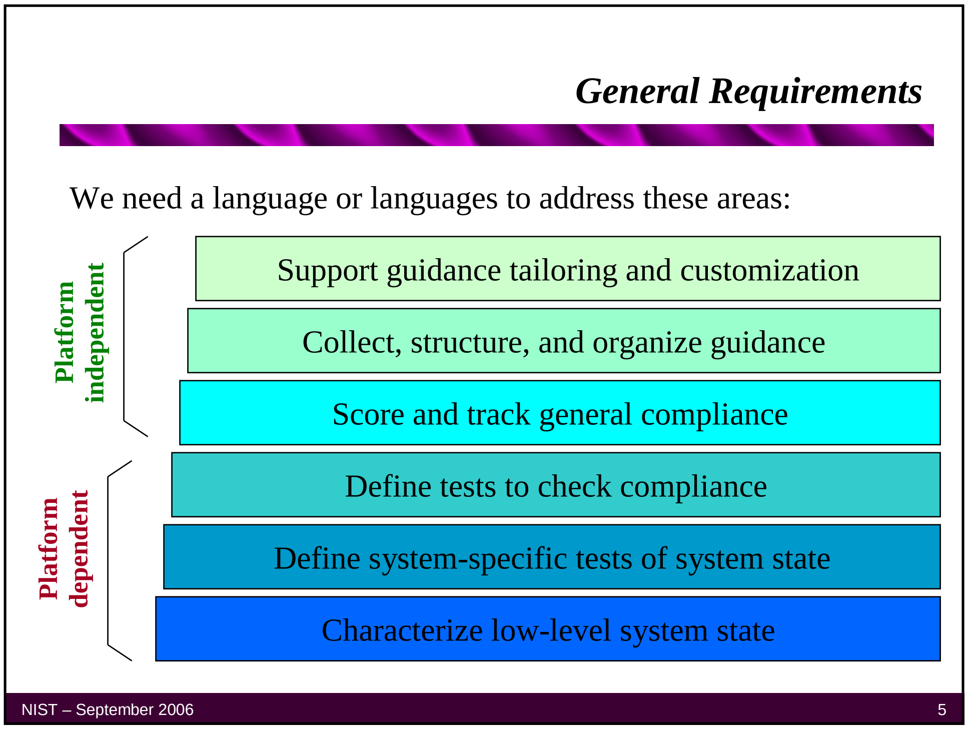


Review of XCCDF Structure



Goals for XCCDF

- **Creating security benchmarks**
  - Conveying security configuration guidance
  - Weighting compliance scoring
  - Binding automated checks with rationale
  - Conveying remediation information
  - Supporting benchmark tailoring, customization, & re-use
- **Generating benchmark documents and report**
- **Storing benchmark results**



General Requirements

We need a language or languages to address these areas:

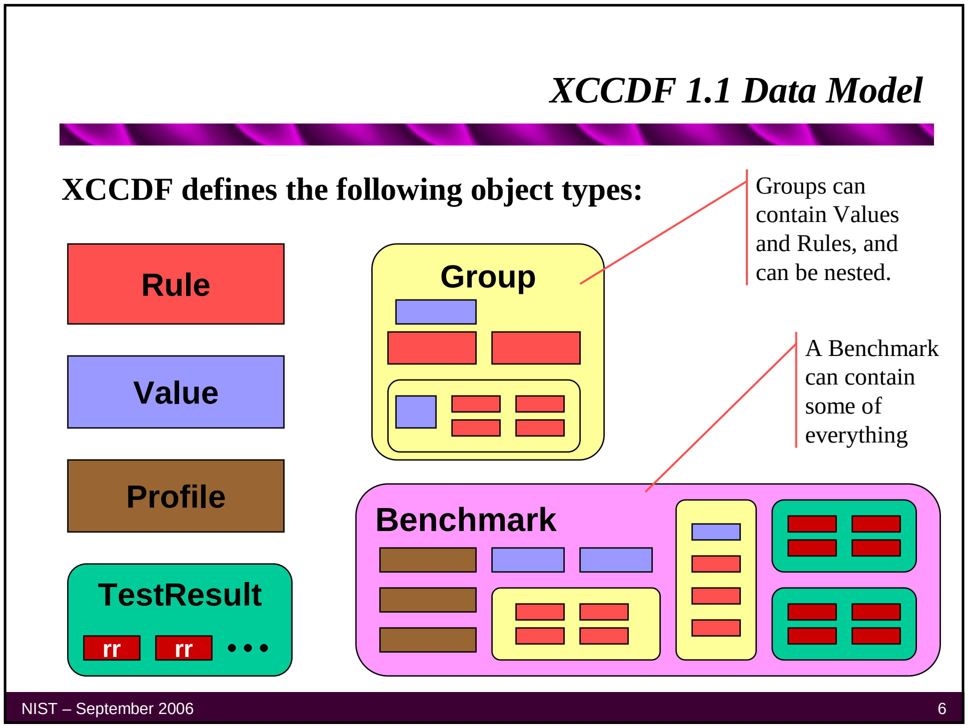
Platform independent

- Support guidance tailoring and customization
- Collect, structure, and organize guidance
- Score and track general compliance

Platform dependent

- Define tests to check compliance
- Define system-specific tests of system state
- Characterize low-level system state

NIST – September 2006 5



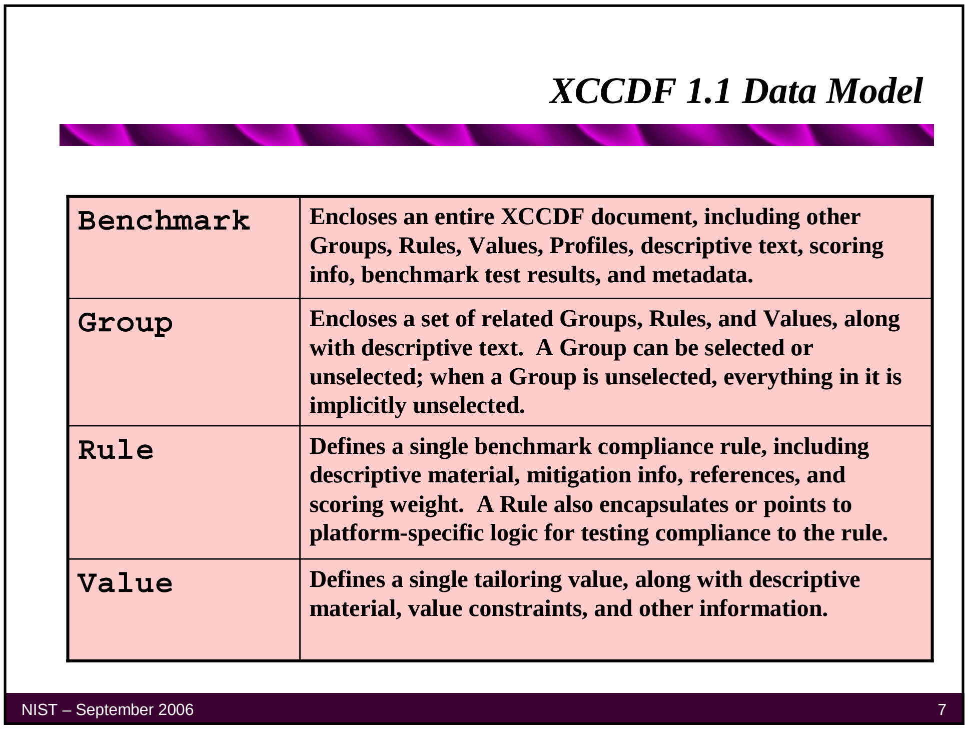
XCCDF 1.1 Data Model

XCCDF defines the following object types:

Groups can contain Values and Rules, and can be nested.

A Benchmark can contain some of everything

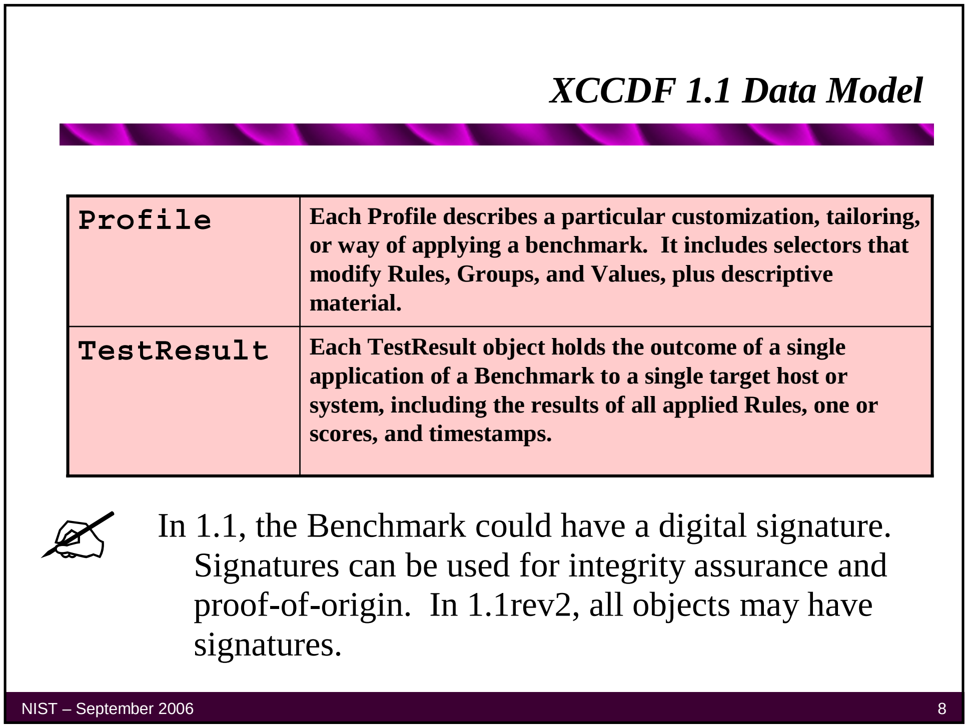
NIST – September 2006 6



XCCDF 1.1 Data Model

`Benchmark`	**Encloses an entire XCCDF document, including other Groups, Rules, Values, Profiles, descriptive text, scoring info, benchmark test results, and metadata.**
`Group`	**Encloses a set of related Groups, Rules, and Values, along with descriptive text. A Group can be selected or unselected; when a Group is unselected, everything in it is implicitly unselected.**
`Rule`	**Defines a single benchmark compliance rule, including descriptive material, mitigation info, references, and scoring weight. A Rule also encapsulates or points to platform-specific logic for testing compliance to the rule.**
`Value`	**Defines a single tailoring value, along with descriptive material, value constraints, and other information.**

NIST – September 2006 7

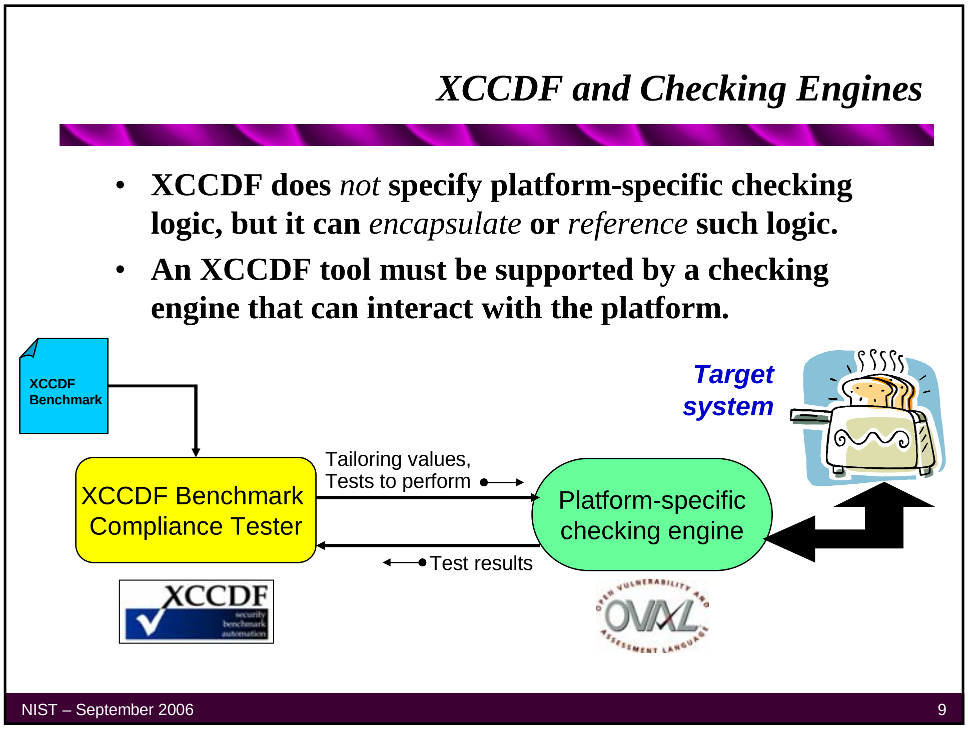


XCCDF 1.1 Data Model

`Profile`	**Each Profile describes a particular customization, tailoring, or way of applying a benchmark. It includes selectors that modify Rules, Groups, and Values, plus descriptive material.**
`TestResult`	**Each TestResult object holds the outcome of a single application of a Benchmark to a single target host or system, including the results of all applied Rules, one or scores, and timestamps.**

✍ In 1.1, the Benchmark could have a digital signature. Signatures can be used for integrity assurance and proof-of-origin. In 1.1rev2, all objects may have signatures.

NIST – September 2006 8



XCCDF and Checking Engines

- **XCCDF does *not* specify platform-specific checking logic, but it can *encapsulate* or *reference* such logic.**
- **An XCCDF tool must be supported by a checking engine that can interact with the platform.**

XCCDF Benchmark

XCCDF Benchmark Compliance Tester

Tailoring values, Tests to perform

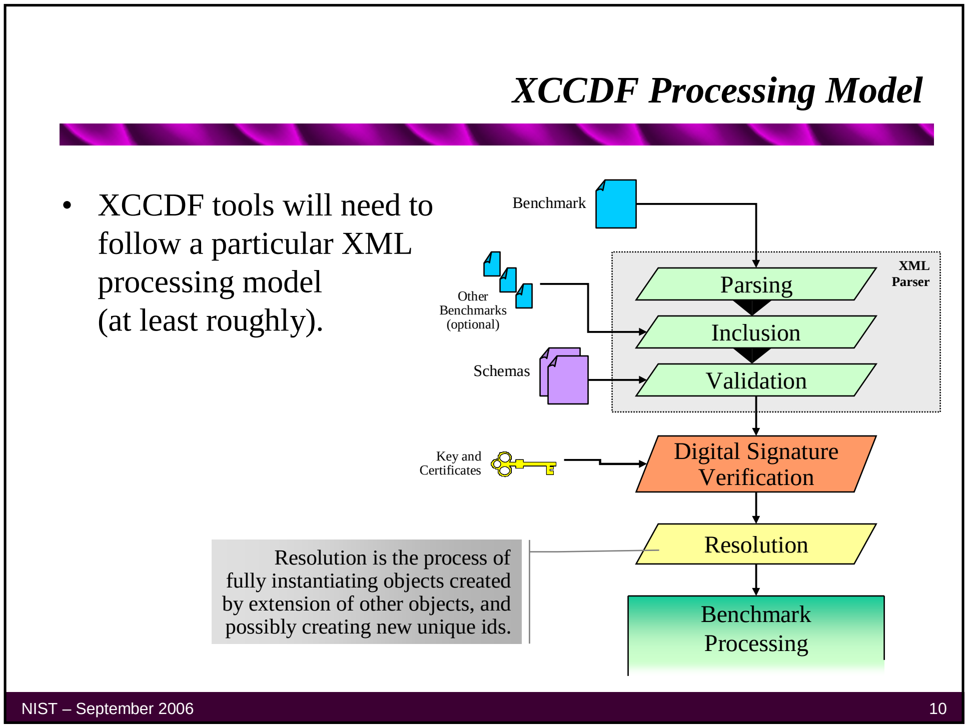
Test results

Platform-specific checking engine

Target system

XCCDF
security benchmark automation
OPEN VULNERABILITY AND ASSESSMENT LANGUAGE
OVAL

NIST – September 2006 9



XCCDF Processing Model

- XCCDF tools will need to follow a particular XML processing model (at least roughly).

Benchmark

Other Benchmarks (optional)

Schemas

XML Parser

Parsing

Inclusion

Validation

Key and Certificates

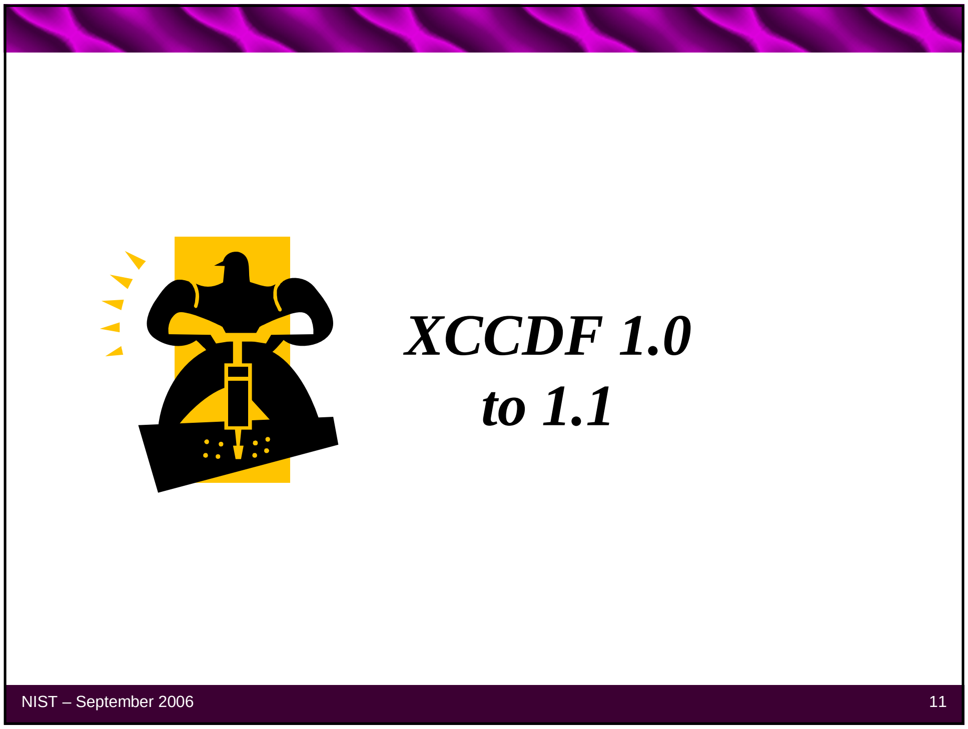
Digital Signature Verification

Resolution

Benchmark Processing

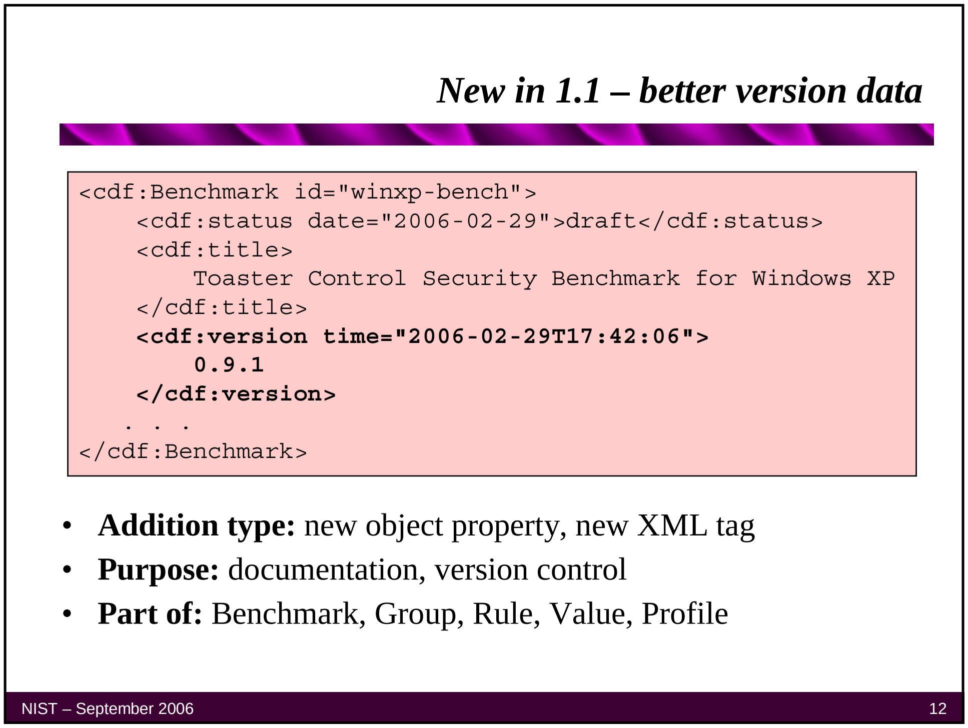
Resolution is the process of fully instantiating objects created by extension of other objects, and possibly creating new unique ids.

NIST – September 2006 10



XCCDF 1.0 to 1.1

NIST – September 2006 11

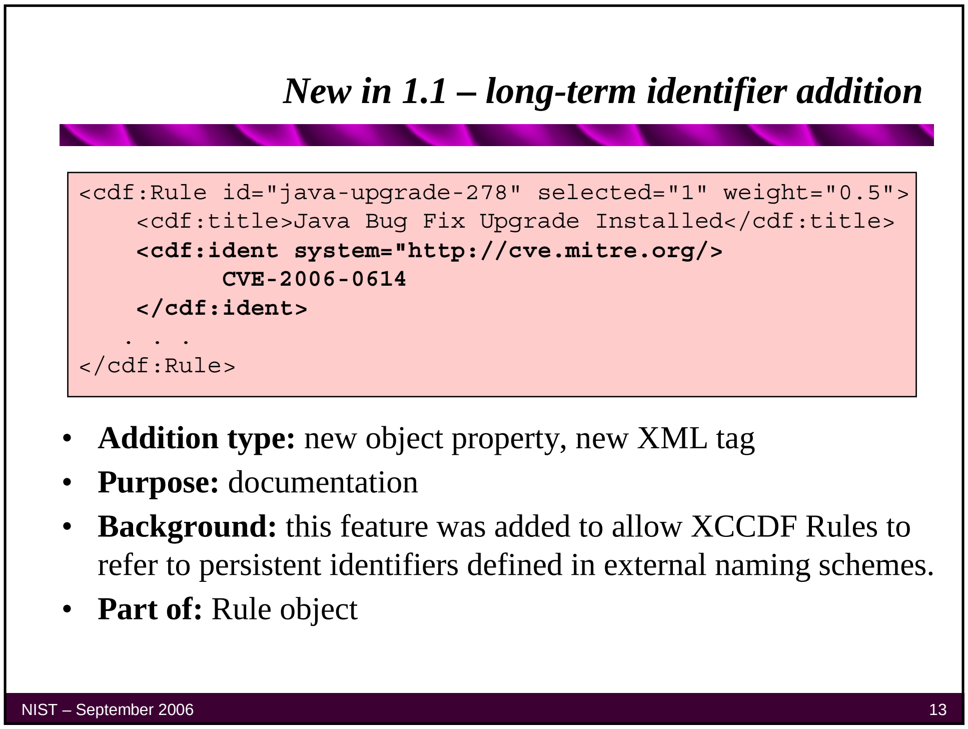


New in 1.1 – better version data

```
<cdf:Benchmark id="winxp-bench">
    <cdf:status date="2006-02-29">draft</cdf:status>
    <cdf:title>
        Toaster Control Security Benchmark for Windows XP
    </cdf:title>
    <cdf:version time="2006-02-29T17:42:06">
        0.9.1
    </cdf:version>
   . . .
</cdf:Benchmark>
```

- **Addition type:** new object property, new XML tag
- **Purpose:** documentation, version control
- **Part of:** Benchmark, Group, Rule, Value, Profile

NIST – September 2006 12

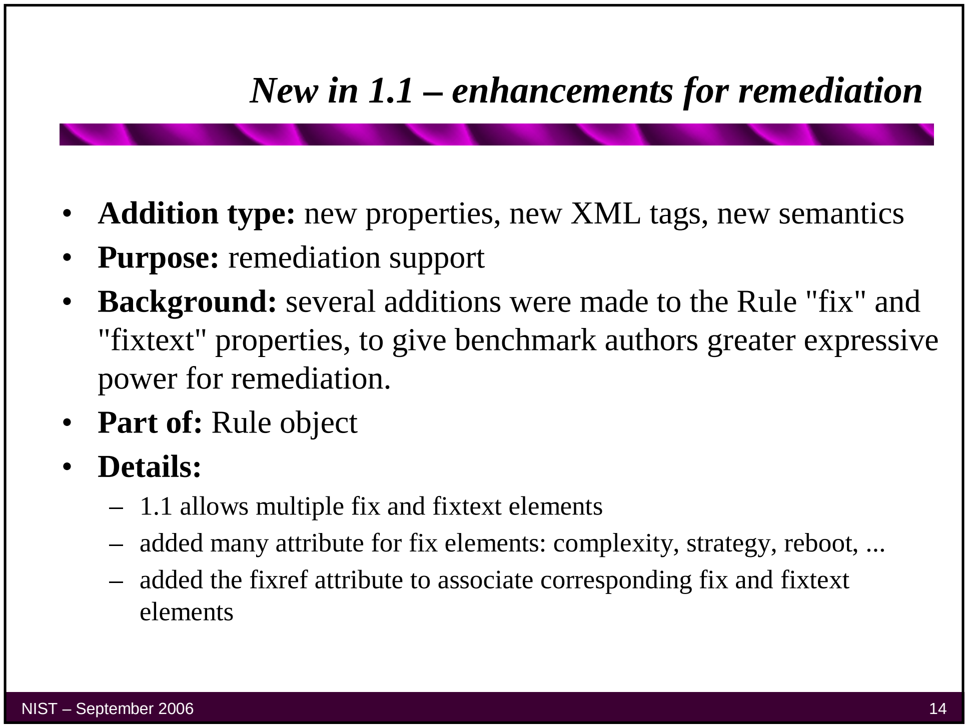


New in 1.1 – long-term identifier addition

```
<cdf:Rule id="java-upgrade-278" selected="1" weight="0.5">
    <cdf:title>Java Bug Fix Upgrade Installed</cdf:title>
    <cdf:ident system="http://cve.mitre.org/>
          CVE-2006-0614
    </cdf:ident>
   . . .
</cdf:Rule>
```

- **Addition type:** new object property, new XML tag
- **Purpose:** documentation
- **Background:** this feature was added to allow XCCDF Rules to refer to persistent identifiers defined in external naming schemes.
- **Part of:** Rule object

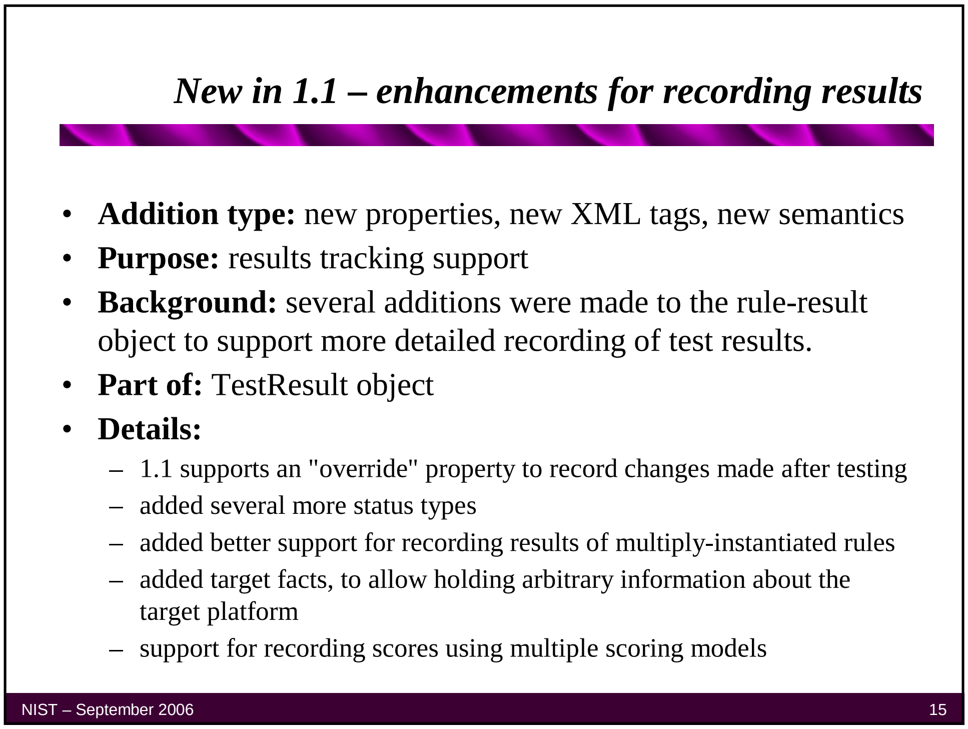
NIST – September 2006 13



New in 1.1 – enhancements for remediation

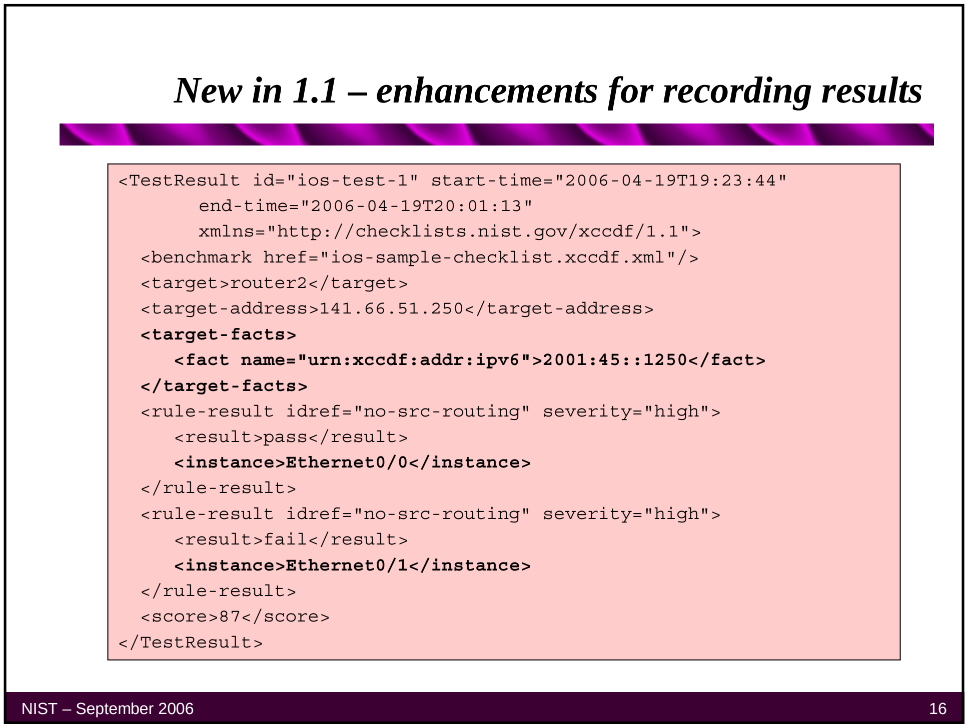
- **Addition type:** new properties, new XML tags, new semantics
- **Purpose:** remediation support
- **Background:** several additions were made to the Rule "fix" and "fixtext" properties, to give benchmark authors greater expressive power for remediation.
- **Part of:** Rule object
- **Details:**
  - 1.1 allows multiple fix and fixtext elements
  - added many attribute for fix elements: complexity, strategy, reboot, ...
  - added the fixref attribute to associate corresponding fix and fixtext elements

NIST – September 2006 14



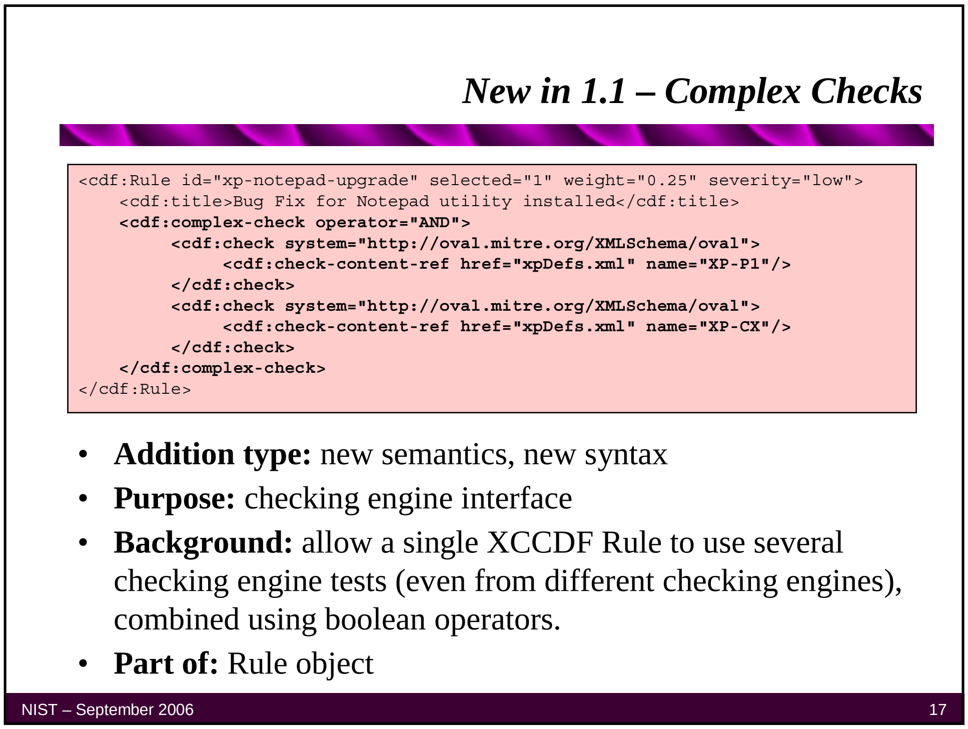
New in 1.1 – enhancements for recording results

- **Addition type:** new properties, new XML tags, new semantics
- **Purpose:** results tracking support
- **Background:** several additions were made to the rule-result object to support more detailed recording of test results.
- **Part of:** TestResult object
- **Details:**
  - 1.1 supports an "override" property to record changes made after testing
  - added several more status types
  - added better support for recording results of multiply-instantiated rules
  - added target facts, to allow holding arbitrary information about the target platform
  - support for recording scores using multiple scoring models



New in 1.1 – enhancements for recording results

```
<TestResult id="ios-test-1" start-time="2006-04-19T19:23:44"
       end-time="2006-04-19T20:01:13"
       xmlns="http://checklists.nist.gov/xccdf/1.1">
  <benchmark href="ios-sample-checklist.xccdf.xml"/>
  <target>router2</target>
  <target-address>141.66.51.250</target-address>
  <target-facts>
     <fact name="urn:xccdf:addr:ipv6">2001:45::1250</fact>
  </target-facts>
  <rule-result idref="no-src-routing" severity="high">
     <result>pass</result>
     <instance>Ethernet0/0</instance>
  </rule-result>
  <rule-result idref="no-src-routing" severity="high">
     <result>fail</result>
     <instance>Ethernet0/1</instance>
  </rule-result>
  <score>87</score>
</TestResult>
```

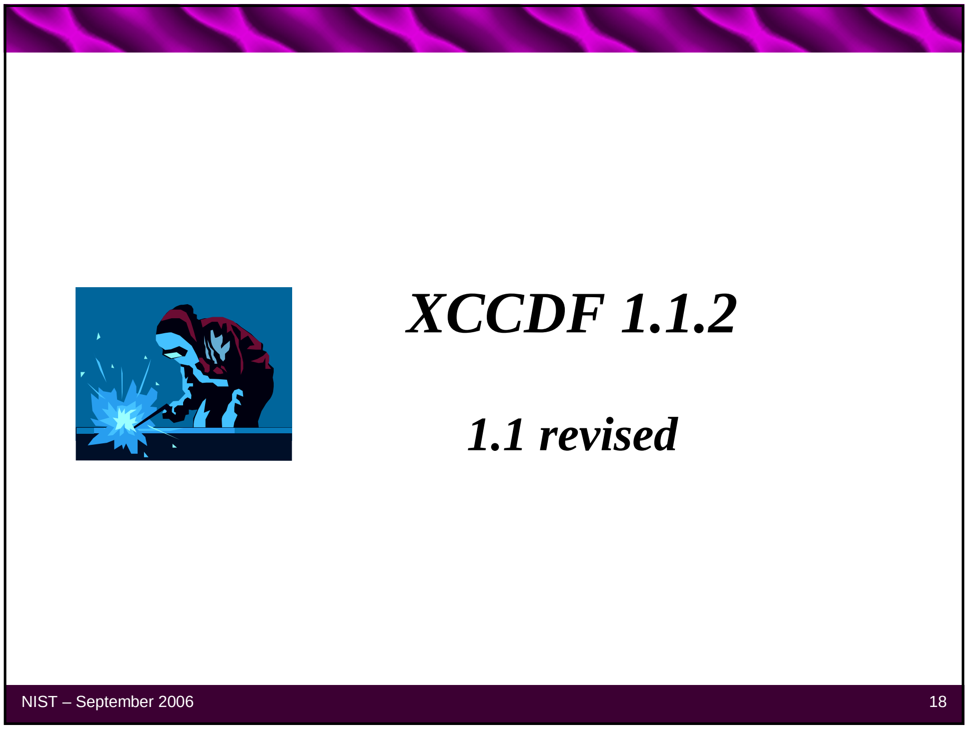


New in 1.1 – Complex Checks

```
<cdf:Rule id="xp-notepad-upgrade" selected="1" weight="0.25" severity="low">
    <cdf:title>Bug Fix for Notepad utility installed</cdf:title>
    <cdf:complex-check operator="AND">
         <cdf:check system="http://oval.mitre.org/XMLSchema/oval">
              <cdf:check-content-ref href="xpDefs.xml" name="XP-P1"/>
         </cdf:check>
         <cdf:check system="http://oval.mitre.org/XMLSchema/oval">
              <cdf:check-content-ref href="xpDefs.xml" name="XP-CX"/>
         </cdf:check>
    </cdf:complex-check>
</cdf:Rule>
```

- **Addition type:** new semantics, new syntax
- **Purpose:** checking engine interface
- **Background:** allow a single XCCDF Rule to use several checking engine tests (even from different checking engines), combined using boolean operators.
- **Part of:** Rule object

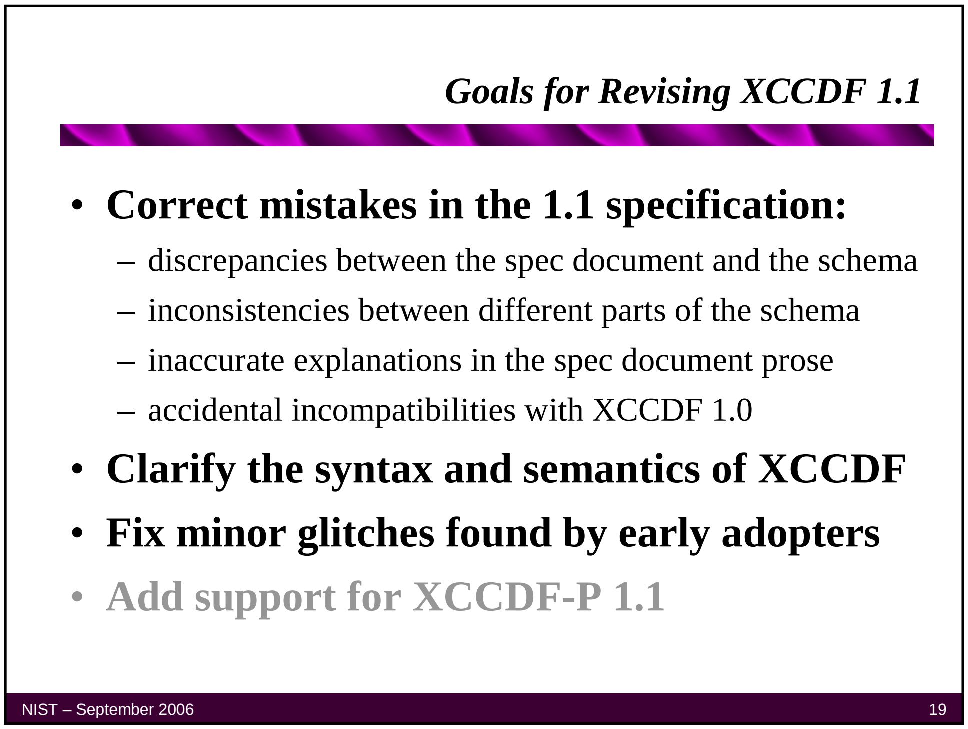
NIST – September 2006 17



XCCDF 1.1.2

1.1 revised

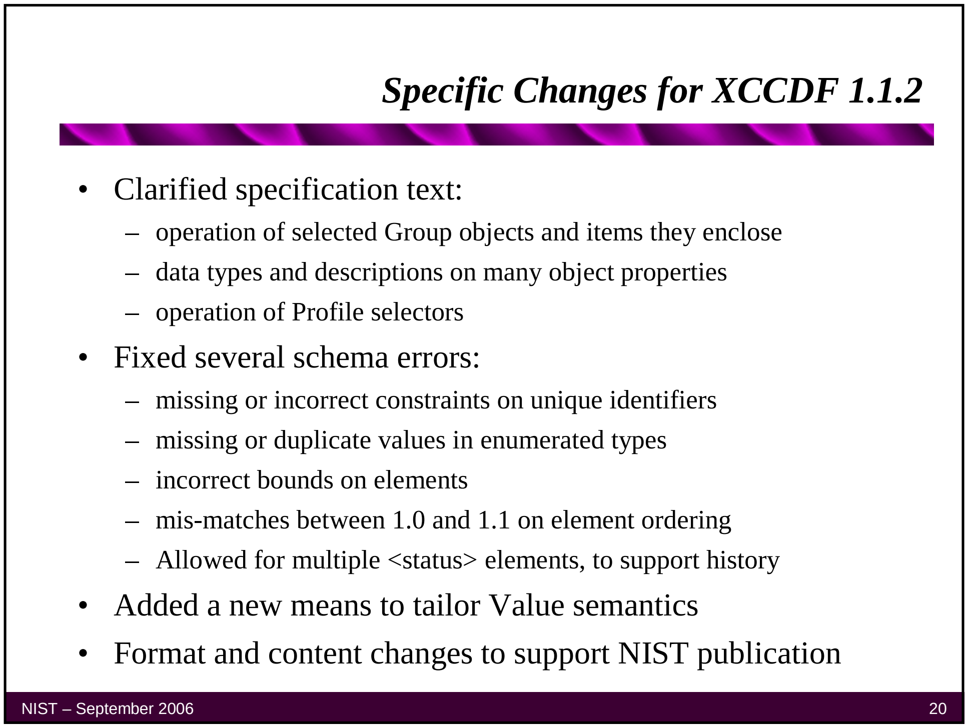
NIST – September 2006 18



Goals for Revising XCCDF 1.1

- **Correct mistakes in the 1.1 specification:**
  - discrepancies between the spec document and the schema
  - inconsistencies between different parts of the schema
  - inaccurate explanations in the spec document prose
  - accidental incompatibilities with XCCDF 1.0
- **Clarify the syntax and semantics of XCCDF**
- **Fix minor glitches found by early adopters**
- **Add support for XCCDF-P 1.1**

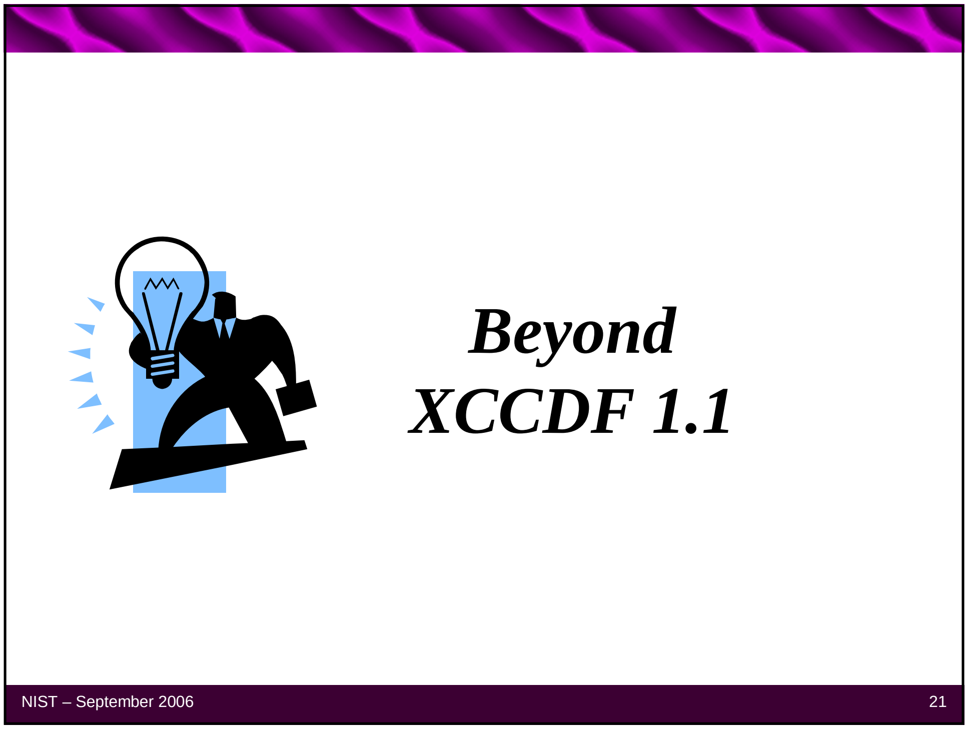
NIST – September 2006 19



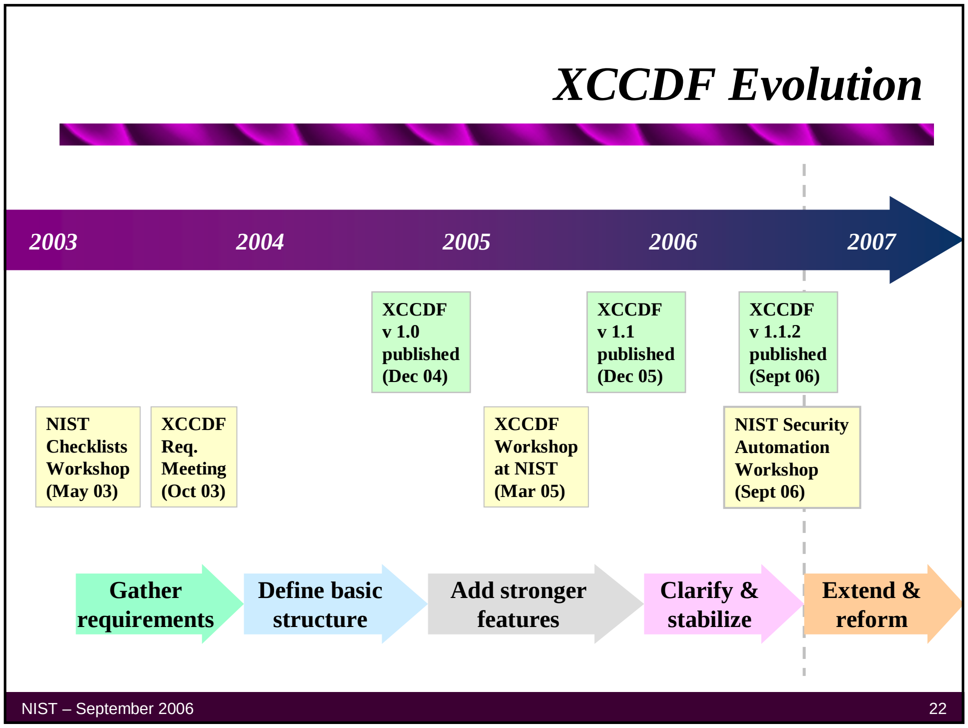
Specific Changes for XCCDF 1.1.2

- Clarified specification text:
  - operation of selected Group objects and items they enclose
  - data types and descriptions on many object properties
  - operation of Profile selectors
- Fixed several schema errors:
  - missing or incorrect constraints on unique identifiers
  - missing or duplicate values in enumerated types
  - incorrect bounds on elements
  - mis-matches between 1.0 and 1.1 on element ordering
  - Allowed for multiple <status> elements, to support history
- Added a new means to tailor Value semantics
- Format and content changes to support NIST publication

NIST – September 2006 20



Beyond XCCDF 1.1



XCCDF Evolution

XCCDF v 1.0 published (Dec 04)

XCCDF v 1.1 published (Dec 05)

XCCDF v 1.1.2 published (Sept 06)

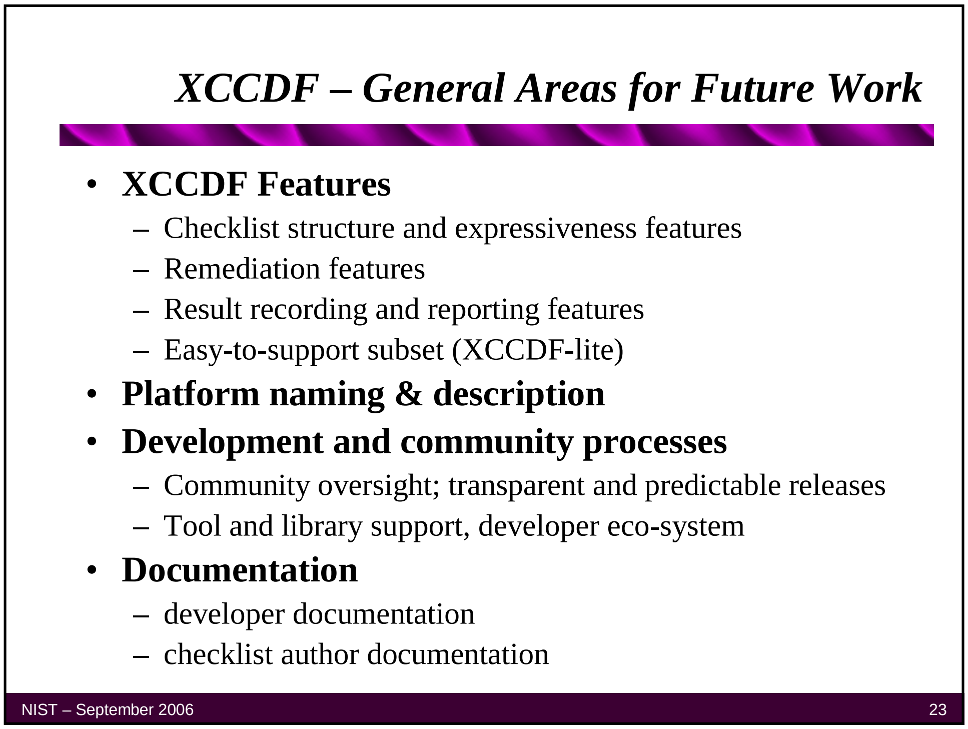
NIST Checklists Workshop (May 03)

XCCDF Req. Meeting (Oct 03)

XCCDF Workshop at NIST (Mar 05)

NIST Security Automation Workshop (Sept 06)

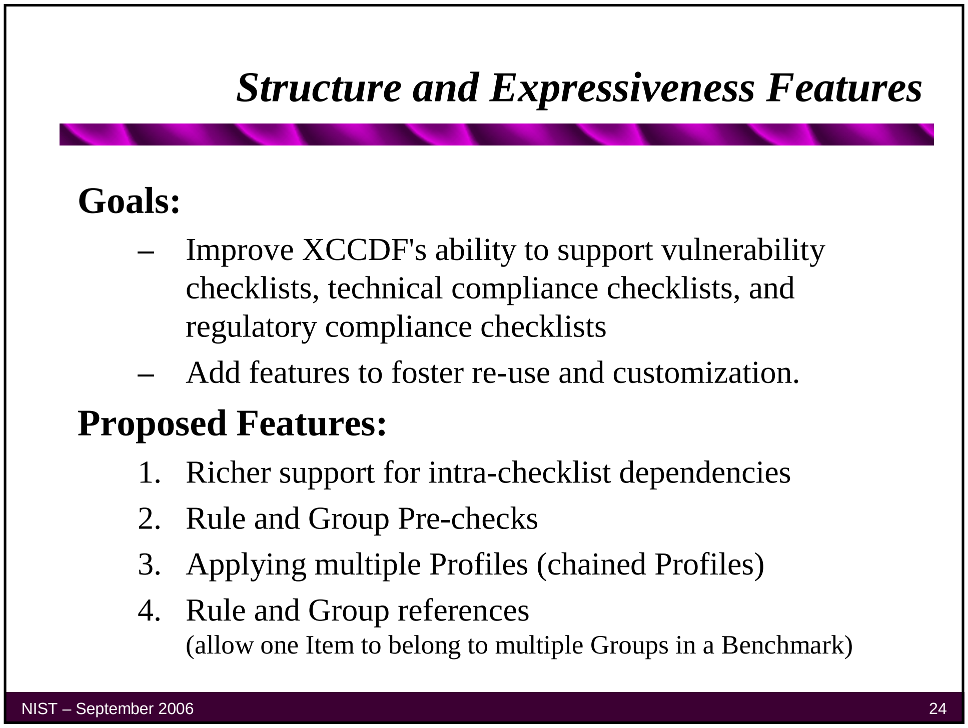
Gather requirements → Define basic structure → Add stronger features → Clarify & stabilize → Extend & reform



XCCDF – General Areas for Future Work

- **XCCDF Features**
  - Checklist structure and expressiveness features
  - Remediation features
  - Result recording and reporting features
  - Easy-to-support subset (XCCDF-lite)
- **Platform naming & description**
- **Development and community processes**
  - Community oversight; transparent and predictable releases
  - Tool and library support, developer eco-system
- **Documentation**
  - developer documentation
  - checklist author documentation

NIST – September 2006 23



Structure and Expressiveness Features

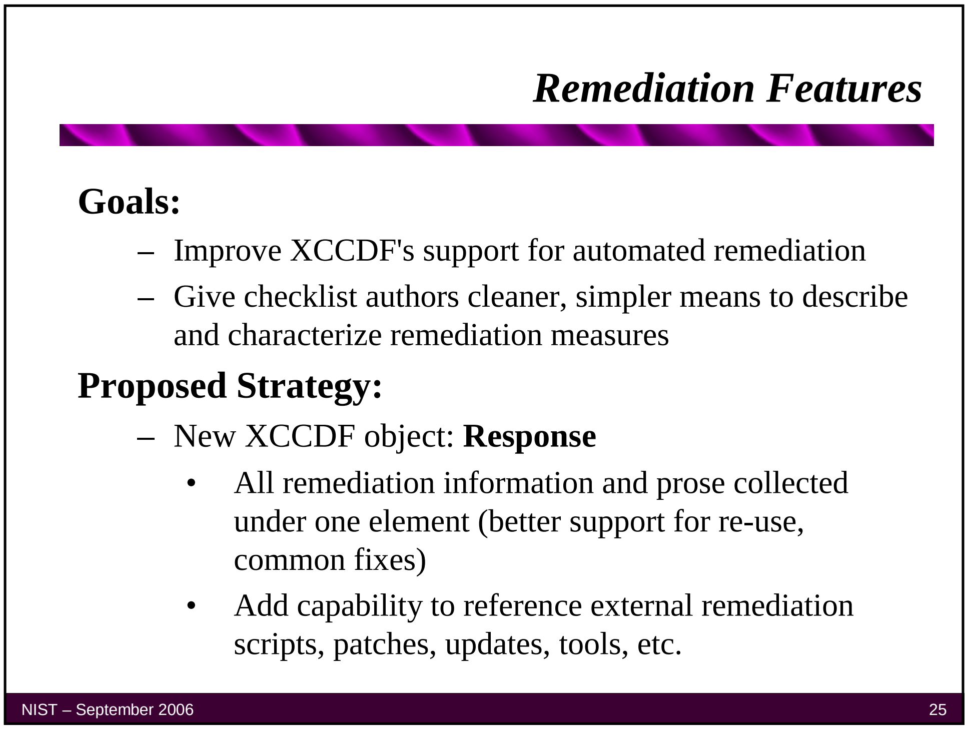
Goals:

- Improve XCCDF's ability to support vulnerability checklists, technical compliance checklists, and regulatory compliance checklists
- Add features to foster re-use and customization.

Proposed Features:

1. Richer support for intra-checklist dependencies
2. Rule and Group Pre-checks
3. Applying multiple Profiles (chained Profiles)
4. Rule and Group references
   (allow one Item to belong to multiple Groups in a Benchmark)

NIST – September 2006 24



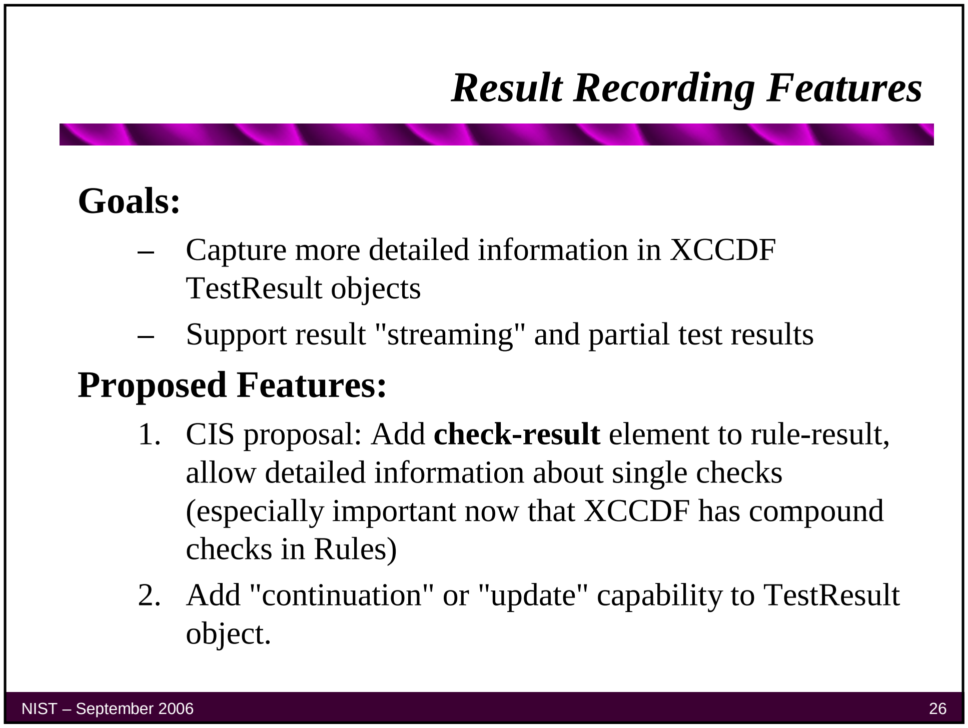
Remediation Features

Goals:

- Improve XCCDF's support for automated remediation
- Give checklist authors cleaner, simpler means to describe and characterize remediation measures

Proposed Strategy:

- New XCCDF object: **Response**
  - All remediation information and prose collected under one element (better support for re-use, common fixes)
  - Add capability to reference external remediation scripts, patches, updates, tools, etc.



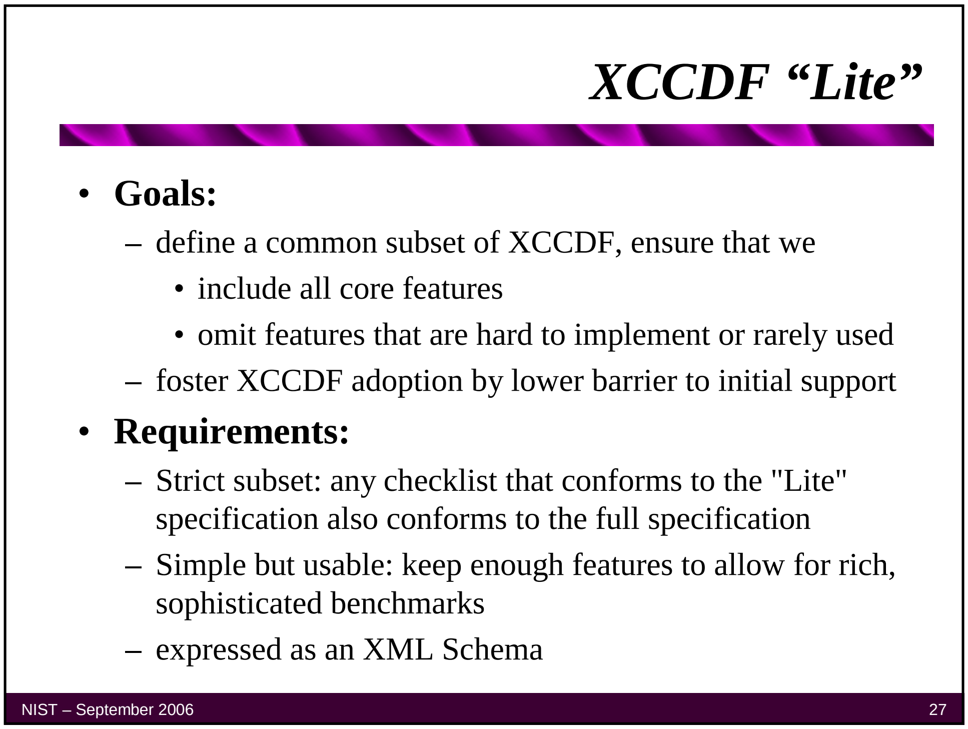
Result Recording Features

Goals:

- Capture more detailed information in XCCDF TestResult objects
- Support result "streaming" and partial test results

Proposed Features:

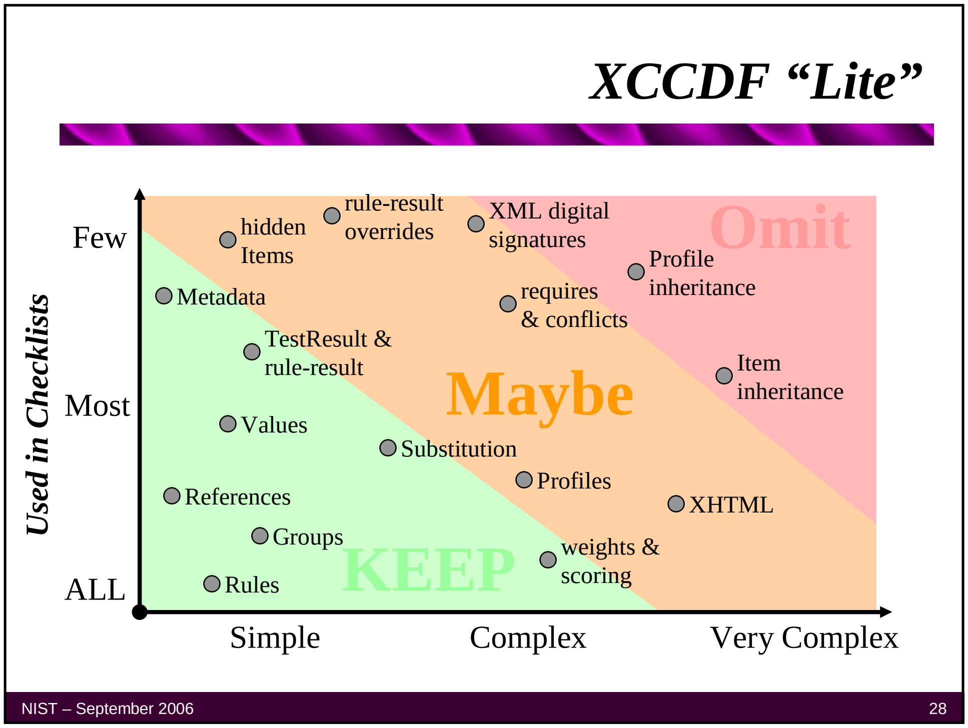
1. CIS proposal: Add **check-result** element to rule-result, allow detailed information about single checks (especially important now that XCCDF has compound checks in Rules)
2. Add "continuation" or "update" capability to TestResult object.



XCCDF “Lite”

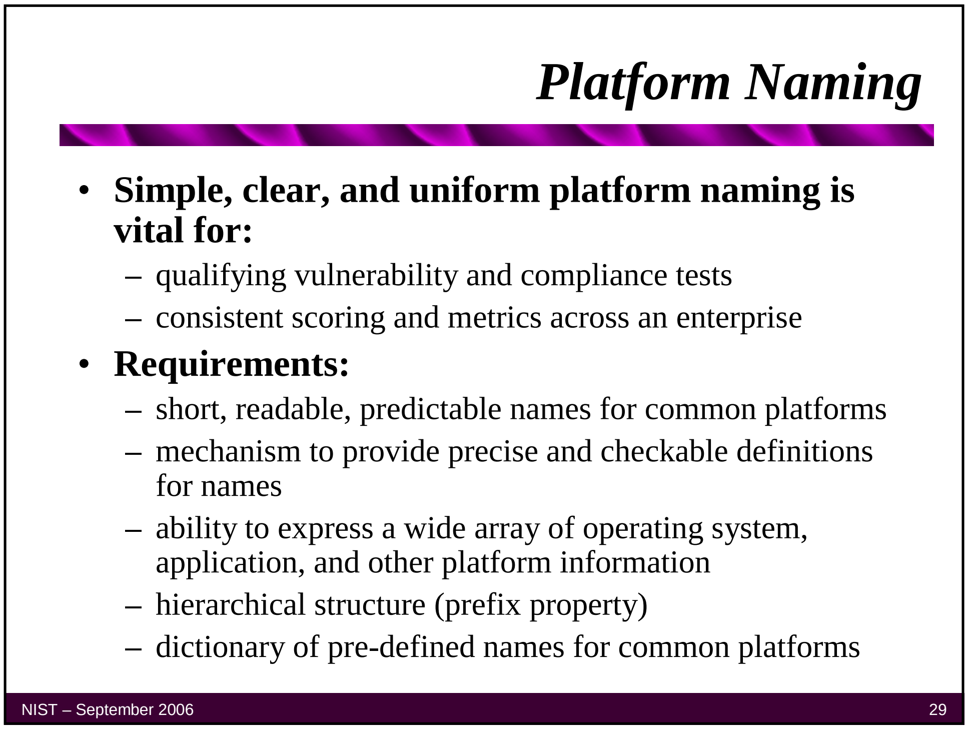
- **Goals:**
  - define a common subset of XCCDF, ensure that we
    - include all core features
    - omit features that are hard to implement or rarely used
  - foster XCCDF adoption by lower barrier to initial support
- **Requirements:**
  - Strict subset: any checklist that conforms to the "Lite" specification also conforms to the full specification
  - Simple but usable: keep enough features to allow for rich, sophisticated benchmarks
  - expressed as an XML Schema

NIST – September 2006 27



XCCDF “Lite”

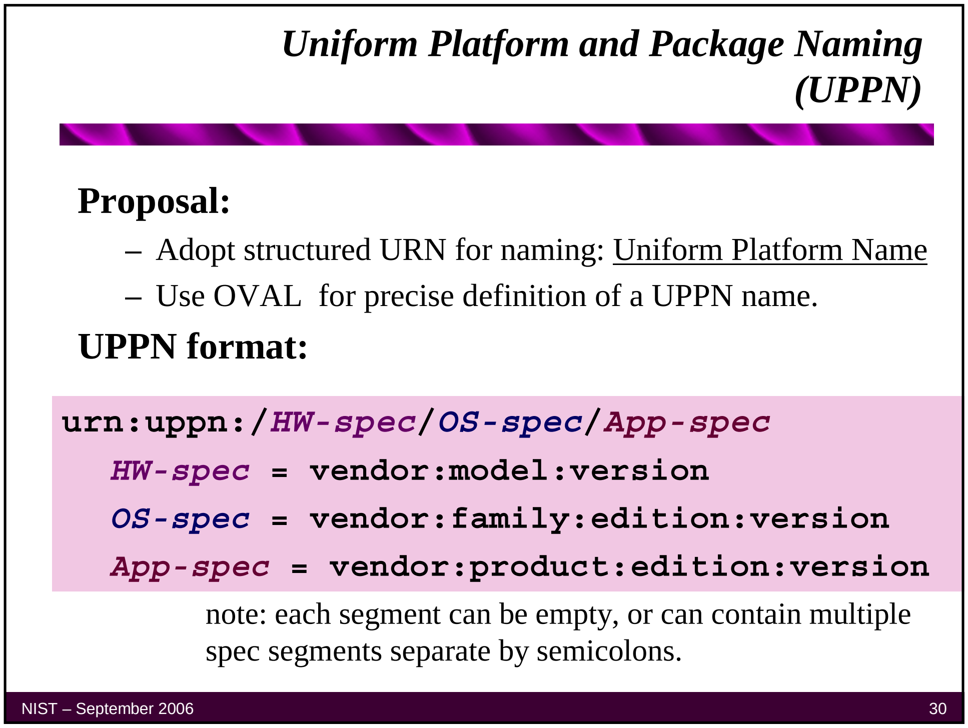
NIST – September 2006 28



Platform Naming

- **Simple, clear, and uniform platform naming is vital for:**
  - qualifying vulnerability and compliance tests
  - consistent scoring and metrics across an enterprise
- **Requirements:**
  - short, readable, predictable names for common platforms
  - mechanism to provide precise and checkable definitions for names
  - ability to express a wide array of operating system, application, and other platform information
  - hierarchical structure (prefix property)
  - dictionary of pre-defined names for common platforms

NIST – September 2006 29



Uniform Platform and Package Naming (UPPN)

Proposal:

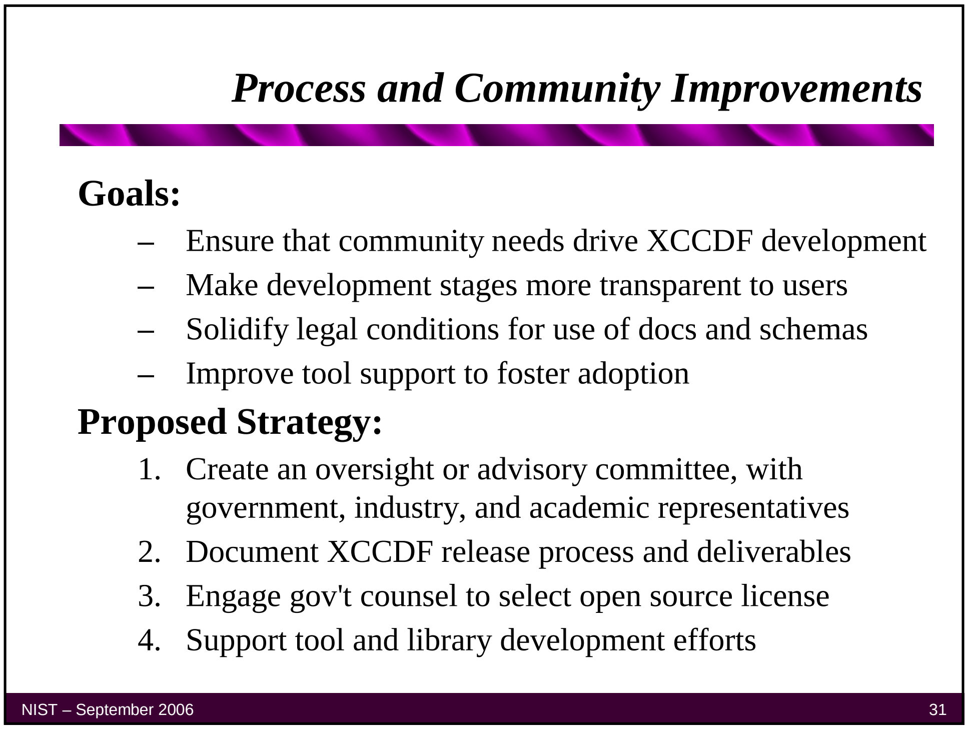
- Adopt structured URN for naming: Uniform Platform Name
- Use OVAL for precise definition of a UPPN name.

UPPN format:

```
urn:uppn:/HW-spec/OS-spec/App-spec
   HW-spec  = vendor:model:version
   OS-spec  = vendor:family:edition:version
   App-spec  = vendor:product:edition:version
```

note: each segment can be empty, or can contain multiple spec segments separate by semicolons.

NIST – September 2006 30



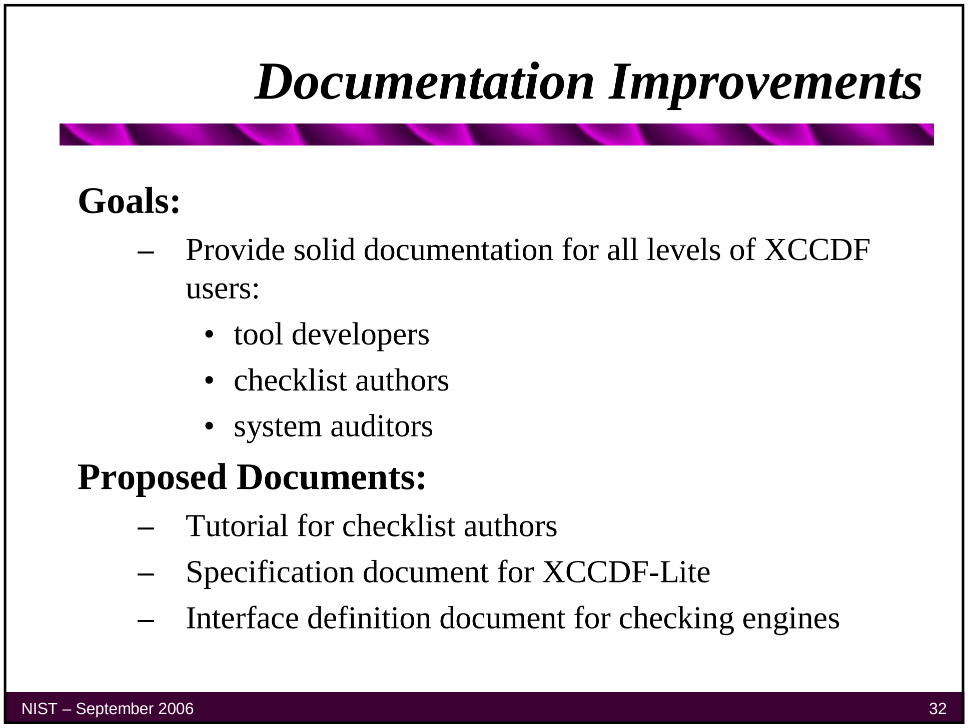
Process and Community Improvements

Goals:

- Ensure that community needs drive XCCDF development
- Make development stages more transparent to users
- Solidify legal conditions for use of docs and schemas
- Improve tool support to foster adoption

Proposed Strategy:

1. Create an oversight or advisory committee, with government, industry, and academic representatives
2. Document XCCDF release process and deliverables
3. Engage gov't counsel to select open source license
4. Support tool and library development efforts



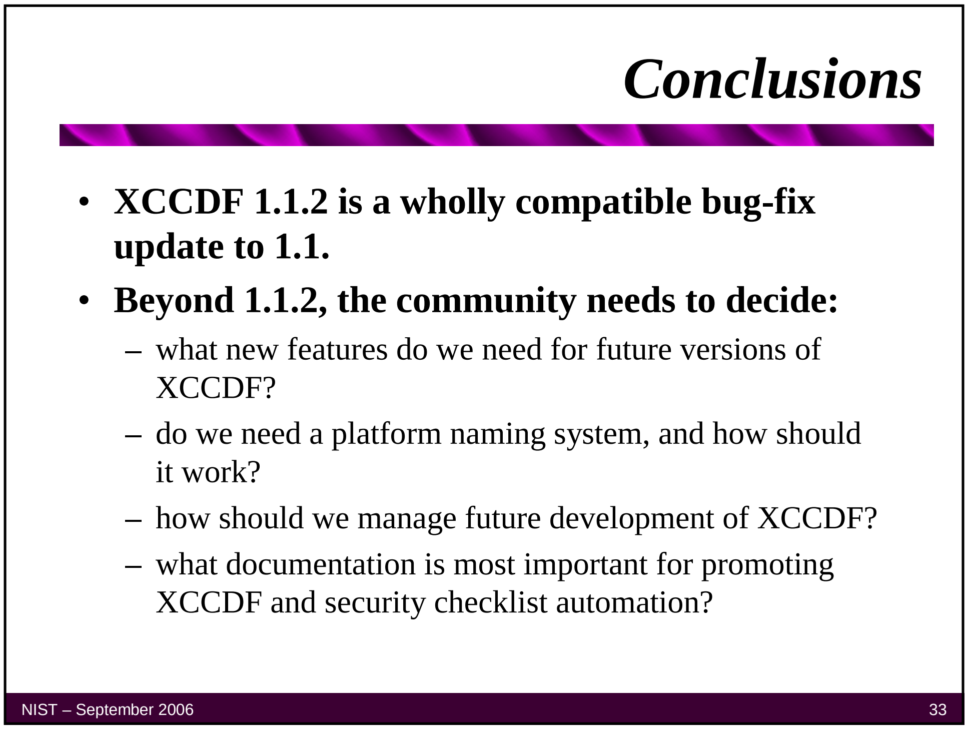
Documentation Improvements

Goals:

- Provide solid documentation for all levels of XCCDF users:
  - tool developers
  - checklist authors
  - system auditors

Proposed Documents:

- Tutorial for checklist authors
- Specification document for XCCDF-Lite
- Interface definition document for checking engines



Conclusions

- **XCCDF 1.1.2 is a wholly compatible bug-fix update to 1.1.**
- **Beyond 1.1.2, the community needs to decide:**
  - what new features do we need for future versions of XCCDF?
  - do we need a platform naming system, and how should it work?
  - how should we manage future development of XCCDF?
  - what documentation is most important for promoting XCCDF and security checklist automation?

NIST – September 2006 33