

Technology and Processes for Software Assurance Evaluations

W. Konrad Vesey

Tools & Techniques Lead

Center for Assured Software

National Security Agency



Technology and Processes for Software Assurance Evaluations

- √ Evaluation Problems
- √ Some Solution Requirements
- √ Use of Tools & Techniques
- √ Building an Evaluation Service Community



Evaluation Problems

- √ Vast amount of software
 - ◆ Growing diversity of program languages and characteristics
 - ◆ Growing dependence on external components
 - ◆ Software release cycles are approaching continuous evolution
 - ◆ More software is yet to be written than has been written so far



Evaluation Problems

- √ Current evaluation services are either inadequate or incommensurable or both
 - ◆ Many services look only at documentation and functional tests
 - ◆ Security evaluations generally rely on substantial manual review
 - ◆ Standard approach is to look for vulnerabilities vs. assess positive assurance
 - ◆ Software program managers are not able to separate the wheat from the chaff in evaluation tools
 - ◆ Skilled security evaluators are rare—evaluator training is even more rare
 - ◆ Evaluation results don't have a common meaning



Evaluation Problems

- v Evaluation science is inadequate and fragmented
 - ◆ Measurement of software assurance properties not widely studied or applied
 - ◆ Formal methods still a few years away from wide applicability



Some Solution Requirements

- √ Software Assurance Evaluations need to be:
 - ◆ Scalable
 - ◆ Repeatable
 - ◆ Goal-based vs. Flaw-based
 - ◆ Criteria-based vs. Tool-based
 - ◆ Useful
 - ◆ Measurably effective
 - ◆ Rational



Use of Tools & Techniques

- √ Assurance evaluations are served by the use of tools across the lifecycle
 - ◆ Requirements, Design, Implementation, Testing, Deployment
 - ◆ The use of tools early in the lifecycle can make evaluations later in the lifecycle more thorough, more efficient, and more accurate.
 - ◆ The use of tools later in the lifecycle verifies the correctness of the implementation.



Use of Tools & Techniques

√ Tool evaluations

- ◆ Tools need to be assessed to determine their true strengths and weaknesses
- ◆ Tool standards are helpful for integration or interoperability

√ Scientific Methods

- ◆ Assurance evaluation techniques need to be grounded in accepted science, not conventional wisdom



Building an Evaluation Service Community

√ Sharing results

- ◆ Common goals
- ◆ Common metrics
- ◆ Accepted well-documented processes
- ◆ Common assessment languages

√ Process improvement

- ◆ Evaluation framework within which incremental improvements can be shared and accumulated



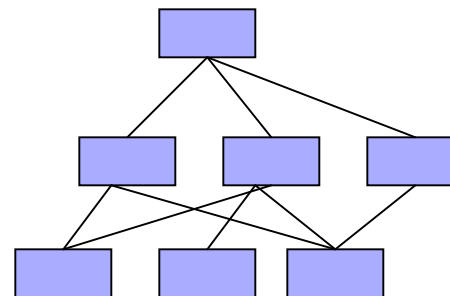
Building an Evaluation Service Community

- √ Example: The Virtual Binder
 - ◆ Assurance Goals
 - ◆ Assessment Languages/Interpretive Methodologies
 - ◆ Assurance Metrics
 - ◆ Measurement Strategies

Building an Evaluation Service Community

v Hierarchy of Assurance Goals

- ◆ A community of evaluation service providers agrees on a hierarchy of assurance goals
- ◆ Each goal definition includes a set of terms that express degrees of partial goal achievement.





Building an Evaluation Service Community

√ Assessment Languages

- ◆ A suite of report formats for communicating assurance evaluation results that are useful for a specific set of customer decisions
 - √ Binary: Deploy/Don't deploy
 - √ Unidimensional: 1-7, ★ ★ ★ ★
 - √ Multidimensional: "Consumer Reports" style comparison of multiple applications or of multiple aspects of a single application
- ◆ Interpretive methods for expressing degrees of partial goal achievement in terms of a chosen assessment language



Building an Evaluation Service Community

√ Assurance Metrics

- ◆ For each goal, metrics are identified that serve as proxies for direct goal measurement
- ◆ Metrics are applicable to an identified class of software
- ◆ Metrics are documented with specific configurations of tools and environments
- ◆ A catalog of metrics can be collected from diverse sources and published throughout the evaluation community



Building an Evaluation Service Community

✓ Measurement Strategies

- ◆ For each goal, an evaluation service provider selects a set of metrics that will represent a reasonably independent suite of proxy measurements
- ◆ These strategies are published with the report
- ◆ Measurements are captured with respect to these metrics and the results are reported in an assessment language appropriate to the needs of the evaluation customer



Building an Evaluation Service Community

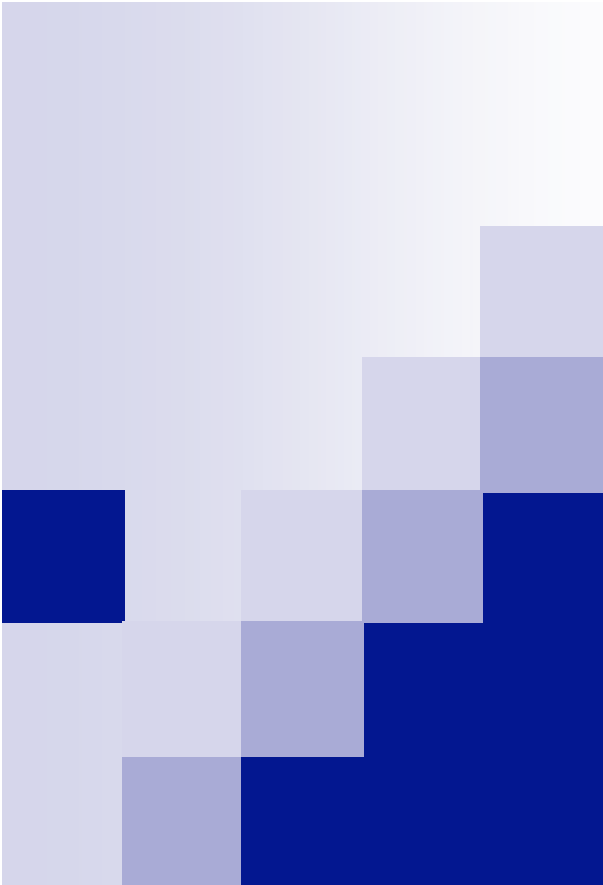
v Applying the process

- ◆ Determine if suitable measurement strategies are available to evaluate the customer's class of software
- ◆ Select an assessment language compatible with the customer's purpose for the evaluation
- ◆ Select measurement strategies for the class of software and apply them using principles of Bayesian inference to determine a probability distribution over the goal achievement terms given the measurement results. This distribution characterizes the relative likelihood that a given goal achievement term is the most accurate description of the software under evaluation
- ◆ Probabilities for degrees of goal achievement are combined in the goal hierarchy to derive distributions over parent goals
- ◆ The resulting likelihoods of partial goal achievement are expressed in the assessment language



Building an Evaluation Service Community

- v Collaboration within the community
 - ◆ Evaluation service providers can produce reports that have a known degree of commensurability—critical for informed comparisons
 - ◆ Measurements for individual metrics can be captured at multiple locations and then combined to generate goal and overall results
 - ◆ A measurement strategy used by one provider can be extended by another with almost no duplication of work
 - ◆ Raw tool output need not be distributed with the report in order for it to be meaningful and useful



Technology and Processes for Software Assurance Evaluations

W. Konrad Vesey

Tools & Techniques Lead

Center for Assured Software

National Security Agency