# CMU Task RT-119

—

# Systemic Assurance

Bill Scherlis
Professor and Director
Institute for Software Research (ISR)
School of Computer Science (SCS)
scherlis@cmu.edu

Final
Nov 2016



The aim of any testing scheme is to ensure that the customer gets substantially the software that he ordered and it must provide the customer with convincing evidence that this is so.

— NATO Software Engineering report 1968 .

institute for SOFTWARE RESEARCH
School of Computer Science
Carnegie Mellon

## RT-119 Systemic Assurance – Focus and Themes

- Game-change approaches to scalable systems assurance: recertification
  - Continual evaluation for continually evolving software-reliant systems
  - Direct evaluation based on accumulation of evidence:
    - Across all lifecycle stages
  - Evidence as additional engineering artifacts
    - Models, analyses, linkages
  - ROI for developers and evaluators:
    - Incremental benefits and amortized costs

> The aim of any testing scheme is to ensure that the customer gets substantially the software that he ordered and it must provide the customer with convincing evidence that this is so.
>
> — NATO Software Engineering report 1968

- Software-reliant SE domains
  - Self-adapting systems for resiliency, security, CPS, etc.
    - Emerging area of concern: AI and autonomy
  - Complex framework-based and web-based systems
    - Component-based and diversely sourced



## RT-119 Systemic Assurance – Focus and Themes

- Technical themes
  - Evidence, traceability, and use of data
    - Accumulation of assurance-related evidence
    - Creation of traceability structures during development
  - Direct analysis of artifacts
    - Semantics-based techniques for frameworks, protocols, concurrency, etc.
    - Enhance confidence, scalability, cost, and devt/evolution tempo
  - Requirements and architecture support for assurance
    - Drives potential to assure, support for variabilities, resiliency
    - Address assurance goals at the earliest development phases
  - Combined methods for heterogeneous systems
    - Combining informal/formal, static/dynamic/isolation, devt/ops
    - Enable composition of judgments
    - Assure as you go – limit both “assurance risk” and technical debt

## CMU RT-119 faculty areas of primary focus

- Faculty
  - Bill Scherlis, PI
  - Jonathan Aldrich
  - Christian Kästner
  - Joshua Sunshine
  - Travis Breaux
  - Claire LeGoues
  - David Garlan
  - Bradley Schmerl
  - Javier Cámara

Analysis, modules, APIs

Requirements and policy

Testing and repair

Resilient/adaptive architecture, CPS

- PhD students
  - Waqar Ahmad, Jaspreet Bhatia, Zack Coker (NSF Fellow), Vishal Dwivedi, Gabriel Ferreira, Thomas James Glazier, Mauricio Soto Gonzales, Hanan Hibshi, Darya Kurilova, Ivan Ruchkin, Daniel Smullen, Roykrong Sukkerd



## Background: Assurance and modern systems

Challenges of modern systems – embracing rapid capability enhancement

- Integrations include hardware, software, and human operators
  - Integrations are more complex and hererogeneous, with larger numbers of components
- Operating environment involves interlinking of systems
  - Integrations across weapon systems and business systems
  - Use of mobile and personal devices
  - Diverse civil and coalition and international partners
- Supply chains are complex, extensive, and geographically diverse
  - Libraries, frameworks, generators
  - Diverse sourcing
  - Modern socio-technical ecosystems are proliferating: mobile, big data, …
- Systems are more autonomous (AI based) and actively resilient
  - Rules of engagement are embodied in the systems
  - Systems can learn and self-adapt
  - Systems may interact with human operators and with other smart systems
  - Compliance responsibility moving from operators to evaluators
- Systems are under continuous attack (network and supply chain)
  - They should degrade gracefully
  - Reliability is influenced by software response to hardware faults and human errors

## Background: Assurance and current practices

Challenges of current practices – overcoming adverse norms

- Requirements for higher assurance and greater complexity
  - Higher levels of assurance are needed for a wider range of systems
  - Architectures are complex, multi-sourced, with internal trust gradients
  - Adversaries are now becoming highly sophisticated
- Process compliance does not assure quality
  - There is no substitute for direct evaluation of artifacts
  - Process compliance can create inappropriate incentives
- Product-focused T&E practices present difficulties
  - After-the-fact product practices are out of phase with modern devt and tool reality
    - Models, rationale, and evidence must be obtained by reverse engineering
    - Poor support for incremental or iterative development approaches
    - Poor support for evaluation of self-adapting systems
  - Engineering approaches / tools are not at pace with cybersecurity requirements
- Many evaluation practices and standards are out of phase
  - After the fact – information loss and reverse engineering requirements
  - Technical difficulties – concurrency, cloud, autonomy
  - System snapshot – no dynamism, few configurations
  - Whole system – vs incremental re-certification
  - No verification – design focus (even at EAL7, A1)
  - IP difficulties – evaluator; prime, third parties



## The four threads of RT-119 activity

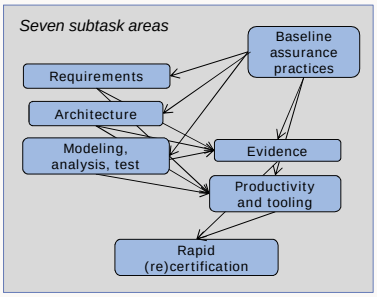
1. Baseline analysis of codified existing best practices
   - Under preliminary "warm-up" consideration:
     - DoD 8500.02, RMF (NIST 800-171, 800-53), STIGs, 5000.x for secure IT
     - DO 178C, DO-333 for avionics
     - CC ISO 15408 and NIAP for security
     - IEC 62304, ISO 14971 for embedded medical

Each team member is involved in *all* four threads

2. Identification and validation of meta-criteria
   - Encompass technical, business, and SE/heterogeneity dimensions
3. Advancement in four areas of technical practice
   - Requirements and validation
   - Architecture and resiliency
   - Technical evidence and analytics
   - Design and code linkage and infrastructure for recertification
4. Concepts for improved best-practice standards based on evidence
   - Build on technical advances, with meta-criteria scrub

## Agenda

- Focus and themes
- Team
- Challenges for assurance
- Task conops – four threads
  1. Practices baselining
  2. Meta-criteria identification
  3. Models, analyses, validation
  4. Improvements to practice
- Summary of status
  - Synergetic activities
- Meta-criteria
- Practices baselining
  - Interviews
- Models, analyses, validation
  - 1. architecture
  - 2. requirements
  - 3. test and repair
  - 4. modular analysis
  - 5. configurations






## Summary of status at conclusion of task

- Practices baselining
  - Summary descriptive analysis nearly complete
  - IRB approval and SME interviews initiated, approx. 20 SMEs interviewed
  - Post-task plan: continue SME interviews; publish analysis
- Meta-criteria
  - Initial set advanced and refined through practices baselining
  - Post-task plan: continue refinement and publish
- Technical advances – areas of emphasis
  - Architecture
    - Autonomy; Legacy recovery; SE and CPS multi-models; code
  - Requirements
    - Evaluator language
  - Modular analyses and defects
    - Composition; Automatic repair; Configurations and variability
- New-generation assurance approaches
  - Preliminary concepts for evidence, models, and dependencies
  - "Sweet spots" in the meta-criteria landscape
  - Post-task plan: continue advancement of evidence-based structures
    - Hazard/safety cases. Argumentation structures. Mathematical analytics.

## Summary of status – recognition; synergies

- Extensive publications in technical areas
  - About 25 papers published and many others in process
    - Mostly top conferences (note CRA evaluation guidance)
  - Awards
    - Best Paper award (CBSE'15)
    - Gold Medal winner (ACM Student Research Award Competition)
- Synergies
  - Nuclear Regulatory Commission (NRC) testimony – Dec 2015
    - Digital Instrumentation and Control (I&C) – prospects for assurance of safety
  - Sponsorship with White House OSTP of a public workshop – June 28, 2016
    - Safety and Control for Artificial Intelligence
  - Hosted HotSoS 2016 – April 19-21, 2016 (NSA sponsored)
    - Science of Security Five Hard Problems
      - (1) Scalability and composability in the construction of secure systems
      - (4) Resilient architectures that can deliver service despite compromised components
    - Focus on assurance and autonomy; sponsors include NSA and NSF
  - Collaboration with Software Engineering Institute – ongoing
    - Line technical themes: assurance and autonomy
    - Nuclear Regulatory Commission: evidence and dependencies
    - Response to Congressional staff queries regarding SE issues
  - DARPA BRASS (building resource adaptive software systems) – Aldrich, Sunshine
    - Model-based adaptation for robotic systems (MARS)
    - Evidence and linking of evidence



## Meta-criteria, v0.4

- Roles for meta-criteria in the Task
  - Evaluate selected baseline standards and best practices
  - Evaluate technical advances in reqts, architecture, design, evaluation
  - Assess value and feasibility of new concepts for assurance evaluation
- Dimensions of meta-criteria
  - Technical factors
    - Quality attribute focus, soundness, evidence, models, etc.
  - Structural factors
    - Support for composition, ecosystems, frameworks, components, etc.
  - Process factors
    - Timing, management, orgn, support for evolution, recertification
  - Affordability factors
    - Cost, risk, visibility and access, incentives, etc.
  - Acquisition and business factors
    - Incentives, roles and stakeholders, IP and observables, access, etc.
- Iterative validation and scrub-down of meta-criteria
  - Direct: Task roles in our evaluations of practices and techniques
  - Indirect: SMEs and initial interview results
  - Currently: 17 meta-criteria identified and articulated

## Meta-criteria, v0.3, items 1 – 9

1. Specific technical quality attributes addressed and overall level of quality attainable and assurable for each
2. Trustworthiness/validity of results – from sound verification to heuristic correlates
3. Phases of process where evaluation activities are undertaken – ranging from early (requirements and architecture) to after-the-fact
4. Access required by evaluators to supplier intellectual property and artifacts
5. Role of evaluation considerations in architectural decisions and implementation choices
6. Role of process indicators versus direct examination of development artifacts
7. Ability to reuse evidence from prior evaluations for incremental re-evaluation and recertification – status of evidence produced
8. Diversity of kinds of evidence to support judgments – kinds of models, informal/formal, linkage and traceability, etc.



## Meta-criteria, v0.3, items 10 – 17

9. Up-front investment (tooling, training) and ongoing cost (based on complexity and scale)
10. Benefits to overall development and sustainment cost and schedule; enhancements to engineer productivity and risk management
11. Composability of results for components, libraries, and frameworks in evaluating aggregates
12. Support for integration within ecosystems: mobile devices, big-data analytics, graphical interaction, etc.
13. Inter-rater reliability in evaluation, including ability to assess and extent of existing assessment
14. Incentives for developers to produce evidence to be used by evaluators
15. Incentives for evaluators to publish evidence back to developers and to end clients
16. Risks of incorrect assessments and assurance judgments
17. Skill requirements for evaluators

## Practices baselining – initiation of SME interviews

- Areas of focus – on-paper evaluation-practice analyses nearly complete
  - DO 178C (and DO-333, DO 178B) – Avionics focus, used by FAA
  - NIAP Common Criteria ISO-15408 – Security evaluation, ex Orange Book
  - Risk Management Framework (RMF – NIST 800-171, 800-53) – just initiatied
  - DoDI 8500.01, Application Security STIG – Security risk management
  - DoDD 5000.02 software-intensive, 3000.09 autonomy – Software assurance metrics and risk remediation
  - IEC 62304 medical device software – Process standard, used by FDA
- SME interviews to gain ground truth – 20 in-depth interviews (to date)
  - Following IRB-approved protocol (to enable publication)
  - Multiple RT-119 investigators have access
- Evaluation of Microsoft SDL experience (Lipner and Howard)
  - Full day discussion with Steve Lipner at CMU – highlights:
    - Central team support and expertise
    - Thread/hazard/safety modeling
    - Dependency modeling and traceability
      - Importance of SDL Tracking Tool
- We seek input/feedback regarding:
  - Selection and focus
  - Access to SMEs
  - Realism and validation for meta-criteria


School of Computer Science

15


## Interviews: Standards, Meta Criteria, …

- SME interview script based on meta criteria
  - Validate meta-criteria and policy perspective
  - Confirm/revise identified technical issues in the various standards
- To date: Formal interviews of about twenty stakeholders
  - Roles: evaluators, developers, policy experts
  - Standards covered: CC and DO178-C primarily
  - Continuing to recruit subject matter experts as interviewees
- Preliminary results:
  - Confirmed certain issues wrt current practices
    - Especially:
      - Gaps, lack of tools/automation, challenging reliability
      - Incentives and coverage
  - Developed a data-driven meta-criteria model (next slide)
- Next steps:
  - Additional interviews. Focus on specific meta-criteria.
  - Refine meta-criteria

*Preliminary notes – not published*


School of Computer Science

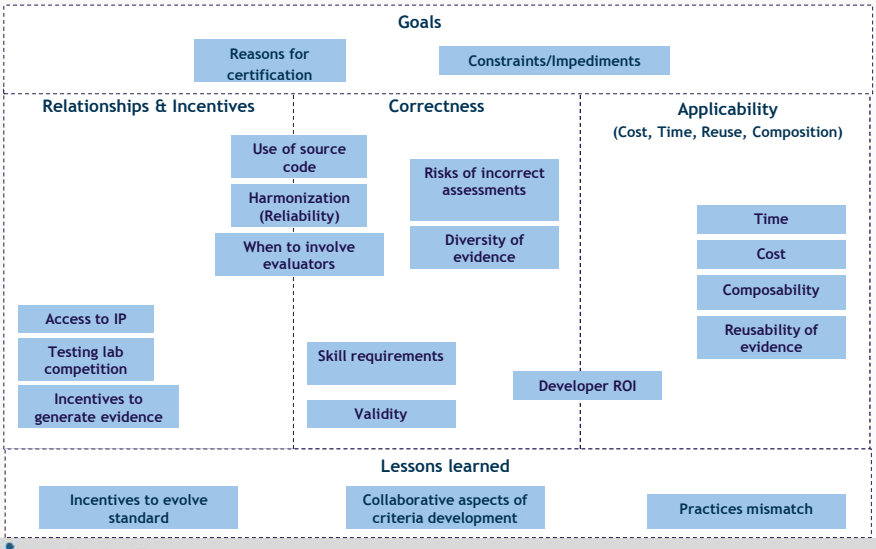
## Interviews: Snapshots (examples)

- Certification as mechanism to deflect risks, not provide assurance
  - "Deflect" = delegate/transfer responsibility for outcomes
  - Emphasis on process compliance (vs. direct evaluation)
- Role of human users often discounted
  - Decreased value of certification when human factors are ignored (misuse of evaluated products)
- Evaluation as pro forma acceptance gate
  - Little or no perceived improvement in quality of evaluated products resulting from evaluation
- Code not a focus for CC, RMF
  - Evaluators may actually prefer to avoid seeing source code
- No clear message regarding composition or evolution
  - Interviewees: "A hard problem – beyond our focus today."
- Recertification glossed over
  - Reuse of evidence only if same evaluator is re-hired

*Preliminary notes – not published*



## Interviews: Reconciling with meta-criteria




institute for SOFTWARE RESEARCH School of Computer Science Carnegie Mellon

## Interviews: Meta-criteria coverage (examples)

- Diversity and reusability of evidence
  - Evaluations heavily rely on text-based evidence
    - In general, it's basically text documents, huge piles of PDF usually. Most of it was text.
  - Re-evaluations and recertification
    - Everyone was trying to reuse as much as possible
- Composition
  - No evaluation of open source libraries/frameworks
    - There was a heck of a lot of open source software. And that, as far as I could tell, was mainly decided by for more technical reasons. This was the standard or the well-regarded software package that did X, that there wasn't a strict evaluation done about how reliable X was/is for security
  - Composition of functionality (coarse-grained)
    - So that's one instance of composition but it's composition of functionality really, not a collection of products, right.

*Preliminary notes – not published*



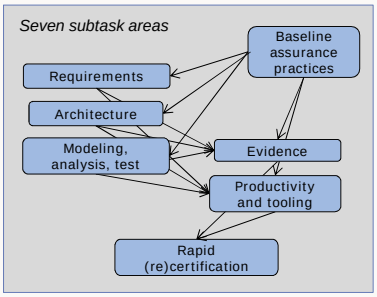
## Interviews: Meta-criteria coverage (examples)

- Access to IP (diverse responses)
  - Regarding sharing code with DO-178C certification authorities
    - Then you have to show it. And you cannot say, "Okay, it's intellectual property. I cannot show you." You will not get the certification for the airplane.
  - Openness (Common Criteria) – a range
    - And if we write our procedures that says okay, evaluator, the first thing you do is you get the source code then you do X. We've had vendors say that's a nonstarter. If you write assurance activities or if you have us do stuff where it requires the labs to have our source code, we're not going to play with you.
    - So it may be that a company is very comfortable with a US lab saying here, here's a source code do this analysis but that same company – if they got their evaluation done in Russia or China – would not want to disclose their source code to those entities.
    - We were pretty open about the artifacts that we're sharing with the evaluators.

*Preliminary notes – not published*

## Agenda

- Focus and themes
- Team
- Challenges for assurance
- Task conops – four threads
  1. Practices baselining
  2. Meta-criteria identification
  3. Models, analyses, validation
  4. Improvements to practice
- Summary of status
  - Synergetic activities
- Meta-criteria
- Practices baselining
  - Interviews
- Models, analyses, validation
  - 1. architecture
  - 2. requirements
  - 3. test and repair
  - 4. modular analysis
  - 5. configurations






## Models, analyses, validation 1 – architecture

- Faculty
  - David Garlan, Bradley Schmerl, and Javier Cámara.
- Students
  - Vishal Dwivedi, Thomas James Glazier, Ivan Ruchkin, Roykrong Sukkerd
- Areas of focus
  - Assurances for Autonomous Systems: How to assure that autonomous systems work correctly in the environments where they are deployed.
    - Includes how to reason about human involvement with autonomous systems
  - Architecture Recovery from Legacy Problems: How to create evidence that links architecture design/rationale with code.
    - New thrust, building on DARPA BRASS related work (legacy architecture recovery)
    - Will integrate with Aldrich BRASS project (adapting in response to resourcing changes
  - Multi-model: How to relate the many models used to reason about social-cyber-physical systems in systems engineering
    - Creation of evidence and dependency/consistency links among multiple kinds of models
  - Code: Relating architectural commitments with code-level decisions

## Models, analyses, validation 2 – requirements

- Faculty
  - Travis Breaux
- Students
  - Daniel Smullen, Hanan Hibshi, Jaspreet Bhatia
- Areas of focus
  - Understanding, reconciling, and combining evaluator verbiage
    - Systematic technique to extract quantitative evaluation judgments
  - Empirical evaluation with more than 200 experts/subjects
- Next
  - Focusing on requirements expression and weighting of attributes



## Models, analyses, validation 2 – requirements

- Early results
  - Express architectural cross-component data flows
    - Detect conflicts due to flow restrictions
    - Enable flow tracing
  - Assess impact on performance of policies
- Publications – a sample
  - Towards Rapid Re-Certification Using Formal Analysis. To Appear: 12th Annual Acquisition Research Symposium, May 2015
  - Managing Security Requirements Patterns Using Feature Diagram Hierarchies. In 22nd IEEE International Requirements Engineering Conference, 2014
  - Formal Analysis of Privacy Requirements Specifications for Multi-Tier Applications (Nominated for Best Paper). In 21st IEEE Requirements Engineering Conference, 2013 (prior to Task initiation)

## Models, analyses, validation 3 – test and repair

- Faculty
  - Claire LeGoues
- Students
  - Mauricio Soto Gonzales, Zack Coker (NSF Graduate Fellow)
- Initial areas of focus
  - Accumulation and application of evidence in the form of tests, coverage analytics, models, and analyses
  - Techniques for automatic program repair based on machine learning and “patch metrics”
  - Semantically-based coverage analytics for testing and repair



## Models, analyses, validation 4 – modular analysis

- Faculty
  - Jonathan Aldrich, Josh Sunshine, Christian Kästner
- Students
  - Waqar Ahmad, Darya Kurilova
- Initial areas of focus
  - Modules and composition benefits
  - Develop mechanisms to limit capabilities of individual modules to access critical resources within a system through isolation (sandboxing) and other means
  - Techniques to model, analyze, and monitor interactions and interference among components
  - Examination of diverse module systems to assess tooling and composition approaches going forward

## Models, analyses, validation 4 – modular analysis

- Faculty
  - Jonathan Aldrich, Josh Sunshine, Christian Kästner
- Early results
  - Sandboxing/encapsulation techniques and tools
    - {static assurance; dynamic monitoring; encapsulation}
  - Architecture-level assurance for quality attributes
  - API protocol usability
  - Concurrency libraries and security
  - Directive mechanisms in mobile frameworks
- Publications – selected
  - Preprocessor-Based Variability in Open-Source and Industrial Software Systems: An Empirical Study. Accepted for Empirical Software Engineering (ESE), 2015.
  - Extracting Configuration Knowledge from Build Files with Symbolic Analysis. Accepted at ICSE workshop, 2015
  - Searching the State Space: A Qualitative Study of API Protocol Usability. International Conference on Program Comprehension, 2015

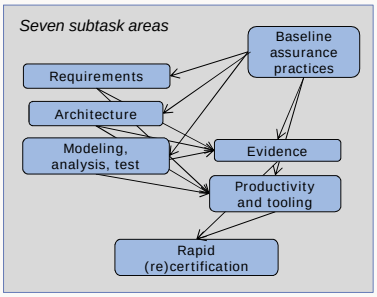


## Models, analyses, validation 5 – configurations

- Complexity of configurations
  - Analysis whether configuration complexity is an indicator for potential vulnerabilities (published: SPLC 2016)
- Management of dependencies within a software ecosystem
  - Reuse and evolution of software packages/supply chains
  - Interviewed developers in node.js, Eclipse, and CRAN
- Inter-app interactions
  - Among Android apps (MSR 2016)
- Dependency management (joint with Jim Herbsleb and Chris Bogart)
  - Software reuse is common/easy with open source and frameworks
  - Rapid change can be common
    - Relying on old versions has security implications
    - node.js/npm: common to use dependencies
      - Malicious change of one package can break ecosystem
  - Issue for configuration integrity in component supply chains
  - Mechanisms to cope with change can cause bad habits
    - Copying code, for example; increasing reliance on automated testing and signaling through version numbers

## Agenda

- Focus and themes
- Team
- Challenges for assurance
- Task conops – four threads
  1. Practices baselining
  2. Meta-criteria identification
  3. Models, analyses, validation
  4. Improvements to practice
- Summary of status
  - Synergetic activities
- Meta-criteria
- Practices baselining
  - Interviews
- Models, analyses, validation
  - 1. architecture
  - 2. requirements
  - 3. test and repair
  - 4. modular analysis
  - 5. configurations






*Achieving assured quality….*

Process enables quality

Engineering delivers quality

Evidence affirms quality

## Review: Incentives and drivers for assurance

- Augmenting process compliance with direct evidence
  - Process enables quality
  - Engineering delivers quality
  - Evidence affirms quality
- Challenges and impediments to evidence-based approaches
  - IP exposure and acceptance evaluation
  - Safe harbors and incentives
  - False trades: performance, cost (lifecycle, devt), quality, security, etc.
- Drivers of evidence-based approaches
  - Acquisition and sustainment
    - Incrementality and evolution
  - Structural realities
    - Dynamic architectures, resiliency, autonomy
    - Frameworks, granular components, rich supply chains, ecosystems
  - Data-intensive modern tooling
    - Modeling and analytic evidence structures
      - Link multiple kinds of models, attributes, ilities, etc.
    - Explicit management of attribute trades



## Looking ahead post-task

- Practices baselining
  - Continue interviews
  - Expand SME access, building on network
- Meta-criteria
  - Application in baselining, leading to refinement and scrub
  - Publication plans
- Technical advances – areas of emphasis
  - Architecture
    - Continue: Autonomy; Legacy; SE / CPS multi-models; Code
  - Requirements
    - Dependency management
    - Safety/hazard/security policy analysis
  - Modular analyses and defects
    - Broaden scope of quality attributes
    - Focusing at component level, composition
- New-generation assurance approaches
  - Build on meta-criteria “sweet spots”
  - Develop initial concepts for evidence-based devt/eval
    - Practices, data, tools, models
    - Incentives and productivity benefits