Assured Design

Dr. John Goodenough

jbg@sei.cmu.edu

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Assured Design (SW-reliant system)

Assured: Having justified confidence that a software-reliant system has particular properties

Assured Design (SW-reliant system)

Assured: Having justified confidence that a software-reliant system has particular properties

- Functional — Actions and outputs in response to inputs
- Run-time — Reliability, security, safety, performance, etc.
- Lifecycle — Modifiability, testability, etc.

Assured Design (SW-reliant system)

Assured: Having justified confidence that a software-reliant system has particular properties

- Functional — Actions and outputs in response to inputs
- Run-time — Reliability, security, safety, performance, etc.
- Lifecycle — Modifiability, testability, etc.

Design: The structure (architecture) of a system — its constituents and their relationships

Assured Design (SW-reliant system)

Assured: Having justified confidence that a software-reliant system has particular properties

- Functional — Actions and outputs in response to inputs
- Run-time — Reliability, security, safety, performance, etc.
- Lifecycle — Modifiability, testability, etc.

Design: The structure (architecture) of a system — its constituents and their relationships

Assured Design: Having justified confidence that a (software-reliant) system design has particular properties

Assured Design (SW-reliant system)

Assured: Having justified confidence that a software-reliant system has particular properties

- Functional — Actions and outputs in response to inputs
- Run-time — Reliability, security, safety, performance, etc.
- Lifecycle — Modifiability, testability, etc.

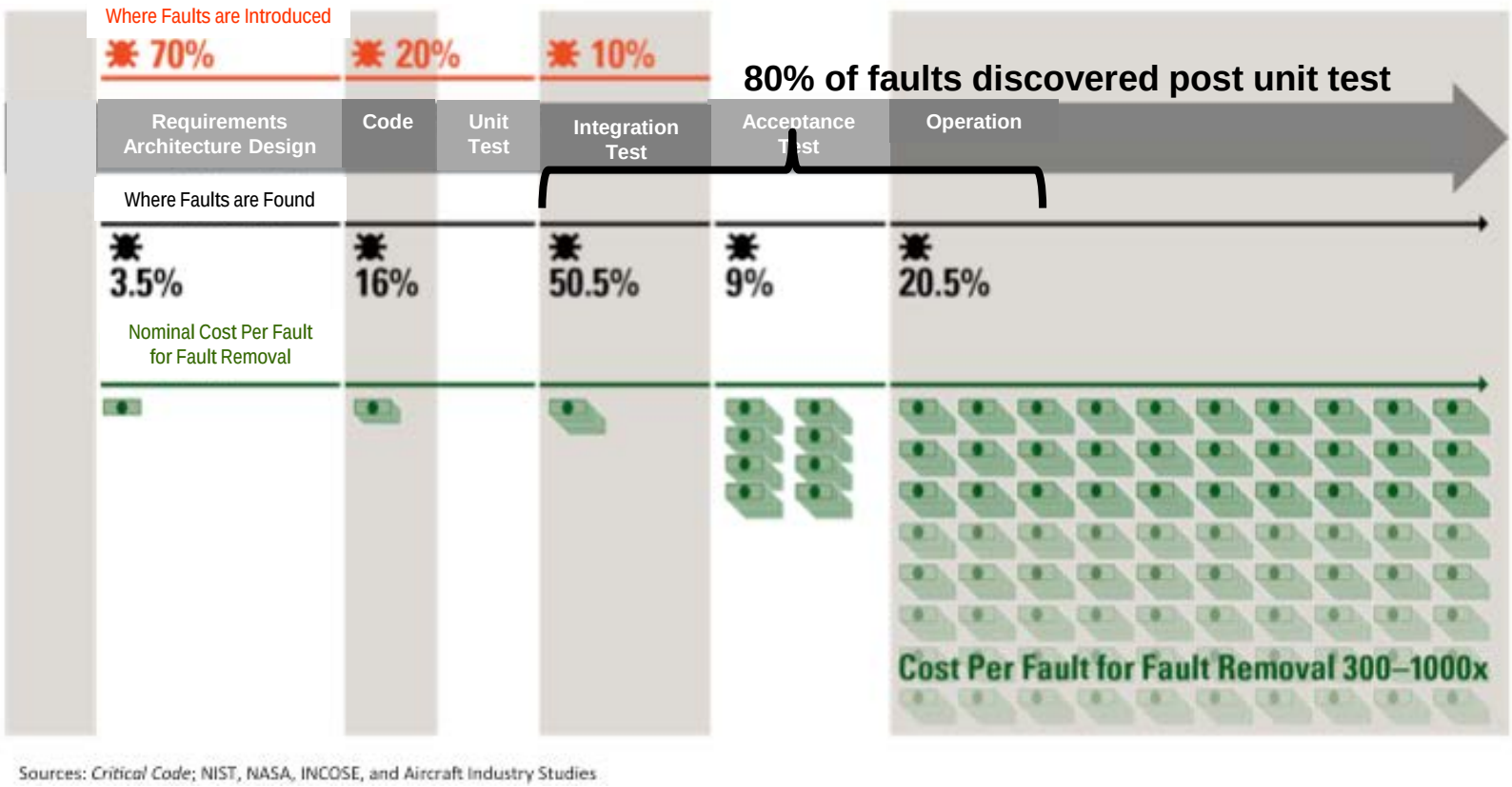
Design: The structure (architecture) of a system — its constituents and their relationships

Assured Design: Having justified confidence that a (software-reliant) system design has particular properties

Design errors are costly and important

Safety-Critical System Challenges

70% of faults introduced in requirements and architecture design
80% of faults discovered post unit test



Sources: *Critical Code*; NIST, NASA, INCOSE, and Aircraft Industry Studies

Total System Dev. Cost
Boeing 777 $12B F-35 $59B

Software as % of total system dev. cost
1997: 45% → 2010: 66% → 2024: 88%

DoD Impact of Poor/Incorrect Design

Time to field and development cost

- Need for rework
- Extended T&E

Degraded sustainability

Reliance-21 C4I COI: Need to field new capabilities faster as threats and technologies change

Better designs are critical

Session Presentations

Design: Modeling and Analysis of Designs

- *Effective Reduction of Avoidable Complexity in Embedded SW*
- *Open Source AADL Workbench*
- *Extending AADL for Security Design Assurance of the Internet of Things*

Implementation: Vulnerability reduction; Exploit new HW

- *Increase Adoption of Secure Coding Standards*
- *Graph Algorithms on Future Architectures*

Session Presentations — Design

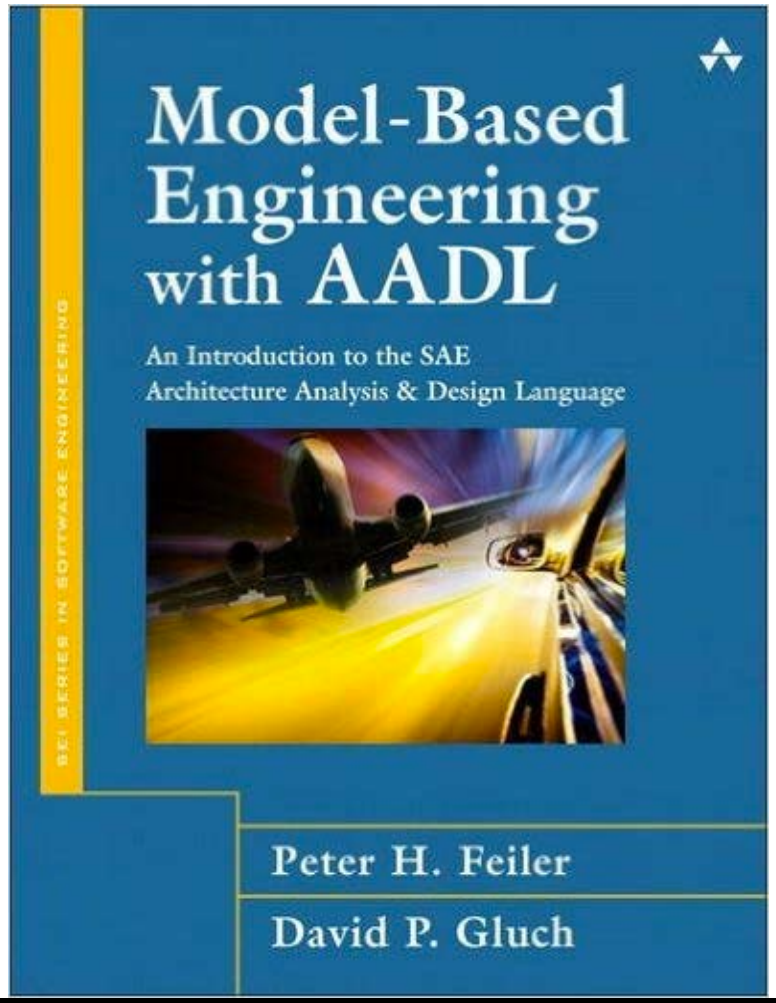
Modeling and Analysis of Designs

Session Presentations — Design

Modeling and Analysis of Designs

AADL (Architecture Analysis and Design Language):

- specifies a static representation of a system architecture
- can model logical flows, binding of software to hardware
- is strongly typed, allowing consistency checks with analysis tools



Session Presentations — Design

Modeling and Analysis of Designs

- *Effective Reduction of Avoidable Complexity in Embedded SW*
  - DoD benefit: Reduced T&E; Less rework → faster deployment; reduced cost
  - SEI edge: Arch. modeling expertise and tools (AADL)

Session Presentations — Design

Modeling and Analysis of Designs

- *Effective Reduction of Avoidable Complexity in Embedded SW*
  - DoD benefit: Reduced T&E; Less rework → faster deployment; reduced cost
  - SEI edge: Arch. modeling expertise and tools (AADL)
- *Open Source AADL Workbench*
  - DoD benefit: Less rework → faster deployment; reduced cost
  - SEI edge: AADL expertise; formal methods; real-time systems theory

Session Presentations — Design

Modeling and Analysis of Designs

- *Effective Reduction of Avoidable Complexity in Embedded SW*
  - DoD benefit: Reduced T&E; Less rework → faster deployment; reduced cost
  - SEI edge: Arch. modeling expertise and tools (AADL)
- *Open Source AADL Workbench*
  - DoD benefit: Less rework → faster deployment; reduced cost
  - SEI edge: AADL expertise; formal methods; real-time systems theory
- *Extending AADL for Security Design Assurance of the Internet of Things*
  - DoD benefit: Fewer operational vulnerabilities
  - SEI edge: Arch. modeling; model checking; SW security

Session Presentations

Design: Modeling and Analysis of Designs

- *Effective Reduction of Avoidable Complexity in Embedded SW*
- *Open Source AADL Workbench*
- *Extending AADL for Security Design Assurance of the Internet of Things*

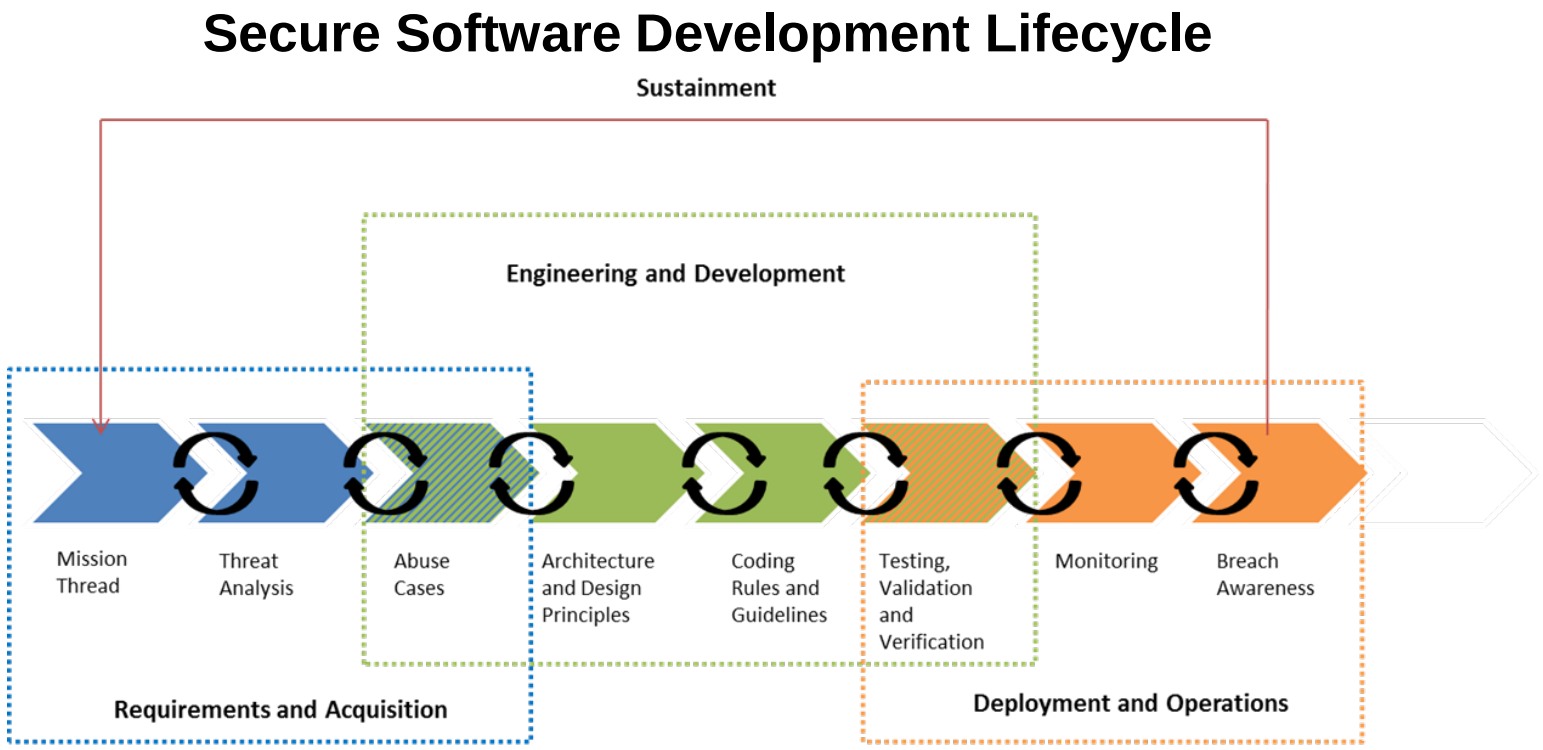
Implementation: Vulnerability reduction; Exploit new HW

- *Increase Adoption of Secure Coding Standards*
- *Graph Algorithms on Future Architectures*

Session Presentations — Implementation

Prevent Vulnerabilities

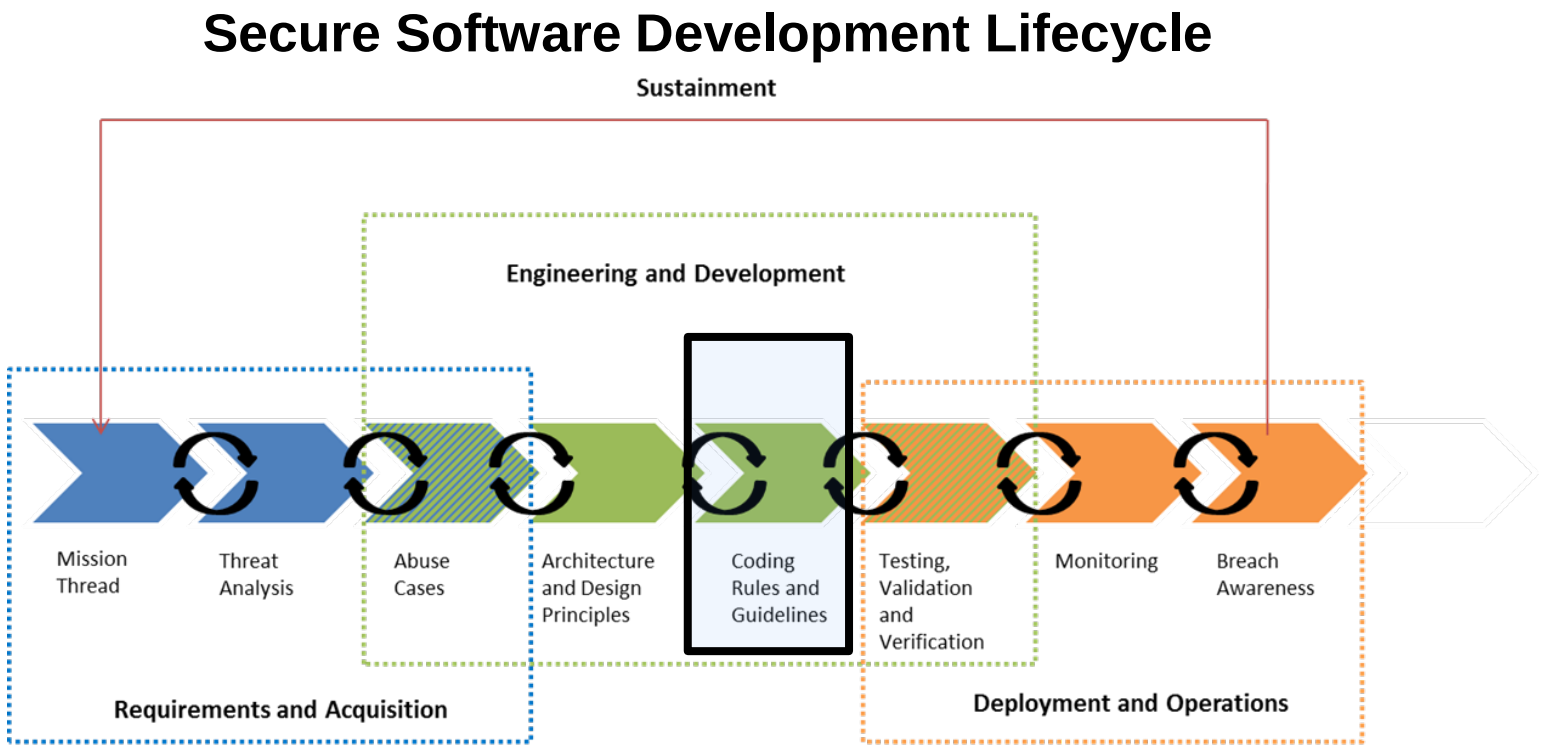
- *Increase Adoption of Secure Coding Standards*



Session Presentations — Implementation

Prevent Vulnerabilities

- *Increase Adoption of Secure Coding Standards*

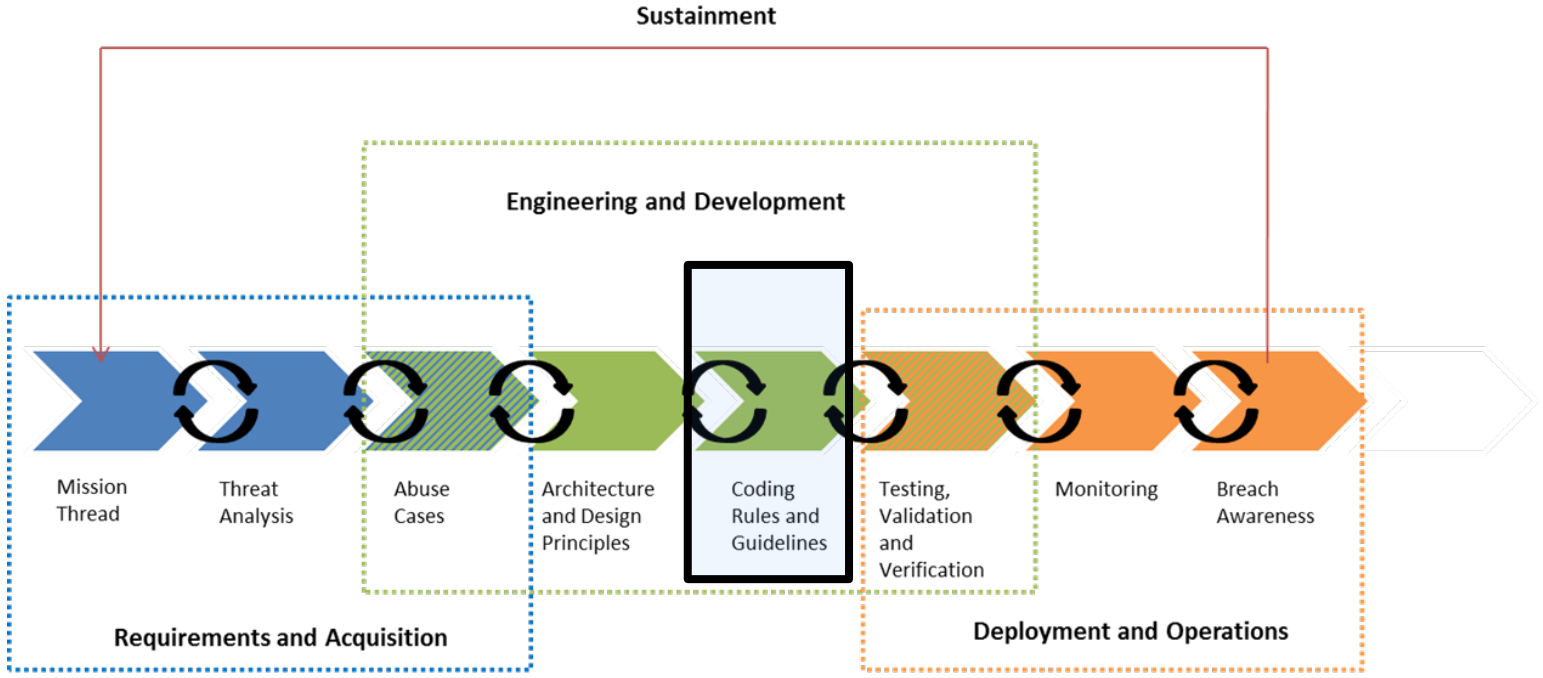


Session Presentations — Implementation

Prevent Vulnerabilities

- *Increase Adoption of Secure Coding Standards*
  - DoD benefit: Fewer operational vulnerabilities
  - SEI edge: CERT's knowledge of vuls and access to code

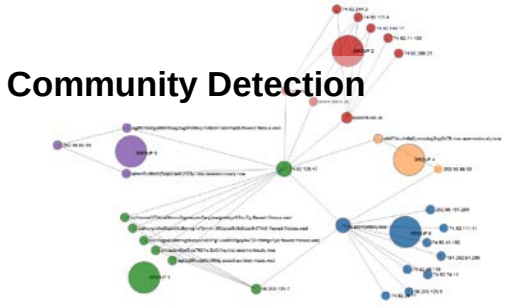
Secure Software Development Lifecycle



Session Presentations — Implementation

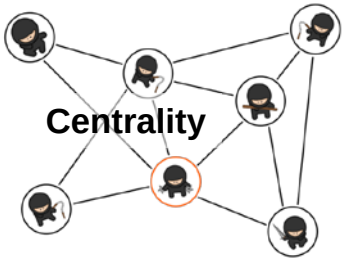
Better Analysis Capabilities

- *Graph Algorithms on Future Architectures*



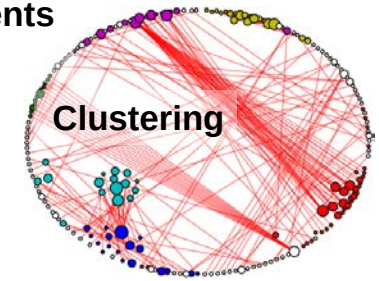
APT Detection in Computer Networks, C3E, 2013

United States Interstate Highway System



Malware Distribution Networks

Social Networks

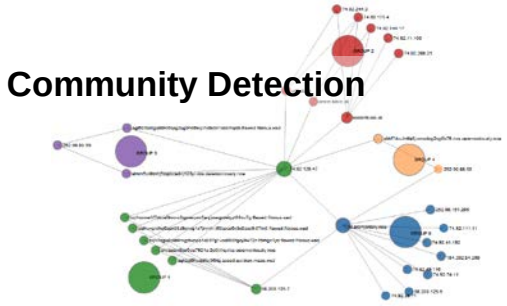


Revert graph showing editor conflict on the "Cyprus dispute" Wikipedia page, 2015

Session Presentations — Implementation

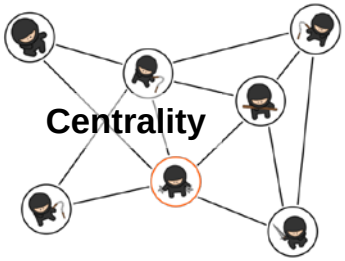
Better Analysis Capabilities

- *Graph Algorithms on Future Architectures*
  - DoD benefit: Faster exploitation of new HW architectures → improved security analysis capability
  - SEI edge: Knowledge of graph analysis algorithms and their use



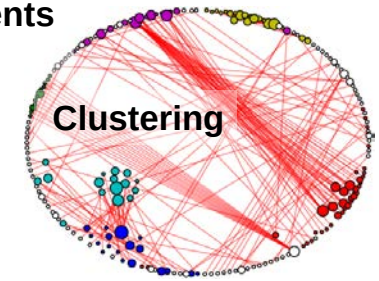
APT Detection in Computer Networks, C3E, 2013

United States Interstate Highway System



Malware Distribution Networks

Social Networks



Revert graph showing editor conflict on the "Cyprus dispute" Wikipedia page, 2015

Summary

Assured Design: Advancing the state of the art and practice

- Modeling and analysis tools (AADL)
- Application in real-world contexts (JMR; SAVI)

Coding Rules: Reducing vulnerabilities in deployed systems

Modifiability: Ensuring graph analysis algorithms can readily exploit new HW architectures

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0002882