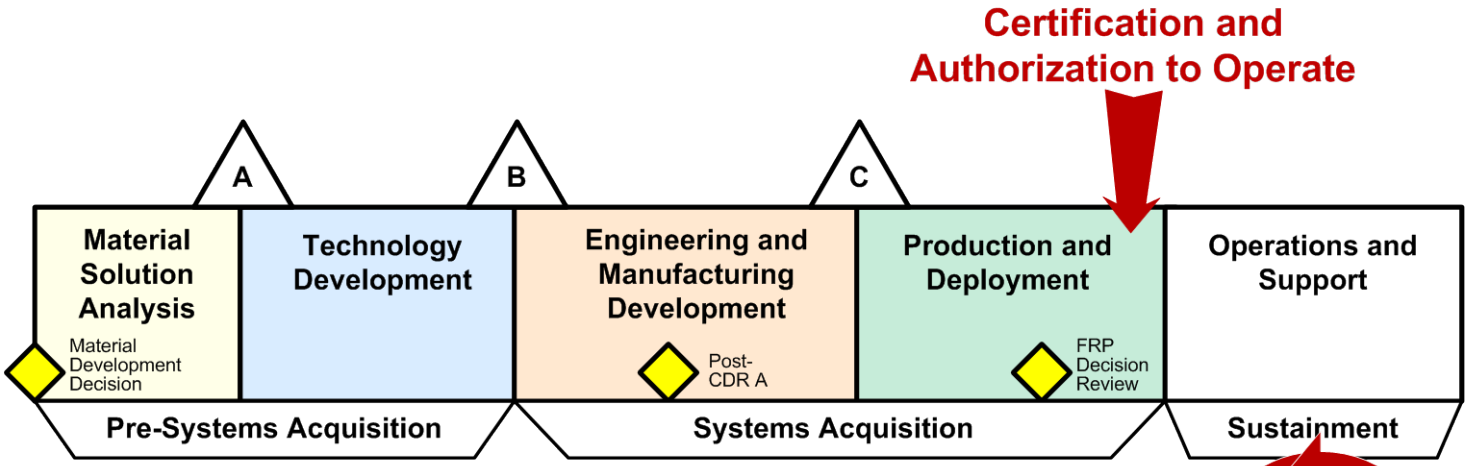
Mission Thread Analysis

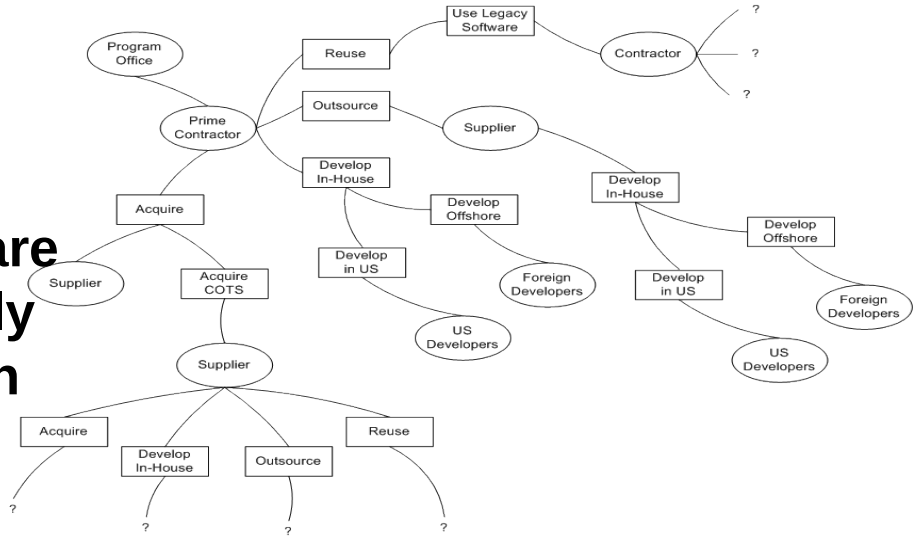
Carol Woody
Software Engineering Institute
cwoody@cert.org

Why Care About Mission Threads?

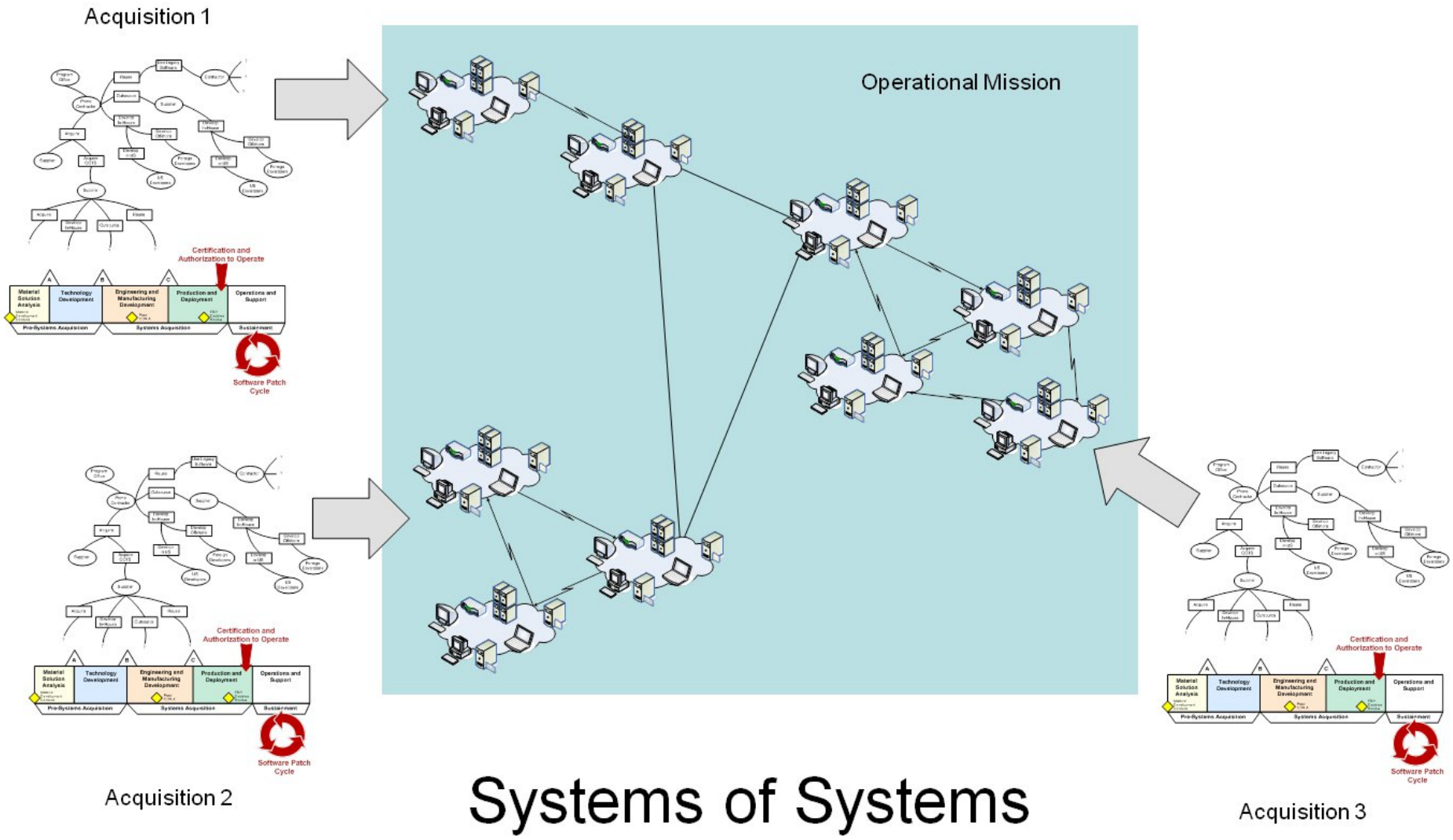
System Life Cycle



Software Supply Chain



Operational Reality



Software Needs to be Trusted

Exploitation of software defects is estimated to cost the U.S. economy $60 Billion annually

What should we be doing to improve this situation?

- Simple Answer: Remove the software defects

Is this feasible?

Role of Software in Systems

From the *NRC Critical Code Report* *

"Software has become essential to all aspects of military system capabilities and operations" p.19

- 1960 – 8% of the F-4 aircraft functionality
- 1982 – 45% of the F16 aircraft functionality
- 2000 – 80% of the F-22 aircraft functionality

* Committee for Advancing Software-Intensive Systems Producibility; National Research Council (NRC). *Critical Code: Software Producibility for Defense* 2010

Why is this change important?

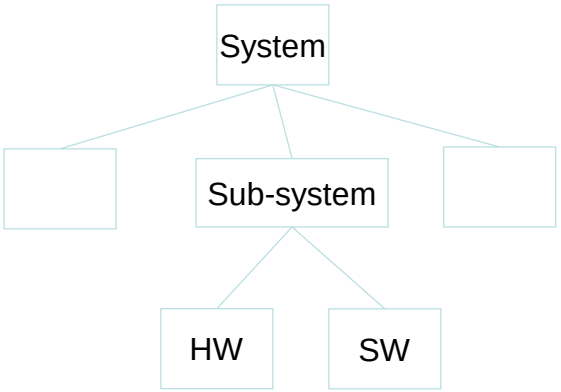
Systems Engineering is Insufficient for Software-reliant Security

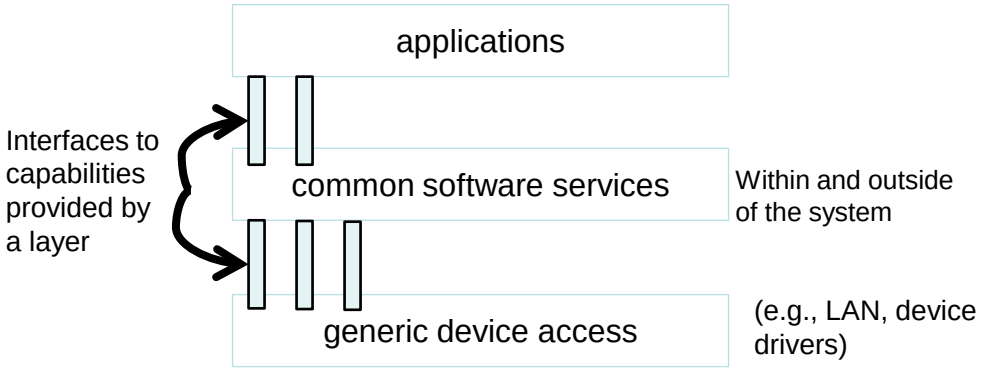
Systems Engineering Assumptions

- Systems can be decomposed into discrete, independent, and hierarchically-related components (or subsystems)
- Components can be constructed and integrated with minimal effort based on the original decomposition
- Quality properties can be allocated to specific components

Software Engineering Realities

- Software components are often related sets of layered functionality (one layer is *not* contained inside another layer)
- Interactions of the components (*not* the decomposition) must be managed
- Security properties relate to composite interactions (*not* to individual components)





System engineering cannot ignore software realities

Mission Thread Analysis

Connecting the software and systems to the operational mission

- Will the implementation work?
- How is security defined and validated?

Establishing the role of mission success (functioning as intended) for system and software assurance

Evaluating mission impact in the event of major technology changes

Agenda

What does mission failure look like?

- Example: 2003 Power Grid failure

Overview of Mission Thread Analysis

Examples using Mission Thread Analysis

Experience to-date

What does mission failure look like?

Power Grid Example

Complex Failure: 2003 Power Blackout [1]

On August 14, 2003, approximately 50 million electricity consumers in Canada and the northeastern U.S. were subject to a cascading blackout. The multiple failures occurred over a four hour period

- The blackout initiated when three high-voltage lines went out of service due to trees too close to the lines
- Race condition disabled alarm system that provided the only effective means for grid operators to identify problems.
  - Hot backup failed: corruption of the data stream caused the backup server to fail - hardware redundancy only (data errors fixed manually)
  - Alarm system restart required full control system reboot (> 30 min)
- IT did not notify grid operations of the alarm system failure and decision not to restart right away
  - Operators assumed all was well – no alarms
  - Operators did not notice power failure – automatic power backup

Complex Failure: 03 Power Blackout -2

Independent power grid monitoring failed

System monitoring applications take current state and project future system state. Warning raised when projected state differs significantly from realized state.

- Data error resulted from the downed lines – IT had to correct errors manually.
- After fixing the data (60 minutes), the IT person forgot to restart the monitor before leaving for lunch.
- Restart failed after lunch as data out of date.
- Monitor finally restarted about 30 minutes before final failure (insufficient time to analyze discrepancies and respond)

Observations – Power Grid

Technical failures required too lengthy a recovery time

Hardware redundancy will not accommodate software failures

Poorly recognized mission dependencies

- Lack of operational ability to identify early warning signs of failure
- Independent monitoring not operationally robust

Fixing software defects is not enough

Other Examples

AT&T Phone Outage – 1957

Lee, Leonard. 1960. *The Day the Phone Stopped - How People Get Hurt When Computers Go Wrong.* New York City: Donald I Fine, Inc., 1960. 1-55611-286-6.

Amazon Cloud recover outage – 2011

ReplyManager and Amazon Web Services Outage Report. *Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region.* [Online] April 21, 2011. [Cited: June 28, 2011.] http://aws.amazon.com/message/65648

Challenges for Software Assurance

Increasingly failures result from a group of errors that are addressed individually but not always considered collectively (operator, unexpected software state, and user)

Mitigating component failure is not sufficient

- Increasing dependencies among development, deployment, and operations – technology focused solutions are incomplete and may make things worse
- Current decomposition techniques to address complexity hide mission risks until deployment
- Increased interaction among systems is needed to support a mission

Decisions at the all levels must support the mission

Mission Thread Analysis for Assurance

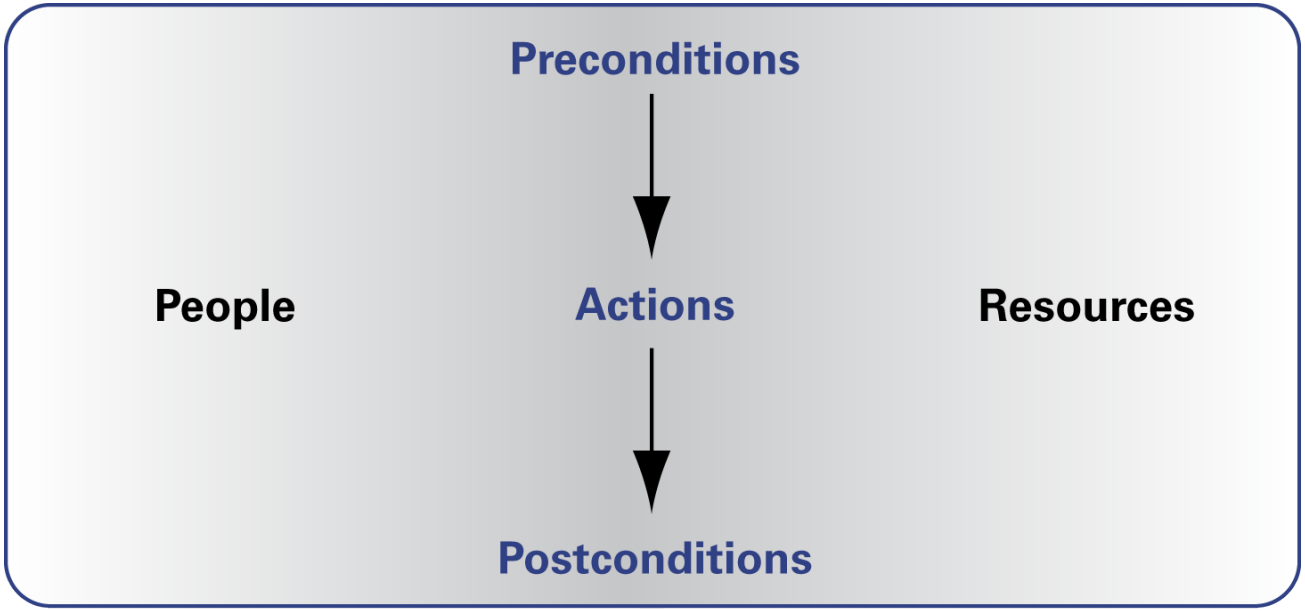
Mission Thread Analysis for Assurance (a.k.a. Survivability Analysis Framework)

Focus on successful completion of a mission (satisfactory execution of each critical step)

Analysis Framework Process:

- Identify a critical mission thread (specific examples)
- Define goals (successful completion criteria) for mission processes
- Describe critical steps required to complete the mission process (end to end) - sequenced activities, participants, and resources
- Define ways that execution can be compromised at each critical step and overall

Mission Step Analysis



Stresses → Acceptable Outcomes

Analysis

- *Potential failure conditions*
- *Likelihood of error conditions*
- *Impact of occurrences*
- *Recovery strategies*

Uses for Mission Thread Analysis

Ensure the operational robustness of a deployed system within the context of a mission thread

Validate that a deployed system meets security and reliability requirements of the mission thread

Balance enterprise and multi-enterprise needs against component and system-specific needs

Construct a shared view of a system and its role in a mission thread for communication among system stakeholders, management, developers, support staff, and users

Applying Mission Thread Analysis Example

Example 1: Doctor Orders Blood Test

A doctor uses blood tests to monitor a treatment plan

- A patient is brought to emergency room with chest pains.
- The doctor suspects a heart-attack and builds a plan of treatment to include initial tests, aggressive treatment, and subsequent tests to verify progress of the treatment.
- The order process with the lab is fully automated.

Example 1: Scenario Steps

A. Patient brought to emergency room with chest pains.

B. Dr. Emergency reviews available records

C. Dr. Emergency develops a treatment plan.

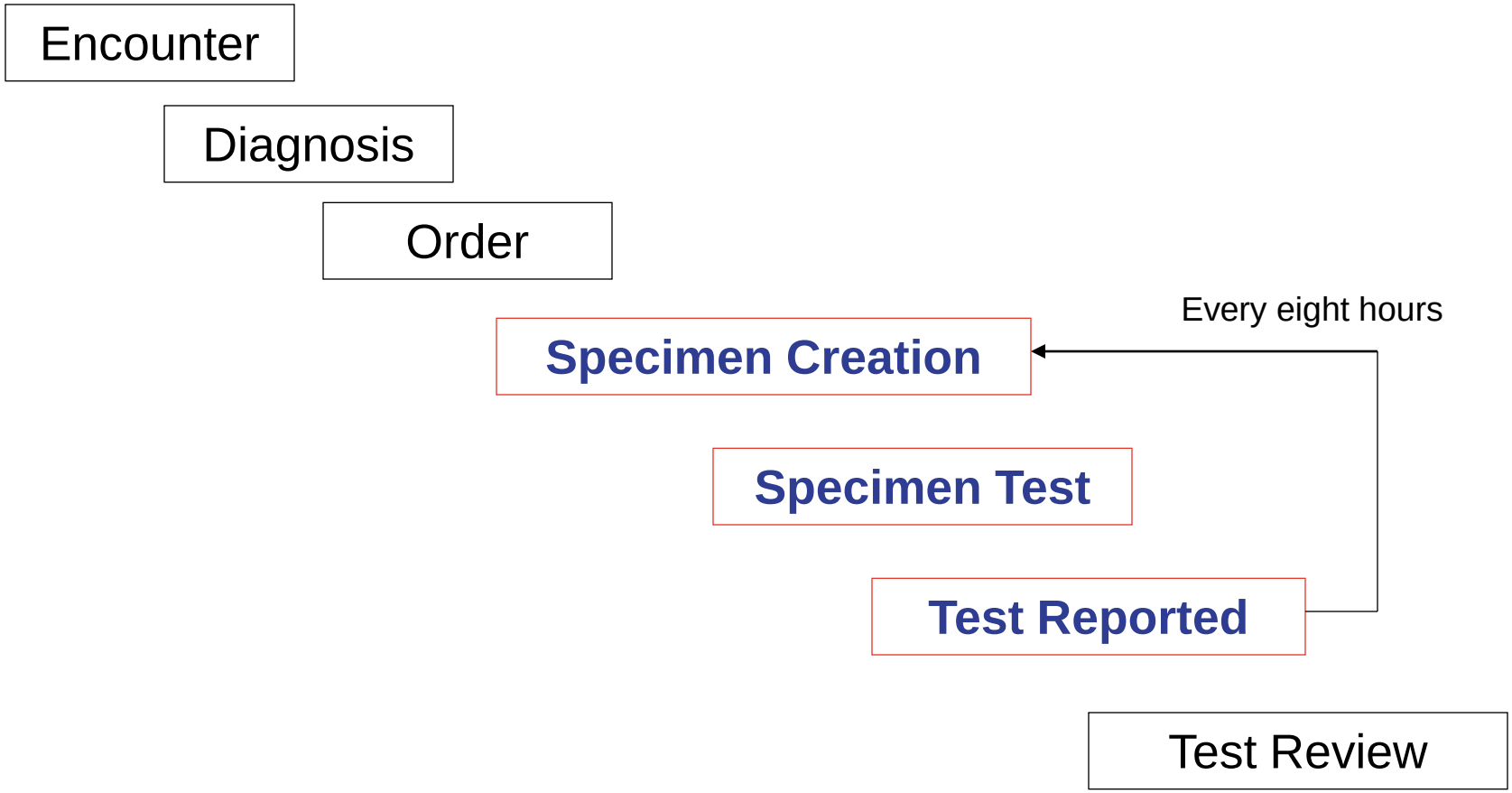
D. Dr. Emergency orders series of blood tests.

E. Phlebotomist from laboratory arrives to collect first specimen which is sent to the lab.

F. Lab receives specimen, performs centrifuge and delivers serum to appropriate testing stations.

G. The Lab system notifies the ordering system that the specimen has been received for testing.

Example 1: Structured View of the Data



Example 1: About the Scenario

Evaluating the success of the treatment plan depends on linking the blood tests and treatment steps.

The business process will be successful if the doctor receives complete lab reports in a timely manner as ordered

The business process steps will be the same if the participants and resources are in the same room, spread across a hospital campus or scattered to multiple companies.

What it takes to establish and maintain security, reliability, and survivability will vary drastically depending on the context.

Example 1: Operational Context

Doctor is using a handheld device to review patient records, to record information during the exam, to order tests, and to receive test results.

The Lab is a separately run business that has contracted to provide services to all of the local hospitals.

For privacy purposes, the Lab does not have patient-specific information. The Lab bills the hospital, and the hospital bills the patient.

Example 1: Describing a Critical Step

Step D	**Doctor orders blood tests**
Precondition	Treatment plan defined Required lab tests identified Handheld connectivity to Lab Dr. E. authorization to order tests for this patient
Action	Enter order on handheld Order accepted, verified, and accepted by Lab
Post-Condition	Test confirmed and scheduled

Example 1: Failure Outcomes for Step D

Missing (or delayed) results:

- some or all tests are not done

Wrong results:

- some unrequested tests were performed
- results do not reflect the actual sample

Disclosure:

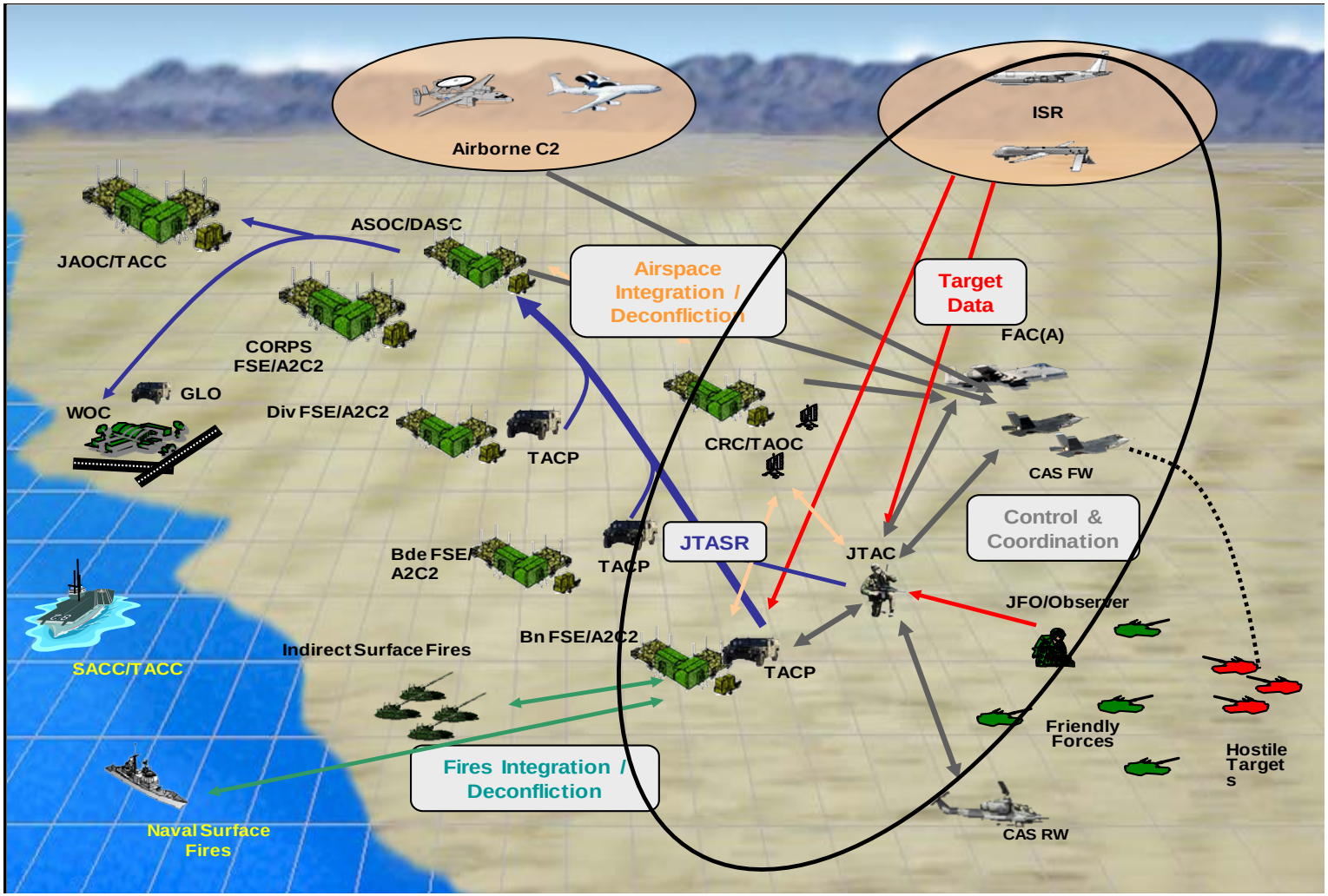
- results are disclosed to unauthorized person
- test results are not associated with the correct patient
- test results are not associated with the correct doctor

Example 1: Causes of Failure

Failure: Missing test results

- Paperwork requiring tests to be run was lost or misplaced
- Blood samples were lost, contaminated, or misplaced
- Some tests were not run by the technician
- Wrong tests were run by the technician
- Some or all test results were not associated with the correct patient by the lab
- Some or all test results were not associated with the right doctor by the lab
- Testing machine did not produce results
- Testing machine was not working and could not produce results

Example 2: Operational View (OV)-1



Derived from Business Case Analysis, Close Air Support Capability Solutions FY10-FY15, Prepared by FCA/ECSA, 6 November 2007

Example 2: Concept of Operations for Time Sensitive Targeting

1. Find
2. Fix
3. Track
4. Target
5. Engage
6. Assess

Example 2: Military Joint Forces Mission Thread

An army unit on patrol spots a missile launcher preparing to fire. The unit calls their commander and provides a description of the launcher and its location. Even though the launcher is in the Army's area of responsibility, in this scenario the Army does not have an appropriate weapon to bring to bear (for example, the artillery could be in use on other targets). However, the Air Force has a suitable platform and is tasked as Executive Agent to further prosecute or strike of the target. The Army remains the authority for the strike even though the Air Force will perform the engagement.

Example 2: TST Scenario Steps -1

A. Army unit sees “something” (e.g. Missile launcher preparing to fire.)
B. Unit calls their command post and provides a description of the launcher and its location. Echelon Units are provided with JFC intent and guidance on TST.
C. TST cell becomes involved.
D. Sharing, interaction, and preliminary planning with other Intel Points
E. Joint TST Cell makes a decision and adds to joint data system . Other component commanders provides input.
F. Each TST Cell does pairings based their rules of engagement and C2 systems. All components do pairing based on assets available. Joint Air Coordination center usually determines best match as delegated from JFC.
G. Select Weapon(s) and delivery Platform (F16 is selected platform) based on timing, collateral damage, desired effect, etc.

Example 2: TST Scenario Steps -2

H. Joint Force Commander decision (approval authority); Only inserts himself if TST is outside normal bounds

I. OPS Staff/Commander in Air Operations Center cuts order; Committing aircraft; sending verbal to pilot usually through command and control unit

J. Order transmitted to pilot usually verbally

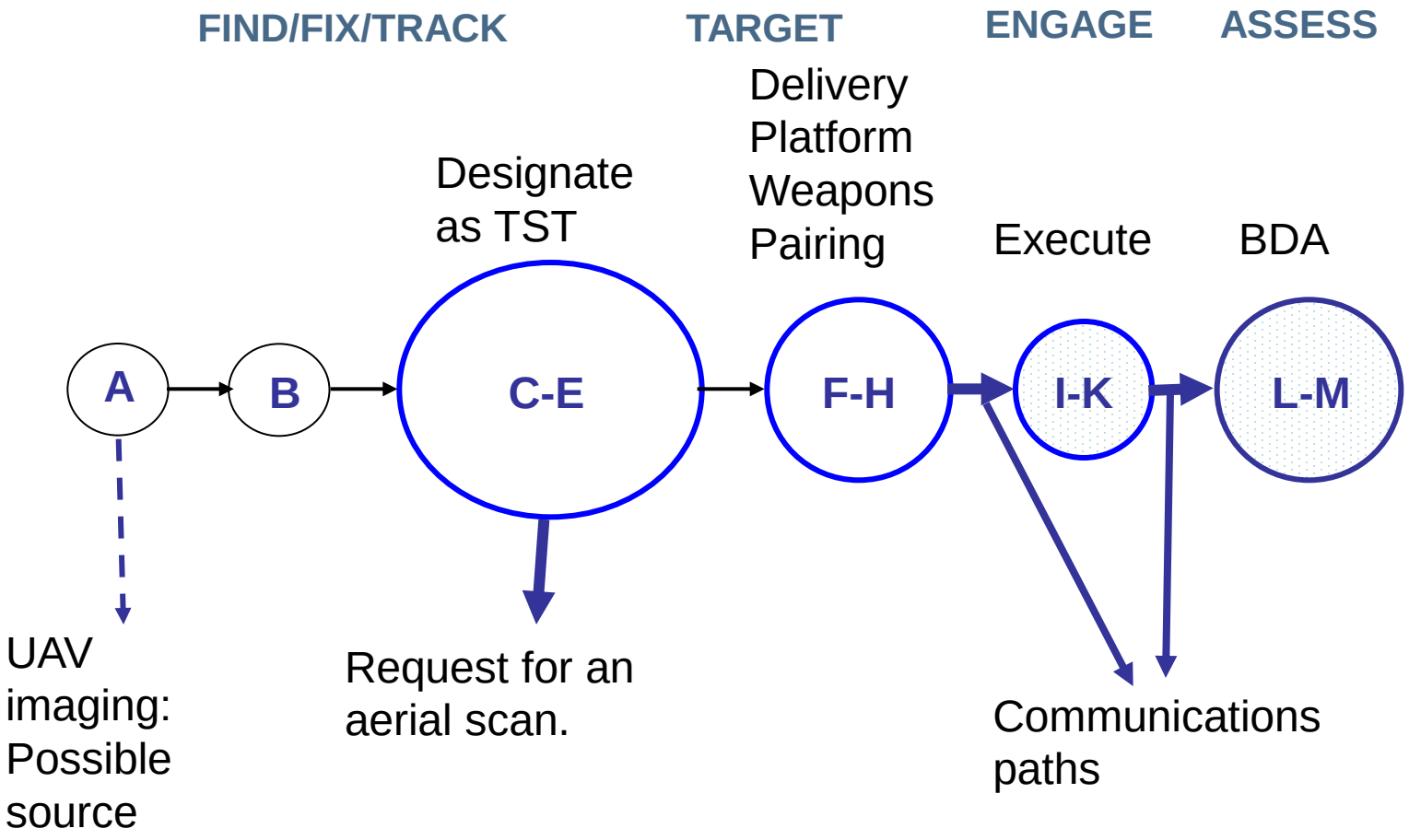
K. Pilot executes

L. Multiple assessments lead to unified Battle Damage Assessment; collection requirements are normally orchestrated before target is attacked. Constant tradeoff issues arise between low-density/high demand assets.

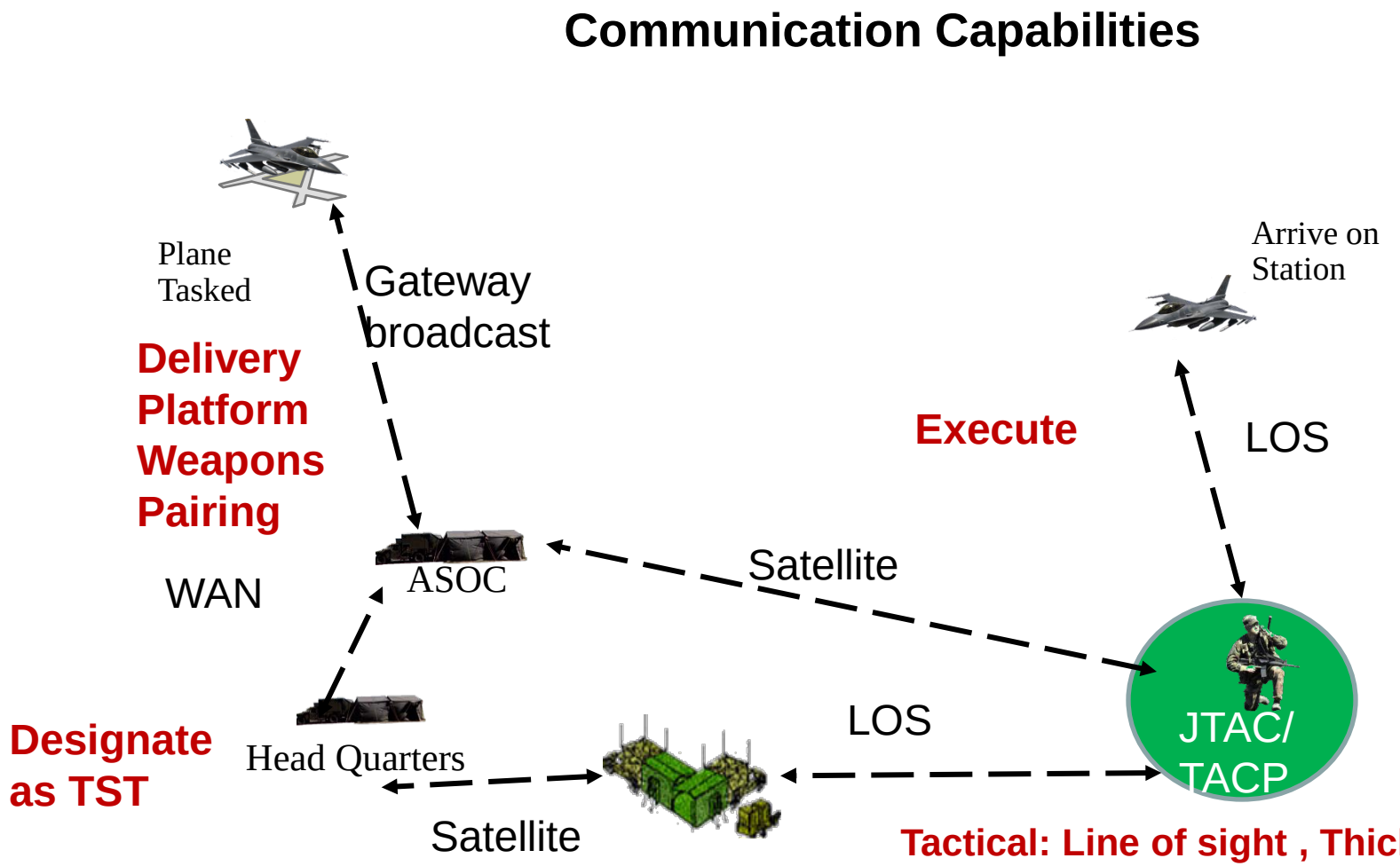
M. Assessment determines if re-strike Is required. If unsuccessful, target is rolled back into queue.

Example 2: Mission Thread Diagram

End-to-end mission analysis applied to a sequence of steps



Example 2: TST Operational Context -1



Tactical: Line of sight , Thick clients, capable of operating with high performance networks, with ad-hoc, peer-to-peer, or with no network.

Mission Thread Analysis - Lessons Learned to Date

Building Mission Threads

Official documents are usually too general for mission analysis

- Idealized – what should happen as opposed to what does (a primary basis for requirements)
- Stressful operating conditions not considered

One project built the mission thread documents from the detail design with no relationship to operational reality

Explore Security Related Issues

Consider mission failure scenarios

- Who identifies and manages a reported error?
  - Coordination of responses across multiple systems – how do you do concurrent end-to-end management (multiple contractors)
- Which faults should be reported to user software?
  - Fault reporting can easily overload resources (esp. mobile devices)
  - Will the receiver understand an error and know what to do?
- How could an attacker go undiscovered in the “cracks” between systems (e.g. UAV malicious code)

Support for an Assurance Case

Assurance case requires context of mission and use

- Defines successful performance – normal conditions
- Establishes range of threats/vulnerabilities to be considered

Mission and use must be structured for assurance analysis

- Dependencies
- Failure outcomes
- Potential causes of failures

Reference: Ellison, Goodenough, Weinstock, Woody, Survivability Assurance for System of Systems, CMU/SEI-2008-TR-008,May 2008, www.sei.cmu.edu/reports/08tr008.pdf

Summary

Summary

Mission thread analysis

- provides visibility for operational completion of actions across systems and components that are independently designed and developed to optimize local needs
- supports failure analysis and mission impact of interacting systems and components
- addresses gaps in system requirements with the analysis and evaluation of failure potential and mission assurance
- builds the case for justified confidence of the delivered system

Resources

Survivability Analysis Framework, Robert Ellison and Carol Woody. http://www.sei.cmu.edu/library/abstracts/reports/10tn013.cfm

Survivability Assurance for System of Systems, Robert J. Ellison, John Goodenough, Charles Weinstock, & Carol Woody, CMU/SEI-2008-TR-008 May 2008 www.sei.cmu.edu/reports/08tr008.pdf

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN “AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.