Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

“The Software Engineering Institute (SEI) of Carnegie Mellon University-Programs, Initiatives and Responses to U.S. DoD Great Challenges”

Singapore Management University, July 22, 2004

Angel Jordan
University Professor Emeritus
Provost Emeritus, Carnegie Mellon University
Acting Director, CEO
Software Engineering Institute
ajordan@sei.cmu.edu
www.sei.cmu.edu
412-268-7740

Sponsored by the U.S. Department of Defense

Outline of the Presentation

The Software Engineering Institute and its Mission
State of Practice Versus SEI's Vision
SEI Technical Programs
Product Lines systems
Dynamic Software systems
Software Engineering Process Management
Acquisition Support Program
The Challenge Problems from DoD
Network Systems Survivability Program
Survivable Network Technology
CERT Analysis Center
Survivable Enterprise Management
Practices, Development and Training
CERT/CC
Network Situational Awareness

Software Engineering Institute

Applied R&D laboratory, Federally Funded R&D Center, at Carnegie Mellon University, Pittsburgh PA

Mission is to provide leadership in software engineering and to transition new software engineering technology

Encouraged to support industry in precompetitive technology R&D and in technology transition activities

Software Engineering Institute

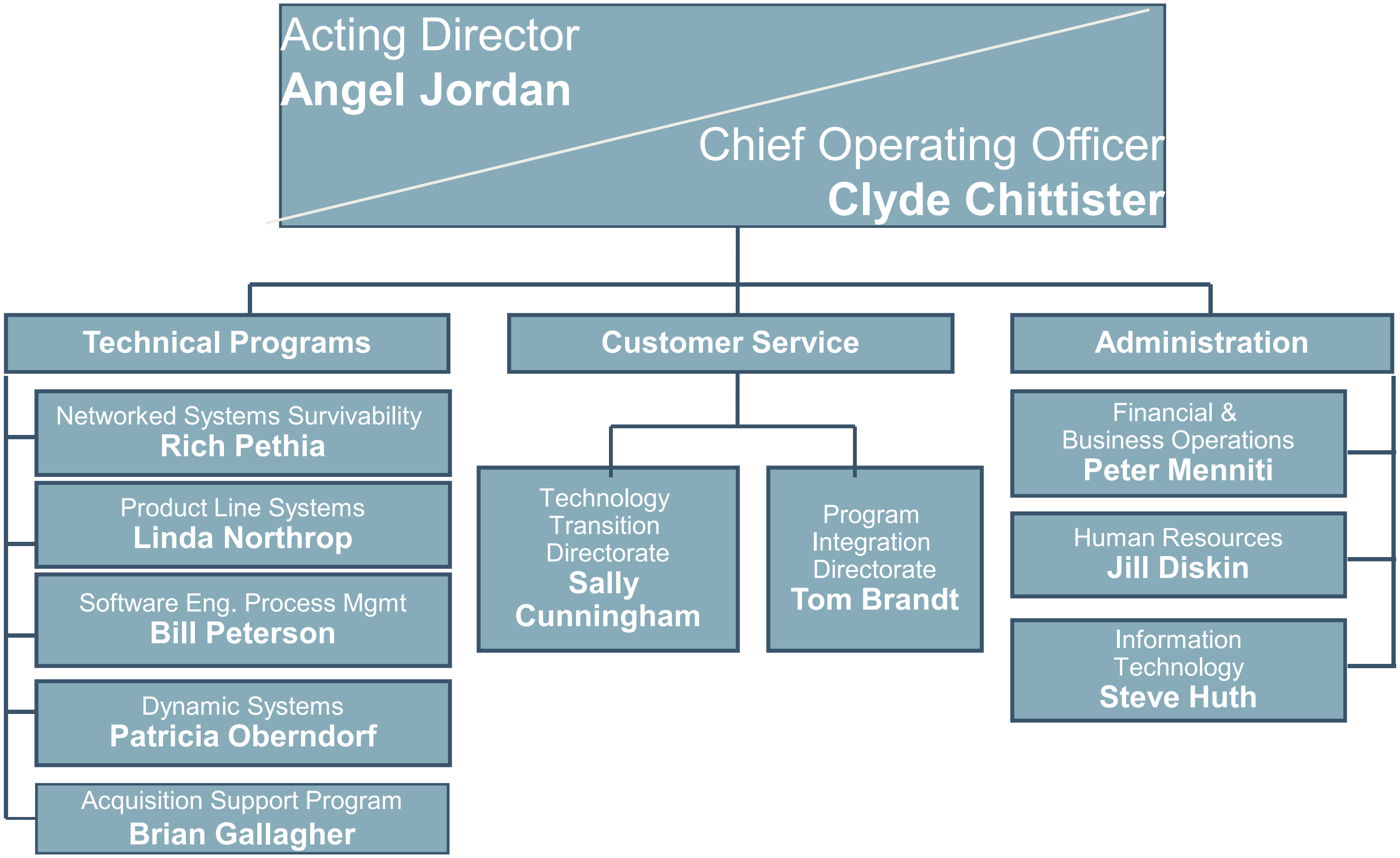
Mission

Provide technical leadership to advance the practice of software engineering so the DoD can acquire and sustain its software-intensive systems with predictable and improved cost, schedule, and quality.

Role as R&D Laboratory FFRDC

1. maintain long-term competency in areas where Government cannot rely on in-house or private-sector capabilities
2. develop and transfer new technology to private sector for Government's benefit

SEI Organization Chart



Acting Director
Angel Jordan

Chief Operating Officer
Clyde Chittister

Technical Programs

Networked Systems Survivability
Rich Pethia

Product Line Systems
Linda Northrop

Software Eng. Process Mgmt
Bill Peterson

Dynamic Systems
Patricia Oberndorf

Acquisition Support Program
Brian Gallagher

Customer Service

Technology Transition Directorate
Sally Cunningham

Program Integration Directorate
Tom Brandt

Administration

Financial & Business Operations
Peter Menniti

Human Resources
Jill Diskin

Information Technology
Steve Huth

SEI Strategic Themes

Predictably better, faster, and cheaper by -

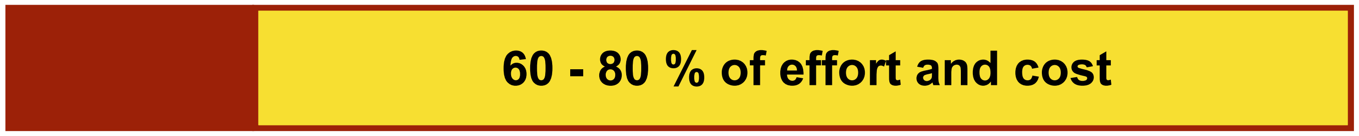
Moving to the left
Embrace a systems engineering approach and make better decisions before coding to predictably improve quality, cost, and schedule.

Reusing everything
Reuse code, but also the architecture and knowledge from building similar systems.

Never making the same mistake twice
Leverage lessons learned.

State of Practice Versus SEI’s Vision

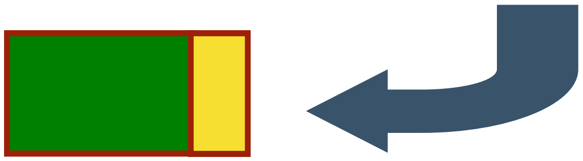
Software state of practice (“test in” quality)



60 - 80 % of effort and cost

Development Integration and System Test

World-class developers “design in” quality



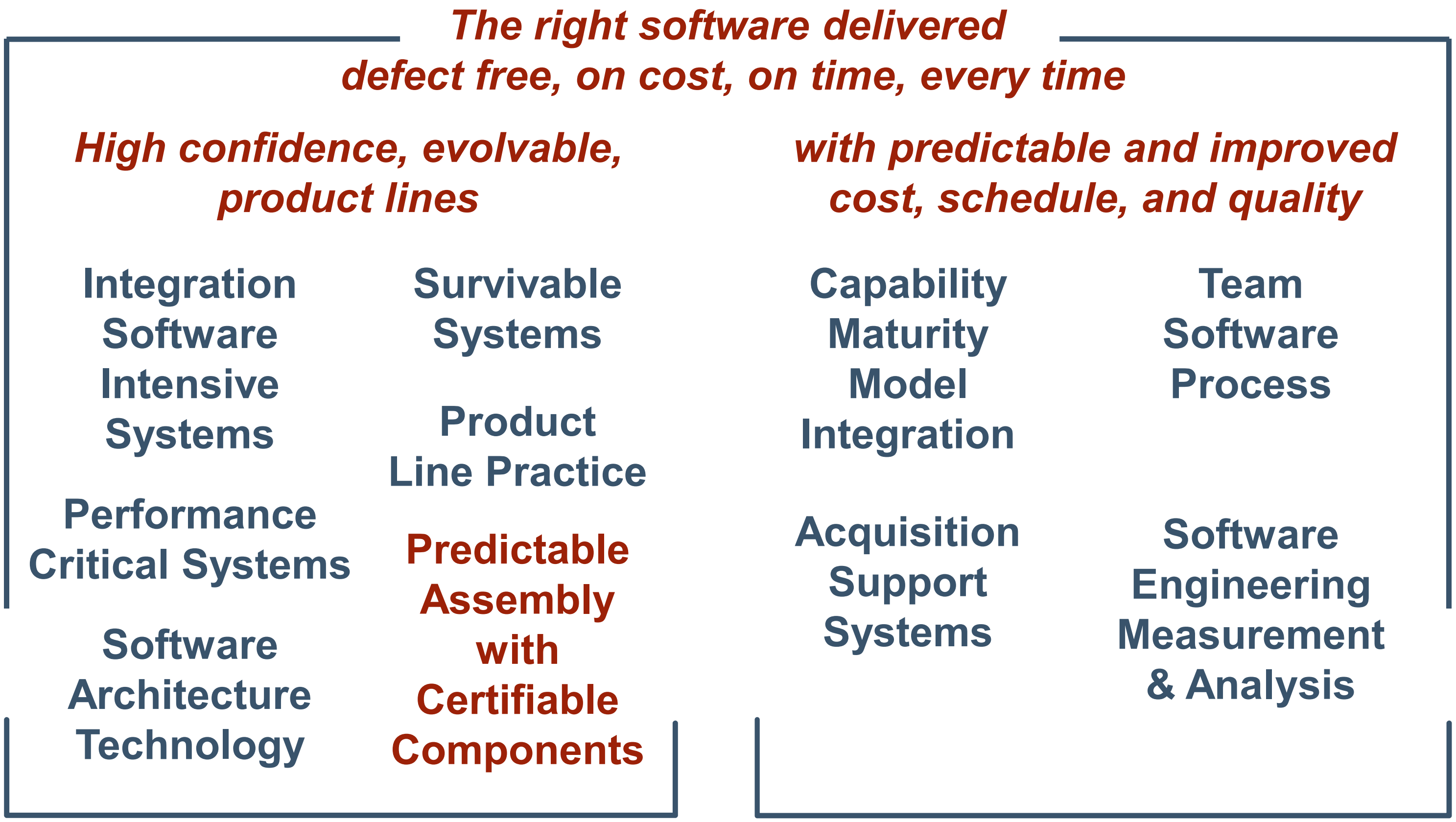
*** move to the left !**

*** reuse <u>everything</u>**

*** never make the same mistake twice**

*Ref: Standish Group, www.standishgroup.com, 1999

SEI Technical Program



The right software delivered
defect free, on cost, on time, every time

High confidence, evolvable, product lines

Integration Software Intensive Systems

Performance Critical Systems

Software Architecture Technology

Survivable Systems

Product Line Practice

Predictable Assembly with Certifiable Components

with predictable and improved cost, schedule, and quality

Capability Maturity Model Integration

Acquisition Support Systems

Team Software Process

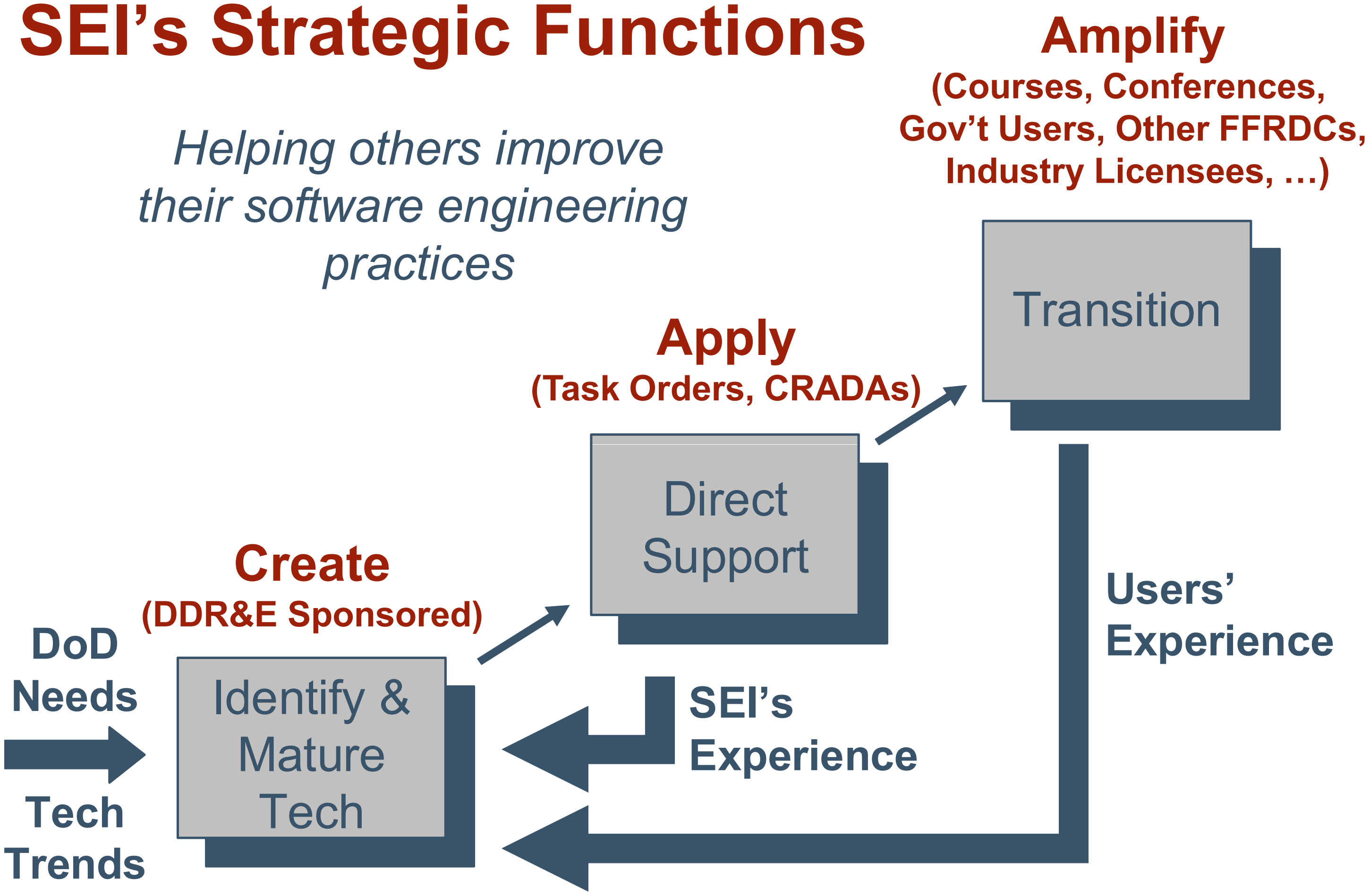
Software Engineering Measurement & Analysis

Technical Practice Initiatives

Management Practice Initiatives

SEI's Strategic Functions

Helping others improve their software engineering practices



Product Line Systems Program Strategy

Program Goal

Enable widespread product line adoption through architecture centric development

Program Strategy

- Product Line Practice Initiative
- Architecture Tradeoff Analysis Initiative
- Predictable Assembly from Certifiable Components Initiative

Product Line Practice Initiative Summary

Enable developers and acquirers to exploit the demonstrated commercial and DoD benefits of software product line practice.

Mature techniques for

- Finding and exploiting system commonalities and for controlling variability
- Developing, managing, and acquiring a software product line

and ensure that these techniques become routine software engineering practice among developers and acquirers.

Make product line development and acquisition a low-risk, high-return proposition by transitioning proven techniques that lower product

*(A software product line is a set of software-intensive systems sharing a common, managed set of features that satisfy specific needs of a selected market or mission and that are developed from a common set of core assets in a prescribed way.)

Software Architecture Technology (SAT) Initiative Summary

Ensure product quality by guiding and optimizing architectural design and by preventing software architectural risks in software-intensive systems

Provide acquirers and developers with an effective and integrated set of life-cycle architectural practices (design, documentation, evaluation, reconstruction), all based on proven software architecture techniques and methods.

Provide acquirers and developers with a spectrum of architecture evaluation techniques for software.

PACC Summary

Ensure that the builders of software-intensive systems have the ability to select software components on the basis of their certified runtime behavior and can reliably predict the runtime behavior of assemblies of components.

Provide a technology to certify components and thereby engender a class of trusted components from which specific systems can be built.

Provide the necessary technical underpinnings that make software component technology trusted, predictable, and successful in yielding software-intensive systems with the desired runtime behavior.

Dynamic Systems Impact Program Strategy

Improve software system-of-systems engineering by maturing the processes and models necessary to integrate systems and validate their performance and dependability qualities.

Performance Critical Systems

Summary

Establish methods for credibly analyzing and predicting performance, dependability, and interoperability properties of software systems prior to implementation and test.

Goal, Transition Objectives, Actions

Technical analysis processes, methods, and tools applied to software system models enable developers to diagnose performance, dependability, and interoperability problems while a system (and a system-of-systems) is being designed (i.e., prior to system integration and operational use).

Software systems engineers are using standards-based methods and tools to produce model-based predictions of system performance and dependability.

Project managers require and use predictive analyses of system behavior to assess the feasibility (risk) of developing a planned system.

Goal, Transition Objectives, Actions

A handbook of software systems engineering approaches guides software systems engineers in producing credible predictions of system performance and dependability behaviors.

Provide case study analyses of actual systems that have experienced performance and dependability problems to show the utility of proposed specification, modeling, and analysis techniques.

Provide parallel analyses of selected architectures of systems undergoing development to validate specification and analysis methods and to provide case study materials.

Create tutorials and case studies encouraging the application of AADL.

Provide a handbook of techniques for developing credible predictions of operational properties of software-intensive systems.

Integration Software Intensive Systems Summary

Identify, mature, and transition* software engineering practices and technologies to accomplish sustainable integration and interoperation across systems of systems.

We will accomplish this by:

- Defining key practices for constructive and programmatic interoperability
- Developing and piloting tools and technologies to support the key practices
- Establishing the integrated transition infrastructure to support DoD and industry partners

*transition is to both the acquisition and development communities

Software Engineering Management Program

“The quality of a software system is governed by the quality of the process used to develop and evolve it.”

Focus of Software Engineering Process Management Program

- Capability Maturity Models® (CMMs®)
- Team Software Process$^{\text{sm}}$ (TSP$^{\text{sm}}$)
- Software Engineering Measurement and Analysis (SEMA)

Summary of CMMI Objectives

Ensure that best engineering and management practices are implemented by organizations throughout the DoD and industry by means of integrated capability maturity models that support process improvement across an enterprise

Develop and make available CMMI adoption materials that aid organizations as they implement CMMI, the Software Acquisition Capability Maturity Model (SA-CMM), and the People Capability Maturity Model (P-CMM)

Hold workshops, deliver courses, conduct Standard CMMI Appraisal Method for Process Improvement (SCAMPI) appraisals, provide direct assistance to organizations, and support a vast transition partner program to ensure the acquisition and development communities can implement process improvement programs, understand the coverage of CMMI, SA-CMM, and P-CMM best practices, and understand the relationships these models have to other sets of best software engineering and management practices and standards

Summary Team Software Process -1

1. Create and transition into practice a scalable software engineering process that predictably produces secure, high-quality software to committed costs and schedules

2. Foster a professional, disciplined approach to software development based on defined processes, quality management, and industry standard metrics that are embraced and used by software developers, their management, and the DoD software acquisition community

3. Ensure that the transition mechanisms, professional resources, and infrastructure required to initiate and sustain transition are available within DoD, government, industry, and academia.

Summary Team Software Process -2

1. Work with DoD, government, and industry organizations to apply Team Software Process (TSP) to key software development projects. Demonstrate the applicability of TSP in typical software development settings where requirements changes, shifting priorities, and an evolving technology base are common.

2. Accelerate organizational process improvement in support of software business goals and objectives, and process maturity as defined in Capability Maturity Model Integration (CMMI).

3. Establish benchmarks for software process performance based on data gathered from participating TSP user organizations and promote their use by software developers and acquirers to improve management and engineering decision making.

SEMA Summary

1. Develop measurement and analysis guidance, information resources, and practices that assist the DoD and industry suppliers of software-intensive systems in quantitatively managing and improving their projects, processes, and organizations.

2. Measure the value and impact of selected innovations in the practice of software engineering by helping Software Engineering Institute (SEI) efforts identify, measure, and report the costs and benefits of their work.

International Process Research Consortium

On January 19 the SEI announced the launch of the new International Process Research Consortium (IPRC), a collaborative effort with academia, industry, and government worldwide to explore the frontiers of software process research. The principal goal of the IPRC is to formulate a research agenda, or roadmap, of “what’s next” in software process research, that will enable the research community and early adopters of process technology, to prepare for the next generation of software engineering challenges.

To date, notables from the research community including Vic Basili, Mario Fusani, Dieter Rombach, Barry Boehm, Terry Rout, and others have accepted their invitation to join.

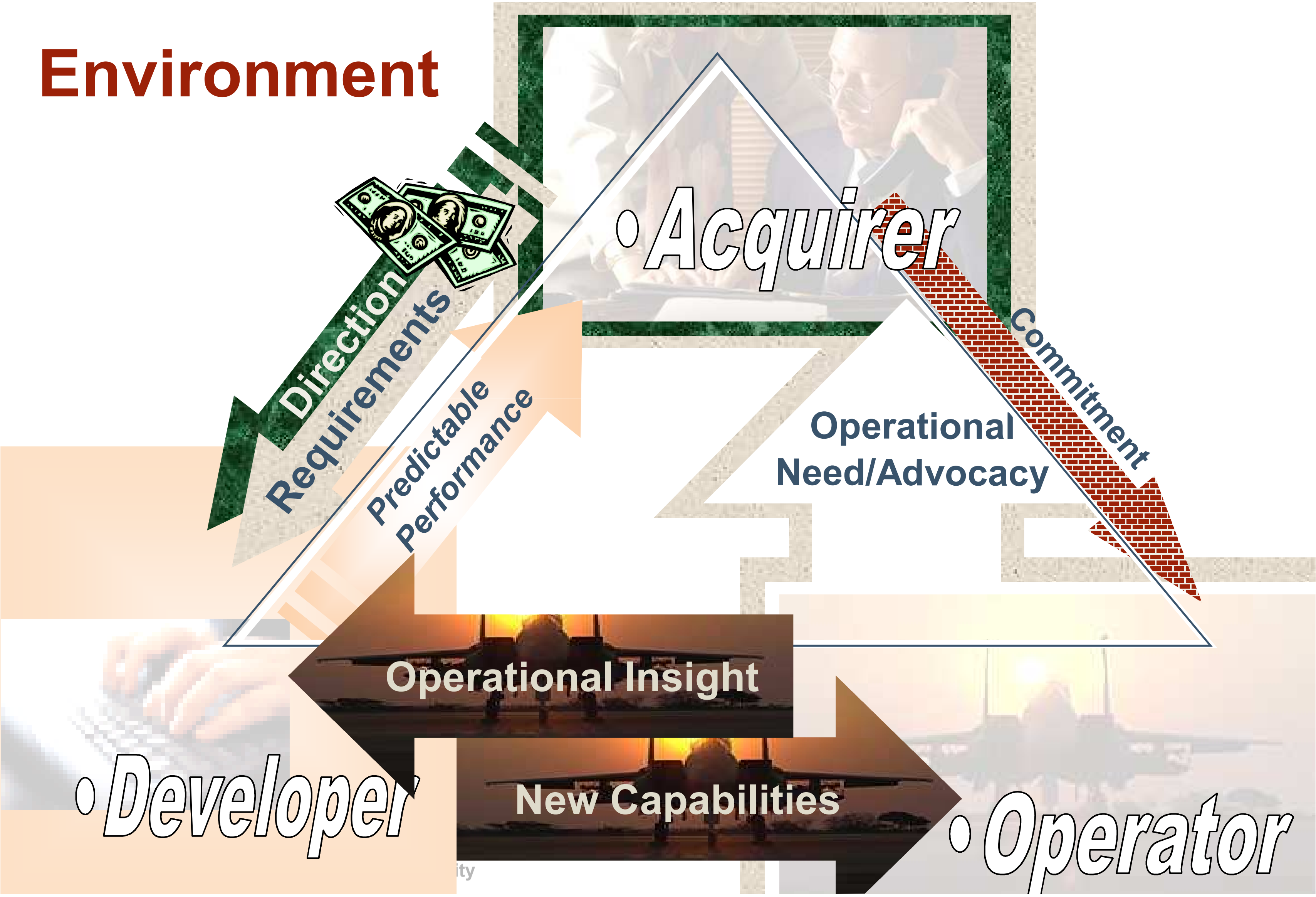
Acquisition Support Program Strategies

Understand and characterize the acquisition environment

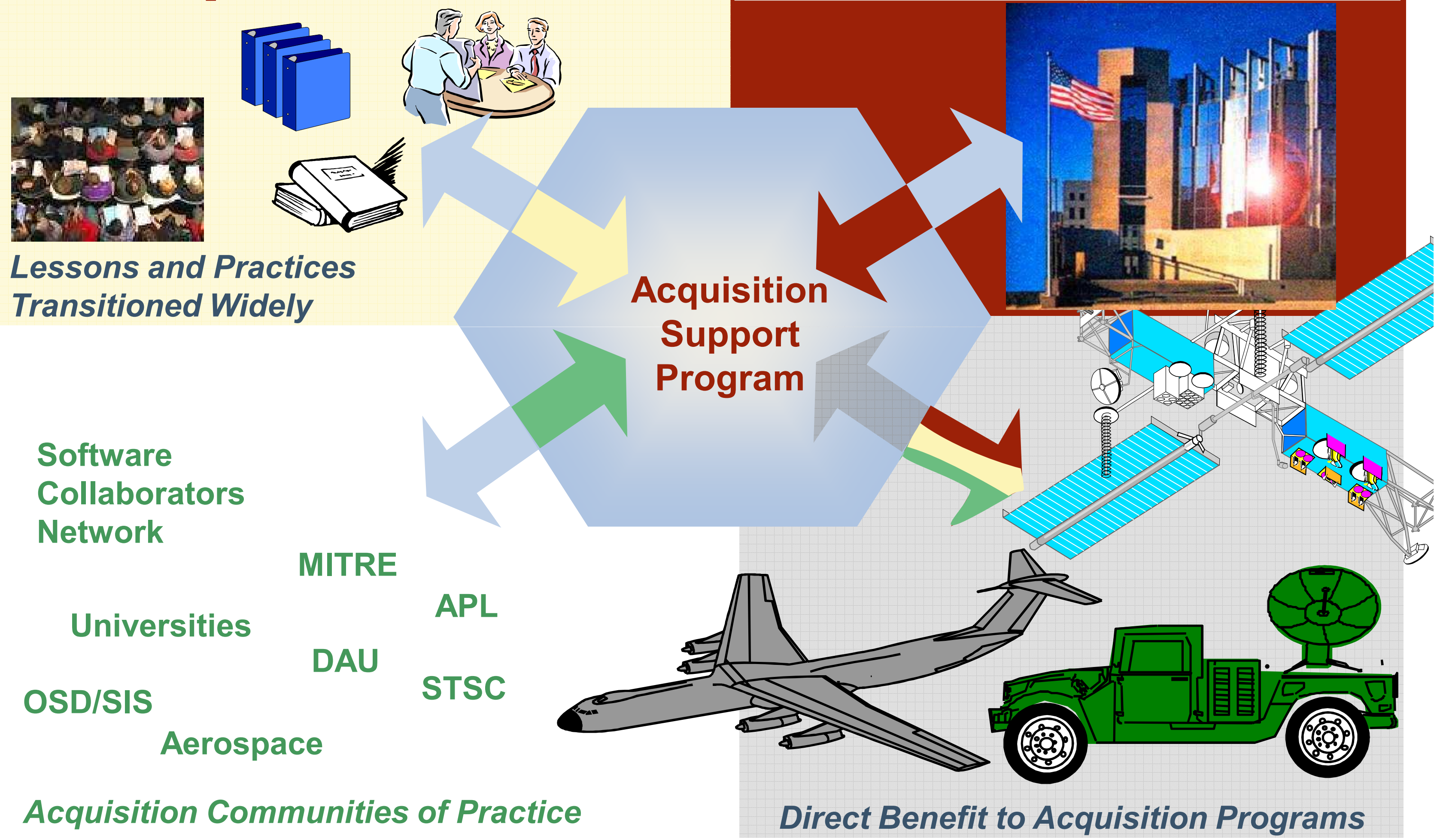
Work directly with key acquisition programs to help them achieve their objectives

Disseminate lessons and best practices widely

Environment



Operational Plan



Technical Practice Initiatives

Management Practice Initiatives

Lessons and Practices Transitioned Widely

Acquisition Support Program

Software Collaborators Network

MITRE

APL

Universities

DAU

STSC

OSD/SIS

Aerospace

Acquisition Communities of Practice

Direct Benefit to Acquisition Programs

“Challenge Problems from DoD to SEI”

1. Security, Survivability, and Assurance

- 1.1 Improve software development practices to include security and survivability as explicit design requirements.
- 1.2 Provide tools and controls used by software developers and buyers to test and evaluate software for security.
- 1.3 Maintain the integrity, availability, and survivability of narrow bandwidth wireless communications in mobile, ad hoc, and rapidly changing networks.
- 1.4 Provide tools and modeling capabilities to match the intended and actual functionality of mobile code to the policies of its execution environment; establish appropriate execution constraints. Provide tools and modeling techniques to design networked systems that are resistant to cascade failure effects, both malicious and accidental.
- 1.5 Provide tools and techniques to assure that reused and off-the-shelf software does not contain malicious or damaging code.

Challenge Problems….

2. Interoperability and Integration

•2.1 Improve the understanding, control, and predictability of the impact of modifications in systems-of-systems.

•2.2 Develop improved methods to enable coordinated evolution and interoperability of system-of-systems.

•2.3 Develop methods in software development and test to complement principles and practices in information architectures, enterprise architectures, and net-centricity.

Challenge Problems…

3 Software Technology R&D

•3.1 Develop improved, enhanced or new processes, principles, methods, and tools for determining expected properties of software systems before they are built and for confirming their as-built properties. Determine how to incentivize use of these technologies by DoD acquisition programs and their contractors.

•3.2 Dramatically decrease the amount of effort required for implementing new software-intensive systems. Incentivize use of productivity improvement technologies by DoD acquisition programs and their contractors.

•3.3 Minimize impact of run-time faults on system operation.

•3.4 Approaches to understanding, estimating, and measuring software reliability.

•

•3.5 Strengthen methodologies for evaluating COTS and other reused software to more accurately estimate expected cost and schedule impacts and understand the risks and benefits of using such products in a DoD system.

Challenge Problems….

4. Acquisition Management

4.1 Improve the software engineering skills of acquisition program managers and the acquisition workforce, workforce, including identifying and educating key personnel.

4.2 Collect and disseminate effective practices and lessons learned for acquisition of software-intensive systems including evolutionary and system-of-system acquisitions.

4.3 Improve evaluation criteria for use in contracting for software-intensive systems.

Challenge Problems…

4. Acquisition Management

4.4 Improve approaches for acquisition, design and implementation of DoD systems so they obtain the benefits of COTS product upgrades and guard against the risks of vendor and product instability.

4.5 Develop software-related acquisition metrics to:

- Judge the success of an acquisition organization
- Judge software product maturity
- Assess incremental benefits of products, artifacts, and processes spanning a system’s life.

Challenge Problems….

5 Sustainment

5.1 Enhance ability to sustain software systems including:

- Control and predictability over changes (requirements, fixes, tech refresh, etc.) in complex systems
- Insertion of those changes.

5.2 Establish a technology refresh strategy that encourages competition, improves quality, encourages innovation, and facilitates system interoperation.

5.3 Establish strategies for transition of software support from development to deployment and successful sustainment.

TSP and Secure Systems

The TSP provides a framework, a set of processes, and disciplined methods for producing quality software.

Software produced with TSP has one or two orders of magnitude fewer defects than current practice.

- 0.02 defects/KSLOC vs. 2 defects/KSLOC
- 20 defects per MSLOC vs. 2000 defects per MSLOC

If 5% of the defects are potential security holes, with TSP there would be 1 vulnerability per MSLOC.

TSP For Secure Systems -1

TSP for Secure Systems is a joint effort of the TSP team and SEI’s NSS (CERT) group.

The work is based on proven TSP quality practices and CERT’s extensive security skills and knowledge.

TSP secure augments PSP training and TSP introduction with specialized security training.

- secure design process
- secure implementation practices
- secure review and inspection methods
- secure test process
- security-related predictive measures

TSP For Secure Systems -2

The goal of the project is to develop a TSP-based method that can predictably produce secure software.

The TSP for Secure Systems project is developing a process and support system that will

- support secure systems development practices
- predict the likelihood of latent security defects
- be dynamically tailored to respond to new threats

TSP for Secure Systems will be tested in several pilots.

TSP-Secure Pilot Workshop

The purpose of the workshop was to

- Convince the team that
  - software security is synonymous with software quality
  - the quality methods the team is already using with the TSP and the PSP can be easily extended to address security issues
- prove the feasibility of using the TSP to develop secure software
  - pilot and test initial ideas for TSP-Secure
  - establish a baseline from which to expand and refine TSP-Secure

SAT and PACC

SAT:

- Quality attribute requirements drive software architecture design.
- Software architecture drives software development throughout the life-cycle

PACC:

- “Smart” architectural constraints lead to predictability.
- Using component technology as a carrier of quality attribute-based restrictions.

Common Approach:

- Precisely define quality attribute requirements in terms of scenarios.
- Exploit the “structure” of quality attribute models to define the structure of well-formed architectures.
- Define transformations between architecture models, quality attribute models, quality attribute scenarios, quality attribute measures and implementations.

Security for SAT

Currently researching the key architectural decisions used to realize quality attribute requirements – we call these “architectural tactics”

- In collaboration with NSS we are investigating architectural tactics for security

Currently developing a prototype rule-based system that can serve as an expert architecture design assistant

- Will be investigating the incorporation of security-based design rules

Security for PACC

Planning to investigate several lines of research for including security in the PACC agenda

- Using “modeling checking” (which is already being used to ensure reliability/safety for PACC) to expose security attack scenarios (current research by Jeanette Wing at CMU)
- Using “proof carrying code” as a vehicle for formally certifying component security (current research by Peter Lee at CMU)

Networked Systems Survivability Program Strategy

Ensure that appropriate technology and systems management practices are being used to design and implement systems that recognize, resist, and recover from attacks on networked systems.

Survivable Network Technology (SNT) Summary

Historically, security has been addressed in networked systems through the addition of boundary controls such as firewalls, intrusion detection systems, and virtual private networks which are only partly effective. New techniques are needed across the software development life cycle to produce systems with “built-in” security, systems that are better able to resist, recognize, and recover from attacks.

Function extraction (FX) helps engineers faced with the task of building on legacy code to ensure they have a detailed understanding of the existing code base and ensure all of a system’s software is free of embedded malicious code.

Intrusion-aware design (IAD) provides a structured way to analyze a system’s response by using realistic scenarios of likely attacks and use the analysis results to improve the design to defend against those types of attack.

The V-RATE method helps designers assess the security risk of using COTS products.

CERT Analysis Center Summary

Develop and implement applied research into methods and technologies that provide early indications and warnings of attacks against the Internet, critical infrastructure sectors, and large distributed organizations. Specific focus of this work is in the areas of insider threats and emerging wireless technologies and their impact on law enforcement.

CERT Analysis Center

Customers/Collaborators

- U.S. Secret Service
  - Insider Threat Study
  - Wireless Security Issues
  - CSPI
- Department of Homeland Security
  - Insider Threat Study

Survivable Enterprise Management Summary

Assist DoD sites and critical infrastructure organizations in adopting (or maintaining) effective survivability management methodologies and information security practices, thus enabling them to reduce the number of successful attacks and to recognize, resist, and recover more rapidly from attacks that do occur.

Survivability risks and issues are currently addressed by organizations and system operators in an incomplete and ad hoc fashion. Mitigation activities frequently do not consider mission priorities or the organization’s risk tolerance as part of security decision-making activities. Detailed guidance on the effective management of survivability at the enterprise level is not available, and a validated, practice-based approach for organizational security improvement does not exist.

Practices, Development & Training Summary

Promote the adoption and widespread use of security and survivability practices and standards, and increase the quality and quantity of practitioners and of personnel who are well qualified to manage and respond to computer security incidents.

The SEI has identified many deficiencies in the design, development, and implementation of technology and in operational practice that leave organizations vulnerable to a variety of attacks, accidents, and failures. The current challenge is to develop comprehensive training and education programs that cover the spectrum from identifying information assurance needs and understanding core principles to implementing solutions.

CERT/CC Summary

Limit the scope and damage of attacks against Internet sites and the Internet infrastructure and enable the construction of more secure and survivable current-generation systems in the short term. Provide information and services that enable the DoD, federal civilian agencies, and critical national infrastructure operators to protect themselves from known threats and vulnerabilities and to recover from security breaches quickly. Establish an international data collection and analysis capability that improves the response community’s ability to recognize indicators of new attacks and provide timely warnings to minimize damage and interdict intruders.

Network Situational Awareness (NetSA) Summary

Develop a comprehensive strategy for enhancing situational awareness for system and network administrators, technology managers, and policy makers. Activities include research into methods and technologies that provide early indications and warnings of attacks against the Internet, critical infrastructure sectors, and large distributed organizations.

A spiral model of research, prototype development, and transition to the vendor community will be employed to develop and transition operational methodology and Tools.

Worthy of Note in NSS

1. 10th Association for Computing Machinery Conference on Computer and Communications Security presentations and workshop (Oct 2003)
2. Completed training of over 50 U.S. Secret Service agents assigned to three 2004 NSSEs (Nov-Dec 03) (Unplanned work)
3. Public release of OCTAVE-S (both OCTAVE and OCTAVE-S now available as free downloads)
4. Development of OCTAVE-influenced modules for the Critical Systems Protection Initiative (CSPI) training. Three offerings of training delivered to USSS agents in 1Q
5. Published *State of the Practice of Computer Security Incident Response Teams* as SEI TR in Oct 03
6. **Celebration of 15th anniversary**
7. Handled 37,511 incidents

CyLab

1. Carnegie Mellon's CyLab – President Cohon formally announced the creation of CyLab that combines the university's existing expertise and related research centers under one umbrella.

2. CyLab builds upon the university's proven problem-solving approaches and a record of interdisciplinary research with more than 50 researchers and 80 students from Carnegie Mellon including the SEI and CERT Coordination Center.

3. CyLab is co-directed by Rich Pethia and Pradeep Khosla.

Department of Homeland Security US-CERT

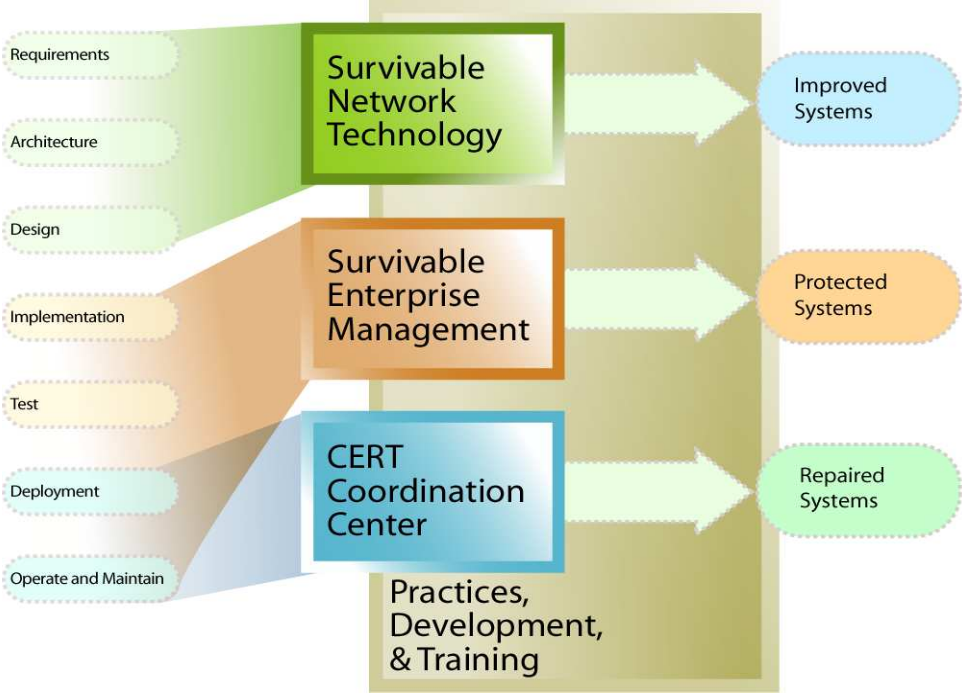
1. In September 2003, the Department of Homeland Security announced a partnership with the CERT Coordination Center to create US-CERT, a coordination point for prevention, protection, and response to cyber attacks across the Internet.
2. Members of the CERT/CC staff including technical experts, writers, editors and others have worked with DHS:
3. to launch the National Cyber Alert System, to launch the US-CERT Web site (www.us-cert.gov)
4. In addition, several SEI staff members are part of the the National Cyber Security Divisions Task Forces to address computer security issues.
5. US-CERT's objectives are to: Aggregate available cyber security information
6. Provide it to individuals and organizations in a timely manner

CERT Centers

CERT was formed in 1988 in response to the Internet Worm

CERT added research, training, and analysis as the Internet matured

September 15, 2003 CERT Centers is named the US CERT (www.us-cert.gov) in partnership with DHS



Carnegie Mellon
Software Engineering Institute

CERT® Coordination Center

Solving today's security problems

Artifact Analysis

Study intruder code to develop defenses

Developing new techniques for analysis

Vulnerability Handling

Analyze flaws in Internet systems

4,000 vulnerabilities handled each year

Publications available at http://kb.cert.org/vuls/

Incident Handling

Respond to security emergencies on the Internet

Measure exploitation of flaws

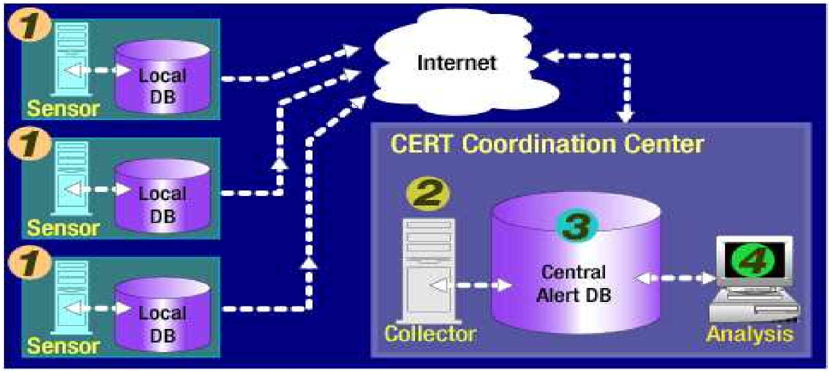
100,000 incidents handled each year

Publications available at http://www.cert.org

Carnegie Mellon
Software Engineering Institute

AirCERT (Automatic Incident Response CERT)

Technology needed to handle exponential growth in incidents & develop systems of indications and warnings



Key Ideas

Open-source infrastructure to automatically gather & report security events from Internet sites to the CERT/CC

Reduce the burden on security analysts by automatically handling well-understood attacks

Spot problems not visible from a local perspective

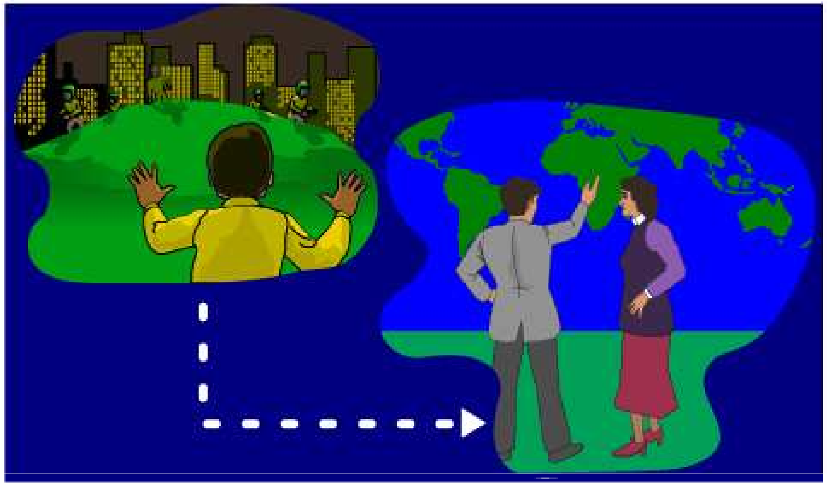
Use and Status

Gather structured, security incident data for analysis to identify current trends, scope of a specific widespread incident, & predictive indicators for attacks

Completed proof-of-concept prototype; some components being tested by the Internet community, piloting with GSA & agencies

Carnegie Mellon
Software Engineering Institute

CERT® Analysis Center



Need

Attacks occur at Internet speed and cause major damage within reaction cycles; we need predictive and preventative capability

Key Ideas

Augment existing, inadequate, IDS technology

Dynamically adjust for rapid changes in environment

Protection against new threats

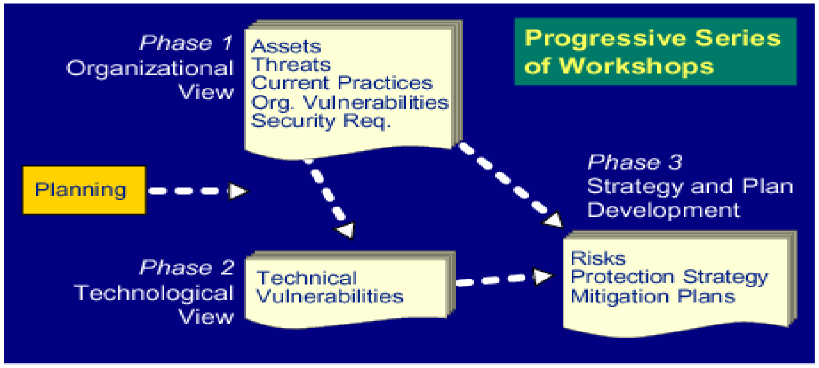
Use and Status

Studying feasibility of data collection, reduction & fusion processes

Initial pilot successful at identifying severe operational anomalies & previously undetected probes

Carnegie Mellon
Software Engineering Institute

OCTAVE℠



Need

Effective security management programs must be sensitive to mission and overall objectives.

Key Ideas

Information security must be linked to an organization’s mission & business objectives for effective planning

Enable interdisciplinary teams to perform information security risk evaluations & act as a focal point for improvement efforts

Use and Status

Actively piloting in DoD, government, & industry sectors

Created first derivative method: OCTAVE-S for small organizations

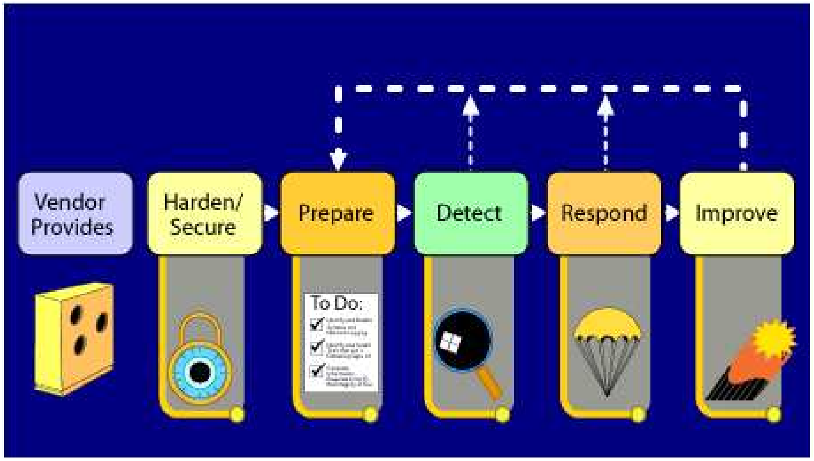
Offering training

Seeking transition opportunities

Security Practices

Need

Pervasive understanding of security policy, management practices and technical practices



Key Ideas

Organizations can improve the security & survivability of networked systems by adopting CERT® security practices

Use and Status

Practices are published on the web & taught in training courses

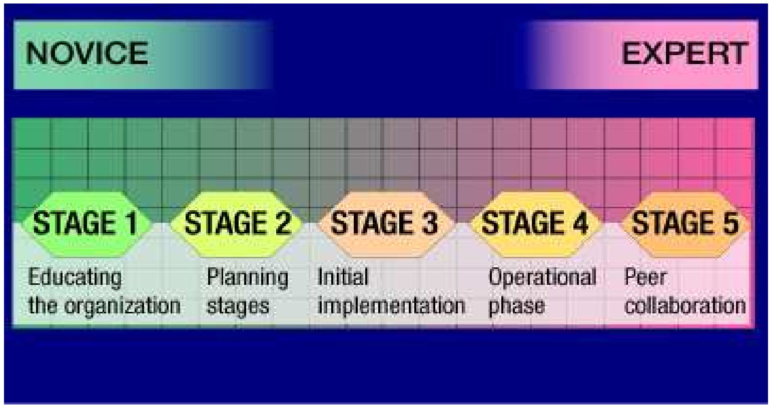
Working on certification standards

Seeking DoD pilot sites & transition opportunities

CSIRT (Computer Security Incident Response Team) Development

Need

Organizations need teams to respond to computer security incidents



Key Ideas

Develop a community of CSIRTs to share resources and respond to global incidents

Engage organizations as partners depending on the maturity of their CSIRT capability

Use and Status

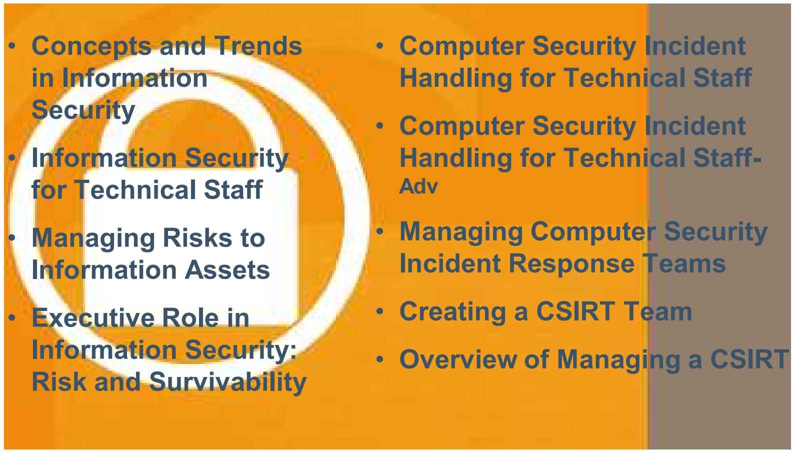
Assisting DoD and other sectors to develop a certification and accreditation process for CSIRTs

Using CSIRT training courses as a transition mechanism for our knowledge and experience

Training

Need

Improve the information security skills of technical staff and managers to address the increasing gap between core competencies required and number of qualified personnel



- Concepts and Trends in Information Security
- Information Security for Technical Staff
- Managing Risks to Information Assets
- Executive Role in Information Security: Risk and Survivability
- Computer Security Incident Handling for Technical Staff
- Computer Security Incident Handling for Technical Staff-Adv
- Managing Computer Security Incident Response Teams
- Creating a CSIRT Team
- Overview of Managing a CSIRT

Key Ideas

Approaches exist to protect critical information assets and systems

All levels of staff need training to facilitate adoption of security practices

Use and Status

Offering public and customer deliveries

Seeking transition and licensing partners