INN OVA TIO NS

Software Engineering Institute | Carnegie Mellon University

Results, Relevance, and Impact

Software increasingly becomes integral to how we work, communicate, and fight our nation's adversaries. As we have come to depend more on software, we face the risks that arise from this dependence. The size and complexity of software, as well as the interconnectedness of software-enabled systems, mean possible exposure to disruptive, damaging events. These stem from not only software quality issues, emergent behavior, and unforeseen dependencies—but also cyber-attack by hackers, insiders, criminals, nation states, and terrorists.

Meeting challenges from these ever-evolving capabilities, use cases, and threats requires continual innovation in the way we build and secure software and the systems on which we depend.

Since its establishment in 1984 as a federally funded research and development center (FFRDC) sponsored by the U.S. Department of Defense (DoD), the Carnegie Mellon University Software Engineering Institute (SEI) has delivered innovative methods, tools, algorithms, and frameworks to meet current software and cybersecurity needs and serve as a foundation for combatting future ones.

Because of our role as a research leader, SEI technical work often produces fundamental approaches that bear fruit years later. For example, SEI pioneering work in software architecture in the 1990s led to the accepted understanding today that architecture determines the quality and longevity of a software system.

We also address emergent needs with our work. For instance, SEI experts closed gaps in network data collection and analysis by developing a suite of cybersecurity tools and a system that now provides traffic monitoring and protection throughout Federal networks (the Einstein system).

SEI innovation continually advances. Our compilation of capsule stories in this booklet ends with work that made impact in 2015. However, our researchers and engineers continue to investigate the toughest problems, develop frameworks for solutions, create prototypes and algorithms, and deliver software tools to help with critical areas such as trust in autonomous systems, real-time system verification, and decision analytics.

Our Core Technical Areas

We effect the transfer of our research into operations through work in our seven core technical areas. Any innovation we produce stems from at least one of these areas. In this compilation, we note the connection between innovation and our core technical areas by a subtle color-coding on each innovation capsule story.

- 🟪 **Cyber Missions**
- 🟥 **Data Modeling & Analytics**
- 🟥 **Human-Machine Interactions**
- 🟧 **Autonomy & Counter-Autonomy**
- 🟩 **Software & Information Assurance**
- 🟩 **C4ISR Mission Assurance**
- 🟦 **Systems Verification & Validation**

Table of Contents

2015

Creating a New Language to Verify Complex Systems

Distributed Adaptive Real-Time (DART) systems (e.g., Unmanned Air Systems) are key to DoD capability. These systems are safety critical, resource constrained, and sensor rich, and they adapt autonomously to their physical environments.

In general, formally verifying a DART system is intractable. Coordination, adaptation, and uncertainty pose key challenges to assuring their safety- and mission-critical behavior. The typical approach to verifying DART systems is to test, rigorously and exhaustively. Testing, however, is usually performed later in development and cannot account for all reactions of an essentially autonomous system.

One innovative approach that SEI researchers are using involves creating a new programming language for DART systems called the DART Modeling and Programming Language (DMPL). DMPL is a C-like language that can express distributed, real-time systems. The semantics of this language are precise; they support formal assertions usable for model checking, an evolving area for testing complex software systems. In addition, system developers can express physical and logical concurrency in DMPL to perform timing analysis.

SEI investigation into verifying DART systems will also produce other tools for mixed criticality scheduling and model checking. In addition, work to verify DART systems continues longer-term SEI research into mixed-criticality and real-time scheduling, model checking, and High Confidence Cyber-Physical Systems (HCCPS).

2015

Enhancing Computing Power at the Edge

As part of its mission to transition the technologies it develops into use, the SEI in 2015 made its implementation of tactical cloudlets, KD-Cloudlet, freely available in its open-source code repository on GitHub.

To support their missions, military and emergency personnel operating in crisis and hostile environments increasingly use mobile applications. Most of these applications perform computation-intensive tasks such as speech and image recognition, natural language processing, and situational awareness enhancement. These tasks take a heavy toll on the mobile device's battery power and computing resources. Unfortunately, battlefield and disaster environments are not only at the edge of the network infrastructure but also resource constrained.

Cyber-foraging augments the capabilities of resource-limited mobile devices by leveraging compute resources in the surrounding environment. Cloudlet-based cyber-foraging relies on discoverable, generic, and forward-deployed servers located in single-hop proximity of mobile devices.

Using KD-Cloudlet, developers can turn any system running Linux—from a laptop to a more powerful server—into a discoverable source that can be used by nearby mobile devices for computation offload and data staging.

The KD-Cloudlet tool's release springs from several years of SEI research into the use of cloud computing at the tactical edge. The research into the needs and constraints of tactical environments drove the development of the tactical cloudlets. SEI researchers collaborate in this ongoing research with the creator of the cyber-foraging and cloudlet concepts, Dr. Mahadev Satyanarayanan of CMU.

2015

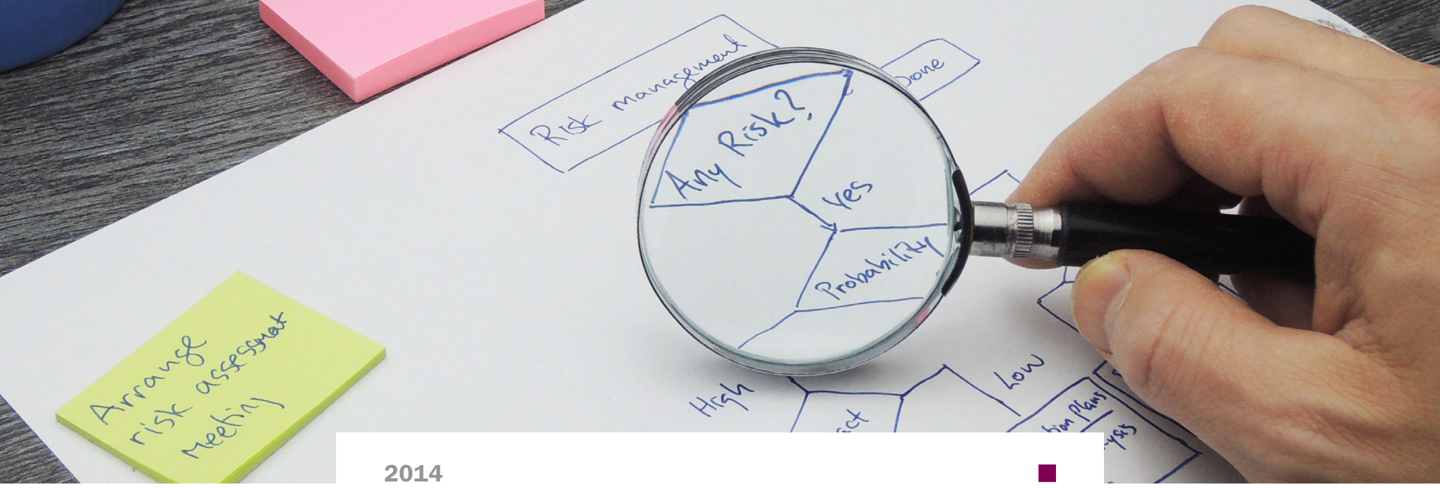
Integrating Early to Prevent Costly Problems

In a 2014-15 shadow exercise, an SEI technical team rapidly detected potential integration issues early in the military's Joint Multi-Role (JMR) technology demonstration program that traditional approaches missed, using its Architecture-Centric Virtual Integration Practice (ACVIP). The findings led to ACVIP adoption by JMR contractors and its inclusion in RFPs for new projects. JMR technology demonstrations are a precursor to the Future Vertical Lift military helicopter program.

The roots of ACVIP are in SEI research into virtual integration that began in 1998. Unlike the traditional development approach of *design-build components-integrate-test*, the virtual integration approach employs architectural modeling to make sure the components work together before building components in conformance to the model.

DoD line funding enabled SEI researchers to lead the technical development of the SAE Architecture Analysis and Design Language standard (established in 2004) for the specification, analysis, automated integration, and code generation of real-time, performance-critical, distributed computer systems. Line funding, together with sponsored work for the Army and others, enabled the SEI to produce the Open Source AADL Tool Environment (OSATE) workbench for implementing virtual integration.

In 2008, the international Aerospace Vehicle Systems Institute (AVSI), whose membership includes defense industry organizations, chose AADL and OSATE for its System Architecture Virtual Integration Initiative (SAVI), based on evidence that the technologies offer a means to achieve an *integrate-then-build* approach to evolving complex systems.



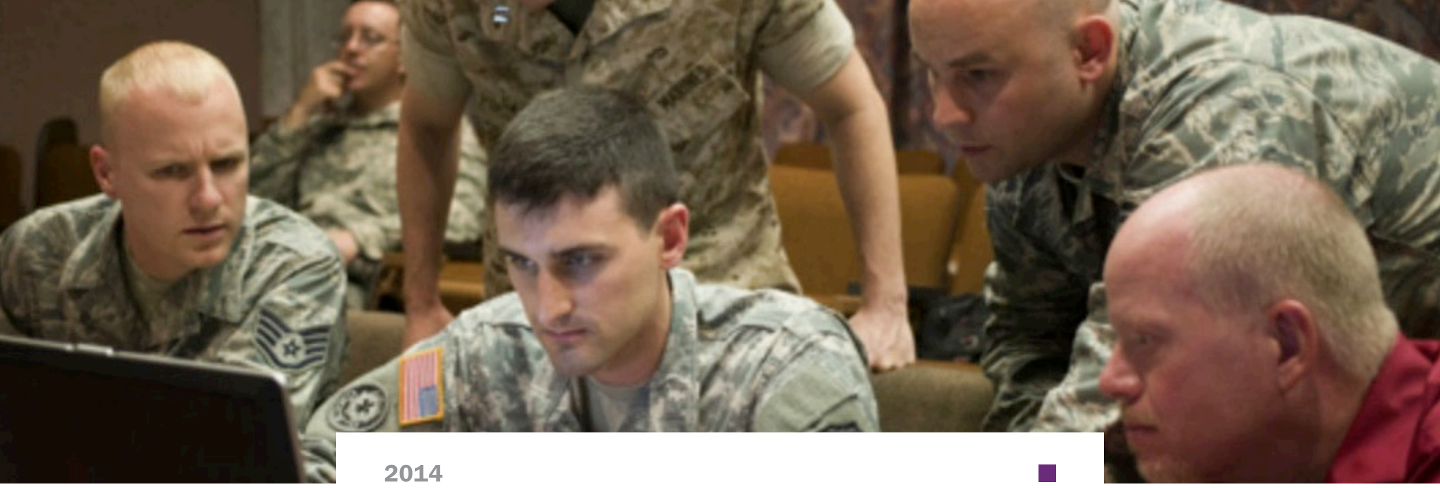
2014

Taming Uncertainty in Software Cost Estimation

In 2014, SEI researchers used their Quantifying Uncertainty in Early Lifecycle Cost Estimation (QUELCE) method in a workshop with a live major defense acquisition program (MDAP). This milestone along the way to transitioning innovation into the acquisition lifecycle is the result of focused research and development.

DoD acquisition regulations call for early (pre-Milestone A) estimates that stretch across the entire program lifecycle, including operations and support. These early cost estimates rely heavily on expert judgments about cost factors. However, the ways in which cost factors may change through the lifecycle has received little attention.

Based on research initiated in 2011, the QUELCE method provides an explicit, quantified consideration of the uncertainty of the change drivers. In doing so, QUELCE enables calculation (and re-calculation) of the cost impacts caused by changes that may occur during the program lifecycle. The result is that this approach enhances decision-making through transparency about the expert assumptions that underlie the cost estimate.



2014

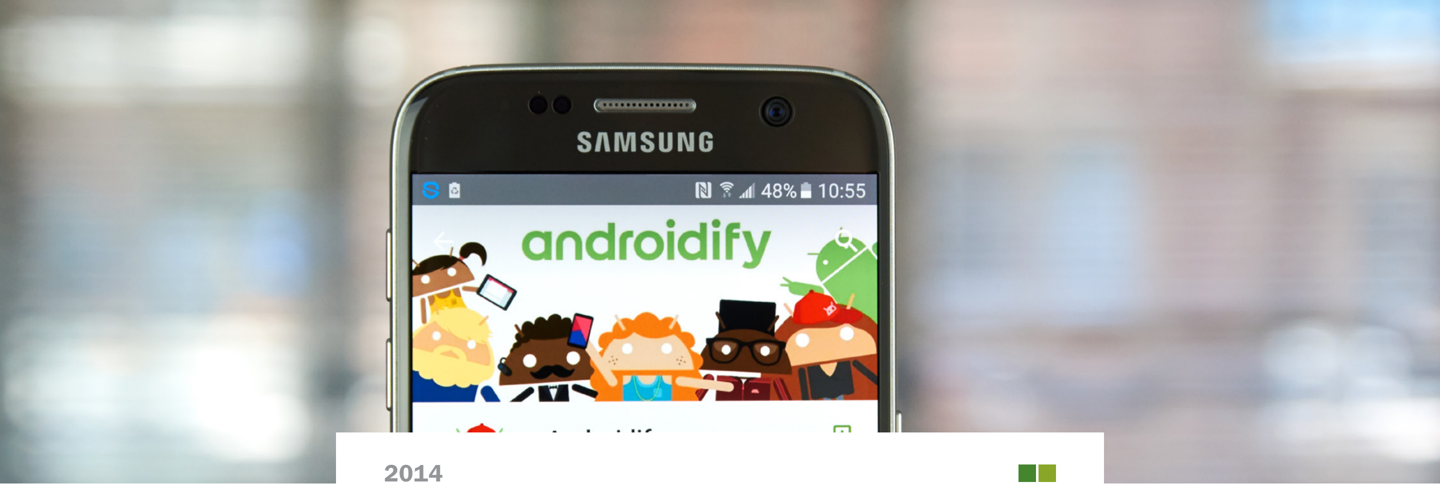
Enabling a Stronger Cyber Workforce

For more than 15 years, the SEI has been investing in developing learning platforms and courseware for cyber warrior readiness in DoD and other government organizations.

The SEI CERT Division developed an initial Virtual Training Environment (VTE) platform using DoD Research funding in 2001. By 2005, VTE was being used to address DoD training and capability-building challenges related to information security. In 2012, VTE was redesigned to meet cyber workforce training requirements and transitioned as FedVTE to serve tens of thousands of government and military users. The CERT Division estimates that FedVTE has saved the government over $70M through 2015 by providing the equivalent of 24,000, 5-day training courses.

The CERT Division followed VTE with a web-based system, the CERT Exercise Network (XNET). The USCYBERCOM Exercise Network is a customized instantiation of XNET. In 2012, CERT introduced the Simulation, Training and Exercise Platform (STEP), a flexible, multimedia, e-learning environment that students can access anywhere, anytime. STEP has formed the backbone infrastructure for USCYBERCOM's Cyber Flag and Cyber Guard joint exercises since their inception.

Most recently, in 2015, CERT researchers prototyped an Automated Cyber Readiness Evaluator platform to provide a scalable, objective assessment that validates the technical knowledge and skills of the government's cyber workforce.



2014

Attacking Software Vulnerabilities

In 2014, SEI's CERT division introduced the Tapioca tool to check Android apps for vulnerabilities. In the first year of use, Tapioca was used to check more than 1 million Android apps.

The release of the open source Tapioca tool, a network-layer man-in-the-middle proxy virtual machine, is one bit of evidence of the CERT Division's continuing commitment to proactive vulnerability discovery. The CERT Division vulnerability analysis team maintains over 1,400 vendor contacts, creating vulnerability reports that eventually appear as entries in the National Vulnerability Database.

The SEI also works directly with US-CERT to publish Vulnerability Notes directly to the US-CERT website, where they are considered the authoritative statement from the government regarding a given vulnerability. In addition, CERT is the only organization that has proven to be able to, repeatedly and successfully, coordinate responses to a vulnerability across industry, the DoD, and the federal government.

2014

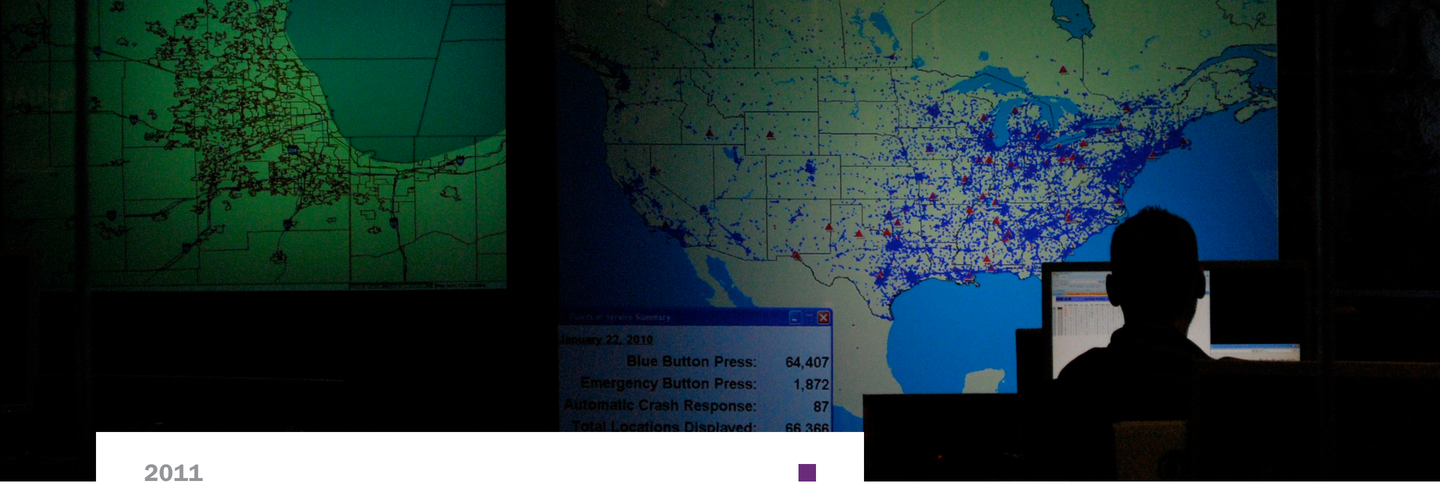
Building Capability to Defend Against Malware

Malicious code, or malware, is a piece of software that runs without the user's explicit consent, maybe without the user's knowledge. Historically, malware has caused seen nuisance-type results (such as delivering some unwanted content). In the last decade or so, more malware has focused on committing crimes, such as stealing identity or taking control of a computer.

For malware analysts, a significant challenge derives from the fact that malware rarely has source code available. Analysts must grapple with sophisticated data structures exclusively at the machine code level.

To help analyze malware, SEI CERT researchers are developing a suite of binary static analysis tools based on a framework called *Pharos*. SEI built this framework on Lawrence Livermore National Laboratory's (LLNL) ROSE compiler infrastructure. The Pharos tool suite includes many extensions to the binary analysis features of ROSE that SEI has jointly developed with LLNL. The Pharos tools use static analysis techniques, such as control flow analysis and dataflow analysis, to reason about the behavior of data structures in binary files.

In 2014, SEI's CERT Division completed research to eliminate bottlenecks in the process of deriving actionable insight from malware by automating tasks and providing more semantically rich abstractions used by a malware analyst.


January 22, 2010
Blue Button Press: 64,407
Emergency Button Press: 1,872
Automatic Crash Response: 87
Total Locations Displayed: 66,366

2011

Assessing Cyber Risk Readiness

One lesson of the past 20 years is that organizations cannot expect to prevent every cyber-attack. Instead, they must be ready to continue operations and meet their missions when disruption occurs.

The SEI's CERT Division tools for cyber risk and resilience promote a structured approach to managing security risks, business continuity, and information technology operations within the context of business objectives.

Created by the CERT Division for the U.S. Department of Homeland Security (DHS) in 2011, the Cyber Resilience Review (CRR) is a no-cost, voluntary, non-technical assessment to evaluate an organization's operational resilience and cybersecurity practices. The CRR assesses enterprise programs and practices across a range of 10 domains based on the CERT Resilience Management Model (CERT-RMM), including asset management, vulnerability management, incident management, risk management, and situational awareness. In 2014, DHS released a CRR self-assessment guide to allow organizations to conduct a CRR without outside facilitation. In 2015 alone, the CERT Division conducted 48 CRRs in 10 critical infrastructure sectors.

In 2012, the CERT Division developed the Risk and Vulnerability Assessment (RVA) to aggregate vulnerability data in support of informed decisions regarding the security and safety of information systems. An RVA combines national level threat and vulnerability information with assessment data to provide specific risk analysis reporting and remediation steps. An RVA provides information on network mapping, penetration testing, wireless networks, databases, and other areas. During 2015, the CERT Division worked with DHS to conduct 46 RVAs.

In 2015, the CERT Division and DHS launched the External Dependencies Management (EDM) assessment. This in-person, DHS-facilitated evaluation measures how well an organization can handle cyber disruptions in key services provided by third parties. Any external dependency presents possible risk, from service agreements for cloud computing to business relationships that depend on a third-party's computing infrastructure and security.

2009

Certifying the Software Architect Role

In 2009, the U.S. Army mandated that all Project Executive Offices (PEOs) appoint a chief software architect (CSWA) to be responsible for oversight and management of software development within each PEO. The memo specified that the CSWA must earn a Software Architecture Professional Certificate from the SEI (or equivalent). The Army based its decision on an understanding of SEI work in software architecture and, in particular, a recent impact study of the use of SEI architecture evaluation techniques in the Army [SEI 2009, Nord 2009].

The Army's mandate reflected appreciation for the value of more than 15 years of SEI innovation and leadership in software architecture definition, evaluation, analysis, and documentation. SEI work included the first software architecture book for practitioners, *Software Architecture in Practice*, winner of the prestigious JOLT award from *Software Development* magazine. Three other equally seminal books followed. All SEI software architecture books are cited often, have been updated in multiple editions, and have collectively sold more than 150,000 copies.

These books form the foundation of training courses and certificate programs, in which people from more than 900 organizations in industries such as defense, financial, healthcare, insurance, and energy have been trained by SEI experts. More than 80 colleges and universities around the world have adopted SEI software architecture curriculum. The work also spawned the annual Software Engineering Institute (SEI) Architecture Technology User Network Conference.

2009

Augmenting T&E with Assurance

SEI work on the use of assurance cases in the development of medical devices [Weinstock 2009] led directly to the FDA's issuing draft guidance to manufacturers recommending the use of assurance cases. As a result, infusion pump manufacturers are beginning to make use of them.

It is difficult to assure the safety, security, or reliability of complex, net-centric systems of systems (such as medical devices and military weapons systems) because of their size, complexity, and continuing evolution. In addition, those types of systems can exhibit undesired and unanticipated emergent behavior (that is, actions of a system as a whole that are not simple combinations of the actions of the individual constituents of the system).

Traditional software and systems engineering techniques, including conventional test and evaluation (T&E) approaches, cannot provide the justified confidence needed. The assurance case provides a means to structure the reasoning that engineers use implicitly to gain confidence that systems will work as expected. It also becomes a key element in the documentation of the system and provides a map to more detailed information.

The concept of an assurance case derives from the safety case, a construct that has been used successfully in Europe for over two decades to document safety for nuclear power plants, transportation systems, automotive systems, and avionics systems.

2009

Codifying Resilience Practice

In the aftermath of the 9/11 terror attacks, organizations began to seek answers to predictably and systematically controlling operational resilience through activities such as security and business continuity.

In October 2003, a group of 20 Information Technology (IT) and security professionals from defense organizations, the financial services sector, IT, and security services met at the SEI to identify what could enable and accelerate IT operational and security process improvement. The bodies of knowledge identified included IT and information security governance, audit, risk management, IT operations, security, project management, and process management.

Soon after, in March 2005, the SEI began work with the Business Continuity Committee of the Financial Services Technology Consortium (FSTC), exploring the development of a reference model to help determine an organization's capability to manage operational resilience. Drawing on its experience with developing and evolving the widely used Capability Maturity Model Integration (CMMI) framework, the SEI developed the CERT Resilience Management Model (CERT-RMM) of 26 process areas.

Since 2009, organizations in the DoD, the U.S. defense industrial base, U.S. federal civilian agencies, the financial services sector, and academia have been using the CERT-RMM to institutionalize improved processes for managing operational resilience and measure their benefit.

2007

Strengthening Network Traffic Analysis

In 2007, the National Cyber Initiative made Einstein mandatory for all federal civilian agencies. The Department of Homeland Security (DHS) Einstein program helps protect federal computer networks and the delivery of essential government services.

First deployed in 2004, Einstein's capabilities for situational awareness are used throughout the federal government in part because of a casual conversation between SEI staff members and the DoD. That conversation led to the research and collaboration that produced a sophisticated suite of tools that can characterize network threats, assess the impact of security events, and identify vulnerable network infrastructure. Einstein integrates several distinct data collection/analysis systems and toolsets for network traffic analysis developed at the SEI CERT Division.

Initially, Einstein collected summary network traffic information at agency gateways and provided a high-level view of federal government network connections. The program has grown to provide an automated process for collecting, correlating, analyzing, and sharing computer security information across the federal government to improve our nation's situational awareness.

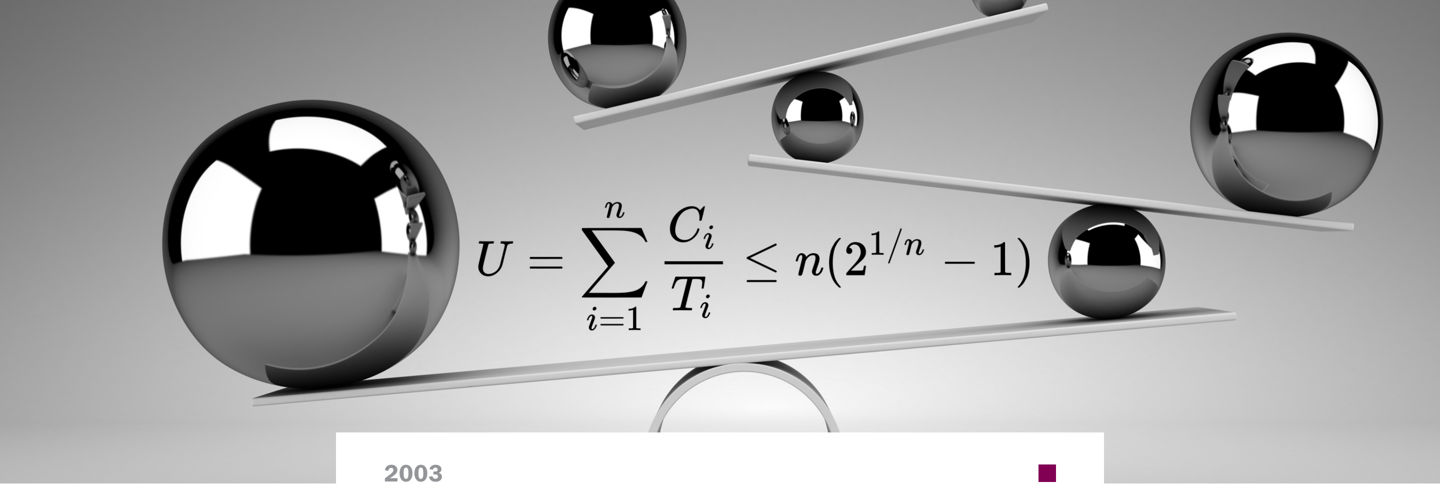
2004

Leading the Growth of an Architectural Modeling Standard

In 2004, SAE International published the industry standard SAE Architecture Analysis and Design Language (AADL). Growing from DARPA-funded research into MetaH and Acme architectural languages a decade or more before, the development of the AADL was shepherded by the U.S. Army Aviation and Missile Research Division (AMRDEC) Software Engineering Directorate (SED) with technical leadership by the SEI.

Focused for several years in its research on architectural modeling and analysis for safety- and mission-critical systems, the SEI worked effectively across industry, government, and academic organizations to fashion the initial standard language and subsequent annexes. As technical lead for the standard, the SEI integrated several research technologies into the AADL standard, making it extensible, semantically well defined, and consistent.

Through its creation of the Open Source AADL Tool Environment (OSATE), in addition, the SEI has fostered pilot applications of AADL in a range of industrial pilot projects. Also, developers and researchers are finding that AADL and OSATE provide a technology transition platform, as shown by their integration with formal analytical frameworks such as SysML (Systems Modeling Language) and MARTE (Modeling and Analysis of Real-Time and Embedded systems).



2003

Defining Non-Functional System Qualities

Quality (or non-functional) attribute scenarios form a common language between users and software developers, playing a significant role in both requirements specification for an architecture and integration testing to see that requirements will be met. Assessing how secure, timely, reliable, and usable systems must be is now a fundamental component of the processes used in all software development projects.

The ideas that quality attributes influence the shape of the architecture and that the architecture is fundamental to the system emerged from SEI work in Rate Monotonic Analysis (RMA) in the early 1990s. From work on the RMA, SEI gained the insight of considering system structure using an analytical framework. By analogy, SEI researchers realized that such a framework could be applied to quality attributes.

Developing systematic ways to relate the software quality attributes of a system to the system's architecture provides a sound basis for making objective decisions about design tradeoffs. It also enables engineers to make reasonably accurate predictions about a system's attributes that are free from bias and hidden assumptions.

SEI researchers tested and validated this insight into the primacy of quality attributes through conducting architecture evaluations. Whether they were evaluating a financial system or an avionics system, or any other system, they succeeded in finding risks by evaluating the systems from the point of view of different quality attributes. A lasting influence of the SEI work in the field of software architecture and software development can be seen in the pervasive attention paid to quality attributes and a general acknowledgment that requirement specifications need to include them.

2003

Standardizing More Secure Software

Software vulnerabilities expose the DoD, other federal agencies, our nation’s critical infrastructure, and businesses to attacks that could compromise their systems’ integrity or modify their critical information. Preventing the introduction of software vulnerabilities during software development is a proactive, efficient way to reduce risk before the software is deployed.

Since forming its Secure Coding Initiative in 2003, the SEI CERT Division has analyzed and cataloged thousands of software vulnerabilities and discovered that many share the same common errors. By engaging more than a thousand security researchers, language experts, and software developers, CERT Division produced secure coding standards for common software development languages such as C and Java. These standards guide programmers to avoid coding errors that lead to vulnerabilities and provide example solutions.

The U.S. military, other government agencies, and system developers from industry have adopted CERT Division secure coding standards, and Siemens and Computer Associates have licensed the SEI’s training courses on secure coding in C and C++. Programmers and others in military, government, and industry organizations have taken the SEI courses.

In addition, courses based on the CERT Division standards for C and C++ are taught at major software engineering universities and colleges, such as Carnegie Mellon University, Purdue, Stevens Institute, University of Florida, and Santa Clara University.

Finally, through its security contributions to the ISO/IEC C-language specification, the CERT Division is also influencing developers of C language compilers, who conform their code to the ISO/IEC C-Standard and thus to countless software products written in the C language.

2002

Tailoring Risk Management Practice

SEI research in the 1990s produced standards for software risk management, enabling managers in all types of software-relevant programs to do a better job of identifying what could go wrong and mitigating the worst of those risks.

In 1996, SEI produced the *Continuous Risk Management Guidebook* [Dorofee 1996], which brought together several concepts developed through work with DoD agencies and Service branches in the preceding five years. This approach had widespread influence. A Cutter Consortium's report a few years later, *The State of Risk Management 2002*, revealed that 21% of respondents to a survey about risk management techniques said that they used SEI standards for risk management. Only ISO ranked higher, with 36% of respondents.

In the decades since it published the *Guidebook*, the SEI has continued to conduct research and development in various aspects of risk management. In 1998, SEI CERT researchers began developing a new approach for managing cybersecurity risks within an organization, the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) [Alberts 2003]. OCTAVE continues to be a widely used information security risk assessment method.

Other SEI-developed applications of risk management principles include the COTS Usage Risk Evaluation (CURE), the widely used Architecture Trade-off Analysis Method (ATAM), and the Mission Risk Diagnostic (MRD) (to assess risk in interactively complex, socio-technical systems across the lifecycle and supply chain).

Much of the SEI's risk management work today is focused on software assurance. SEI researchers are developing the Security Engineering Risk Analysis (SERA) method, a systematic risk-based method for building security into software-reliant systems rather than deferring security to later lifecycle activities such as operations.

2001

Setting a Foundation for Software Architecture

Safety-critical components need to interact safely with less reliable and even unsafe components. For example, the flight control component in an autopilot is certified to DO178B Level A (the highest level). However, it needs to accept guidance commands from a flight guidance system that is only certified to Level C. Nevertheless, avionics certification requires that Level A software must still function correctly in spite of the software failures in less critical components [RCTA 1992].

The SEI developed an architecture template, called the Simplex architecture, which supports overall safety when a system is composed of both reliable/safe components and less reliable/less-safe components [Sha 2001].

The Simplex architecture divides a system into two parts: (1) a complex component that cannot be fully verified but is needed to provide important service and (2) a high-assurance control subsystem that is simple and fully verified. The Simplex architecture also ensures predictable and guaranteed timing behaviors (in spite of failures of complex components) and allows restarting or replacing complex components during operation. Notable applications of Simplex architecture principles include the F-22 and F-35 aircraft.

2001

Changing Software Contractor Selection Criteria

At the beginning of the 21st century, the Capability Maturity Model Integration (CMMI) framework team published CMMI appraisal requirements, ushering in a new era for appraisals [CMMI 2001]. In partnership with government and industry, SEI published the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) [AMIT 2001], along with the specification for two other appraisal classes.

Later, the SEI developed SCAMPI B and C as a 100 percent community-funded project [Hayes 2005]. Factors that might influence an organization’s choice of a SCAMPI (A, B, or C) include cost, schedule, accuracy, efficiency, and the desired results. SCAMPI continues to have wide range of uses, including internal process improvement and external capability determinations.

The SEI’s contribution also includes creating the SCAMPI Lead Appraiser role through certification (500 as of 2013), based on the SCAMPI Lead Appraiser Body of Knowledge (SLA BOK) [Masters 2007].

SEI work on assessing/evaluating contractors led the DoD and other government acquisition organizations to change their criteria for selecting contractors. In awarding contracts, they now consider the how well the contractors’ software development practice follows the defined processes.

2000

Bringing Science to Insider Threat Mitigation

For nearly two decades, the SEI CERT Division has focused on gathering and analyzing data about actual malicious insider acts—including espionage, IT sabotage, fraud, theft of confidential information—and potential threats to U.S. critical infrastructures.

In 2001, the DoD Personnel Security Research Center (PERSEC) sponsored the first CERT Division research into the malicious actions of insiders. A few years later, the Department of Homeland Security (DHS) added its sponsorship to build a database of information on more than 150 actual insider threat cases. The database now contains more than 1,000 cases, which CERT analyzes from technical and behavioral perspectives.

Carnegie Mellon University's CyLab published the first edition of the *Common Sense Guide for Mitigating Insider Threats* in 2005, based on CERT Division research. Cylab establishes public-private partnerships to develop new technologies for measurable, secure, available, trustworthy, and sustainable computing and communications systems. Subsequent editions of the *Guide* were released in 2006, 2009, and 2012.

Applying analytical methods to the cases, the CERT Division has produced additional guidance and tools for government programs to detect, mitigate, and prevent insider threats that include

- Interactive training simulation and workshop (since 2007)
- Insider Threat Vulnerability Assessment (since 2009)
- The *CERT Guide to Insider Threats* (first published in 2012)
- Transition of linguistic analysis tools to DoD/IC customers (2015)
- Certificate programs to build skills in preventing and handling insider threats (2015)

2000

Enabling Large-Scale Network Flow Analysis

Today, network analysts in the DoD and federal agencies use SEI CERT Division network situational awareness technologies to characterize network threats, assess the impact of security events, and identify vulnerable network infrastructure.

In the early 1990s, the CERT Division developed Argus, one of the first software-based network flow analysis tools, to support incident response activity. In 2000, the Automated Incident Reporting to CERT (AirCERT) initiative released data conversion, sharing, and analysis tools (Analysis Console for Incident Data—ACID) and supported the development of Internet Engineering Task Force (IETF) standards to establish a data format for exchanging information on computer security incidents among response teams around the world.

The Einstein program, mandatory for all federal civilian agencies, integrates several distinct data collection and analysis systems and use toolsets for network traffic analysis developed by CERT. Continually through the years, the CERT Division has developed and released open source tools such as the System for Internet Level Knowledge (SiLK) tool suite, for the DoD to conduct security analysis not driven by known-bad signatures; and Yet Another Flowmeter (YAF) [Inacio 2010], which leverages additional data sources, including Domain Name System, Secure Socket Layer certificates, and application banners stored in the IPFIX standard format.

Industry has adopted these tools. Telecommunication providers, government defense contractors, and many other high tech companies use this technology to help protect their own networks and the networks of their clients.

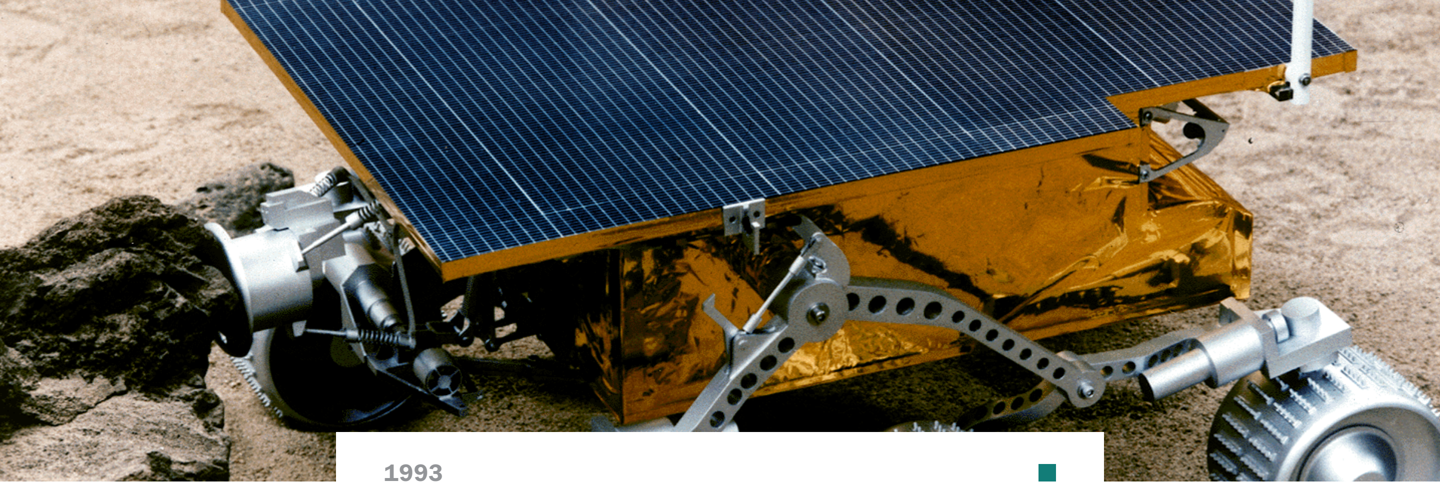
1994

Evaluating System Architecture

One recurring theme in defense challenge problems is the need to predict runtime behavior before a software-reliant system has been built. Maintenance and improvement costs represent more than half the total cost of a system, a percentage that has grown steadily since 1960 [Jones 2006]. The problem for DoD is to predict problems with modifiability before the system is constructed and before these problems occur.

SEI pioneered the use of scenario-based methods in the evaluation of software architectures for modifiability and other qualities. The first SEI-developed architecture analysis method, the Software Architecture Analysis Method (SAAM), introduced the concept of a quality attribute scenario, giving specific modifications against which the system is to be tested. The SAAM led directly to the Architectural Trade-off Analysis Method (ATAM), which evaluated a system for a collection of quality attributes.

Major defense contractors, such as Boeing and Raytheon, now include architecture evaluation in their architect certification process. The U.S. Army staff reported that use of scenario-based architecture evaluation methods reduced risk in schedule and cost, improved documentation, and resulted in a higher quality product [Nord 2009].



1993

Meeting Real-Time Scheduling Needs

Today, rate-monotonic analysis (RMA) is part of real-time computing textbooks and the only real-time scheduling technology approved by FAA for Level A avionics software in networked control applications with distributed computers, sensors, and actuators.

The importance of RMA became clear when a software bug that caused the computer on the Mars Pathfinder to reset put the 1997 mission in jeopardy. Computer scientists patched the software to fix the bug, using the rate-monotonic scheduling algorithm. Years before, the SEI had been instrumental in the development of the rate monotonic scheduling paradigm, and its technical staff played a crucial role in the development of the theory.

In 1993, the SEI published *A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time Systems*, which contains quantitative methods that enable real-time systems developers to understand, analyze, and predict the timing behavior of many real-time systems. In addition, SEI technical staff created workshops and consultation services for RMA early adopters.

SEI's and others' work in RMA transformed not only real-time engineering practice. Emerging from SEI work in RMA in the following years were two other ideas that underpin contemporary practice in software system development: (1) quality attributes influence the shape of the architecture and (2) the right architecture is fundamental to system success.

1991

Transforming Software Quality Assessment

SEI’s publication of the Software Capability Maturity Model (Software CMM) in 1991 changed the view in government and industry about software quality. The model consisted of best practices in key process areas, giving organizations an objective standard for software development.

By 1986, the DoD and defense contractors recognized that some software engineering practices produced working software with greater consistency. Unfortunately, those practices were not documented or widely recognized.

Asked to conduct a study of “best practices,” the SEI met with leading software professionals in the DoD, defense industry, commercial industry, and academia to develop a consensus on the practices that consistently lead to improved software development. To help organizations determine how well their work stacked up against these practices, the SEI produced a Maturity Questionnaire [Humphrey 1988]. Response to this questionnaire was overwhelmingly positive, from both the DoD and the defense industry.

After assisting several organizations with their assessments and subsequent improvement efforts, the SEI produced a guide for how organizations might manage that process [Fowler 1990]. As the community began to adopt these ideas, they expressed a need for a more precise definition of the practices and the underlying model. As a result, the SEI published the Software CMM.

Many people contributed to the ideas in the Software CMM, and more than a few of those ideas preceded the SEI effort. The SEI’s leadership brought software community experts and practitioners together. Its role as assimilator filtered the ideas into a consistent framework, which became a worldwide de facto standard for software process improvement. The new structure for improvement, the capability maturity model, became a seminal information architecture that has been mimicked and adapted over time.

Eventually, the Capability Maturity Model Integration (CMMI) framework, managed with software community guidance by the SEI for more than a decade, evolved from the Software CMM.

1990

Establishing a Basis for Software Reuse

Systematic software reuse is a strategy that can bring products to market or field more quickly, improve quality, and lower costs. Recently, this strategy has become more popular in the increased competitive development environment brought about by budgetary restrictions. For example, the DoD Systems Engineering FY 2014 Annual Report (issued in March 2015) notes that the CH-53K Heavy Lift Replacement Helicopter includes 7 million software lines of code with 64 percent reuse.

Underlying today's efforts to reuse software is a 1990s technology called feature-oriented domain analysis (FODA). Developed by the SEI, FODA [Kang 1990] analyzes a problem domain across multiple similar systems to identify common and variable features. In developing FODA, the SEI demonstrated that managing variation was essential to systematic software reuse and that simply identifying common elements and features is insufficient.

At the SEI, FODA later evolved into product line analysis, which extended the analysis of commonality and variability beyond features to quality attributes.

1989

Building the Master of Software Engineering Curriculum

Today, there are more than 100 accredited software engineering schools in the U.S. and about 1.5 million people work in software-development-related fields. Nearly all university software engineering-related curricula trace their lineage to SEI-led efforts.

The SEI education effort provided needed leadership during the early years of curriculum development in software engineering education. In shaping a software engineering curriculum, the SEI engaged the academic community in creating the materials and amplified technology transition with government and industry by making materials available to allow other organizations to teach material it had developed.

In the winter of 1988, the SEI held a workshop of leading software engineering educators to design a recommended curriculum for a Master of Software Engineering (MSE) degree. The SEI curriculum recommendations that grew from that workshop were published at the annual Conference on Software Engineering Education and Training (CSEE&T) [Ardis 1989], a series started by the SEI that continues today with its own independent steering committee and sponsorship.

The number of software engineering programs nearly doubled in the first three years after the publication of the guidelines. Most of those programs followed the recommended guidelines. Another outgrowth of the curriculum project was the development of materials called curriculum modules, which helped to transition the MSE curriculum and support faculty members who wished to offer software engineering courses.

In subsequent years, the SEI worked with the Association of Computing Machinery, the Institute of Electrical and Electronics Engineers, and others to influence on the quantity and quality of undergraduate software education.

1988

Pointing the way toward a Software Architecture Discipline

In studies dating back to 1978, data showed that the cost of development and modification of the user interface contributed over 50 percent of the total cost of ownership [Sutton 1978]. Attempts to reduce the cost of developing defense systems clearly had to include reduction in the cost of developing and maintaining the user interface.

The high cost of developing and modifying the user interface led to a class of systems intended to reduce this cost, user interface management systems (UIMSs). Serpent was a UIMS that separated the user interface and functional portions of a system, allowing for modifications to the user interface with minimal impact on the remainder of the system.

Through its work on Serpent, the SEI contributed to a greater understanding by a generation of user interface researchers about the impact of software engineering architectural decisions on the ease of modifying the user interface, introducing an important concept to the discipline of software architecture that emerged in the 1990s.

1988

Fostering Growth in Professional Cyber Incident Management

The SEI CERT Coordination Center (CERT/CC) was born from newfound national concern about malicious attacks on communications networks. Graduate student Robert Morris jarred the network-connected world from ambivalence regarding cybersecurity on November 2, 1988, by releasing a worm that brought the nascent internet to its knees [Marsan 2008].

In the aftermath of the Morris Worm attack, DARPA asked the SEI to establish a computer emergency response team, which has come to be known as the CERT/CC. As a neutral third party, the CERT/CC reports vulnerabilities to vendors without revealing the identity of the reporter. This position allows CERT/CC to work with competing vendors whose products contain the same vulnerability, free of conflict of interest.

Since its formation, the CERT/CC has facilitated mitigation vulnerabilities and disseminated the information through the publication of Vulnerability Notes, which include summaries, technical details, remediation information, and lists of affected vendors. CERT/CC maintains a knowledgebase of that includes a publicly available Vulnerability Notes Database.

In addition, the organization has been instrumental in building and supporting a network of more than 50 national computer security incident response teams (CSIRTs), with tools and training to help managers, project leaders, CSIRT staff, and computer forensic professionals.

References

[Alberts 2003] Alberts, Christopher & Dorofee Audrey. *Managing Information Security Risks: The OCTAVE Approach.* Addison-Wesley Professional, 2003 (ISBN 03211188630).

[AMIT 2001] Members of the Assessment Method Integrated Team. *Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document* (CMU/SEI-2001-HB-001). Software Engineering Institute, Carnegie Mellon University, 2001. resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5325

[Ardis 1989] Ardis, M. & Ford, G. "SEI Report on Graduate Software Engineering Education." *Proceedings of the Third SEI Conference on Software Engineering Education. CSEE*, Pittsburgh, PA, July 18-21, 1989. Published as Springer Lecture Notes in Computer Science 376, 1989.

[CMMI 2001] CMMI Product Team. *Appraisal Requirements for CMMI, Version 1.1 (ARC, V1.1)* (CMU/SEI-2001-TR-034). Software Engineering Institute, Carnegie Mellon University, 2001. resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5781

[Dorofee 1996] Dorofee, Audrey et al. *Continuous Risk Management Guidebook*. Software Engineering Institute, Carnegie Mellon University, 1996. resources.sei.cmu.edu/library/asset-view.cfm?assetID=30856

[Fowler 1990] Fowler, Priscilla & Rifkin, Stanley. *Software Engineering Process Group Guide* (CMU/SEI-90-TR-024). Software Engineering Institute, Carnegie Mellon University, 1990. resources.sei.cmu.edu/library/asset-view.cfm?AssetID=11253

[Hayes 2005] Hayes, William et al. *Handbook for Conducting Standard CMMI Appraisal Method for Process Improvement* (SCAMPI) B and C Appraisals, Version 1.1 (CMU/SEI-2005-HB-005). Software Engineering Institute, Carnegie Mellon University, 2005. resources.sei.cmu.edu/library/asset-view.cfm?AssetID=7281

[Humphrey 1988] Humphrey, Watts et al. *A Method for Assessing the Software Engineering Capability of Contractors* (CMU/SEI-87-TR-023). Software Engineering Institute, Carnegie Mellon University, 1988. resources.sei.cmu.edu/library/asset-view.cfm?AssetID=10345

[Inacio 2010] Inacio, Chris & Trammell, Brian, "YAF: Yet Another Flowmeter" *Proceedings of Large Information System Administration Workshop (LISA)* https://www.usenix.org/legacy/events/lisa10/tech/slides/inacio.pdf

[Jones 2006] Jones, C. "The Economics of Software Maintenance in the Twenty First Century." 2006. compaid.com/caiinternet/ezine/capersjones-maintenance.pdf

[Kang 1990] Kang, Kyo et al. *Feature-Oriented Domain Analysis (FODA) Feasibility Study* (CMU/SEI-90-TR-021). Software Engineering Institute, Carnegie Mellon University, 1990. resources.sei.cmu.edu/library/asset-view.cfm?AssetID=11231

[Marsan 2008] Marsan, Carolyn Duffy. "Morris Worm turns 20: Look what it's done" *Network World*. networkworld.com/news/2008/103008-morris-worm.html (October 30, 2008).

[Masters 2007] Masters, Steve et al. *SCAMPI Lead Appraiser Body of Knowledge (SLA BOK)* (CMU/SEI-2007-TR-019). Software Engineering Institute, Carnegie Mellon University, 2007. resources.sei.cmu.edu/library/asset-view.cfm?AssetID=8455

[Nord 2009] Nord, Robert et al. *Impact of Army Architecture Evaluations* (CMU/SEI-2009-SR-007). Software Engineering Institute, Carnegie Mellon University, 2009. resources.sei.cmu.edu/library/asset-view.cfm?assetid=8859

[RCTA 1992] RCTA, Inc. *Software Considerations in Airborne Systems and Equipment Certification. (DO-178B)*, December 1, 1992.

[SEI 2009] Software Engineering Institute. “Army Requires PEOs to Appoint Chief Software Architect.” *2009 Year in Review*. Software Engineering Institute, Carnegie Mellon University, 2010.

[Sha 2001] Sha, L. “Using Simplicity to Control Complexity.” *IEEE Software* (July/August 2001): 20-28.

[Sutton 1978] Sutton, Jimmy A & Sprague, Ralph H., Jr. *A Study of Display Generation and Management in Interactive Business Applications* (Technical Report RJ2392), IBM Research, November 1978.

[Weinstock 2009] Weinstock, Charles & Goodenough, John. *Towards an Assurance Case Practice for Medical Devices* (CMU/SEI-2009-TN-018). Software Engineering Institute, Carnegie Mellon University, 2009. resources.sei.cmu.edu/library/asset-view.cfm?AssetID=8999

Copyright

Locations

SEI Pittsburgh, PA
4500 Fifth Avenue Pittsburgh,
PA 15213-2612

SEI Arlington, VA
Suite 200
4301 Wilson Boulevard
Arlington, VA 22203

SEI Los Angeles, CA
2401 East El Segundo Boulevard
El Segundo, CA 90245

Contact Us

Phone:	412.268.5800 | 888.201.4479
Web:	sei.cmu.edu | cert.org
Email:	info@sei.cmu.edu