Software Solutions Symposium 2017

March 20–23, 2017

Secure Software Workforce Development Panel Discussion

Girish Seshagiri
Nancy Mead
William Newhouse
James Over

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Agenda

Introduction

- Community Initiative Center of Excellence for Secure Software (CICESS)
- Software Assurance Curriculum Project
- The NICE Workforce Framework & Software Development
- Software Quality and Security

Discussion

Software Solutions Symposium 2017

Secure Software Workforce Development Panel Discussion

Community Initiative Center of Excellence for Secure Software

Girish Seshagiri
Executive Vice President | CTO
Ishpi Information Technologies, Inc.

Why We Are Here

Vanishing middle class jobs

High youth unemployment and large number of under-employed

Student debt > 1.0 trillion

1.5 million cybersecurity jobs currently unfilled

Increasing number of cyber attacks against critical infrastructure

Workforce capable of developing software which is secure from cyber attacks

Apprenticeships for skill building and talent pipeline

Takeaways

Defective software is insecure

Sense of urgency to solve cybersecurity skills gap and create hundreds of thousands of middle class jobs

Leverage and build upon existing resources – software assurance curriculum, cybersecurity workforce framework, high maturity processes for use by individual programmers and agile teams

Connect education directly to a job through a dual learn and earn registered apprenticeship program

Develop skilled workforce based on validated competencies and industry standard certifications

Apprenticeships are good for business with positive return on investment

Cybersecurity Workforce Demand

1.5 Million

MORE cybersecurity professionals will be needed to accommodate the predicted global shortfall by 2020

Source: (ISC)² 2015 Global Information Security Workforce Study

On average, 52%

of IT professionals surveyed stated fewer than **25%** of all applicants were qualified

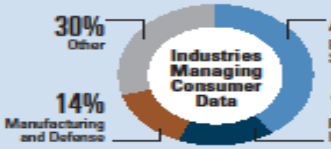
Source: State of Cybersecurity: Implications for 2015: An ISACA and RSA Conference Survey

The biggest skill gaps of today's cybersecurity professionals

- 72% Ability to Understand the Business
- 46% Technical Skills
- 42% Communication Skills

Source: State of Cybersecurity: Implications for 2015 An ISACA and RSA Conference Survey

Fastest cybersecurity demand sectors are in industries managing consumer data



Source: Job Market Intelligence: Cybersecurity Jobs, 2015–2016 Burning Glass Technologies

Cybersecurity

job postings took **8%** longer to fill than IT job postings overall

Source: (ISC)² 2015 Global Information Security Workforce Study

Expertise required for various cybersecurity roles in demand

- Information Security
- Network Setup
- Auditing
- Network Protocols
- Core Database, Coding and Scripting
- Systems Administration

Source: Job Market Intelligence: Cybersecurity Jobs, 2015

Approximately 10%

of the current cybersecurity workforce are comprised of women

Source: (ISC)² 2015 Women in Security: Wisely Positioned for the Future of InfoSec

18% Growth

Computer and mathematical occupations will grow much faster than the average job during 2012–2024

Source: Bureau of Labor Statistics, U.S. Department of Labor

Fastest growing skills in cybersecurity job postings

- Python
- HIPAA
- Risk Management
- Internal Auditing
- Audit Planning

Source: Partnership for Public Service

Hardest to fill skills in cybersecurity job postings

Source: Job Market Intelligence: Cybersecurity Jobs, 2015–2016 Burning Glass Technologies

- Software Architecture
- Network Attached Storage (NAS)
- Software Issue Resolution
- Internet Security
- Legal Compliance
- Data Communications
- Platform as a Service (PaaS)
- Computer Forensics
- Internal Auditing
- Apache Hadoop

NICE NATIONAL INITIATIVE FOR CYBERSECURITY EDUCATION

nist.gov/nice

NIST National Institute of Standards and Technology U.S. Department of Commerce

Cost of Status Quo

Item	Cost
Two-year and four-year college	$400 billion per year
Workforce education and on-the-job training	$600 billion per year
Skills gap	$160 billion per year
Time for new employees to reach full productivity	> Five months on average
Replacing an employee	Ranges from 6 to 24 months of the position's salary

Common Benefits of Apprenticeship

Production

- Output during the apprenticeship at a reduced wage
- Higher post-apprenticeship productivity relative to similarly tenured employees
- Reduction in mistakes or errors

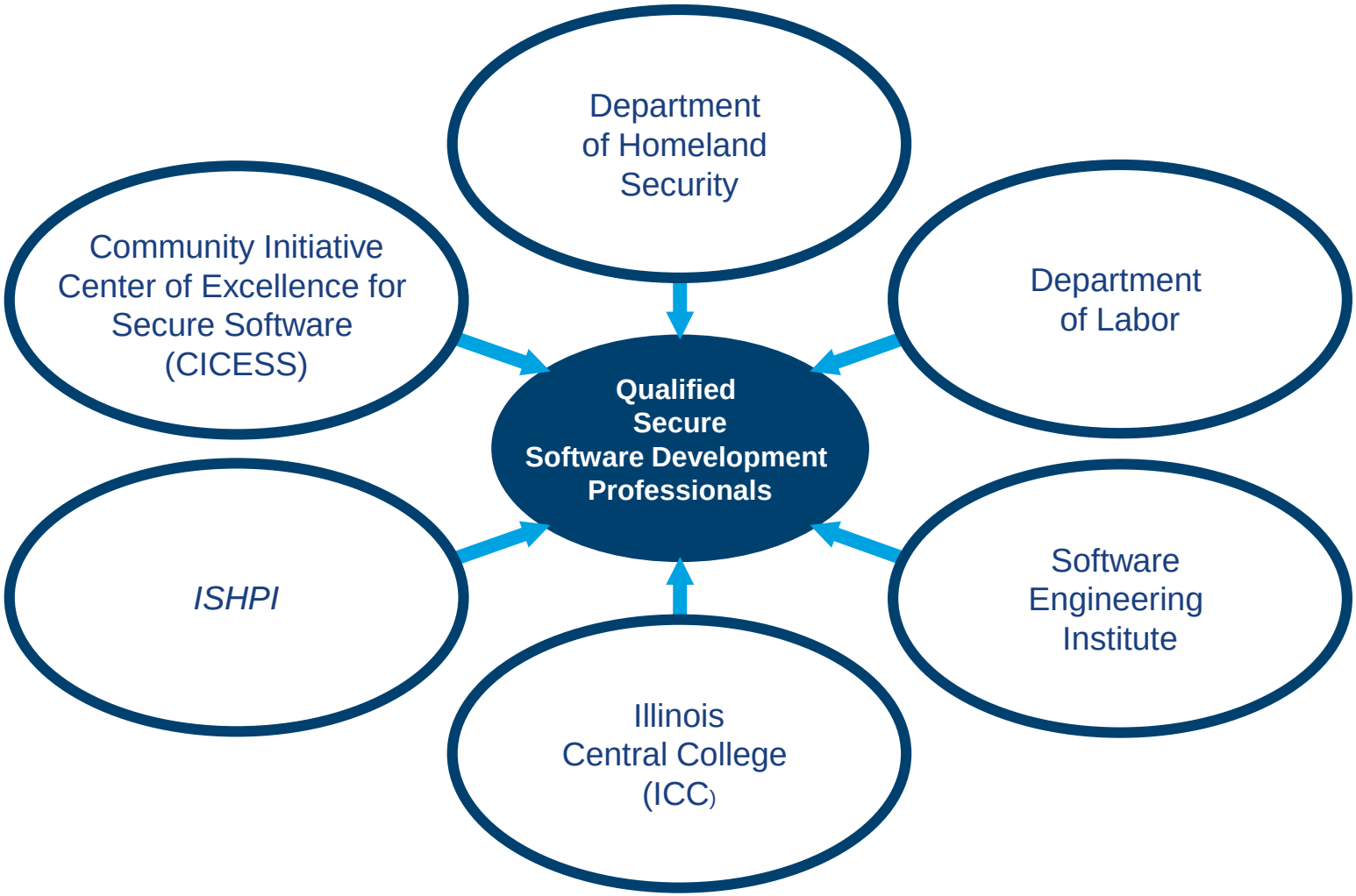
Workforce

- Reduced turnover
- Pipeline of skilled employees
- Better matching of employee skills and character with employer needs and firm culture
- Lower recruiting costs
- Development of future managers

Soft Skills

- Employee engagement and loyalty
- Greater problem-solving ability and adaptability
- Reduced need for supervision

A Unique Collaboration – Industry, Government, Academe



Goals

One of the largest available skilled workforce for secure software

Direct connection between education and a job without accumulating debt

World-class education providers with core common standard curriculum offerings

A standard competency-based, registered, apprenticeship program with uniform guidelines

Central Illinois is the destination choice for an exciting career

A skills formation and workforce development model scalable to other occupations and other communities across the nation

The Swiss Dual Track Model

CICESS Design – 1

Standard academic curriculum leading to first-in-the-nation AAS Degree in Secure Software Development

Berger Aptitude Test (B-Apt) for Computer Programming for entry to the apprenticeship program

Standard apprenticeship curriculum based on Carnegie Mellon University Software Engineering Institute (CMU/SEI) process models

Validate secure software development competencies – $(ISC)^2$ CSSLP, SEI PSP Developer certifications

CICESS Design – 2

Alternating blocks of weeks of academic instruction and apprenticeship on-the-job training in the dual model

Recurring and one-time-only fees from participating employers for ongoing program administration, apprenticeship curriculum development, and train-the-trainer materials

Guidelines for minimum hourly wages for the apprentices with flexibility to meet varied human resources practices of participating employers

Alignment with Federal Initiatives

NIST Cybersecurity Workforce Framework

NIST National Initiative Cybersecurity Education

NSA Centers of Academic Excellence

DoL American Apprenticeship Initiative

DoL Registered Apprenticeship standards

CICESS Value Proposition

Augmentation of your current workforce development methods

Ability to plan for and satisfy future needs for hard-to-fill secure software developers

Ability to build a secure software talent pipeline that includes women and minorities who are trained, mentored, and certified

A cost-effective solution to training and retaining new workers in secure software development

High retention rates when apprentices become full-time employees

Takeaways

Sense of urgency to address unsustainable trends and exploit rare economic development opportunity to create hundreds of thousands of middle class jobs

Industry/government/academic coalition led by industry to address cybersecurity “skills gap” and talent pipeline

Connect education directly to a job through a dual learn and earn registered apprenticeship program

Develop skilled workforce based on validated competencies and industry standard certifications

Apprenticeships are good for business with positive return on investment

Software Solutions Symposium 2017

Secure Software Workforce Development Panel Discussion

Nancy R. Mead
SEI Fellow and Principal Researcher
Carnegie Mellon Software Engineering Institute

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

DM-0004332

Software Solutions Symposium 2017

Secure Software Workforce Development Panel Discussion

Software Assurance (SwA) Curriculum Project

Definition of Software Assurance

We used the following definition of software assurance:

Application of technologies and processes to achieve a required level of confidence that software systems and services function in the intended manner, are free from accidental or intentional vulnerabilities, provide security capabilities appropriate to the threat environment, and recover from intrusions and failures.

Software Assurance (SwA) Curriculum Project

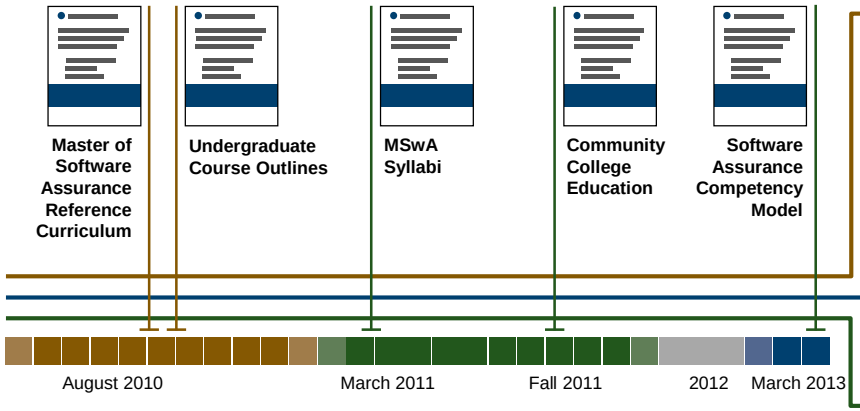
Initially sponsored by the Department of Homeland Security (DHS) National Cyber Security Division (NCSD)

Goals

- Develop software assurance curricula.
- Define transition strategies for future implementation.

Timeline

Goals: Develop software assurance curricula
Define transition strategies for implementation



Community Outreach
- 20+ published papers
- 7 SEI reports
- 20+ talks, webinars, podcasts, media
- Thousands of downloads
- LinkedIn group of 500+ members
- Course materials and videos

Integrated Into Course Offerings
- Carnegie Mellon University
- Stevens Institute of Technology
- US Air Force Academy
- University of Detroit Mercy
- University of Houston
- $(ISC)^2$

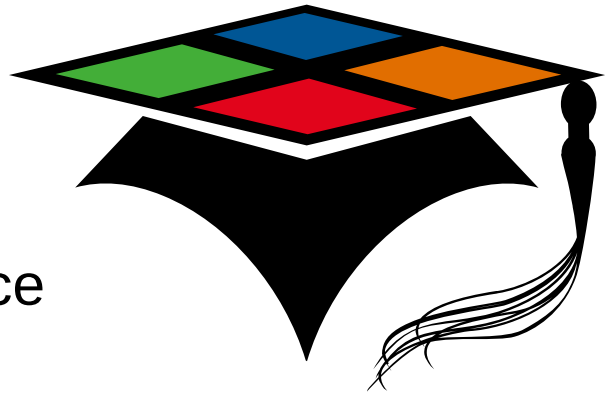
Transition
- Degree offerings
  - Polytechnic U. of Madrid– full MSwA Fall 2014
  - Illinois Central College Community College Program Fall 2015
- SwA Courses
  - Assurance Management
  - Assured Software Development 1
  - exec course
  - practitioner version of ASD1
- SEI SwA Certificate Program
  - under development in 2017

Professional Society Recognition

IEEE acm

Objectives

Improve the state of software assurance education.



- Develop a Master of Software Assurance Reference Curriculum (Volume I).

  Identify educational offerings at other levels:

  - undergraduate (Volume II)
  - MSwA syllabi (Volume III)
  - Community College (Volume IV)
  - integration with IS curricula (SEI report)
  - SwA Competency Model

Audiences

Faculty responsible for

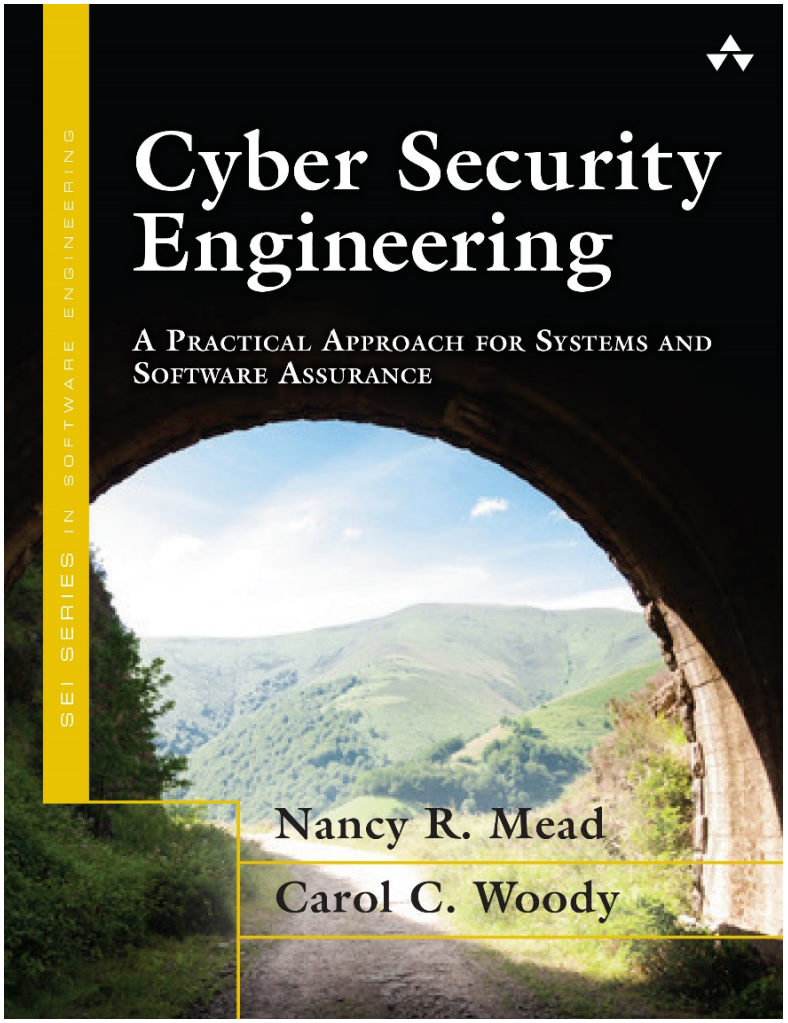
- design, development, and maintenance of degree programs focusing on software assurance knowledge and practices

Those in development and acquisition organizations responsible for either

- staffing positions in software assurance
- providing current software engineers with increased software assurance capabilities

Those who assess software assurance oriented programs

Book Published in Late 2016



Search for the book on informit.com.

Software Solutions Symposium 2017

Secure Software Workforce Development Panel Discussion

Community College Program

SwA Curriculum Community College Recommendations

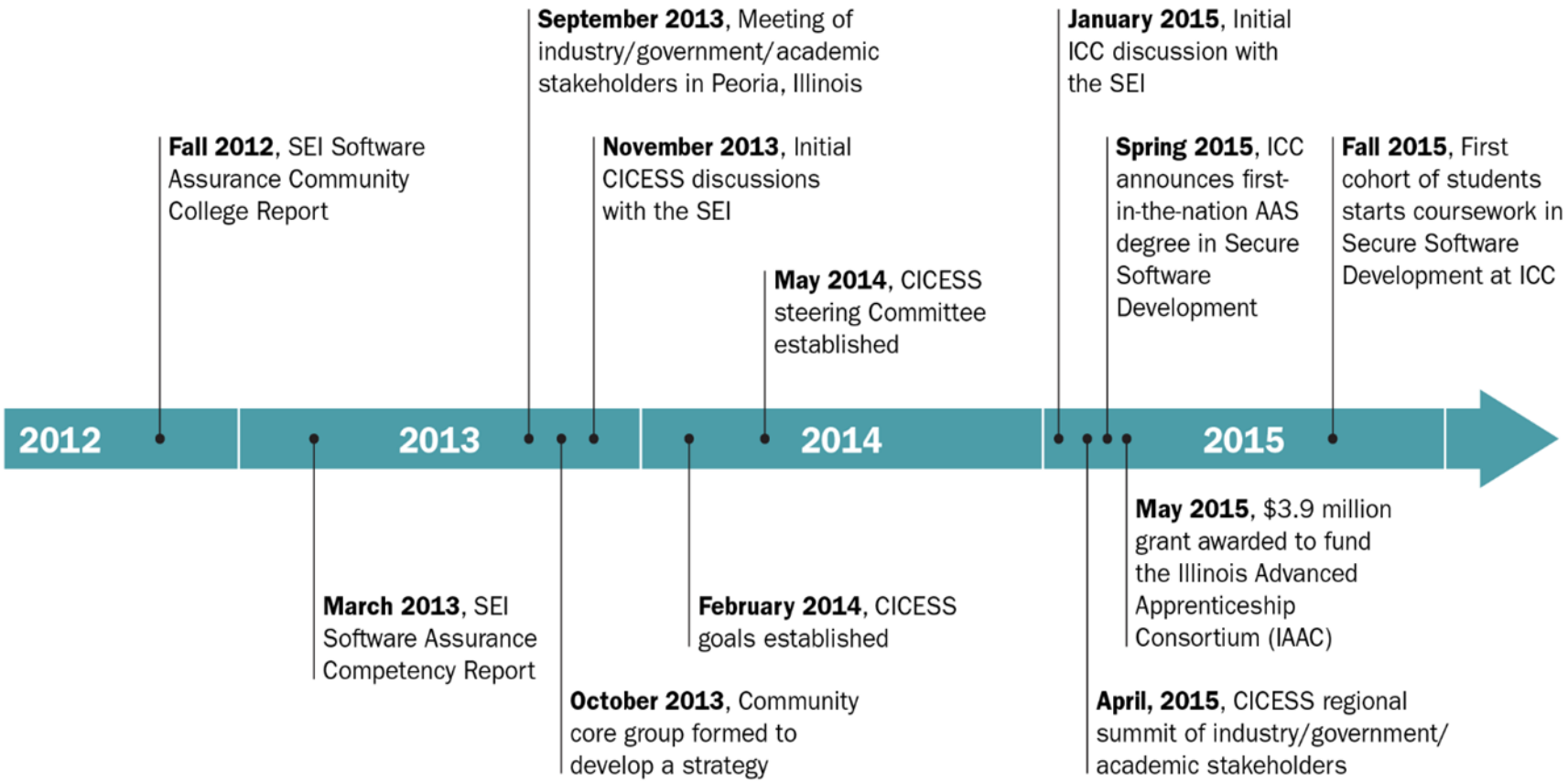
Volume IV in the SwA Education Curriculum Reports

- Modifications to Computer Science I, II, and III
- Additional recommended courses
  - Introduction to Computer Security
  - Secure Coding
  - Introduction to Assured Software Engineering

SwA Curriculum webpage

- www.cert.org/curricula/software-assurance-curriculum.cfm

Timeline for the Community College and Industry Apprenticeship Program



Illinois Central College Implementation

CS I: Programming in Java

CS II: Programming in Java

CS III: Advanced Programming in Java

Structured Query Language

Introduction to Relational Database

C# Programming

Mobile Application Programming

Introduction to Computer Security

Secure Coding

Introduction to Assured Software Engineering

Database Administration

Structured System Analysis

Two electives in computer programming, web, or networking, depending on employer needs

General education courses (19 credit hours)

Contact Information

Nancy R. Mead
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213

nrm@sei.cmu.edu

www.sei.cmu.edu/about/people/profile.cfm?id=mead_13121

Software Solutions Symposium 2017

Secure Software Workforce Development Panel Discussion

The NICE Workforce Framework & Software Development

Bill Newhouse

Deputy Director, National Initiative for Cybersecurity Education, NIST

National Initiative for Cybersecurity Education (NICE)

- The NICE strategic plan http://csrc.nist.gov/nice/about/strategicplan.html
- The NICE Cybersecurity Workforce Framework http://csrc.nist.gov/publications/PubsDrafts.html#SP-800-181

Resources

- The NICE Working Group and subgroups (K-12, Collegiate, Competitions, Training and Certifications, and Workforce Management) http://csrc.nist.gov/nice/nicewg/index.html
  - Forum to identify and share best practices that help us as a nation make progress towards the NICE Strategic goals and objectives.
- NICE provide a grant to support the creation of Cyberseek http://cyberseek.org/
- NICE provided grants for the creation of 5 Regional Alliances and Multistakeholder Partnerships to Stimulate (RAMPS)

NICE Strategic Goals

Accelerate Learning and Skills Development

- *Inspire a sense of urgency in both the public and private sectors to address the shortage of skilled cybersecurity workers*

Nurture A Diverse Learning Community

- *Strengthen education and training across the ecosystem to emphasize learning, measure outcomes, and diversify the cybersecurity workforce*

Guide Career Development & Workforce Planning

- *Support employers to address market demands and enhance recruitment, hiring, development, and retention of cybersecurity talent*

NICE Strategic Goal #3: Guide Career Development and Workforce Planning

Support employers to address market demands and enhance recruitment, hiring, development, and retention of cybersecurity talent

Objectives:

3.1 Identify and analyze data sources that support projecting present and future demand and supply of qualified cybersecurity workers

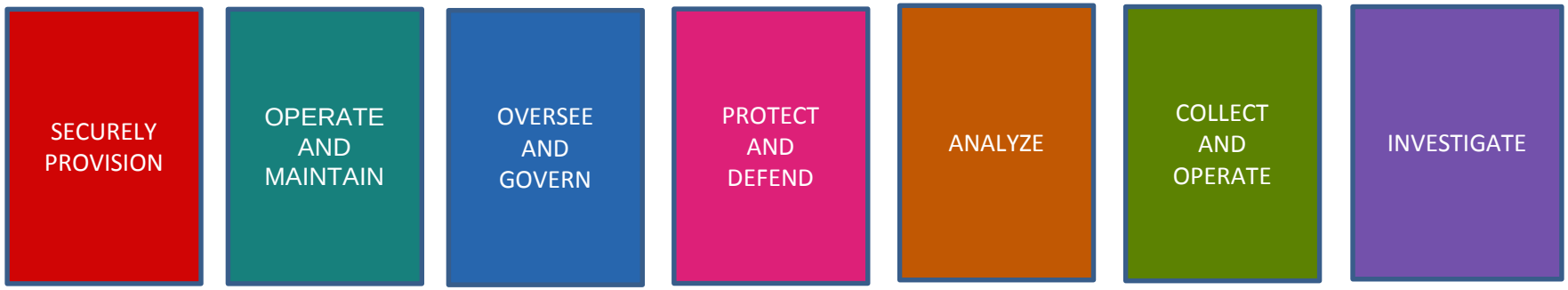
3.2 Publish and raise awareness of the NICE Cybersecurity Workforce Framework and encourage adoption

3.3 Facilitate state and regional consortia to identify cybersecurity pathways addressing local workforce needs

3.4 Promote tools that assist human resource professionals and hiring managers with recruitment, hiring, development, and retention of cybersecurity professionals

3.5 Collaborate internationally to share best practices in cybersecurity career development and workforce planning

Cybersecurity Work Roles



- Specialty Areas (33) – Distinct areas of cybersecurity work;
  - Work Roles (52) – The most detailed groupings of IT, cybersecurity or cyber-related work, which include specific knowledge, skills, and abilities required to perform a set of tasks.
    - Tasks – Specific work activities that could be assigned to a professional working in one of the NCWF's Work Roles; and,
    - Knowledge, Skills, and Abilities (KSAs) – Attributes required to perform Tasks, generally demonstrated through relevant experience or performance-based education and training.

- Audience:
  - Employers
  - Current and Future Cybersecurity Workers
  - Training and Certification Providers
  - Education Providers
  - Technology Providers

Securely Provision (7 Specialty Areas, 11 Work Roles)

Category	Specialty Area	Work Role
Securely Provision	Risk Management	Authorizing Official/Designating Representative
		Security Control Assessor
	Software Development	Software Developer
		Secure Software Assessor
	Systems Architecture	Enterprise Architect
		Security Architect
	Technology R&D	Research & Development Specialist
	Systems Requirements Planning	Systems Requirements Planner
	Test and Evaluation	Testing and Evaluation Specialist
	Systems Development	Information Systems Security Developer
		Systems Developer

Software Development Specialty Area

Software Development (DEV)	Develops and writes/codes new (or modifies existing) computer applications, software, or specialized utility programs following software assurance best practices.	Software Developer (SP-DEV-001)	Develops, creates, maintains, and writes/codes new (or modifies existing) computer applications, software, or specialized utility programs.	621	Click to view KSAs	Click to view Tasks	
		Secure Software Assessor (SP-DEV-002)	Analyzes the security of new or existing computer applications, software, or specialized utility programs and provides actionable results.	622	Click to view KSAs	Click to view Tasks	

Securely Provision (SP) Software Development (DEV)	Software Developer (621): Develops, creates, maintains, and writes/codes new (or modifies existing) computer applications, software, or specialized utility programs.
Knowledge	
K0001	* Knowledge of computer networking concepts and protocols, and network security methodologies.
K0002	* Knowledge of risk management processes (e.g., methods for assessing and mitigating risk).
K0003	* Knowledge of national and international laws, regulations, policies, and ethics as they relate to cybersecurity.
K0004	* Knowledge of cybersecurity principles.
K0005	* Knowledge of cyber threats and vulnerabilities.
K0006	* Knowledge of specific operational impacts of cybersecurity lapses.
K0014	Knowledge of complex data structures.
K0016	Knowledge of computer programming principles such as object-oriented design.
K0027	Knowledge of organization's enterprise information security architecture system.
K0028	Knowledge of organization's evaluation and validation requirements.
K0039	Knowledge of cybersecurity principles and methods that apply to software development.
K0044	Knowledge of cybersecurity principles and organizational requirements (relevant to confidentiality, integrity, availability, authentication, non-repudiation).
K0051	Knowledge of low-level computer languages (e.g., assembly languages).
K0060	Knowledge of operating systems.
K0066	Knowledge of Privacy Impact Assessments.
K0068	Knowledge of programming language structures and logic.
K0073	Knowledge of secure configuration management techniques.
K0079	Knowledge of software debugging principles.
K0080	Knowledge of software design tools, methods, and techniques.
K0081	Knowledge of software development models (e.g., Waterfall Model, Spiral Model).
K0082	Knowledge of software engineering.
K0084	Knowledge of structured analysis principles and methods.

Securely Provision (SP) Software Development (DEV)	Software Developer (621): Develops, creates, maintains, and writes/codes new (or modifies existing) computer applications, software, or specialized utility programs.
Knowledge	
K0085	Knowledge of system and application security threats and vulnerabilities.
K0086	Knowledge of system design tools, methods, and techniques, including automated systems analysis and design tools.
K0105	Knowledge of web services, including service-oriented architecture, Simple Object Access Protocol, and web service description language.
K0139	Knowledge of interpreted and compiled computer languages.
K0140	Knowledge of secure coding techniques.
K0152	Knowledge of software related information technology (IT) security principles and methods (e.g., modularization, layering, abstraction, data hiding, simplicity/minimization).
K0153	Knowledge of software quality assurance process.
K0154	Knowledge of supply chain risk management standards, processes, and practices.
K0170	Knowledge of local specialized system requirements (e.g., critical infrastructure systems that may not use standard information technology [IT]) for safety, performance, and reliability.
K0179	Knowledge of network security architecture concepts including topology, protocols, components, and principles (e.g., application of defense-in-depth).
K0199	Knowledge of security architecture concepts and enterprise architecture reference models (e.g., Zachman, Federal Enterprise Architecture [FEA]).
K0202	Knowledge of the application firewall concepts and functions (e.g., Single point of authentication/audit/policy enforcement, message scanning for malicious content, data anonymization for PCI and PII compliance, data loss protection scanning, accelerated cryptographic operations, SSL security, REST/JSON processing).
K0219	Knowledge of local area network (LAN) and wide area network (WAN) principles.
K0260	Knowledge of Personally Identifiable Information (PII) data security standards.
K0261	Knowledge of Payment Card Industry (PCI) data security standards.
K0262	Knowledge of Personal Health Information (PHI) data security standards.
K0263	Knowledge of information technology (IT) risk management policies, requirements, and procedures.
K0322	Knowledge of embedded systems.
K0331	Knowledge of network protocols (e.g., Transmission Critical Protocol (TCP), Internet Protocol (IP), Dynamic Host Configuration Protocol (DHCP)), and directory services (e.g., Domain Name System (DNS)).
K0342	Knowledge of penetration testing principles, tools, and techniques.
K0343	Knowledge of root cause analysis techniques.

Securely Provision (SP) Software Development (DEV)	Software Developer (621): Develops, creates, maintains, and writes/codes new (or modifies existing) computer applications, software, or specialized utility programs.
Skills	
S0001	Skill in conducting vulnerability scans and recognizing vulnerabilities in security systems.
S0014	Skill in conducting software debugging.
S0017	Skill in creating and utilizing mathematical or statistical models.
S0019	Skill in creating programs that validate and process multiple inputs including command line arguments, environmental variables, and input streams.
S0022	Skill in designing countermeasures to identified security risks.
S0031	Skill in developing and applying security system access controls.
S0034	Skill in discerning the protection needs (i.e., security controls) of information systems and networks.
S0060	Skill in writing code in a currently supported programming language (e.g., Java, C++).
S0135	Skill in secure test plan design (e. g. unit, integration, system, acceptance).
S0138	Skill in using Public-Key Infrastructure (PKI) encryption and digital signature capabilities into applications (e.g., S/MIME email, SSL traffic).
S0149	Skill in developing applications that can log and handle errors, exceptions, and application faults and logging.
S0174	Skill in using code analysis tools.
S0175	Skill in performing root cause analysis.
Abilities	
A0007	Ability to tailor code analysis for application-specific concerns.
A0021	Ability to use and understand complex mathematical concepts (e.g., discrete math).
A0047	Ability to develop secure software according to secure software deployment methodologies, tools, and practices.

NISTIR 8151

Dramatically Reducing Software Vulnerabilities

Report to the White House Office of Science and Technology Policy

Paul E. Black
Lee Badger
Barbara Guttman
Elizabeth Fong
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8151

November 2016

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Under Secretary of Commerce for Standards and Technology and Director

February 2016 Federal Cybersecurity Research and Development Strategic Plan

- This plan starts by describing a well-known risk: current systems perform increasingly vital tasks and are widely known to possess vulnerabilities.
  - These vulnerabilities are often not easy to discover and difficult to correct.
  - Cybersecurity has not kept pace, and
  - The pace that is needed is rapidly accelerating.
- The R&D Strategic plan defines goals for the near, mid and long term.
- NISTIR 8151 addresses the first mid-term goal:
  - Achieve Science and Technology advances to reverse adversaries' asymmetrical advantages, through sustainably secure systems development and operation. …
  - This goal is two-pronged: first, the design and implementation of software, firmware, and hardware that are highly resistant to malicious cyber activities (e.g., software defects, which are common, give rise to many vulnerabilities) …

The Goal of NISTIR 8115

- Present a list of specific technical approaches that have the potential to make a dramatic difference reducing vulnerabilities – by stopping them before they occur, by finding them before they are exploited or by reducing their impact.
  - Stopping vulnerabilities before they occur generally includes improved methods for specifying, designing and building software.
  - Finding vulnerability includes better testing techniques and more efficient use of multiple testing methods.
  - Reducing the impact of vulnerabilities refers to techniques to build architectures that are more resilient, so that vulnerabilities cannot be exploited for significant damage

Software Solutions Symposium 2017

Secure Software Workforce Development Panel Discussion

Defective Software Is Not Secure

James W. Over
Technical Director/Principal Engineer
Carnegie Mellon Software Engineering Institute

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

DM-0004574

Defective Software Is Not Secure

Many vulnerabilities are caused by common software defects.[1]

- buffer overflow, failure to validate input, logic errors, etc.

```
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;  /* MISTAKE! THIS LINE SHOULD NOT BE HERE */
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
    goto fail;
```

Poor quality development practices are a principal cause.

From 1-5% of defects are potential vulnerabilities.[1,2]

Practices to improve software quality are critically needed.

1. Woody, Carol; Ellison, Robert J.; Nichols, William. “Predicting Software Assurance Using Quality and Reliability Measures.” CMU/SEI-2014-TN-026
2. Alhazmi, Omar H.; Malaiya, Yashwant K.; & Ray, Indrajit. “Measuring, Analyzing and Predicting Security Vulnerabilities in Software Systems.” *Computers & Security 26*, 3 (May 2007): 219–228.

Software Quality State of the Practice

Software Defect: an engineering artifact, that if not changed, could cause improper design, implementation, test, use, or maintenance.

Defect Density: Count of the defects removed divided by product size. A measure of product quality that is related to the quality of the development process.

	Defects per KSLOC		Defects per MLOC		Est. Vulnerabilities per MLOC	
Source	Min	Max	Min	Max	Min	Max
ADA and Beyond: Software Policies for the DoD; 1997	1.0000	7.0000	1,000	7,000	10	350
Delivered Defect Density by Maturity Level; C. Jones; 2003	1.0500	7.5000	1,050	7,500	11	375
Software Engineering Best Practices; C. Jones; 2010	1.1600	5.2933	1,160	5,293	12	265
SEI TSP Data; 2014	0.00092	0.5625	1	563	0	28

Improving Software Quality

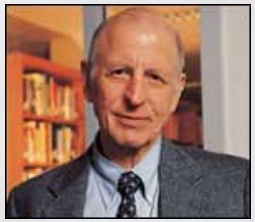
"The only way for errors to occur in a program is by being put there by the author. No other mechanisms are known. Programs can't acquire bugs by sitting around with other buggy programs." – Dr. Harlan Mills

Software development relies on testing to find and fix defects.

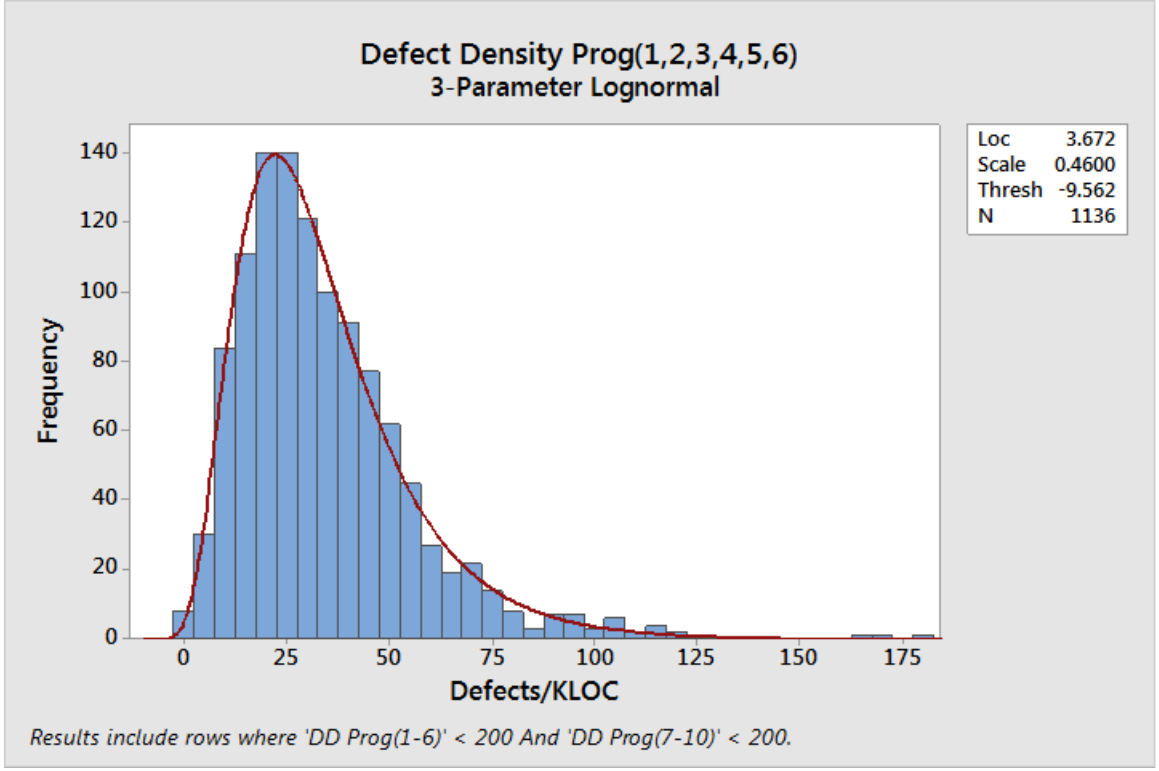
As a defect removal practice testing is

- Expensive; 30% to 60% of development costs
- Slow; 50% or more of schedule
- Ineffective; removes 80% to 85% of defects (and some vulnerabilities)

Low defect content is an essential prerequisite to a quality software process, but testing only finds a fraction of the defects. The most effective way to reduce defects is with the individual software engineer. – Watts Humphrey



Software Engineer Quality Data



Developers	1,136
Mean	34.739
Median	29.762
Std. Deviation	25.278

Defects found in test and injected during detailed design and coding

Source: PSP training data

Personal Software Process Quality Practices

Personal Software Process (PSP)

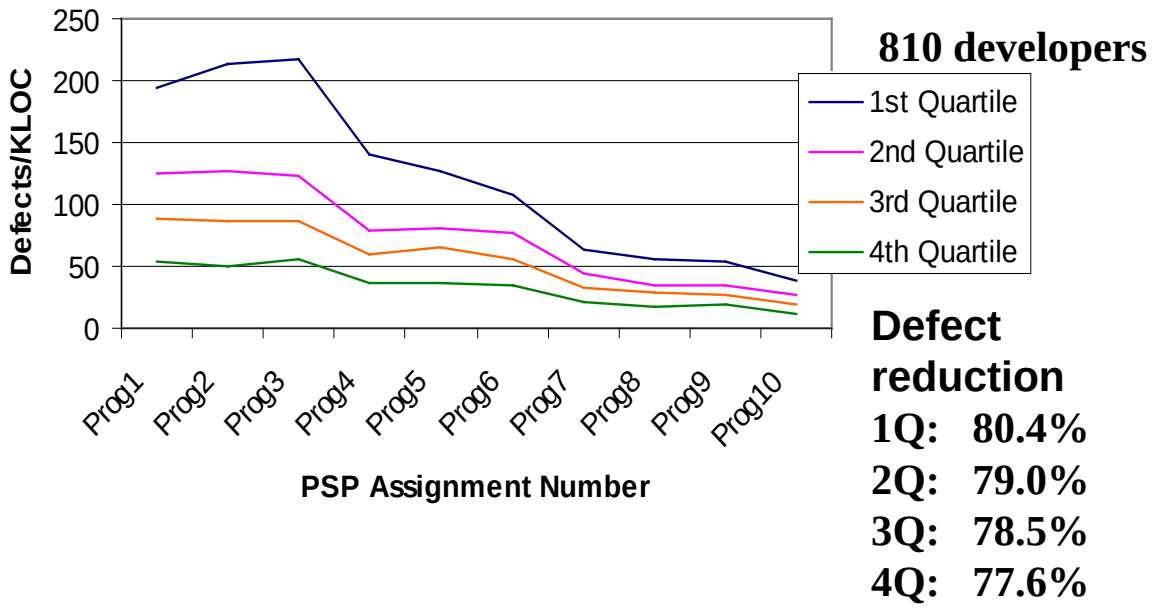
Process framework for developers.

Scaled to small projects

Developer controlled

Metrics to manage variation and speed improvement

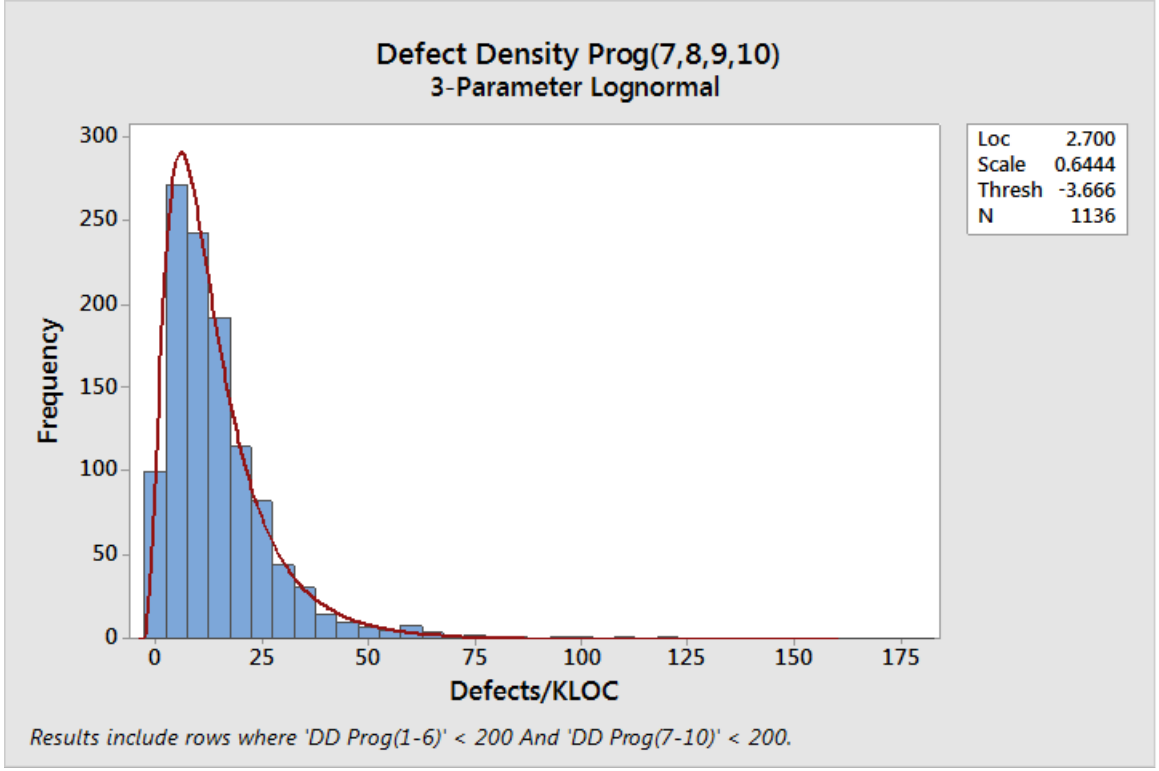
Training and certification



PSP Facts

- Training – developers write10 small programs
- PSP levels – 3; baseline, planning, quality
- Measures – size; development time; defects found
- Data – 50 data points/assignment; 10 assignments/ developer; 3383 developers as of 2016

PSP-Trained Software Engineer Quality Data



Developers	1,136
Mean	15.043
Median	11.547
StDev	16.071

Defects found in test and injected during detailed design and coding

Source: PSP training data

Xtreme Quality

Team Software Process (TSP)

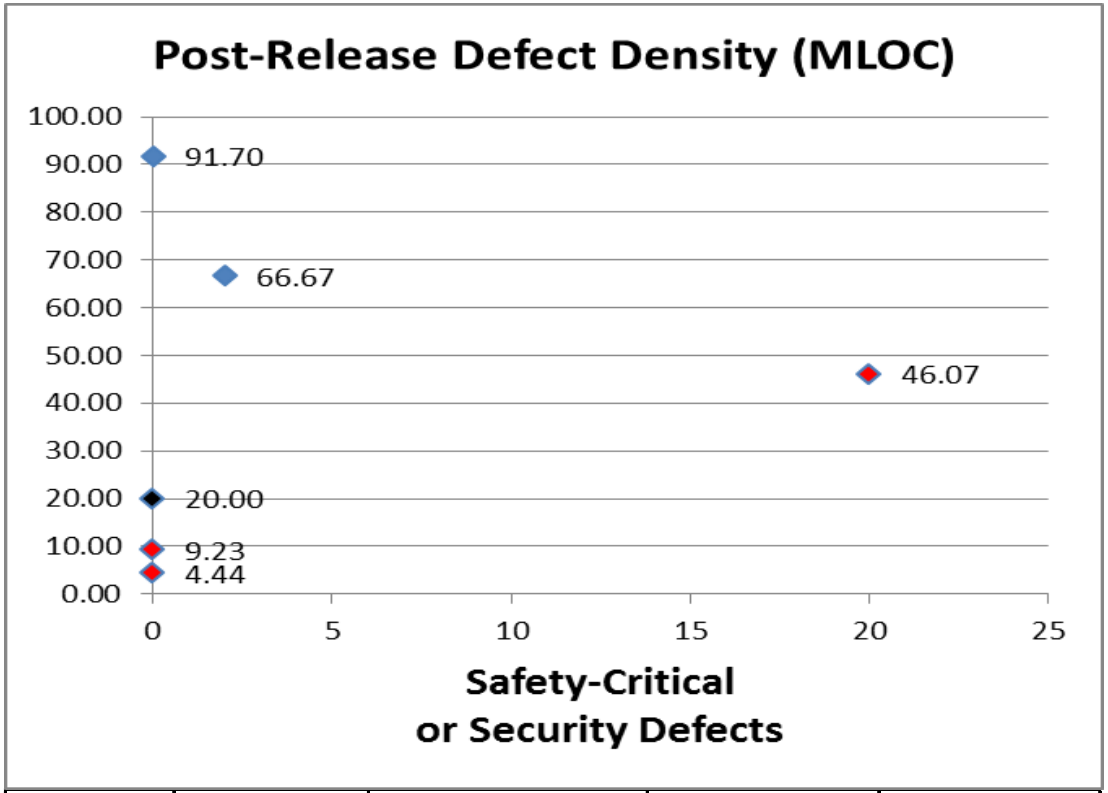
Process framework for PSP-trained developers.

Scaled to medium to large applications

Team controlled

Metrics to support team management.

Coach training and certification



Project	Type	Critical Defects in first 1+ year	Defect Density (MLOC)	Size
D1	Safety Critical	20	46.07	2.8 MLOC
D2	Safety Critical	0	4.44	.9 MLOC
D3	Safety Critical	0	9.23	1.3 MLOC
A1	Secure	0	91.70	.6 MLOC
X1	Secure	0	20.00	.1 MLOC
B1	Secure	2	66.67	.45 MLOC

A Path Forward

Improving software quality is a necessary part of the solution to software security.

The means of achieving improved quality are proven and available.

The challenges

- Educating the workforce
- Changing the behavior of software people
- Raising consumer awareness

What Government Can Do

Change industry behavior

- Software quality standards and policies that address reducing defect content in the software.
- Incentives for government software providers designed to produce continued, incremental improvements in quality.

Sponsor secure software development apprenticeship programs

- Training programs that emphasize software quality
- Hands-on demonstration of knowledge/skills transfer
- On-the-job application of knowledge/skills learned under the supervision of a qualified coach/mentor