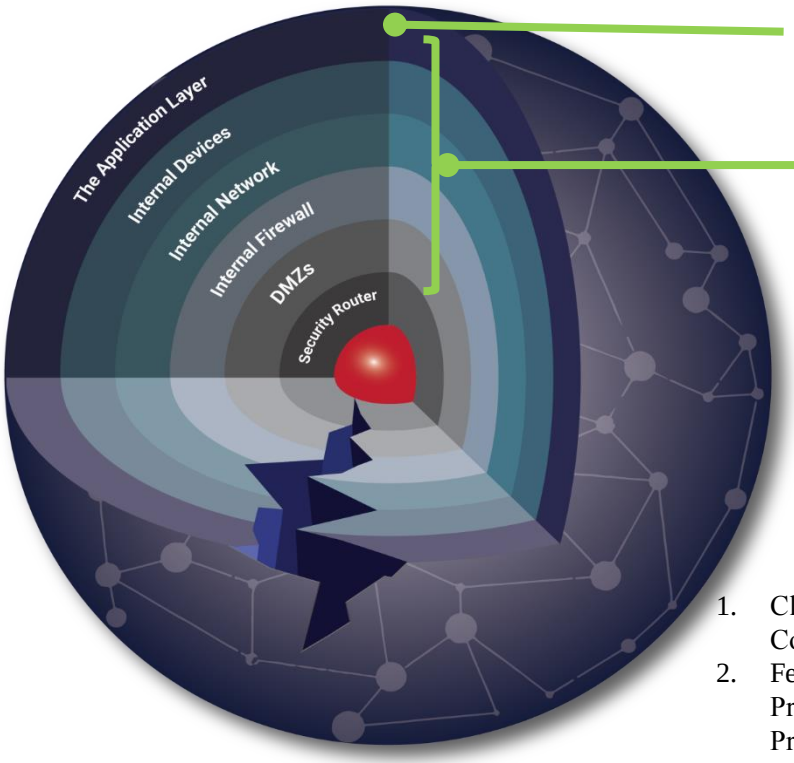
Achieving DoD Software Assurance (SwA)

Thomas Hurt
Office of the Deputy Assistant Secretary of Defense for Systems Engineering

20th Annual NDIA Systems Engineering Conference
Springfield, VA | October 26, 2017

First Line of Defense in Software Assurance Is the Application (Software) Layer

Software assurance (SwA) provides the required level of confidence that software functions as intended (and only as intended) and is free of (known) vulnerabilities, either intentionally or unintentionally designed or inserted in software, throughout the life cycle.

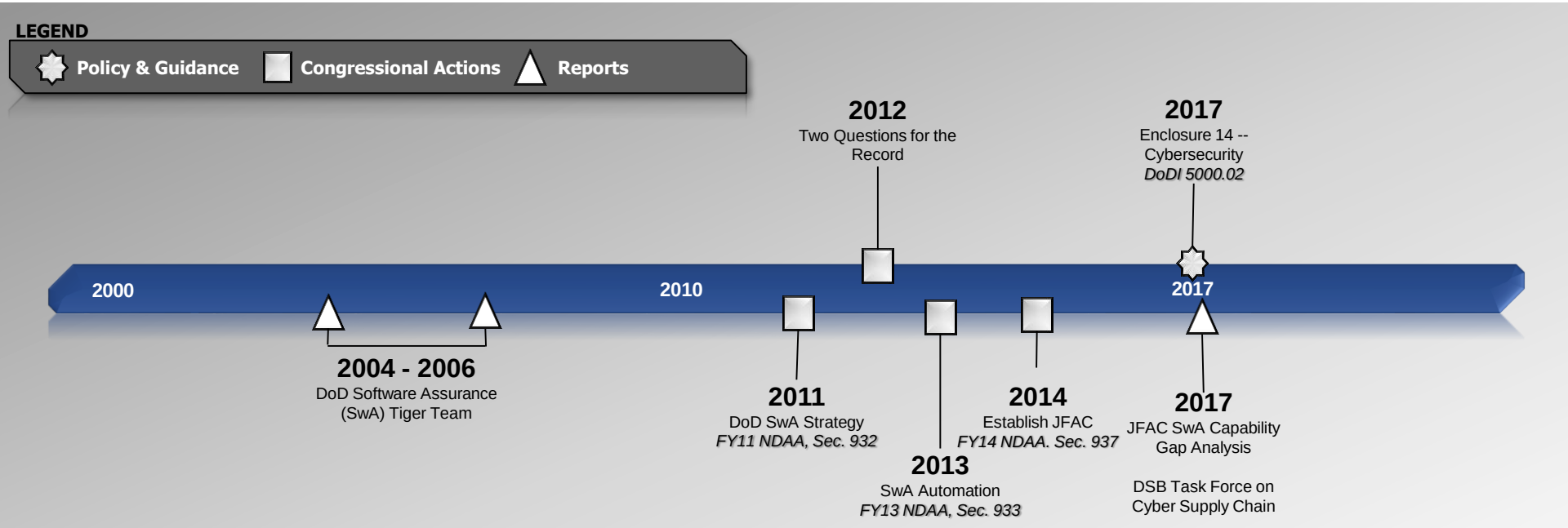


84% of breaches exploit vulnerabilities in the application[1]

Yet funding for IT defense vs. software assurance is 23 to 1[2]

1. Clark, Tim, "Most Cyber Attacks Occur from This Common Vulnerability," *Forbes*, 03-10-2015
2. Feiman, Joseph, "Maverick Research: Stop Protecting Your Apps; It's Time for Apps to Protect Themselves," *Gartner*, 09-25-2014. G00269825

How Did We Get Here?

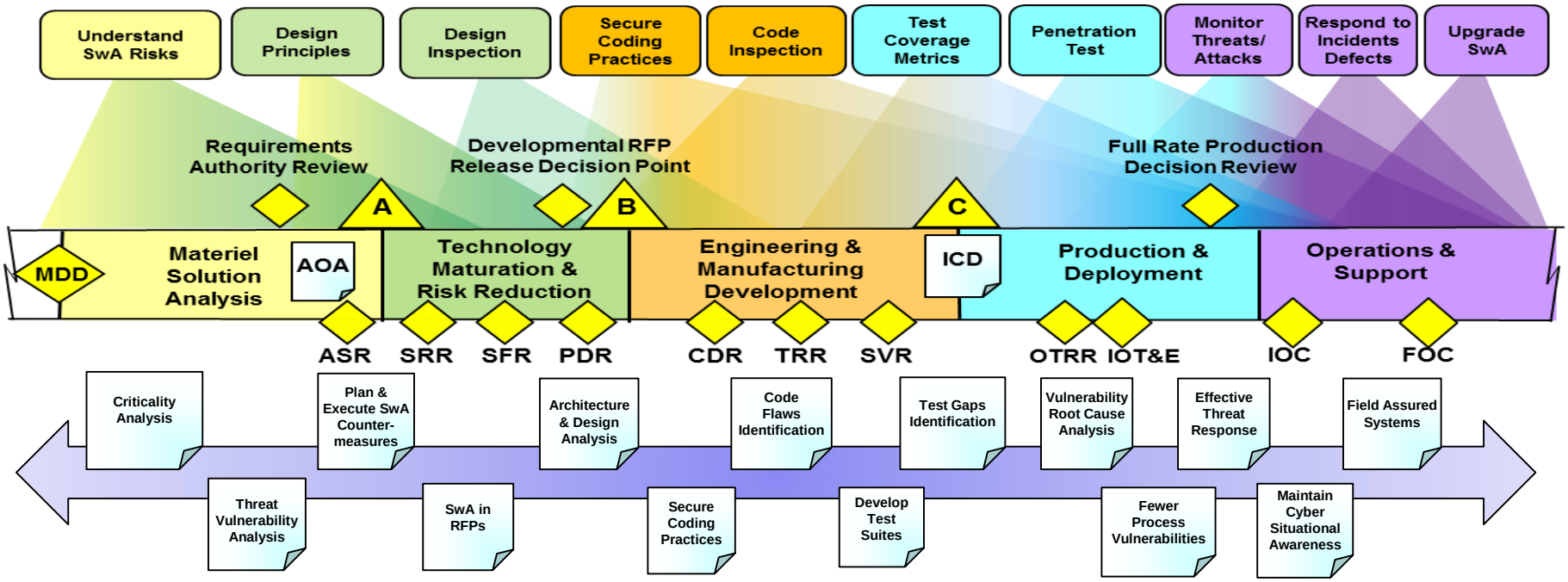


Congress and DoD have acknowledged the need for increased software assurance to improve confidence in secure and resilient weapon systems for over a decade.

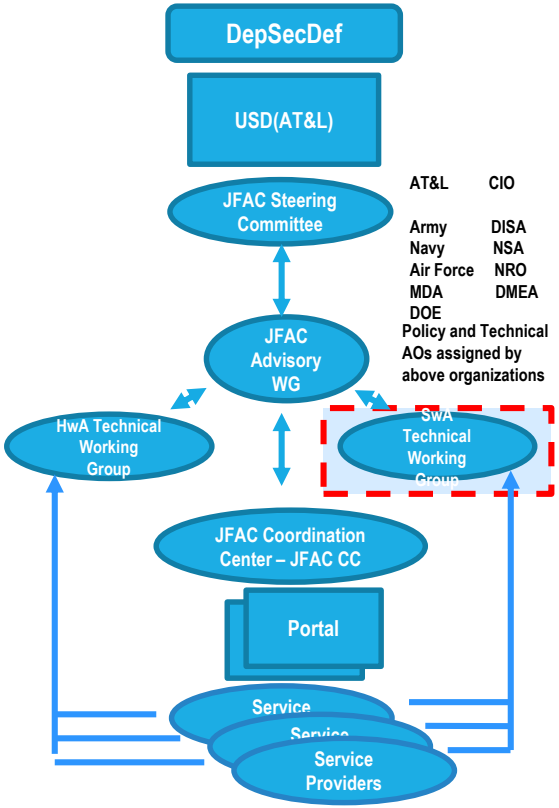
JFAC: Joint Federated Assurance Center

How to Engineer Software Assurance Across the DoD Acquisition Life Cycle

Software Assurance best practices, as a part of Systems Engineering, focus on increasing the level of confidence of software functioning as intended.



SwA within DoD



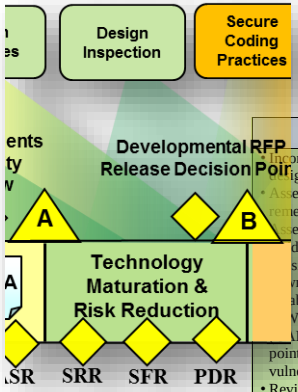
- **JFAC SwA Working Group**
  - Collaboration and shared prioritization in daily/weekly activities, meet on a regular basis
  - Recommend SwA policy and guidance
  - Provide community forum for "hard problem" analysis and question/answer
- **DoD SwA Community of Practice**
  - Tri-leads; meets quarterly with various DoD stakeholders' participation
  - Sponsors research and pilots into hard SwA problems

What's Going on Now? (1 of 3)

- **DoD Software Assurance Community of Practice**
  - Past products include: Contract language for integrating SwA; State-of-the-Art Resource (SOAR) for SW Vulnerability Detection, Test, and Evaluation; SwA metrics
  - Recent Topics and Ongoing Activities
    - SwA Risk Assessment process
    - Malware discovery in binary code
    - SwA analysis of mobile software
- **The Journal of Cyber Security and Information Systems: Design & Development Process for Assured Software–Vol 1***
  - Software Assurance in the Agile Software Development Lifecycle
  - Is Our Software REALLY Secure?
  - Development and Transition of the SEI Software Assurance Curriculum
  - Keys to Successful DoD Software Project Execution
  - Hacker 101 & Secure Coding: A Grassroots Movement toward Software Assurance

* https://www.csiac.org/journal-issue/design-and-development-process-for-assured-software-volume-1/

What's Going on Now? (2 of 3)



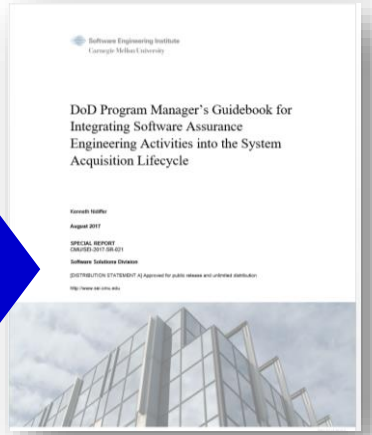
Acquisition Phase Considerations

SOFTWARE ASSURANCE CONSIDERATIONS (TMRR Phase)
• Incorporate SwA requirements, tool use, metrics, and assurance thresholds into solicitations. Architectures, designs, and code developed for prototyping are frequently reused later in development. • Assess system functional requirements and verification methods for inclusion of SwA tools remediation across the development life cycle. ssess requirements for SwA are correct and complete regarding assurance. Consider mea ders and adversaries using malicious inserts; system characteristics; interoperability wi sion threads; and other factors. Assure that mapping and traceability are maintained as wnstream assessments. ablish baseline architecture and review for weaknesses (e.g., use of Common Weakness VE)) and susceptibility to attack (e.g., use of Common Attack Pattern Enumeration and APEC)), and likelihood of attack success considering each detected weakness; identify points and mission impacts. Consider which families of automated SwA engineering tools vulnerability or weakness detection. • Review architecture and design for adherence to secure design principles and assess sound decisions considering likely means of attack; programming language choices; development frameworks; and use of open source software, etc. • Identify and mitigate technical risks through competitive prototyping while engineering in prototypes may be physical or math models and simulations that emulate expected perform concepts may require scaled models to reduce uncertainty too difficult to resolve purely by emulation. SW prototypes that reflect the results of key trade-off analyses should be demo TMRR phase. These demonstrations will provide SW performance data (e.g., latency, secu integration of legacy services, graceful function degradation and re-initiation, and scalabili decisions as to maturity; further, EMD estimates (schedule and life cycle cost) often depen components developed in TMRR; therefore to prevent technical debt, SwA considerations must have been taken into account. • Develop a comprehensive system-level architecture, then design (address function integrity, assurance of the functional breakout, function interoperation, and separation of function) that covers the full scope of the system in order to maintain capabilities across multiple releases and provide the fundamental basis to fight through cyberattack. The program focused on a given SW build/release/increment may only produce artifacts for that

Systems Engineering Technical Review Success Criteria

Objective	SwA Success Criteria
Preliminary Design Review (PDR)	
Recommendation that allocated baseline fully satisfies user requirements and developer ready to begin detailed design with acceptable risk. Allocated baseline is established such that the design provides sufficient confidence that the program demonstrates a high likelihood of accomplishing its intended mission, including in a cyber-contested environment. Preliminary design and basic system architecture support capability need and affordability target achievement.	• Determine that baseline fully satisfies user requirements, with assurance engineered in. • Determine that likely means of attack through software have been assessed and used in architecture and design implementation. • Review architecture and design against secure design principles; including system element isolation, least common mechanism, least privilege, fault isolation, input checking and validation. Consult JFAC planning tools, best practices in architecture and design, and guidance. • Determine if initial SwA Reviews and Inspections from prior SETR activities capture planning and requirements appropriately, including assurance. • Confirm that SwA requirements that were previously mapped from tactical use threads, mission threads, system requirements, and system interoperability requirements, are mapped to module test cases and to the final acceptance test cases. • Establish automated regression testing procedures and tools as a core process, and assure regression testing is conducted for remediated vulnerabilities, defects, and weaknesses.

PM's Guidebook for SwA Activities



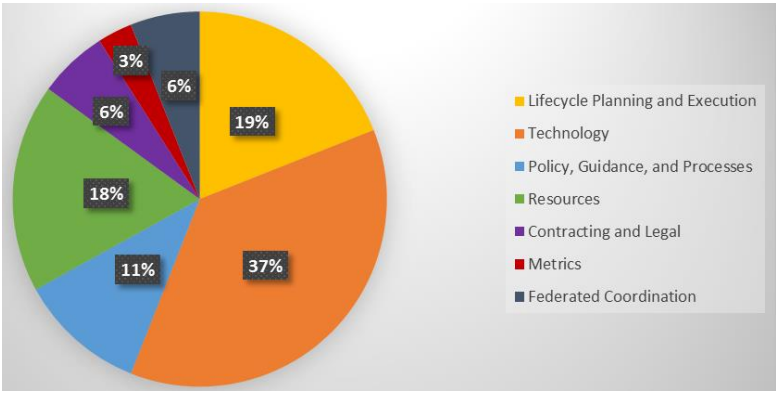
To be published by SEI.

Upcoming Journal of Cyber Security and Information Systems article:
"Engineering SwA into Weapon Systems during the DoD Acquisition Life Cycle"

What's Going on Now? (3 of 3)

In July 2016, the JFAC SwA Technical Working Group identified **63 DoD capability gaps** that prevent the effective planning and execution of software assurance within the DoD acquisition process. The gaps were organized into seven categories:

Gap Examples:
2.2.2 - SwA requirements lacking in system requirements
5.2.1 - Lack of SwA training for Program Managers
6.1 - Lack of definitive contract language for SwA planning and execution activities, as early in the lifecycle as possible

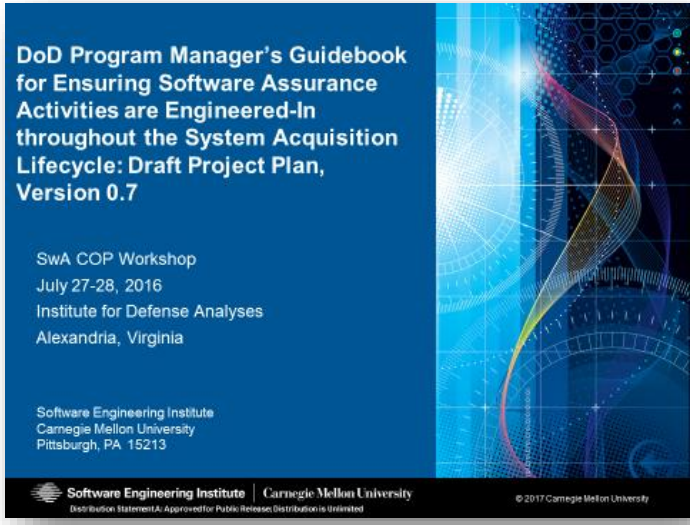


As chair of the JFAC Steering Committee, Ms. Kristen Baldwin, Acting Deputy Assistant Secretary of Defense for Systems Engineering (DASD(SE)), approved the analysis* and directed the Technical Working Group to **develop a strategy to address the identified gaps**. DASD(SE)'s JFAC lead, Mr. Tom Hurt, supported the **NDIA-sponsored joint industry-government workshop**.

*Distribution C, available upon request.

What's Next?

- **DoD Program Manager's Guidebook for Integrating Software Assurance Engineering Activities into the System Acquisition Life Cycle**
  - To be written and published by SEI in collaboration with JFAC SwA Technical WG
  - Partner Document: Software Developers Guidebook
- **DASD(SE) Activities**
  - FY18 Business Case Analysis for SwA Tools
- **JFAC website on SIPR, JWICS**
  - One-stop shop for SwA tools and best practices
  - New S&T and Assessment Knowledge Base portals
  - https://jfac.army.mil
- **Develop JFAC Full Operational Capability (FOC) strategy**
  - Improve DoD SwA throughout Lifecycle Planning, Execution and Sustainment
  - Linking Sustainment to Early Program Development



Conclusion

- **DoD has been focused on software assurance for over a dozen years.**
  - DASD(SE) leads the development and implementation of the supporting best practices, guidance, tools, and workforce competencies to ensure PMs have the means to mitigate SwA vulnerabilities and risk.

- **The JFAC's goal is to provide DoD programs a one-stop shop to request, evaluate, and obtain resources to improve their software assurance practice.**
  - SwA analysis tool license distribution and management
  - Service providers for programs' SwA work; SMEs focused on hard problems
  - SwA best practices

- **JFAC and DoD SwA COP is addressing key software assurance gaps.**
  - Developing FOC strategy to execute as resourcing becomes available

Systems Engineering: Critical to Defense Acquisition










Defense Innovation Marketplace
http://www.defenseinnovationmarketplace.mil

DASD, Systems Engineering
http://www.acq.osd.mil/se

For Additional Information

Mr. Thomas Hurt

ODASD, Systems Engineering

571-372-6129

thomas.d.hurt.civ@mail.mil