

21st Annual NDIA Systems Engineering Conference

## **Software Assurance Concept of Operations**

## Advancing Software Assurance in the Modern Age – Part 2

**Bradley Lanford, Engility Corporation** 

In Support for Office of the Under Secretary of Defense for Research and Engineering

October 2018



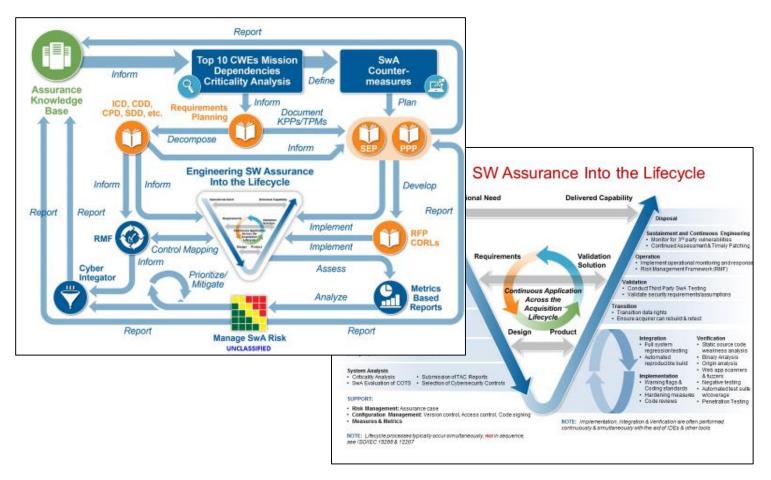




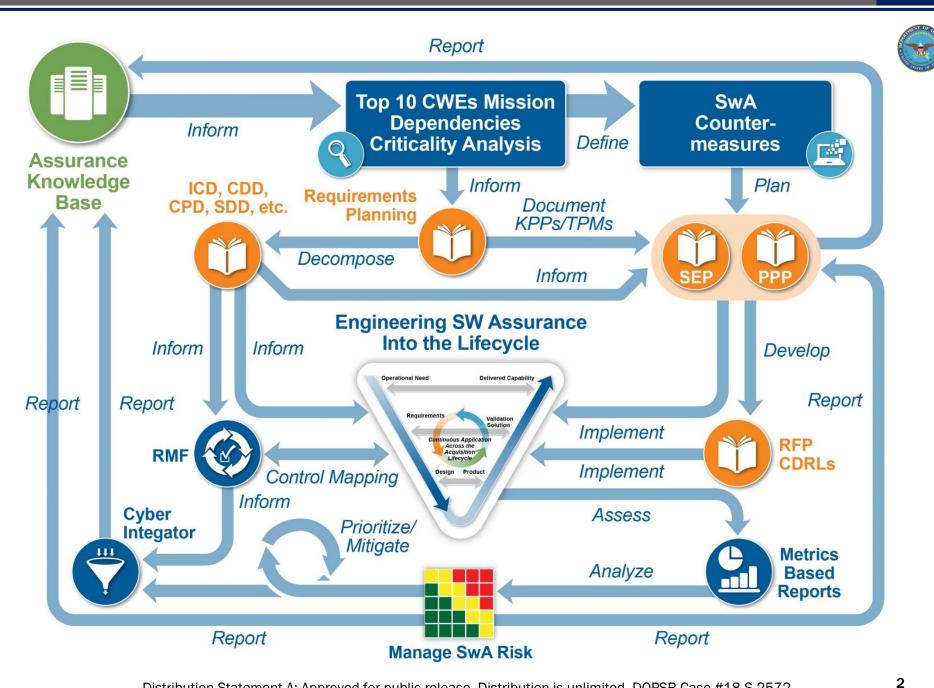


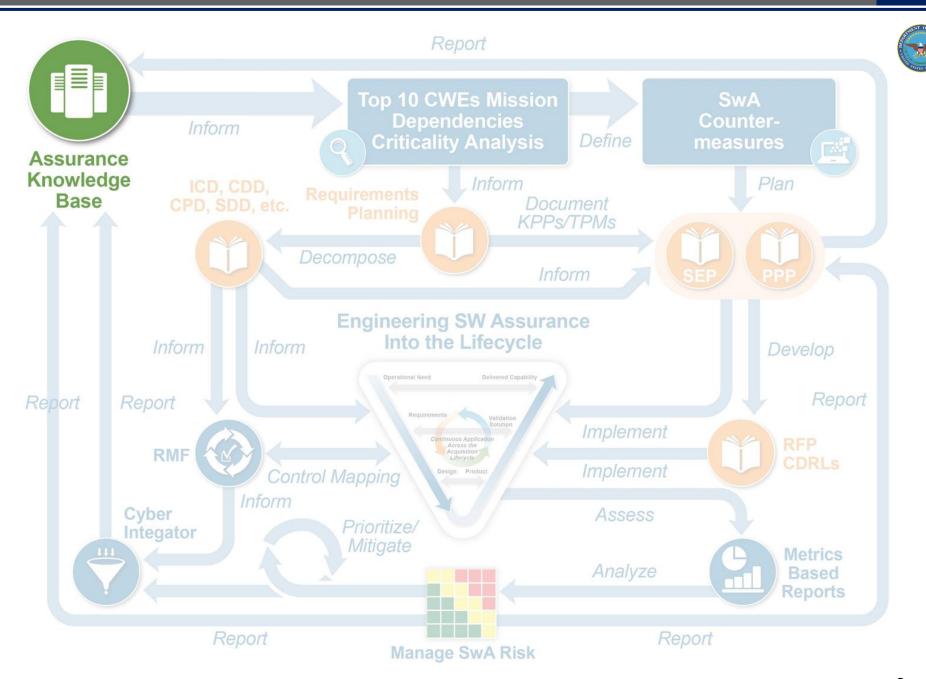


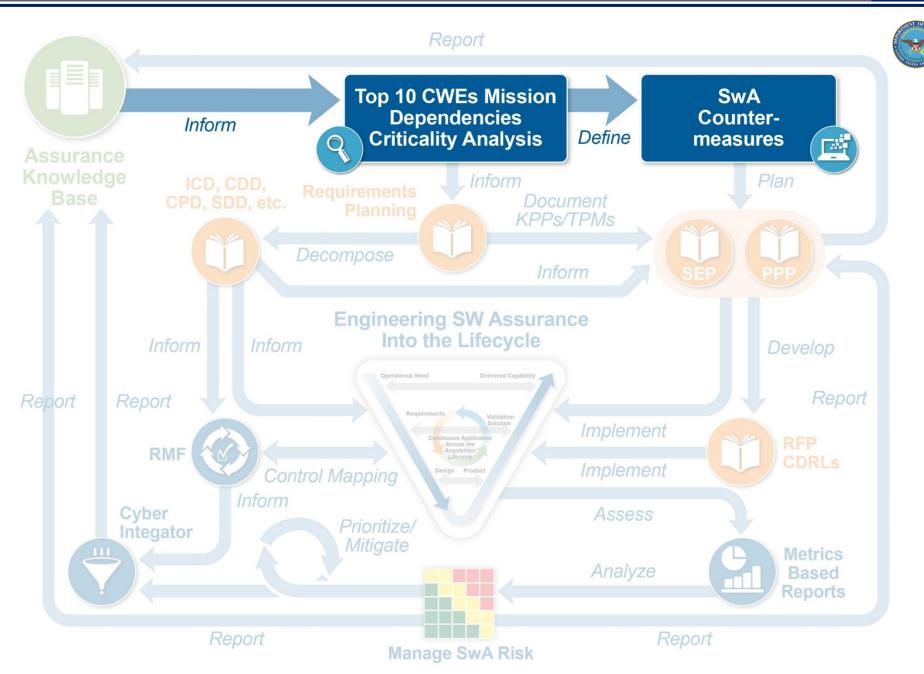
# Software Assurance Concept of Operations (CONOPS)

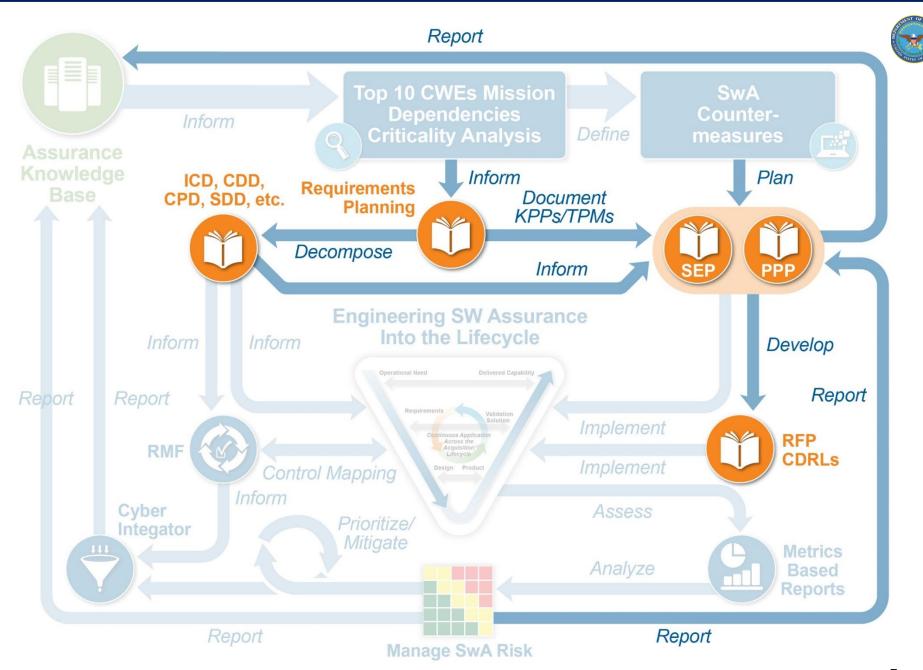


This CONOPS looks at the fundamental software assurance related processes that program managers should expect to see built in to a system's development lifecycle.









### **Engineering Software Assurance** into the Lifecycle



**Operational Need** 

Requirements

**Delivered Capability** 

**Validation** 

Solution

#### **Business or Mission Analysis**

Identify threat environment and opportunities for attack

#### Stakeholder Needs & Reg Definition

Define Functional requirements for operation in cyber contested environment

#### System Req. Definition

Derive Non-functional SwA Requirements

#### Architecture Definition

- Develop secure architecture
- Obtain data rights

#### **Design Definition**

Design system with considerations for SwA

#### System Analysis

- Criticality Analysis
- SwA Evaluation of COTS

  - Selection of Cybersecurity Controls

· Submission of TAC Reports

#### SUPPORT:

- · Risk Management: Assurance case
- · Configuration Management: Version control, Access control, Code signing
- Measures & Metrics

Continuous Application

Across the

Acquisition :

Lifecycle

Product

Design

Disposal

#### Sustainment and Continuous Engineering

- · Monitor for 3rd party vulnerabilities
- Continued Assessment & Timely Patching

#### Operation

- Implement operational monitoring and response
- Risk Management Framework (RMF)

#### Validation

- · Conduct Third Party SwA Testing
- Validate security requirements/assumptions

#### Transition

- · Transition data rights
- Ensure acquirer can rebuild & retest

#### Integration

- Full system regression testing
- Automated reproducible build

#### Implementation

- · Warning flags & Coding standards
- Hardening measures
- Code reviews

#### Verification

weakness analysis

Static source code

- Binary Analysis
- · Origin analysis
- · Web app scanners & fuzzers
- Negative testing
- · Automated test suite w/coverage
- Penetration Testing

NOTE: Implementation, Integration & Verification are often performed continuously & simultaneously with the aid of IDEs & other tools.

NOTE: Lifecycle processes typically occur simultaneously, not in sequence; see ISO/IEC 15288 & 12207

