# Software Acquisition Environment

3.4.1

Version 5.2 26 MAR 2025 3.4.1 Software Acquisition Environment TIME: 1.0 HR

## TOPIC LEARNING OBJECTIVES

Upon successful completion of this topic, the student will be able to:

1. Recognize the major components of a computer and the languages it uses to interface with itself, and the language we use to interface with it.
2. Recognize the importance of software-intensive systems.
3. Recognize the definition of a software-intensive systems.
4. Recognize the DoD software domains.
5. Recognize the differences between Weapons, Command Control Communications Computers Intelligence Surveillance Reconnaissance (C4ISR), and Defense Business Systems (DBS) systems.
6. Identify key laws and software acquisition management policies and practices that are required for the acquisition of a DoD software-intensive system.
    a. Recognize the major provisions of the Information Technology Management Reform Act (ITMRA) contained in the Clinger-Cohen Act (CCA) of 1996.
    b. Recognize the basic provisions of DoDI 5000.02 and DoDI 5000.75 that are applicable to software-intensive systems.
7. Recognize a software item (SI) and how it functions as a building block of software.
8. Recognize the role of interoperability and architecture in software acquisition.

## STUDENT PREPARATION

Student Support Material

1. None

Primary References

1. DoDI 5000.87
2. DoDI 5000.02
3. DoDI 5000.75

Additional References

1. DAU ISA 1011 Information Systems Acquisition

# Overview

- Computer Basics
- Importance of Software
- Software-Intensive Systems
  - Weapon Systems
  - C4ISR Systems
  - Defense Business Systems
- Software Acquisition Policies and Practices
- Software Items
- Software Interoperability and Architecture

# Computer Basics

- Computer
  - A device that receives or senses data through input devices, processes that data, then provides that processed data as an output
  - Computer components:
    - Input (keyboard, mouse)
    - Processing (CPU)
    - Storage (memory)
    - Output (monitor, printer)
  - Military Input/Output (I/O) device examples:
    - Pressure and temperature sensors
    - Electromagnetic spectrum sensors (Radar, EW sensors, etc.)

# Computer Basics

- Software
  - Software contains the instructions that make the computer components operate and communicate
    - System Software
      - Operating systems (UNIX, Windows, MAC OSX, Apple IOS)
    - Application software
      - Microsoft Word
      - Cell phone apps
  - Written using programming languages, which come in many forms

# Programming Language Categories

- Programming languages can be placed in one of four broad categories:
  - Machine Language
    - The only language that is actually “understood” by computers
    - Composed only of the binary digits 0 and 1 known as binary code which control computer actions
  - Assembly-Level Language
    - Uses mnemonics, or memory aid to represent machine language
    - Uses a program called an assembler which "assembles" machine language instructions
  - Higher-Order Languages (HOLs) (C++, Java, etc.)
    - General purpose programming language that allows people to write programs without having to understand the inner workings of a computer
    - HOLs must be translated into machine language by a process called compilation and done by a software tool called a compiler
  - Fourth-Generation Languages (4GLs) (Mathematica, SQL, LabVIEW, etc.)
    - Closer to human language than typical high-level programming languages (better semantic properties)

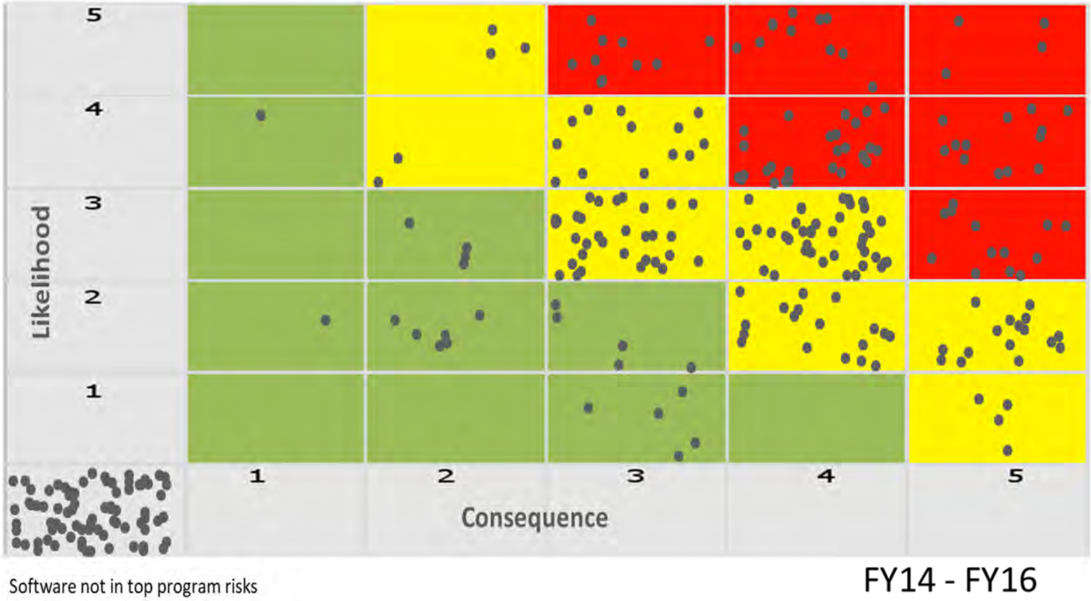
# Overview

- Computer Basics
- Importance of Software
- Software-Intensive Systems
  - Weapon Systems
  - C4ISR Systems
  - Defense Business Systems
- Software Acquisition Policies and Practices
- Software Items
- Software Interoperability and Architecture

# The Importance of Software

- Software is a crucial and growing part of our weapon systems and the national security mission
  - "The DoD is experiencing an explosive increase in its demand for software-implemented features in weapon systems...in the meantime, defense software productivity and industrial base capacity have not been growing as quickly." –Institute for Defense Analyses, 2017

- SW among most frequent and most critical challenges, driving program risk on ~60% of acquisition programs

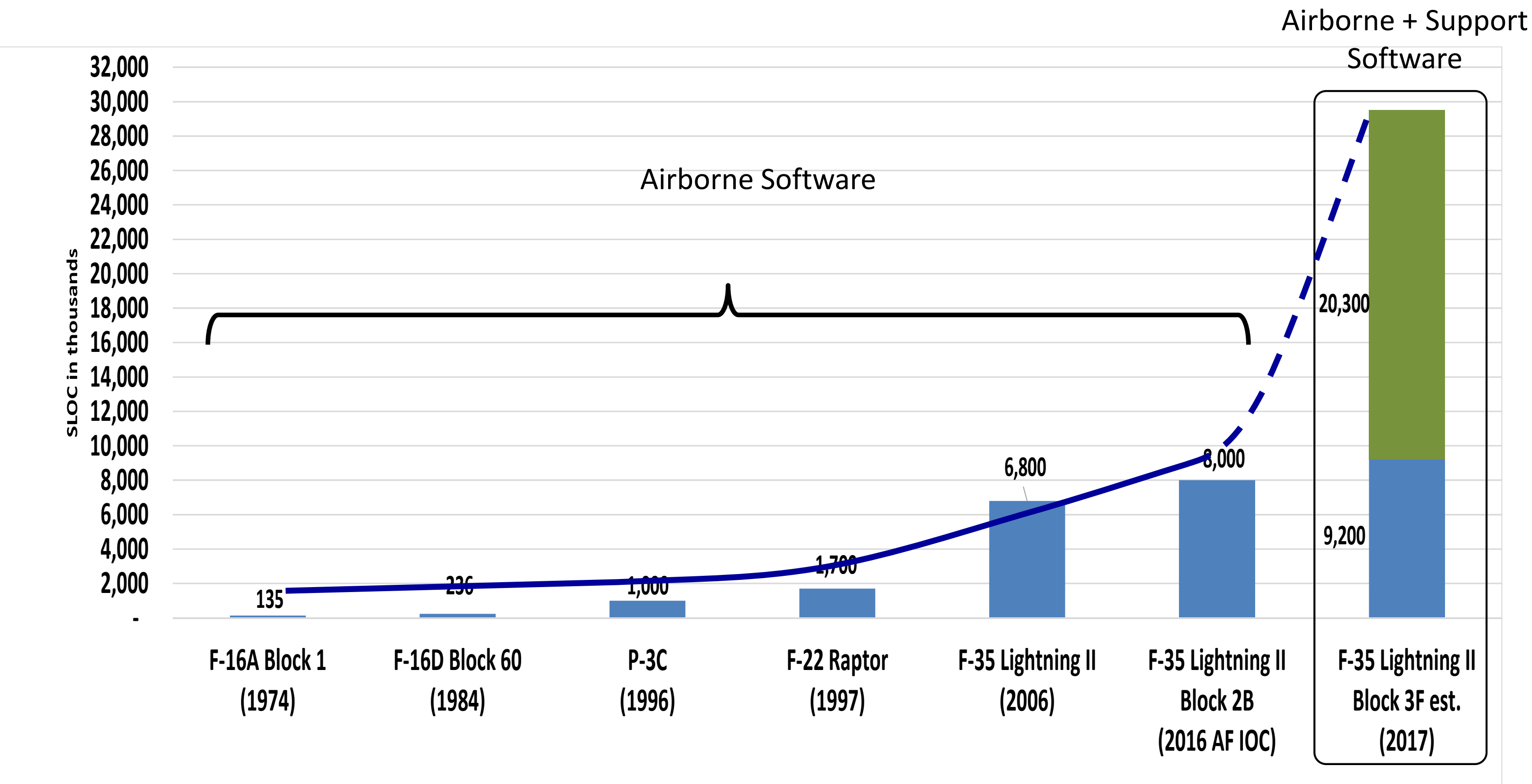


FY14 - FY16

***SW never dies; it requires DoD to update continuously and indefinitely***

# The Importance of Software

- DoD Software complexity and size rapidly growing



*Explosive growth of Source lines of code (SLOC) in Avionics Software*

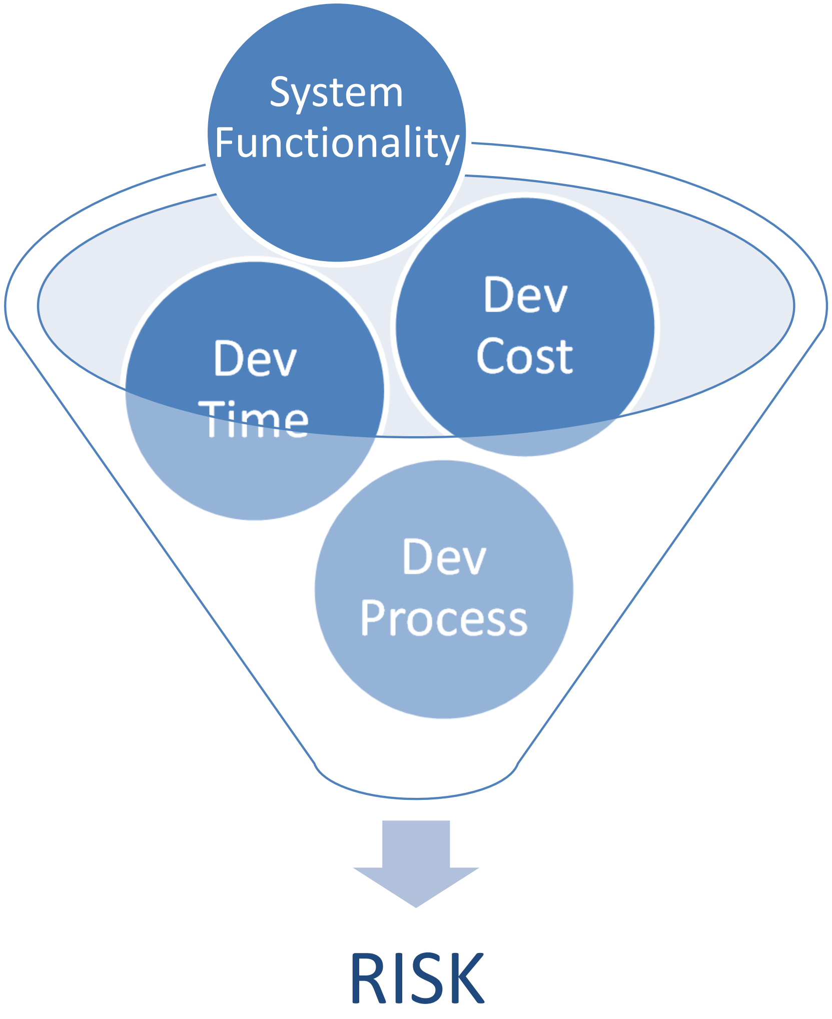
*Functions performed by software have increased significantly*

# Overview

- Computer Basics
- Importance of Software
- Software-Intensive Systems
  - Weapon Systems
  - C4ISR Systems
  - Defense Business Systems
- Software Acquisition Policies and Practices
- Software Items
- Software Interoperability and Architecture

# Software-Intensive System Definition

- A software-intensive system is any system where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole [IEEE-Std-1471-2000]
- In DoD, a system is considered software-intensive ***if software is a major risk driver*** in any of these areas:
  - Development costs
  - Development process
  - System functionality
  - Development time



# Classifications of Software-Intensive Systems

- Software-Intensive Systems are classified into 3 categories:
  - Weapon Systems
  - C4ISR Systems
  - Defense Business Systems

# Weapons Systems

- Software is specifically designed for the system
  - Performs highly specific function
  - Normally embedded into the hardware
- Characteristics
  - Unique and/or proprietary architecture
  - ***Safety is primary design feature***
  - Data life-span measured in nano-seconds (real-time data)
- Examples
  - Guidance system in a missile
  - Navigation equipment in an aircraft
  - Control systems in a tank

# C4ISR Systems

- Communicate, assimilate, coordinate, analyze, and interpret information to provide decision support to military commanders
- Characteristics
  - Mix of proprietary and Commercial Off-The-Shelf (COTS) architecture
  - ***Security is primary design feature***
  - Data life-span measured in seconds (near real-time data)
- Examples
  - Global Command and Control System – Maritime (GCCS-M)
  - Navy Multiband Terminal (NMT)
  - Consolidated Afloat Networks and Enterprise Services (CANES)

# Defense Business Systems

- Systems that are used for routine administrative business applications such as accounting, finance, and personnel
  - They do not directly support combat operations
- Characteristics
  - COTS architecture
  - ***<u>Privacy is primary design feature</u>***
  - Data life-span in measured in hours or longer (not real-time data)
- Examples
  - Defense Travel System (DTS)
  - Navy Enterprise Resource Planning (NERP)
  - Navy Marine Corps Intranet (NMCI)

Defense Travel System Program Management Office

NMCI
NAVY MARINE CORPS INTRANET

# Software-Intensive System Summary

| System | Purpose | Data Life Span | Environment |
|---|---|---|---|
| **Weapons** | - Safety<br>- Performance | - Real-time<br>- Nano-seconds | - Sensor to shooter<br>- Life or death |
| **C4ISR** | - Security<br>- Decision support | - Near real-time<br>- Seconds | - Tactical command<br>- Strategic command |
| **DBS** | - Privacy<br>- Administrative | - Hours or longer<br>- Not real-time | - Data intensive<br>- Business |

# Overview

- Computer Basics
- Importance of Software
- Software-Intensive Systems
  - Weapon Systems
  - C4ISR Systems
  - Defense Business Systems
- Software Acquisition Policies and Practices
- Software Items
- Software Interoperability and Architecture

# Software Management Policies/Guidance

- The PM should base software systems design and development on robust systems engineering principles:
  - Developing architectural-based software systems that support ***open system concepts***
  - ***Exploiting Commercial Off-The-Shelf (COTS)*** computer system products
  - Exploiting, where practicable, ***software reuse*** opportunities before developing new software
  - Selecting contractors with ***domain experience*** in developing comparable software systems; with successful past performance; and with a mature software development capability process
  - ***Planning for transition*** of fielded software to the support/maintenance activity
  - Preparing for ***life-cycle software support or maintenance*** by developing or acquiring the necessary documentation, host systems, test beds, and computer-aided software engineering tools consistent with planned support concepts

# Clinger-Cohen Act (CCA) of 1996

- A combination of the Federal Acquisition Reform Act (FARA) and the Information Technology Management Reform Act (ITMRA)
- ITMRA sets the DoD policy for software acquisition - streamlines IT acquisitions, emphasizes life-cycle management of IT and establishes accountability through Chief Information Officers (CIOs)
  - Establishes capital planning and investment control through the Office of Management and Budget (OMB)
  - Establishes performance and results-based management policy
  - Encourages incremental acquisitions and modular contracting
  - Encourages acquisition of COTS products
  - Defines National Security Systems as intelligence systems, cryptologic systems, C2 systems or software that is an integral part of weapon systems
- FARA streamlined and simplified many contracting procedures throughout the Federal Government

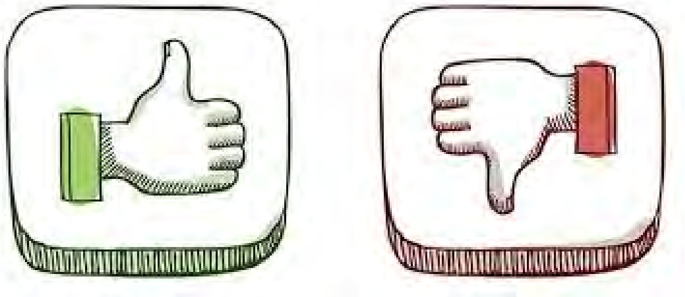
*Clinger-Cohen Act (CCA) governs the acquisition of software-intensive systems within the Federal Government*

6 6a

# Clinger Cohen Act

| CLINGER-COHEN ACT REQUIREMENT | Applicable Documentation |
|---|---|
| Conduct an analysis of alternatives | Acquisition Strategy (Business Strategy section) |
| Conduct an economic analysis that includes a calculation of the return on investment; or for non-AIS programs, conduct a life-cycle cost estimate | Component Cost Estimate, Component Cost Position |
| Determination that the acquisition supports core, priority functions of the DoD | Capability Needs Statement (Capabilities section) |
| Determine that no private sector or government source can better support the function | Acquisition Strategy |
| Develop clearly established measures and accountability for program progress | Acquisition Strategy (Program Metrics, Value Assessment) |
| Ensure that the acquisition is consistent with the DoD Information Enterprise policies and architecture, to include relevant standards | Systems Architecture |
| Ensure that the program has a Cybersecurity Strategy that is consistent with DoD policies, standards and architectures, to include relevant standards | Cybersecurity Strategy |
| Ensure, to the maximum extent practicable, (1) modular contracting has been used, and (2) the program is being implemented in phased, successive increments, each of which meets part of the mission need and delivers measurable benefit | Acquisition Strategy (Contracting Strategy section) |
| Establish outcome-based performance measures linked to strategic goals | Capability Needs Statement (Performance Attributes section) |
| Redesign the processes that the system supports to reduce costs, improve effectiveness and maximize the use of commercial off-the-shelf technology | Capability Needs Statement (Program Summary Section) |
| Register Mission-Critical and Mission-Essential systems with the DoD CIO | DoD Information Technology Portfolio Repository |

# Commercial Off-The-Shelf Software



PROs

- Saves money in design & development
- Can reduce development risk
- Selection of COTS early will enable requirements to be driven by commercial software capabilities
- Very useful in rapid prototyping
- Leverage industry expertise and infrastructure for development

CONs

- Still must be integrated into the system and qualified
- Difficulty in configuration management and support for older releases
- Additional testing may be required to qualify system
- Subsequent releases determined by vendor
- Security aspects of COTS not well understood
- Licensing costs set by vendor

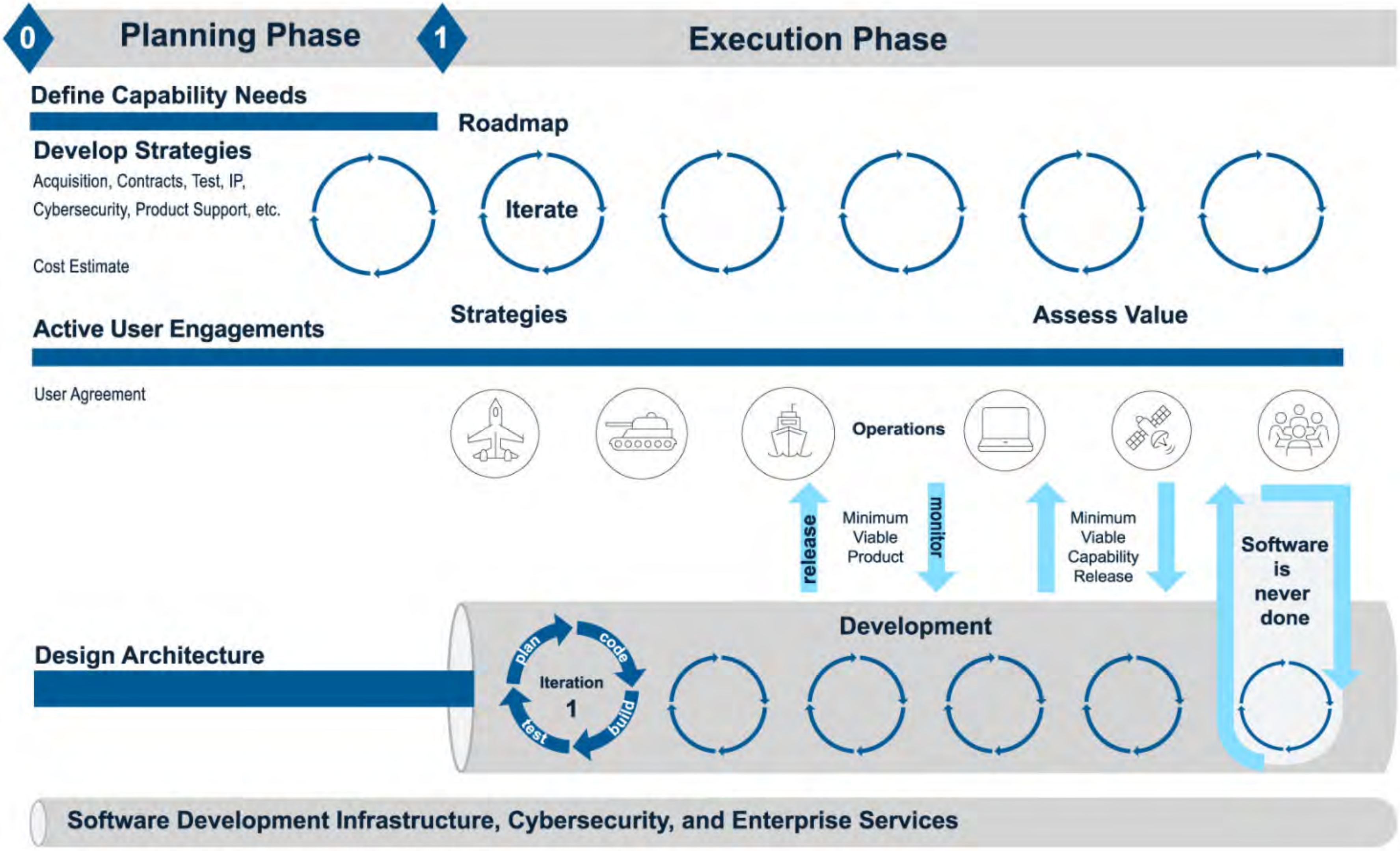
# DoD Instruction (DoDI) Provisions

- DoDI 5000.02 Operation of the Adaptive Acquisition Framework
  - Acquisition categories, phases, and milestones
  - Systems Engineering
  - Program Management and Intellectual Property
  - Test and Evaluation
  - Acquisition of Information Technology
  - Human Systems Integration (HSI)
  - Cybersecurity
- DoDI 5000.75 Business Systems Requirements and Acquisition
  - Implements statutory requirements of CCA
  - Established framework to assess risk
  - Minimize customization of COTS
  - Increase use of Government-Off-The-Shelf (GOTS)/COTS to maximum extent
  - Established the ***tailorable*** Business Capability Acquisition Cycle (BCAC) with authority to proceed (ATP) decision points in lieu of formal milestones

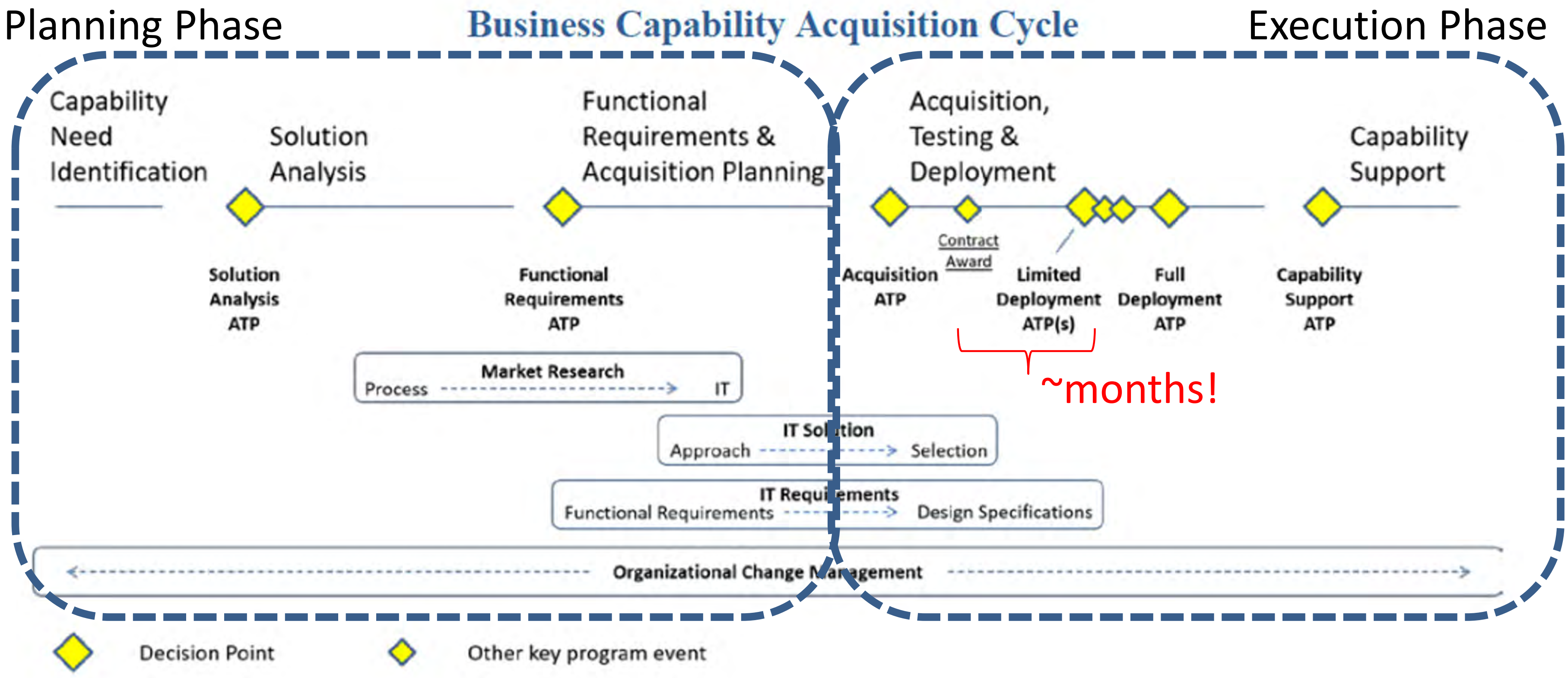
# DoD Instruction (DoDI) Provisions

- DoDI 5000.87 Operation of Software Acquisition Pathway
  - Software programs shall not be treated as a MDAP
  - Exempt from JCIDS
  - Streamline SW requirements, budget, and acquisition processes
  - Facilitate rapid and iterative delivery of software capability to the user
    - Enables DoD to deliver better software faster
  - Integrates modern software practice such as Agile Software Development, DevSecOps, Lean Practices, and User Centered Design

# Software Acquisition Pathway



# Business Capability Acquisition Cycle



- Streamline activities to align with rapid, frequent, and measureable deliveries
- Bring the user closer to the acquisition community
- Deploy meaningful capability early and often

# Overview

- Computer Basics
- Importance of Software
- Software-Intensive Systems
  - Weapon Systems
  - C4ISR Systems
  - Defense Business Systems
- Software Acquisition Policies and Practices
- Software Items
- Software Interoperability and Architecture

# Software Item (SI)

- As part of the Systems Engineering process, software is broken down into smaller building blocks called Software Items (SIs)
- A SI is a collection of software that performs closely related functions that is uniquely designated for the purposes of:
  - Managing requirements
  - Conducting qualification testing
  - Controlling interfaces
  - Ensuring configuration management
  - Mitigating risk
- To facilitate programming, SIs can be broken down into smaller, logically related pieces, usually referred to by commercial standards as Software Units. Software Units include individual programs like routines to perform specific functions
- Failure to control a SI can result in risks and vulnerabilities to a software intensive system

# Overview

- Computer Basics
- Importance of Software
- Software-Intensive Systems
  - Weapon Systems
  - C4ISR Systems
  - Defense Business Systems
- Software Acquisition Policies and Practices
- Software Items
- Software Interoperability and Architecture

# Interoperability

- DoD systems must be Net-Ready (NR), interoperable, and in compliance with Federal and DoD enterprise architecture

- The NR Performance Attribute ensures interoperability between individually developed and fielded capability solutions
  - Fielded IT solutions must achieve Net-Ready Certification IAW DoDI 8330.01, Interoperability of Information Technology (IT), Including National Security Systems (NSS), 21 May 2014, Incorporating Change 1, 18 December 2017

- The DoD Architecture Framework (DoDAF) is the overarching framework used to develop architectural descriptions in DoD
  - Interrelated set of viewpoints and associated models to address interoperability
  - DoDAF views submitted are determined by Sponsor where no Joint integration aspects are present
  - Programs select the DoDAF views that fit their purpose and tailor accordingly

# Summary

- A software intensive system is a system in which one or more of the following is a major ____________ driver:

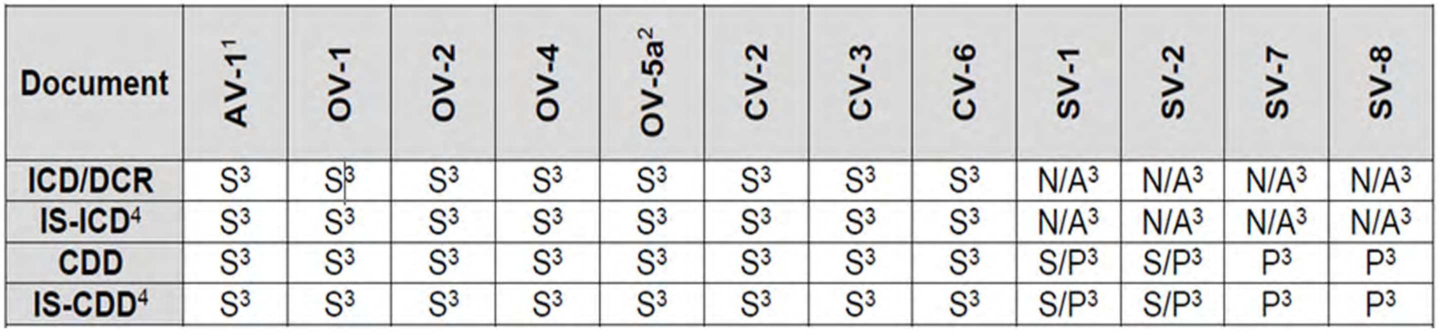
- What are the classifications of Software-Intensive Systems?

- Which law is the major driver in DoD software acquisition policy?

# Backup Slides

# Enterprise Architecture

- DoDAF views required to support all capability documents are:



| Document | AV-1[1] | OV-1 | OV-2 | OV-4 | OV-5a[2] | CV-2 | CV-3 | CV-6 | SV-1 | SV-2 | SV-7 | SV-8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ICD/DCR | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | N/A[3] | N/A[3] | N/A[3] | N/A[3] |
| IS-ICD[4] | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | N/A[3] | N/A[3] | N/A[3] | N/A[3] |
| CDD | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | S/P[3] | S/P[3] | P[3] | P[3] |
| IS-CDD[4] | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | S[3] | S/P[3] | S/P[3] | P[3] | P[3] |

**DoDAF Key**

**AV:** All
**OV:** Operational
**CV:** Capability
**DIV:** Data/Information
**SV:** System
**StdV:** Standards
**SvcV:** Services

**IS-:** Information Systems
**DCR:** DOTMLPF-P Change Recommendation
**ICD:** Initial Capabilities Document
**CDD:** Capability Development Document
**RDP:** Requirements Definition Package
**CP:** Change Proposal

- Additional views required to support NR Certification are:

| Document[1] | AV-1[2] | DIV-1 | DIV-2 | DIV-3[3] | SV-1 | SV-2 or SvcV-2 | SV-4 or SvcV-4 | SV-5a[4] or SvcV-5 | SV-6 or SvcV-6 | SV-7 or SvcV-7 | StdV-1[5] | StdV-2[5] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CDD | S/P[6] | S[6] | S[6] | P[6] | S/P[6] | P[6] | P[6] | P[6] | S/P[6] | P[6] | P[6] | P[6] |
| IS-ICD/ IS-CDD[7] (RDPs/CDs) | S[6] | N/A[6] | N/A[6] | N/A[6] | S[6] | N/A[6] | N/A[6] | N/A[6] | N/A[6] | P[6] | N/A[6] | N/A[6] |

S: Program Sponsor Provides P: Program Office Provides

# Types of Software

- The four primary types of software:
  - Type A (Commercial Off-the-Shelf [COTS] applications)
  - Type B (Customized Software)
  - Type C (COTS Hardware/Operating Systems)
  - Type D (Custom Software/Hardware)

https://media.defense.gov/2019/May/01/2002126693/-1/-1/0/SWAP%20MAIN%20REPORT.PDF

# COTS

- COTS applications:
  - Consists of applications that are available from commercial suppliers
  - Business processes, financial management, HR, software development, collaboration tools, accounting software, and other “enterprise” applications in DoD are generally not more complicated nor significantly larger in scale than those in the private sector
  - Unmodified commercial software should be deployed in nearly all circumstances

https://media.defense.gov/2019/May/01/2002126693/-1/-1/0/SWAP%20MAIN%20REPORT.PDF

# Customized Software

- Customized Software:
  - Applications that consist of commercially available software that is customized for DoD-specific usage
  - Customization can include the use of configuration files, parameter values, or scripted functions tailored for DoD missions
  - These applications generally require (ongoing) configuration by DoD personnel, contractors, or vendors

https://media.defense.gov/2019/May/01/2002126693/-1/-1/0/SWAP%20MAIN%20REPORT.PDF

# Open-Source Software (OSS)

- Open-source software (OSS) is software whose source code is made publicly available, allowing users to study, modify, and distribute it under defined terms and conditions

- There are two fundamental concerns for the DOD that are specific to OSS
  - Using externally maintained code in critical systems potentially creates a path for adversaries to introduce malicious code into DoD systems. This concern requires a careful supply chain risk management approach
  - Imprudent sharing of code developed for DoD systems potentially benefits adversaries by disclosing key innovations. This risk is managed through a Modular, Open-Systems Approach (MOSA), which allows systems to benefit from OSS while protecting critical, innovative components as separate modules

Reference: https://dodcio.defense.gov/portals/0/documents/library/softwaredev-opensource.pdf

# Custom Software/Hardware

- Custom Software/Hardware:
  - This class of software focuses on applications involving real-time, mission-critical, embedded software whose design is highly coupled to its customized hardware.
    - Examples include primary avionics or engine control, or target tracking in shipboard radar systems
  - Requirements such as safety, target discrimination, and fundamental timing considerations demand that extensive formal analysis, test, validation, and verification activities be carried out in virtual and “iron bird” environments before deployment to active systems
  - These considerations also warrant care in the way application programming interfaces (APIs) are potentially presented to third parties

https://media.defense.gov/2019/May/01/2002126693/-1/-1/0/SWAP%20MAIN%20REPORT.PDF

# Custom Software/Hardware

- Custom Software/Hardware:
  - This class of software focuses on applications involving real-time, mission-critical, embedded software whose design is highly coupled to its customized hardware.
    - Examples include primary avionics or engine control, or target tracking in shipboard radar systems
  - Requirements such as safety, target discrimination, and fundamental timing considerations demand that extensive formal analysis, test, validation, and verification activities be carried out in virtual and "iron bird" environments before deployment to active systems
  - These considerations also warrant care in the way application programming interfaces (APIs) are potentially presented to third parties

https://media.defense.gov/2019/May/01/2002126693/-1/-1/0/SWAP%20MAIN%20REPORT.PDF