ISA 201
Intermediate
Information Systems Acquisition
Lesson 20
Software Support
Overall 1: Given a Department of Defense (DoD) IT acquisition scenario early in the lifecycle, analyze software sustainment planning factors critical to the success of a potential Software Support Activity (SSA).

- During the planning stage of DoD system software acquisition, identify relevant considerations that influence the lifecycle support of a software-reliant system.
- Identify the important software life cycle planning documents and their major components.
- Identify at least one major activity related to the Product Support BCA that should occur at relevant acquisition milestones.

Overall 2: Given a Department of Defense (DoD) IT acquisition scenario, manage IT system transition to a Software Support Activity (SSA) for Post Deployment Software Support (PDSS).

- Recognize the advantages and disadvantages of software transition deployment strategies.
- Recognize critical success factors for software transition.
Learning Objectives

Overall 3: Given an IT acquisition scenario during the Operations and Sustainment (O&S) Phase, manage the optimal life-cycle sustainment approach for system software sustainment.

- Describe the keys to successful software sustainment and support.
- Compare the types of software maintenance.
- Describe the purpose of the DoD software disposal process.
- Describe the impacts of not following the DoD software disposal process.
- Identify the items that must be considered to execute the DoD software disposal process.
- Describe the outcomes of successful DoD software disposal.
• Software Lifecycle Planning
  • Software Transition
  • Sustainment and Support
  • Disposal
  • Exercise
Lesson Overview

Team discussions from Camtasia:

- Team 1 -- Explain the difference and relationship between design for support and support the design

- Team 2 – Discuss the primary planning considerations for software sustainment that should be addressed in program IPTs

- Team 3 – Identify and discuss some of the categories that should be assessed when considering software for reuse

- Team 4 – Discuss the various Software Transition Success Factors

- Team 5 – Identify and discuss the deployment methodologies

Homework review

Software Support
Must plan to manage more than just this

What are some considerations when conducting software sustainment planning?
Many considerations when conducting software sustainment planning. SW development environment needs to be replicated.
Software Support Planning and the Acquisition Life Cycle

- Develop software support strategy
- Investigate alternative support concepts
- Influence product
- PMO establishes a Computer Resources IPT (CR-IPT)
- Prepare initial Computer Resource Plans

- Select support concept
- Propose software quality requirements
- Propose software support acquisition requirements
- Identify SSA resource requirements
- Review developer’s support documentation and transition plans for software (e.g., STrP)
- Update Computer Resource Plans (LCSP)

- Ensure software supportability
- Evaluate software quality
- Certify software documents & technical data
- Incorporate developer’s support documentation and transition plans for software
- Update Computer Resource Plans (LCSP)

- Implement Transition Plan
- Acquire & install SEE/STE
- Acquire SSA resources
- Staff and train personnel
- Demonstrate support capability
- Update Computer Resource Plans (LCSP)

- Manage support
- Conduct support
- Provide software logistics support
- Perform software configuration management
- Maintain Computer Resource Plans (LCSP)
1. What are the projected costs to fix/update the Software Item (SI) to be developed in O&S?

2. Is the software architecture designed with open interfaces to ensure plug and play updating with new module?

3. Are data rights in place to ensure we can update/fix bugs?

4. What tools will the SSA need to fix/update the software?

5. How is the developed data managed and stored so the SSA knows how to manage and store data during O&S?

6. Is the code being developed using secure coding practices? Static and Dynamic code analysis (NDAA)

7. What is the proposed Security Architecture of the SI being built?
Business Case Analyses help identify pros/cons of various software development/sustainment COAs.

2 BCA versions related to IT management:

- **IT BCA**
  - Required to support IT investment decisions for the Investment Review Board (IRB)
  - Goal is to justify ROI
  - DOD CIO Memo 23 Oct 2014

- **PS BCA**
  - Required to support each MS decision and every 5 years after deployment
  - Goal is to optimize holistic sustainment approach
  - USD AT&L WSAR 2009, NDAA 2010 sec. 805
  - Can support selection of Software Support Activity (SSA)
  - May also consider warranty implications (CBA) for COTS sustainment
Lesson Overview

Lesson Plan

• Software Lifecycle Planning 4 to 10

• Software Transition
  • Sustainment and Support
  • Disposal
  • Exercise
Software Deployment—Strategies

- Immediate cut-over
- Phased cut-over
- Parallel cut-over
### Deployment Strategies
**Cost, Effort, and Risk**

<table>
<thead>
<tr>
<th>Method</th>
<th>Cost</th>
<th>User Effort</th>
<th>Team Effort</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Pilot</td>
<td>Medium</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Phased</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Immediate</td>
<td>Low</td>
<td>Medium</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>

“When you deploy your system, expect the unexpected!”
Necessary activities prior to transitioning to sustainment:

- Source of Repair Assignment Process (SORAP)
- Test & Evaluation
- Stable Software Baseline
- Complete Documentation
- Authority to Operate
- Software Transition Plan
- Staffing and Training Plan (MER/MP&TP)

The Software Transition Plan identifies the resources needed to support delivered software and describes the developer’s plans for transitioning delivered software to the support agency.
What must be considered for transition?

- Support database(s)
- Development environment infrastructure
- Software Test Lab/Software Integration Lab
- Release processes/procedures
- Staffing
- Operations and Maintenance training
- System documentation
- Stable software baseline
- ATO
Software Support Activity (SSA)

- Assumes the role of providing post-deployment software support for modifications or upgrades made to a system's software following the system's initial fielding.

- The SSA organization typically compiles updates into formal software releases to avoid disrupting the fielded system.

- Software Maintenance activities performed by a SSA are the same as those carried out during the development effort that led to the first fielding. They are tailored, as appropriate, to reflect the effort required to implement each change package, update pertinent documentation, verify the changes, and distribute the changes to users.
Due to the scope and nature of SSA activities, and given the ‘7 factors’, they need to be involved, early identification and incorporation of SSAs into planning is a key enabler to successful sustainment.

Considerations/Lessons Learned include:
- Identifying the sustainment strategy early: What functionality/authority will the SSA have, and what information will they require?
- Use an early PS BCA to identify candidate SSAs
- Working with the SSA to identify their needs, ensure appropriate development contracts/deliverables are in place.
- Establish SSA performance expectations – measures/metrics
- Use SLA/MOA/MOU to articulate the SSA relationship.
- SSA can help create the Software Transition Plan (STrP) to discuss how they will assume control from the developer.
• Software Lifecycle Planning
• Software Transition

• Sustainment and Support
• Disposal
• Exercise
Software Sustainment Key Considerations: Sustainment vs Maintenance

- **Software Sustainment**
  - Comprehensive requirements to support, maintain and operate the software capabilities of a system.
  - Includes processes, procedures, people material and information required to support.

- **Software Maintenance**
  - One facet of software sustainment.
  - Software failures not subject to the ‘laws of physics’
  - Software maintenance is essentially development, creating a new version (baseline).

- **Hardware Maintenance**
  - Failures are subject to the ‘laws of physics’
  - Hardware maintenance typically returns a failed system to its previously established baseline.

- **Implications**
  - Software maintenance requires a ‘redevelopment’ environment
  - Software maintenance drives additional considerations inherent to a new version (baseline), including configuration management/control, updated manuals and training, distribution process (release management), etc.
There are 8 drivers of software change throughout sustainment:

- Defect corrections
- Responding to Threats
- Policy and doctrine
- Safety
- Enhance Interoperability
- Hardware changes
- Technology insertion
- Functional changes based on user requests
Software Sustainment Key Considerations: Types of Software Maintenance

- **Corrective Maintenance**
  - Reactive modification to correct discovered problems
- **Adaptive Maintenance**
  - Modification to maintain usability in a changed environment
- **Perfective Maintenance**
  - Provide functional enhancement to users
- **Preventative Maintenance**
  - Enhance maintainability

Traditional ‘Maintenance’ (fixing ‘bugs’)

Types of Software Maintenance

- Preventative, 5%
- Corrective, 20%
- Perfective, 50%
- Adaptive, 25%

Perfective, Adaptive, and Preventative (80%) are considered “software evolution”

In 1995, it was estimated that $70Billion was spent maintaining 10Billion LOC in the U.S. As seen in other acquisition areas, ~70% of SW costs are in sustainment. In 1995, sustainment costs were ~$7 per year per KLOC. What might it be today?
• **Strive for Commonality**
  - Ultimate goal is to reach consensus on a common sustainment solution and, thereby, minimize the incidence of multiple system/software configurations.

• **Apply Industrial Engineering Practices to Software**
  - Parameters like the frequency and quantity of software changes, the number of versions, and the time required for development, validation, and distribution must be considered early on.

• **Engage Customers**
  - Each stage of sustainment should engage users and relevant subject matter experts.
  - Ensure adequate training is provided when needed
  - Use a closed loop system to capture customer feedback (e.g. Help Desk metrics)

• **Adopt a Holistic Approach to Sustainment**
  - The effect of any software change should be evaluated in terms of net worth provided to the warfighter.
Keys to Successful Sustainment (2 of 2)

• Develop Highly Maintainable Systems and Software
  - Enhance ability for SSA to ‘understand the product’ by reducing complexity and ensuring documentation is accurate.
  - Ensure documentation is kept up to date
  - Leverage an open and scalable architecture to increase maintainability.
  - Allows for expansion with minimal impact to unchanged elements.

• Manage Off-the-Shelf Software
  - Up-front savings of COTS frequently offset by risk/expense later in the product life cycle.
  - This wreaks havoc on the program without appropriate management.

• Plan for the Unexpected
  - Develop realistic and detailed scenarios to minimize surprises.
  - Plan for unexpected growth – the ‘1% rule’.

• Analyze and Refine the Software Sustainment Business Case
  - Address business case analysis sustainment, annual sustainment total ownership cost estimates, and software cost estimation practices that support these analyses.
System obsolescence, technology refresh, and upgrade planning
- Each COTS software product life cycle includes updates, refreshes, and obsolescence (i.e., unsupported releases).
- Life cycle is not based on the users’ requests or budgetary cycles, but rather on marketplace demands and COTS software vendors’ business plans.

Source code escrow
- Source code may be owned by the COTS vendor or the third-party integrator.
- Problems can arise when the COTS vendor goes out of business or no longer exists due to a business merger or acquisition.

Vendor license management
- During development, licenses may be managed by the system integrator.
- The transition of license management tasks to the sustainment organization needs to be jointly planned by the program office and sustainment organization.

Architecture and COTS software interfaces
- During system development, third-party integrators/developers may capitalize on relationships with COTS software vendors to acquire system-specific capabilities.
- These capabilities may not be in the official version of the product and there is no guarantee that these “extra” features will be maintained as the product evolves.
• Choosing a government SSA/Product Support Provider
  - Data rights and licensing considerations with COTS items
  - Service Level Agreements/MOU/MOA – (GOTS?)
  - Functional authority (Help Desk, Configuration Management/Change Control, Information Assurance, etc)
  - Performance measures reporting/frequency
• ‘Modified’ COTS
  - Discouraged, creates unique configurations that marginalize COTS benefits.
• Counterfeit/Grey Market/Malicious software considerations
  - DODI 4140.67 Counterfeit Prevention Policy, Apr 2013
    - CLL032/CLL062
  - DODI 5200.39 Critical Program Information Protection within DOD
• Warranty considerations
  - BCA to consider ROI for extended warranties
• Software Lifecycle Planning
• Software Transition
• Sustainment and Support

• Disposal

• Exercise
Objective is to retire a system's existing software products or services while preserving the integrity of organizational operations.

IT/Software Disposal Process—Purpose
(per ISO/IEC 12207:2008, Systems and Software Engineering)

- To end the existence of a system’s software entity.
- Ends active support by the operation and maintenance organization, or deactivates, disassembles and removes the affected software products, consigning them to a final condition and leaving the environment in an acceptable condition.
- Destroys or stores system software elements and related products in a sound manner, in accordance with legislation, agreements, organizational constraints and stakeholder requirements.
- Where required, it maintains records that may be monitored.
NIST Special Publication 800-88 also contains information regarding security and privacy in cloud computing.
• US Policy Guidance
  - Reuse within own agency/organization
  - Post notice on the Defense Information Systems Agency (DISA) surplus equipment list
  - Transfer useful equipment to schools
  - Dispose of as other surplus equipment
• Environmental considerations
  - Specific hazards: lithium batteries, Cathode Ray Tubes (CRTs)
  - Volume of solid waste
  - Other Environmental, Safety and Occupational Health (ESOH) impacts
• Secure data archiving considerations
  - Determine what data must be archived, and for how long.
  - Who will archive data – SSA?
    - PM responsibility DAG 4.2.3.1.7.3
As a result of successful implementation of the IT/Software Disposal Process:

- a software disposal strategy is defined;
- disposal constraints are provided as inputs to requirements;
- the system's software elements are destroyed or stored;
- the environment is left in an agreed-upon state; and
- records allowing knowledge retention of disposal actions and any analysis of long-term impacts are available.
Lesson Overview

Lesson Plan Status

- Software Lifecycle Planning
- Software Transition
- Sustainment and Support
- Disposal

Exercise
The initial Joint Training and Maintenance System (JTAMS) Increment 1 software support strategy was to use the developer, JUGGERNAUT, to provide software support. Due to declining budgets the Program Management Office (PMO) is considering changing the support strategy for JTAMS Increment 2 software. It has been proposed that the PMO use support provided by an existing DoD SSA.

- What are the pros and cons of moving to support to a government SSA?
- What factors should be considered when making that decision?
Summary

Today we learned to (1 of 2):

Overall 1: Given a Department of Defense (DoD) IT acquisition scenario early in the lifecycle, analyze software sustainment planning factors critical to the success of a potential Software Support Activity (SSA).

• During the planning stage of DoD system software acquisition, identify relevant considerations that influence the lifecycle support of a software-reliant system.
• Identify the important software life cycle planning documents and their major components.

Overall 2: Given a Department of Defense (DoD) IT acquisition scenario, manage IT system transition to a Software Support Activity (SSA) for Post Deployment Software Support (PDSS).

• Recognize the advantages and disadvantages of software transition deployment strategies.
• Recognize critical success factors for software transition.
Today we learned to (2 of 2):

Overall 3: Given an IT acquisition scenario during the Operations and Sustainment (O&S) Phase, manage the optimal life-cycle sustainment approach for system software sustainment.

- Describe the keys to successful software sustainment and support.
- Compare the types of software maintenance.
- Describe the purpose of the DoD software disposal process.
- Describe the impacts of not following the DoD software disposal process.
- Identify the items that must be considered to execute the DoD software disposal process.
- Describe the outcomes of successful DoD software disposal.
When are BCA’s Conducted?

Product Support BCAs are accomplished throughout the life cycle.
- Rationale for software sustainment:
  - Software provides an increasing proportion of functionality in our weapon systems, growing at an exponential rate.
  - Software Maintenance is growing faster than new software development
    - ~1996, software development was roughly 18Million LOC, while Maintenance was ~20Million LOC (111%)
    - ~2011, software development was roughly 38 Million LOC, while Maintenance was ~54Million LOC (142%)
September 2009

Some Defense Department organizations haven't scrubbed data from information technology equipment before disposing of the hardware, resulting in the possible release of information that could be used for identity theft, or releasing other sensitive DoD information, according to an Inspector General audit.

An investigation by DoD's IG also found that one organization had lost track of one unclassified computer entirely, the report said.
Software Sustainment entails developing a maintenance capability that mirrors developmental environment. Software maintenance activities include:

- Understanding Requirement
- Understanding Existing Code
- Implementing Change
- Checkout & Deployment

However, poor documentation of existing code can dramatically increase the effort required to ‘understand the product’ – often up to 50% of the entire sustainment effort. The effect is:

- Increased sustainment costs,
- Longer schedule requirements, or
- Reduced testing and debugging (higher failure rates post-deployment).

“Pay me now or pay me later…”