Overcoming the Challenges of Sustaining Multi-Mission Software Capabilities

Spencer Border
LM Aero / Avionics & Software
770-494-1643
spencer.border@lmco.com
What is Sustainability?

- **Traditional:** A method of using a resource so that the resource is not depleted or permanently damaged and has the capacity to endure

- **Software:** The processes, procedures, material, and information required to support, maintain, and operate a software system with the capacity to endure without damaging system resources
Software Sustainment

- Maintenance and operations considered part of software sustainment

- Consist of correcting failures, improving performance, adapting to a changing environment, preventive maintenance

- Involves more than modifying and updating source code
Software Sustainment Process

Identification

- Identify Problems
- Design new software around existing features

Analysis

- Validate Needs, Useful Analysis

Design

- Modify or Add new code

Programming

- Integrate/Test new development, Regression Testing

Integration & Test

- Acceptance test, Review updated baseline

Acceptance

- Install & move to new system

Deployment
Why Sustain Software?

• Multi-mission systems are becoming more dependent on software
  – "The B-52 lived and died on the quality of its sheet metal. Today our aircraft will live or die on the quality of our software." – Air Force General\(^1\)

• Software is deployed in a constant evolving environment

• Software does not wear & tear – sustainment costs are affected by the number of variants
Multi-Mission Software Sustainment Challenges

- Modernizing and Recapitalizing legacy systems
- Repurposing systems to meet the needs for new threats and interoperability
- Performance driven by Moore’s Law, COTS products, Engineers with modern skills, more
Software Sustainment Effort

Can account for 60% - 90% of total software effort

Too High!
What we already do

• Manage software commonality and variability via software-product lines
  – Building sets of variable systems from common assets
  – Block Upgrades, Incremental Builds
  – Software development follows a templated architecture

• Management of integrating vendor software

• Customer specific sustainment contracts
What can we do?

• Create common software across all products
• Understand and alleviate rising software sustainment cost
• Determine approach to software – Operating standards, estimation models, work structure
• Recruit, develop, and retain a software sustainment workforce
• Be a versatile and competitive organization
Common Software Product

- Baseline includes all customer software
- Software execution based on aircraft/customer configuration
Common Software Product

• Re-Verify unique requirements
  – Do they really need to be unique from the baseline? Example: ACAWS definition

• Allows for faster release of software systems

• A common software baseline for all customers will help alleviate sustainment cost
Understanding Multi Mission Software Sustainment Cost

- Sustainment cost are a function of early lifecycle phase development
- As variations in capabilities are introduced and products age, it becomes more difficult to sustain
- External, Internal influence factors
  - Environments, Processes, Personnel
Understanding Multi-Mission Software Sustainment

Organizational Environment
- Changes in policies
- Competition in the marketplace

User Environment
- User Requirements

Maintenence

Software
- Difficulty of application domain
- Quality of Documentation
- Complexity of Program
- Program Structure

Process & Personnel
- Capturing Requirements
- Undocumented Assumptions
- Variation in Programming Practice
- Error Detection and Correction

Operational Environment
- Staff Turnover
- Domain Expertise

Hardware and software innovation
Recruiting and Retaining Necessary Skill Sets

- Recruit, Develop, and Retain a software sustainment workforce
- Recruit Engineers with modern skills
- Retain Engineers who understand legacy systems
Adaptability/Flexibility is Key to Succeeding

- Prepare ourselves for a shrinking defense budget and increased multi-mission complex systems on top of legacy systems

- Plan for the future

- Goal is to provide advantages today, while competing in the changing environment of tomorrow
References


