Challenges of Employing Agile DevOps Practices for Embedded Operational Flight Software

Shawn Taylor
Technical Director
76th Software Engineering Group
Tinker AFB, OK
shawn.taylor@us.af.mil
405-736-5834
Embedded Operational Flight Software Growth

SLOC estimates at initial production

Software is Today’s Catalyst in enabling Weapon System Capability

Weapon Systems
Increasingly Software Intensive

Percentage of Functionality Relying on Software

Improved Sensors
Electrical-Optical
Sensor Fusion
Smart Weapons
GPS/Navigation
Multi-Mission Platforms
Situational Awareness
Predictive Maintenance
Digital Flight Controls
Controls & Displays
Heads Up Displays

76th Software Engineering Group
The on-board operational flight software for the B-2 has increased by 67% over its lifecycle as modernization programs have ensured the relevancy of the weapon system for the war fighter.
Need for Speed

• Near peer adversaries are closing the gap
• Speed with discipline enhances security
• Delivery of outdated requirements reduces value
• Fail fast, learn fast
• National security concerns compel us to take action
  – Shorten software release cycles to operate within our adversaries’ observe-orient-decide-act (OODA) loop
  – Identify and attack impediments prohibiting deployment at the speed required to ensure relevance
  – Leverage Agile DevSecOps practices to put operational flight software in the hands of the war fighter sooner

We Must Change the Delivery Cadence of Operational Flight Software
Barriers to Agile DevSecOps Implementation

- Limited Stakeholder Involvement/Experience
- Operator Training Cadence
- OSS&E/Airworthiness/Nuclear Certification
- Developmental/Operational/Flight Test
- Joint Interoperability Test Certification
- Authority to Operate (ATO)
- Collaboration Tools – Network Limitations/Classified Environments
- Tightly-coupled Architecture
- Scarce Tool Chain Support for Legacy Languages
- Overreliance on System Integration Labs vs. Emulation
- Complex Algorithm Development

Operational Flight Software Faces Unique ADSO Implementation Challenges
• Agile DevSecOps requires dedicated stakeholder involvement
  – Program Office/Owning Command
  – Cyber Security Authorities
  – Developmental/Operational Test Community
  – End Users (Pilots and Maintainers)

• Clarification and maturation of requirements
  – Essential for meaningful prioritization
  – Ensures continuous evaluation of relevance
  – Completes the feedback loop

• Stakeholders must be knowledgeable of ADSO methodologies
  – Training
  – Hands-on Experience

Culture Change Across the Entire Spectrum of Stakeholders
Operator Training Cadence

- Pilot training cadence affected by Agile DevSecOps pace
  - Status quo geared toward large block changes
  - Iterative and incremental builds create fewer training requirements per release
  - Minimum acceptable cadence much longer than sprint cycle

- Not all software updates require training changes
  - Identification of Pilot Vehicle Interface (PVI) changes
    - Track PVI changes to understand impacts
    - Potentially bypass training for releases without PVI changes
  - Training system synchronization must be considered

- User representative must be integral to development team
  - Current or former pilot input extremely valuable
  - Most software developers aren’t pilots and vice versa

Operational Flight Software Demands Rigorous Training
• Operational Safety Suitability and Effectiveness (OSS&E)
  – Airworthiness and nuclear certification
• Developmental/Operational Test
  – No substitute for flight test in certain cases
  – Bring DT/OT requirements into development teams – shift left
• Joint Interoperability Test Certification
  – Weapon systems must communicate with one another
  – Net Ready Key Performance Parameter (NR-KPP)
• Look for ways to streamline compliance

Regulations Inevitable for Systems that Fly and Deliver Weapons
Toolchain Procurement and Authorization

• Authority to Operate (ATO)
  – Move to Continuous ATO
  – Authorize the pipeline, not the product
  – Security baked in not bolted on
• Procurement of Software Tools
  – Typical timeline of 6-18 months from need to usability
  – Approval for use process too slow
• There is no one-size-fits-all solution, but…
  – Cloud architectures enable buying power
  – DoD Enterprise DevSecOps Initiative
    • Hardened Software Factory
    • Avoids vendor lock

These Processes Must be Vastly Improved to Enable Successful Implementation
• Many Tightly Coupled Legacy Architectures
• Limited Support for Languages and Hardware
  – Ada, Jovial, ANSI C, assembly codebases
  – MIL-STD-1750A spec processors
• Overreliance on hardware-in-the-loop System Integration Labs
  – Virtual labs leveraging robust emulation enable faster development
  – Development environments must be kept modernized
• Complex Algorithm Development
  – Certain “science project” solutions not easily decomposable
  – Exasperated by lack of modularity in architecture
Takeaways

• Growth of operational flight software in quantity and complexity
• Adoption of iterative and incremental build cycles is vital
• Operational flight software presents unique challenges
• Stakeholders must commit resources to enable success
• Users of these systems require rigorous training
• Regulatory requirements must be considered
• DevSecOps infrastructure requires deliberate investment
• Retooling of legacy system development environments
• Degree of implementation will vary by platform

Implementation of DevSecOps Imperative to National Security