Space C2 Evolution: Adapting Software Development to Foster International Collaboration (and vice versa)

Caroline T. Jones
Ann L. Chervenak
Michael L. Campbell

March 1, 2021

Motivation for International Collaboration on Space Command and Control (Space C2)

- The space domain is becoming increasingly congested and contested
- Both the US and its partner nations are developing capabilities that bring critical services to warfighters to facilitate timely, quality battlespace decisions
- Currently, there is limited collaboration among allied nations
- International collaboration offers many attractive potential benefits including sharing capabilities, services, and data sets
- But there are also many technical and social/cultural challenges
- To be successful, we must seek out opportunities where the solutions to technical and social challenges reinforce each other
- Focus particularly on collaborative capability development
- Solving these challenges will require coordination and collaboration from all nations involved

Explore what Collaborative Space C2 would mean for the US and its coalition partners

Goals and Desired Outcomes of International Collaboration

- The **goals** of international collaboration include:
  - *Facilitating joint software development, deployment, and updates among partners*
  - *Ensuring secure deployment and operation of shared applications on US and partner nation infrastructure*
  - *Meet the needs of all of the partner nations' Space Operations Centers by co-developing products*
  - *Securely sharing assets, including data sets at multiple classification levels*
  - *Improve interoperability of operations, testing, and training*
  - *More efficiently use limited resources via reduced duplication of effort*
  - *Automate coordination and management of assets across the partners' enterprises*
- The **desired outcomes** of international collaboration include:
  - *More effective joint and collaborative space warfighting capabilities*
  - *More responsive development and deployment of new & updated capabilities for the warfighter*
  - *Greater resiliency*
  - *Improved resistance and faster response to cyber-attacks*
  - *Improved cost-effectiveness during development, deployment, and operations*
  - *Improved communication between operators and among development teams*

Challenges for International Collaboration

Technical

- Use of different coding practices, languages, collaboration tools, and standards
  - Differing hardware and software platforms for development, testing and deployment
    - Requirement to share code and data at multiple levels of classification

Policy, Social, Cultural

- Differences in laws, policies, and oversight requirements
- (Human) language differences
- Distance and time zone differences
- Differing national and organizational priorities and risk tolerances

Common

- Differing Information Assurance (IA) Practices and Policies
- Differing perceived threats
- Creating cohesive multinational teams

Solutions to the Challenges

Technical

- Establish Common Collaborative Development & Deployment Environment
  - Enable Interoperability Between Disparate Environments
    - Provide Security, Including Identity & Access Management

Policy, Social, Cultural

- Manage multiple baselines across the coalition
- Balance tradeoff between autonomy and efficiency / effectiveness
- Define policies to govern ownership and control of software and other intellectual property

Common

- Build trust between disparate teams
- Embrace Agile Principles
- Adopt Development, Security, and Operations (DevSecOps)

Technical Solutions

Establish Common Collaborative Development & Deployment Environment

- If possible, create a single environment for development and deployment with common hardware and software standards
- This common collaborative environment may include:
  - Standardized toolsets
  - Well-defined processes and pipelines
  - Integrated security protocols, tools, reviews
  - Development, test, integration, operational environments
  - Sandboxes deployed to each partner for testing of functionality and interoperability
- Provide mechanisms to promote newly developed capabilities to nations' operational environments

Enable Interoperability Between Disparate Environments

- Another approach is to create disparate but interoperable environments
- Consider development strategies such as independent microservices that communicate via well-defined APIs
  - Reduces need for extensive integration between work done by different teams
- Additional testing and time may be needed to ensure that all software features function properly
- Automate testing (wherever practical) in the DevSecOps pipeline
- Utilize common standards and tools to reduce the work required to deploy and update applications for different environments

Provide Security, Including Identity & Access Management

- Establish interoperable standards for Identity and Access Management (IdAM) to enable authentication, authorization, and access control
- Provide encryption for data transfers and stored data
- Support multi-level security (MLS) to access different classification levels
- Consider zero trust models, where devices are untrusted by default and strict security verification is used both within & across a network perimeter
- Avoid coding inflexible security features that are difficult to change as policy and requirements evolve

Policy, Social, Cultural Solutions

Managing multiple baselines across the coalition

- Due to differing requirements, partners may maintain multiple baselines (versions) of the same code base
  - For example, partners may have different policy, Information Assurance (IA), and feature priorities
- The coalition should maintain a “coalition baseline” for as long as possible before individual partner baselines branch off
- As the coalition baseline evolves, branches should ideally incorporate the changes or attempt to merge back into the coalition baseline

Balancing the tradeoffs between autonomy and efficiency/effectiveness

- Desire by individual partners for autonomy is not going to go away and is appropriate
- Can lead to inefficiencies, which often must be accepted
- Agile methodology facilitates early discovery of conflicting priorities and performance issues
- If a partner nation has unique requirements, it can branch off from the coalition baseline
- Alternatively, stakeholders may choose to relinquish some autonomy to serve the overall mission

Defining policies to govern ownership and control of software and other intellectual property

- Conventions, standards of behavior, and processes borrowed from open source communities may be applicable to resolve these issues
- Carefully select open source and commercial software based on their associated license restrictions
  - Even with open source, license restrictions vary significantly
- Source code dependencies should be examined carefully to prevent potential issues, for example multiple versions of a library being used
- Software supply chain should be carefully examined

Solutions to Common Technical and Social/Cultural Challenges

Building trust between teams

- Team members must trust the quality of others' work and their expertise to maximize team and individual contributions
- Avoid favoritism or withholding of critical information among partners
- Agile methodology can build trust through frequent interactions, continuous evaluation of status and priorities, consensus-based decision making, and code review
- Verify development milestones through joint demonstrations and exercises

Embrace Agile Software Development Principles

- Iterative development in small units of work which can be completed in a short time
- Developers and end users communicate early and often
- Differing interpretations of application features or different end user priorities among nations are identified and addressed early
- Collaboratively create user stories, assign tasks
- Mitigate potential issues regarding distribution of important, challenging work among partners
- Schedule daily scrum meetings with flexibility to accommodate time zone differences

Adopt Development, Security, and Operations (DevSecOps)

- DevSecOps pipelines may support the requirements of partner nations (security, deployment cadence, etc.)
- Initially, create individual pipelines for each nation to run their security scans
- Ideally, implement a common DevSecOps pipeline to implement automatic security scans and requirements
- Partner nations run this pipeline before deploying software to their own systems
- Integrate other security steps that cannot be automated (e.g., manual code reviews and interpretation of code analysis results)

Summary and Conclusions

- Explored what Collaborative Space C2 entails for US and Coalition Partners
- Collaboration Goals:
  - Joint software development and deployment on common and disparate environments
  - Improved interoperability, efficiency, security, automated coordination and management, and sharing of assets
- ***Desired Outcome: Joint development of more effective, responsive, resilient, secure, cost-effective space warfighting capabilities***
- Key Takeaways
  - It is essential to address both technical and cultural challenges for the collaboration to succeed
  - International collaboration can complicate distributed software development & deployment, but also enriches the pool of talent and ideas
  - Technology can support building collaboration and trust, but applying technology also requires establishing some trust and recognizing shared interests at the beginning
  - Building software collaboratively and building collaborative teams are different aspects of the same thing
  - Many current processes, such as the agile framework, already lend themselves well to international collaboration, but may need some changes
  - Start with desired end state in mind
  - Start small and build incrementally

Solving the challenges of international Space C2 collaboration will demand the evolution of how we do software development and how we work with our international partners