



An operational perspective on the power, utility, and challenges of exploiting DevSecOps methods

Briefing Date -- 28 August 2024

Lt Col Jeremy "JJ" Homan
CAO 16 Aug 2024

Highest Overall Classification:

UNCLASSIFIED

VIA VINCIMUS



Overview

- Research Purpose & Method
- Acquisition Approaches for Software-based Capabilities
- Lessons from Combat Development in the Field
 - Lessons Learned & Best Practices Vignettes
- Summary of Major Takeaways



Research Purpose & Method

Purpose:

- Inform other DoD practitioners on benefits & challenges of leveraging DevSecOps & agile-software methods to improve their organizations

Approach/Method:

- Experiential study with a focus on lessons learned & recommended best practices from real-world experiences & DoD guidance

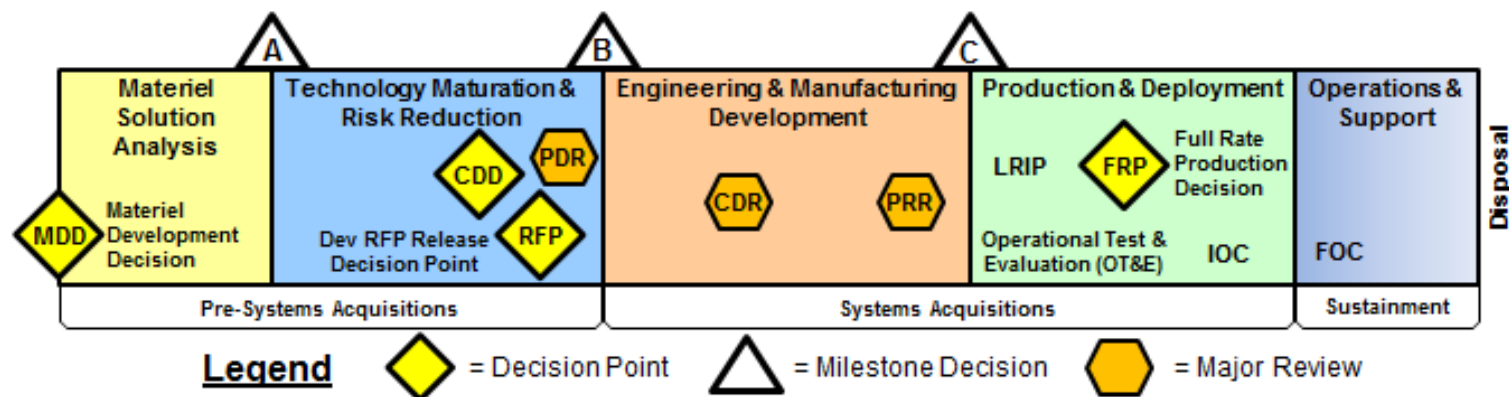
Lens of Author:

- 18-year Air Force servicemember (Operations Research Analyst)
- Ops Test, Assessment, Academia, Program Office, MAJCOM/FIELDCOM experience
- Education in Systems Engineering (Computer Systems) & Business Management
- Current Assignment:
 - Combat Force Enhancement & Innovation at Space Operations Command



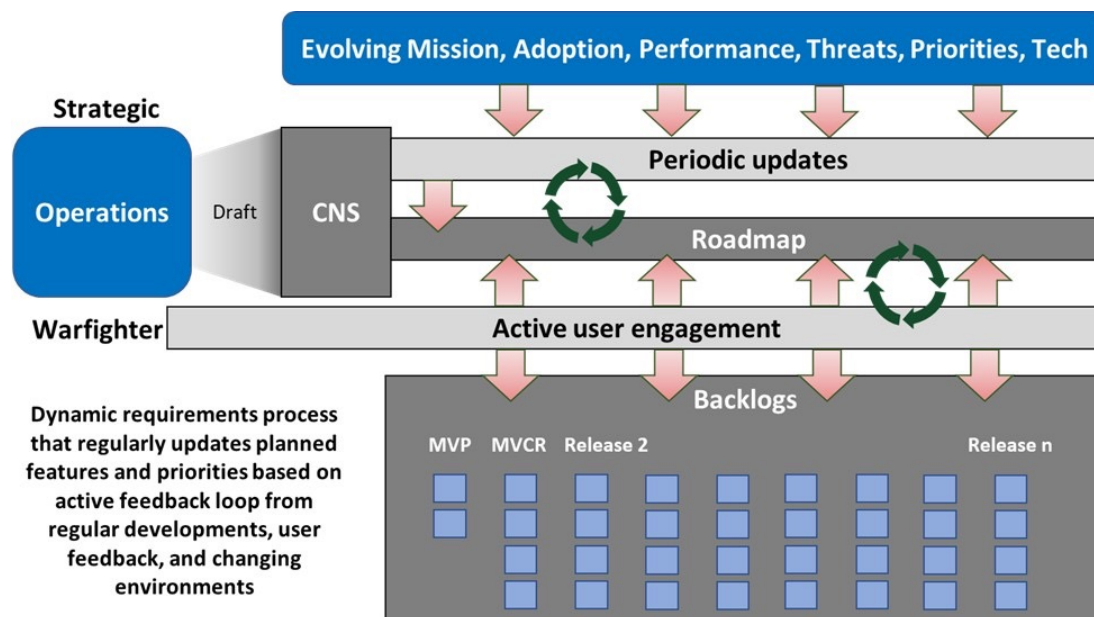
Acquisition Approaches for Software-based Capabilities

Major Capability Acquisitions



- "JCIDS / Traditional"
- Typically HW/SW
- Longer lead/\$\$\$\$
- Vendor delivered

Software Acquisitions Pathway (SWP)



- Newer to the DoD
- Solely SW on dedicated platforms/systems
- Complexity varies
- Rapid/\$\$
- Vendor/Gov delivered
- Emergent cape gaps



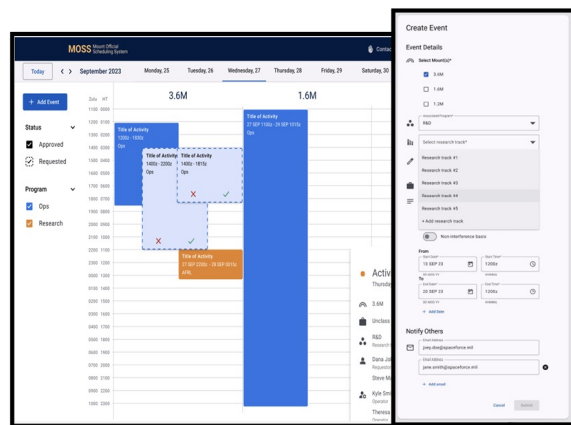
Lessons from Combat Development in the Field

What: Combat Force Enhancement, Innovation, Capability, Digital Tools/Apps/Software products

Who: Combat Development Teams, Supra Coders, Combat Enhancement Teams, Other Stakeholders

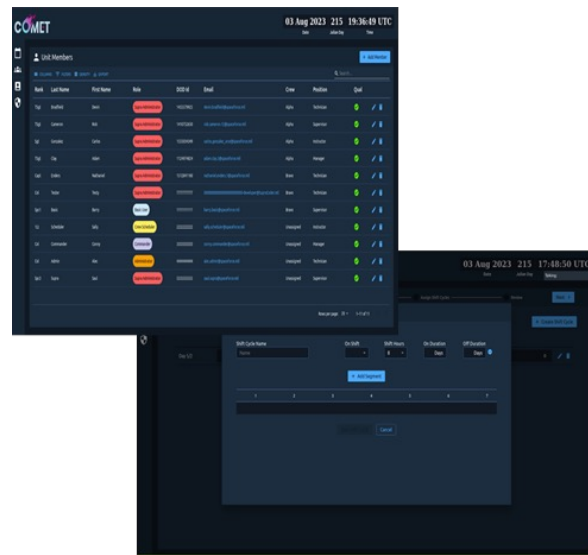
Why/How: Improve the mission through rapid, impactful, and sustainable combat force enhancement

Asset Scheduling Tool

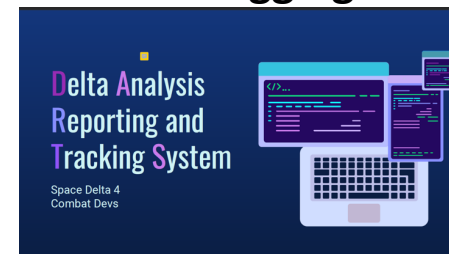


<https://www.spoc.spaceforce.mil/News/Article-Display/Article/3773396/15-sps-innovates-with-mount-official-scheduling-system>

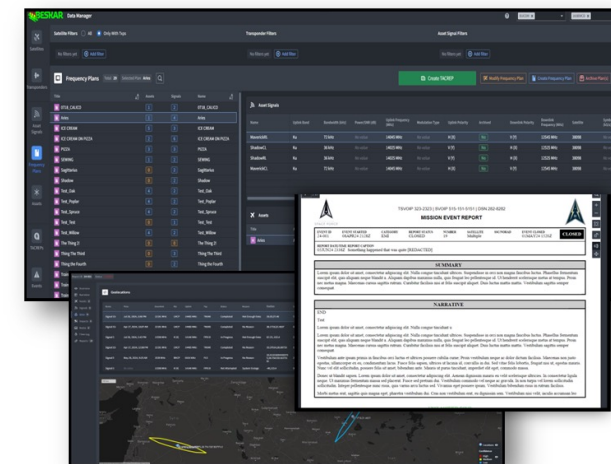
Crew Scheduling Tool



Mission Logging Tool



EMI Reporting Tool





Lessons Learned & Best Practices

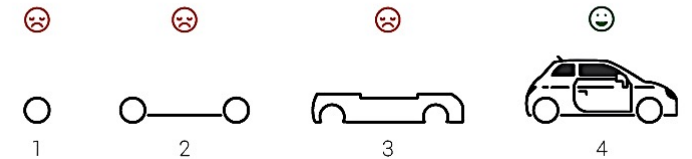
Observation: The minimum-viable product (MVP) convention is used often, but defining what “min-viable” means upfront is rare

Example(s): Space Crew Scheduling Tool, Space Domain Awareness Visualization/COP

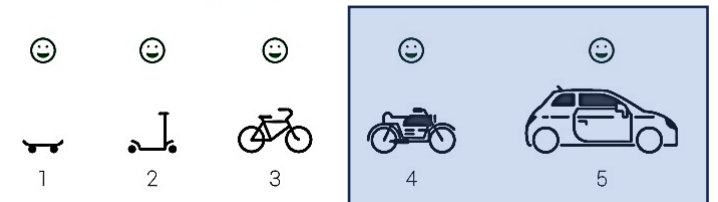
Takeaways:

- Need to “write-down” what the product must do first & agree on the ops-acceptable MVP for fielding (e.g. CNS & SW-ICD) or high risk for scope creep
- Agile does not mean zero requirements
- Need leadership direction (informed by end-user inputs to ensure design for the human is considered)
- Non-functional requirements matter (e.g. releasable to allies, availability, interoperability, usability, etc.)
- “Viable” implies operationally-relevant/realistic (e.g. biking 500 miles is not viable for most folks)

————— How **not** to build a minimum viable product —————



————— How to build a minimum viable product —————



Only "MVPs" for
very long distances



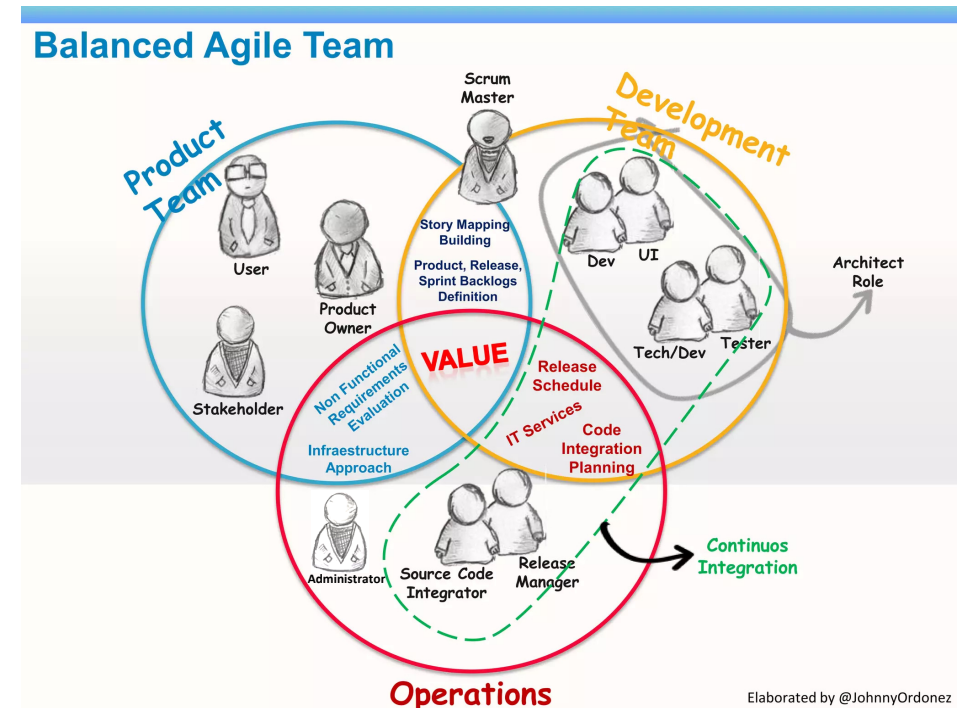
Lessons Learned & Best Practices

Observation: Developing & delivering software apps is often the focus compared to the fully-burdened ops & sustainment phase

Example(s): EMI Reporting Tool

Takeaways:

- “SW is never done” also means “the people are never done”...true CI/CD takes resources
- Even perfectly user-friendly software needs ops & support (e.g. access to help-desk, administrators, ops test, training plan, config/release management, etc.)
- Software infrastructure is not free, even if it appears to be free to the end-user
- Good leadership & management are always necessary even after initial delivery...coding is only part of delivering capability & CI/CD





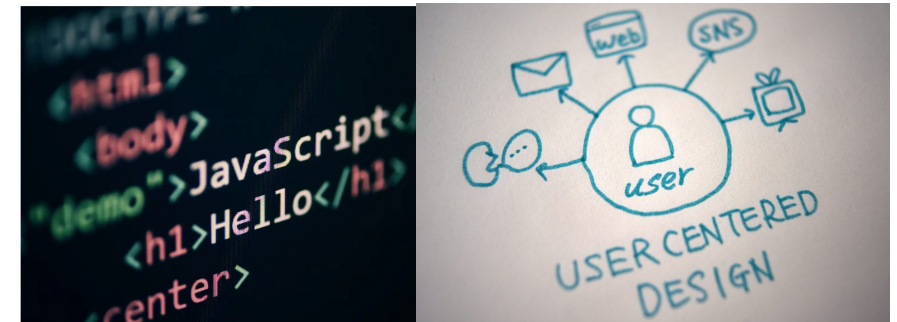
Lessons Learned & Best Practices

Observation: General expectation that coding skills equal full-spectrum software development/management/design skills

Example(s): All digital tools

Takeaways:

- Ability to code vs ability to engineer/design/manage are not necessarily synonymous
- Coding and other skills need regular opportunities to employ, else skills will atrophy...need to cultivate a willing & capable digital workforce
- Balanced software products teams can enable success for junior government developers
- Proficiency with one coding language/tech stack does not equal proficiency for all...tailored training & continuous education opportunities are needed



CREDIT: BIGO STUDIO / SHUTTERSTOCK Credit: Jared S. Smith

FY22	FY23	FY24
Release-10	Release-11	Release-12
Capability X – Epic X-1		
Capability X – Epic X-2		
Capability X – Epic X-3		
Capability X – Epic X-4		
	Capability X	
Capability Y – Epic Y-1	Capability Y – Epic Y-3	
Capability Y – Epic Y-2	Capability Y – Epic Y-4	
Capability Z – Epic Z-1	Capability Z – Epic Z-2	
	Capability Z – Epic Z-3	
		Capability Z

Mission A
 Mission B
 Mission C

Greater detail in the nearer-term because more is known and less is uncertain Less detail over a longer timeframe because less is known and more is uncertain



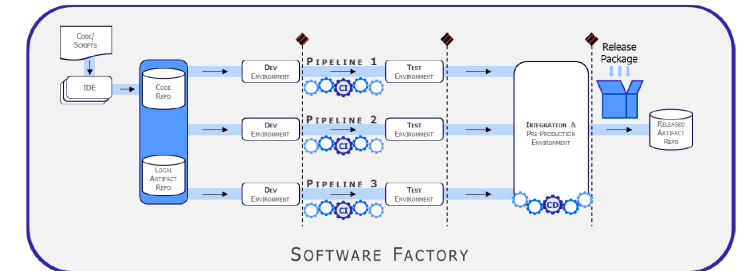
Lessons Learned & Best Practices

Observation: SW Factories purpose & role are often misunderstood by workers, customers, and other stakeholders

Example(s): Crew Force Management, Crew Scheduling, Mission Logging Tools

Takeaways:

- SW Factory model varies by organization with dedicated roles/requirements... "open/free" SW Factories to any developer or user are rare
- Unchecked platform proliferation is costly and drives a fractured portfolio that is difficult to sustain & learn
- Better "containerization" can help, but it is not magic; platform choice matters and re-factoring/re-deploying code takes time, money, and skill
- SW Factories need the 4P's (Platforms, Pipelines/Products, Process, People)





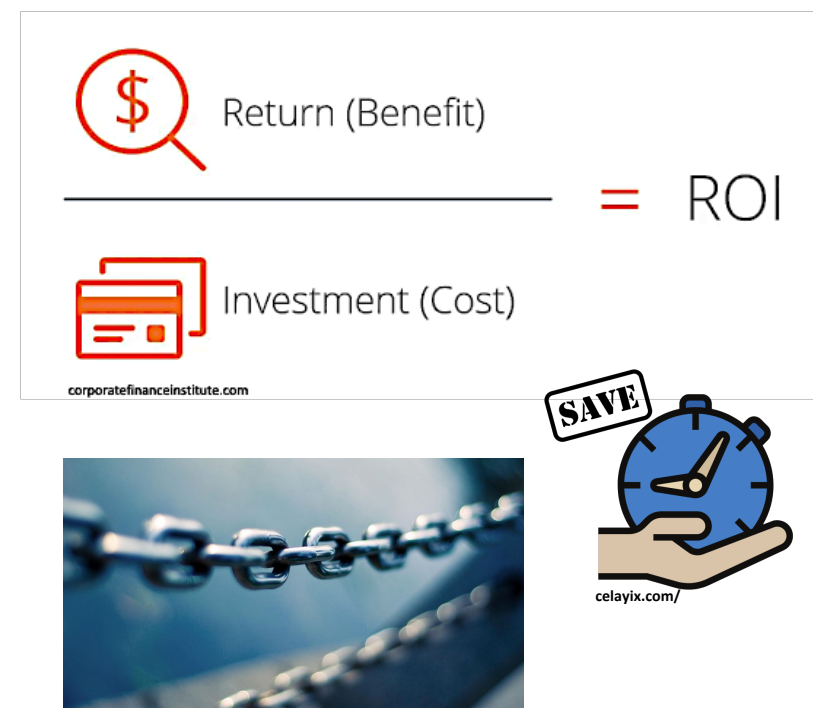
Lessons Learned & Best Practices

Observation: Common challenge for ops community to articulate the return on investment (ROI) for organic digital tools

Example(s): All digital tools; EMI Reporting Tool

Takeaways:

- Avoid buzzwords & best-case scenarios to describe the ROI...focus on a defensible impact to real-world ops
- Majority of ops-focused tools will save planning steps/time, reduce errors, shorten kill-chain time to decision making, allow for predicative analysis, save money, increase asset utilization, etc. ... promote "S.M.A.R.T" metrics
- Near impossible to state the ROI without a defensible baseline of the current state (e.g. process mapping)
- Saving hours of someone's time does not always equal manpower savings...avoid hopeful ROI
- Estimated benefits are a great start, but real-world/exercises can better demonstrate a digital tools ROI for ops





Lessons Learned & Best Practices

Observation: Organic software development is nascent for most of the operational community and navigating the PBBE & acquisition rules/policies can be difficult

Example(s): All digital tools

Takeaways:

- Majority of software regulations & policy focused on JCIDS “traditional” Major HW/SW systems model
- Software applications/tools/products do not always equal an Information System or Weapon System
- Understanding cybersecurity & RMF accreditation processes is critical for organic software development
- Understanding colors of money/SW acquisitions rules is important even in ops
- Data/IP rights can matter for future sustainment/integration needs
- Software capacity vs Product-based contracts can benefit
- Commission on PBBE Reform is taking a hard look at software development & sustainment policies/methods





Summary of Major Takeaways

- The SW Acquisitions Pathway presents new opportunities for rapid force design & improvement, and increased participation by stakeholders & end-users
- Capability gaps & requirements still matter for organic agile software
- Investments in digital infrastructure & training have created new opportunities for digital improvement by a willing & capable workforce
- No free lunch when it comes to software
- Software is never done, and neither are the resources required
- Demonstrating digital ROI starts with a defensible performance baseline
- Organic software development within the ops community is still in the early stages and needs continued investment & dedicated resources
- Early & often collaboration and sharing of digital lessons learned & best practices benefits everyone



References

- Software Acquisitions Pathway, <https://aaf.dau.edu/aaf/software/>
- Major Capability Acquisitions, <https://aaf.dau.edu/aaf/mca/>
- Air Force Software Ecosystem, <https://software.af.mil/>
- DoD Enterprise DevSecOps Fundamentals, March 2021, <https://dodcio.defense.gov/>
- Commission on PBBE Reform, March 2024 Final Report, <https://ppbereform.senate.gov/>

The logo is a light blue triangle with a grid pattern. Inside the triangle, there are several red lines and arrows pointing towards the center. The words "SPACE" and "OPERATIONS COMMAND" are written in a light blue font at the bottom of the triangle.

Discussion - or - Questions?