Where Our National Security Begins...
What is an Industry Advisory Working Group?

Industry partners self-organizing to discuss matters of mutual concern and provide pragmatic recommendations regarding the industrial base.

An IAWG is...
- Volunteer-based
- Strategic in nature
- Objective (pros & cons)
- Open to participation
- Company-agnostic
- Problem-centric
- Focused on outcomes

An IAWG is not...
- Sponsored by the government
- Restricted in participation
- Proprietary
- A pursuit/capture venue
- A shaping & positioning opportunity
- A venue to recommend products
- An open ended discussion forum

Future Business Models are of Strategic Importance to the Industry Base
**NRO Industry Advisory WG (IAWG): Charter & Objectives**

- **Mission:** Help NRO and Industry jointly achieve transformation objectives
  - Identify business models that will support government and industry objectives
  - Identify potential pitfalls and recommend potential solutions

- **Charter:** Provide expert industry resource and sounding board focused on:
  - Business aspects of emerging acquisition models used to acquire software services
  - Ramifications of componentizing software applications,
  - Benefits accrued to the government & industry,
  - Intended and unintended consequences against the industry base,
  - Limitations and viability as a reasonable course of action

- **Objectives:**
  - Provide strategic industry input to a changing acquisition landscape
  - Provide an objective and neutral venue for discussing approaches to business models
  - Foster effective communication between government and industry leadership

**“Action Team” approach:**
*Deliverable-based, issue-specific, and pragmatic*
Observations on Agile & DevOps:

- It is a culture shock to developers, program managers, SED, and MOD
- It has great promise but faces major culture, training, and business process obstacles

Questions:

- Is DevOps fundamental to NRO cloud adoption and Future Ground Architecture?
- How does DevOps impact/change the NRO Operating & Business Models?
- What Contracting approaches are optimal for Agile & DevOps?
Introduction & Overview

I. Agile & DevOps Overview: Value delivery paradigms

II. Benefits: Why should NRO implement Agile & DevOps?

III. Adoption: What does NRO need to do to accrue the benefits

IV. Program examples & Contracting practices

V. Lessons Learned and Recommendations
Agile & DevOps Overview

What they are, What it means

Agile is a Rigorous, Proven Approach that prioritizes activity appropriate to its impact and importance to “working” mission capability
Agile Principles*  
Applicable to NRO Software Development

• **Individuals and interactions** over processes and tools. Teams of people build software systems, and to do that they need to work together effectively. This includes all disciplines working together towards common end goals and not handoffs of disconnected deliverables.

• **Working software over comprehensive documentation**. Working software producing desired outcomes is the goal. Documentation is important as a capture of the working software but not as isolated, unmaintained deliverables.

• **Customer collaboration** over contract negotiation. Frequent feedback from customers is the most reliable way to determining if the right thing is (still) being built. Having a contract with a customer is important to establish boundaries, but the goal is working software satisfying a need.

• **Responding to change** over adherence to a plan. Agile software development methods are focused on quick responses to change and continuous development.

* As adapted from [http://agilemanifesto.org/history.html and http://www.ambysoft.com/essays/agileManifesto.html, among others]
What is “DevOps”?  

DevOps is a method of collaborative effort among a group of stakeholders to more efficiently and quickly create value

• Maximize value across enterprise, not just locally for parts of enterprise
• Impacts definition of value: not just developing software and handing it off to someone else

DevOps is most typically applied to software but is also applicable to other efforts that create value for eventual consumer

Three Aspects of DevOps: Flow, Feedback, Learning

• Flow: how efficiently and quickly we go from need to delivery, i.e. Dev to Ops
• Feedback: how quickly we find problems or where our proposed solution will miss the mark, i.e. Ops to Development
• Learning: how we find better ways to do things, even in areas we think we do well already. Examples may include engineering & transition processes, project planning, procurement, development & delivery.
DevOps: Working Definition

DevOps (clipped compound of “development” and “operations”)

- Culture & Practice emphasizing collaboration and communication between stakeholders, including consumers, developers, operators, and testers, to improve software delivery and infrastructure changes.
- Describes techniques for automating repetitive tasks within the software development lifecycle (SDLC) including software build, testing (including functional, nonfunctional, security and compliance) and deployments.

Lot of tools involved with goal of maximum consistency, done via maximum automation and minimum manual effort:

- Code - Code development & review, continuous integration tools
- Build – Version control tools, code merging, build status
- Test – Test and results determine performance
- Package – Artifact repository, application pre-deployment staging
- Release – Change management, release approvals, release automation
- Configure – Infrastructure configuration and management, Infrastructure as Code tools

A convergence of culture, process and automation/tools to achieve faster delivery of robust, correct features in small batches from idea to operations.
As a Concept, Agile & DevOps Aren’t New

- **Agile & DevOps have roots in Lean manufacturing (Toyota, 1980s)**
  - Reduce lead time from concept to customer value
  - Increase communications among stakeholders
  - Limit work in progress (WIP)
  - Reduce batch sizes (e.g. size of change)
  - Reduce number of handoffs
  - Identify and respond to constraints

- **Agile is a collection of principles and common practices that emerged from waterfall, spiral, and RUP engineering methodologies.**
  - Waterfall and Spiral were slow to bring value, with multi-year timelines for requirements decomposition, prototyping, and full scale development.
  - Agile reduced size of “Work in Progress”, reduced cycle time for requirements decomposition, development and deployment.
  - Agile Manifesto (2001) applies Lean Principles to software development

- **Despite the rigor of Agile process, some in government perceive it as lacking rigor in contrast to Waterfall/Spiral processes**
  - Planning sessions converge current needs to match evolving understanding
  - Automation ensures build, test, and deploy processes are done consistently throughout the software lifecycle
Agile & DevOps as a Value Delivery Paradigm

- Scaled Agile describes what is needed
- Lean-Agile describes the minimum viable product & how to build it
- How do we deliver value faster to the customer? “DevOps”

- Continuous everything (testing, integration, delivery, ...)

Customer Need → Scaled Agile → Lean-Agile → DevOps → Continuous VALUE Delivery

Planning/Goal Setting → “Minimum Viable Product” → Delivery to customer
DevOps Principles: Continuous Testing & Learning

**Continuous testing**
- Initiated by code commits
  - automated regression testing
  - functional testing
  - performance testing
  - conformance testing for security and coding standards
- Goal is to find problems early: fixing code written last week is easier than fixing code written with bug last year

**Continuous learning**
- Review effort and results
- Look for improvements
- encourage experimentation and innovation
DevOps Principles
Continuous Integration (CI)*

- Continuous Integration is the practice of merging all developer working copies to a shared mainline, continually merging source code updates from all developers on a team
  - With CI, the “system always runs,” meaning it’s potentially deployable, even during development of the solution.
  - CI is most readily applied to software solutions where small, tested vertical threads can deliver value independently.

- CI avoids ever-increasing set of nonobvious integration disconnects that traditionally do not surface until mammoth, stressful integration events
  - Complex systems with mechanical subsystems, software, electrical/electronic subsystems, suppliers, subassemblies, and the like require additional considerations beyond the scope of this report.
  - In general, however, integrating and testing collaborating components together frequently is the only practical way to fully validate a solution.

- CI offers real-time window into actual state of software system and associated quality measurements, throughout project lifecycle
  - Teams need a balanced approach to build quality in AND receive fast feedback from the integrated increments

- CI enables consistent alignment between code base and the integrated software product

- CI enables automated Integration processes to ensure build-consistency as software grows
  - CI platforms have technical constructs and must be continuously integrated to prove new functionality.

DevOps Principles
Continuous Delivery & Deployment Pipeline

Continuous delivery is enabled through the deployment pipeline. The team ensures every change is capable of being deployed to production, making the decision to deploy a business decision. Components include:

- **Visibility** – All aspects of delivery system including building, deploying, testing, and releasing are visible to every team member

- **Feedback** – Team members learn of problems as they occur so they are able to fix them quickly, rather than after the fact

- **Continuous deployment** – Reduces time to mission by using a fully automated process (with possible manual go/no-go decision) to deploy and release any version of software to any environment.

- **Reduced Complexity** – Simplifies, standardizes and executes releases with minimal errors. Automates complex workflows required for software movement between development, test and production environments and removes manual interactions (human error)
Getting to Continuous Delivery

- Utilize Continuous Development & Integration to maximize value, minimize risk
  - Lot of testing, but can be limited by physical test servers available.
  - Bring in other resources as needed

- Implement IaaS/PaaS cloud provisioning models and management practices
  - Deploying exact replicas of test servers on demand: repeatability, consistency, speed
  - Bring in other resources as needed: minimize “blockers” to continuous flow

- Re-tool for Speed: Front- and Back-ends of the Systems Engineering “Vee”
  - Evolve SE RFC processes from “Waterfall Requirements Decomposition” to “Continuous Requirements & Product Backlog Grooming”
  - Adapt/Evolve NSIS delivery, Transition to Ops, and Security A&A processes
# DevOps Principles

## Platform as a Service vs Infrastructure as a Service

<table>
<thead>
<tr>
<th>Capabilities</th>
<th>IaaS Implementation</th>
<th>Paas Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Privileged User Access</td>
<td>Required</td>
<td>Not Required - Only need access to PaaS</td>
</tr>
<tr>
<td>Scaling</td>
<td>DevOps team is responsible for scaling operations</td>
<td>Scalability is automatically managed by PaaS policy.</td>
</tr>
<tr>
<td>Communication</td>
<td>DevOps team responsible for configuration valid communication interfaces</td>
<td>Communication endpoints managed by PaaS</td>
</tr>
<tr>
<td>Patching</td>
<td>Patching is managed by DevOps team</td>
<td>Patching is provided by PaaS</td>
</tr>
<tr>
<td>Container Lifecycle Management</td>
<td>The DevOps team manages IaaS instances used for service deployment</td>
<td>Container Lifecycle managed by PaaS</td>
</tr>
<tr>
<td>Continuous Delivery</td>
<td>DevOps engineers need to install and configure build-automation tools and integrate them with a project repository to provide continuous delivery</td>
<td>Delivery Process and Automation Tools provided by PaaS</td>
</tr>
<tr>
<td>Application Runtimes</td>
<td>The DevOps team needs to configure an application runtime on IaaS instances.</td>
<td>Application runtime automatically deployed in a PaaS container.</td>
</tr>
</tbody>
</table>
Agile & DevOps Benefits

Why NRO needs to implement & adopt
Why DevOps at NRO?

- **Improved Code Quality**
  - Iterative development and smaller code segments make it easier to detect and resolve code defects
  - Code is fixed rapidly before it can be duplicated elsewhere

- **Reduced Time to Deployment**
  - Developers are better equipped to uncover and correct defects from code written several days ago than several months ago
  - Testing, including security, is automated, repeatable, and consistent
  - Time to ATO: automated security artifact generation and DAA review

- **Eliminating Process Handoffs as a source of waste**
  - Eliminates monthly review boards (Work In Progress can be tested and deployed when ready)
  - Product backlog and prioritization allows dev staff to address ratified requirements without additional process time (CCB/ERB)
  - Validating automation mitigates redundant process iterations
What are the potential benefits to NRO? Government Relatable “ROI”

- **Shorter Development Cycle**
  - Promotes a culture of increased collaboration and communication between the development and operations teams

- **Increased Release Cadence**
  - Waterfall based releases of 6-18 months reduced to daily or hourly cycles

- **Improved Defect Detection**
  - Iterative development and modular programming breaks larger codebases into smaller manageable features, making it easier to detect code defects.

- **Reduced Deployment Failures and Rollbacks**
  - Taking an operational viewpoint into account during development, combined with improved defect detection significantly decreases the number of pre- and post-deployment issues.

- **Efficient Operations**
  - Agile and DevOps are built on LEAN principles. The reduction of waste, such as unnecessary waiting times or process overhead through continuous improvement, is a core principle.

- **Achieve Greater Visibility and Transparency**
  - Deployments produce actual working software every few weeks, which enables the demonstration of software to stakeholders. Managers see rapid, weekly progress, and can quickly determine whether the product will meet expectations and quality.
DevOps is more than the use of tools / services to automate workflows:
- Requires the evolution of an organization’s culture and business processes on both sides of the Government / Contractor paradigm
- Emphasis on close collaboration between those who perform development, operations, and security

Organizational culture must evolve to enable DevOps
- Best practice is to start small but significant, learn, grow
- Build firm foundation with growing body of success, not Big Bang
- Overcoming the reluctance to automate the build/test/delivery of software
- Developers must embed security compliance behaviors into their DevOps teams

NRO operating model requires changes for DevOps
- Waterfall based NSIS approach appropriate for systems with well-known requirements and well-known solutions, but not appropriate for software intensive activities with evolving requirements
- “One size fits all” approach for assessing operational readiness, performing A&A, and change management limits the ability to deliver and sustain software services in a more effective manner

Recognize learning component: evolving effectiveness vs designing perfect processes
- Ex: evaluating which infrastructure is appropriate for program operations (Cloud/Virtual/Bare Metal) - One size doesn’t fit all
- Ex: how NRO/Developer community adopts & uses GFE where appropriate

Evolving culture and business processes with respect to software development and sustainment will facilitate NRO benefits from DevOps
Must we Choose Between DevOps OR Waterfall?

“Bimodal IT”* - the practice of managing two separate but coherent styles of work

Mode 1 (Waterfall)
- Single sequential progression from requirements to development to test to deployment
  - Typically requires long timelines (e.g. years) from problem to delivered solution
  - Concentrates on development of many (possibly diverse) capabilities to be delivered at the same time (or as a small number of drops)
- Baseline for Government programs with which any other approach must likely combine and coordinate

Mode 2 (DevOps)
- Evolving requirements prioritized as backlog
- Multiple cycles (e.g. sprints) to develop, test, and deploy additional capability as capability becomes available
  - More frequent delivery of more focused capability
  - Delivery of small batches, fully tested and avoiding accumulation of issues that harder to isolate in larger batches
- Goal is rapid delivery of value to stakeholders
- Numerous methodologies, e.g. Agile Scrum, SAFe

* Source: Gartner
Recommendation: Elect a Hybrid Approach

DevOps AND Waterfall

- Recognizes transition state between legacy system development and emerging framework and services development
- Combines modes for optimum benefit
  - DevOps mode can provide capabilities scheduled for waterfall delivery
  - DevOps contributions to waterfall delivery can have benefits of small batch size
  - Do not need Big Bang adoption of DevOps to start seeing DevOps value
  - Recommend starting DevOps with small but significant project
    - Show value, Learn, grow
    - Needs to be a real program… “pilot” project should mean “first” not “trial”
  - Migrate larger Waterfall development to appropriate mix with DevOps
- Challenges:
  - DevOps delivery speed can overwhelm Waterfall transition processes
  - Waterfall governance models tend toward “one size fits all”
Agile & DevOps Adoption

What NRO needs to do to accrue the benefits of Agile & DevOps
Finding: Industry is Embracing Agile and DevOps Despite Perception of Government Reticence

- NGA is providing contractors the ability to realize benefits of Agile (& DevOps) via frameworks such as Scaled Agile Framework (SAFe). Additional frameworks are available for adoption.
- GED Primes / Programs are implementing Agile & DevOps projects and see business benefits
  - New Projects: Agile Masters are training program teams and SETA's & Government PMO's
  - Existing Projects: Flexibility by COTRs enables the learning curve for the teams
- C2S and NRO S2P provide the tooling needed to manage Agile / Dev Ops (eg. Jira, Confluence)
  - Scaled Agile often requires “Premier Class tools” not yet available as services (eg. Version 1)
  - Tool choices are often preference of Agile Masters and program architects – They like tools they are familiar with
- Agile at scale is maturing – metrics, Systems Engineering, EVMS and other techniques have matured through use and business need
- DevOps being tried / deployed at factories – not yet mature – but lessons being learned quickly
- Waterfall process “culture” constrains speed to mission available from Agile & DevOps
  - Transition to Ops & NSIS delivery models are blockers to Continuous Deployment
  - Traditional Waterfall requirements decomposition constrains Agile Release / Sprint speed
What does NRO need to do to make DevOps work?

- **Leadership Commitment to Changing Culture**
  - Must take place at the Director level and include all echelons
  - Recognize conflict between Waterfall & Agile/DevOps approaches
  - Ensure MOD, Security, and SED share in **accountability** to accelerate speed to operations. There is a mission opportunity cost to delay.

- Adoption requires ongoing collaboration between:
  - Acquisition
  - Program Management
  - Developers
  - Infrastructure Providers (NISP)
  - Software Service Provider (NASP)
  - Security

- Start with Program Development environments, labs, & factories
What does NRO Acquisition need to do?

**Acquisition**

- RFP language: encourage industry use of IaaS/PaaS from NISP/NASP and require explanation of value differential for not using GFE environment
- Language includes need to bid agile approach using DevOps best practices
  - Avoid asking for agile processes but requiring waterfall deliverables
- Contracts need to require end results, not instructions on how to do things
  - Are SOO/PWS more applicable than SOW?
- Contract needs to reflect how developers get to picture of the end state
  - Importance of defining collaboration expectations, expected communications mechanisms
  - Specify details on the collaboration required between developers/ops/security and all other stakeholders as part of the solicitation

*Emphasize speed to delivered capability without directing implementation*
What do NRO PMs need to do?

Program Management

- Identify value stream, stakeholders, and process which will identify who has to collaborate with whom and when
- PMs must be Agile/Lean trained, preferably DevOps advocates
- Manage via Scaled Agile Framework (SAFe) Lifecycle - minimize handoffs
- Set guidance for needed activities [“managing” these is more local than PM]
  - Swarm when something breaks
  - Scrums and Scrums of Scrums on regular basis
What do NRO Developers need to do?

Developers

- Collaboration with security and other compliance regimens are built into the beginning of the process
  - Work from beginning to build security requirements into the program
  - Collaborate with Security to develop automated artifacts that are acceptable to evaluate mission systems for accreditation.
  - Conform to relevant security processes such as ICD-503
- Remain responsive and responsible through deployment
- Ensure Deployable code is always available
- Plan for short cycle, continuous small updates
- Take responsibility for writing tests that fully cover code
- Ensure documentation and progress through Development Cycle visible to all
- Maintain responsibility for continuity of operations in collaboration with IaaS and PaaS providers
- Allocate periodic sprints to burn off prioritized technical debt (e.g. DR workoff)
What does NISP need to do?

**Infrastructure (NISP)**
- Provide common highly available IaaS environment
  - Across multiple infrastructures
    - Bare Metal
    - Virtual Environments
    - Cloud Environments
  - Across multiple locations
  - Across multiple networks
- DevOps team: developers for the developers
  - Maintain infrastructure
  - Be aware of advances and incorporate into infrastructure
- Dev, test, and prod environments should be as nearly identical as possible
  - Standard, repeatable configurations under configuration management
  - Ideally single enterprise repository of everything under CM
What does NASP need to do?

- Provide common, highly available PaaS
- Provide Services appropriate to locations and environments
- Maintain consistency for Common Services across environments
- Provide consistent version control across all versions of applications
- Daily backups; resiliency and disaster recovery
- Evangelize proper use of frameworks
- Capture lessons learned
- Provide Subject Matter Expertise on services and best practices
- Evolve Frameworks based on new mission requirements from programs adopting and reusing the framework(s) and/or applications.

Same design patterns apply to both Mission Applications and DevOps Common Services.
What does NRO Security need to do?

**Security**

- Must be part of end-to-end process and not put off until expecting to deploy
- Recognize the mission cost of manual accreditation processes & timelines
- Work from beginning to build security requirements into program’s process
- Maintain continuous collaboration with developers for code/tool approval
- Collaborate with Mission Owners to develop automated artifacts that are acceptable to evaluate mission systems for accreditation.
- Embrace the “continuous” spirit of Dev/OPS – enable Xacta “type accreditations” where feasible
- Must accept automated compliance checking of all software commits to CM
  - Limit manual reviews to things cannot (yet) automate
- Ensure Programs conform to relevant security processes such as ICD-503
  - ICD-503 process is automated to allow continuous security testing and integration beginning in development
  - Automated Security Scans connected to NESSUS
Agile & DevOps Adoption Impacts NRO Business (Industry) AND Operating (Government) Models

- Agile & DevOps redefine what is meant by “requirements”
  - Requirements become a collection of agreements between product owner and developer over course of development
  - User feedback validates or corrects what has been built (delivered outcomes)
  - How do we measure this as progress?

- Contracting approaches must evolve to account for DevOps
  - Outcome based contracting
    - Deliver desired solution (“working software, met requirements”) vs. list of CLINs
    - Focus on delivered capability, speed & total price as opposed to FTE labor rates
  - Incentivize collaboration and timely, accurate, secure delivery
  - Incentivize risk taking and fast failure on SMALL THINGS to LEARN FAST
  - Concept idea: Payments based on development velocity
    - Developer/PM/PO bid and assign points for sprint
    - Developer is paid on delivery and commitment accuracy
    - Identify and encourage stretch goals
Agile / DevOps:
Contracting Practices & Program Examples

NGA Case Study: Agile Web Presence
NRO Case Study: Jarvis Software Services Platform
Agile DevOps Contracting Challenges & Questions

- Government Observation: Government not trained or experienced in DevOps contracting
- Industry Observation: Contracts language, Award Fee approaches can dis-incentivize
- Assumption: Government requires Deliverable based Solutions vs Labor Hour FTE
- Question: How do you pay for speed?
  - Incentivizing sprint team velocity improvements?
  - Incentivizing accurate estimation of velocity (story points) and delivery?
- Question: How to Account for Development vs “Support” functions?
  - Engineering/Development = Scrum Teams
  - Support = CM, CDRL management, Security etc
- Key Concept is “Time boxing”: schedule and resource loading as independent variables
  - Sprints and Epics are fixed duration (i.e. schedule is fixed)
  - Resources over a fixed duration are known
  - Capability delivered is dependent variable…how far down the priority list did you get?
  - Success based on a ratio: maximizing innovation or capability delivered “per” dollar
- IDEA: Re-introduce Cost as an Independent Variable (Govt states price)
Purpose: Provide enterprise shared software services and offerings to reduce cost and schedule of customer application developers

Scope: Platform development for the NRO Application Service Provider (NASP)
  • Architecture, engineering, and O&M (ITIL Tiers 1-3) functions
  • Enterprise Services, LLC is prime with multiple subcontractors and vendor support

Jarvis has four primary activities
  • Development of new platform services (primarily COTS products)
  • Integration of 3rd Party services onto the Platform
  • Operations and Maintenance of the Platform
  • Engagement and advocacy for adoption for other IC elements
Jarvis Contract Approach

- Contract structure: CPAF with IF
  - Cost plus allows for program flexibility while ensuring required deliverables
  - Award fee given every 6 months based on criteria

- Incentive fee given every 6 months based on ability to deliver 13 services
  - Equates to 26 new services annually
  - Current velocity delivers ~9 services per quarter, so far exceeding expectations

- Services align to the S2P architecture
  - Product Backlog priorities maintained by government
  - Release Plans developed jointly by government and Jarvis prime
  - Technical definition of “service” is loose…in addition to COTS/OSS, can include technical debt retirement, O&M enhancements, or upgrades to existing COTS/OSS
Jarvis Agile & DevOps Approach

• (U) Jarvis team practices an Agile (Scrum) development methodology
  • 7 technical Scrum teams matrixed with program support

• (U) Services are scheduled and delivered via Agile Releases
  • Per Agile methodology, specific services by Release are subject to change

• (U) High Level Release Schedule (4 Releases, per calendar year)

<table>
<thead>
<tr>
<th>Event</th>
<th>Length (Business Days)</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release Planning</td>
<td>3</td>
<td>Engage with stakeholders to determine content of the Release</td>
</tr>
<tr>
<td>Sprint 1</td>
<td>15</td>
<td>Develop, integrate, test and deploy Release content</td>
</tr>
<tr>
<td>Sprint 2</td>
<td>15</td>
<td>Develop, integrate, test and deploy Release content</td>
</tr>
<tr>
<td>Sprint 3</td>
<td>15</td>
<td>Develop, integrate, test and deploy Release content</td>
</tr>
<tr>
<td>Sprint 4</td>
<td>15</td>
<td>Develop, integrate, test and deploy Release content</td>
</tr>
<tr>
<td>Release Retrospective</td>
<td>2</td>
<td>Reflect upon what went well and what didn’t, on order to improve upon the next Release</td>
</tr>
</tbody>
</table>
Jarvis Results Snapshot
Software Services Platform (S2P)

- Customer base growing >40% CAGR quarterly: from 800 to >5000 in 15 months
- 13 services in first 3 months compared to 2 services in previous 2 years
- Over 250 developer tools available within first 9 months of contract
- Developers typically provisioned in less than 2 weeks

Initiatives include:
- ATO automation
- Software Supply Chain automation & life cycle security
- API Integration and Continuous Integration
NGA Exemplar Contract: Agile Web Presence (AWP)

Scope

• NGA PoR responsible for designing, building, and maintaining NGA’s corporate web presences across all security domains including NGA Globe, NGA.mil and CAPNET (Congressional portal)

• Effort includes O&M support to the applications, cloud infrastructure engineering to build and deploy the applications in C2S, and NGA’s UC2 cloud

• Included an Enterprise Search (ES) effort for 24 months to reengineer legacy ES solution to deliver 7M+ GEOINT records including imagery, reporting, and foundation GEOINT products
AWP Contract Approach

• $38M Small Business set aside contract awarded in 2015 to Octo Consulting Group, Inc. to deliver firm-fixed price Agile software development, O&M, and cloud engineering capacity to the IT Services (T) directorate at NGA

• Government used Scaled Agile Framework (SAFe) to provide process rigor and approach to Government/Contractor integration

• Contractor receives a fixed price payment to deliver a set velocity (70 pts) of development capacity every 2 weeks per Scrum Team

• The Government works hand-in-hand with contractor to iteratively plan for upcoming sprints and releases, managing a roadmap of features, often bucketed into overarching release “themes”

• Contractor delivers 3 development Sprints and an integration and planning sprint every 8 weeks with releases into test, staging and production environments 2-3 weeks later
AWP Results Achieved

• Successfully met or exceeded all contractual performance requirements including velocity/sprint (pts/sprint), system availability (% uptime), and incident response time (# of hours per levels identified in PWS)

• Government significantly reduced overall contract cost while maintaining capacity and affording contractor significant staffing flexibility (and modest profitability) by building software at an unclassified level and moving code into classified, staging and production environments

• One of first NGA applications to deploy into C2S and receive authority to operate

• Led major transformation of legacy SharePoint-based application architecture towards Node, Angular, and Loopback based open source architecture that promotes code reuse and eases development

• Proved an innovative small business can move a legacy large business contract based on local, bare-metal infrastructure into a cloud-based development and production environment with no impact to development cadence or system availability
Agile & DevOps Contracting: Industry Observations

Observation: Industry has found ways to implement Agile & DevOps internally despite government’s reliance on waterfall processes for most programs. There is good to leverage.

CP or FFPLOE Contracting approaches can work…but differ in application & management
- CP is workable with government controlling priorities and industry controlling velocity and capacity
- FFPLOE is workable with govt/industry jointly defining Story Point LOE with fixed capacity
- Minimum Period of Performance is 1 year. 6-month PoP found to be disruptive and usually lacks business case

Implementation: Scrum Team & “Support function” metrics (fixed cost, fixed time)
1. Capacity delivered
2. Quality/defects
3. Availability & incidence response

Incentivizing speed? Consider Incentive Fee, whether using FFP or CP approaches
- It allows industry to say “yes” instead of trading requirements.
- For example, higher velocity against fixed cost/time means early delivery and govt/industry split the delta 50/50. Or higher velocity means more off the product backlog and govt/industry split the delta representing the sprint points.

Key Considerations in Agile & DevOps contract implementation
- Active management of staffing mix
- Government partner..no playing “gotcha” on speed #s.
- Govt must focus on product roadmap & priority. Govt MUST own the priorities
- Transparency of tools with criteria
- Regular engagement/rhythm with government, e.g. re-prioritization meetings
- Government discipline to NOT revert to LOE behavior by asking for people or FTE
Agile / DevOps
Lessons Learned & Best Practices
Greater understanding of mission need leads to more impactful solutions

- Focusing on meeting user needs, within the time allotted, reduces “gold plating” and drives the program to focus on value added activities and avoid wasted effort.
- Agile and DevOps enable environments where government end-users can work side by side with developers, allowing solution providers a more intimate understanding of mission needs.

Maximum benefits are realized when Agile & DevOps processes and Customer processes are aligned (e.g. Security, Requirements, Transition)

- This is achieved by integrating Agile & DevOps processes with government standards (eg. NSIS), IT Service Management (ITSM), acquisition practices, contracting procedures and operational processes.
- Agile & DevOps practices can be more effective when blended with best practices of other methodologies like Systems Engineering, Deployment & tailored/adaptive Waterfall approaches, transparent EVMS, and well defined PMP/RACI approaches that clearly define roles, responsibilities, and business processes.

The ability to reprioritize requirements, shift schedules, change design paths, and adapt to evolving mission needs can lead to wasted effort if stakeholders do not understand the value of a traded capability or unsatisfied requirements.

- Successful Agile & DevOps developments focus on development “flow” and sprint “velocity” as measured by speed of new mission capability from dev to production (aka “Value”)
- Programs must groom future requirements ahead of release planning. Agile & DevOps do not absolve government product owner from having a plan and prioritizing requirements in time. Recommended approach is to use epochs in an Agile cadence.
- Training is fundamental and absolutely critical
Agile & DevOps practices include Capability & Architecture Roadmapping... Use them! They provide critical context to Release Planning as projects evolve and teams learn.

- The flexibility inherent in Agile projects may seem to be at odds with long-term architectural objectives, but Agile & DevOps encourages the use of Agile Architectures and capability roadmaps.
- This architecture is emergent, starts with the simplest architecture possible, and uses the concept of an Architectural Runway couple with capabilities that define the needs of the architecture based on the capability roadmaps – on a just-in-time basis.

Agile is perceived in some government circles as lacking rigor when in fact Agile & DevOps, if implemented correctly, are as rigorous, if not more, than legacy processes.

- Agile & DevOps can provide processes that enables the team to be creative, innovative and flexible while, at the same time, ensuring that lean practices, best practices, and systems engineering discipline are applied appropriately thought the program lifecycle.
- Agile & DevOps eliminate “Cowboy Coding”, replacing it with a repeatable, detail oriented process that seeks to deliver working systems on a large scale.

Agile & DevOps provides transparency and new metrics to government, allowing government to have new levels of insight into the status of a program.

- Agile reporting practices allow government to understand status and/or progress of programs on a weekly or even daily basis – including plan vs actuals measurements for technical progress, risks, and velocity.
Agile development is not a silver bullet, and it may not be right for your project
- Depending on program/project goals, not all projects need to use Agile & DevOps. However, multiple studies show Agile development is more effective ASSUMING the project team is properly trained and equipped to implement. The more stable a development team, the more effective the implementation.
- New projects should be considered for Agile or Agile & DevOps implementations.
- Virtually any project type (even non-software development projects) can benefit from an Agile approach with an experienced coach to assist in setting up the project and appropriate training for the personnel participating in the program.

Agile development is simple in theory, but complex in practice. Scrum can be learned in two days, but learning how to setup a successful and high functioning agile program takes years of experience.
- Support team/staff must scale in proportion to the number of scrum teams. PM, Security, and other functions must be appropriate to the number of developers.
- Organization/Management. Successful Agile & DevOps programs require an organizational culture that fosters decentralized decision making, complete transparency, and lean thinking.
- The first step in establishing an Agile/Agile & DevOps program is educating the leaders, executives, and key stakeholders in order to assist them in establishing the required culture.

Agile & DevOps programs must have Agile & DevOps contracts and contract structures.
- The government and its support contractors MUST be trained and certified in Agile & DevOps.
- If contracts are “Agile” in name only and do not allow for continuous changes and evolving requirements the end goal will not be reached.
Agile & DevOps Success Factors

Agile & DevOps contracting with fixed price software development requires numerous factors to be successful:

- A Government team with the overarching vision and engineering/technical ability to decompose that vision into features, and one that is collaborative and includes the development team as part of vision and technical-solution development

- Buy-in and collegial working relationship between the Government program office, SETA support, and the development contractor

- Fully articulated and documented engineering features, ready for development contractor to review, decompose, and commit to during sprint/release planning” without changing meaning

- A Government team ready to protect the development contractor mid-Sprint from unexpected and unplanned activity that would jeopardize the team’s ability to deliver what it committed to during planning

- A contractor that is nimble, flexible, and innovative and able to partner with the Government to achieve program success
Agile & DevOps

Summary, Recommendations, Future Work
Agile & DevOps are vital to realizing the NRO Future Ground Architecture vision
- Uptake across Industry and Government reflects the maturity of Agile DevOps practices
- Next generation of software developers will not know anything else
- Quality + security + speed to delivery requirements cannot be met without it

Both NRO and NGA are demonstrating successful DevOps programs on contract

Must address culture clash between Waterfall and Agile & DevOps processes
- “Agile PLUS Waterfall”, not “Agile OR Waterfall”...recognizes legacy install base
- ATO Automation and Transition to Operations processes are #1 and #2 challenge areas
- Recognize learning component: evolving effectiveness vs designing perfect processes

Software Procurement processes must evolve to support DevOps timelines

Contracting approaches/language must evolve to harness power of Agile & DevOps
- Both CP and FFPLOE approaches can work
- AF and IF components are recommended
- Emphasis on delivery vs FTE, Product Backlog prioritization, and incentivizing speed

Government CO, PM, COTR, and Engineer training & certification is essential
State of DevOps Adoption Action Team: Government & Industry Survey (Proposed)

- Challenge: anecdotal evidence from a few government programs
  - Is industry adopting DevOps on its own or waiting for government direction?
  - Minimal lessons learned and information flow between industry and government.

- IDEA: “crowd source” or “survey” the industry base for...
  - Government programs where DevOps is used
  - Industry partner use (DevOps vs Waterfall) mix.

- Suggestions for information to gather would start with:
  - Name of project
  - Sponsoring organization
  - Who doing the hands on work
  - What capabilities have they demonstrated
  - Planned next steps
  - Eventual goals
What should government specify ref DevOps implementation?

What standard contract language should be implemented?

Sample concepts

- Setting a CAIV/Price for procurement to which bidders would respond with how much of the requirements they can meet for that level
- Procuring “story points” as a measure of development capacity and velocity.
- Tell me what your 1, 3, 5 point stories look like?
- Using scrum team velocity as a tracked performance metric that determines viable speed & capacity to deliver a set amount of functionality.
- Incentivize on “delivered velocity” and accurate estimation of story points in the bid (Agile Dev version of the budgetary “closest to the pin”)
- Government assesses realism based on demonstrated velocity and P/P

Discussion on how these approaches affect program funding profiles

What is the business impact of?

- ATO automation
- Continuous delivery
- Multi-Domain DevOps
Future Topics & Discussion Areas

- Address ATO Automation and Transition to Operations processes
- As DevOps platforms become common, data dependencies must be addressed
  - Mission data essential to create value via applications
  - Data structure dependencies (today the handoffs are overly structured)
  - Tightly coupled Workflow structures are inhibitors
- NRO IAWG Questions
  - What is the business impact of ATO automation and Multi-domain DevOps?
  - What is industry assessment of NRO S2P completeness?
  - How do you measure ROI of DevOps?
  - What should government specify ref DevOps implementation?
For Additional Information: NRO IAWG Contact Info

- **NRO IAWG Chair**
  CAPT Nick Buck, USN (Ret)
  nick@buckgroup.net
  (703) 801-3405

- **USGIF Volunteer Coordinator**
  Justin Franz
  justin.franz@usgif.org
  (571) 392-7205
Where Our National Security Begins...