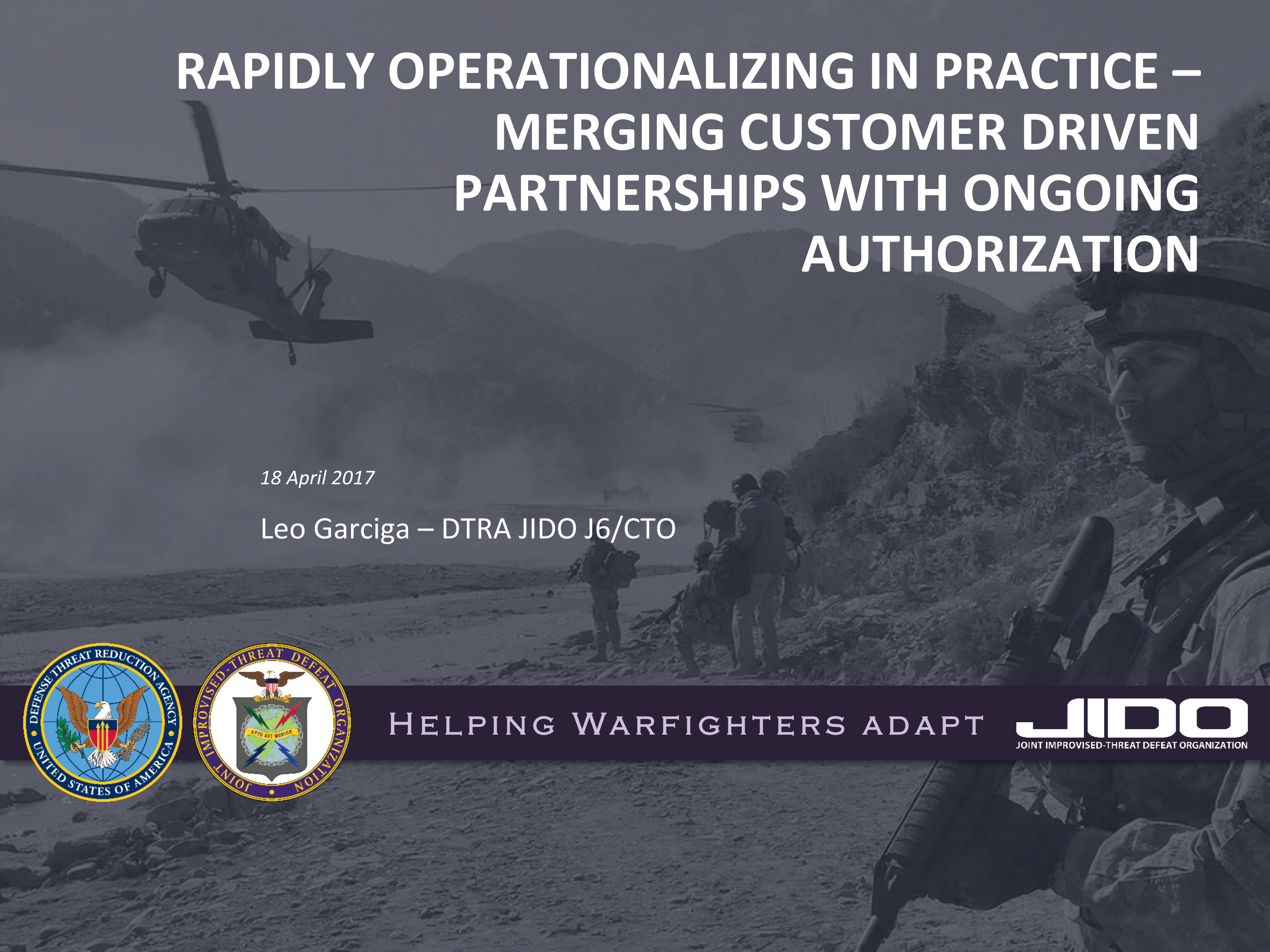


# RAPIDLY OPERATIONALIZING IN PRACTICE – MERGING CUSTOMER DRIVEN PARTNERSHIPS WITH ONGOING AUTHORIZATION

*18 April 2017*

Leo Garciga – DTRA JIDO J6/CTO

HELPING WARFIGHTERS ADAPT

JIDO
JOINT IMPROVISED-THREAT DEFEAT ORGANIZATION

# AGENDA

About Us

Who is JIDO & What is our Mission?

The Foundation

JIDO AGILITY is key

DevOps is Awesome!

Security/ Accreditation

Bringing It All Together

Questions?

# JIDO'S MISSION

## ✓ JIDO is a QRC: Quick Reaction Capability

- Bringing timely solutions to warfighters
- Focused on "what's real" (0-2 years out)
- "Operationalize" new technologies
- Defeat Threat Networks

## ✓ JIDO J6 Mission IT

- Operationalize Big Data analytic platform, "Catapult", and tool suite based on real-time, tactical needs
- Embed with users world-wide to understand data available, analytic methodologies, & capability/data gaps
- Solutions required same day at times

"Adapt or Die"

# JIDO J6: EVOLVING FOUNDATIONS

Rapid capability delivery of new technology is achieved through adaptable contracts**,** rigorous methodology and laser like focus on the user needs:

- **Adopt, Buy, Create**:  Ensures best solutions: GOTS, COTs or develop on site

- **Path to Production**:  Allows rapid development and fielding of secure, compliant and impactful solutions on operational IT networks

- **Operational Experimentation**: Promotes discovery of new technologies and allows the operator and analyst to test, evaluate and provide input into the development of those technologies

JIDO J6 works directly with the Warfighter and technology partners at the tactical, operational, and strategic levels to improve outcomes

# SECURE AGILE AT JIDO – THE FOUNDATION

- ✓ **Running true Agile SDLC for 5 years**
- ✓ **Sprints: 3 weeks, release to production (almost) every sprint**
- ✓ **Approved the “Catapult” Big Data Platform to support rapid delivery**
  - Design ATO concurrently with technical design
  - Accredited/Approved as a “system of systems” with a “type authorization”
  - Secure “docking port” for new tools
  - “Widgets” or webapps get an IT Product assessment only
- ✓ **Continuous Integration already implemented with nightly security scans**
- ✓ **Release management with traditional CM/CCB still hard**
  - Takes more time to review and approve than to develop systems/software
  - JIDO - 50+ releases per year (historically), but not nearly enough
  - JIDO able to move quickly due to IA/Ops/Dev all report to one Gov lead – not the case in most organizations/agencies

> “Agile alone is not enough!”

# JIDO TENETS OF SECURE AGILE DELIVERY

- ✓ **Everything Revolves around Mission: This is cultural – the end user is the ultimate authority. EVERYONE's job is to make the end user successful.**
  - ▪ Dev team builds solutions that ACTUALLY MEET end user end state (not just needs)
  - ▪ The dev team IS THE END USER for IA and OPS.  IA and OPS creatively solve problems to make the Dev team happy.
  - ▪ This is how all teams support the mission, nothing else is acceptable.
- ✓ **Invest in software delivery:** Build secure environments focused on DELIVERY of software, not capabilities themselves.  Automate everything.
- ✓ **Everyone uses the same tools:** All stakeholders (IA, Dev, Ops, & Users) have access to delivery tools to monitor status of software delivery.  Full disclosure and measurement of bottlenecks in order to optimize the process.
- ✓ **Adaptability is greater than correctness:** For every design decision focus on the ability to change later as needed over being right the first time.  Perfect design and perfect security is impossible, so minimize the cost to change when needed.

# DEVOPS IS AWESOME

✓ **DevOps:** Fully automated deployments from dev TO PRODUCTION without human review. Dev or lab-only automation is just CI, not CI/CD and DevOps!

✓ **Proven Benefits:** “High-performing companies are getting better at what they are doing because they are doing it more often.” **(2015 State of DevOps)**

- Change Frequency– 30x more frequent changes
- Meet mission faster – 200x faster from code commit to production (Lead Time)
- More reliable - 60x fewer failures, 168x faster recovery time (MTTR)
- Better Software – 20x better change success, 4x faster to fix bugs

✓ **Adopted tooling to support a DevOps pipeline:**

- Linux Containers (Docker, Trusted Registry, etc.)
- Secure dependency management (SonaType)
- Real-time container security (Twistlock – awesome)

✓ **Security & compliance built in up-front**

✓ **Goal – Full automation from dev to production**

- **SMALL** changes – like every commit
- **No** manual/human review gates
- Adoptable by other agencies

> “Secure DevOps, anyone??”

# WHAT ABOUT SECURITY/ACCREDITATION?

- ✓ **Doesn't DevOps bypass many of the security checks?**
- ✓ **We believe the vision and intent of policy supports an automated delivery pipeline (NIST SP 800-37):**
  - **Appendix F.4 - Ongoing Authorization:** *"The ultimate objective is to achieve a state of* ***ongoing authorization*** *where the authorizing official maintains sufficient knowledge of the current security state of the information system … to determine whether continued operation is acceptable based on ongoing risk determinations… Formal reauthorization actions are avoided in situations where the continuous monitoring process provides authorizing officials the necessary information to manage the potential risk…"*
  - **Appendix G - Continuous Monitoring:** *"… The monitoring program is integrated into the organization's system development life cycle processes… Near real-time risk management of information systems can be facilitated by employing automated support tools to execute various steps in the RMF* ***including authorization-related activities."***

Secure Agile

+

DevOps

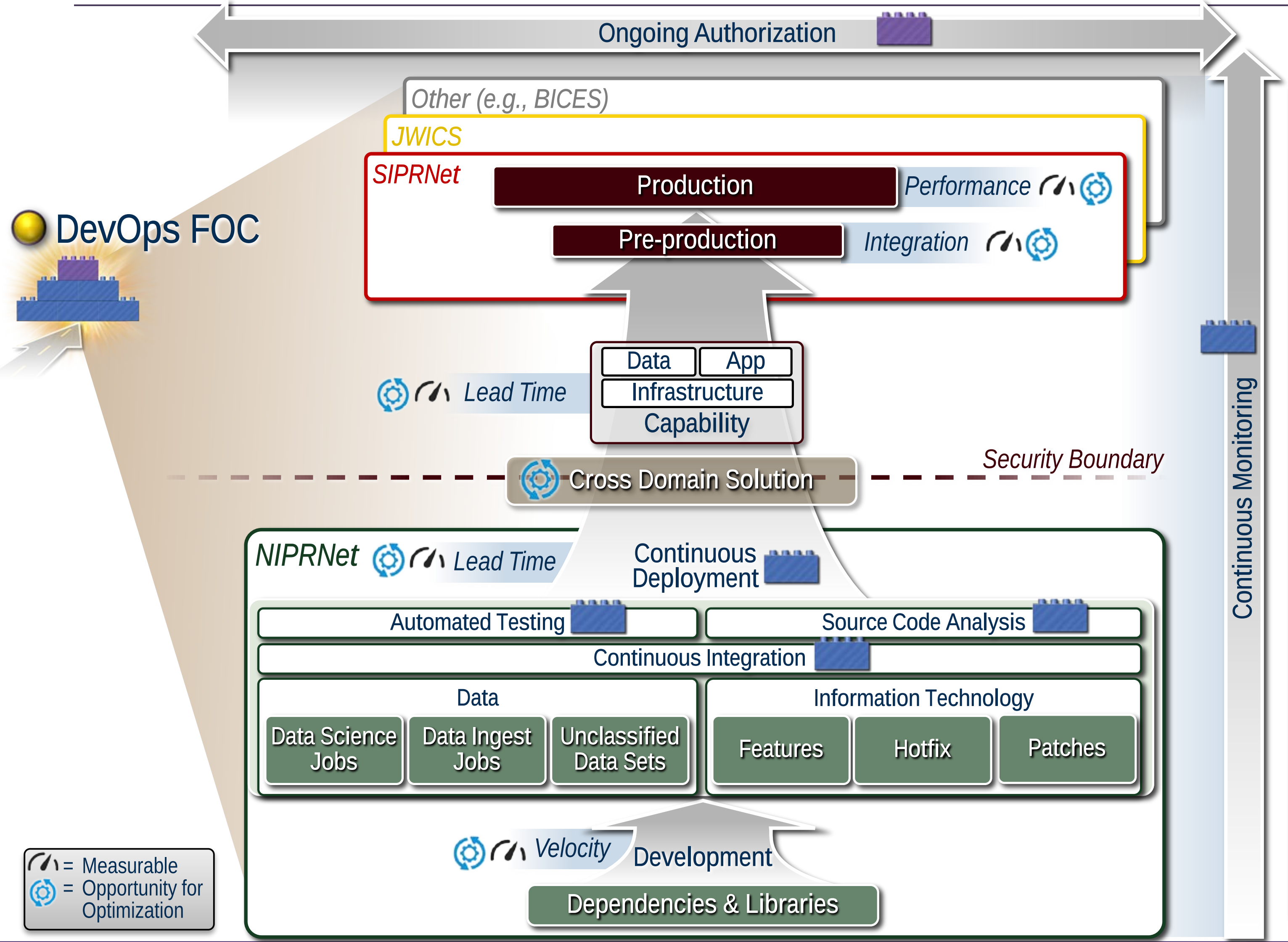
+

Continuous Monitoring

=

**Ongoing Authorization**

# JIDO'S INTEGRATED DEVOPS LIFECYCLE



Ongoing Authorization

Other (e.g., BICES)

JWICS

SIPRNet

Production

Performance

Pre-production

Integration

DevOps FOC

Data | App

Infrastructure

Capability

Lead Time

Security Boundary

Cross Domain Solution

NIPRNet

Lead Time

Continuous Deployment

Automated Testing

Source Code Analysis

Continuous Integration

Data

Data Science Jobs

Data Ingest Jobs

Unclassified Data Sets

Information Technology

Features

Hotfix

Patches

Velocity

Development

Dependencies & Libraries

Continuous Monitoring

◔ = Measurable

⟳ = Opportunity for Optimization

## Key Features

### Streamlined

- Dev through Production in seconds
- Secure-container transfer between networks
- Supports pre-prod and direct to prod paths

### Automated

- Code test, analysis, and build are orchestrated on commit
- Data, app, and infrastructure bundled for portability
- Capability bundles transported, verified, and deployed without manual effort

### Measurement/ Optimization

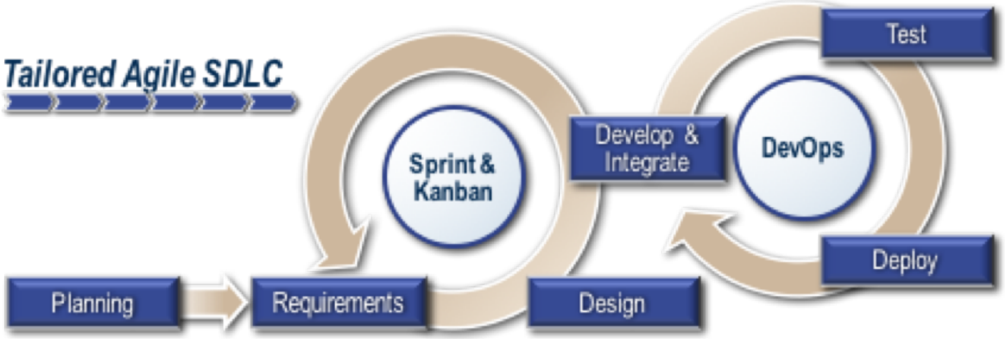
- Development velocity tracked and monitored
- Lead time, build and deployment failures tracked and alerted
- Capability performance monitored and optimized at scale

### Reporting

- Dashboards within DevOps tools give real time insight

Our streamlined approach achieves Ongoing Authorization

# SHIFT IN AGILE APPROACH FOR DEVOPS



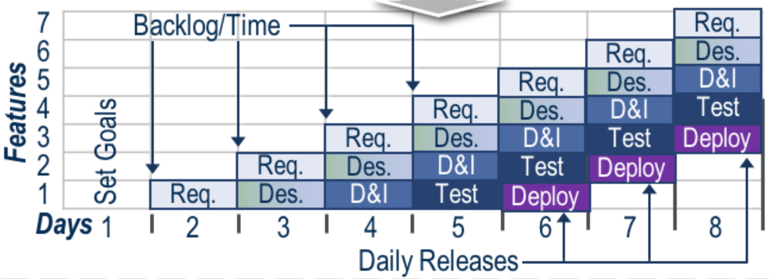
Mission Agility

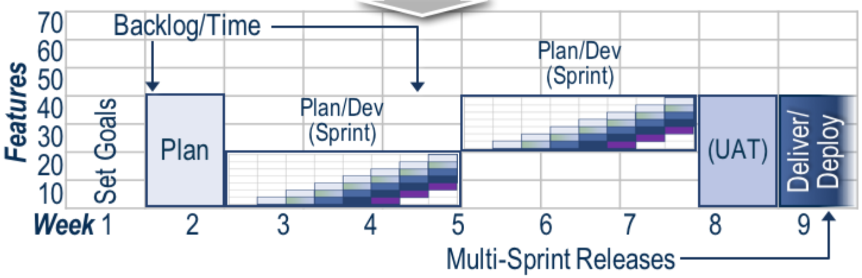
***Kanban-based Delivery (Quick-turn)***

- Single feature tool updates, integrations, and hot fixes
- Frequent feature release using DevOps continuous delivery
- Oversight and control via JIRA virtual Kanban board

***Scrumban-based Delivery (Large Scope)***

- Projects of large scope, many features, or prototypes with unknowns
- Frequent integration test releases to pre-production using DevOps
- Oversight and control via SDPP, kickoff, sprint planning, and review





**Procedures**

- **DevOps Scrum** evaluate features for security, operations concerns
- **Team Scrum** with developers to discuss daily tasking
  - Testing, Requirements, Design, Risks

***Daily***

- **All Daily Plus:**
  - **Constant Product Owner Engagement** (Backlog grooming)
  - **Technical Review Board** to validate technical Implementation
  - **Design Review Board** to guide UI/UX implementation

***Weekly***

- **All Weekly Plus:**
  - **Sprint Reviews** with PO and Stakeholders to review work done
  - **Sprint Retrospective** with team **Sprint Planning** scopes next sprint's requirements

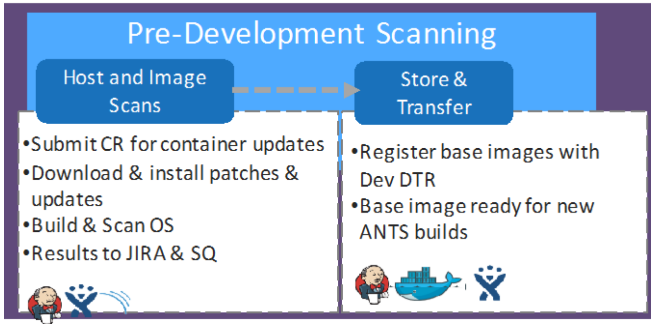
***3 Weeks (Sprint)***

- **All Sprint Plus:**
  - **Project/Release Planning**
  - **Release Design**
  - **SDPP**
  - Kickoff, Release Review, Release Retrospective

***Over 1 Month (Projects)***

Software Development Lifecycle Optimized for DevOps

# JIDO ONGOING AUTHORIZATION PIPELINE



Pre-Development Scanning

Host and Image Scans

- Submit CR for container updates
- Download & install patches & updates
- Build & Scan OS
- Results to JIRA & SQ

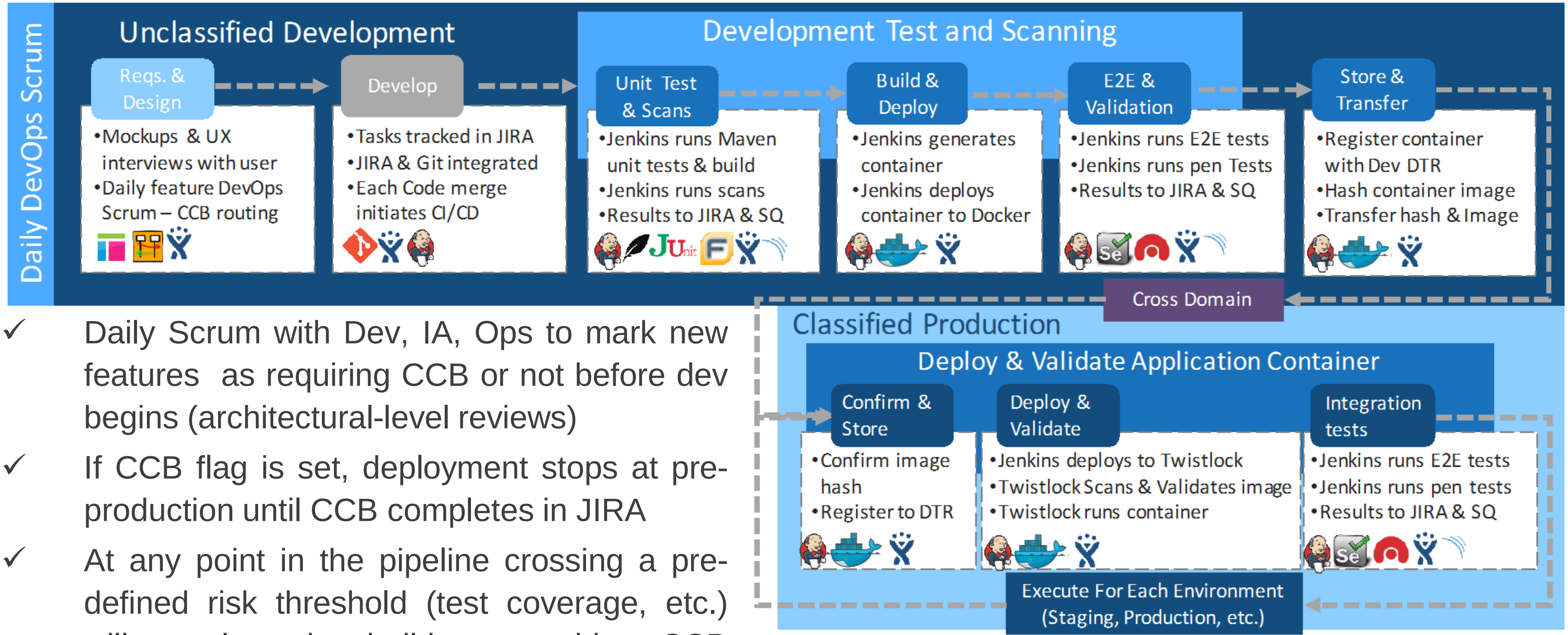
Store & Transfer

- Register base images with Dev DTR
- Base image ready for new ANTS builds

✓ Container-compatible platform certified & approved up front

✓ OS image and common containers (e.g. Java) are hardened & approved up front, registered in Docker Trusted Registry

✓ Workflow is integrated with Jenkins, Git, SonarQube, etc. for a complete picture



Daily DevOps Scrum

Unclassified Development

Reqs. & Design

- Mockups & UX interviews with user
- Daily feature DevOps Scrum – CCB routing

Develop

- Tasks tracked in JIRA
- JIRA & Git integrated
- Each Code merge initiates CI/CD

Development Test and Scanning

Unit Test & Scans

- Jenkins runs Maven unit tests & build
- Jenkins runs scans
- Results to JIRA & SQ

Build & Deploy

- Jenkins generates container
- Jenkins deploys container to Docker

E2E & Validation

- Jenkins runs E2E tests
- Jenkins runs pen Tests
- Results to JIRA & SQ

Store & Transfer

- Register container with Dev DTR
- Hash container image
- Transfer hash & Image

Cross Domain

Classified Production

Deploy & Validate Application Container

Confirm & Store

- Confirm image hash
- Register to DTR

Deploy & Validate

- Jenkins deploys to Twistlock
- Twistlock Scans & Validates image
- Twistlock runs container

Integration tests

- Jenkins runs E2E tests
- Jenkins runs pen tests
- Results to JIRA & SQ

Execute For Each Environment (Staging, Production, etc.)

✓ Daily Scrum with Dev, IA, Ops to mark new features as requiring CCB or not before dev begins (architectural-level reviews)

✓ If CCB flag is set, deployment stops at pre-production until CCB completes in JIRA

✓ At any point in the pipeline crossing a pre-defined risk threshold (test coverage, etc.) will escalate the build to requiring CCB review

Automated, Secure Delivery Reduces Risk

# QUESTIONS?

JOINT IMPROVISED-THREAT DEFEAT ORGANIZATION

APTO AUT MORIOR