

Sponsored by:



DevOps for Defense

December 2020

DevOps for System Engineering & Architecture

JD Black

<https://devopsfordefense.org>

<https://www.meetup.com/DevOps-for-Defense/>

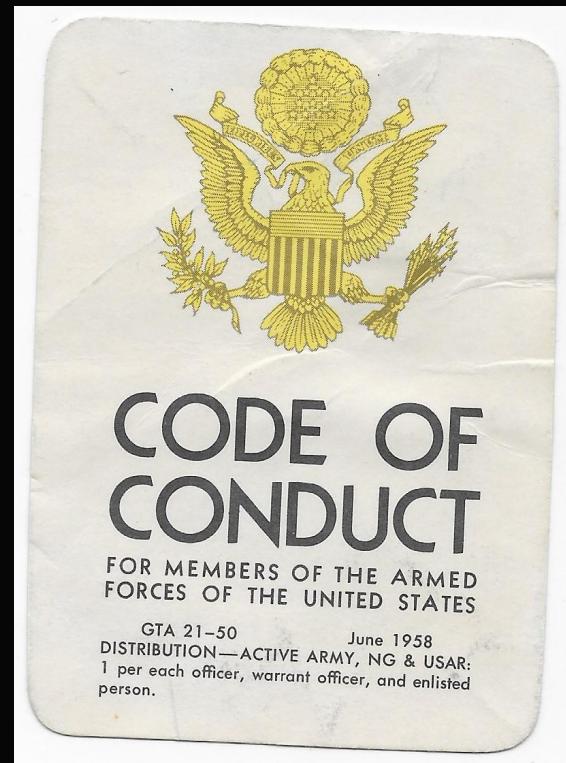
<https://github.com/jondavid-black/DevOpsForDefense>

devopsfordefense@gmail.com

<https://twitter.com/devops4defense>

DevOps for Defense Meetup: Code of Conduct

- UNCLASSIFIED ONLY!!!!
- Treat each other with respect and professionalism.
- Do not talk about private, sensitive, or proprietary work.
- Do talk about your experiences, needs, desires to improve work in our domain.
- Do share your thoughts.
- Do learn from others.
- Do mute yourself while others are speaking!



DevOps for Defense
3 Year
Anniversary

2020 DevOps for Defense Highlights

February

Life of a microservice
@Shipt

Narrated by Vadim Uchitel



March

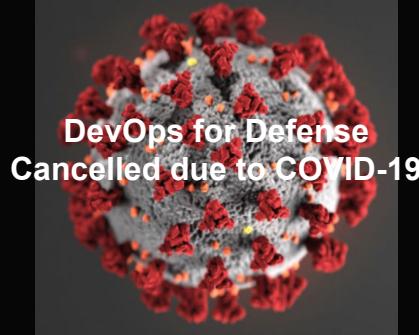
Developer Productivity
Engineering



Hans Dockter
Gradle Founder & CEO



April - June



July



DevOps for Defense
goes Virtual

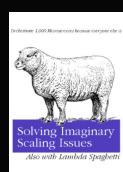
August



September

Scaling
services in the
cloud

by Vadim Uchitel



November

DevOps for
Defense
November 2020

Configuration Management:
#1 Leading Indicator of
High Performing Teams

JD Black

<http://devopsfordefense.com>

First Direct to
YouTube Demo

Everything is on the YouTube Channel if you missed it!

PSA - DevOps Enterprise Summit Now Online



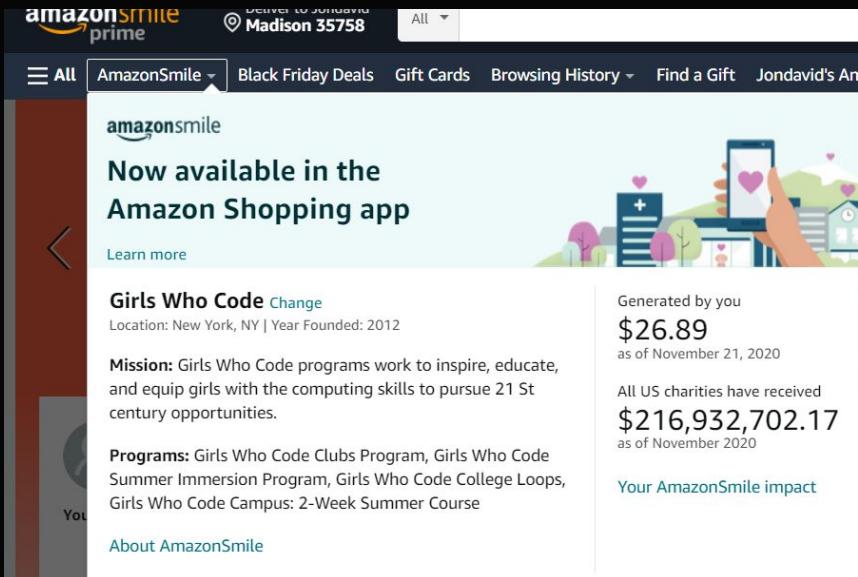
All 100+ videos from the 2020 DevOps Enterprise Summit virtual conferences are available there exclusively. We'll also be uploading videos from our previous conferences, dating back to 2014. Wow!

Want access? You're invited to create a Free Membership account, which allows you to view up to 10 videos every 28 days.

Register - <https://doesvirtual.com/videolibrary/free-membership>

Watch - <http://videolibrary.doesvirtual.com/>

DevOps for Defense Community Challenge



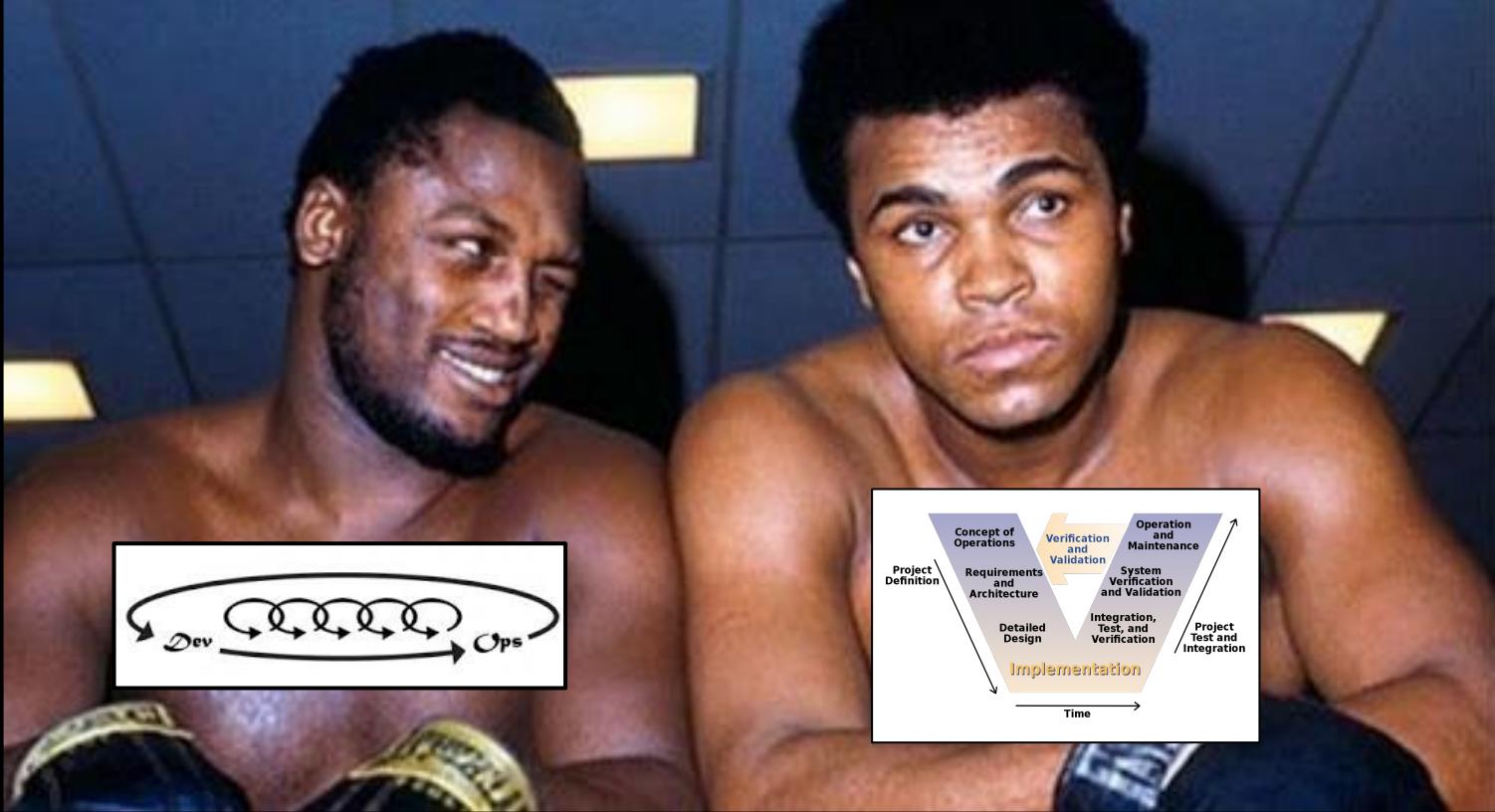
<https://girlswhocode.com/>



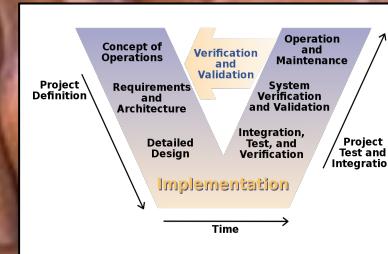
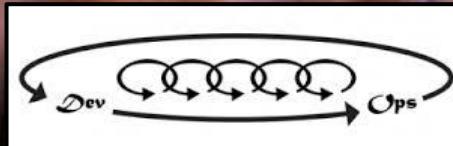
Easy to Support Positive Change

Support Girls Who Code through smile.amazon.com

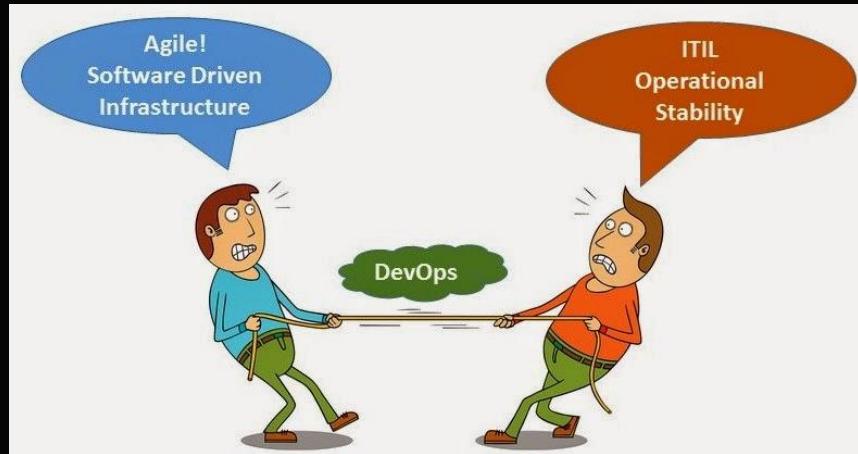
Fight of the Century?



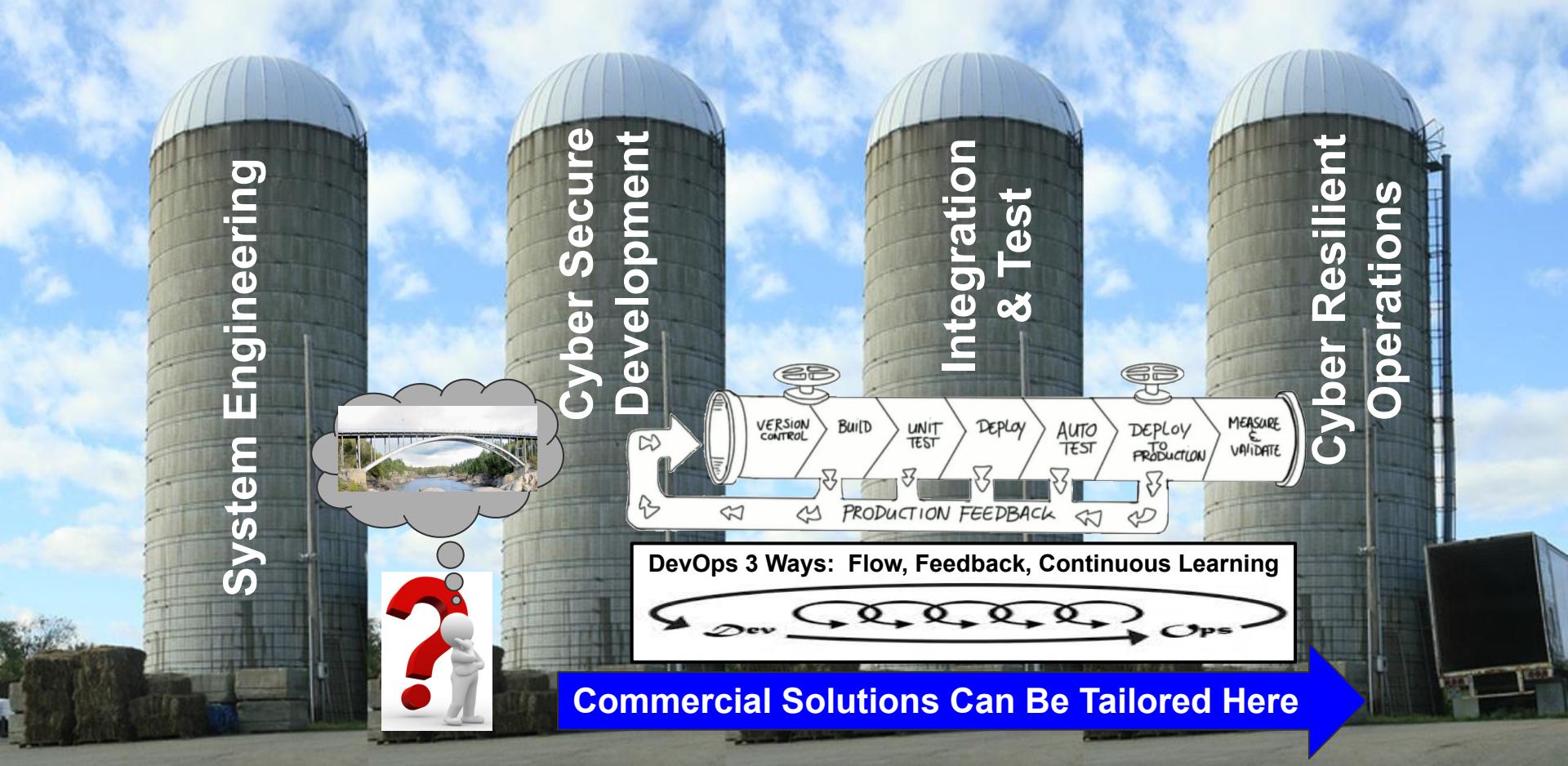
DevOps vs System Engineering



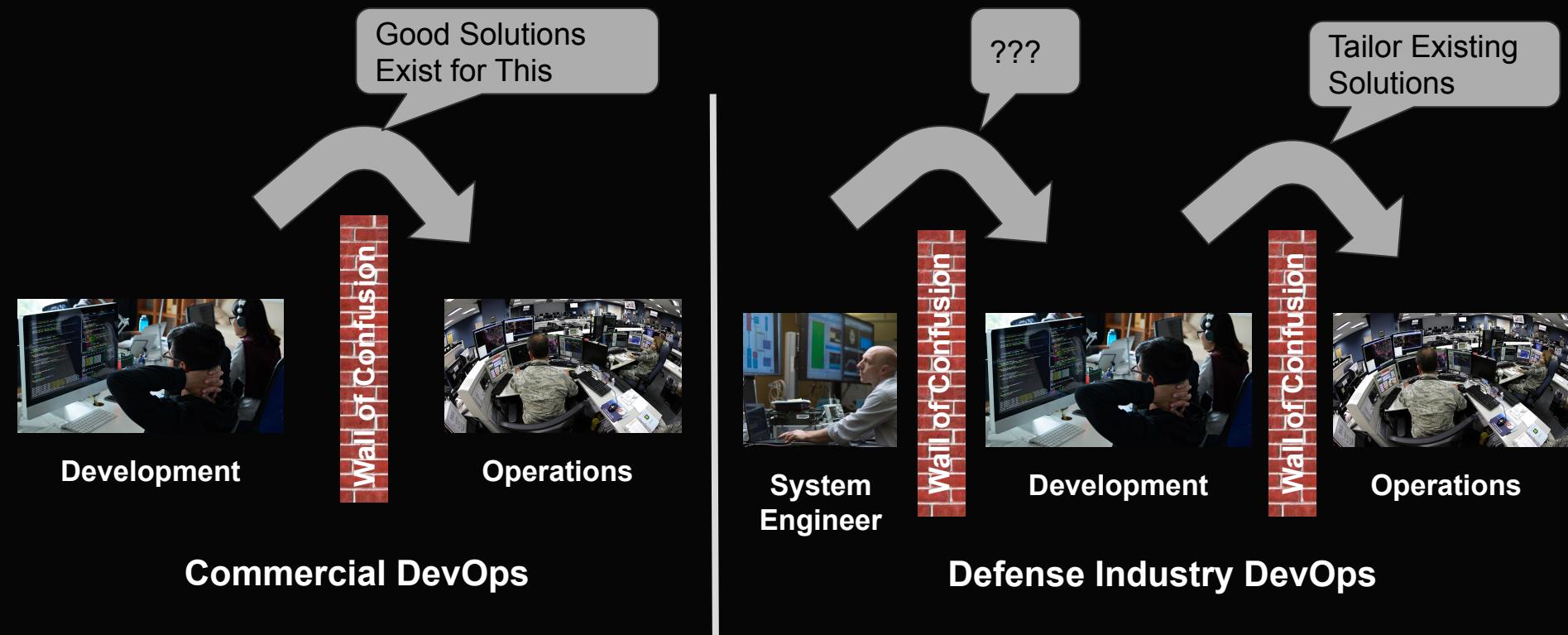
Commercial World - Dev vs Ops



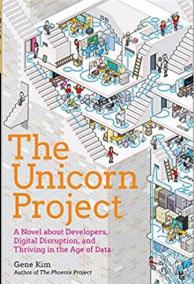
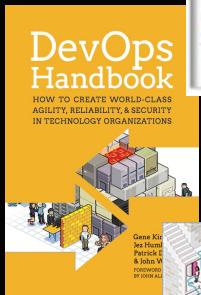
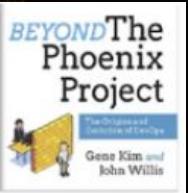
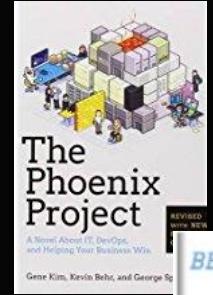
Defense Industry - Even More Handoffs & Challenges



The DevOps Flow: Defense Industry Challenge



Ask the Expert!



Hey Gene, how do you successfully integrate Systems Engineering in a DevOps approach?

Um, I don't know. This doesn't look like something the commercial world deals with in the way the defense industry must.

What helped the commercial world improve? Can we leverage these principles to address our gap?

Technology and automation

- ★ Version control
 - Deployment automation
- ★ Continuous integration
- ★ Trunk-based development
- ★ Test automation
- ★ Test data management
- ★ Shift left on security
 - Continuous delivery
- ★ Loosely-coupled architecture
- ★ Architect for empowered teams

@nicolefv

Process

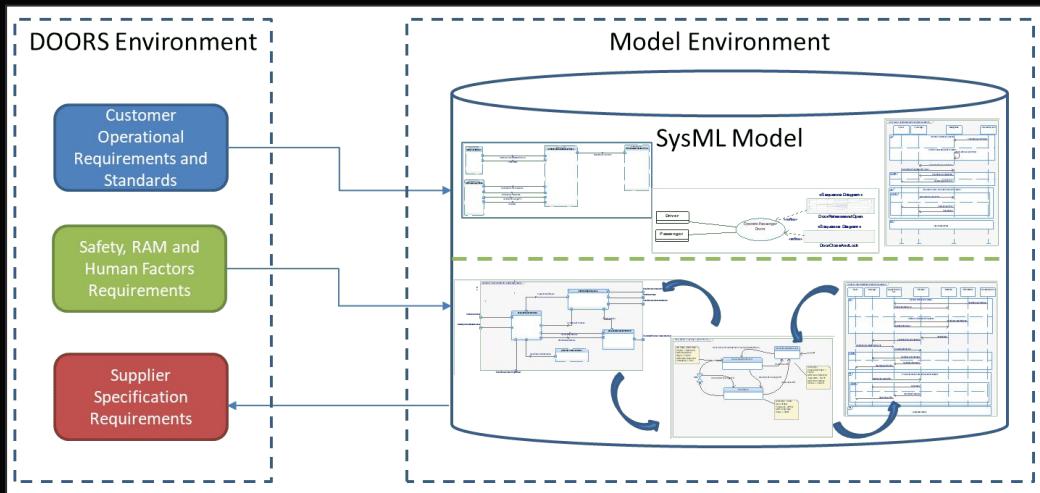
- ★ Gather and implement customer feedback
- ★ Work in small batches
- ★ Lightweight change approval process
 - Team experimentation

@nicolefv

★ = Opportunities to Improve System Engineering with DevOps Practices

Assertion: Infrastructure-as-Code was the Critical Technical Enabler

What Does a System Engineering / Architecture Baseline Look Like to You?



INCOSE defines MBSE as 'the formalised application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases'.

MBSE

Model-Based System Engineering



Volume 1
System Engineering
Ad H. Alves, et al.
Bentley Systems, Inc.
2018
www.bentley.com

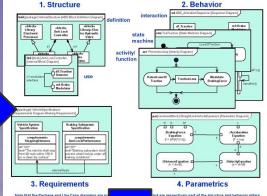


Beware the SE / MBSE Ivory Tower

Architect /
System Engineer



MBSE
System Model



Engineering
Technical Review



Stakeholders

Wall of Confusion

Development



Test



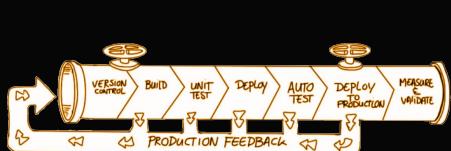
Operations



There's Something Missing...

I want an unambiguous model of my system that...

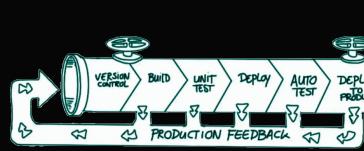
- a. Can be rigorously configuration managed
- b. Can support small batch execution - do just enough to enable dev
- c. Can be automatically checked for quality
- d. Can capture the decomposition of the system
- e. Can capture the interactions within the system
- f. Can allocate behavior to system components using test specifications
- g. Provides direct value to downstream engineering teams**



System
Engineer



Development



Exists



Operations



We believe that

- building Architecture-as-Code, using Infrastructure-as-Code and MBSE principles
- for Architects & System Engineers to Collaborate within cross-functional teams
- will achieve rigorously configuration managed unambiguous engineering artifacts with direct value to downstream engineering teams.

We know we are successful when we see

- Continuously integrated system models, with automated quality assurance frequently delivered to downstream engineering
- with full traceability across requirements, architecture, and test specifications
- that supports faster rigorous end-to-end capability delivery.

Architecture-as-Code

Jellyfish

A structured architectural design language for complex systems.

Captures:

- System Decomposition
- System Interactions
- Behavioral Allocations
- Deployment Specification

Enables:

- Rigorous CM
- Automated QA
- Document Generation
- Code Generation

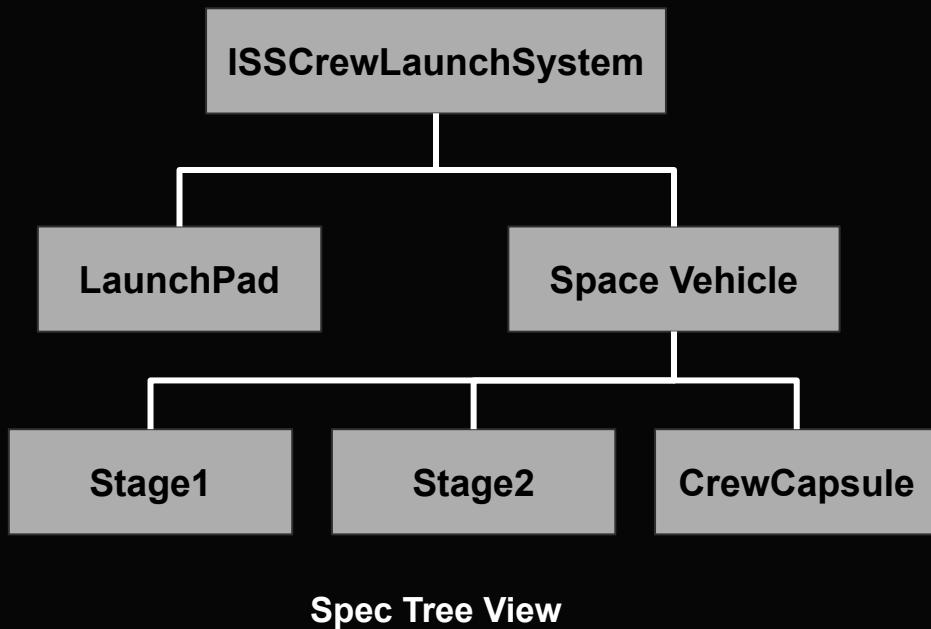


Now available on GitHub under MIT Open Source License.

- <https://github.com/NorthropGrumman/jellyfish>
- <https://github.com/NorthropGrumman/seaside-gradle-plugins>

```
model ThreatEvaluation {  
    metadata {  
        "stereotypes": ["system"]  
    }  
  
    input {  
        SystemTrack systemTrack  
        TrackEngagementStatus trackEngagementStatus  
        ImpactAssessment impactAssessment  
        DroppedSystemTrack droppedSystemTrack  
    }  
  
    output {  
        PrioritizedSystemTrackIdentifiers prioritizedTracks  
    }  
  
    scenario calculateTrackPriority {  
        when receiving systemTrack  
        and receiving impactAssessment  
        and receiving trackEngagementStatus  
        then willPublish prioritizedTracks  
    }  
  
    scenario calculateTrackPriorityWhenTrackDropped {  
        when receiving droppedSystemTracks  
        then willPublish prioritizedSystemTracks  
    }  
}
```

Simple Example: Top Level System Decomposition

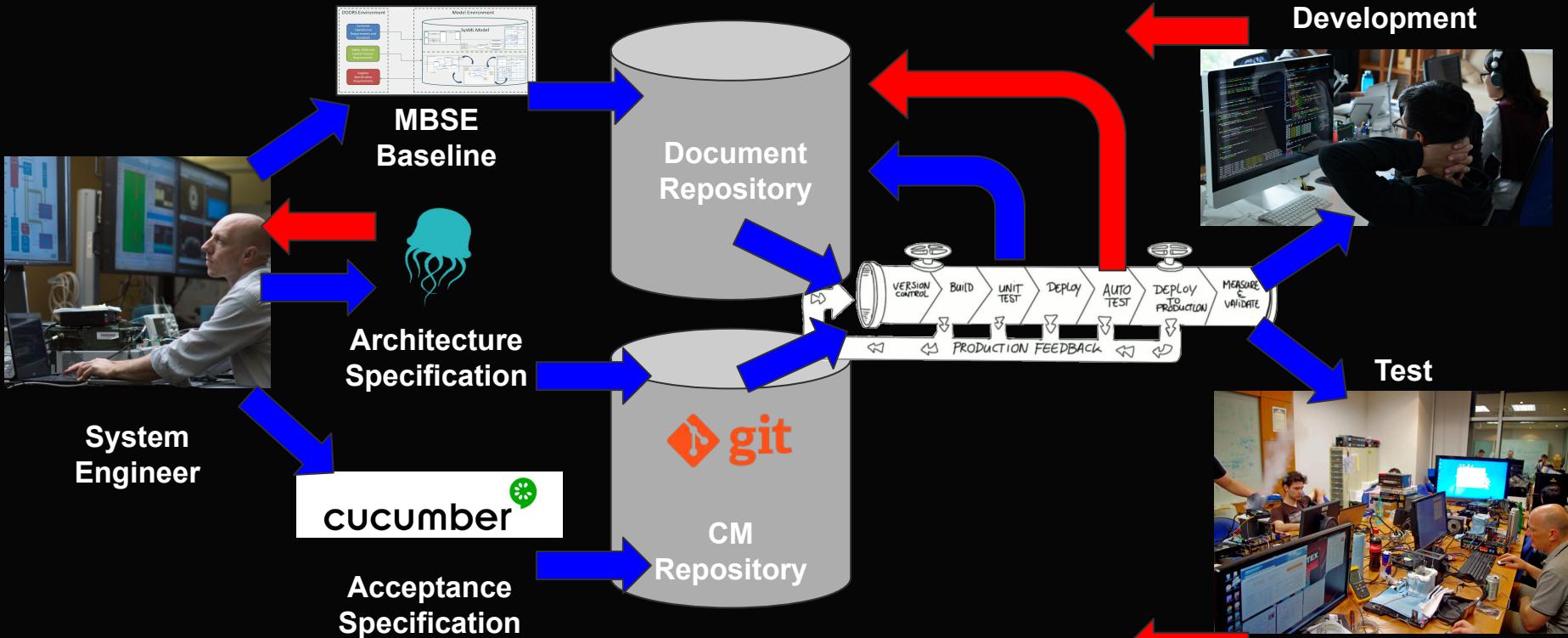


```
model ISSCrewLaunchSystem {  
    metadata {  
        "stereotypes" : ["system", "product"]  
    }  
  
    parts {  
        LaunchPad pad39A  
        SpaceVehicle falcon9  
    }  
}
```

```
model SpaceVehicle {  
    metadata {  
        "stereotypes" : ["system", "product"]  
        "requirements" : ["Falcon9-001", "Falcon9-002", "Falcon9-003"]  
    }  
  
    parts {  
        Stage1 atmosphericBooster  
        Stage2 vacuumBooster  
        CrewCapsule dragonCrew  
    }  
}
```

Architectural Model

How does Architecture-as-Code help?



Establish Fast Flow & Feedback for System Engineering, Frequently Release Artifacts for Dev & Test

That's 2020...Hope 2021 Gets Normal!

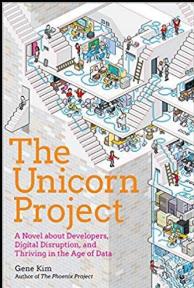
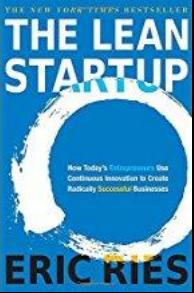
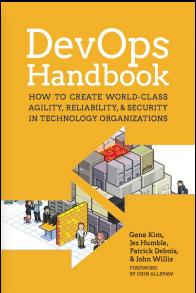
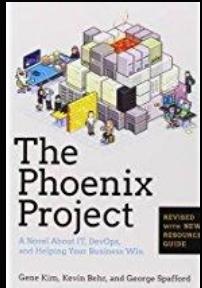


Have a Merry and Safe Christmas Holiday!

DevOps Resources

<https://devopsfordefense.org/resources/>

Books / Publications:



<https://www.meetup.com/DevOps-for-Defense/>
<https://github.com/jondavid-black/DevOpsForDefense>
devopsfordefense@gmail.com

Conference Presentations (YouTube):

- DevOps Enterprise Summit (DOES)
- IT Revolution
- Velocity
- GoTo

