32nd Annual INCOSE international symposium

hybrid event

Detroit, MI, USA
June 25 - 30, 2022

Industrial DevOps for Cyber-Physical Solutions

Robin Yeman
Chief Technical Officer / Advisory Board
Catalyst Campus / ProjectandTeam

Dr. Suzette S. Johnson
NG Fellow, Lean-Agile
Northrop Grumman Corporation

Robin Yeman

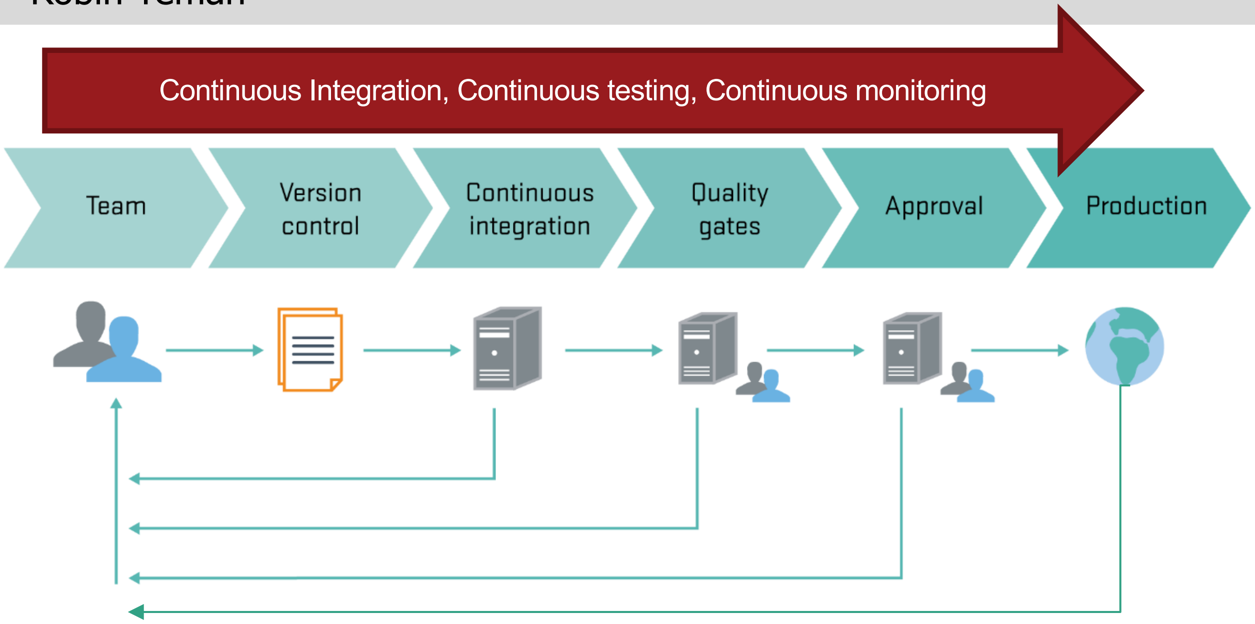
Chief Technical Officer / Advisory Board
Catalyst Campus / Project andTeam
Robin.Yeman@catalystcampus.org

Suzette Johnson

NG Fellow, Lean Agile
Northrop Grumman
Suzette.Johnson@ngc.com

What is DevOps?

"DevOps is a mixture of people, process, and technologies that provides a delivery pipeline enabling organizations to move both responsively and efficiently from concept to business outcome." - Robin Yeman



What is Industrial DevOps?

- The application of continuous delivery and DevOps principles to the *development, manufacturing, deployment, and serviceability of significant cyber-physical systems to enable these programs to be more responsive to changing needs while reducing lead times.*
- Focuses on building a continuous delivery pipeline that provides a multi-domain flow of value to the users and stakeholders of those deployed systems.
- The bodies of knowledge that inform Industrial DevOps principles and practices include DevOps, Lean manufacturing, Lean product development, Lean startup, systems thinking, and scaled Agile development.

Intent of Industrial DevOps

- Scale DevOps principles and practices across large complex systems composed of hardware, firmware, and software to improve flow and delivery of value
- Address the misconception that rapid iteration and flow is only for software development efforts or small applications or systems
- Provide an extended definition and perspective for DevOps
- Provide recommendations

Why?

Industrial DevOps Applied

Companies like Lockheed Martin and Northrop Grumman build cyber-physical solutions with critical human-safety requirements like F-35 and the B-2.

F-35

B-2

Common Problems on large cyber-physical solutions

- Lack of alignment among stakeholders on practices used to engineer, develop, integrate, test, certify
- Lack of alignment among stakeholders on tools used to engineer, develop, integrate test, certify
- Lack of transparency – data, measures, decisions – among stakeholders
- “Nothing is done until everything is done”—large batch processes and mindset
- Delays due to governance and cadence are routine
- Long lead time for hardware procurement

Reference: Carnegie Mellon/SEI, Hasan Yasar

Benefits of Industrial DevOps

- Delivery of value in the shortest, sustainable lead time
- Improved collaboration and knowledge sharing across functional areas
- Build competitive advantage through rapid learning and experiments
- Improved quality
- Improved customer happiness
- Happier, more engaged employees

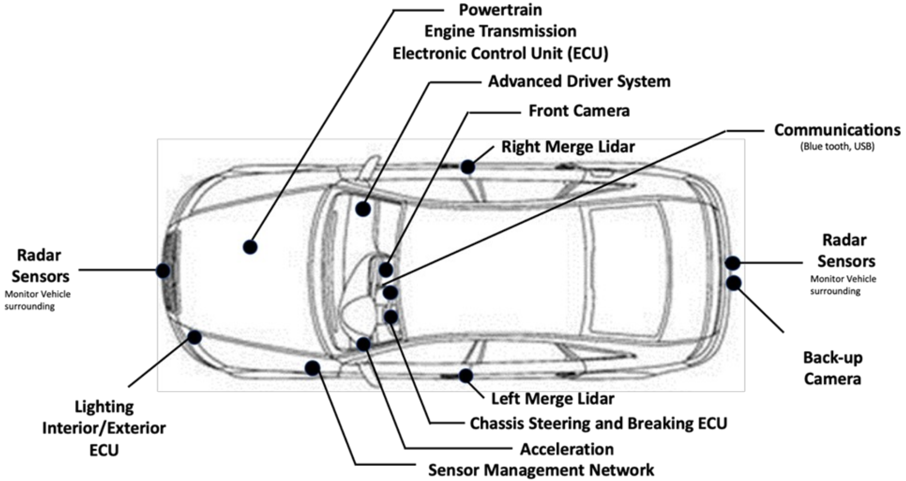
“Sooner, Safer, Happier” ----Jonathan Smart

Industrial DevOps Principles

1. Visualize and organize around the value stream
2. Multiple Horizons of Planning
3. Base decisions on objective evidence of system state and performance
4. Architect for Scale, Modularity, and Serviceability
5. Iterate / Reduce batch size / Get fast feedback
6. Cadence and Synchronization
7. Continuish Integration
8. Test Driven Development

Example: Autonomous Vehicle

Autonomous vehicles are cyber-physical solutions with high complexity, human safety details, certification, and compliance requirements



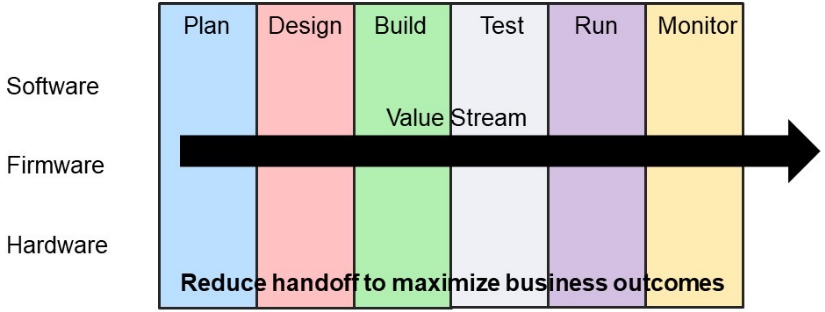
Principle: Organize around value stream

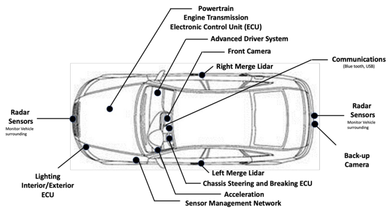
1. Visualize and organize around the value stream
2. Multiple Horizons of Planning
3. Base decisions on objective evidence of system state and performance
4. Architect for Scale, Modularity, and Serviceability
5. Iterate / Reduce batch size / Get fast feedback
6. Cadence and Synchronization
7. Contin-uish Integration
8. Test Driven Development

Vehicle *Value Stream*

→ **Safety and Security** *Value Streamlet*

→ **Collision Avoidance** *Value Streamlet*





Principle: Multiple horizons of planning

1. Visualize and organize around the value stream
2. Multiple Horizons of Planning
3. Base decisions on objective evidence of system state and performance
4. Architect for Scale, Modularity, and Serviceability
5. Iterate / Reduce batch size / Get fast feedback
6. Cadence and Synchronization
7. Contin-uish Integration
8. Test Driven Development

Epic	Q1 Features	Q2 Features	Q3 Features	Q4 Features
Add new camera, associated technology, and hardware updates	Interoperability camera (spike) New sensor Lidar enhancement Recording playback	Procure camera's ongoing traffic surveillance Obstacle-detection system enhancement	Enhanced communications with braking-system sensors Seat belt–sensor coordination Camera prototype on vehicle	Camera instantiation Full regulatory compliance

Forecast

Table 3: Epic 2 Example: Decomposed Work across the Quarter

Each quarter is planned in 2-week iterations

Short Term Minimal Viable Product To Long Lead Items

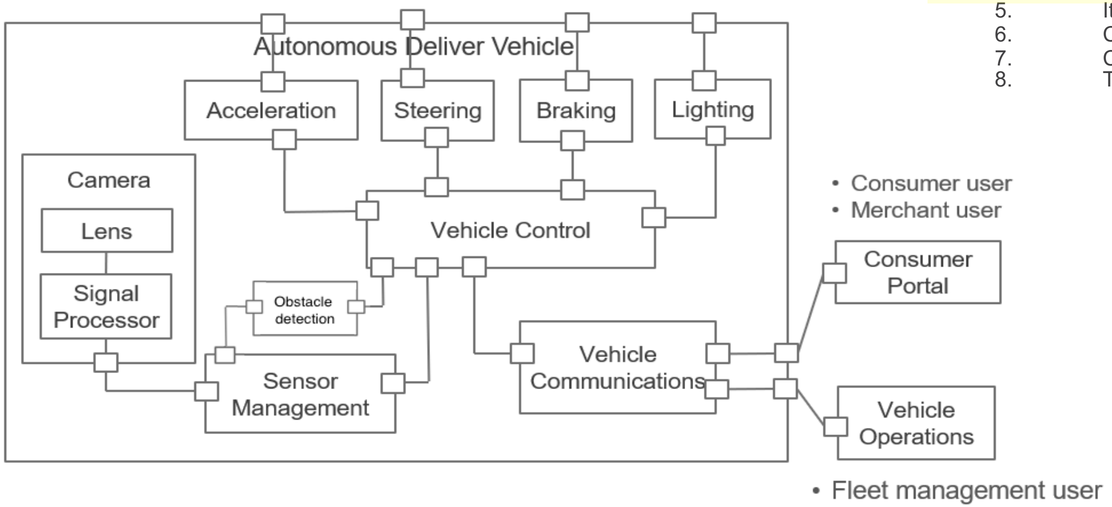
Principle: Base decisions on objective evidence

	Time Horizon	Capability	Evidence
epic	Annual	Enhance obstacle detection through updates to sensor types; refactoring architecture	Drive vehicle through multiple scenarios to validate sensor types; Evaluate deployment rate for new updates
feature	Quarterly	Enhanced Lidar sensor color profile	View colors in simulator to verify improvement
User story	Iteration	Split Lidar by component value	Validate demonstration of Lidar split by through test of
task	Day	Update cloud point extents in ESRI	CI/CD Pipeline has identified no errors with change

1. Visualize and organize around the value stream
2. Multiple Horizons of Planning
3. Base decisions on objective evidence of system state and performance
4. Architect for Scale, Modularity, and Serviceability
5. Iterate / Reduce batch size / Get fast feedback
6. Cadence and Synchronization
7. Contin-uish Integration
8. Test Driven Development

Evidence at each level

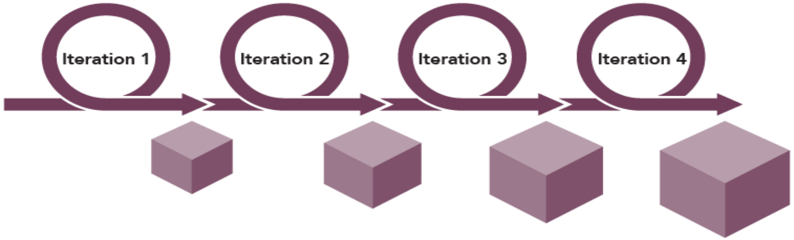
Principle: Applying architecture for modularity



1. Visualize and organize around the value stream
2. Multiple Horizons of Planning
3. Base decisions on objective evidence of system state and performance
4. Architect for Scale, Modularity, and Serviceability
5. Iterate / Reduce batch size / Get fast feedback
6. Cadence and Synchronization
7. Contin-uish Integration
8. Test Driven Development

Modularity enables continuous flow in software and hardware

Principle: Iterate and reduce batch size



1. Visualize and organize around the value stream
2. Multiple Horizons of Planning
3. Base decisions on objective evidence of system state and performance
4. Architect for Scale, Modularity, and Serviceability
5. Iterate / Reduce batch size / Get fast feedback
6. Cadence and Synchronization
7. Contin-uish Integration
8. Test Driven Development

Epic 1: Software-Only Updates to Existing Fleet Vehicles

- Sensor system with an available test environment
- Simulated environment and small, code-based sensors, such as a forward and backup camera.

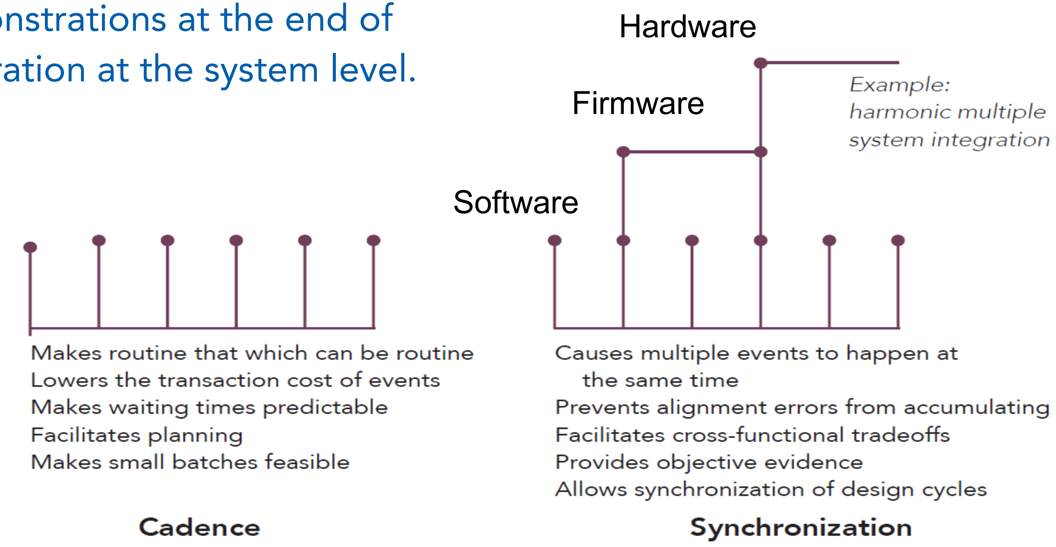
Epic 2: New Camera

- New functionality for radar sensors and using previously generated tests on a radar simulator to check the new code deployment.
- Code integration with the camera and radar system and resolve any integration errors.
- Upon successful demonstrations in the simulated environment, the team tests the firmware and software updates on a couple production cars.

Principle: Apply cadence & synchronization

1. Visualize and organize around the value stream
2. Multiple Horizons of Planning
3. Base decisions on objective evidence of system state and performance
4. Architect for Scale, Modularity, and Serviceability
5. Iterate / Reduce batch size / Get fast feedback
6. Cadence and Synchronization
7. Contin-uish Integration
8. Test Driven Development

Both teams established a cadence of regular quarterly planning and iterations at a two-week period.
Regular synchronization occurs by conducting demonstrations at the end of each two-week iteration at the system level.



Principle: Apply continu-ish integration

Epic 1: Software-Only Updates to Existing Fleet Vehicles

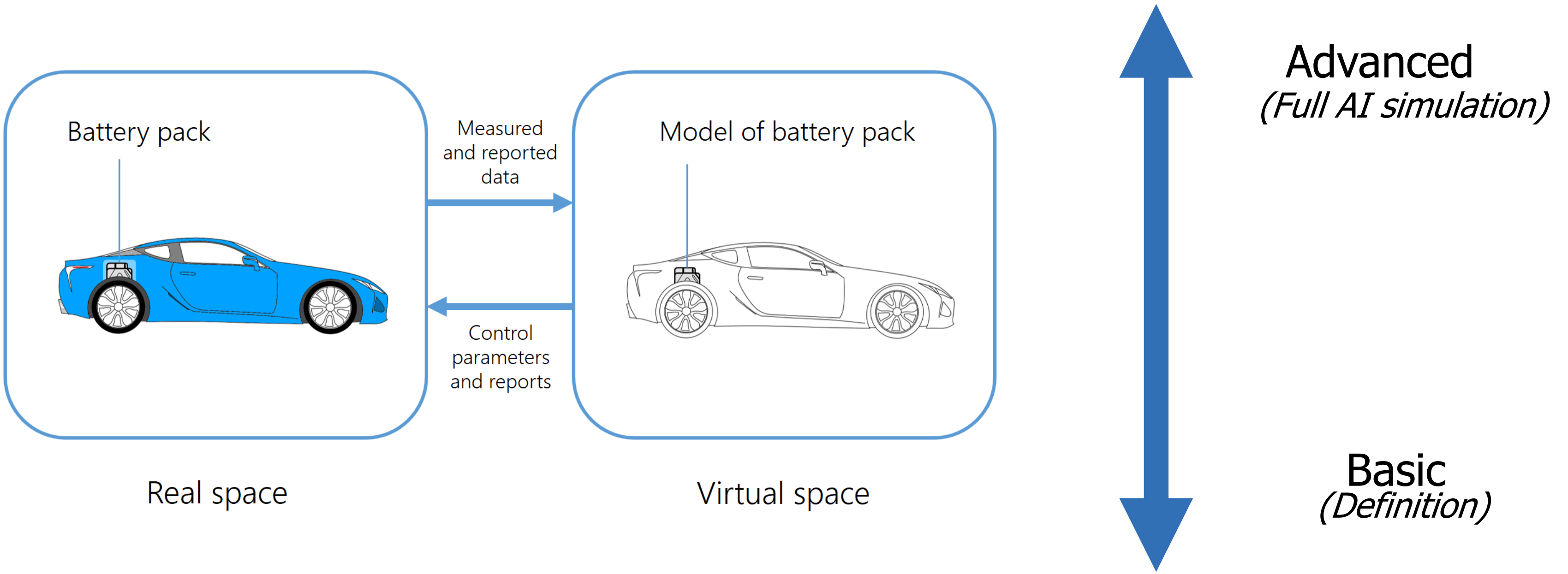
- Digital twin and test environment make developer-level testing cheaper and faster; code-level integrations can happen routinely, daily, or even hourly
- Routine DevOps practices of source code control, automated builds, and automated build verification tests apply well in this case

1. Visualize and organize around the value stream
2. Multiple Horizons of Planning
3. Base decisions on objective evidence of system state and performance
4. Architect for Scale, Modularity, and Serviceability
5. Iterate / Reduce batch size / Get fast feedback
6. Cadence and Synchronization
7. Contin-uish Integration
8. Test Driven Development

Epic 2: New Camera

- Specs provided
- Build a camera-emulator software device which feeds simulated camera data in via the predetermined API and protocols. Deploy the device into the test bed and on a test vehicle to allow new algorithm development for enhanced obstacle detection without the physical device. May also employ field-testing mocks on an actual test vehicle.
- The mechanical design is tested with mechanical mock-ups, which are consistent with the intended physical properties of the camera. The supplier provides pilot-camera hardware in the test environment.

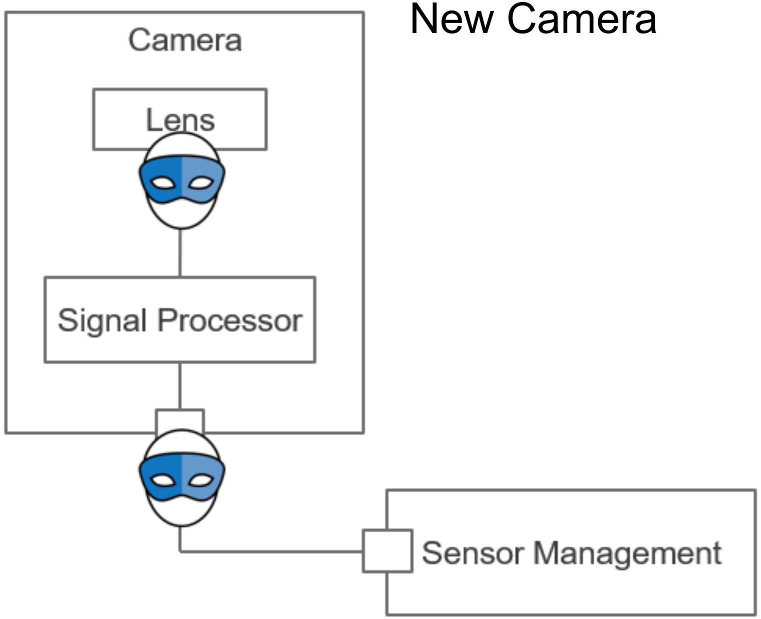
Complete/Partial Digital Thread Enabler



Principle: Apply test driven development

1. Visualize and organize around the value stream
2. Multiple Horizons of Planning
3. Base decisions on objective evidence of system state and performance
4. Architect for Scale, Modularity, and Serviceability
5. Iterate / Reduce batch size / Get fast feedback
6. Cadence and Synchronization
7. Contin-uish Integration
8. Test Driven Development

New Camera

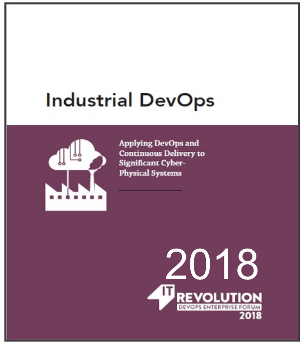


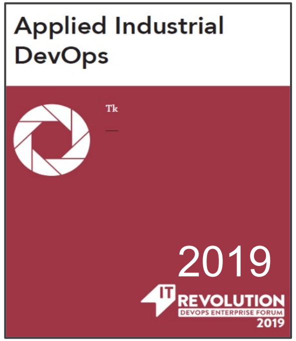
- Model-based systems engineering (MBSE) and computer-aided design (CAD) environments for electrical and mechanical development provide rich support for testing and validating designs.
- Mocks and test doubles isolate hardware changes for early evaluation and testing.

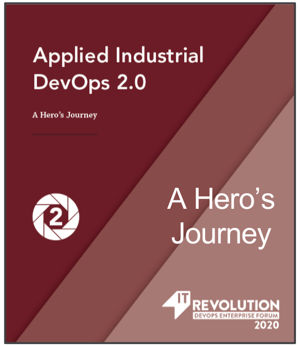
Test-driven development applies to software and hardware development.

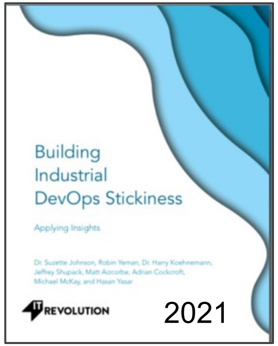
Continuing the Industrial DevOps journey

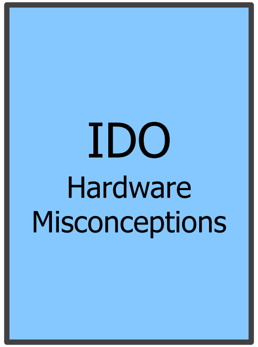
Industrial DevOps expands the definition of DevOps outside of software to enable significant cyber-physical systems development programs to be more responsive to changing needs while reducing lead times.











https://itrevolution.com/book/industrial-devops//
https://itrevolution.com/book/applied-industrial-devops/
Building Industrial DevOps Stickiness (itrevolution.com)

Thank you

32nd Annual **INCOSE**
international symposium

hybrid event

Detroit, MI, USA
June 25 - 30, 2022

www.incose.org/symp2022