Penny @ MathWorks, Rise of DevOps

- Revision control
- Issue tracking
- Project management
- Test harness
- Continuous integration
- Release processes
- ...

MATLAB 5.0
Road to Software Defined Vehicles

Electrification

Automated driving

Connected vehicles

Software Defined Vehicle

Challenges for SW Development
- Adopting Central Computers and Service-oriented Architectures
- Deploying AI and Data-driven features
- Adopting Agile & DevOps methodologies

Challenges for Organizations
- Reshaping of the value chain
- Mindset shift to virtual development
- Workforce mobility and talent acquisition
Road to Software Defined Vehicles

Challenges for SW Development
- Adopting Central Computers and Service-oriented Architectures
- Deploying AI and Data-driven features
- Range Optimization

Challenges for Organizations
- Reshaping of the value chain
- Mindset shift to virtual development
- Workforce mobility and talent acquisition

Adopting Agile & DevOps methodologies
DevOps: a set of practices, tools and philosophy to automate and integrate processes between Development and Operations

**develop**
Create and refine model. Perform unit testing, using CI to automate when possible.

**build**
Model passes unit testing, prepare for system level testing

**test**
System-level test. Qualify release candidate in final form – in workspace with I/O, initiated as it will be initiated in operation. Extensive use of CI system.

**release**
Deliver component to repository for availability to operation. Often uses CI/CD system.

**feedback → plan**
Use performance data and new requirements to plan next iteration, e.g., develop new model.

**deploy**
Put a released and configured component (with other parts of system) in operation

**operate**
Run qualified, released component in production

**monitor**
Collect performance and other operational data: debug, monitor, and trigger actions if warranted
We see customers at different levels of DevOps maturity
An Automotive View of the DevOps Lifecycle

Perpetually Upgradeable Vehicles

- Faster Development and Release Cycles
- Better Leverage Data from System Operation
- Enable New Types of Algorithms and Functionality in the Field
Before Model-Based Design: Highly manual with late integration

Software Development

- Code and Check
- Build and Static Testing
- Integration Testing

Vehicle Fleet Testing
From the 2019 MAC

Model-Based Design
Systematic use of models throughout the development process

- Modeling
  - Simulation
    - Fast repeatable tests

- Automation
  - Coding
  - Verification
    - Fast agile development loops

Andy Grace, VP MathWorks
A look to the future with Model-Based Design
Model-Based Design

**Technical Safety Concept**

- Customer Need
- System Requirements
- System Architecture Models
- MIL Early Validation

**Model Verification**

- Discover design errors at design time
- MIL Unit and Integration Testing
- Review and Static Analysis at the model level

**Code Verification**

- Gain confidence in the generated code
- SIL Unit and Integration Testing
- MISRA Compliance

**Software Level**

- Object Code
- Automatically Generated C/C++ Code
- Implementation Models
- Software Architecture Models
- Software Requirements
- Model Verification

**System Level**

- Find Defects Sooner
The Broader Software Development Landscape

**System Engineering**
- SE tools
- Software Architecture Models
- Requirements and Architecture Design

**Software Development**
- "Code-centric"
- "Model-based"
- Code and Check Virtual Integration
- Build and Static Testing Dynamic Testing
- Integration Testing Virtual Integration Virtual Scenarios, etc.
- Source repo
- Test cases
- CI Pipeline
- Production artifacts
- Test results
- System Models
- Implementation Models
- Scenario Models
- Generated C/C++ Code
- Scenario Models
- System Models
- Object Code
- Scenario Models

**Vehicle Fleet Testing**
- Continuous Deploy, Test and Operate
These changes often involve new teams that need to align
Changes in how feedback is done for test fleets

System Engineering
- "Code-centric"
  - SE tools
  - Software Architecture Models
  - Requirements and Architecture Design

Software Development
- Code and Check Virtual Integration
- Build and Static Testing Dynamic Testing
  - Source repo
  - Test cases
  - CI Pipeline
  - Production artifacts
  - Test results

  - System Models
  - Implementation Models
  - Scenario Models

  - System Models
  - Generated C/C++ Code
  - Scenario Models

  - System Models
  - Object Code
  - Scenario Models

Vehicle Fleet Testing
- Continuous Deploy, Test and Operate
These changes are steps to a broader DevOps view
These changes are steps to a broader DevOps view

A. DevOps for add-on services for systems

B. DevOps to monitor test fleets

C. DevOps for systems in operation

Operational Data via 5G/Cloud

Start of production (SOP)

Requirements and Architecture Design

Software Development

Vehicle Fleet Testing

Continuous Deploy, Test and Operate

Dashboards

Operational Data via 5G/Cloud

Mobility Products and Services

Start of production (SOP)

Requirements and Architecture Design

Software Development

Vehicle Fleet Testing

Continuous Deploy, Test and Operate

Dashboards

These changes are steps to a broader DevOps view
This broader view involves more than embedded code

- Engineer the operational data; look for insights and apply analytics
- Stream operational data to large-scale streaming frameworks
- Train and deploy models in big data systems/in the cloud
- Use models as a digital twin
- Deploy models as microservices, in containers
- CI integration for models and code
Capabilities on the Road to DevOps

Data Analytics for Automotive

Containers, Microservices and more

Continuous Integration
Perform Fleet Analytics at Scale
We can work with you to analyze your fleet data with MATLAB
Develop & deploy analytics for large-scale streaming environments

Streaming Data

Time-coherent data grouped by device ID

Stream Processing Framework

MATLAB Desktop

Apache Kafka

Kafka Connector for MATLAB Production Server
Scale analytics to big data in the cloud with Databricks

- Use existing MATLAB Code
- Run it faster
- Self-service access corporate (big) data
- Share model and algorithm with non-MATLAB users

Learn more: MATLAB with Databricks
Capabilities on the Road to DevOps

Data Analytics for Automotive

Containers, Microservices and more

Continuous Integration
Deploy your simulations

• Enable others to run your simulations

• Leverage digital twins for in-operation assets

• Interchange with other simulation environments via FMU
Deploy your simulations as microservice Docker containers

MATLAB & Simulink

Simulink Compiler

MATLAB Compiler SDK

Integrate into enterprise applications

Speed up simulation

Microservice

HTTP endpoint

kubernetes

docker

docker

docker

docker

docker

docker

R2022a
Capabilities on the Road to DevOps

Data Analytics for Automotive

Containers, Microservices and more

Continuous Integration
Integrate with Continuous Integration platforms

(7) Documented MathWorks CI integrations to reduce risk and simplify use and setup

Learn more: Continuous Integration with MATLAB and Simulink
Leverage CI/CD Automation for MBD workflows

Prebuilt & Tailorable Model-Based Design Pipeline

Build system to generate pipeline and optimally execute

Prequalification with Process Advisor

Examples to run process on common CI Systems

Learn more: Continuous Integration for Model-Based Design
Continuous Integration with Polyspace

Implement Best Practices in your Software Factory to Improve DevOps Metrics

October 20, 2022 | Stuttgart
Tjorben Gross
Skanda Naglapur Ramamurthy

DevOps Result review workflow with Polyspace

- Direct Feedback
- IDE Plugins are used
- Learning
- New issues break the build
- Flow/ Automation
- Critical and high issues resolved
Summary: On the road to software-defined vehicles, DevOps is an important aspect

- We have experience working on DevOps with customers in many other industries

- We want to work with your MBD teams AND your DevOps/Infrastructure teams as you navigate the software trend

- Join us on this DevOps journey!
Enjoy the Conference!