MathWorks
AUTOMOTIVE
CONFERENCE 2024
Europe

How the software-defined vehicle trend is transforming development processes

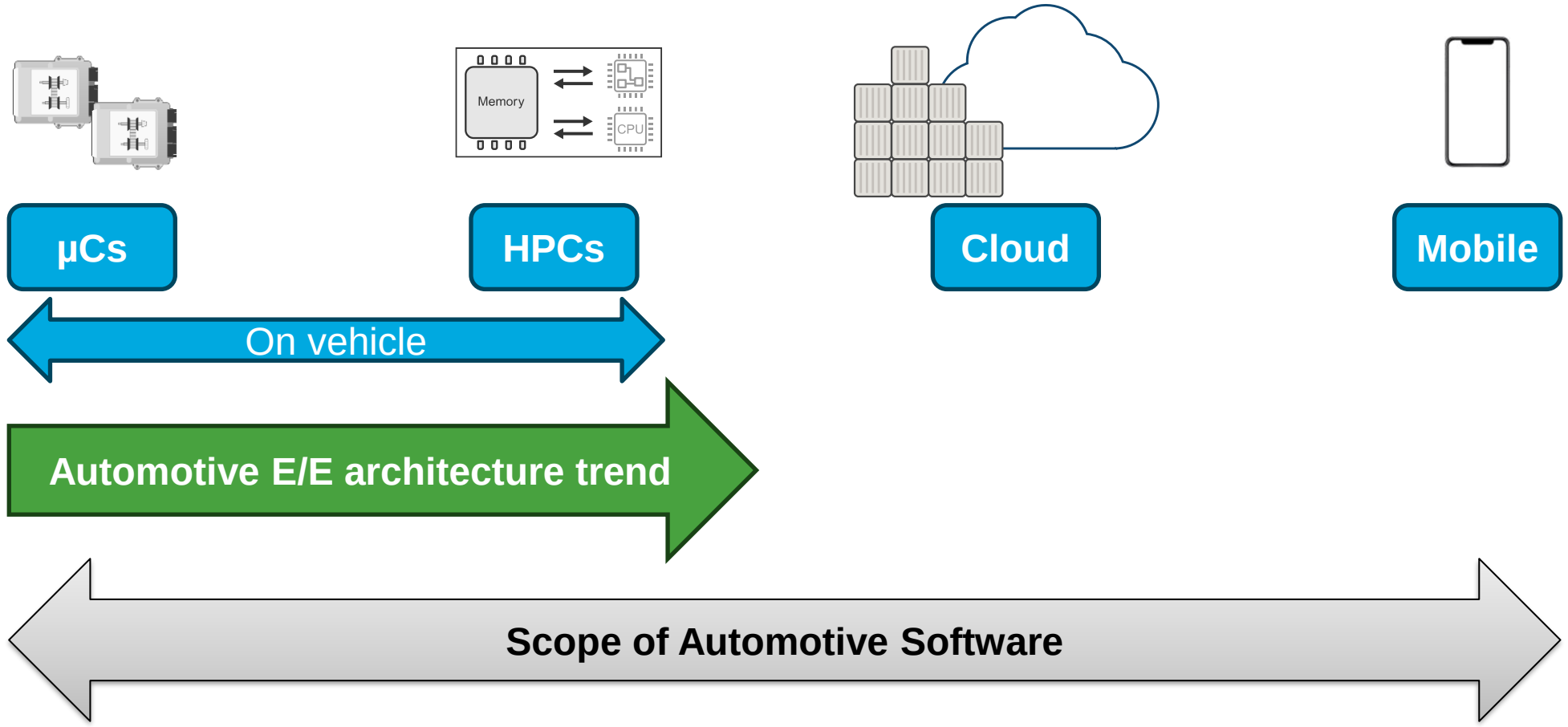
Dr. Hans Martin Ritt, The MathWorks

MathWorks®

Identifying development processes that are transformed by SDV

- Automotive E/E architecture evolution
- Software factory
- Closing the DevOps loop
- …

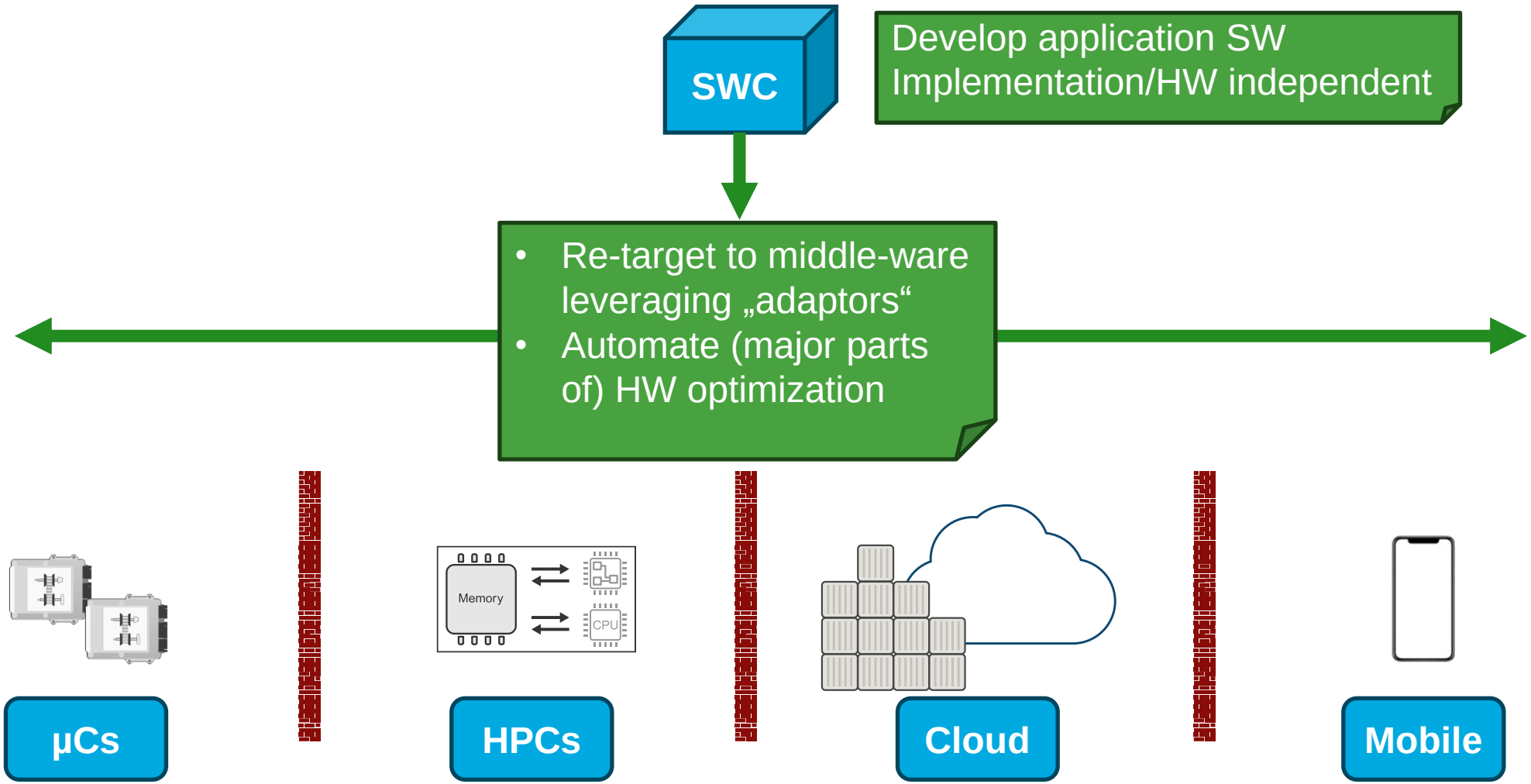
Address the full scope of SDV workflows - together



Demonstrate how collaboration can enhance SDV workflows

- Collaboration with FEV on an SDV demonstrator
- Cover the range of uCs, High performance computers (HPCs), cloud and mobile
- Development environments need to integrate – addressing
  - function development and
  - verification and validation

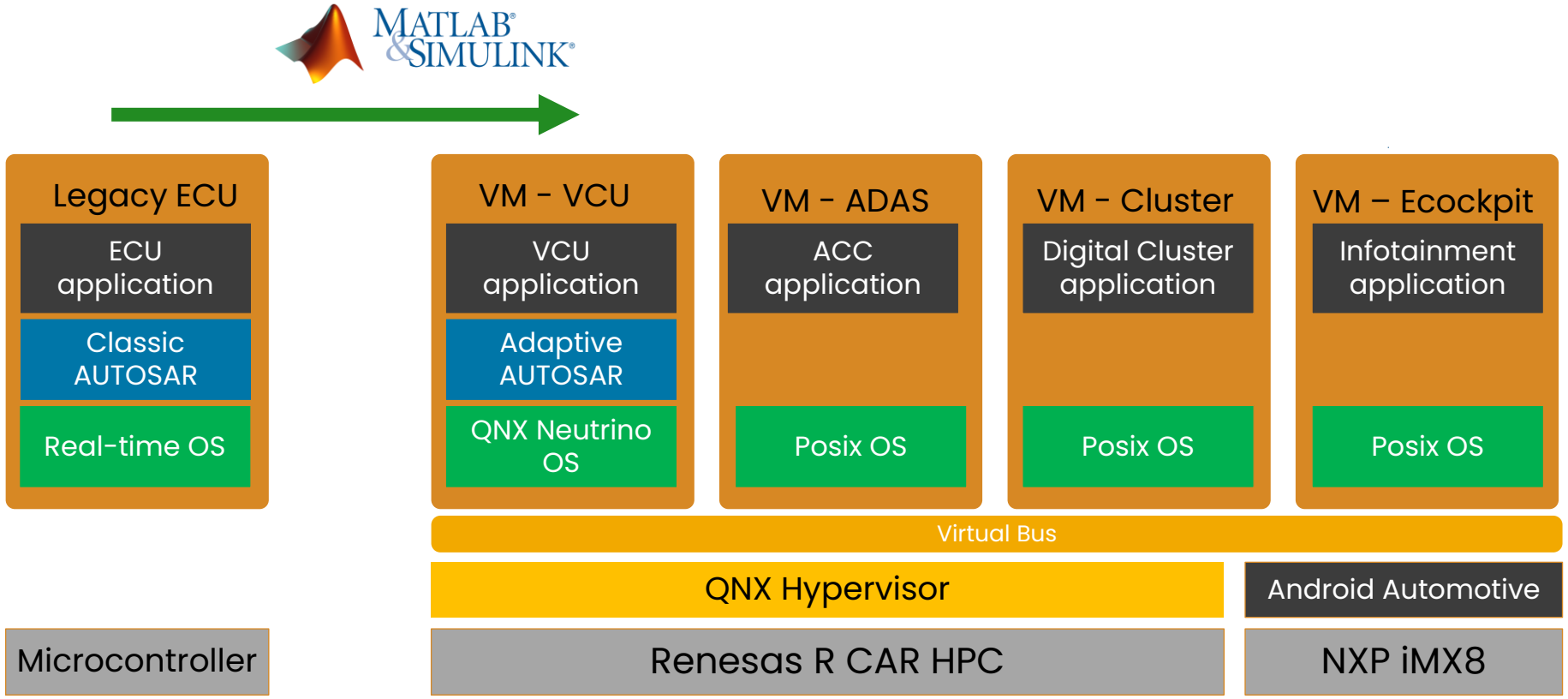
Maximize reuse through modular and adaptable development



SWC = Software Component

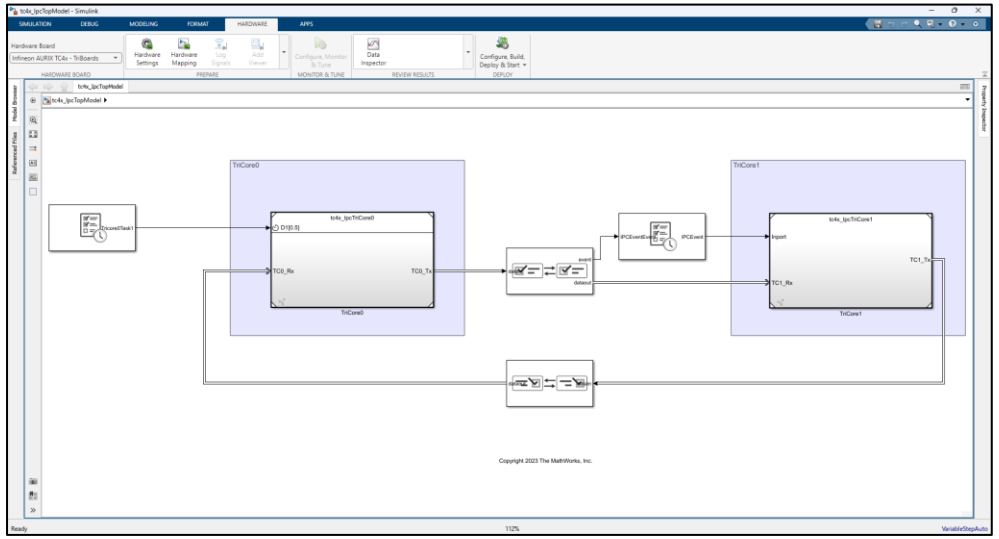
FEV's SDV HPC Software Development demonstrator

- Architecture Overview
- PoC application transition legacy ECU to VCU



Rapidly develop on AURIX without deep device knowledge

- SoC Blockset Support Package for Infineon AURIX™ TC4x Microcontrollers



Multicore model example simulates IPC and then deploys to hardware

- **Partition algorithms** to run on **multiple** TriCore **cores** and on **PPU**
- **Analyze** algorithm **resource usage and task execution**
- **Generate optimized code** for SIMD PPU Core and multiple TriCore cores
- **Performing Processor-In-Loop tests** on PPU and TriCores
- **Supports Synopsys Metaware**

Shift left to increase update frequency for high integrity software

Traditional automotive development

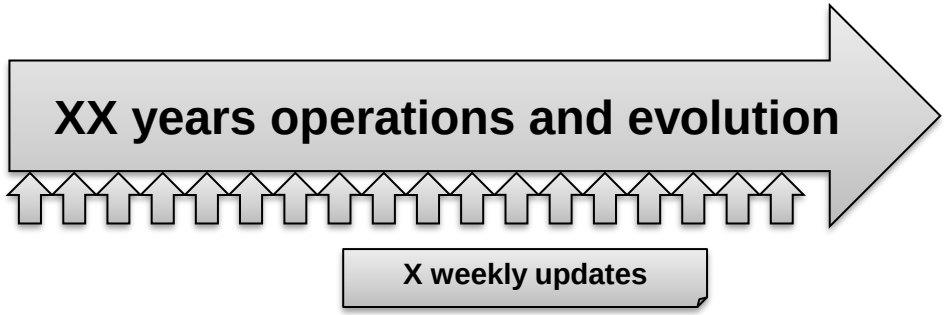
Conflict:
- Months long verification and validation process
- Significant level of manual work

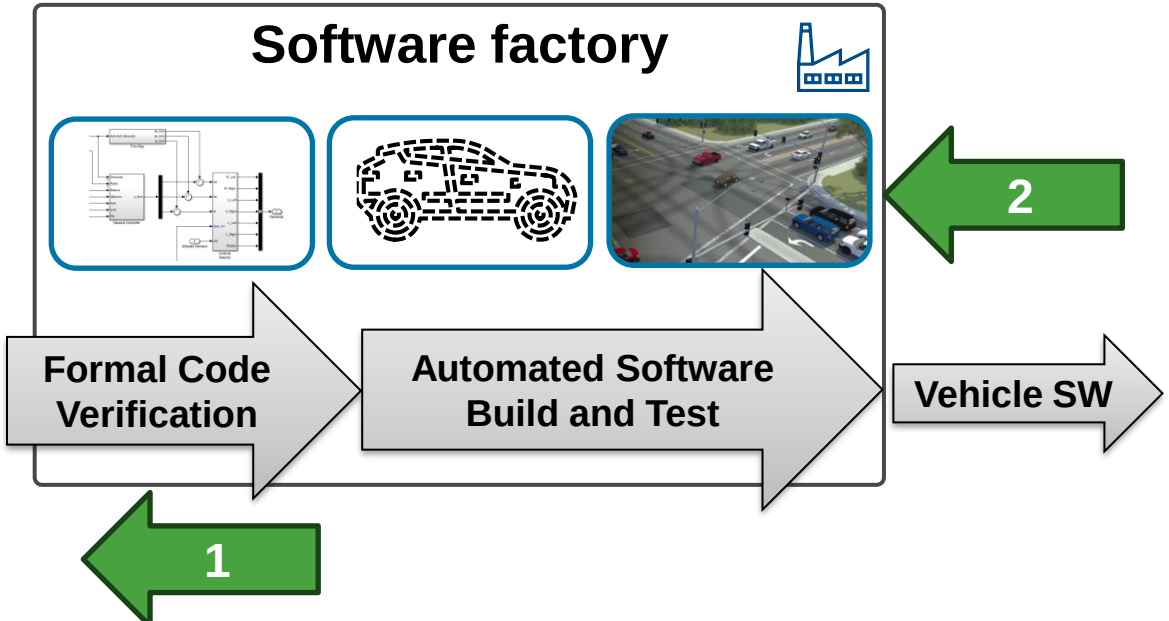
➲**Reduce and limit update frequency**

Solution: Shift left
1. Formalize code verification
2. Virtualization and automation of physical/manual work

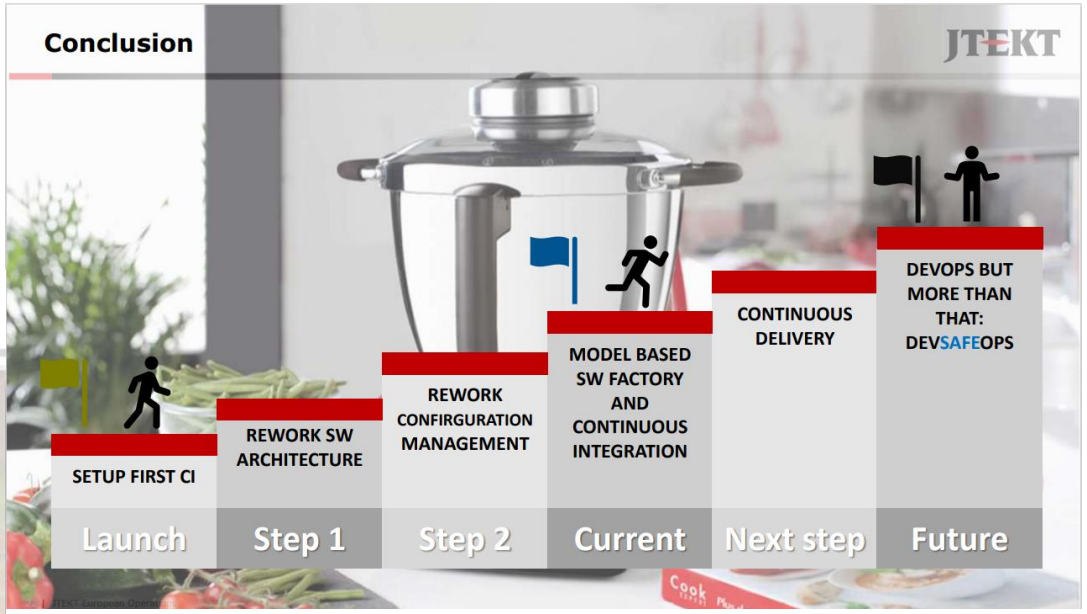
- Infrastructure as code

Frequently update vehicles in operation

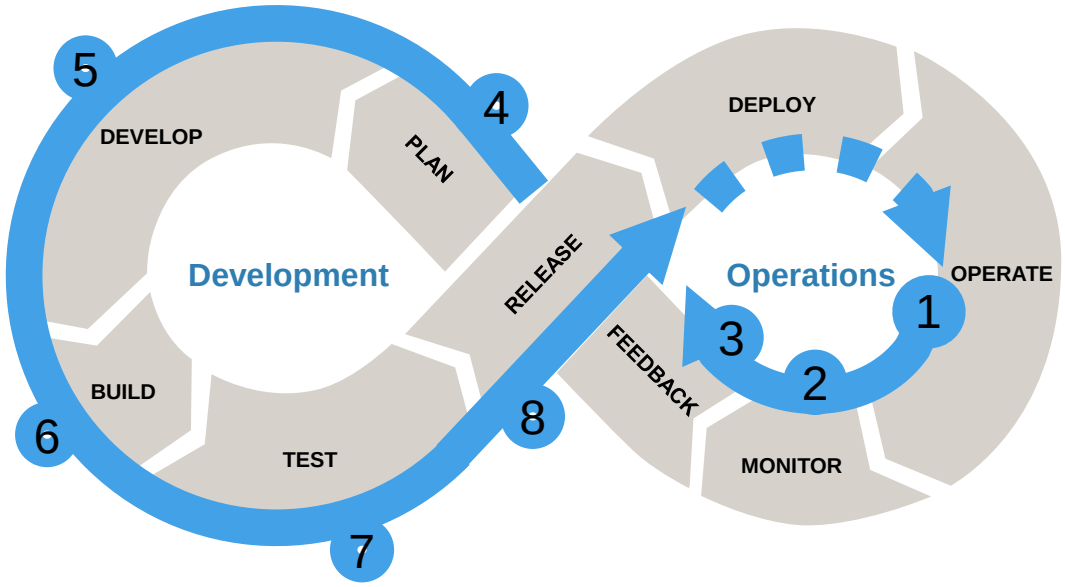




Enhance workflows to succeed with the SDV transition



Accelerate Time to Recover from Incidents



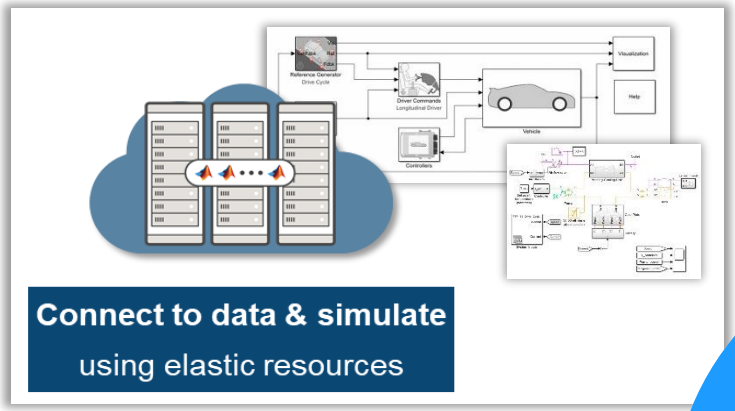
1. Incident in the fleet
2. Make data available for developers
3. Interpret data with re-simulation
4. Dependency analysis with architecture models
5. Develop update
6. Rebuild the necessary parts
7. Validate complete updated system in simulation
8. Update the necessary parts

Accelerating SDV Development with Virtual Verification

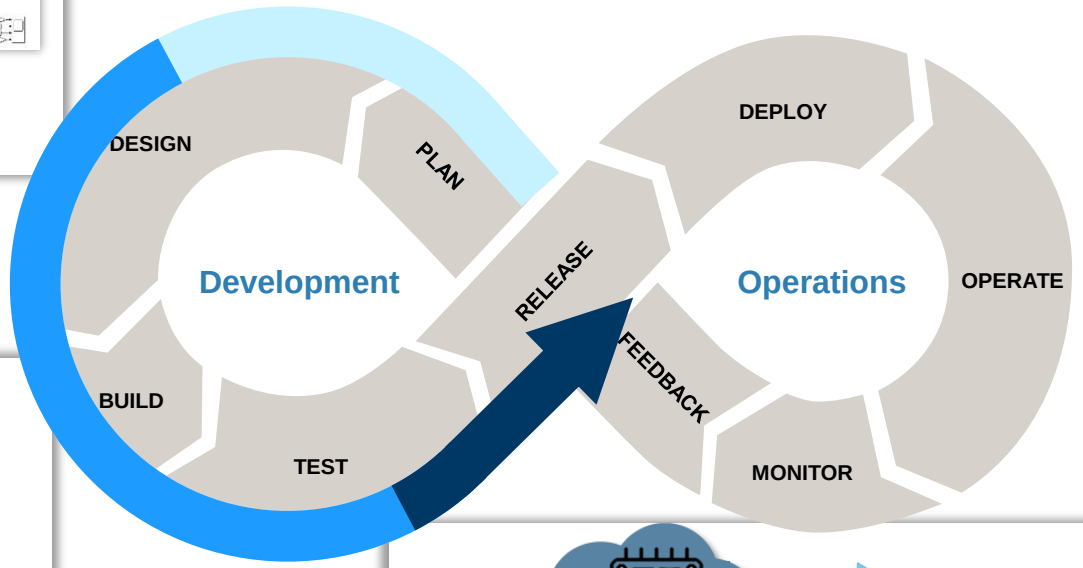
MathWorks

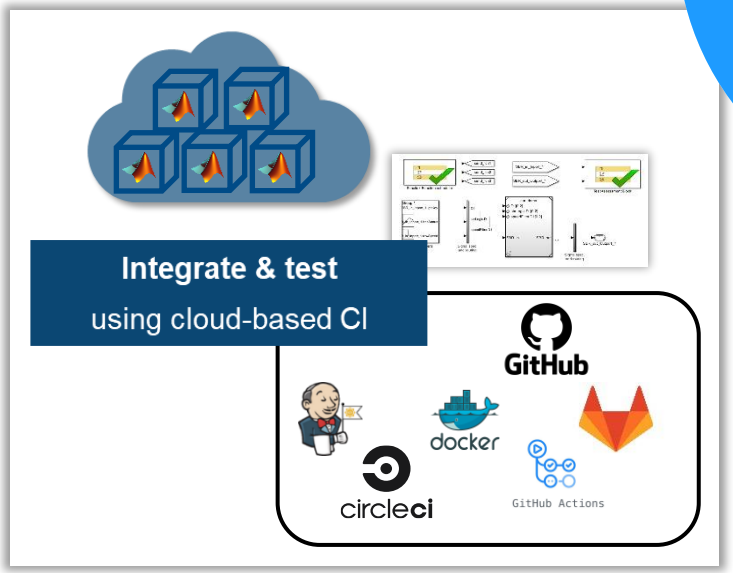
Elektrobit

aws automotive



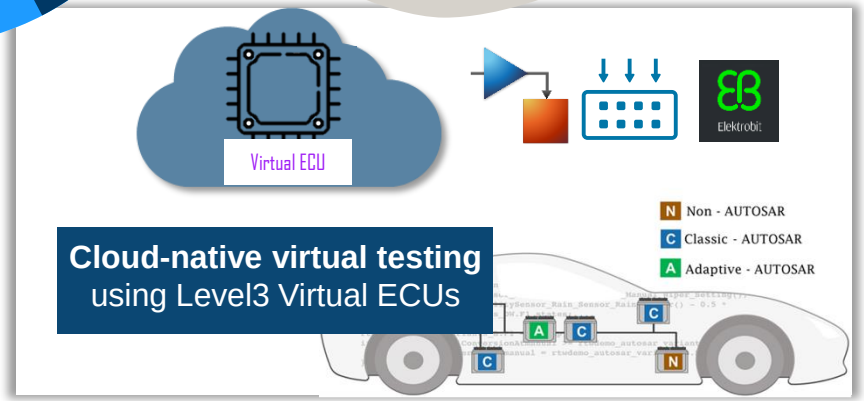
Connect to data & simulate
using elastic resources





Integrate & test
using cloud-based CI

GitHub
docker
circleci
GitHub Actions



Cloud-native virtual testing
using Level3 Virtual ECUs

Collaborate to transform development processes

- What are relevant development processes/workflows?
  - Automotive E/E architecture evolution
  - Software factory
  - Closing the DevOps loop
  - …
- What are your priorities?
- Collaborate with MathWorks to drive the transformation