Expanding DevSecOps to Embedded Systems: Is it possible? If so, how?

Hasan Yasar

Technical Director, Adjunct Faculty Member

Software Engineering Institute | Carnegie Mellon University

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM22-1109

Outline

- Overview of DevOps
- HW/SW Development, Testing & Deployment
- Solution

About me:

- 25+ years of software development experiences with EE major
- Certified Scrum Practitioner
- Certified Ethical Hacker
- Various roles throughout SDLC ; Manager, Architect, Tester, Developer, QA, IT Manager, Project Manager, VP…
- Started with waterfall in 1990
- Started with agile in 2003
- Started with DevOps in 2010
- Faculty Member on delivering DevOps course at CMU, SEI since 2015
- DevOps, DevSecOps community organizer, frequent Speaker
- PC members in various research conferences,
- Editorial board member, IJSS, AJSE
- Member of IEEE 2675 DevOps, 982.1 SW reliability, ISO WK29 Agile/DevOps WKs

Overview of DevOps/DevSecOps

What is DevOps?

DevOps is a set of principles and practices which enable better communication and collaboration between relevant stakeholders for the purpose of specifying, developing, continuously improving, and operating software and systems products and services [1]

What isn't *DevOps*?

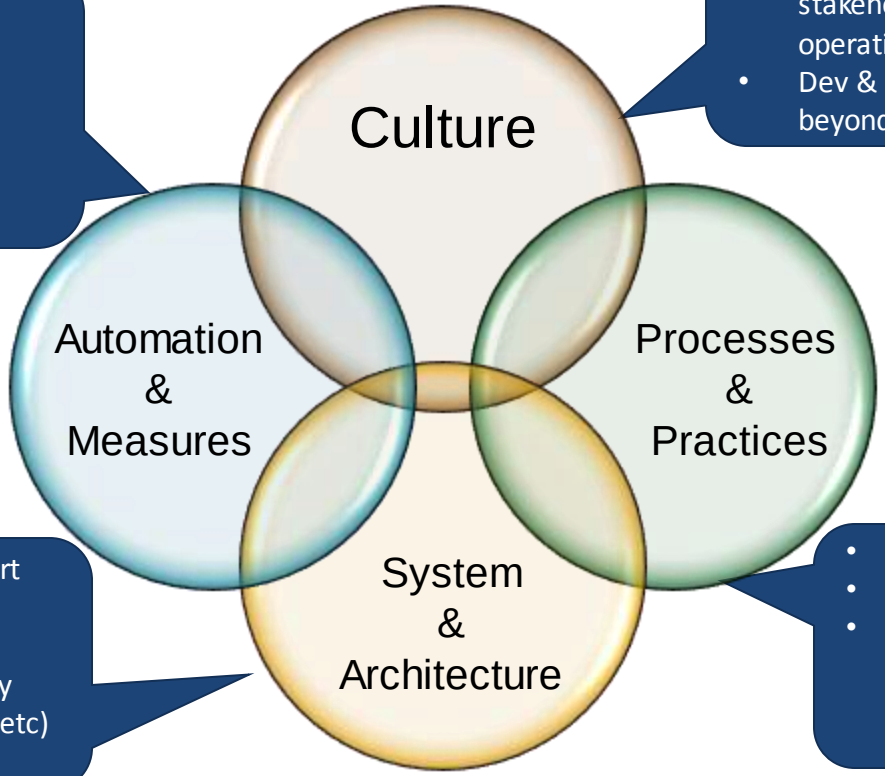
Systems Engineering, Tools, Waterfall

[1] IEEE 2675 DevOps Standard for Building Reliable and Secure Systems Including Application Build, Package and Deployment

Might Seem Simple, but not EASY!

- What Some People Think Boundaries of DevSecOps is!
- Automate repetitive, error-prone tasks
- Static & Dynamic Systems Analysis
- Performance dashboards

- All roles collaborate
- Dev, Ops, Sustainment have stakeholders that understand operational drivers
- Dev & Ops support products beyond delivery



- System architected to support integration and automation goals
- Represents important quality attributes (scalable, secure, etc)

- Value stream understanding
- Whole pipeline accounted for
- Continuous integration, automated test, virtualization, self-serve, scripting, automated deployment…

DevOps Has Four Fundamental Principles

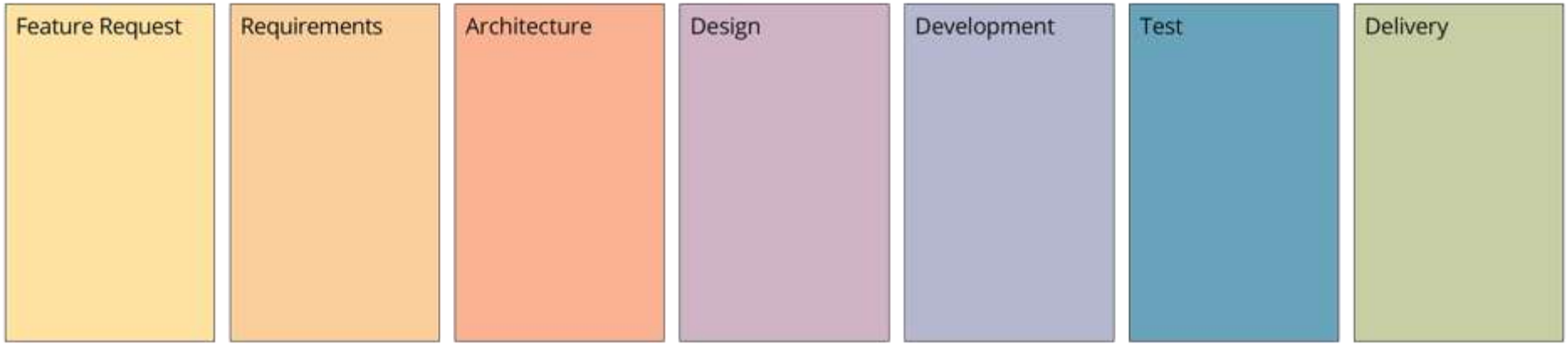
1. **Collaboration:** creating ‘cross-functional’ teams

2. **Infrastructure as Code:** all assets are versioned, scripted, and shared where possible

3. **Automation:** deployment, testing, provisioning, any manual or human-error-prone process

4. **Monitoring:** any metric in the development or operational spaces that can inform priorities, direction, and policy

Reminder: We still need to follow SW Development Phases



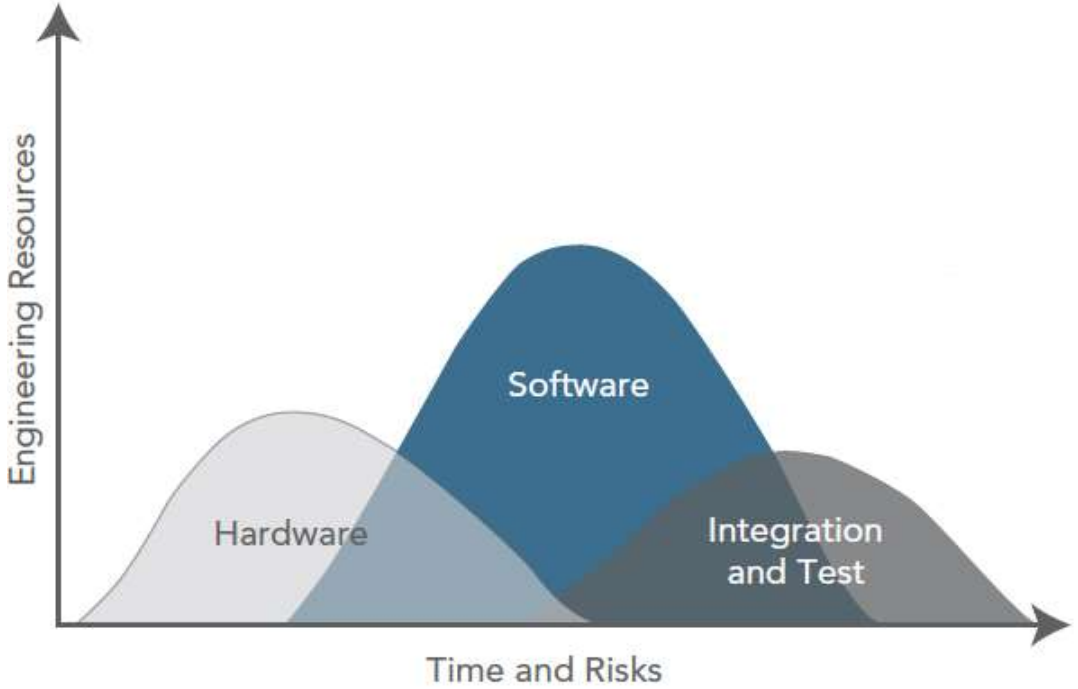
HW/SW Development & Deployment

DevSecOps & Hardware

- Modern software development benefits from DevSecOps approach:
  - Integrated development and deployment pipeline
  - Automated testing, including security
  - Continuous Integration/Deployment
- Much harder to do if hardware is involved:
  - Requires hardware testbeds
  - Complicated and unstable toolchains
  - Unstable I/O to external dependencies
  - Often slow response time

The Problem - HW /SW integration

Traditional Software / Hardware Development and Testing

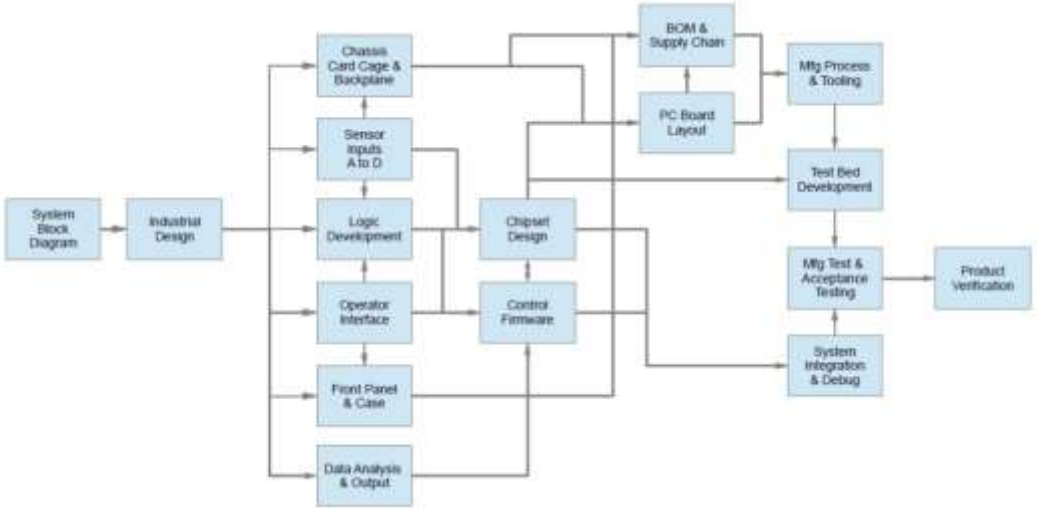


- *Embedded HW Availability* ***Delays*** *Final Integration and Test*
- Software and hardware **issues identified late** the development life cycle costing schedule and cost impact.
- HW/SW **defects released** into fielded system
- HW design spec verification **Delay**
- Software architecture **risks will not be identified and mitigated** until much later in the software life-cycle
- **Requires expensive hardware** and association maintenance
- ***Minimum support for*** *milestone with working virtual system*
- *Limited and* ***delayed*** *M&S support*

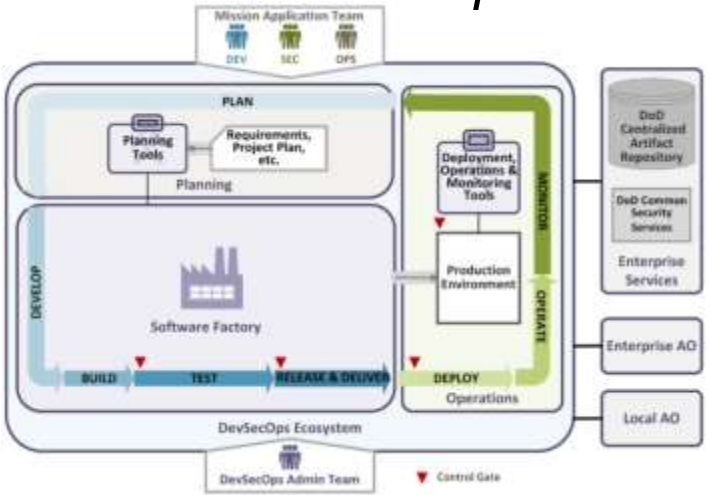
HW & SW Design flaws identification delay resulting in cost and schedule overrun

DevSecOps Helps, *But There Are Barriers*

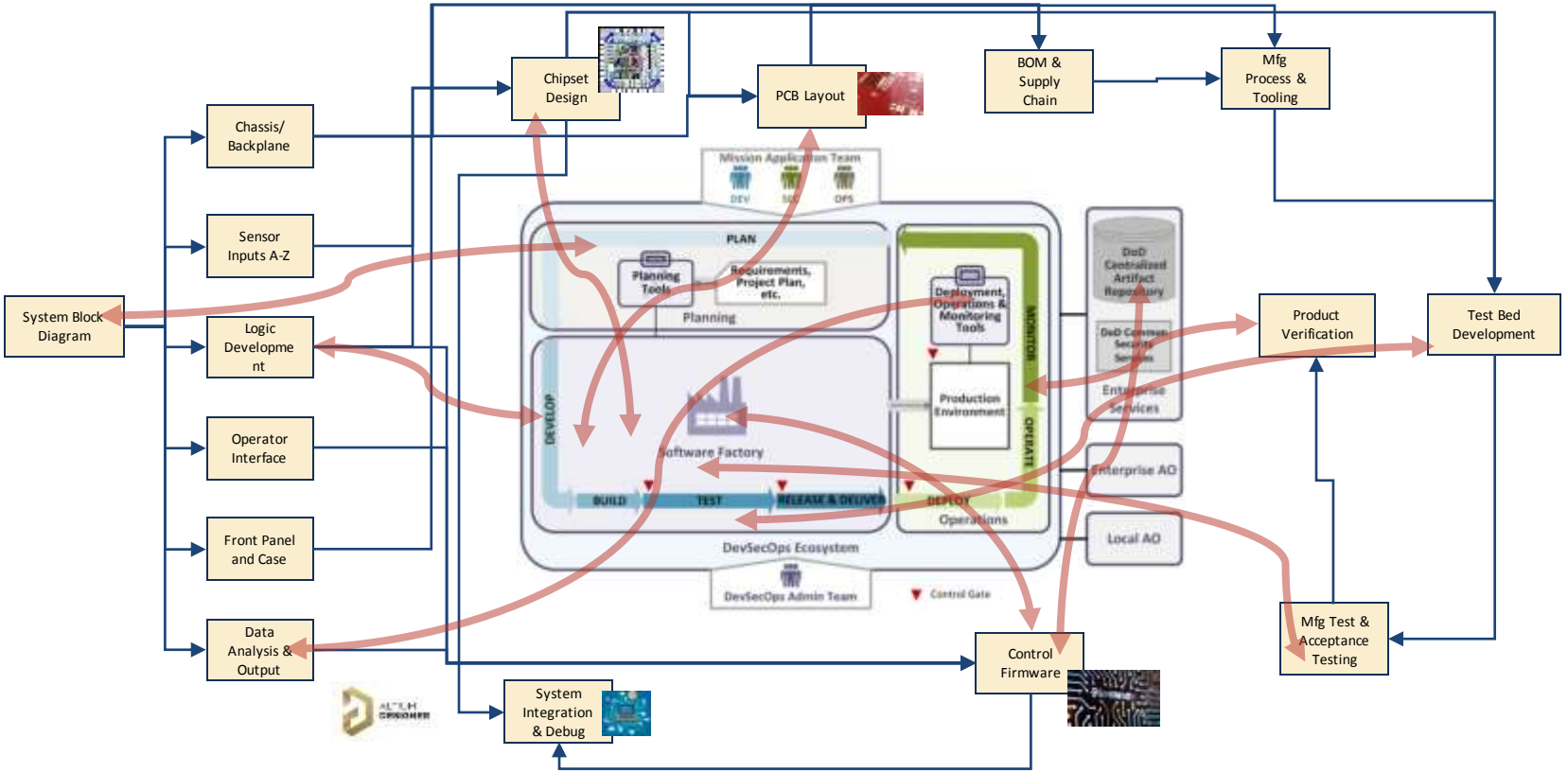
Hardware Development



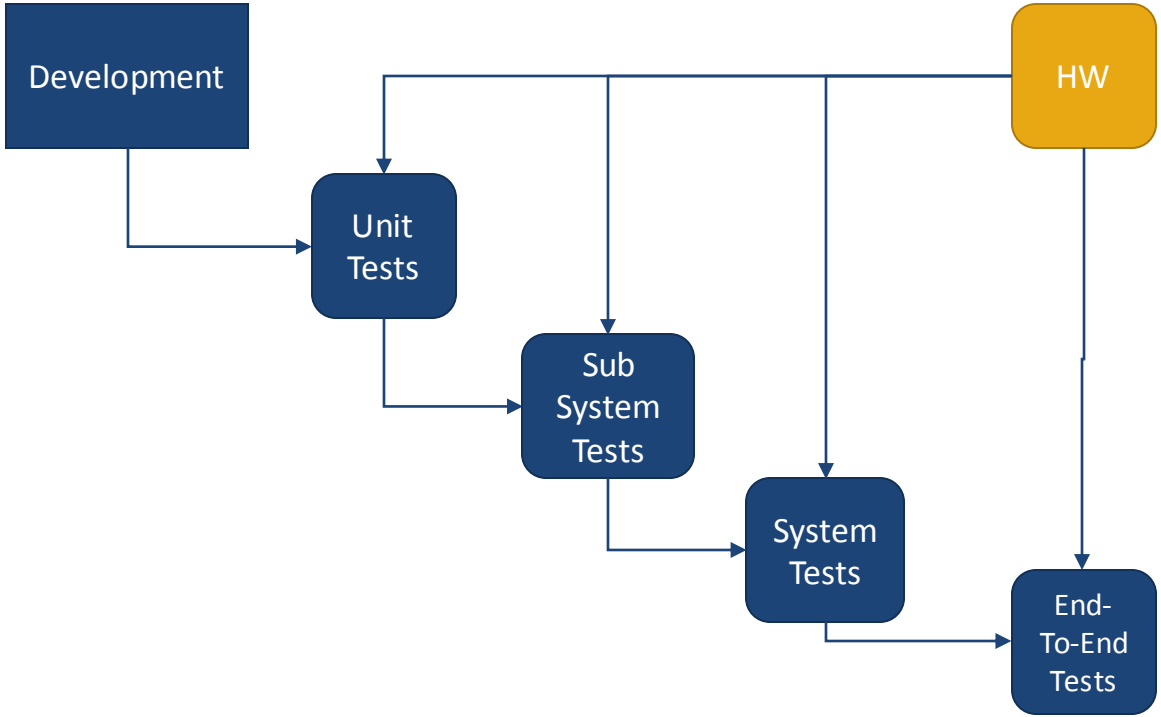
Software Development with *DevSecOps*



Breaking Barriers Between HW & SW



Typical (SW/HW) Development Process



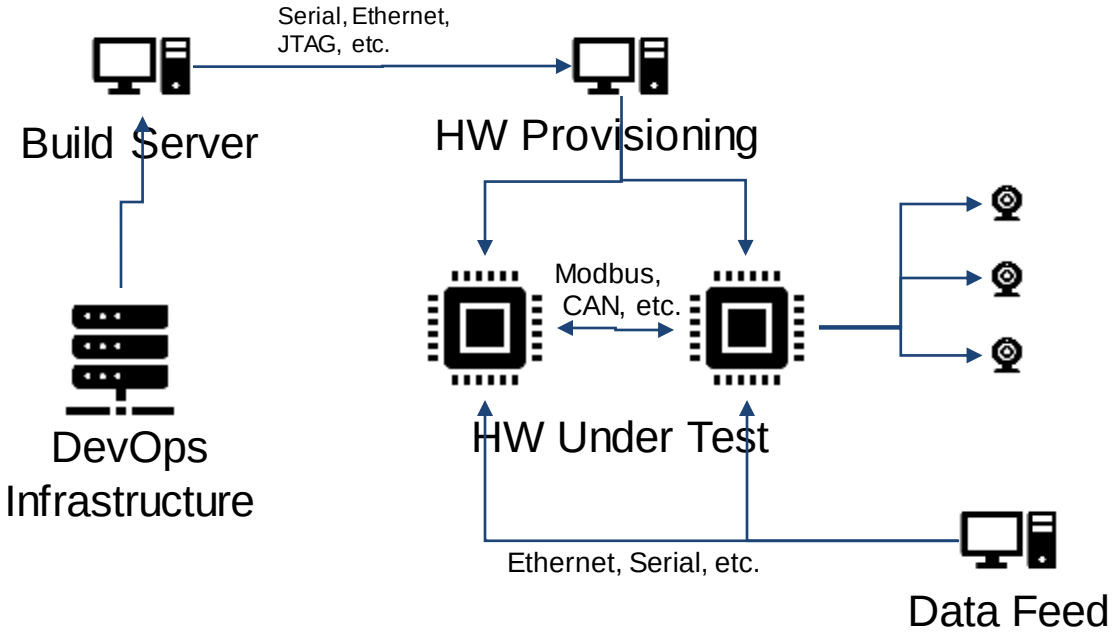
HW Development boards and prototypes

- Typically used during initial prototyping but can also extend into testing
- Convenient due to multitude of available IO options already built in
- Very limited as they usually represent the controller, no custom hardware, sensors, etc.
- Not the actual hardware, requires system level tests on actual hardware

DevSecOps and Hardware Components

- Simulate HW when you can for unit, sub and system tests
  - Don't expect for simulation to completely replace HW CI
- Don't wait until end-to-end testing to test with real HW components
  - Perform HW "arming" tests frequently
- HW/SW configuration for reliable CI is very challenging unless simulating
  - Consider full memory snapshots for SW and HW components

Hardware-Based Testing with DevSecOps



Hardware substitution with API

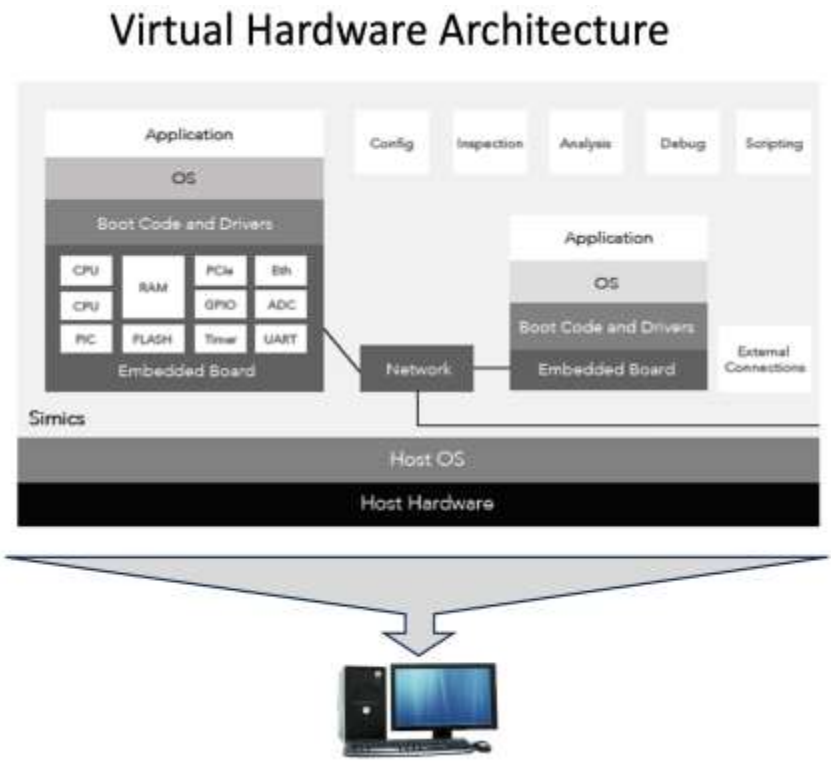
- At the edge of software and hardware
- Hardware replaced in software with a set of API calls
- Closer to simulation but easier to develop
- Not a bad option to do early unit tests on the software side only
- Does not test hardware at all and does not fully test the software either

Hardware simulation

- Fully or partially implemented hardware functionality in software
- Emulation is often down to the instruction set
- Many forms exist, each has its limitations
- Emulation introduces latency
- Significant effort to create and maintain a functional simulator
  - Many companies have a dedicated simulator “SIM” team
- Efficient development of complex systems “requires” a simulator, often custom
- Off the shelf simulators exist, provide generic simulation and test integration capabilities
  - May not be sufficient for a complex system
  - Proprietary technology maybe difficult to extent

Diving into Modelling & Simulation

- CPU/Controller emulation at instruction set level
  - By itself is not very useful for complex systems
  - Also, very powerful as it allows snapshots and replays c the entire state of the system
- Component/peripheral devices emulation
  - Serial interfaces, USB controllers, network, I2C, SPI, Flash, etc.
  - Very important for a reliable simulation
  - Also, difficult to customize
- Perfect for Kernel or embedded system development
  - Relying primarily on the CPU and I/O virtualization
  - Commercially available CPU virtualization modules
  - Virtualizes dev boards



Hybrid Approach

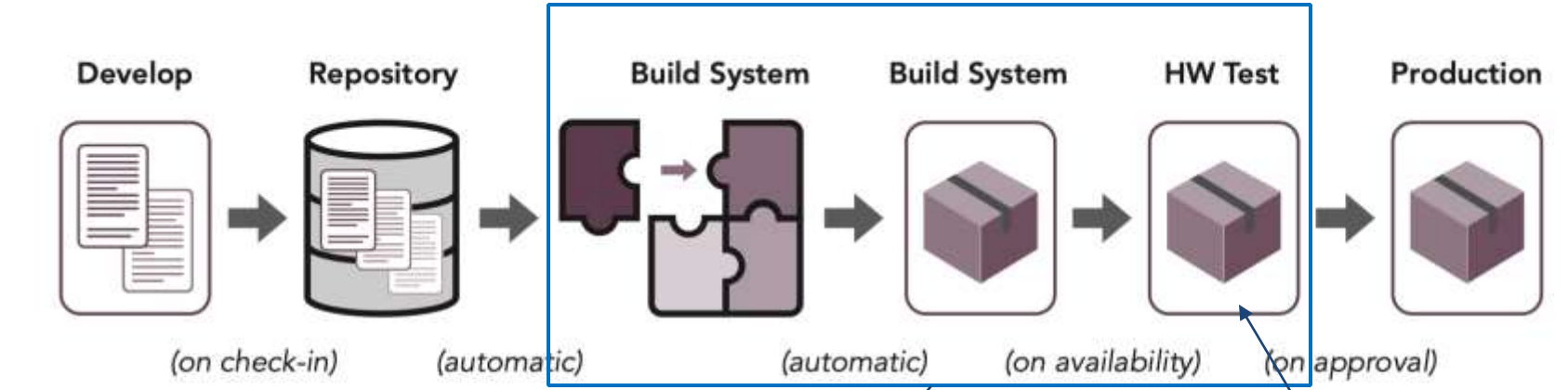
- Start up the simulation team early in the dev process
- Perform early SW development with API-based substitution and HW dev boards
- Basic simulator ready in time for sub-system and unit tests in CI
- Hardware testbed and simulator management with configuration and memory snapshots to improve test stability
- Daily/Weekly “arming” tests with real HW
- End-to-End tests on real hardware once ready

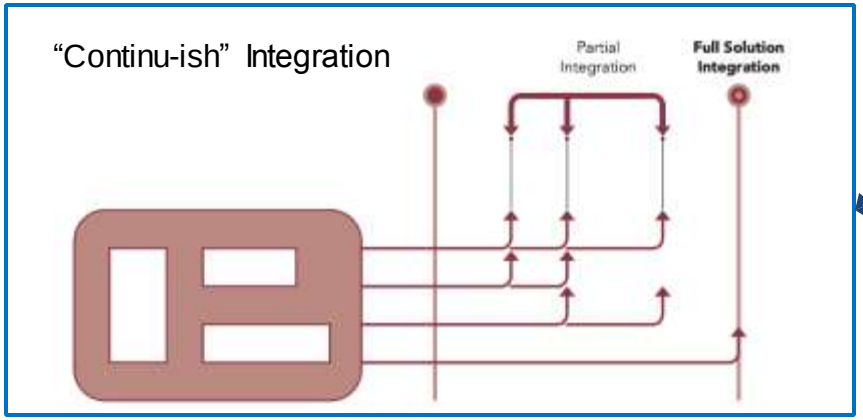
Industrial DevOps Principles* – HW/SW delivery

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. Continuous-ish Integration
8. Test Driven Development

* IT Revolution Industrial DevOps & Applied Industrial DevOps Paper

Software/System development pipelines workflow



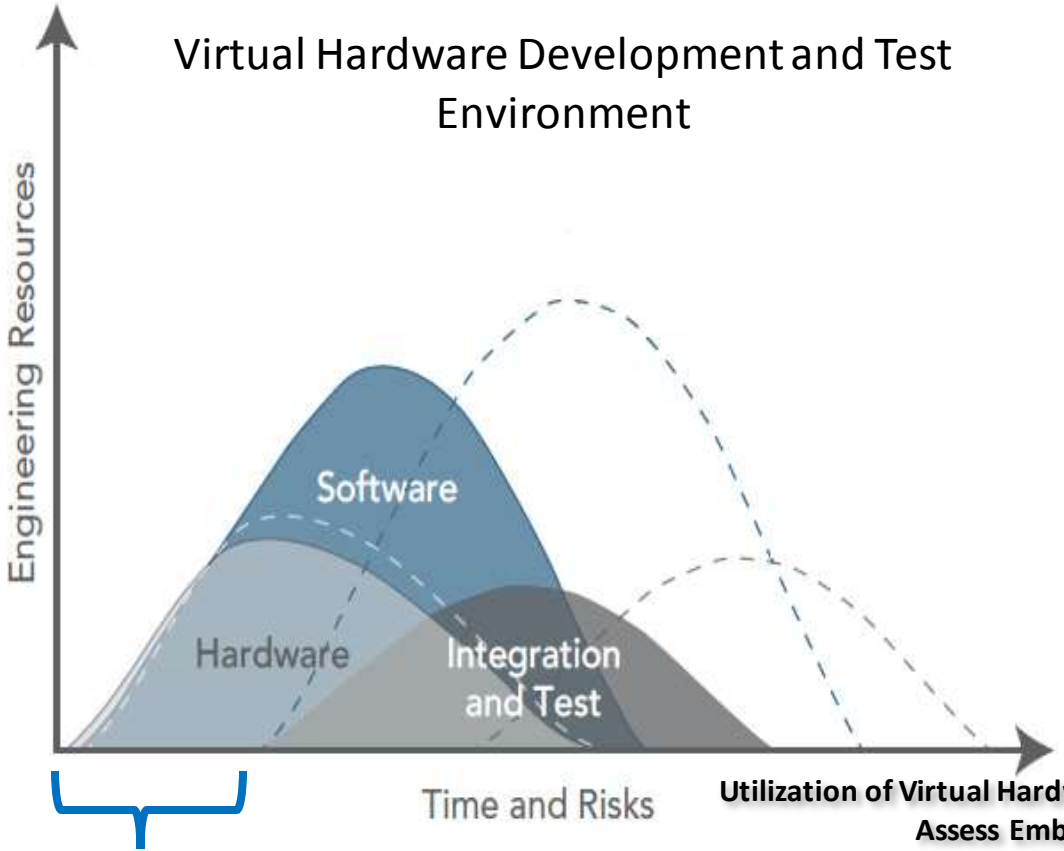


Proxy HW

Testing with Hardware

Yes it is possible!

The Solution - HW /SW integration

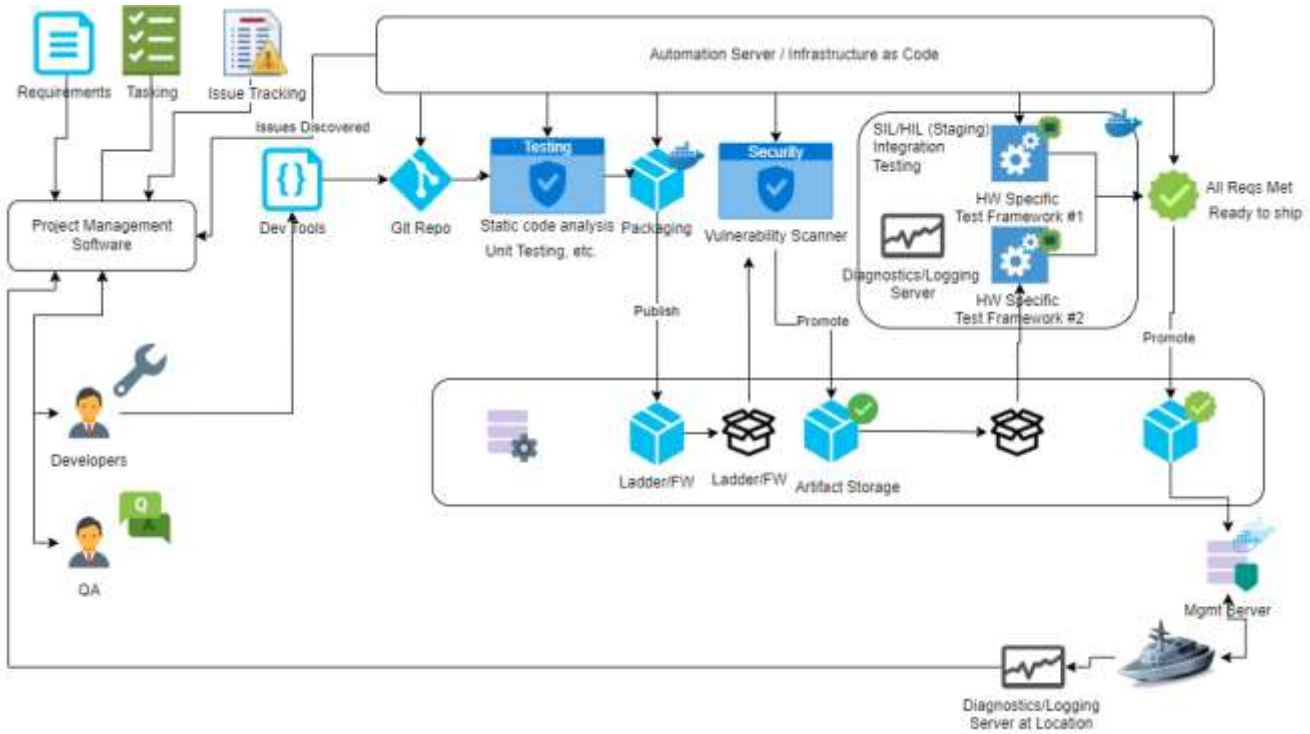


Early Virtual HW Solves the Problem:

- ✓ *Embedded HW available **Early** for SW (including firmware) & HW **Integration and Test***
- ✓ *SW & HW **defect identification early and Minimizes Rework**: Cost avoidance using virtualization design verify design meets requirements and design specification.*
- ✓ HW Support **Design Specs verification**
- ✓ *SIV&V analyst can **perform dynamic analysis***
- ✓ ***Less expensive** then hardware*
- ✓ ***Support for** milestone with working virtual system*
- ✓ ***Architecture** Risk Mitigation*
- ✓ ***Higher Fidelity** capability for M&S and Training environment **early***

Utilization of Virtual Hardware Environments will Accelerate Organization's Ability to Assess Embedded SW and Provide Detailed SW Analysis Much Earlier in the Development Cycle

Exemplary DevSecOps & HIL Development And Testing Process



Leveraging the power of HW/SW DevOps pipeline for large complex systems is an industry step change and the companies that solution this problem first will increase transparency, reduce cycle time, early HW/SW integration, test automation, increase value for money, and innovate faster.

For more information…

DevOps: https://www.sei.cmu.edu/go/devops

DevOps Blog: https://insights.sei.cmu.edu/devops

Webinar : https://www.sei.cmu.edu/publications/webinars/index.cfm

Podcast : https://www.sei.cmu.edu/publications/podcasts/index.cfm

Thank You

Hasan Yasar
Technical Director, Adjunct Faculty Member
Continuous Deployment of Capability
hyasar@sei.cmu.edu
@securelifecycle

It is question and answer time:

What does this mean to you?

How can we put these ideas into action?