Agile for High Assurance: Lessons learned

Jeffery Payne, CEO, Coveros

jeff.payne@coveros.com

@jefferyepayne

About Coveros

- Coveros helps organizations build and deliver mission-critical applications using agile

- Services
  - Agile Transformations
  - High Assurance Agile Software Development
  - Rugged DevOps Implementations
  - Agile Testing & Automation
  - Agile Coaching

- Agile, DevOps, App Security Training

- SecureCI – Open source CI/CD toolchain

- GSA Schedule 70 holder, TS Facility Clearance, DCAA approved rate structure

Areas of Expertise

Agile High Assurance Case Study

- Coveros hired by medical device company in early 2015 to transition their software division to an agile process while assuring regulatory compliance
  - FDA Category III Medical Device (Safety-critical)
- Performed agility assessment from Feb – March 2015
- Recommendations report April 2015
  - Improvements to people skills
  - High Assurance agile process/practices
  - Cultural & organizational shifts
  - Recommended automation & tooling

Agile High Assurance Case Study (cont.)

- Began implementation of high assurance agile process in March 2015 and completed release of FDA approved version of medical device software in November 2015
  - Led agile pilot project leveraging new high assurance agile process
  - Provided agile coaching for teams
  - Supplied agile engineers to model high assurance agile behavior
  - Implemented test automation and DevOps capabilities to accelerate delivery
- Completed agile rollout and transformation In June 2016
- Software organization has fully transitioned to agile and is now spreading agile principles to hardware engineering and other business functions

Lesson #1: Agile works just fine for HA

- Articles written on why agile doesn't work for high assurance often cite reasons that are common Agile Myths:
  - Agile doesn't value planning
  - Agile doesn't believe in documentation
  - Agile is ad-hoc and doesn't support secure design
  - Agile doesn't provide sufficient software assurance

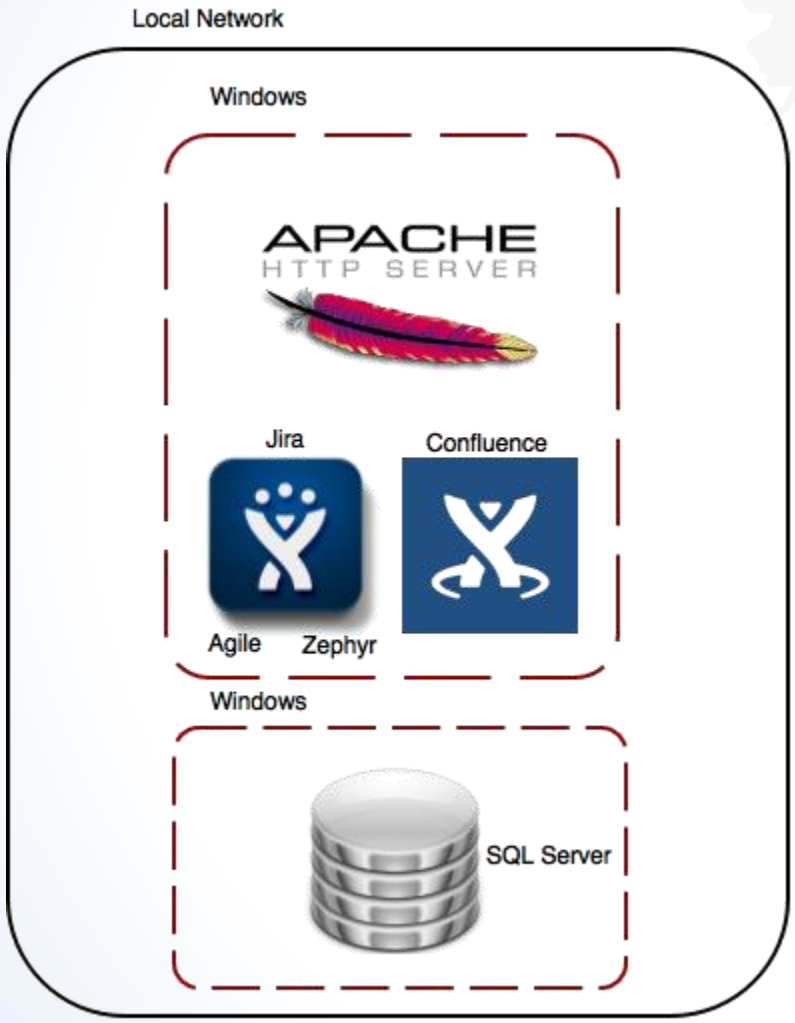
- Agile Principles call out values that support high assurance:
  - Our highest priority is to **satisfy the customer** through early and continuous delivery of **valuable software**.
  - **Working software** is the primary measure of progress.
  - Continuous attention to **technical excellence and good design** enhances agility.
  - The **best architectures, requirements, and designs** emerge from self-organizing teams.

Lesson #2: You must accelerate your documentation process

- Auditors demand that proper documentation exists for high assurance software
  - Evidence you followed your defined process
  - Evidence your requirements trace to design/code, tests, and test results
  - Evidence your assurance practices meeting regulatory requirements

- Documentation for Agile High Assurance
  - Understand what documentation is needed when by auditors, regulators, compliance officers, etc.
  - Plan time in Sprints for lightweight documentation tasks associated with each Story you design, build, and test
  - Leverage tooling to maintain traceability and lightweight documentation
  - Automated document creation

Documentation for FDA medical device



Document

Generation

- Safety case
- Traceability docs
- Design docs
- Test reports
- FDA Pre-Market Approval (PMA)

Lesson #3: Continuous assurance IS critical

- The only way to avoid costly (and lengthy) end-of-lifecycle assurance activities is to automate your build, test, delivery process and integrate software assurance activities as you go.
- Even though you cannot 'continuously deploy' into high assurance environments, you should assuring software during every check-in build and continuous integration cycle.
- Testing activities that are too long to be performed as part of check-in builds should be done during continuous hourly / nightly / weekend regression testing runs performed during each Sprints.
- Always continuously deliver release candidates to downstream test environments as a means of practicing deployments and mitigating the risk of faulty deployments

Continuous assurance for FDA medical device

Development teams

git

Code repository

Windows slave

Jenkins

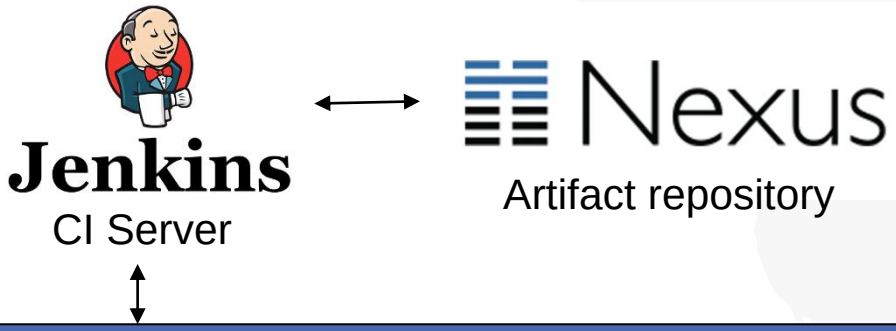
CI Server

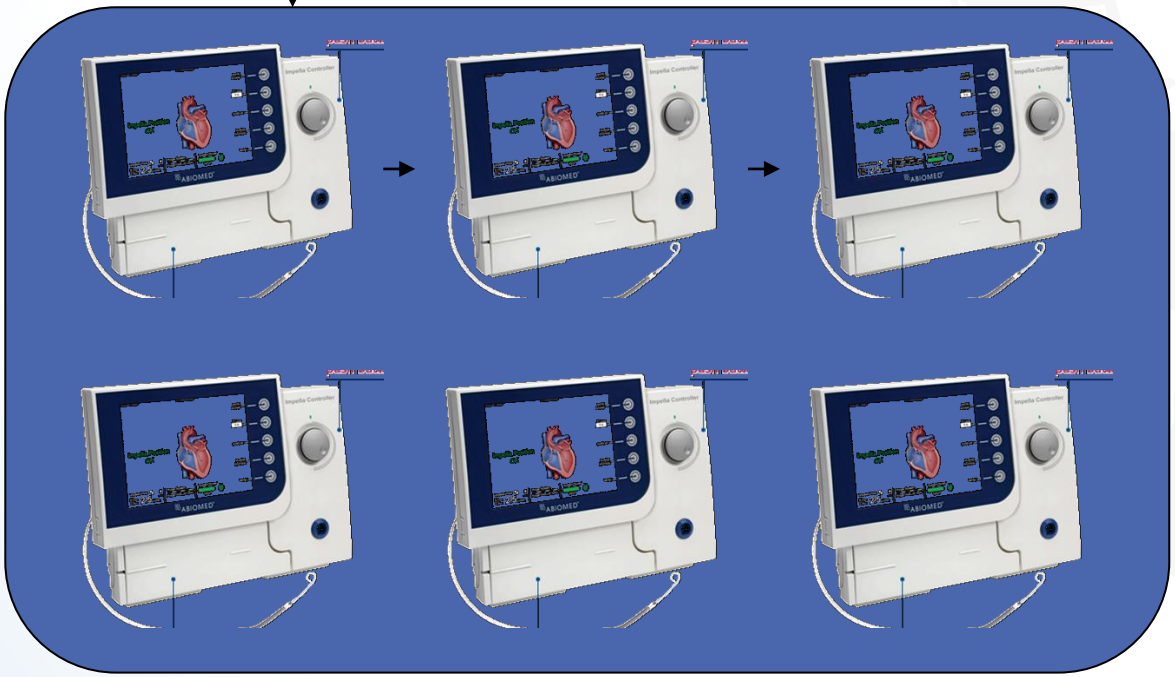
Nexus

Artifact repository

Continuous Integration

Continuous assurance for FDA medical device





Enabling capabilities:

- 'Expect' to provide interactive scripting
- 'sh' to manipulate console
- IP locks to notify jobs a console is busy

Lesson #4: Integrate independent teams into Sprints

- Often independent teams are a hurdle to speedy delivery as they are late in the lifecycle
  - Independent test
  - Information assurance
  - Internal audit / compliance
- Incorporate independent groups into Sprints and allow them to audit / review automated scripts and lightweight documentation builds their confidence in you
  - Ideally independent teams work side-by-side with your dev team
  - This cultural shift will often take some time to happen but keep pushing
- Support their efforts by automating their manual activities to increase their effectiveness and to reduce delays.
  - The more you understand about their process, the better able you are to help

Thank You!

More Info: jeff.payne@coveros.com

 @jefferyepayne