

Continuous Integration

Bringing Medical Device Software
Development out of the **DARK AGES**

Jeff Gable

IT application development circa 2005

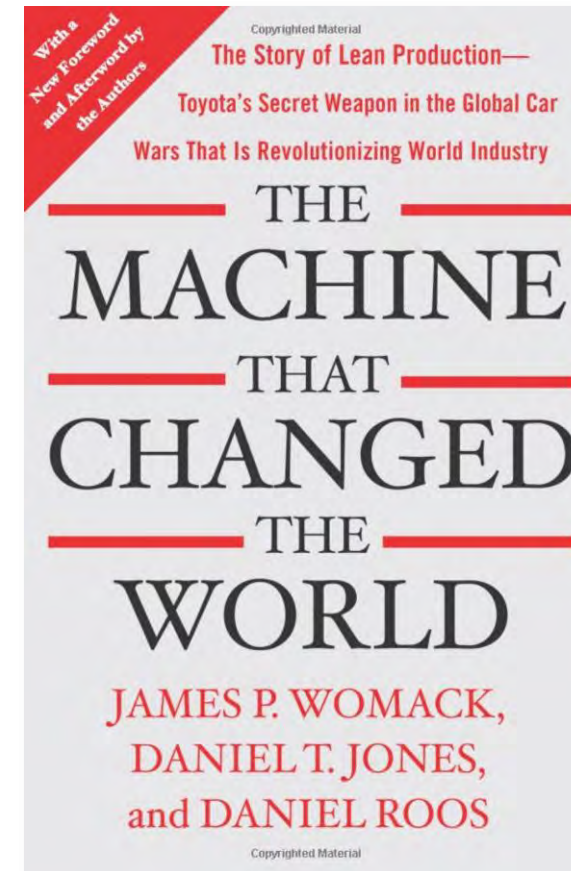
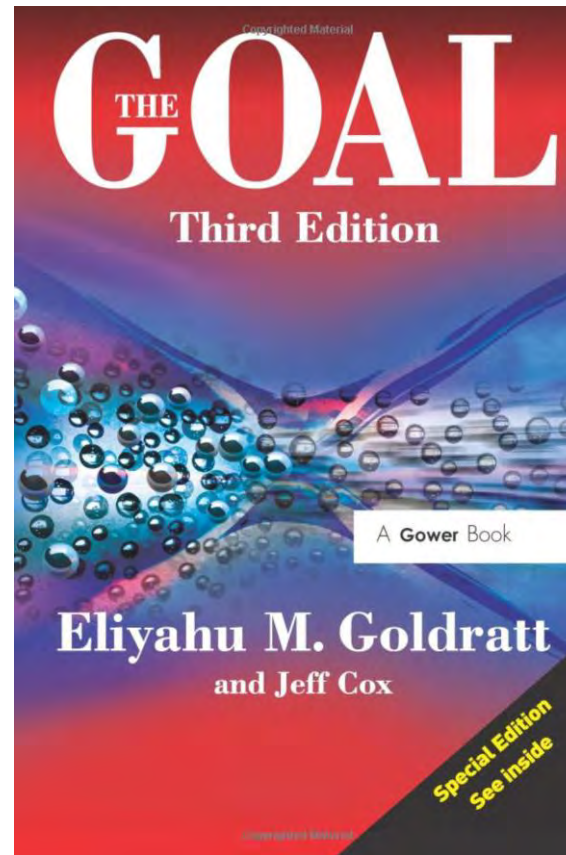
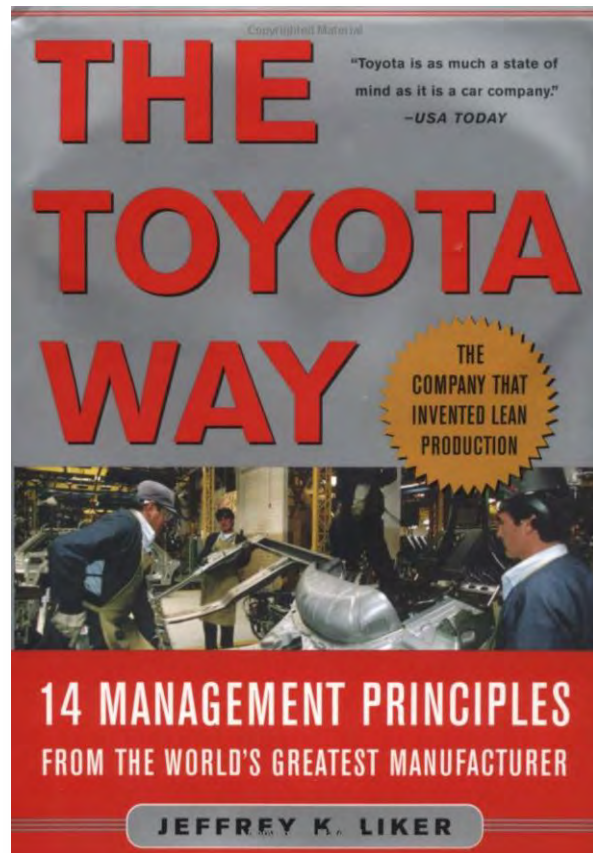
- Painful merges of development branches
- **“Throw it over the wall” attitude**
- Manual, error-prone deployment

IT application development circa 2005

Low-quality software

Iteration time measured in months

Apply Principles from Lean Manufacturing



Devops Revolution

Apply Systems-Level Thinking

Improve Feedback Loops

Foster Culture of Continuous Improvement

Devops Revolution

Apply Systems-Level Thinking

Improve Feedback Loops

Foster Culture of Continuous Improvement

Devops Revolution

Apply Systems-Level Thinking

Improve Feedback Loops

Foster Culture of Continuous Improvement

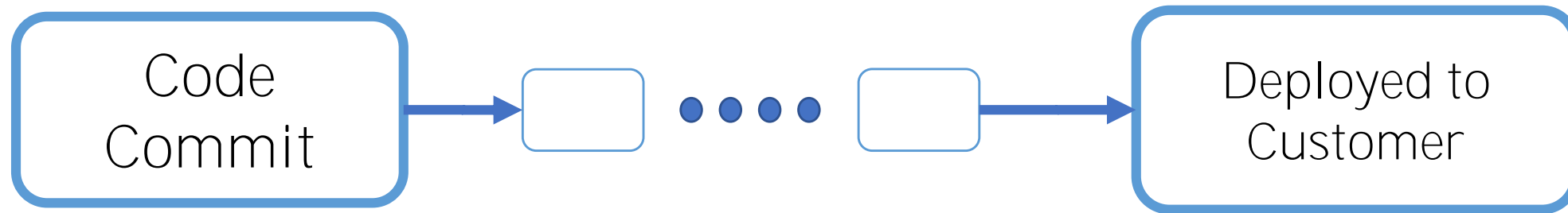
Devops Revolution

Apply Systems-Level Thinking

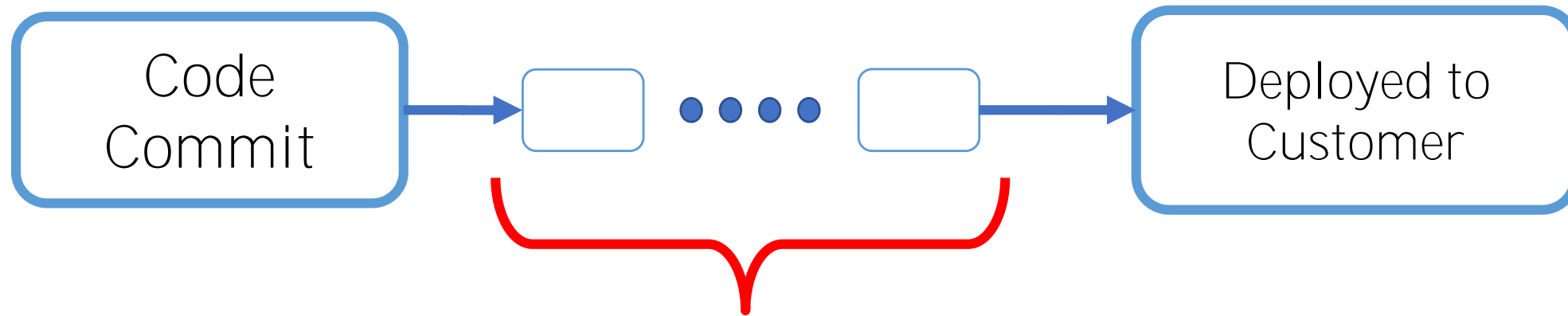
Improve Feedback Loops

Foster Culture of Continuous Improvement

Continuous Integration



Continuous Integration



Automate all of the things

Continuous Integration

Infrastructure as code

All configuration for deployment defined in readable text configuration files and stored in version control with application code

Continuous Integration

Infrastructure as code

All configuration for deployment defined in readable text configuration files and

stored in version control with application code

Repeatable Deployments

Continuous Integration: Results

High-quality software

Iteration time measured in hours

Continuous Integration: Results

Company	Deploy Frequency	Deploy Lead Time	Reliability, Quality, and Responsiveness
Typical Enterprise	Every 9 months	Months	low
Amazon	23,000 / day	Minutes	high

Continuous Integration is now
the industry standard
in web development



How can we apply these lessons
to software development
for medical devices?

Prerequisites

- 1) Software in Version Control
- 2) Fully Automated Build Process

Add Automated Testing

Unit Tests

- Break dependency on hardware with stubs, mocks
- Practice Test Driven Development

Can't be done? Not worth the effort?

Bull\$#!

%

Add Automated Testing

Integration Tests

- Run special test images on real hardware to exercise hardware
- Add scripting and data instrumentation to embedded code to enable automation

Infrastructure as Code

Build and Development Environments can be virtual machines that are defined with readable text configuration files.

Everyone has the
same build environment

Get Started with Continuous Integration

- Buy a build server
- Install CI Software (Jenkins, GitLab)
- Start with build only, email on failure
- Slowly add tests, defined infrastructure, etc.
- Invest 20% of effort in improving process over time. Slow, steady progress wins.

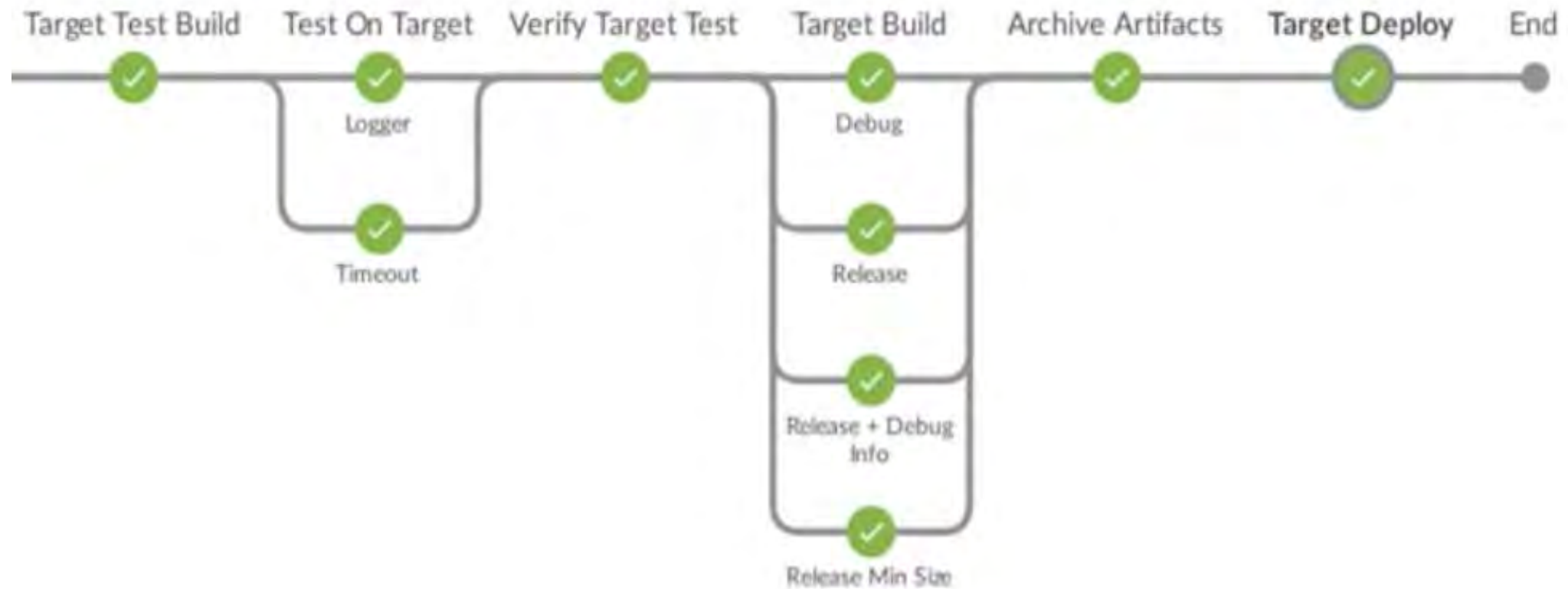
What's possible



What's possible



What's possible



Open Questions:

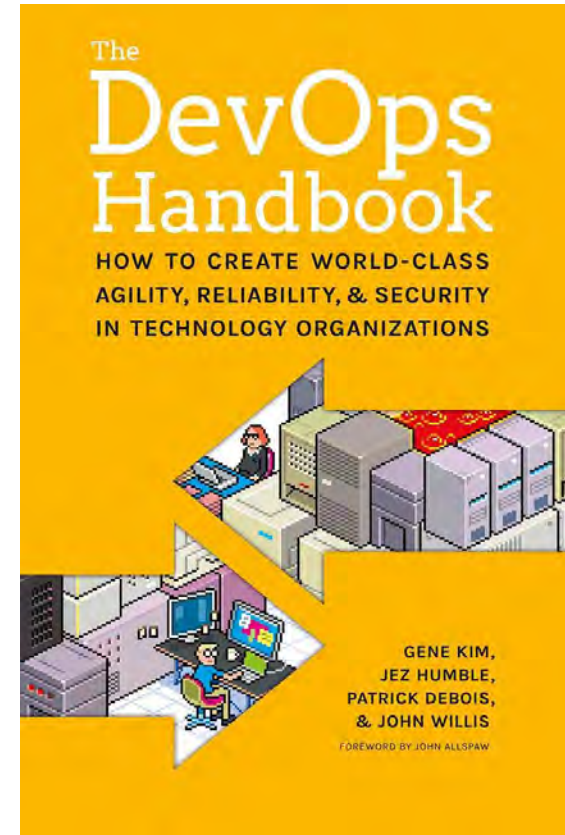
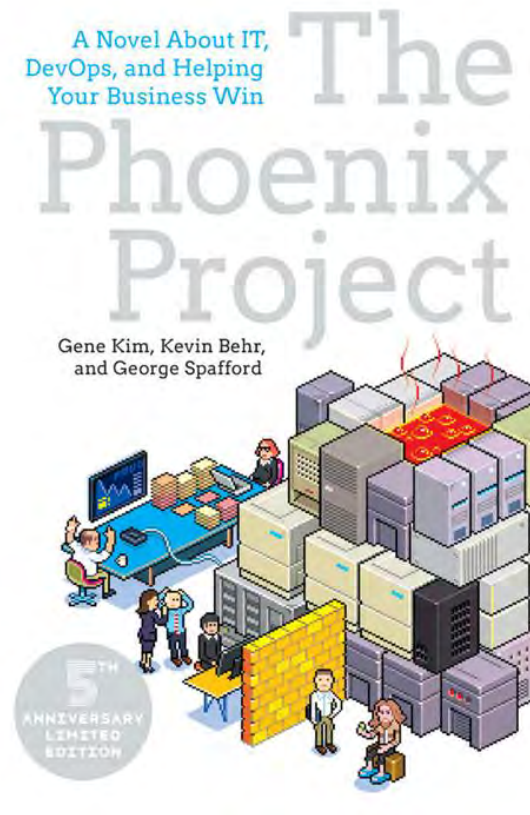
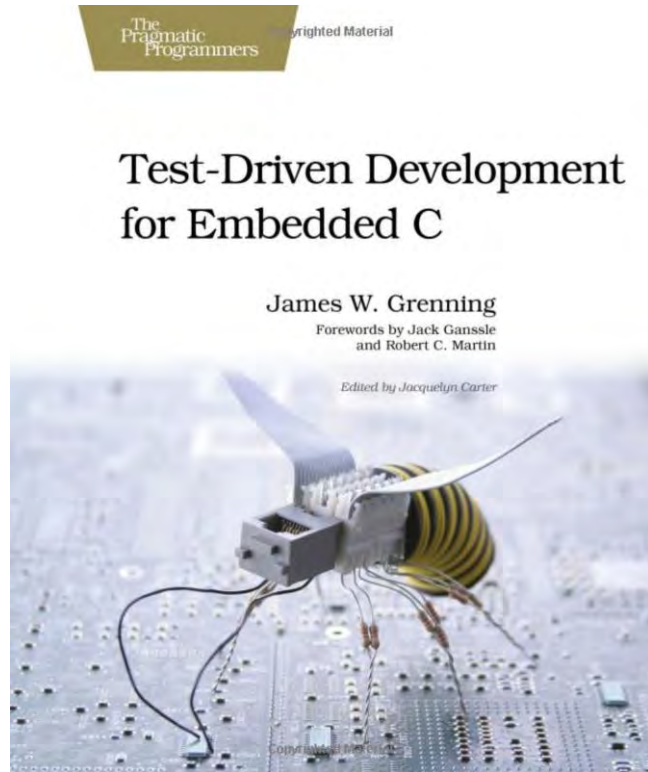
If FDA is part of the deployment chain, what steps can you take to minimize that time?

Open Questions:

Imagine what can be possible with over-the-air updates. Can medical devices achieve continuous delivery of software updates?

Example: Tesla

Learning Resources



Thank you!

Jeff Gable

website: jeffgable.com

email: jeff@jeffgable.com

Phone: (949) 226-4574
