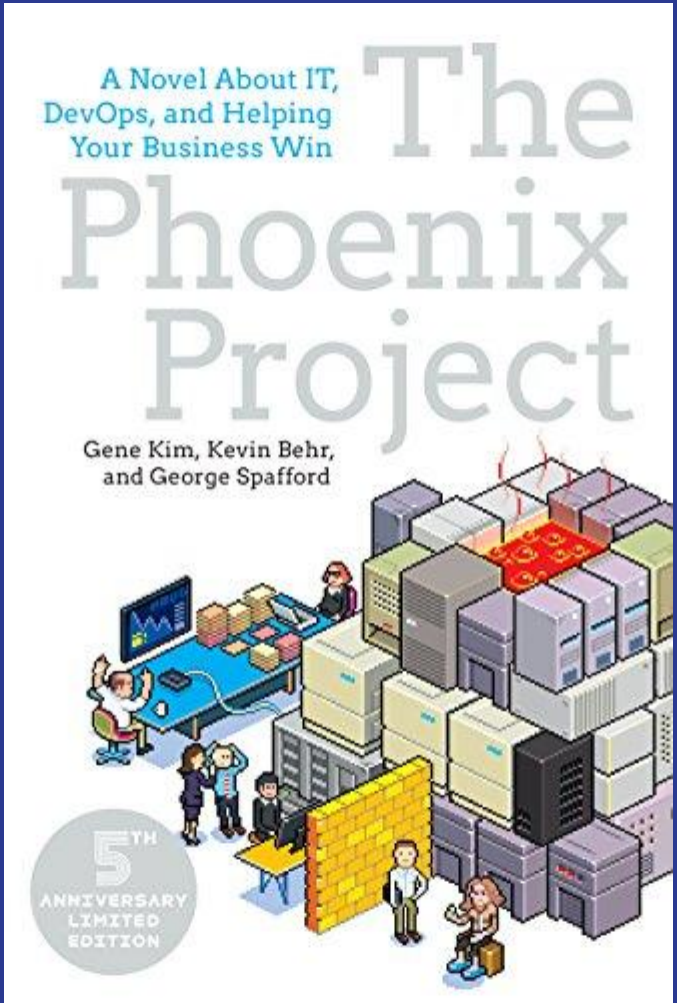
What You Should Know About DevOps:

The Importance of Company Culture

Cam Herringshaw
camherringshaw@gmail.com

Motivation for this talk

As developers, we tend to focus on the technical aspect of DevOps, and often brush over the fact that it’s as much of a philosophical movement as a technical one.

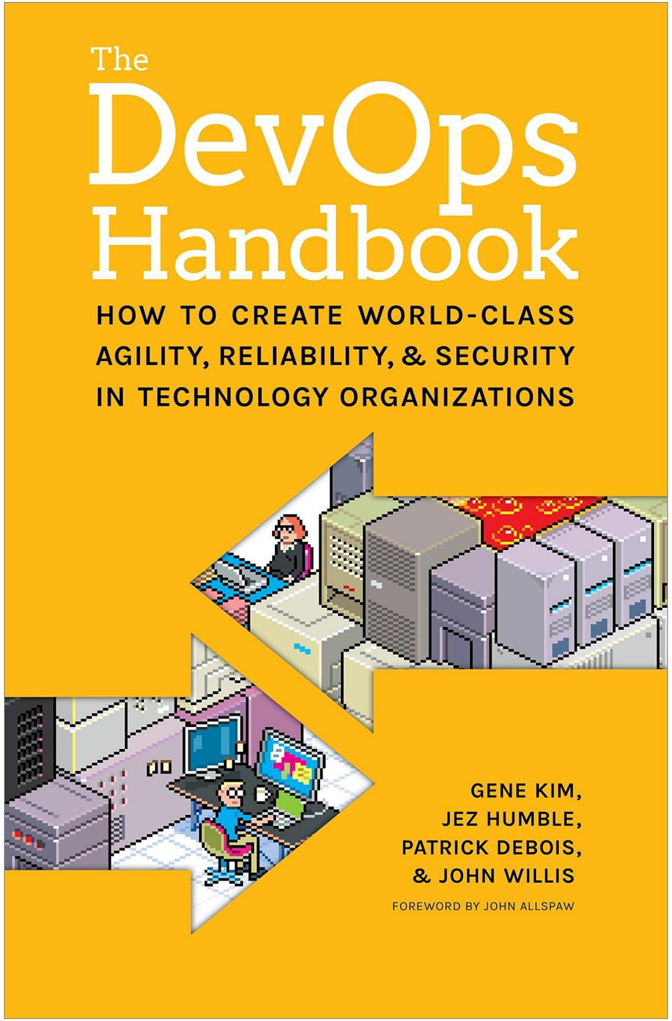


A Novel About IT, DevOps, and Helping Your Business Win

The Phoenix Project

Gene Kim, Kevin Behr, and George Spafford

5TH ANNIVERSARY LIMITED EDITION



The DevOps Handbook

HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS

GENE KIM,
JEZ HUMBLE,
PATRICK DEBOIS,
& JOHN WILLIS

FOREWORD BY JOHN ALLSPAW

What is DevOps?

What is DevOps?

“DevOps is the combination of cultural philosophies, practices, and tools that increases an organization’s ability to deliver applications and services at high velocity… [enabling] organizations to better serve their customers” - AWS

“DevOps isn’t about automation, just as astronomy isn’t about telescopes” - Christopher Little (Gartner Analyst)

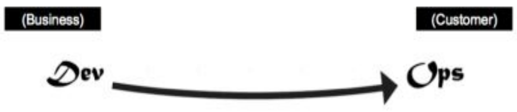
“A much expanded understanding of the value of collaboration between development and operations staff throughout all stages of the development lifecycle when creating and operating a service” - The Agile Admin

The Three Ways

Flow

Systems Thinking

Think about performance in terms of the entire system, not just one area/team

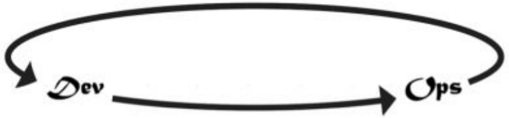


The First Way:
Systems Thinking

Feedback

Amplify Feedback Loops

Create feedback loops, then shorten and amplify

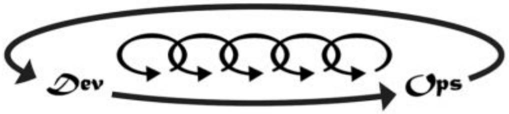


The Second Way:
Amplify Feedback Loops

Continual Learning

Experimentation

Take risks and learn from failure



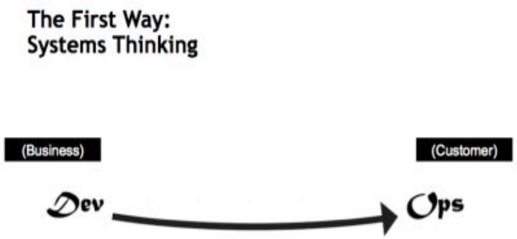
The Third Way:
Culture Of Continual Experimentation And Learning

Flow

Flow

Systems Thinking

Think about performance in terms of the entire system, not just one area/team



- Often talked about in terms of “value streams”
- Emphasize communication and collaboration between teams
- Never pass a known defect downstream

Technical Aspects of Flow

- Identify key teams, people, and processes
- Where are things handed off?
- What are restrictions at each handoff?

Starting a DevOps Culture - Flow

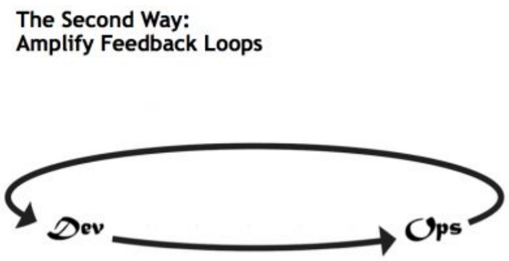
- Knowledge-focused and blame-free culture
- Emphasize communication
- Celebrate other team’s achievements

Feedback

Feedback

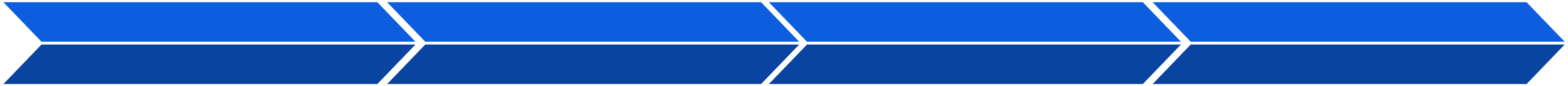
Amplify Feedback Loops

Create feedback loops, then shorten and amplify



- Create a “right to left” feedback loop
  - User testing
  - Code Review
  - Business cases
- Effects:
  - Understand and respond to all customers (internal and external)
  - Embedding knowledge where needed in teams

Flow



Product/Design

Identify requirements, talk to customers, create wireframes

Build

Develop, build the feature

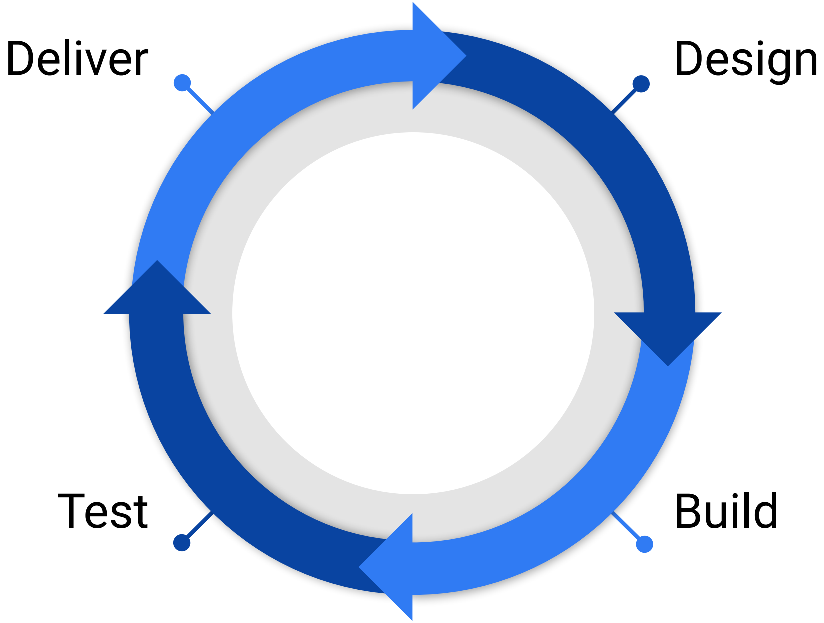
Test

Make sure it meets all requirements, passes all tests, legal restrictions

Deliver

Allow customers to access the new feature

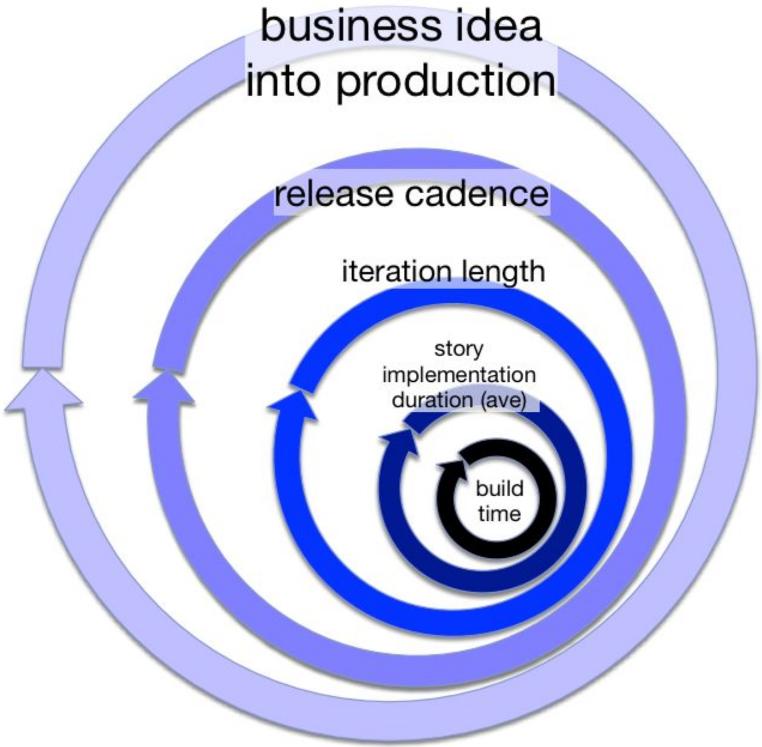
Creating Feedback Loops



- Establish feedback loops between teams
- Give each team an idea of how their work affects others (up- and down-stream)

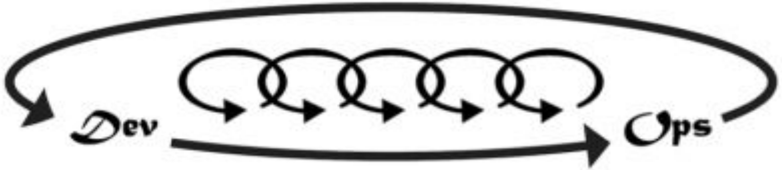
Shrinking Feedback Loops

- After initial feedback loops are in place, work on making them smaller and faster
- Embedded teams are really useful here


business idea into production
release cadence
iteration length
story implementation duration (ave)
build time

Experimentation

The Third Way:
Culture Of Continual Experimentation And Learning



- Culture is critical here
- Encourage risk-taking (within reason)
- Don’t blame or shame people for failures - learn from the failures

Experimentation

- "It's not a public flogging: the goal is to build up collective wisdom. So the response to an error is never, "No big deal, don't beat yourself up about it." It's "Let's pull on that" - the idea being that a mistake is like a loose thread you should tug on, hard, to see if the whole fabric unravels." - Chris Hadfield on NASA mission debriefs, *An Astronaut's Guide to Life on Earth*

- "Management has to create a climate where owning up to mistakes is permissible and colleagues have to agree, collectively, to cut each other some slack."

Summary

- Flow - get teams talking to each other, improve process for handoff
- Feedback - learn from what happens at each stage in the process
- Experimentation - encourage people to find what works for your team

Takeaways

- DevOps is a cultural endeavor, as much so (or more!) as a technical endeavor
- Feedback should be given/received as often as possible

Questions?

camherringshaw@gmail.com

Resources

https://devops.com/author/clittle/

https://medium.com/@neonrocket/devops-is-a-culture-not-a-role-be1bed149b0

https://docs.microsoft.com/en-us/azure/devops/learn/what-is-devops-culture

https://www.ca.com/en/blog-automation/what-is-devops-culture.html

https://www.excella.com/insights/5-key-aspects-of-devops

http://itrevolution.com/the-three-ways-principles-underpinning-devops/

https://blog.scottlogic.com/2018/03/13/organizations-journey-to-devops-culture.html

https://mckellpartners.com/publications/retirement-planning/what-job-did-you-hire-that-milk-shake-for/