CHICAGO
INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE 2015

goto; conference

Microservices and Evolutionary Architecture

Rebecca Parsons

WHY DO I CARE?

Organizations expect business agility, time to market, and the ability to adapt quickly from IT

WHAT DID WE USED TO DO?

Remember component re-use?

Remember integration by database?

Remember SOA?

SOA GOT SOME THINGS RIGHT

- Break monoliths down into independent services
- Integration over internal coupling
- Encouraged greater acceptance of eventual consistency
- At least made us think differently about the old approach

SOA STILL TENDED TO MESS UP

- Monster services
- Producer driven services
- Orchestration
- Tooling often got in the way
- Much harder to change (even though that's why we did it)
- Much harder to test

AND NOW?

Micro-services

WHAT ARE MICRO-SERVICES?

… an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API.

martinfowler.com/articles/microservices.html

CHARACTERISTICS OF MICRO-SERVICES

- Small
- Built around business capabilities
- Independently deployable
- Little centralized management
- Smart end points and dumb pipes
- Lack of centralized, shared database

IMPLICATIONS OF USING MICRO-SERVICES

- Granularity question is crucial
  - Must balance cohesion and coupling
- Independently scalable
- Monitoring crucial
- Explicit design for service failures
- Infrastructure automation and deployment automation essential
- Platform flexibility must be managed
- Eventual consistency must be managed
- Interface churn must be managed

HOW DO I DECOMPOSE MY MONOLITH?

- Think about bounded contexts as defined in Domain Driven Design
- Think about business capabilities
- Think about what consumers need
- Think about communication patterns
- Think about data architecture
- Think about correlated change patterns
- Be prepared to combine services and split services
- Tolerant reader

MICRO-SERVICES AND EVOLUTIONARY ARCHITECTURE

- Micro-services exhibit many of the principles of evolutionary architecture
  - Focus on evolvability
  - Tolerant reader
  - Exploiting Conway's Law
  - Appropriate coupling
  - Contracts
  - Testability

ROLE OF CONTINUOUS DELIVERY

- Micro-services increase the burden on operations
  - More things to deploy
  - Monitoring must be more sophisticated
  - More permutations of failure
- Inadvisable without a strong DevOps culture
- Inadvisable without rigor of continuous delivery
  - Infrastructure automation
  - Deployment automation
  - Test automation

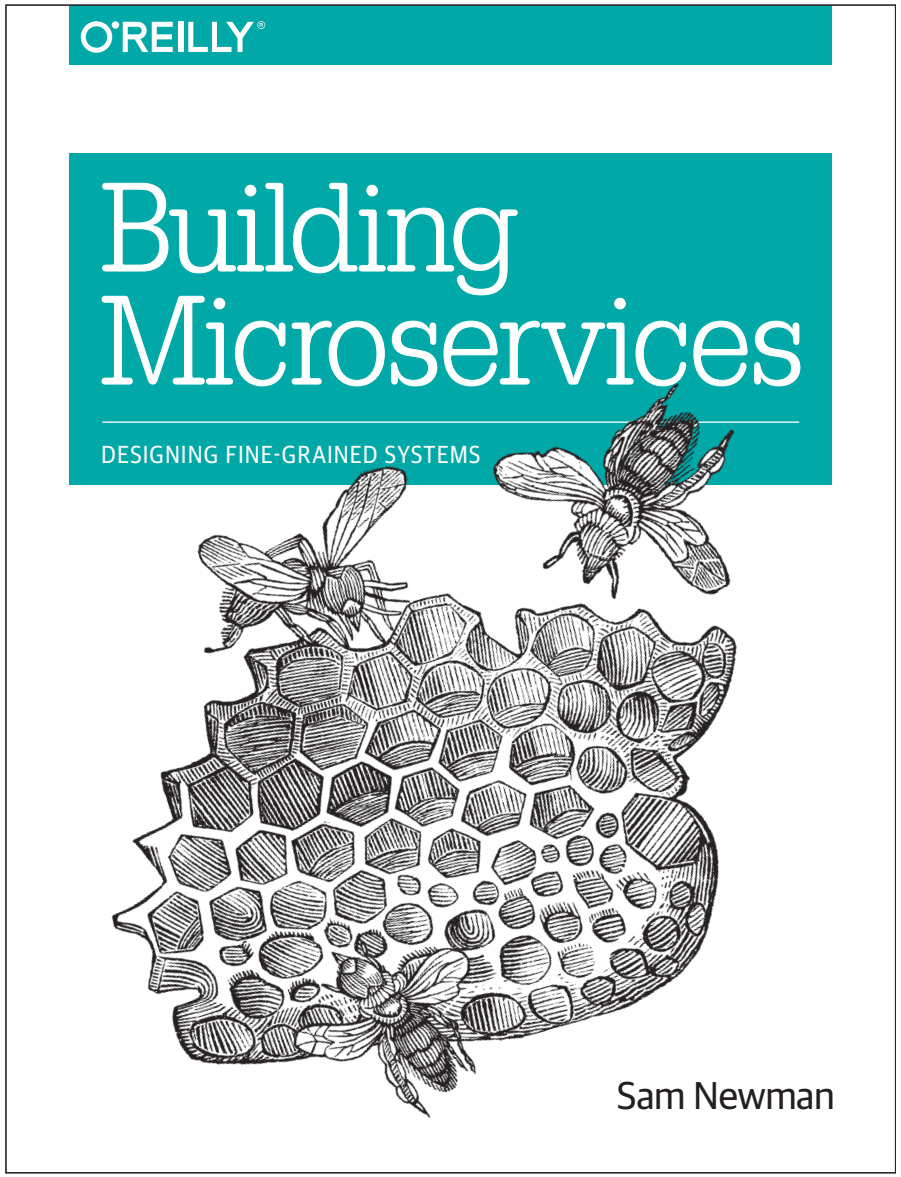
How do I get started?

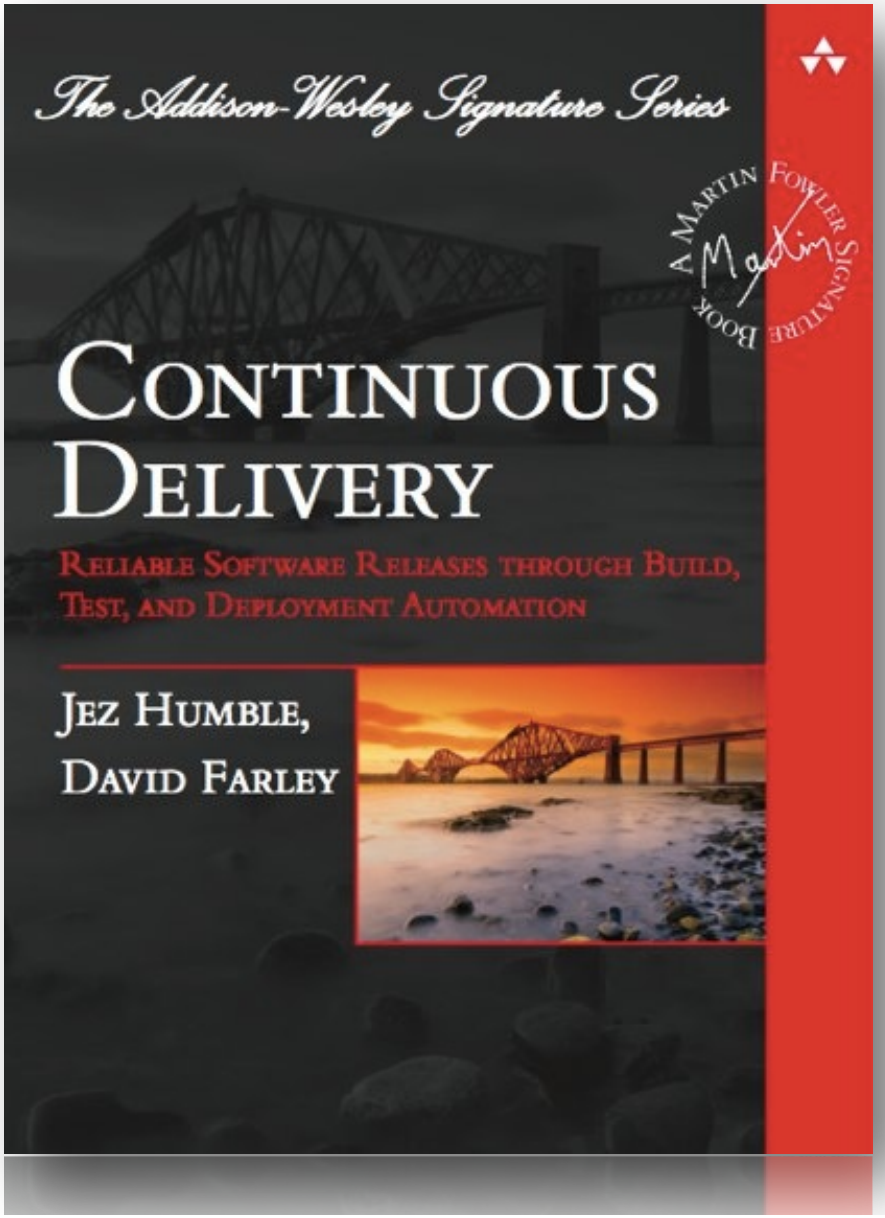
Is this a silver bullet?

Sorry, but no.

Still, used effectively in many, varied organizations.

To Learn More





Thank you!

@rebeccaparsons

http://rebeccaparsons.com

CHICAGO
INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE 2015

goto; conference

Questions?

Please remember to evaluate via the GOTO Guide App

follow us @gotochgo

Conference: May 11-12 / Workshops: 13-14