Accelerating Your DevOps Journey

Peter Eeles
Executive IT Architect
DevOps Global Tiger Team, IBM Hybrid Cloud
peter.eeles@uk.ibm.com
Agenda

1. The Business and IT Context
2. The Relevance of DevOps
3. DevOps Capabilities
4. Transforming to DevOps
5. Getting Started
Years till 50 Million Users

- Airplane: 68 years
- Automobile: 50 years
- Telephone: 62 years
- Electricity: 46 years
- Radio: 38 years
- Microwave: 31 years
- VCR: 29 years
- Credit Cards: 28 years
- Television: 22 years
- ATM: 18 years
- PC: 14 years
- Cell Phone: 12 years
- Debit Cards: 12 years
- Internet: 7 years
- PayPal Accounts: 5 years
- iPods: 4 years
- WiFi: 4 years
- Contactless Credit Cards: 4 years
- YouTube: 4 years
- Facebook: 3 years
- Twitter: 2 years
- Mobile App Banking: 2 years
- NFC-Enabled Handsets: 2 years
- E-Paper/iPad: 1 year
Agenda

1. The Business and IT Context
2. The Relevance of DevOps
3. DevOps Capabilities
4. Transforming to DevOps
5. Getting Started
WORKED FINE IN DEV

OPS PROBLEM NOW
What is DevOps?

1. Get ideas into production fast
2. Get feedback
3. Repeat 😊
The “Business-IT” and “IT-IT” gaps
How long does a small change take?

Idea

Production
DevOps is a risky business 😊

- Risk exploration period
- Risk resolution period
- Controlled risk management period
- Traditional Project Profile
- Modern Project Profile
- Risk Reduction

https://devops.com/2016/05/19/devops-risky-business/
Agenda

1. The Business and IT Context
2. The Relevance of DevOps
3. DevOps Capabilities
4. Transforming to DevOps
5. Getting Started
5. Digital Innovation Platform

2. Integration
   - API Management
   - Integration
   - Process Automation

1. Mobile Mobile Platform

3. DevOps
   - Idea
   - Shift Left Testing
   - Automated Deployment
   - Production Application Performance Monitoring

4. Hybrid Cloud
   - Traditional IT
   - Dedicated On-Prem
   - Dedicated Off-Prem
   - Shared Off-Prem

Multi-speed IT

Optimization

Industrialised Core

Innovation Edge

Interface

Innovation
A Typical Deployment Landscape

- **Development**
  - Developer
  - Build
  - Unit Test
  - Integration Test

- **Build**
  - Build Engineer
  - Integration Build
  - Deploy
  - Component Test

- **QA**
  - QA Team
  - Deploy
  - QA Test

- **SIT**
  - Integration Tester
  - Deploy
  - SIT

- **UAT**
  - User
  - Deploy
  - UAT

- **Production**
  - Operations Engineer
  - Deploy
  - Monitor
Let’s do some math …

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># Applications</td>
<td>250</td>
</tr>
<tr>
<td># Releases per year (per app)</td>
<td>28</td>
</tr>
<tr>
<td># Prod deployments per year</td>
<td>7000</td>
</tr>
<tr>
<td># Non-prod environments</td>
<td>2</td>
</tr>
<tr>
<td># Deployments per non-prod environment</td>
<td>5</td>
</tr>
<tr>
<td># Non-prod deployments per year</td>
<td>70000</td>
</tr>
<tr>
<td># Deployments per year</td>
<td>77000</td>
</tr>
</tbody>
</table>
The transformation: As it prepared to launch a critical new application, Fidelity Worldwide Investment wanted to replace its manual release processes with an automated release solution. The solution helped reduce the time required for software releases by 99 percent, from 2 - 3 days to just 1 - 2 hours. The company also achieved cost avoidance of more than USD2.3 million per year.

“Applications that took days to release now take just an hour.”
— Tony Green, Technology, Architecture and Engineering, Fidelity Worldwide Investment
Agenda

1. The Business and IT Context
2. The Relevance of DevOps
3. DevOps Capabilities
4. Transforming to DevOps
5. Getting Started
Transformation Best Practices

**IBM**

- Consider all elements of a delivery ecosystem
- Implement a center of excellence
- Plan improvements around capabilities
- Adopt capabilities incrementally
- Embrace principles of organizational change

**Kotter**

- Establish a sense of urgency
- Create the guiding coalition
- Develop a vision and strategy
- Communicate the change vision
- Empower employees for broad-based action
- Generate short-term wins
- Consolidate gains and produce more change
- Anchor new approaches in the culture
Consider all Elements of a Delivery Ecosystem

Technology focus

Method
Roles, work products, tasks, processes, standards, guidelines

Tools
Development tools & their integrations

Infrastructure
Locations, nodes & connectivity

Enablement
Training curriculum & courses

Cross-cutting Concerns
Functionality, qualities, constraints

Adoption
Adoption plan, organizational change, metrics

Organization
Organizational roles & units

People focus
Implement a Center of Excellence

Center of Excellence

Creates & Maintains

Delivery Environment

Supports

Delivery Project

Creates & Maintains

Application
The evolution of delivery practices

<table>
<thead>
<tr>
<th>Traditional</th>
<th>Iterative</th>
<th>Agile</th>
<th>Scaled Agile</th>
<th>DevOps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple Views</td>
<td>Iterative Development</td>
<td>Test-Driven Development</td>
<td>Measured Performance</td>
<td>Collaborative Development</td>
</tr>
<tr>
<td>Quality Attribute-Driven</td>
<td>Risk-Value Lifecycle</td>
<td>Continuous Integration</td>
<td>Formal Change Management</td>
<td>Continuous Testing</td>
</tr>
<tr>
<td>Development</td>
<td>Shared Vision</td>
<td>Refactoring</td>
<td>Concurrent Release</td>
<td>Continuous Release</td>
</tr>
<tr>
<td>Component-Based Development</td>
<td>Use Case-Driven Development</td>
<td>Whole Team</td>
<td>Concurrent Testing</td>
<td>Continuous Monitoring</td>
</tr>
<tr>
<td>Asset Reuse</td>
<td>Release Planning</td>
<td>User Story-Driven Development</td>
<td>Continuous Optimization and</td>
<td></td>
</tr>
<tr>
<td>Decision Capture</td>
<td></td>
<td>Whole Team</td>
<td>Optimisation</td>
<td></td>
</tr>
<tr>
<td>Architecture Proving</td>
<td></td>
<td>Refactoring</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Team Change Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Plan Improvements around Capabilities

Large-grained

Capability

Practice

Delivery Environment Component

Requirements Definition & Mgt.

Architecture & Design

Construction

- Shared vision
- Use case-driven development
- Requirements management

- Evolutionary architecture
- Evolutionary design
- Component software architecture

- Continuous integration
- Test-driven development

- Iterative development
- Two-level project planning
- Whole team approach

Method

Tools

Enablement

Organization

Infrastructure

Adoption

20
Adopt Capabilities Incrementally

Current State

Increment 1

Increment 2

Increment 3
Agenda

1. The Business and IT Context
2. The Relevance of DevOps
3. DevOps Capabilities
4. Transforming to DevOps
5. Getting Started
Summary

- All industries are seeing a changing emphasis: from optimization to innovation
- DevOps is a key enabler in helping deliver an innovation agenda
- Shift Left Testing and Automated Deployment are core DevOps Capabilities
- IBM can help with your DevOps journey