Building a DevOps Culture

2016 MTUG IT Summit and Tradeshow, June 2nd 2016
Agenda

- What is DevOps
- Why DevOps
- Developing a DevOps Mindset
- Crafting a successful DevOps Journey
- The DOs and DON’Ts of DevOps
- Common DevOps Use Cases
- Tools of the Trade
What is DevOps
DevOps is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes.

- Loukides, Mike (2012-06-07). "What is DevOps?"
What is DevOps

- Understanding of the interdependencies between application development and IT Operations
- Aims to help organizations rapidly produce quality applications, products and services
- Responds to the dynamic business needs and demands of the stakeholders
- Supports the use of agile development processes
Characteristics of DevOps

- Same team develops and operates the solution.
- Focus on business goals
- Pragmatic automation
- High value placed on learning by experimentation
- Rapid innovation cycles
What is DevOps Not

- A role, person or organization
- A set of task performed by system administrators or developers
- Writing Chef or Puppet Scripts
- Application or set of automation tools
DevOps Lifecycle

- Cross Functional DevOps Team
- Application Delivery Pipeline (SDLC)
- Continuous Delivery Tool Chains
- Re-engineer
- Optimize Processes
- Measure
Pillars of DevOps

DEVOPS

Business Enabling Responsiveness
Reduced Lead Time for Changes

PRACTICES

Monitor Everything
Continuous Delivery
Automated Infrastructure
Continuous Integration
Automated Testing
Version Control Everything

CULTURE

High Trust
Innovative
Performance Oriented
Empowered Associates
Reduce Variation
High Cooperation

Continuous Flow and Visibility

Lean & Agile Principles
System Flow

Product Centric
Amplify Feedback
Continuous Experimentation
Why DevOps
Why DevOps

Application development and IT operations teams travel different roads with different points of view.

- App development team:
  - Deliver quickly
  - Add new functionality

- Operations team:
  - Avoid outages
  - Contain issues

Caution

These separate roads slow the delivery of new apps and features to customers.

Source: HPE, DevOps: Accelerate to the speed of business
Why DevOps
Hey Ops, you wanted a chat?

Yes - I'd like to congratulate you on a fantastic job!

I released your latest build yesterday to prod and it's been running like Ben Johnson on drugs.

Are you feeling OK?

I'm speechless. Why all the love?

I was home by 6pm yesterday which hasn't happened since 2005!
## The 4 Step Process

<table>
<thead>
<tr>
<th>Assess your DevOps Strategy</th>
<th>Identify the DevOps maturity of your core dev and IT ops processes</th>
<th>Adopt and implement change to improve velocity</th>
<th>Measure progress and plan next improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Identify your DevOps business drivers and challenges</td>
<td>• Identify key stakeholders in business, development, QA, and operations</td>
<td>Pilot and roll out changes to accelerate delivery for:</td>
<td>Based on results and business metrics, update and adjust plans.</td>
</tr>
<tr>
<td>• Examine critical success factors and best practices</td>
<td>• Determine communication plan for stakeholders</td>
<td>• Development and build automation</td>
<td>• Identify results of improvements</td>
</tr>
<tr>
<td>• Review critical DevOps areas that align with your business goals</td>
<td>• Understand the app development methodologies—Agile, waterfall, both—that are most important</td>
<td>• Testing</td>
<td>• Locate next opportunity to improve</td>
</tr>
<tr>
<td>• Identify priorities for short, midsize, and long-term projects</td>
<td>• Determine the IT processes that are most important</td>
<td>• Installation and deployment</td>
<td>• Plan next phase</td>
</tr>
</tbody>
</table>
Crafting a DevOps Journey
Getting Started

- Start with a Low Risk Application
- Identify Desired Business Outcome(s)
- Current State Gap Analysis
- Secure Executive Support
- Properly set expectations
- Focus on Small Iterations of Improvement
- Ruthlessly Remediate Inefficiencies
DevOps Evolution

CRAWL

• Invest in engineering thought leadership
• Focus on shipping product
• Microservices & Cloud strategy definition

WALK

- Invest in Lean mindset and practices
- DevOps adoption
- Microservices & Cloud implementation
- On-demand releases
- Establish baseline metrics

Defined path forward

Begin unlocking productivity and speed to market

RUN

• Spread talent across organization
• Manage to metrics
• Optimize and extend

Optimized. Scalable site that can innovate at the speed of business
Application Modernization Journey

DevOps

Less Mature

- Legacy Application
- Business Objectives
- Gap Analysis
- Infrastructure As Code
- Application Architecture

More Mature

- Cloud-Native

Less Mature:
- Pragmatic DevOps
- Understand the Business Objectives
- Use 12 Factors to Assess Current State

DevOps:
- Continuous Improvement
- Adopt Infrastructure as Code
- Reprocess, Refactor, Replatform, Rebuild

More Mature:
Dos and Don'ts of DevOps
The DOs

- Mentor and train the organization to embrace the DevOps mindset
- Select DevOps tools and patterns that will ease the evolution of your application development platform
- Include automated testing in your DevOps playbook
- Instill pride of ownership in your teams
- Encourage innovation and experimentation
- Embrace effective feedback, metrics and monitoring
The DON’Ts

- Take **SECURITY** for granted
- Forget about service and resource governance
- Promote a culture of secrecy, judgement and fear of retribution
- Do something manual more than twice
- Force every application, from the start, through the DevOps process
- Re-invent the wheel
- Create bottlenecks that lead to a single point of failure in your teams
Use Cases: Application Modernization
What is Application Modernization?

Application modernization is the refactoring, re-purposing or consolidation of legacy applications to align more closely with current business needs.
What Are Your Application Modernization Goals?

 ✓ Improve Reliability - Antifragility
 ✓ Reduce Costs
 ✓ Alignment of IT and Business Goals
 ✓ Easier Compliance Audits
 ✓ I’m Just Generally Masochistic
 ✓ Improve Speed/Agility
 ✓ Improve Performance
 ✓ Improve UI/Features/Functionality
 ✓ Reduce Learning Curve
Use Case: Infrastructure as Code
Infrastructure Maturity Model

**Pre Virtualization**
- Low Hardware Utilization Rates
- Limited Virtualization Strategy
- Services Tightly Coupled to Hardware
- Traditional Provisioning Process

**Server Virtualization**
- Consolidated Physical to Virtual Servers
- Non Mission Critical Applications Virtualized
- Infrastructure Applications Virtualized
- Basic Management and Monitoring

**Enterprise Virtualization**
- Mission Critical Applications Virtualized
- Virtual Infrastructure Provides HA
- Virtualized Security Services
- Virtual Infrastructure and Storage Integrated
- Advanced Management and Monitoring
- Limited Cloud Strategy

**Cloud**
- Monitor Everything
- Service Catalog
- Self-Service Portal
- Automated Change
- XAAS
- Infrastructure As Code

Less Mature → More Mature
Programmable Infrastructure

**Infrastructure as Code** - The concept of applying methods and tooling established in software development onto the management of IT infrastructure. This includes but is not limited to:

- Automation
- Versioning
- APIs
- Immutability
- Agile Techniques
## Lead Time Comparison

<table>
<thead>
<tr>
<th>Action</th>
<th>Traditional Infrastructure</th>
<th>Infrastructure as Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Provision a Server</td>
<td>Days or Weeks</td>
<td>Minutes</td>
</tr>
<tr>
<td>Build a New Environment</td>
<td>Weeks or Months</td>
<td>Minutes</td>
</tr>
<tr>
<td>Reconfigure a Existing Environment</td>
<td>Days or Weeks</td>
<td>Minutes</td>
</tr>
</tbody>
</table>
Tools of the Trade
We may not have all the answers but we can help you figure them out.
And Finally…to Sum it up

- IT Performance improves organizational performance. DevOps practices lead to better IT and organizational performance.
- High-performing IT organizations deploy 30x more frequently with 200x shorter lead times.
- Lean management provides higher quality, shorter cycle times with quicker feedback loops, and lower costs.
- Painful deployments generally typically indicate poor IT performance, organization performance, and culture.
Thank You