DevOps Without Dev is Dead on Arrival

Ashish Kuthiala
Director, Marketing
Electric Cloud
Software Apps

Engines of Every Business

Web apps and app services have become critical to business success
Today’s Software Delivery Process
Complexity Everywhere!

Dev
Software Dev

Build

Dev
Build

QA
Test

Release
Release

Ops
Deploy & Operate

Complex Apps

Web Server
App Server
Database

Complex Process

Complex Environments

Dev
QA
Production

Many/Complex Tools

Tools:
- git
- Visual Studio
- COLLABNET
- HRA
- Bugzilla
- gerrit
- COREYNT
- PERFORCE
- amzn
- VMware
- android
- PERFORM
- Rational
- Clover
- Klocwork
- SCons
- SONS
- Parasoft
Agile Delivers Applications Faster

But increases delivery process challenges

Today’s process: Manual/script based
Slow, siloed, error-prone, limited visibility
## Cost of not solving these problems

<table>
<thead>
<tr>
<th>Issue</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop to Deploy</td>
<td>90 Days</td>
</tr>
<tr>
<td>Build to Release/Deploy</td>
<td>10+ errors/cycle</td>
</tr>
<tr>
<td>Audit application changes (who, what, how, why, when)</td>
<td>Days</td>
</tr>
<tr>
<td>Time to troubleshoot problems</td>
<td>20 Days</td>
</tr>
<tr>
<td>Improved infrastructure utilization</td>
<td>1000’s of dedicated resources</td>
</tr>
</tbody>
</table>
What is DevOps

Tight Collaboration and merging of principles, guidelines, best practices

Dev/QA
- Software Development
- Build, Compile, and Integrate
- Quality Assurance
- Software Release
- Support deployment and production tasks

Ops
- Provisioning and Configuration
- System Administration
- IT infrastructure health and monitoring
- Change and Release management
- Other Infra Support

Dev + Ops = DevOps
DevOps – Broad Goals

- Reduce Software Delivery Times
- Improve Application Quality
- Reduce Application:
  - Complexity
  - Errors
  - Risk of application deployment failures
- Enhance productivity of Dev and Ops teams
Why is there a divide between Dev and Ops?

Cultural Differences

The Technology Divide
Cultural Differences regarding change

Stereotypes fuel conflict

Dev

Ops
The Technology Divide

Process and Tools

Agile = Apps Faster

Application Deployments

IaaS/Cloud = Infra Faster

Business Demand

Business

OPS
Why is the migration to DevOps happening now?

Business Dynamics

Technology Dynamics
Extraordinary pressures to deliver more, faster

Businesses now demand answers to challenging questions such as:

- What changes went into this application?
- Who approved these alterations?
- What packages comprise this application?
Technology Dynamics

Dev investing in best practices and tools

- Holistic and coherent enterprise-wide processes
- End-to-end visibility
- Capturing and disseminating best practices
- Closed-loop analysis and reporting
- Consistent security policies

Ops leveraging virtualization and cloud computing

- Eliminating physical infrastructure bottlenecks
- Enable resources on demand
- Consistency
Value of Dev in DevOps
Reasons to focus on Dev

- Comprehensive understanding of the application
- Experience with the application deployment process
Dev’s comprehensive understanding of the application
General Structure of an Application

Online Banking Application

- Web
  - Apache Configs
  - Banking Files

- App Srv
  - Weblogic Configs
  - Banking WAR1
  - Banking WAR2

- DB
  - DB Changes
  - DB Baseline

Artifacts
(aka Components)
- WAR, JAR, EAR
- DLL’s
- Build output
- File sets
- Scripts
- Configurations
- Graphics
- Etc.
Complexity From Many Moving Parts

Development Processes

Online Banking Application

Web
- Apache Configs
- Banking Files

App Srv
- Weblogic Configs
- Banking WAR1
- Banking WAR2

DB
- DB Changes
- DB Baseline

SCM
- Create
- Package
- DevTest
- Publish

SCM
- Build
- DevTest
- Publish
Dev’s experience with the application deployment process
Many Steps of Deployment

Complex Apps

Online Banking Application
- Web
  - Apache Configs
  - Banking Files
- App Sv
  - Weblogic Configs
  - Banking WAR1
  - Banking WAR2
- DB
  - DB Changes
  - DB Baseline

Many Environments

System Integration
- Web
- App
- DB

UAT
- Web
- App
- DB

Prod
- Web
- App
- DB

Many Steps

- Pull Artifacts
- Shutdown AppServer
- Copy .WAR to AppServer
- Register .WAR AppServer
- Copy SQL scripts to database server
- Run SQL scripts on database
- Start application server
- Run validation scripts
Many deployments per release cycle

Many more deploys in Dev/QA than production
Phased Approach
A Phased approach to achieving DevOps

Phase 1: Share the goals

Phase 2: Share the requirements

Phase 3: Share the process
Phase 1: Share the goals

- Common goal – Successful Application Releases
- Ops:
  - Understand and Reconcile with Dev’s Goals
  - Stop being a roadblock to Rapid Application Releases
- Dev:
  - Understand business goals that drives Ops
  - Share and alleviate Ops’ challenges

Shared Pager Duty
Phase 2: Share the requirements

- Complete visibility into each others’ requirements
- Dev:
  - Knows and Uses Ops requirements and data feeds
- Ops:
  - Complete visibility into what’s coming from Dev
  - How to release applications starting from within Dev environments

Transparent Deployments
Phase 3: Share the process

- Share the processes
- Dev:
  - Share understanding of applications, deployment processes
- Ops:
  - Leverage Dev’s knowledge

Joint End-to-End management of all application releases
Conclusion
How to Make DevOps Work

Summary

- Take a Dev-centric approach
- Establish shared goals
- Phased Approach
- Change incentives
- Measure

Iteration Makes Perfect
EC Software Delivery Solution

Software Delivery Acceleration

Build Automation & Acceleration

Test Automation & Acceleration

Deploy Automation

Software Delivery Platform

Dev/Test/System Tools
App Infrastructure

Physical/Virtual/Cloud

Orchestrate, Accelerate, Visibility
## Cost of not solving these problems

<table>
<thead>
<tr>
<th>Issue</th>
<th>Before Electric Cloud</th>
<th>With Electric Cloud</th>
<th>Business Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop to Deploy</td>
<td>90 Days</td>
<td>10 Minutes</td>
<td>99.93%</td>
</tr>
<tr>
<td>Build to Release/Deploy</td>
<td>10+ errors/cycle</td>
<td>~0 errors/cycle</td>
<td>90+%</td>
</tr>
<tr>
<td>Audit application changes (who, what, how, why, when)</td>
<td>Days</td>
<td>Minutes</td>
<td>90%</td>
</tr>
<tr>
<td>Time to troubleshoot problems</td>
<td>20 Days</td>
<td>minutes</td>
<td>90+%</td>
</tr>
<tr>
<td>Improved infrastructure utilization</td>
<td>1000’s of dedicated resources</td>
<td>400+ UCS blades</td>
<td>Massive savings on capital, power</td>
</tr>
</tbody>
</table>
Electric Cloud Customers

200+ in over 250 locations

- **Fin. Services**
  - NYSE
  - Morgan Stanley
  - E*TRADE
  - Nomura
  - Trading Technologies

- **Mobile**
  - RIM
  - Motorola
  - Nokia
  - ZTE
  - Samsung
  - MediaTek

- **ISV**
  - HP
  - VMware
  - Symantec
  - Oracle
  - IBM

- **Internet**
  - Facebook
  - Intuit
  - FamilySearch
  - eCollege
  - Gap

- **Gaming**
  - EA
  - Bioware Corp
  - Bioware
  - Gaikai

- **Service Providers**
  - Equinix
  - CenturyLink
  - CenturyLink
  - DIRECTV

- **Networking**
  - Cisco
  - Huawei
  - Ericsson
  - Juniper Networks

- **Semiconductor**
  - Texas Instruments
  - Qualcomm
  - Broadcom
  - Intel

- **Embedded**
  - Philips
  - Roche
  - Cerner

- **Defense**
  - Lockheed Martin
  - Boeing
  - Thales
  - Mercury Computer Systems
  - General Dynamics
Backup Slides
**ElectricAccelerator**

Parallelize & distribute build tasks

- Dependency-aware
- Workload-aware
- Infrastructure-aware

Dramatically accelerated & accurate software builds

Work with gmake, nmake, visualStudio, SCONs, MSBuild

Cluster/Cloud

Multicore Servers

Multi-core desktops
ElectricCommander

Automate & Accelerate entire process
ElectricDeploy

Application Deployment Automation

• Model driven for consistent deployments
  • Application model, Environment model, workflow model

• Refinement through managed iteration
  • Code-safe, run-safe, retry-safe

• Comprehensive visibility
  • Deployment pipeline, deployment troubleshooting