DevOps Practices that Deliver Transformation and Cost Reduction

Stephen Elliot
Research Vice President, DevOps
IDC
Defining DevOps

“A methodology and a set of enabling technologies that unify a highly collaborative team consisting of business leadership, design, development, testing, process and portfolio management, deployment, and operations to be responsible for the creation and delivery of business capabilities.”
Amazon Deployment Stats: The Bleeding Edge

1,079         Max deploys in a single hour
10,000        Mean # of hosts receiving deploys simultaneously
30,000        Max # of hosts receiving deploys simultaneously

(production and host environments only)
DevOps Practice Traits

- Transformational change agent
- Deep business acumen
- Cultural and process awareness
- Collaboration
- Continuous delivery
- Continuous integration
- Feedback mechanisms
- Automated testing
- Politically savvy
- Organizational development knowledge
- Sustainability focus
- Automation
- Collaboration
- Configuration and change
- Performance engineering
- Application performance and analytics
DevOps Maturity Model

**People**

- Cultural Identity
  - IT and business culture that limits risk taking, learning, and experimentation, exhibiting a lack of accountability in a highly political and bureaucratic structure.
  - Business Outcome: “No risk taking leads to slow growth”
  - Weak executive management is enabled through politics.

**Culture**

- Selective Collaboration
  - Pockets of standardized processes and technologies driving increased teamwork, collaboration, and risk taking. Some teams are rewarded through the achievement of shared metrics and objectives.
  - Business Outcome: “Standardization leads to cost reductions”
  - Process and technology rationalization simplify development and operational processes.

**Technology**

- Consistent Standards
  - Pervasive automation across development and operations teams enable standardized change management policies and processes across development, test, and change deployment, and monitoring teams. Skills and career training increase.
  - Business Outcome: “Service Delivery team productivity increases”
  - Performance metrics are used to measure and communicate progress.

**Business**

- Continuous Expansion
  - Highly collaborative teams across development, testing, and deployment, empowered through a "blameless" culture, and continuous learning. Customer feedback used throughout the SDLC, with a focus on Minimal Viable Products.
  - Business Outcome: “Customer focused innovation delivered continuously”
  - Teams use customer input to deliver faster business outcomes, while expanding speed and quality improvements.

**Process**

- Optimized
  - Cultural transformation driven by customers, with fast and automated product deployments integrated with quality, security, audit, and compliance teams. Experimentation and failure, if expected, outcomes are analyzed with extensive application driven analytics.
  - Business Outcome: “Sustainable competitive advantage”
  - Business advantages scale to deliver measurable and transparent business outcomes.

Five Dimensions
- People
- Culture
- Technology
- Business
- Process
Defining DevOps Business Outcomes

Speed and Throughput

- Sprint counts, work in progress, project done count, features in progress, commits per day, builds per day, average build duration

Quality

- Monitoring status, static analysis, unit and functional test results, security analysis, code error rates, code change customer impact timeframe

Costs

- Cost per unit of work, number of incidents, MTTR, MTTF, customer experience monitoring, application availability and performance
Use Metrics to Communicate Value and Judge Progress

<table>
<thead>
<tr>
<th>Technology Leadership and their teams</th>
<th>Business Executives and their teams</th>
<th>Perceived value</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Deployment frequency</td>
<td>• Revenue &amp; Profit</td>
<td>• Productivity</td>
</tr>
<tr>
<td>• Lead time for changes</td>
<td>• Customer feedback</td>
<td>• Quality</td>
</tr>
<tr>
<td>• Change error rates</td>
<td>• Cash flow</td>
<td>• Costs (avoidance and/or reduction:</td>
</tr>
<tr>
<td>• Failure rates</td>
<td>• Time to market</td>
<td>– Operating expense</td>
</tr>
<tr>
<td>• Availability</td>
<td>• ROI &amp; NPV</td>
<td>Capital expense</td>
</tr>
<tr>
<td>• Recovery time</td>
<td>• Customer satisfaction</td>
<td>• Speed (throughput)</td>
</tr>
<tr>
<td>• Job satisfaction</td>
<td>• Cost per service/unit</td>
<td></td>
</tr>
<tr>
<td>• Defect density</td>
<td>• Internal Rate of Return (IRR)</td>
<td></td>
</tr>
<tr>
<td>• MMTR and MTTF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Technology
- Business
- Value

Perceived value includes:
- Productivity
- Quality
- Costs (avoidance and/or reduction:
  - Operating expense
  - Capital expense
- Speed (throughput)
Organizing and Incentivizing Teams

<table>
<thead>
<tr>
<th>Business Objectives</th>
<th>Speed/Quality/Cost Reduction/Time to Market/Customer Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical and Business Metrics</td>
<td>DevOps Adoption, Productivity, MTTR, MTTF, Availability, Error Reduction, Deployment Frequency, IRR, Churn Rate Reduction</td>
</tr>
</tbody>
</table>
Creating a DevOps Center of Enablement (CoE)

There are significant acceleration advantages for IT leaders that decide to create a DevOps team or Center of Excellence versus a less-organized DevOps organizational approach.

- Increased automation across silos
- Empathy increases across domain expertise owners, driving more trust
- IT project success increases, and there is acceleration in the speed of success
- Shadow or stealth IT decreases as business stakeholders are included early
- IT's job satisfaction and career development improvements
Gauge Project Readiness for DevOps Adoption

- **Culture and leadership**
  - What are your IT and business cultural traits?
  - Who will lead DevOps, and what’s their motivation?
  - How will you manage cross functional change?

- **Metrics and Measurement**
  - Have metrics been defined, and are they measurable?
  - Have HR performance metrics adjusted for DevOps?
  - Are there dashboards?

- **Technology and automation**
  - Where is the most pain?
  - What’s the automation and integration strategy?
  - Which processes will be impacted?
  - What’s your DevOps architecture?

- **Collaboration and teamwork**
  - Is teamwork and collaboration incented?
  - Which project will deliver the biggest ROI?
  - How will IT silos be deconstructed?

- **Collaboration and teamwork**
  - What are your IT and business cultural traits?
  - Who will lead DevOps, and what’s their motivation?
  - How will you manage cross functional change?
Determine DevOps Risks

<table>
<thead>
<tr>
<th>DevOps Reality</th>
<th>Business and executive case: Does the organization believe that the status quo is unacceptable?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Compelling intent: Is the philosophy of DevOps compelling enough to garner support from the organization?</td>
</tr>
<tr>
<td></td>
<td>Credible solution: Are the proposed new principles and operating model appealing to the organization, and will they work in the culture?</td>
</tr>
<tr>
<td>Inspiration and commitment</td>
<td>Create top team/CoE: Have the right team members and leadership been identified?</td>
</tr>
<tr>
<td></td>
<td>Committed executive sponsorship: Do we have the right leaders at all levels actively engaged, and communicating and supporting DevOps driven changes?</td>
</tr>
<tr>
<td></td>
<td>Influential supporters: Are the most credible and trusted opinion leaders involved and supporting DevOps?</td>
</tr>
<tr>
<td>Team and staff success</td>
<td>Personal commitment: How will disruption and political resistance be managed? How will commitment be communicated?</td>
</tr>
<tr>
<td></td>
<td>Critical skills: Are plans in place to retain, develop, train, or acquire DevOps talent?</td>
</tr>
<tr>
<td></td>
<td>Desired behaviors/attitudes: What are the critical behaviors that drive results? Can HR change metrics and compensation to reinforce behaviors?</td>
</tr>
<tr>
<td>Outcome Delivery</td>
<td>Achievable plan: What projects should use DevOps practices, while not negatively impacting the business?</td>
</tr>
<tr>
<td></td>
<td>Governance plan: Does DevOps have program governance to manage change and deliver key decisions?</td>
</tr>
<tr>
<td></td>
<td>Leading indicators: Do teams have the right information and technology and business metrics to track results, identify risks, and adjust course before it’s too late? Is failure accepted?</td>
</tr>
<tr>
<td>Sustainability</td>
<td>Effective organization: Have teams addressed the organizational obstacles that might hinder DevOps changes?</td>
</tr>
<tr>
<td></td>
<td>Enabling technologies: Are the right technologies in place to deliver the results on time? Has automation become a key across the organization?</td>
</tr>
<tr>
<td></td>
<td>Continuous improvement: Have we designed fast feedback loops (Security, etc.) to learn and improve DevOps over time?</td>
</tr>
</tbody>
</table>
Reducing Operational Costs with DevOps Practices

Traditional I&O Organization
- Keep the “lights on mentality”
- Reactive silo
- Technology first mentality, with finger pointing
- Focused on system availability and cost-centric metrics

Transformational I&O Organization
- Partner with dev. and security teams
- Trusted partner advising delivery teams
- Match DevOps with ITIL
- Focused on performance tuning for services, and speed and quality metrics
Reducing Application Development Costs: Using Containers

Technology Description
A new way to package applications that logically collect code and dependencies using fewer resources than virtual machines.

Adoption
Less than 7% of large enterprise IT organizations are using container technology today; however, IDC expects this rate to triple in 2017.

Benefits
Faster time to market and more agility for customers requesting new application functionality.

Risks
Expect significant evolution in the container ecosystem; the container format is stable, but orchestration solutions are vying for market share.

Critical Success Factor
Process integration, automation, and agile development are expected benefits.

Investment
Executives should allocate at least 1-2% of their application development budget toward container experimentation.

Source: IDC #US42252817, January 2017
## Vendor Challenges Increasing IT Project Failure Rate Through Poor Alignment

<table>
<thead>
<tr>
<th>Vendor Decision</th>
<th>Customer Questions</th>
<th>IT's Risk Exposure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product commitment and budget reductions</td>
<td>Identify product lines. Are product management and engineering teams impacted?</td>
<td>Disrupts product lines and impacts features, capabilities, and road maps</td>
</tr>
<tr>
<td>Staff reductions</td>
<td>Which teams are impacted? Why? Where is head count being allocated?</td>
<td>Shows lack of commitment across impacted product lines</td>
</tr>
<tr>
<td>Executive and senior management changes</td>
<td>Why were the changes made? What background does the CEO bring?</td>
<td>Management can kill momentum in prior regime strategic product areas</td>
</tr>
<tr>
<td>Financial performance and Wall Street pressures</td>
<td>What are the financial metrics management is judged on?</td>
<td>Metrics drive behavior and actions; financial reengineering limits budgets</td>
</tr>
<tr>
<td>Corporate strategy misalignment</td>
<td>How will strategy changes impact budget allocations and acquisitions?</td>
<td>Corporation/business unit misalignment kills innovation and slows execution</td>
</tr>
<tr>
<td>Internal reorganizations</td>
<td>How often do internal reorganizations occur? What is the new reporting structure?</td>
<td>Slows product development; creates political instability and turnover</td>
</tr>
<tr>
<td>Mergers and acquisitions or divestitures</td>
<td>What’s driving the M&amp;A activity? Is the acquirer committed to this business?</td>
<td>Product discontinued, support reduced, and no future investment leading to technology obsolescence</td>
</tr>
</tbody>
</table>

Source: IDC #US42252817, January 2017
## Creating an Automation Strategy

<table>
<thead>
<tr>
<th>Top down</th>
<th>Bottom up</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strategic planning and executive leadership:</strong> Overarching IT automation strategy creation and ownership</td>
<td><strong>Technology integrations</strong></td>
</tr>
<tr>
<td><strong>Business capabilities:</strong> Assess, create, and prioritize business requirements, metrics, and value</td>
<td><strong>Process and policy definitions and integrations</strong></td>
</tr>
<tr>
<td><strong>Value assessment:</strong> Evaluate short-term and long-term automation value, expectations, and road maps</td>
<td><strong>Impact assessment</strong></td>
</tr>
<tr>
<td><strong>Governance:</strong> Include security, audit, and compliance for business risk analysis</td>
<td><strong>Organizational change</strong></td>
</tr>
<tr>
<td><strong>Tool inventory assessment:</strong> Document the existing IT Portfolio; determine migrations</td>
<td><strong>Integration and orchestration road maps:</strong> Technology, process, and staffing for physical, virtual, datacenter, and cloud environments</td>
</tr>
<tr>
<td><strong>Automation analysis:</strong> Define processes; align ownership and accountability</td>
<td><strong>Organizational alignment:</strong> Identify existing and new skills for automation; align organizational structure and consider a DevOps COE</td>
</tr>
</tbody>
</table>
Continuous Delivery: Automation Use Case

CD requires taking a broad focus on creating a pattern and model for getting software from development to release through the use of automation and a consistent, repeatable deployment pipeline.

- Transparency across engineering, product management, build, release, and operations teams as well as the deployment pipeline is critical to success and collaboration.

- Executives should make the transition to CD, starting with a small project and measuring its success with business and technology metrics.

- While highly technical, CD discussions must have a business benefit and use business metrics that enable IT executives to communicate its value with business executives.

CD is associated with the process of software delivery, while DevOps takes a strong focus on the cultural IT changes; there is interlocking support between the two practices.
Continuous Delivery: What to Automate?

1. Automate the workflow to approve deployments and environment requests

2. Automate the unit, functional, and regression testing

3. Automate the build process when source code changes are promoted

4. Implement artifact repository with dependency management

5. Automate the code quality metrics as deployment gates

6. Automate the deploy process to release code to environments

Standardization is the fastest way to increase ROI
Increasing Application Security with DevOps

**Define what "security" means.** There are many layers of security that can be included in DevOps projects. Examples include application security, information security, infrastructure security, and mobile security. Make sure the team has a clear definition of what is possible and what the objectives and pre-requirements are on the security front.

1. **Communicate** with the audit, security, and compliance teams in their language, or make an attempt to understand what they need and how it needs to be delivered. Be transparent.

2. Make DevOps part of an [enterprise resiliency strategy](#).

3. **Organize the separation of duties**, a major, unwritten audit challenge; DevOps teams can help actively organize this perspective.

4. **Consider different types of controls**, as auditors and compliance teams often want to achieve similar objectives, regardless of the processes (manual or automated) and regardless of the tools.

5. **Show process transparency** and how various process pieces fit together, integrate with various partners and providers (external and internal), and deliver consistent outcomes — one of the major challenges for security and audit teams is the common lack of understanding of the different pieces of an automated process.
Consider the notion of **continuous governance** whereby DevOps team is focused on the quality of software code from suppliers. Teams can use test automation on applications for compliance requirements and create a theme of "compliance-driven development."

Create **continuous monitoring of production** for vulnerabilities in areas such as ongoing authorization, feedback from production to development on security, and vulnerability topics. They can also conduct static code analysis and have development teams run this against their code. Push compliance to the start of the development cycle.

Recognize app security is often **both dynamic and static** via various scanning technologies. Conduct bug readouts and bring input back to engineering via feedback loops. The goal should be to be measured on successful remediation of security bugs. Shift security to the development processes.

Apply security for code vulnerability scans, access, and improved availability; identify what could be **embedded** into the DevOps process.
Invest in Agile Coaching for DevOps

Agile Coaching
- Organizational (Guided on-boarding and transition)
- Clear roles and responsibilities
- Sustainer programs

Backlog Management
- Release/Roadmap planning
- Portfolio and program planning

Execution
- Leveraging technical practices
- LEAN documentation
- Feedback loops: Dashboards, reporting, metrics
9 Critical Pitfalls to Avoid

1. Lack of management commitment
2. Weak team leadership
3. Allowing departmental chaos
4. Failure is not an option mentality
5. Poor project management and work planning
6. Lack of demonstrating results with data
7. Not reviewing the entire service lifecycle
8. Sidestepping political inhibitors
9. Poor communications
Summary

• DevOps enables organizations to take advantage of team-based collaboration and reinvent integrated service delivery across development, operations, and security teams.

• Automation and shared tooling are essential ingredients for successful DevOps and cost reduction.

• DevOps practices provide value to both greenfield and existing (brownfield) application development projects as well as front-end application development and system-of-record, mission-critical application development.

• DevOps can deliver cost reduction through different practices and activities.