The ultimate goal of DevOps is to unify development and operations end to end, but many organizations struggle to realize the full adoption journey from a single application to the enterprise level. Tactics and challenges vary at every stage; thus even the most promising efforts fail to scale products and services through the entire scope of adoption. A comprehensive strategy is critical to delivering sustainable business value through DevOps.

Most Accenture clients who find success with DevOps adoption use an approach spanning four key layers of the organization structure, each with its own set of measurable practices that must be synced and streamlined together.

- **Optimize** Individual Applications and Projects
- **Organize** Application Clusters and Programs
- **Empower** Lines of Business (LoB) and IT Divisions
- **Align** the Organization

**A DevOps journey is an organization-wide journey across all layers. Even if your scope of DevOps adoption is within one single layer, you need to sync it with other layers.**
This guide aims to help owners and stakeholders of DevOps adoption in each layer:

1. **Understand each layer** and its corresponding goals, challenges and recommendations.
2. **Decide where to start** and which recommendations to implement.
3. **Plan and scale** adoption across all layers.

<table>
<thead>
<tr>
<th>LAYER</th>
<th>AUDIENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimize Applications/Projects</td>
<td>Developers, Testers, Operators, Architects, Infrastructure Team</td>
</tr>
<tr>
<td>Organize Application Clusters/Programs</td>
<td>Program Management and Release Management Teams</td>
</tr>
<tr>
<td>Empower LoBs/IT Divisions</td>
<td>LoB/IT Division Leadership, Strategy and Change Management Teams</td>
</tr>
<tr>
<td>Align The Organization</td>
<td>Top-Level Leadership, Strategy and Enterprise Architecture Team</td>
</tr>
</tbody>
</table>

**Definitions**

When reading this document: "Application" refers to both applications and projects. "Cluster" refers to both clusters and programs. "LoB" refers to both LoBs and IT Divisions.
Individual applications are the basis consideration in every DevOps strategy. All other layers are entirely dependent on solid DevOps implementation at this level; thus, stakeholders should keep the broader adoption plan in mind when deciding which tools and practices to implement here.

A core tenant of DevOps is identifying dependencies among related applications and grouping them by release time and strategy. These groupings are known as “clusters”. This allows for the harmonious implementation of DevOps to all applications within a cluster, thereby promoting consistent release speed and quality.

Organizations often overlook this critical step, resulting in negative business impact. If DevOps practices are not applied to every application within a given cluster, the sluggish applications will delay release of the entire cluster. This could result in higher time-to-market for business functions served by those applications.
EMPOWER LOBS

Most large organizations are structured by LoBs. (For instance, Core Banking, Private Banking, Corporate Banking, Internal Functions, and Data and Warehouse may all be considered LoBs at a financial institution.) Each LoB is serviced by a number of application clusters.

LoBs often act as sub-organizations—it’s not uncommon for each one to have its own IT team, with independent objectives, strategies, roadmaps, infrastructures, enterprise architectures, governance, tools, processes, etc. Applying DevOps uniformly among LoBs is as challenging as it is crucial to the success of the overall adoption.

ALIGN THE ORGANIZATION

In today’s fast-changing ecosystem, IT helps drive business forward. Successful organizations need cutting-edge IT supported by an organizational culture that promotes innovation and agility. This is where LoBs align with enterprise strategy and governance to build a culture that fosters DevOps adoption at every level.

Let’s break it down

Aside from sharing a few common elements, each layer in DevOps adoption has its own goals and practices.

GOAL

Embrace DevOps at the highest organizational levels to drive business and stay competitive.

Adopt DevOps culture across subdivisions to improve quality, speed and cost-effectiveness of software development.

Improve combined maturity and time-to-market of dependent applications.

Improve individual application quality and time-to-market.
EMPOWER LINE OF BUSINESS AND IT DIVISIONS
ALIGN ORGANIZATION
ORGANIZE APPLICATION CLUSTERS AND PROGRAMS
OPTIMIZE INDIVIDUAL APPLICATIONS AND PROJECTS

GOAL
PRACTICES
• Consolidate LoB governance
• Align DevOps with enterprise strategy and architecture

COMMON CONSIDERATIONS
• DevOps Transformation Alignment with Other Transformations e.g. Agile, Application Modernization

• Governance (enabling standardization)
• Organizational structure
• DevOps-as-a-Service and Self-Service models
• Scalable multi-tenant platform

• Identify clusters
• Create a consolidated release plan

• Select the right applications
• Devise an implementation plan
• Tool setup and process design
• Pilot for quick wins
• Continuous integration, delivery and more

• DevOps Alignment with Infrastructure Optimization e.g. Journey to Cloud
• Cultural and Change Management
• Stakeholder and Vendor Management
• DevOps Analytics and Artificial Intelligence
COMMON GROUND
CONSIDERATIONS THAT CUT ACROSS EACH LAYER
The DevOps journey is not merely technical; it also includes governance and metrics, tools and technology, people and culture, and change management. The following principles apply to almost every layer of DevOps adoption.

**Transformation Alignment**

Most organizations undergo several transformations at various levels. Infusing DevOps into these transformations can streamline and reduce the overall implementation effort.

**EXAMPLE**

When an application is transitioning from Waterfall to Agile, DevOps can augment the benefits of Agile through continuous integration and delivery. DevOps principles can also inform resourcing decisions, e.g., structuring operations roles into scrum teams.

**Alignment with Infrastructure Optimization**

Bloated infrastructure drives up IT costs. DevOps unlocks efficiency through both optimization and modernization, i.e., considering how the organization might benefit from modern infrastructures, such as cloud. These types of improvements directly contribute to business competitiveness.

**EXAMPLE**

When an organization is undergoing cloud adoption, DevOps can significantly enhance infrastructure setup and operations by automating practices like provisioning/deprovisioning, scalability, monitoring, backup, and security provisioning through infrastructure-as-code techniques.

**Cultural and Change Management**

Studies show organizations that ignore cultural and change management during a transformation journey fail to transform successfully. Like any other transformation, DevOps adoption requires training, mentorship, up-skilling/cross-skilling, behavioral change, motivation/reward, sentiment analysis, and assessment across all levels of the organization. Cultural and change management experts should approach DevOps with a concrete roadmap of implementation details, executional checkpoints and feedback loops.
Stakeholder and Vendor Management

Most organizations have multiple stakeholders and vendors managing various IT and business functions (e.g., development or testing), but when vendors fail to collaborate cohesively, adoption tends to fail. Becoming an “owner of the adoption” is a success factor in scaling the DevOps transformation. This requires commitment to change and effective coordination of stakeholders, with a strong show of support from senior leadership.

DevOps Analytics and Artificial Intelligence

It is critical to continue implementing new technologies that bring added value to the adoption process as it matures. Two such technologies include DevOps analytics and artificial intelligence.

DevOps analytics turns data from DevOps tools into insights that aid in decision-making. It also gives stakeholders visibility into various DevOps practices, helping them identify strengths and opportunities for improvement across every aspect of the adoption process.

**EXAMPLE**

Adoption owners can find the root cause of a bottleneck in software agility much faster among large application portfolios using DevOps analytics.

Artificial intelligence also helps in making DevOps ecosystems more intelligent by replicating aspects of human behavior. Through pattern recognition, learning, logic and decision-making, artificial intelligence can facilitate DevOps practices and significantly improve adoption maturity.

**EXAMPLE**

Organizations can automate the rectification of software delivery issues as well as create systems that learn by themselves using only the input of sample data.
Optimize Applications

Select the Right Applications

Some applications benefit more from DevOps than others. Selecting and prioritizing the right ones can be tricky, especially in large organizations with broad application portfolios. Typically performed at the same time, the following assessments are helpful in culling down a shortlist.

- **Cost-Benefit Analysis** – Compare the tentative cost of implementing DevOps with the potential benefit for each application. In the Figure 2, applications A and G represent optimal targets, based on this analysis.

- **Application Characteristics Analysis** – Factors like risk, complexity and time-to-market also impact selection. Analyzing ROI for each characteristic in a given application helps to illustrate the potential for improvement from DevOps as a whole. Figure 3 suggests the “System of Engagement” application should be prioritized, as it stands the gain the most.

- **Business and IT Priority Analysis** – Outline business and IT priorities of each application under consideration. The higher the priority, the earlier application should be slated for DevOps adoption, assuming the above analyses support the same finding.

Devise an Implementation Plan

After selecting your applications, proceed to assess each application’s current state of DevOps maturity across the following aspects:

- Delivery Approach
- Release Management and Governance
- Build Management
- Continuous Integration
- Deployment and Platform Provisioning
- Continuous Delivery (including Quality Engineering and Infrastructure)

Next, work with stakeholders to identify target maturity levels. The gap between current and target levels will help you create a detailed implementation plan and schedule.
Figure 3 — Application Characteristics Analysis

<table>
<thead>
<tr>
<th>System Type</th>
<th>Engagement with Business</th>
<th>Requirement for Time to Market</th>
<th>Opportunity for Innovation</th>
<th>Degree of Changes and Need for Team Collaboration /Co-location</th>
<th>Risk and Cost of Implementing Changes</th>
<th>Organization and Technology Complexities</th>
<th>Potential Benefits for Adopting DevOps/Agile</th>
<th>Priority of Implementing DevOps</th>
</tr>
</thead>
<tbody>
<tr>
<td>System of Engagement</td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
</tr>
<tr>
<td>System of Insight</td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
</tr>
<tr>
<td>System of Processing</td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
</tr>
<tr>
<td>System of Records</td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
<td><img src="Score.png" alt="Score" /></td>
</tr>
</tbody>
</table>

**Tool Setup and Process Design**

Tools and process should be established prior to implementation. Following the “standardization at higher levels and industrialization at lower levels” model of governance, strategic decisions at the LoB and application cluster layers should drive tool and process selection here. Setup should also be completed in tandem with other activities in those layers.

To ease standardization and scaling of DevOps adoption, the tool set should be hosted in a centralized DevOps platform, as opposed to each application building its own.

**Pilot for Quick Wins**

“Start small, prove right and then scale the transformation quickly.” This philosophy aims to provide early benefits and assurance to stakeholders before they invest to fully scale the DevOps transformation.

Make a small investment to pilot one of your selected applications with the goal of demonstrating initial DevOps improvements and benefits to others. Then continue with other applications and further layers of adoption.

**Continuous Integration, Delivery and More**

This is the heart of DevOps adoption. Your applications are selected, your tools and processes are in place, and you created a promising pilot. Now it’s time to execute the following practices against your implementation plan.

**Continuous Integration**

- Application Lifecycle Management
- Project Collaboration
- Software Configuration Management
- Peer Review
- Automated Code Analysis
- Automated Security Code Scanning
- Automated Build/Packaging and
- Dependency Management

**Continuous Delivery**

- Automated Deployments
- Automated Functional and Non-Functional Testing
- Automated Test Data Management and Service Virtualization
- Integrated Monitoring and Operations
**Organize Application Clusters**

**Identify Clusters**

All dependent applications should move through development and testing cycles together at the same pace, aiming for a consolidated release. Grouping related applications into clusters is the first step to improving overall release speed. This requires coordination, which can be difficult due to lack of leadership, circular dependencies, or disparities among stakeholders, vendors, geographies, release plans and processes.

Once a cluster is established, DevOps should be applied to all or most applications within it. If not, the non-DevOps applications will move slower than others, delaying the target consolidated release speed. In these cases, it can be beneficial to implement techniques like feature toggling and backward compatibility in faster applications so as to maintain speed while allowing slower applications catch up when they are ready.

**Create a Consolidated Implementation Plan**

Planning the consolidated release of an application cluster requires an assessment of the challenges at hand as well as the techniques to overcome them. Understanding both strengths and opportunities for improvement is a key part of this assessment, which should recommend an optimal release speed that is attainable by every application in the cluster.

The assessment and implementation plan should respect all genuine caveats as well as issues that are beyond the team’s control or those cannot be improved within reasonable cost and time.
Empower LoBs

These practices help to scale adoption across various LoBs. An organization cannot realize the benefits of DevOps until it has successfully scaled and matured at all levels.

Standardization at Higher Levels, Industrialization at Lower Levels

Governance is one of the most critical success factors in scaling a transformation. Lean governance accelerates adoption by avoiding overhead and promoting faster decision-making. Using this model, DevOps standards are defined at the LoB level and systematically adopted at lower levels.

Lean governance expedites adoption at lower levels with bureaucracy-free workpaths, lightweight processes, regular checkpoints, continuous monitoring, and a clear escalation and authority hierarchy with key stakeholder involvement. This type of governance is fueled by a strong managerial commitment to adapt and fine-tune guidelines with speed, lead by example, and facilitate and authorize tools and processes at lower levels.

DevOps governance can also be combined with governance of other ongoing transformation initiatives.

Organizational Structure

Well-structured IT teams greatly enhance DevOps adoption, but there is no single recommendation in this area. Distinct LoBs often build structure differently while maintaining the ability to work together efficiently. When devising new structures, it’s critical to respect the boundaries of other IT functions and adhere to LoB guidelines.

A shared DevOps team (also called a “Center of Excellence” or “CoE”) is a highly effective structural solution in most cases. A CoE is a virtual team of DevOps experts of varying experience and skill levels designed to aid adoption in lower layers. This model relies on the DevOps principles of cooperation and collaboration.

CoE experts should be organized in two levels, as shown in Figure 4.

1. **Application and Cluster Level** - These experts have a dual agenda: The first is to implement DevOps practices and accelerate adoption. The second is to train and mentor application teams to become self-sufficient in DevOps methods.
2. Centralized CoE Level – These experts are responsible for defining DevOps practices, building and maintaining a centralized platform, developing innovations, and executing pilots and proofs of concept.

The CoE model requires a great degree of alignment, communication and sharing of work between groups. It is also beneficial to rotate experts regularly to help them gain experience and diversify learning. This enables the CoE to improve both functions by leveraging the collective knowledge of the whole.

A Scalable Multi-Tenant Platform

One of the primary services provided by the CoE is the creation of a DevOps platform using the tools outlined by the governance structure. This platform is used at lower levels to onboard applications and facilitate DevOps practices.

Depending on the strategy defined by the governance team, this secure platform can be hosted on premise or in the cloud, with continuous monitoring.

DevOps-as-a-Service and Self-Service Models

CoEs become more sophisticated with scale. For instance, application teams can request CoE services from a published catalog using the DevOps-as-a-Service model, where they can also track the status and quality of service delivery. Alternatively, self-service may be a practical way to speed up certain aspects of adoption at the ground level. The catalog and other service features are typically accessible through an online portal.

Sample DevOps Portal Features

- Service catalog
- Service status and tracking (using service metrics/KPIs/SLAs)
- Self-service access
- Guidelines and best practices
- Training and resources

Figure 5 depicts the process by which a CoE would develop a service catalog, which offers the following information at minimum:

- Service name
- Description and full scope
- Complexity level
- Unit of effort consumed by the CoE in providing said service
- Metrics/KPI/SLA associated with service

Building on this model, a CoE may also offer reusable DevOps artifacts through an App Store-like interface, where consumers can download and use various applications locally.
Align the Organization

The intent to become a DevOps organization and the commitment required to do so lie at the highest level of the pyramid. Here, it’s important to streamline adoption by syncing LoBs with organizational strategy and enterprise architecture.

Consolidated Governance

If multiple governance models are at play among various LoBs, they should be consolidated at the organizational level to facilitate DevOps adoption. This is a key principle of “lean” governance, which allows for more responsiveness and collaboration among LoBs to promote reusability, collective learning, thought leadership and measurability.

The DevOps Effect

DevOps adoption is a multidimensional journey that affects many functions within an organization. As such, it is critical to align DevOps with the overall enterprise strategy in addition to business and IT architecture. This enables measurement of results against organizational objectives.

For example, if an organization’s strategy is to become a digital organization within three years, it will likely undergo several transformation programs across various LoBs at the same time. DevOps must align with these programs at every level to accelerate them and reach its own maximum business potential.
The answer is: Both.

Ideally, DevOps adoption should begin at higher levels to foster the proper setup of standardized governance structure, platforms, tools, processes and guidelines—all of which are to be industrialized at lower levels.

However, application teams should be empowered to “just start with DevOps” at lower levels prior to standardization, in the spirit of attaining quick wins and pilots. They can implement basic DevOps practices (e.g., SCM, Unit Test Automation, Build Automation, etc.), knowing to expect minor adjustments as platform, governance and CoEs are formally established down the road. At that point, higher levels become the driving force of DevOps adoption across all layers.

In this way, an organization can start and scale (i.e., standardize and industrialize) DevOps adoption with continuous learning.
The content of this paper is based on the authors’ experience and understanding of organization’s requirements, challenges and what has worked in various projects.

Please reach out to the key contacts below if you would like to provide any feedback or experiences on the content of this paper.

Disclaimer
This paper contains generic advise for DevOps adoption in an organization, based on the experience of the authors, which may also overlap the views/ideas of other DevOps practitioner in the industry shared in public domain. The organizations and individuals should consider their respective ecosystem/landscape/requirements/scope and do the required due diligence before applying them. Accenture or the authors are not responsible for the implementation or the outcome of DevOps adoption (advised in this paper) unless consulted/engaged in the adoption.

The paper is the property of Accenture and its affiliates and Accenture be the holder of the copyright or any intellectual property what-soever in the Paper. No part of this Presentation may be reproduced, distributed, copied in any manner without the written permission of Accenture. Opinions expressed herein are subject to change without notice.
AUTHORS

Manoj Seth
Global Lead of DevOps Solution Architecture
Manoj Seth is Global Lead of DevOps Solution Architecture in Accenture Technology Services. In this role, he has been helping various industry clients in building and implementing DevOps strategy for their transformational and modernization programs.
manoj.seth@accenture.com

Vijeth Hegde
Lead of DevOps Accenture Technology Services
Vijeth Hegde is the Lead of DevOps within Accenture Technology Services-India. In this leadership role, Vijeth focuses on the overall DevOps capability building, asset creation, solutioning, consulting, architecture, new offerings, industrialize, training, delivery and sales support.
vijeth.s.hegde@accenture.com

Rajendra Prasad
Global Lead of DevOps, Agile, Automation and AI
Rajendra Prasad is the Global Lead of DevOps, Agile, Automation and Artificial Intelligence in Accenture Technology Services. In this leadership role, Rajendra Prasad focuses on driving efficiency in Application Services and IT operations for clients across industry. He leads the team that created and currently deploys Accenture myWizard®, an intelligent automation platform with artificial intelligence at its core and integrated with DevOps/Agile tools/technologies.
rajendra.t.prasad@accenture.com

CONTACTS

Federica Falcone
Global DevOps Lead
federica.falcone@accenture.com

Marianne Breen – Hart
Global DevOps Strategy and Operations Lead
marianne.l.breen@accenture.com

Copyright © 2018 Accenture
All rights reserved.

Accenture, its logo, and High Performance Delivered trademarks of Accenture.