Adopting the IBM DevOps approach for continuous software delivery

Adoption paths and the DevOps maturity model

Paul Bahrs

October 11, 2013

IBM DevOps introduces an enterprise capability for continuous software delivery, enabling organizations to seize market opportunities and reduce time to customer feedback. This article outlines four paths to adopting or improving continuous software delivery in your organization. In addition, this article presents a practice-based framework (the DevOps maturity model) that can help you assess your current practices, set your goals, and measure your improvement along a continuum as you adopt the IBM DevOps approach.

Introduction

IBM DevOps introduces an enterprise capability for continuous software delivery, enabling organizations to seize market opportunities and reduce time to customer feedback. This capability unites the people, practices, technologies, and information that support your software delivery process.

DevOps: The IBM approach

The IBM white paper DevOps: The IBM approach explains why continuous software delivery is an essential enterprise capability.

This article outlines four paths to adopting or improving continuous software delivery in your organization. These adoption paths focus on: Planning and measuring, Developing and testing, Releasing and deploying, and Monitoring and optimizing. In addition to the adoption paths, this article also presents a practice-based framework, called the IBM DevOps maturity model, which can help you assess your current practices, define your roadmap, and measure your improvement along a continuum as you adopt the IBM DevOps approach.

Adoption framework

The IBM DevOps maturity model is your communication tool to inform your enterprise of your strategy for improvements. This model transcends organizations, practice areas, and technologies. Our practice-based model reflects the broader context of an enterprise adoption framework. It
defines best practices for executing a strategy for adopting new solutions iteratively and enables your organization to assess, plan, define, and deploy improvements that achieve measurable outcomes to your enterprise. The adoption framework prescribes the steps to prepare, pilot, and release improvements into your enterprise.

One of the first steps in the adoption framework, shown in Figure 1 and Table 1, is to assess your current maturity level and decide on the scope, architecture, and roadmap for adoption or improving your enterprise DevOps capability.

### Figure 1. Adoption framework

![Adoption framework diagram](image)

### Table 1. Activities in the adoption framework

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assess and decide</td>
<td>Assess your current maturity and decide on the scope, architecture, and roadmap for adopting or improving your enterprise DevOps capability.</td>
</tr>
<tr>
<td>Establish a core team</td>
<td>Establish a leadership and mentoring group to drive the improvements/adoption and support a broader, virtual team of subject matter experts.</td>
</tr>
<tr>
<td>Define a working model</td>
<td>Define a usage model that addresses processes and activities for each role, as well as tool configurations and supporting architecture for the capabilities planned in the pilot. This will then be used for user acceptance test of the solution and by the pilot teams during the pilot.</td>
</tr>
<tr>
<td>Prepare for a pilot</td>
<td>Define pilot outcomes/objectives and measurements for validating the objectives. Using the usage model, produce assets to support introducing the solution to pilot teams. Assets may include training, job aids, plugins, reports, and queries.</td>
</tr>
<tr>
<td>Pilot the project(s)</td>
<td>Use defined improvements with real data in a production project or initiative. Measure objectives, validate the usage model, and then improve the solution for the next pilot/release.</td>
</tr>
<tr>
<td>Execute the adoption strategy</td>
<td>Following the pilot, deploy the capability or capabilities piloted across organizations, technologies, and applications according to the roadmap.</td>
</tr>
<tr>
<td>Release</td>
<td>The release is a well-defined improvement that produces measurable outcomes for users, teams, projects, programs, departments, and the enterprise.</td>
</tr>
</tbody>
</table>

### Adoption paths

This table, derived from the white paper DevOps: The IBM approach, defines the DevOps capabilities used to explain what people, practices, and technology should be considered when defining current maturity as well as defining the improvement roadmap.

Adoption paths
Table 2. Adoption paths

<table>
<thead>
<tr>
<th>Adoption path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan and measure</td>
<td>Consists of a business planning practice. <strong>Business planning and measuring</strong> employs lean principles to start small by identifying the outcomes and resources needed to test the business vision/value, adapt and adjust continually, measure actual progress, learn what customers really want, and shift direction with agility, and update the plan.</td>
</tr>
<tr>
<td>Develop and test</td>
<td>Consists of collaborative development and test practices. <strong>Collaborative development</strong> enables collaboration between business, development, and QA organizations—including contractors and vendors in outsourced projects spread across time zones—to deliver innovative, quality software continuously. This includes support for polyglot programming and support of multiplatform development, elaboration of ideas, and creation of user stories complete with cross-team lifecycle management. <strong>Continuous testing</strong> reduces the costs of testing while helping development teams balance quality and speed. It eliminates testing bottlenecks through virtualized dependent services, and it simplifies the creation of virtualized test environments that can be easily deployed, shared, and updated as systems change. These capabilities reduce the cost of provisioning and maintaining test environments and shorten test cycle times by allowing integration testing earlier in lifecycle.</td>
</tr>
<tr>
<td>Release and deploy</td>
<td>Consists of a release and deployment practice. <strong>Release and deployment</strong> provide a continuous delivery pipeline that automates deployment to test and production environments. It reduces the amount of manual labor, resource wait-time, and rework by means of push-button deployments that allow higher frequency of releases, reduced errors, and end-to-end transparency for compliance.</td>
</tr>
<tr>
<td>Monitor and optimize</td>
<td>Consists of monitoring, customer feedback, and optimization practices. <strong>Monitoring</strong> offers easily accessible and consumable reporting that helps developers and testers understand the performance and availability of their application, in production as well as before it's deployed to production. The early feedback provided by continuous monitoring is vital for lowering the cost of errors and change, and for steering projects toward successful completion. <strong>Continuous customer feedback</strong> provides the visual evidence and full context for analyzing customer behavior and pinpointing customer pain points. Feedback can be applied during both pre- and post-production phases to maximize the value of every customer visit and ensure that more transactions are completed successfully. This allows immediate visibility into the sources of customer struggles that affect their behavior and impact business.</td>
</tr>
</tbody>
</table>

DevOps maturity

IBM’s experience in helping organizations successfully adopt the IBM DevOps approach supports a business-driven strategy for adopting or improving DevOps capabilities and outcomes. Creating a business-driven strategy includes prioritizing a set of measurable business goals, benchmarked against your current practices, and developing an incremental adoption roadmap that maintains alignment to your goals. The roadmap should provide a guide to capability improvements and
identify incremental steps to achieving your business goals. The best first step in defining a roadmap is to start with assessing your current maturity levels and defining objectives to support your goals.

Using IBM’s practice-based maturity model and developing a prioritized heatmap, described in the following sections, can provide a framework to define and achieve your outcomes, as well as a plan and a way to measure your improvement.

**Prioritized capability improvements**

IBM recommends defining priorities for improvement across your organization by assessing each high-level capability associated with each adoption path. Underlying dependencies across organizations or roles may also be identified to show where collaboration is breaking down or weak. The assessment can be tailored for specific initiatives already in progress in your organization to narrow the focus to specific areas of concern. Prioritizing improvements through a high-level capability model, as shown in Figure 2, helps you define the scope and key capabilities needed to adopt DevOps in a common way to all stakeholders.

**Figure 2. Sample capability improvement heatmap**

![Sample capability improvement heatmap](image)

**Practice-based maturity model**

The IBM DevOps practice-based maturity model defines four levels of maturity correlating to how your organization currently performs practices aligned to each adoption path. The maturity level increases as you move from *practiced* to *scaled*. The model includes maturity of practices for each of the adoption paths.

**Maturity levels**

The IBM DevOps maturity model defines four levels of maturity that describe how well an organization can perform practices aligned to each adoption path. The levels consider:
consistency, standardization, usage models, defined practices, mentor team or Center of Excellence (COE), automation, continuous improvement, and organizational or technical change management.

Table 3. Maturity levels

<table>
<thead>
<tr>
<th>Maturity level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Practiced</td>
<td>Some teams may exercise activities associated with the practice, but are inconsistent. No enterprise standards defined. Automation may be in place but without consistent usage models.</td>
</tr>
<tr>
<td>Consistent</td>
<td>Enterprise standards for practice are defined. Some teams exercise activities associated with the practice and follow the standards. No core team or COE to assist with practice adoption. Automation, if used, follows enterprise standards.</td>
</tr>
<tr>
<td>Reliable</td>
<td>Mechanisms exist to assist adoption and ensure that standards are being followed. A core team of mentors is available to assist in best practices, education, and adopting improvements.</td>
</tr>
<tr>
<td>Scaled</td>
<td>Adoption of the practice is institutionalized across the enterprise. COE is a mature and integral part of continuous improvement and enablement. Practices are mainstreamed across the enterprise. A feedback process is in place to improve the standards.</td>
</tr>
</tbody>
</table>

The maturity model

Figure 3 shows the IBM DevOps maturity model. The vertical columns align to adoption paths, and the horizontal rows indicate maturity levels that increase as you move from bottom to top, where **practiced is the lowest measurable level** and **scaled is the highest**. Each column represents the set of practices and maturity levels for a specific adoption path. This model provides a way to assess and view maturity across the DevOps continuum and is intended to be consistent with the objectives and outcomes of other specialized models such as Agile, SOA, and CMMI.

Figure 3. DevOps maturity model
Maturity progression

The following sections explain maturity level progression for each adoption path. They describe how an organization would mature along an adoption path by adopting new practices and supporting technologies. It is important to understand that to achieve a level maturity in an adoption path, an organization may have to also mature in other, dependent practices. For example, to effectively implement a real continuous testing capability, an organization would need a capability for continuous integration and build of applications, as well as a capability to continuously deploy the builds into the test environment.

Plan and measure

At the practiced level, organizations capture business cases or goals in documents for each project to define scope within the strategy, but resourcing for projects is managed at the department level. Once projects are executed, changes and scope are managed within the context of the project or program to achieve goals within budget and on schedule. As organizations mature, business needs are documented within the context of the enterprise and measured to meet customer value metrics. Those needs are then prioritized, aligned to releases, and linked to program or project requirements. Project changes and scope are managed at the portfolio level.

Develop and test

At the practiced level, project and program teams produce multiple software development lifecycle products in the form of documents and spreadsheets to explain their requirements, design, and test plans. Code changes and application-level builds are performed on a formal, periodic schedule to ensure sufficient resources are available to overcome challenges. Testing, except for unit level, is performed following a formal delivery of the application build to the QA team after most, if not all, construction is completed. As organizations mature, agility in software development emerges to improve business alignment, driving the need for accelerated and continuous feedback. Software delivery, integration, and build are performed routinely and then on a continuous basis for individual developers, teams, applications, and products. The move to early feedback and continuous delivery encourages test maturity by leveraging automated testing and virtualization. A centralized testing capability provides services across projects to continuously run regression and other automated testing, provided that the infrastructure and application deployment can also support these tests. Linked lifecycle information begins to support improved collaboration in the context of cross-functional information. This eventually provides the basis for development intelligence used to assess the impact of continuous process and technology improvements.

Release and deploy

At the practiced level, releases are planned annually for new features and maintenance teams. Critical repairs and off-cycle releases emerge as needed. All releases are managed in a spreadsheet updated through face-to-face meetings. Impact analysis of change is performed manually as events occur. Application deployments and middleware configurations are performed consistently across departments using manual or manually staged and initiated scripts. Infrastructure and middleware are provisioned similarly. As the organization matures, releases
are managed centrally in a collaborative environment that leverages automation to maintain the status of individual applications. Deployments and middleware configurations are automated and then mature to a self-service model that provides individual developers, teams, testers, and deployment managers with a capability to continuously build, provision, deploy, test, and promote. Infrastructure and middleware provisioning evolves to an automated then self-service capability similar to application deployment. Operations engineers cease manually changing environments; instead they focus on optimizing the automation.

Monitor and optimize

At the practiced level, deployed resources are monitored, and events or issues are addressed as they occur, without context of the affected business application. Development and Operations organizations coordinate informally and usually are driven by specific events or issues. Feedback of user experience with business applications is limited and only achieved through formalized defect programs. As organizations mature, monitoring is performed within the context of business applications, and optimization begins in QA environments to improve stability, availability, and overall performance. Customer experience is monitored to optimize experiences within business applications. Optimization to customer key performance indicators that reflect business value attainment is part of the continuous improvement program.

DevOps assessment and planning

How to begin your DevOps journey? The most difficult part is deciding on the right first step and then maintaining support throughout the adoption process. We have found that a great first step is to outline your organization's DevOps investment strategy through initiatives that achieve maturity improvements. These initiatives form the basis of incremental improvement starting with your current practice maturity level. Hold a cross-functional discussion to define the initiative goals, roadmap prioritized improvements, and identify quick wins that your organization can achieve. Then, use the quick wins that can be achieved in the first 90 days to build momentum and maintain grassroots energy and support from management.

One way to define your initiatives, or identify and execute on those quick wins, is to take advantage of IBM’s DevOps assessment and planning workshop. It's designed to accelerate your efforts to get started with DevOps improvements and adoption. Using our practice-based approach, we leverage the DevOps maturity model to benchmark your current maturity and suggest incremental improvements. During the workshop, our expert consultants facilitate a collaborative discussion with your team to prioritize a set of DevOps improvement initiatives, define outcomes, and lay out an overall strategy. We can then assist you in defining a series of prioritized practice improvements that form the basis for an executable roadmap and architecture to support each initiative. We review your current capabilities and explore improvements for each practice area (such as requirements, build, deploy, management, etc.) to formulate a prioritized, prescriptive roadmap for iterative DevOps improvements. The workshop focuses on specific areas
of the overall software delivery continuum to deliver an executable strategy for implementing a set of incremental initiatives to improve your DevOps capability. The workshop accelerates your first steps in your DevOps journey and delivers a defined set of initiatives, a strategy, and a detailed improvement roadmap and architecture.
Related topics

- **DevOps learning circle**: Find experts and guided learning activities to help you build your personal learning roadmap.
- **IBM Software Services for Rational**: Learn about expert services and training to help you unlock the power of your Rational software investments.

© Copyright IBM Corporation 2013
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)