Using DevOps’ Intelligent Insights to Deliver Greater Business Value

By applying real-time analytics and end-to-end traceability enabled by DevOps, IT organizations not only make software delivery more efficient, but they also enhance business value delivered throughout the digital journey.

EXECUTIVE SUMMARY

For large enterprises, accelerating their velocity to being digital is becoming increasingly difficult. The process is hindered by historical work silos and the inability to transform older, legacy applications and technology infrastructure.

While DevOps' enables business to address market demand for rapid digitization, the approach primarily focuses on end-to-end automation, from application build and deployment, through release and measurement. This is a great first step, but issues are often uncovered while applications are in production. Some of these issues need immediate resolution, increasing the risk of budget and schedule overruns. Thus, diligent and continuous measurement is necessary to ensure enterprise DevOps success.

In a DevOps world, where delivery is agile, rapid and conducted in short Sprints, emphasis should be on proactive issue detection rather than reactive responses to bugs and challenges. This white paper discusses how organizations can use rapid feedback mechanisms to attain continuous intelligent insights that help continuously improve an application's value.
CIOs who comprehend DevOps’s value should also understand that DevOps is a journey and not an end state, and thus they should use this approach to continuously evolve and provide better application services to their customers.

THE ROLE OF DEVOPS IN DIGITAL TRANSFORMATION

CIOs are leading their IT organizations through ongoing digital transformation that extends to every aspect of the company. Yet, decision-makers are often at a loss as to where to start and sometimes wonder if they have made enough progress based upon their initial targets. Effective digital transformation requires organizations to possess the agility to innovate, deploy software and accelerate time to market. Organizations that are able to quickly turn their ideas into consumer products, absorb rapid feedback and then quickly refine their software to launch the suggested improvements are excelling in a world that is moving at the speed of digital. DevOps gives these organizations this kind of ability to deliver new digital ideas quickly. Organizations adopt the DevOps philosophy to bring sustained continuous development, integration and innovation.

MEASURING DEVOPS FOR DIGITAL SUCCESS

However, the greatest challenge today is that continuous software updates are not so simple to deploy. Multiple apps must be rendered on mobile and web interfaces, especially in industries where customers need consistent anytime, anywhere access to information and application services. Any downtime, however minor, can be extremely costly. Despite pre-launch testing, technical issues still arise in production, some of which could be severe enough to warrant immediate remedies. These can totally disrupt development schedules and, more important, mar the business’s reputation and cause financial hardship, if not remedied quickly and effectively. And often, the repercussions of such glitches depend on how quickly an issue is discovered, diagnosed and fixed. Issue detection needs to be immediate and automatic, and the information captured must facilitate remedial action.

In addition to remediating production issues, there is also a need to continuously improve application services for the customers at a greater velocity, to fast-track their digital transformation. But the velocity should not come at a cost of quality, and this is one prime area where DevOps can help. CIOs who comprehend DevOps’s value should also understand that DevOps is a journey and not an end state, and thus they should use this approach to continuously evolve and provide better application services to their customers.

Next-gen analytics tools and cognitive technologies such as autonomics have opened up the possibility to add adaptive automation within DevOps’ continuous delivery (CD) implementations. This facilitates not only identifying issues but also enables the systems to take mindful decisions in solving some of the most common problems in the software delivery lifecycle. As a result, the velocity of applications changes released to production increases significantly and the customer’s digital journey accelerates.
Organizations that are unable to baseline DevOps outcomes are unable to continuously improve; in the long run, they could potentially become laggards.

DEVOPS MEASUREMENT CHALLENGES

There is an old management saying: If you can't measure it, you can't improve it. DevOps is not a one-time activity, but an ongoing process where IT organizations improve the various activities across the systems development lifecycle (SDLC) of build, deploy, verify, release and measure. In this way, they work to significantly accelerate time to market and business value. Organizations that are unable to baseline DevOps outcomes are unable to continuously improve; in the long run, they could potentially become laggards. This simple thought drives DevOps analytics.

Unfortunately, there is no single standard by which we can define or baseline DevOps indices that need to be measured. This is primarily due to the fact that not all parameters that are measured may be applicable to all technology stacks — even if the concept of DevOps may be still applicable to them. Therefore, if an enterprise wishes to measure its DevOps journey, it would have to first create baseline measurements on how much time is being spent inside SDLC processes and how DevOps can help reduce that. If these indices show progress once an enterprise adopts DevOps, it could justify the investment. But these measurements and computations of progress are made over a period of time and are not simple to measure due to the following:

- The indices change based on the customer and what the customer wants to measure.
- The indices change based on the type of the vector that needs to be measured: build, deploy, verify, release, measure and environment (in some cases).
- There is no one single tool or mechanism that helps to measure DevOps progress.

DEFINING THE METRICS THAT MATTER

An obvious first step is to identify the basic performance indicators to measure — i.e., define the KPIs. Based on our experience, the metrics in Figures 1 and 2 (see next page) are good starting points. One caveat: The figures offer an indicative view of the various performance indicators; they are not intended to be an exhaustive catalog. These could be used for creating baselines, and the trends, either upward or downward (based upon the metrics), can help reveal if the needle is tipping in the right direction.
Metrics to Measure Velocity

<table>
<thead>
<tr>
<th>Performance Indicator</th>
<th>What It Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Lead Time</td>
<td>Time taken for a bit of code to get built, tested and deployed.</td>
</tr>
<tr>
<td>Daily Change Rate</td>
<td>Number of changes committed to the main branch of source control and deployed to the necessary environments.</td>
</tr>
<tr>
<td>Mean Time to Environment Setup</td>
<td>The time taken for developers/testers to create a testing environment for verifying each delivered change.</td>
</tr>
<tr>
<td>Mean Time to Detect</td>
<td>Time elapsed since the original coding until the bug it introduces is detected.</td>
</tr>
<tr>
<td>Mean Time to Resolve</td>
<td>Time taken to resolve an issue after detection.</td>
</tr>
<tr>
<td>Mean Time to Approve</td>
<td>Time taken to approve and verify a release. (This is measured from the moment all release content has been delivered until the release has passed all the defined test and verification cycles.)</td>
</tr>
</tbody>
</table>

Figure 1

Velocity Metrics

These metrics (in Figure 1) essentially measure the rate at which change is being delivered. And “change” could be how quickly a problem is detected, how quickly it can be fixed, or how quickly a basic need can be assigned to an engineering team.

Quality Metrics

Apart from agility, quality is of paramount importance in the DevOps world. The quality metrics listed in Figure 2 help to define the rate at which quality factors can be measured.

Metrics to Measure Quality

<table>
<thead>
<tr>
<th>Performance Indicator</th>
<th>What It Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build Failure Rate</td>
<td>Ratio of failed builds to overall builds.</td>
</tr>
<tr>
<td>Deployment Failure Rate</td>
<td>Ratio of failed deployments to overall deployments.</td>
</tr>
<tr>
<td>Infrastructure-Related Failure Rate</td>
<td>Ratio of build/deployment failures related to infrastructure issues.</td>
</tr>
<tr>
<td>Rework Rate</td>
<td>Ratio of tickets being reopened to total tickets.</td>
</tr>
<tr>
<td>Automated Detection Rate</td>
<td>Ratio of defects detected by automated testing cycles to overall detects.</td>
</tr>
</tbody>
</table>

Figure 2
DATA COLLECTION AND MEASUREMENT

Once key performance metrics are identified, the next step is to gather the data to make these measurements. After collecting data for the individual metric, it is necessary to plot each data point to derive meaningful insights. Some of the data can be used directly to derive meaning, while other data might require complex calculations to quantify the value or the outcome. While the numerical value of an isolated data point may be relevant, generally the real impact of such data is realized when the decision-maker conducts trend analysis over a period of time. This can reveal progress – or the lack thereof.

Given the complexity of data collection, analysis and interpretation, it is obviously suboptimal to perform such processes manually. The principles of DevOps need to be applied to measuring outcomes, as well. In Figure 3, we present a sample setup for data collection to create a DevOps measurement baseline.

This figure illustrates at a high level how data can be collected, and the top characteristics of such a system. The system basically contains touch points (integration points) with the necessary tools that are deployed to collect data and then portray the full picture of how the application code moves through the various phases.

Creating Measurement Baselines

DevOps Measurement Platform

- Provide capability to correlate between data
- Should be able to work with all the tools
- Figure out bottlenecks
- Simple and effective ways to collect data
- Simple setup

Figure 3
LOOKING FORWARD

Most DevOps consultants and architects are challenged to address whether DevOps is more concerned with IT than business outcomes. We believe that this is one of the most challenging questions IT organizations face. Answering this with data-driven measurements and metrics (as described above) is therefore critical. This is the primary goal of large transformation engagements wherein DevOps is the key change lever.

Quick Take

Improving Time to Market, Productivity and Incidence of Bugs

A large enterprise client sought to apply DevOps to improve application time to market. Initially, the company’s IT organization was unable to figure out where the bottlenecks were in its SDLC process, which affected developer productivity.

By applying key principles of DevOps to identify bottlenecks in its development process, the company greatly reduced cycle time from development to production release. This also significantly reduced defects, because the IT team was able to more proactively find and treat root cause issues.

Overall, the company’s IT organization increased developer productivity by 15%, an outcome measured by its own analytics and correlation engine.
Sample Dashboards: DevOps by the Numbers

When business teams are able to realize these outcomes, they will also be part of the transformation journey — and hence DevOps will no longer be an “IT thing.”

Sample dashboards that can be created based on the identified metrics are illustrated in Figure 4. These provide a sense of what exactly needs to be improved and what can be accomplished via DevOps.

By examining the figure’s three reports, DevOps teams can see how a particular Sprint impacted the number of files that changed but didn’t improve the quality of the CI builds and the deployments, given the large number of failures. This basically shows that the team has good velocity but must improve on their quality measures. Inferences like these will help the teams improve in their DevOps journey.

DevOps measurement enables organizations to answer the age-old return on investment (ROI) question. The key first step to ensure that DevOps leads to successful digital transformation is to define and measure DevOps metrics. The following are the priority items that organizations should bear in mind when defining the metrics:

- Identify what needs to be measured.
- Which actors contribute to the KPI — these could be tools, process and technology, not just people — or collect data for metrics?
- Measurement of the metrics with a defined baseline.
- Progress tracking metrics to ensure continued forward motion in adopting of DevOps principles.
Repeating these steps for each KPI, and then taking the actual measurements, will allow an organization to objectively measure DevOps’ outcomes and weed out the bottlenecks — and thereby accelerate the organization’s digital journey. Figure 5 presents some additional outcomes that can be achieved from both a business and technical standpoint when the above measurements are put in place.

**Business Benefits**

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improve Team Efficiency</td>
<td>By identifying – and then acting on – factors that limit the team's velocity, the team can deliver changes at a faster pace.</td>
</tr>
<tr>
<td>Portfolio Performance</td>
<td>This helps in measuring the relative maturity of a given portfolio. For the business, it serves as a measuring point for determining which areas need focus given a particular line of business.</td>
</tr>
<tr>
<td>Time to Market &amp; Trend</td>
<td>Time taken for a business feature to be rolled out from build to deployment, and the trend over a particular time frame. This helps the business to make corrective steps and to deliver products more rapidly.</td>
</tr>
<tr>
<td>Cost of Release</td>
<td>The ability to measure the cost vs. value delivered by a release. In an ideal scenario, the cost of the release should taper once DevOps principles are adopted – thus justifying the investment in this approach.</td>
</tr>
<tr>
<td>Impact of Release</td>
<td>The success/failure of features delivered to customers. This measurement gives the business the capability to gauge positive/negative ROI and help make feature delivery more customer-centric.</td>
</tr>
<tr>
<td>High Friction Zones</td>
<td>This helps to identify areas that choke application development velocity. They could be in development, QA or infrastructure; knowing where helps teams take corrective measures.</td>
</tr>
<tr>
<td>Technical Debt</td>
<td>Costs incurred for not following the best code practices, and the related expenses to fix them. This denotes that when headroom projects are not executed in a timely way it impacts the velocity of the delivered code changes.</td>
</tr>
<tr>
<td>Velocity and Trend</td>
<td>This is a direct measure of the team’s capability to deliver changes at a constant speed. The team should be able to increase its velocity of delivering features when the bottlenecks are removed by the application of DevOps principles.</td>
</tr>
<tr>
<td>Defect Fix &amp; Rate</td>
<td>Average time to fix defects and determine the trend. When defects of a particular type proliferate, it indicates a choke point in that area. For example, if environment defects are high, then the environment area needs some attention. This should decrease when DevOps principles are applied in the environment.</td>
</tr>
</tbody>
</table>
FOOTNOTES

1 DevOps is a term used to refer to a set of practices that emphasize the collaboration and communication of both software developers and information technology professionals while automating the process of software delivery and infrastructure changes - from Wikipedia. en.wikipedia.org/wiki/DevOps.


ABOUT THE AUTHORS

Rajkumar Chandrasekaran
Principal Architect, DevOps Practice

Rajkumar Chandrasekaran is a Principal Architect within Cognizant’s DevOps Practice. He has 15-plus years of experience in the field of large-scale application development and has played varied roles, from application architecting through reengineering of applications. Rajkumar is currently architecting Cognizant’s OneDevOps platform assets for the company’s DevOps offerings. He can be reached at Rajkumar.Chandrasekaran@cognizant.com.

Rangarajan Rajamani
Senior Manager, DevOps Practice

Rangarajan Rajamani is a Senior Manager within Cognizant’s DevOps Practice. He has over 14 years of expertise in multiple industries such as global life sciences, manufacturing and construction, performing business consulting, pre-sales and product evangelization. As a functional expert in digital technologies such as the IoT, cloud, service virtualization, functional QA automation and DevOps, Rangarajan has created point solutions for organizations in life sciences, insurance and retail. He currently anchors platform evangelization for Cognizant’s DevOps Practice. Rangarajan can be reached at Rangarajan.Rajamani@cognizant.com.
ABOUT COGNIZANT
Cognizant (NASDAQ-100: CTSH) is one of the world’s leading professionalservices companies, transforming clients’ business, operating and technology models for the digital era. Our unique industry-based, consultative approach helps clients envision, build and run more innovative and efficient businesses. Headquartered in the U.S., Cognizant is ranked 205 on the Fortune 500 and is consistently listed among the most admired companies in the world. Learn how Cognizant helps clients lead with digital at www.cognizant.com or follow us @Cognizant.