Are Your Embedded Analytics DevOps Friendly?

Don't Let Your Analytics Solution Weigh Down Your Infrastructure

insightsoftware

Contents

1 Why Embedded Analytics Needs to Work with DevOps

Every application is becoming an analytic application. Over 85 percent of software companies are currently embedding dashboards, reports, and analytics in their applications[1].

An increasing number of software teams are also adopting DevOps practices—combining methodologies and tools from software development (Dev) and information technology operations (Ops) to fuel more frequent application deployments. Reliance on DevOps climbed to 33 percent last year. That may not seem huge, but it's nearly double from the year before—and another 35 percent of respondents were on track to adopt DevOps methodologies before year end[2].

The Convergence of Embedded Analytics and DevOps

Decisions you make now about how to embed analytics in your software will affect how successful your application is over time. If your embedded analytics solution doesn't work with your current tech stack and DevOps practices, any update to the analytics could increase deployment complexity. Even if you build next-level dashboards and reports, your DevOps team could kill a solution because it's too complicated to put into production and impossible to maintain over time.

1 2018 State of Embedded Analytics Report, insightsoftware
2 2018 State of DevOps, Interop ITX and InformationWeek

Does your embedded analytics solution work with DevOps?

If you can answer this question with an unequivocal YES, you're all set to deliver compelling applications with analytics at their core while reducing deployment complexity. You're outmaneuvering competitors and reacting to fluid customer requirements. You can speed up releases—moving toward continuous delivery—while keeping product quality and reliability high.

If your answer to this question is a **NO** or **MAYBE**, you're at risk. This ebook covers what you need to get to **YES**.

2 | 3 Pillars of Successful Embedded Analytics

Every time you add or enhance the dashboards and reports in your application, you need to serve your end users, your development team, and your DevOps team. If you forget one, you put the entire project at risk.

1. End Users

By embedding dashboards and reports in your application, you give end users powerful insights in context. Without ever leaving your application, users can see relevant data and take action. You can tailor features to roles and skill levels—from interactive data visualizations to more advanced capabilities like self- service—so they can query data, explore results, and design their own reports.

And by embedding sophisticated features like predictive analytics, your customers can predict churn, forecast demand, and detect anomalies.

2. Dev Team

An embedded analytics platform lets your dev team stay focused on your core product, since your analytics vendor is focused on the dashboards and reports in your application. Developers should be able to easily embed and white-label the embeddeanalytics, then rapidly iterate and improve on them over time.

Analytic platforms save time with libraries of prebuilt customizable features, including input controls, data visualizations, and interactive data discovery. With a platform, your team can extend analytics features with third-party visualizations and components such as jQuery and D3. They can brand them using CSS, JavaScript, and embedded APIs. And they can programmatically address workflow/processing.

3. DevOps Team

A DevOps-friendly embedded analytics solution will make deploying features easy. It should work with your existing tech stack without imposing special operational requirements. As you build features on the platform, you should be able to deploy them on standard application servers and scale horizontally by adding servers, virtual machines, or containers.

Look for a solution that's fully compatible with on-premises, cloud, or hybrid architectures. Analytics should work with any number and mix of data sources; use your existing security framework so there is no replication; and be mobile- ready/ UI responsive.

3 Aligning Dev and Ops

While development and DevOps teams share a goal of frequently releasing standout products, on a day-to-day basis they have very different priorities. To ensure your embedded analytics solution meets everyone's needs, you'll have to align both perspectives.

Rajit

Rajit leads the effort to embed or enhance the analytics in the company's flagship software.

> "My priority is building a one-of-a-kind product in an innovative way—experimenting when I need to and enjoying the freedom to create what I want."

Mary

Mary is the DevOps manager leading the team responsible for operationalizing software products.

> "My priority is providing a stable, reliable, secure infrastructure that supports predictable, frequent, and repeatable successes."

Focus Area	Rajit's Priorities	How a Dev-Ops-Friendly Embedded Analytics Platform Bridges the Gap	Mary's Priorities
Security	If any analytics capability isn't 100 percent secure, it's a non-starter. Security needs to be easy to implement and give us granular control, so we can set access down to the row or column level, depending on the user.	Security doesn't change. The embedded analytics platform uses the company's existing authentication and authorization frameworks to provide single sign-on access. Any changes the application team makes to user roles and rights are instantly reflected in the analytics. Database access can be expanded or rolled out to new groups in minutes.	Ensure security of all infrastructure and product levels and deployments, including cloud multi-tenant implementations, Avoid multiple overlapping security methods that are difficult to maintain and could allow "cracks" to open during faster, more frequent release cycles.
Capabilities	My team needs a rich set of flexible tools so they can build unique dashboards and reports that match the rest of our application. The platform should extend to meet our needs: It shouldn't be an issue if we want to add new features like predictive analytics, support for multiple data sources, database write-back, or self-service capabilities.	The embedded analytics platform supports the latest features without adding burden to DevOps. Every capability deploys via methods Mary's team is already using. These capabilities can run on Linux and Windows OS, .NET and Java platforms, on-premises or in the cloud, including multi- tenant deployments, in container and virtual architectures.	A single unnecessary capability could put extra demand on my team and bog down the application for our users. Avoid adding software tools and capabilities that require separate servers, proprietary data structures, or any other infrastructure elements my team has to install, learn, and maintain. And be careful about adding new features just "because we can."
Extensibility	The platform needs to be fully extensible, so it's easy to add new capabilities as needed. It should address any analytics or integration need by leveraging third-party visualizations, plug-ins, APIs, and components like jQuery and D3.	A fully extensible platform supports sustainable, streamlined product delivery lifecycles. Watch out for elements that work in a proprietary way, forcing your team to learn a whole new skillset or system.	Everything should work with our preferred tech stack. To maximize performance levels and keep long- term maintenance at a minimum, the analytics platform must work with our existing infrastructure.

4 5 Elements of a DevOps-Friendly Embedded Analytics Solution

When embedded analytics providers say their platform and tools will work with your tech stack and are DevOps- friendly, what do they really mean?

> **Consider these five factors when evaluating embedded analytics solutions with DevOps in mind.**

1. The embedded analytics solution integrates with your existing security model

If the solution is DevOps-friendly, it will continue handling security with the same methods your DevOps team is currently using. You won't have to recreate or replicate security information in two different places. The embedded analytics solution will make it easy to partition multi-tenant data, limit access to individual records in your data, and secure different parts of the application, workflow, and capabilities.

The rising trend of DevSecOps teams[3] is an indicator of how essential security is to DevOps teams everywhere.

Security Considerations for DevOps:

> ❯ **Authorization Controls**
> DevOps has no desire to change authorization methods for controlling database access. Whatever they're currently using—credentials, enterprise services account, or something else—the embedded analytics platform should adapt to it.

> ❯ **Standard-Based SSO**
> It's important for user experience that embedded analytics applications play within a single sign-on (SSO) environment—and important for DevOps that they do it based on standards. Any custom elements divert DevOps resources to figuring out SSO peculiarities and decrease the odds of being able to quickly debug any problems.

3 https://www2.deloitte.com/insights/us/en/focus/tech-trends/2019/embedding-security-devops-pipelines-devsecops.html

> ### Defense Against Known Vulnerabilities

Look for an embedded analytics vendor who is already protecting against common security vulnerabilities that any web application will face. Presumably, if your developers are already creating web applications, then your DevOps team is well-versed in the vulnerabilities (such as cross-site forgery) and how to take measures to mitigate them. If your solution provider isn't already defending against known vulnerabilities, someone on your DevOps team will have to "go to school" with the vendor—or at least ask some probing technical questions to understand how applications could be compromised and what additional protections must be put in place.

> ### Incident Visibility

If there is a security incident, does the platform provide detailed event logging? DevOps should be able to quickly find the Who, What, and When of everything going on at the time.

2. The embedded analytics solution will access your data in-place

A DevOps-friendly embedded analytics solution will not force you to use a proprietary data store, replicate data, or change your current data schemas. You can have data stores in diverse environments (relational, big data, web services, proprietary). Applications incorporating analytics should be able to use data in your existing systems and deliver strong performance across disparate sources.

For instance, consider a software provider that has over 70 enterprise divisions, each maintaining its own database on different systems. If the company chooses a DevOps-friendly embedded analytics solution, they'll have no problem combining and presenting data from all these sources in a single visual reporting environment.

Data Considerations for DevOps:

> ### Self-Service Capabilities

When applications provide end users with a managed set of data analysis features, their developers define the queries. But that's not the case for self- service applications that allow end users to do their own interactive data discovery. DevOps needs to know those applications will be generating standard SQL queries.

> **Query Performance**
> DevOps should be able to optimize the data separately from optimizing the application as a whole—which can only be done if the embedded analytics solution is accessing data in your current systems. Your application and the data in it should be treated as two distinct components that work together, so you have the option to either optimize them together or separately.

> **Business Logic**
> Where do you want to ensconce the critical business logic that defines key performance measures? By keeping that logic at the data tier, any application that accesses the data will have consistent results through the organization. If data is replicated or placed in proprietary structures, DevOps must ensure that the business logic is carried through everywhere data is used, adding extra work and unnecessary business risk.

3. The embedded analytics solution will work on clouds, in containers, and across mixed and changing deployment environments

DevOps wants assurance that giving developers the means to embed analytics isn't going to create a snag in the way the organization currently deploys applications or limit deployment options going forward. That means applications work on Windows or Linux; on-premises, cloud (Amazon Web Services, Microsoft Azure, Google Cloud Platform), and hybrid architectures; and for any end-user mobile device or browser. It also means applications can be deployed in a container, or parts of them dispersed across multiple containers.
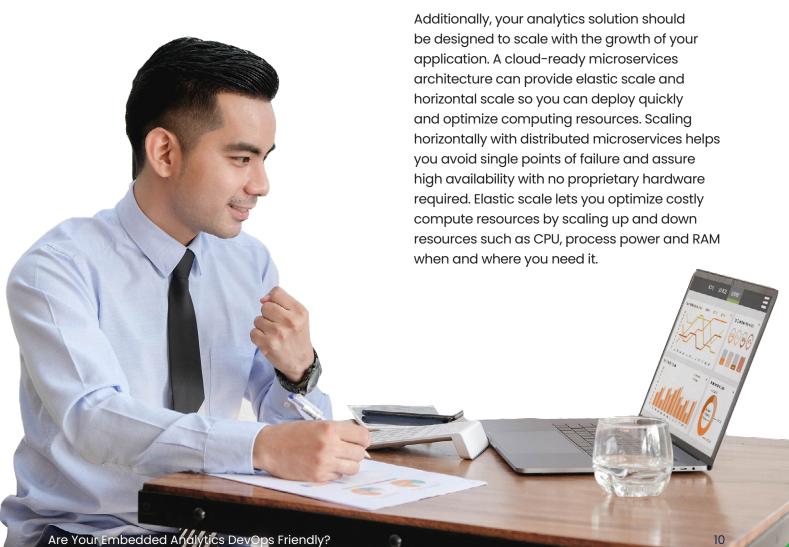
Environment Considerations for DevOps:

› **Deployment & Automation**

Support for container technology (like Docker) and orchestration technology (like Kubernetes), which automates container deployment and scaling, is important because they can facilitate more frequent software releases. Containers help ensure that incremental releases will run reliably when moved from one computing environment to another—through development, testing, and staging into production, be it on a physical or virtual machine.

› **Release Frequency**

Another advantage for increasing release frequency—and moving toward continuous integration and continuous delivery (CI/CD)—is that containers are extremely flexible. Applications are deployed inside containers with everything needed for them to run (libraries, config files, etc.). And unlike virtual machines, containers do not include an OS. Instead, they share the OS kernel with other containers. As a result, developers can build applications in smaller modules, which can be instantiated and removed very quickly.

Additionally, your analytics solution should be designed to scale with the growth of your application. A cloud-ready microservices architecture can provide elastic scale and horizontal scale so you can deploy quickly and optimize computing resources. Scaling horizontally with distributed microservices helps you avoid single points of failure and assure high availability with no proprietary hardware required. Elastic scale lets you optimize costly compute resources by scaling up and down resources such as CPU, process power and RAM when and where you need it.

4. DevOps can deploy the embedded analytics solution into their current architecture in familiar ways

DevOps loves to hear deployment of embedded analytics applications is based on standard methods they already know. Is it as simple as installing an ASP.NET application on an IIS server or a Java application on an Apache Tomcat server? Does the platform make deploying across diverse environments relatively straightforward by minimizing architecture-specific steps?

Architecture Considerations for DevOps:

› **Flexibility for Optimizing Cloud Costs**

DevOps wants the embedded analytics platform to be flexible enough—in both deployment and licensing—for them to make the most cost-effective cloud implementation choices. Depending on cloud pricing (and considering other factors, such as release frequency and agility), it might make more sense to deploy applications to a single large-capacity server or across dozens of containers running on a couple of virtual machines.

› **High Availability Architectures**

Embedded analytics applications should fit nicely beneath the protective umbrella of whatever architectures—load balancing, failover, disaster recovery—are currently used to ensure high availability at the application level. DevOps will also want the freedom to make its own choices for HA clusters, such as sharing user state content via a database or in a centralized file repository on an NAS drive or cloud NAS.

› **Horizontal and Vertical Scaling**

Will it be easy to scale the embedded analytics solution? Is it as simple as multiplying containers or spinning up another application server image in the conventional way? Are there additional licenses to purchase and additional components, like proprietary data stores, that have to be scaled separately?

5. Embedded analytics won't drag release cycles

If you're moving toward continuous integration and continuous delivery (CI/CD), you want to control exactly what gets deployed and when through your build pipelines. Does the embedded analytics platform allow you to decouple application logic from the analytics engine and supporting files so that DevOps can deploy smaller size incremental update packages when broader changes aren't needed? Will DevOps be able to perform these updates using distribution version control systems like Git and open source automation servers like Jenkins or Chocolatey?

Release Considerations for DevOps:

> **Invisible Extensibility**

Can you do a major software enhancement, such as adding self-service capabilities, by deploying self-contained components to the existing application server—or will you have to roll the whole application out just for one new piece? Will DevOps be able to use standards-based tools and mechanisms for orchestrating them? Will the process be invisible to end users—apart from the availability of new features?

> **Incremental Rollouts**

Can you pilot enhancements like self-serve analytics with a small group of end users, then seamlessly roll out to additional users? You should have control to turn specific features on or off, making them visible to only specified users (as access authentication continues to be controlled by your current database security frameworks). There should be no need to touch the databases or reconfigure SSO.

> **Performance Analysis**

Some embedded application platforms provide standalone performance monitoring tools. DevOps may prefer these because they're usable out-of-the-box. Another approach is to provide event logging data to standard tools for web application monitoring. The disadvantage is that DevOps has to do the integration, but the advantage is the opportunity to do far more granular monitoring of user activity. And as analytics become increasingly infused throughout software, there's also the advantage of being able to assess performance of all application elements in a unified manner.

5 Sustainable Innovation and Differentiation

Your embedded analytics solution will affect how frequently you release standout software, how competitive you are, and how sustainable your advantage is over time. Look for a solution that:

> Utilizes your existing technology—including security frameworks and tech stack architecture

> Leverages your existing processes so you can build and release application updates faster

> Offers flexible scaling so it grows with your business

See how Logi Composer delivers an out-of-the box development experience for embedded analytics that sets your end users, developers, and DevOps for success.

Logi Embedded Analytics: Purpose-Built for Software Teams

Product teams need intuitive analytics and data visualization capabilities in their applications, purpose-built for every users' unique role and skills. Logi's embedded analytics solutions, by insightsoftware, empower you to design and deploy analytics into the fabric of your organization and products. These analytics integrate with your existing workflows and security models providing a seamless experience where anyone can analyze data, share insights, and make informed decisions.

Learn more at **insightsoftware.com/logi-analytics/**

About insightsoftware

insightsoftware is a leading provider of reporting, analytics, and performance management solutions. Over 30,000 organizations worldwide rely on us to support business needs in the areas of accounting, finance, operations, supply chain, tax, budgeting, planning, HR, and disclosure management. We enable the Office of the CFO to connect to and make sense of their data in real time so they can proactively drive greater financial intelligence across their organization. Our best-in-class solutions provide customers with increased productivity, visibility, accuracy, and compliance.

insightsoftware

US +1 919 872 7800
UK +44 (0) 845 467 4448
AU +61 2 8985 7777
insightsoftware.com