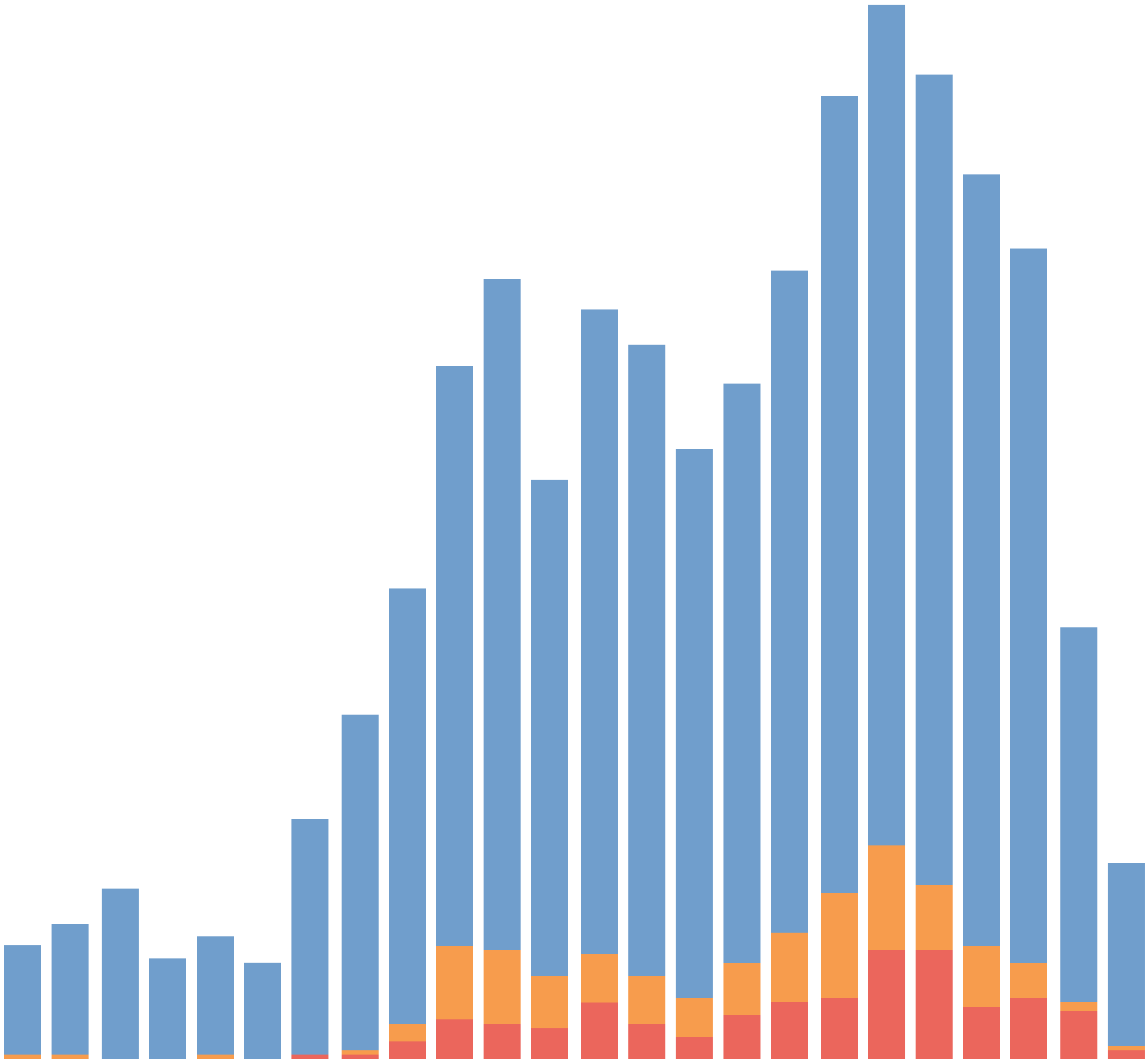


7 Criteria to Consider When Embedding BI

A practical guide for software companies considering whether to "build or buy"

Tri Tu
Embedded BI Consulting Manager

Your application helps customers solve problems. You’re proud of its role in your customers’ operations, problem-solving, or strategic decisions.

Yet you may hear reports from your salespeople. They’ve been losing sales — not because of anything wrong with your application. It’s from something wrong with how customers use the application’s data. Your competition has rolled out whiz-bang new visual analytics. They may call it data visualization, reporting, or dashboards, but the obvious features are shiny dials and colors. You know that their glimmering new analytics aren’t very useful, and that within a short time the thrill will fade. Or perhaps their new displays really are useful, and fast, too. How can your sales people win with analytics and reports that are old, slow, and limited?

Your decision to add new visual analytics, dashboards and reports is simple compared with the next decision: buy or build? Independent software vendors face this choice all the time, of course, and there are traditional criteria.

However, there’s nothing traditional about embedding visual analytics and business intelligence. This is a fast evolving discipline with high stakes. Done well, visual analytics help your customers start productive conversations based on the software you supply. Done poorly, visual displays reflect badly on your company and suggest an inability to keep up.

Though there’s no simple answer to “buy or build,” there are criteria — seven, in fact, that can guide you to make a well informed decision. This paper helps you identify and evaluate each one.

1. Provide compelling visual analytics, dashboards and reports that end-users use and enjoy.
2. Seek fast time to market - speed of development is critical.
3. Deliver so that deployment is easy.
4. Leverage your company’s domain expertise.
5. Look for easy integration into your platform.
6. Ensure data compatibility; no ETL for your data.
7. Weigh the financial commitment against the benefits.

1. Provide compelling visual analytics, dashboards and reports that end-users use and enjoy.

Imagine the big moment when you've released a new version of your software, and all the wrapping drops off. One by one, your customers open your application and see data light up in the new visual analytics, dashboards and reports. How do they feel about what they see, and what do they think?

You want them to see new value. You want them to see in your product effective and beautiful analytics. And behind the data, you want to prove your thought leadership — different than anything the competition delivers.

If instead your analytics interface is old and inadequate, the power of your application can go unseen. Insights and value you could have delivered are left behind. Fairly or not, your customers tend to judge data by its display.

The right display is a truly potent tool. A competent analytics tool is far more than just a report writer or chart builder. It's based on proven principles of cognitive psychology and usability, simple enough to let data tell its story without interference, and flexible enough to answer any question on the fly. — It all adds up to one experience: "Wow."

Wow, users say, this thing works. Wow, I don't need to explain because it's clear as is. The people I show this to get it and don't feel dumb. I didn't have anything to talk about with numbers in columns and rows; now we have something to talk about. This thing is beautiful. This thing lets me do what I need to do. This thing gives me a conversation starter for a sales call. This thing is the beautiful chart I need for that meeting. This thing will persuade. Wow, this view of the data is great, and I can't do without it.

The need for "wow" is a good reason to buy, not build. The company behind a good visual display has spent years developing their analytical applications. That's years that your team does not have to spend.

The "wow" doesn't keep coming by itself. Only a nimble, extensible application can adapt fast enough to flow with evolving customer demands. When customers have new questions, find other ways to use the data, or imagine tweaks to the dashboard that coax even more out of the data, maturity wins — unlike most in-house applications.

For an in-house team faced with revisions, it will feel like just yesterday that the wrapping fell off their analytics display. Now they must go back to work for adjustments or an overhaul — instead of advancing your core application. Or do you let your analytics take the back seat instead? On the other hand, if you had purchased a flexible application, your team could easily make adjustments, adapting to your constantly changing customer needs.

Resist the flow of changes with caution. Users know when someone has restricted their queries, and it frustrates them. For data analysts, satisfaction occurs when analytical software gets out of the way their stream of questions and the simply let's the data answer.

2. Seek fast time to market - speed of development is critical.

One of the requirements is obvious: speed to develop and deploy. Delays become gifts to your competitors and frustration for your customers. So a critical question to ask yourself is how soon do you want to get to market?

Build in-house or not? When considering in-house development, your inevitable question is "Is it worth the wait?"

One of the most widely believed fallacies is that analytical dashboards and reports can be knocked off quickly — over a weekend, perhaps. "Nothing to it" has been more than one software engineer's last words on a Friday afternoon only to return Monday morning with a mess. In fact, designing, building, and implementing user-friendly embedded analytics, data visualization, dashboards and BI solutions is deceptively difficult.

Hidden costs abound. The mess mounts up as days go by. Specifications from the product manager have been misunderstood. Also, the product manager discovers a tiny compromise made for the sake of coding ease and rejects it. The engineer answers a request for another kind of query with, "The application isn't set up for that." Requirements evolve. Gradually, over a long chain of specs and prototypes, meetings and emails, the gap between needs and implementation narrows to a point where everyone can live with the new product. It rarely comes close to the original goal.

Teams without deep expertise in BI and visual analytics invent as they go, creating expense, distraction, and worry. If you do have access to in-house engineers that know visual analytics, then in-house development might be the right option for you. But few companies have such knowledge in-house.

And what about intellectual property concerns? Some ISVs prefer to bear development costs to ensure ownership of the intellectual property. It's a proposition worth careful thought. How much is the IP worth? Will the value of whatever your team develops endure long enough and ensure a strong enough competitive edge to pay off all costs?

The speediest option by far is partnering with a specialized provider — one whose people live and breathe the rapidly advancing field of analytics and data visualization. A supple, competitive analytics solution is rooted in research and developed over years.

3. Deliver so that deployment is easy.

Is deployment easy? Or, like the birthday present marked "assembly required" just under the wrapping, does post-purchase data alignment and display design make implementation a nightmare?

Just connecting to data can be a challenge. An application that came to life in-house can be shot down by real-world data. How many customers will happily beta test your software for you?

Purchased applications can have their own problems, of course. If you buy, make sure the solution requires no rehosting, no data movement, no remaking of data structures, no star schemas, and no special analytic databases. In fact, try to find a solution that can draw from just about any database of any size or variety. Make sure that the solution integrates easily with your infrastructure — whether than means embedding or a standalone option, whether you're a thick client or thin. Make sure the software is as ready-to-run as part of your solution. Why limit yourself or your customers?

4. Leverage your company's domain expertise

You shop for food in a grocery store, have your car fixed by a mechanic, and get your hair cut by a stylist. In the modern world, every business has its own collection of specialties. If you must have the best, you go to a specialist with up-to-date knowledge and reasonable prices. Isn't that why your customers come to you?

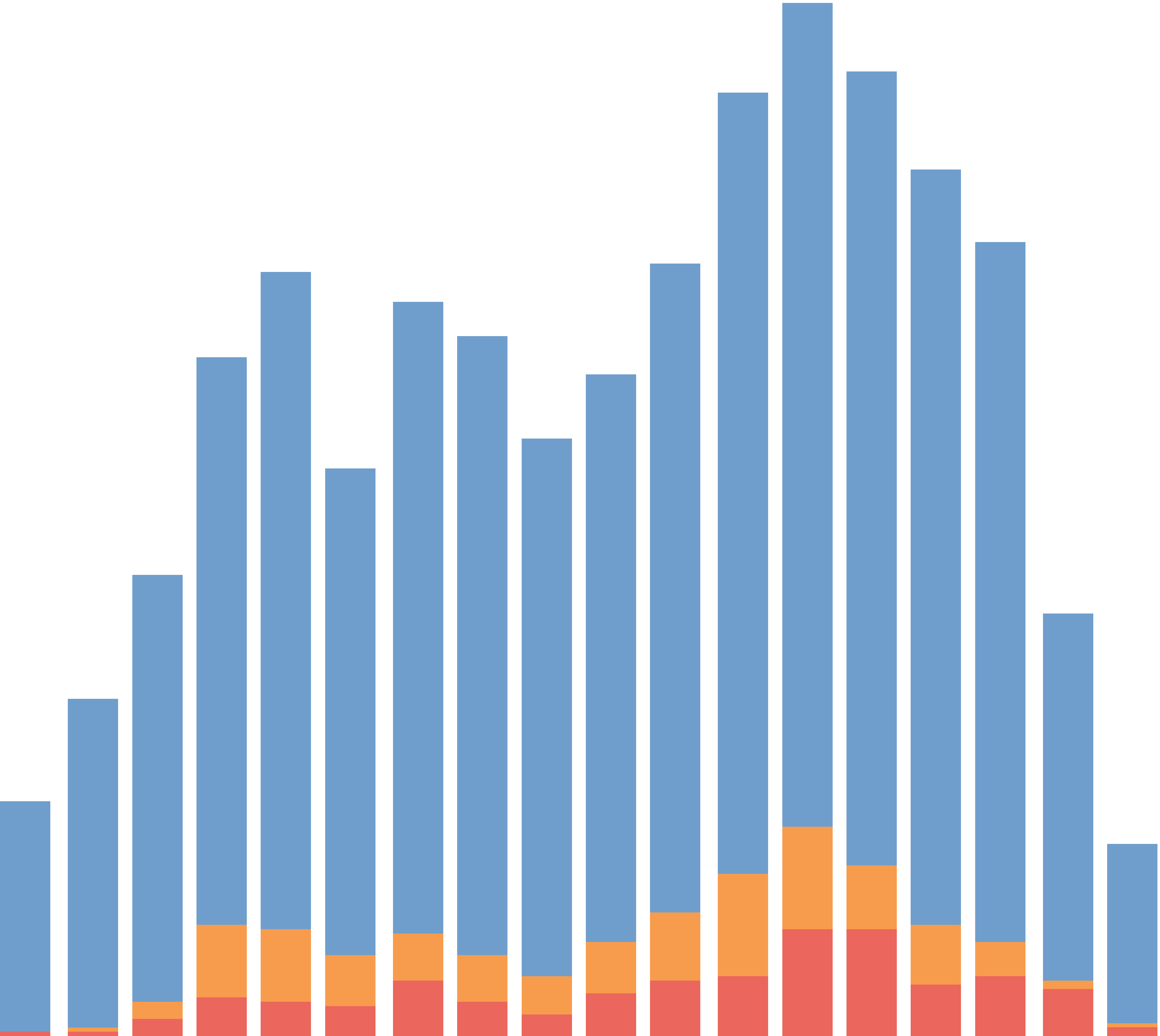
Imagine that your unique value is in software that manages accounting transactions and financial data. Its development isn't a simple matter. Far from it, since your staff works in this particular area of software development every day, and has focused themselves on it for years. You're proud of the product, and rightfully so.

Then one day you want them to write expert BI, analytics, and data visualization software. It's not realistic, even if they say at first, "Nothing to it."

Your greater value in this case comes from other staff — those who face your customers. They imagine vividly the displays of data that your customers would find most useful, and they're eager to create them. These staff are the ones who know your customers, know how they use the data you provide, and can create the right analytical displays.

This is how your company's unique expertise can really show itself. Your customer-facing staff can focus on with what they know best. And your software engineers can concentrate on with improving your core application.

> “Tableau did what they’re best at, which helped us do what we’re best at.”
>
> - ANKUR JETHI, LEAD DEVELOPER



5. Look for easy integration into your platform.

Ultimately, whether you decide to build or buy, you want software that lets your customer work the way they want to. That is, the tool just gets out of the way and lets users work — without another log-in, without awkward interaction with the core application, and without even an inconsistent look-and-feel.

A mature application is transparent. It lays on top of the core application, without APIs or SDKs, as if it belongs there. It starts to work when an engineer drops a single URL into the code — the way someone embeds a YouTube video into a blog. From there, the analytics application passes directives back and forth easily with the host.

When it comes time for users to share their work, a mature tool shows its integration in the easy access to whatever popular formats they may need. Formats should include PDF, web browsers, PowerPoint, or simply JPEGs ready to copy and paste.

6. Ensure data compatibility; no ETL for your data.

Will your customers be able to read and analyze your data immediately? Or will they have to wait for support staff to process the data? Few frustrations are more wasteful for support staff or more painful for users.

Suddenly, users feel that they've lost control. They've had to interrupt their work, call for support, and wait — when they really don't feel that they should have to wait at all. Why can't the software just read the data — any data — as is?

The pain endures even when data transformation is part of the routine, such as when you refresh data marts. Inevitably, there's a gap. The process creates differences between the live transaction records and the transformed data — prompting awkward questions. One user asks another, "It says X over here and Y over there. Which one is correct?" There's confusion, awkwardness, pain, and blame.

It makes far more sense to ensure that the embedded analytics software can read your application's data natively — with no transformation, no proprietary formatting, and no delay. Data read natively can be used instantly for up-to-the-minute analysis of business transactions. Analytical tools that read any data as is let users get on with their work, which is all they really want to do.

7. Weigh the financial commitment against the benefits.

Running underneath most of these criteria has been a common decision: where to invest for the most return? There are several factors. First, you must figure the costs, direct and indirect, over the software lifecycle — typically seven or eight years.

Starting on the first day, in-house development of an analytics application has opportunity cost for your customer-facing experts: they must imagine and describe in detail a kind of software they don't know well. Worse, they must re-imagine and re-describe each time the engineers finish an iteration and offer new tradeoffs or require new compromises. The opportunity cost here is the difference in value between this unfamiliar task and the value of more familiar tasks.

In comparison, these experts produce much higher value when they have the direct control they really want. A well designed, mature data visualization application turns the ordeal of remote-control development into a flow of hands-on expertise. The new analytics displays then come from the full depth of your staff's expertise — turning opportunity cost to zero.

A similar equation works for your software developers. When your customer-facing experts design their own charts, engineers are again free to do what they do best — which again turns opportunity cost to zero.

During development, you have another opportunity cost. Your competition's whiz-bang display continues to show up. Your salespeople assure prospects about your big new improvement, hoping they wait to see it. But how many will actually wait?

Roll-out only begins the next phase of costs. Nothing ever stays the same, especially when technology is concerned. The shape of your data may change, the needs of your customers may change, and your competition will certainly change. Any of that change calls for redesign and re-engineering — and more cost for your engineers, customer, and staff.

In fact, most software costs occur after implementation. A rigorous lifecycle analysis that realistically estimates ongoing maintenance by in-house developers often tips the balance in favor of buying.

All of these costs must be compared with the out-of-pocket for a purchased analytics application.

Case Study: Omega Legal Systems

About two years ago, Omega Legal Systems was hearing ominous reports from their sales people: the competition had been demonstrating new, colorful dashboards — and they were winning customers away.

Since 1975, Omega has been providing top quality, leading edge business intelligence software to mid- and large-size legal firms. They could not let the competition take the lead on such an important aspect of their service.

Competition's flash fizzles How good were the competition's displays? Product manager Mary Steele evaluated. "Their dials and gauges and traffic signals were flashy," she said, "but there was really nothing meaningful being delivered."

Her research took her to articles by Stephen Few, a leading expert in visual display of business data. He mentioned Tableau repeatedly. Steele downloaded a free trial from the Tableau site and within hours emailed lead developer Ankur Jethi. She wrote, "You've got to see this."

Jethi downloaded his own Tableau demo and stayed up overnight experimenting with it. By the next morning, he had concluded that Omega's in-house dashboard project was "dead in the water." He knew they could never match Tableau.

Omega entered into an OEM agreement within three months. Within another six months, Omega released Visual Express, its new product based on Tableau.

The ease of working with the Tableau development team makes Jethi recall it as "a unique experience." He said, "Tableau did what they're best at, which helped us do what we're best at."

Omega first created a stand-alone application with workbooks designed for specific users. Many of these users are executives or others who start with a macro-level focus and often drill down. In one view, for example, a high-level manager can see at a glance the firm's most important metrics, such as outstanding invoices and unbilled work — all without useless, distracting graphics and all on a single screen. One click can take it one level down to more detail. Omega also used Tableau to allow users to make better use of reports they're already familiar with.

Amazing "The response has really been amazing," says Steele. "Clients and prospects really get that this is different. There's no flash and mirrors. Our BI solution really delivers at a fraction of the cost of our competitors. As a result, we've seen a 200 percent increase in sales over the past two years."

Now Omega sales people come back with much different stories. Steele says, "Our competitors' clients are asking them when they'll catch up with us."

Conclusion

There is no simple answer to “buy or build”; it depends on your organization’s current situation, priorities and resources.

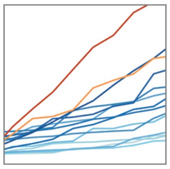
There are seven factors to consider in evaluating your situation:

- Which one of your options deploys fastest?
- With the most ease?
- Which one doesn’t distract your company from its domain expertise but instead builds on it?
- Which meshes with your software as if it were all one piece?
- Which reads any data your customer can throw at it?
- What makes most sense as an investment?

In certain cases it may be better to build your own solution in-house. But for ISVs who want fast time to market with a high quality product that deploys quickly, making use of an existing tool with a satisfied customer base is often the best answer.

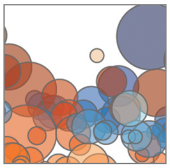
About Tableau

Tableau Software helps people see and understand data. Offering a revolutionary new approach to business intelligence, Tableau allows you to quickly connect, visualize, and share data with a seamless experience from the PC to the iPad. Create and publish dashboards and share them with colleagues, partners, or customers—no programming skills required. See how Tableau can help your organization by starting your free trial at tableau.com/trial.



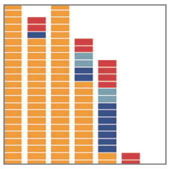
Additional Resources

- Tableau Embedded Analytics

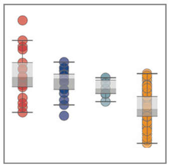


Related Content

- Embedding Tableau: Buy VS. Build Whitepaper
- LeanKit is reducing costs while delivering more value
- Pro sports teams gain customer insights with KORE



Explore Other Resources



- Webinar Series
- Embedded BI Training & Tutorials
- OEM Partner Community