How to Select an Embedded Analytics Product

Twelve Criteria to Consider

By Wayne Eckerson

About the Author

Wayne W. Eckerson has been a thought leader in the data analytics field since the early 1990s. He is a sought-after consultant, noted speaker, and expert educator who thinks critically, writes clearly, and presents persuasively about complex topics. Eckerson has conducted many groundbreaking research studies, chaired numerous conferences, written two widely read books on performance dashboards and analytics, and consulted on BI, analytics, and data management topics for numerous organizations. Eckerson is the founder and principal consultant of Eckerson Group.

About Eckerson Group

Eckerson Group helps organizations get more value from their data. We advise business and technical leaders on how to drive better business outcomes using data and analytics. Our experts each have more than 25 years of experience in the field and are uniquely qualified in every facet of data analytics, from creating data strategies and centers of excellence to building modern data architectures and data governance programs. Get more value from your data. Put an expert on your side—learn what Eckerson Group can do for you!

About This Report

To conduct research for this report, Eckerson Group interviewed numerous industry experts and viewed a dozen or more demos of embedded analytics tools and soffware. The research for this report is made possible by **Yellowfin**.

Table of Contents

Executive Summary

As we move from an information economy to an insight economy, analytics becomes paramount. Every organization, whether it's a business enterprise or a commercial software vendor, needs to embed analytics into core applications that business people use regularly. But until recently, embedding analytics has been a hit-or-miss endeavor—mostly miss.

Today, most independent software vendors and enterprises opt to embed commercial business intelligence (BI) tools into applications. Sometimes they simply select the tool that already serves as the company's BI standard. Although it seems convenient, this may wreak havoc. Most BI tools were not designed for embedding; converting a stand-alone, commercial product into one that can be easily embedded in both single- and multi-tenant environments with full fidelity is challenging.

Although it's easy to discern differences among BI tools when it comes to visualization, authoring, and administrative capabilities, it's difficult to do the same for embedded features. That's because the major differences show up at the code level. Only a developer can really tell how easily and quickly a tool may be embedded and how well it will scale. Pricing is also tricky, since traditional user-based models don't work well when usage can't be predicted.

This report is designed to help readers select an embedded analytics product. It provides 12 criteria to consider when evaluating embedded capabilities. It also comes with a companion spreadsheet that enables users to score and compare products along these 12 dimensions and others they may add.

Key Takeaways

- Independent software vendors (ISVs) are most likely to embed analytics into their products, but enterprises are catching up as they digitize core offerings.
- Most companies are opting to embed commercial BI products rather than build their own reporting and analytics capabilities.
- There are multiple levels of embedding; different tools support different levels.
- There are three key differentiators among embedded BI products: speed of deployment, amount of code that must be written, and the developer mind-set of the vendor.
- There are 12 major criteria to consider when evaluating embedded analytics products.

Recommendations

- Weigh the benefits of building versus buying analytics capabilities. Although it's tempting to build them, it's better in the long run to embed a commercial analytics product.
- Define your requirements before selecting an embedded analytics product. That is, know who will use the analytics and for what purposes.
- Use these requirements to guide your selection of an embedded analytics tool.
- Use the 12 criteria described in this report to guide your discussions with prospective BI vendors.

Embedded Analytics for Everyone

Here's a paradox for data analytics professionals: ***analytics becomes pervasive when it disappears.***

For decades, business intelligence (BI) and analytics tools have failed to penetrate more than 25% of the average organization. And within that 25%, most workers use the tools only once or twice a week. Embedded analytics changes the equation. By inserting charts, dashboards, and entire authoring and administrative environments inside other applications, embedded analytics empowers business users with insights and dramatically increases BI adoption. The catch is that most business users don't know they're "using BI"—it's just part of the application they already use. The best BI tools are invisible.

By placing analytics at the point of need—inside operational or cloud applications—embedded analytics closes the last mile of BI. Workers can see the impact of past actions and know how to respond to current issues without switching applications or context. Analytics becomes indispensable for managing core processes and solving problems. As a result, embedded analytics has a much higher rate of adoption than traditional BI or analytics.

Target Organizations

Embedded analytics makes existing applications more valuable for every organization. Independent software vendors (ISVs) report that embedded analytics increases the value of their applications by 43% and enables them to charge 25% more on average.[1] Enterprise organizations embed analytics into operational applications such as Salesforce.com and intranet portals such as SharePoint. In both cases, embedded analytics puts data and insights at users' fingertips when they need it most—both to gain insights and take action.

[1] "The Definitive Guide to Embedded Analytics," Logi Analytics, 2017–18.

ISV requirements. ISVs must ensure an embedded product looks and feels like the host application, and thus require high levels of customization and extensibility. Cloud ISVs also require embedded products that work in multi-tenant environments, with seamless user administration and custom deployments. Many ISVs offer tiered pricing, which requires that embedded products offer flexible user provisioning. Finally, because ISVs can't always estimate how many customers will purchase analytics add-ons, they need value-based pricing models that don't become prohibitively costly as usage ramps up.

Enterprise requirements. Traditional enterprises have fewer requirements than ISVs, but that is changing. More companies are pursuing digital strategies that require customer-facing Web applications. And although most don't charge for analytics, as ISVs do, many view data analytics as a key part of the online customer experience. For example, mutual funds now provide customers with interactive dashboards where they can slice and dice investment portfolios and take actions such as buying and selling funds. Their requirements for customization, extensibility, multi-tenancy, and security are growing.

Build or Buy?

Once organizations decide to embed analytics, they need to make a few key decisions. The first is whether to build their own analytics capabilities or buy a commercial, off-the-shelf tool.

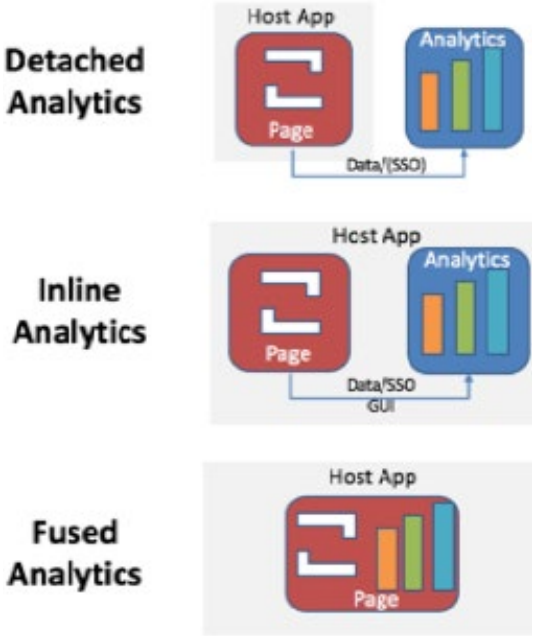
Build. Organizations with internal developers are tempted to build their own analytics capabilities. But unless the analytics are simple and requests for changes are minimal, it's always smart to outsource analytics to a commercial BI vendor. Commercial analytics products deliver best-of-breed functionality that would take in-house developers years to develop, diverting them from working on the host application.

Buy. Recognizing the opportunity for embedding, most BI vendors have made their tools more configurable, customizable, and integrate-able with host applications. Most seek to make their tools easy to customize without coding so they look, feel, and act just like the host application. For example, most embedded customers want to "white label" analytics tools to have the same colors, styles, and layout as the host application and be indistinguishable from it. When more extensive customization or integration is required, customers can use application programming interfaces (APIs) to add new functionality, such as new chart types and data connectors, or fine-tune the way tools integrate with the host application.

Types of Embedding

The second decision is to determine an embedded analytics architecture. There are three primary approaches: detached, inline, or fused (see figure 1).

Figure 1. Three Types of Embedding



Detached analytics. This is a lightweight form of embedding where two applications—host and analytics—run separately but are tightly linked via URLs. This approach works well when multiple applications use the same analytics environment, portal, or service. A common example is Google Analytics, a commercial service that multiple groups inside an organization might use to track Web traffic on various internal websites. There is no commonality between the host application and analytics tool except for a shared URL and shared data. The two applications might also share a common authentication mechanism to facilitate single sign-on (SSO).

Inline analytics. With inline analytics, the analytics tool is part of the host application. It looks, feels, and acts like the host, but runs as a separate element, tab, or module within it. For example, a newspaper might embed a chart within the text of an article on a webpage. Or an ERP application might present users with a dashboard upon log-in that displays summary activity from each module in the application. Or there might be a separate tab where customers can view analytics about their activity within the application. In most cases, the embedded components sit within an iFrame, which is an HTML container that runs inside a webpage.

Fused analytics. Fused analytics delivers the tightest level of integration with a host application. Here, the analytics (e.g., a chart, table, or graphical component) sit side by side with the host application components and communicate bidirectionally with them. For example, an inventory manager can view inventory levels in various warehouses and place replenishment orders without leaving the screen. Or a retail store manager can view daily demand forecasts and then click a button to create the shift schedule for the following week. Fused analytics is facilitated by JavaScript libraries that control front-end displays and REST API calls that activate server functions.

Technology Implications

Embed code. Most analytics vendors can support inline analytics without much difficulty. They simply provide embed code—a snippet of HTML and JavaScript—that administrators can insert into the HTML code of a webpage. The embed code calls the analytics application to display specified content within an iFrame on the webpage. People who use YouTube and

other social media services are familiar with embed code. iFrames are a quick and easy way to embed third-party content, and most analytics vendors support them.

iFrames. But iFrames have disadvantages. Because they are frames or windows inside a webpage that are controlled by an external application or service, they pose considerable security risks. Also, they operate independently of the host webpage or application—the host can't manipulate what's inside the iFrame, and vice versa. For example, hitting the Back button doesn't change what's inside the iFrame. Furthermore, iFrames behave differently depending on the browser, making them difficult to manage. Consequently, a growing number of organizations refuse to allow iFrames. In response, some analytics vendors have revamped the way iFrames work to overcome most of their limitations.[2]

Fused analytics requires tighter integration between host and application than most iFrames can support. Today, this is often achieved through a JavaScript library that customers download and run inside the host application. Fused analytics requires a much greater degree of customization, extensibility, flexibility, and integration than many analytics vendors support out of the box. Companies that want to fuse analytics into an application need to look under the covers of an analytics tool to discover its true embeddability.

Selecting an Embedded Analytics Product

Selecting an embedded analytics product is not easy. You might select a product with great visualization capabilities that are ideal for your application, but only later recognize that it is difficult to white label, customize, or integrate with your host application.

There are many pitfalls. For example, selecting a product that doesn't include a key feature—such as print or export to PDF—can undermine adoption and imperil a project. Or maybe the product doesn't work seamlessly in a multi-tenant environment, making administration cumbersome and time-consuming and contributing to errors that undermine customer satisfaction. Or an embedded deployment might drag out for weeks or months because most customizations require custom coding.

This report is designed to help you select the right embedded analytics product. Whether you are an independent software vendor that needs to know how to embed analytics in a commercial, multi-tenant cloud application or the vice president of application development at a major corporation who wants to enhance a homegrown application, this report will guide your selection of the right analytics product.

[2] These vendors often see these revisions as a stopgap measure until they can replace their iFrame technology with a JavaScript solution.

The report comes with a companion spreadsheet that lets you score and rank multiple vendors against the criteria described in this report, plus any others you might add. The spreadsheet is a handy tool to help you select a short list or choose a finalist.

Create Evaluation Criteria

We've talked with dozens of vendors that offer embedded analytics products. Each has strengths and weaknesses. Analyst firms such as Gartner and Forrester conduct annual evaluations of BI tools, some of which are publicly available on vendor websites. G2 provides crowdsourced research on BI tools, while the German research firm BARC publishes a hybrid report that combines analyst opinions and crowdsourced evaluations. However, these reports generally don't evaluate features germane to embedded analytics. That's because the differentiators are subtle and often hard to evaluate, since it requires diving into the code.

The differentiators among embedded analytics are subtle and often hard to evaluate because it requires diving into the code.

Key Differentiators

For companies that want to tightly integrate analytics with a host application, there are three key things to look for:

1. How quickly can you deploy a highly customized solution?
2. How scalable is the solution?
3. Does the vendor have a developer mind-set?

1. **Deployment speed.** It's easy to deploy an embedded solution that requires minimal customization. Simply replace the vendor logo with yours, change the font styles and colors, copy the embed code into your webpage, and you're done. But if you want a solution that has a truly custom look and feel (i.e., white labeling), with custom actions (e.g., webhooks and updates), unique data sources and export formats, and that works seamlessly in a multi-tenant environment, then you need an analytics tool designed from the ground up for embedding.

The best tools not only provide rich customization and extensibility, but also do so with minimal coding.

The best tools not only provide rich customization and extensibility, but also do so with minimal coding. Every element can be custom-tailored using a point-and-click properties

editor. Templates, themes, and wizards simplify development and customization. And when customers want to go beyond what can be configured out of the box, the tools can be easily extended via application programming interfaces (APIs).

2. **Scalability.** It's important to understand the scalability of an analytics solution, especially in a commercial software setting where usage can skyrocket. The tool needs strong systems administration capabilities, such as the ability to run on clusters and support load balancing. It also needs a scalable database—whether its own or a third party's—that delivers consistent query performance as the number of concurrent users climbs and data volumes grow. Many vendors now offer in-memory databases or caches to keep frequently queried data in memory to accelerate performance. The software must also be designed efficiently with a modern architecture that supports microservices and a granular API. Ideally, it works in a multi-tenant cloud environment where processing can scale seamlessly on demand.

3. **Developer mind-set.** When developers need to get involved, it's imperative that an analytics tool is geared to their needs. How well is the API documented? Can developers use their own integrated development environment, or must they learn a new development tool? Can the tool run on its own application server or does it require a proprietary application server and database? How modern is the tool's software architecture? Was it built using its own APIs? Does it have a front-end environment built using a JavaScript framework, such as Angular JS, that communicates with a back-end Java environment via REST calls that return JSON data?

Companies are adopting modern software architectures and don't want to pollute them with monolithic, proprietary software from third parties.

Increasingly, companies are adopting modern software architectures and don't want to pollute them with monolithic, proprietary software from third parties. The embedded analytics solutions of the future will insert seamlessly into host code running on host applications and Web servers, not proprietary servers and software.

Twelve Evaluation Criteria

Although we have yet to conduct an exhaustive analysis of products, we have done enough research to know what questions to ask vendors to identify their key differentiators and weak spots. Below are the top 12 criteria to evaluate when selecting an embedded analytics product. (Download our companion spreadsheet to score and compare products using these and your own criteria.)

These criteria apply to both ISVs and enterprises, although some are more pertinent to one or the other. For instance, customization, extensibility, multi-tenancy, and pricing/packaging are very important for ISVs; less so for enterprises.

1. **Embedding.** What parts of the analytics tool can you embed, and which can you not? The best embedded analytics tools let you embed everything—including mobile usage, authoring, and administration. Are objects embedded via iFrames (i.e., inline) or JavaScript (i.e., fused)?

   Here are more specific questions to ask prospective vendors:

   a. **What parts of the analytics tool can you embed, and which can you not?** The best embedded analytics tools let you embed everything—charts, dashboards, authoring, collaboration, data preparation, data modeling, database, and administration. Some very popular analytics tools, such as Tableau Software and Power BI, still have a desktop dependency, meaning you can't use them to embed a self-service analytics environment inside an application. The same is true for mobile applications that require users to download an app from an app store.

   b. **How are the elements embedded?** Are objects embedded via iFrames or JavaScript libraries with REST API calls? Some vendors disguise iFrames inside JavaScript embed code, and some have overcome many of the deficiencies of iFrame technology.

   c. **Where can the embedded application run?** Embedded code should display everywhere the host application runs, including on the Web, desktop, cloud, and mobile applications as well as edge devices. However, iFrames may corrupt the display on small screens. If users must download a native mobile application from an external app store, this may nullify the benefits of embedding the analytics tool.

   d. **Does the analytics tool integrate out of the box with popular packaged applications?** Many customers want tools that embed quickly into Salesforce or SharePoint. Some tools have a wizard to accelerate this integration.

2. **White labeling.** What parts of the user interface can you customize without coding? The best tools let you create a custom graphical interface that blends seamlessly with the host application without developer assistance. The less coding, the quicker the project deploys.

   Here are some questions to ask:

   a. **What parts of the tool can you customize without coding?** The best tools offer a properties editor for every object or element in the tool, including charts, tables, text labels, controls (e.g., panels, menus, buttons, filters), headers/footers, borders, backgrounds, icons, and layouts. The editor is a point-and-click interface that non-developers can use to configure and style any element, including changing colors, fonts, borders,

and actions (e.g., click, hover, enter). Most tools let you package styles for various elements into a theme. Customers can apply themes globally or to individual reports, groups of users, and roles.

b. **How do you customize elements that can't be configured?** Most tools provide a CSS configuration file that developers can modify to customize the style of different graphical display elements. For instance, developers can add custom animations to a graphical element, special fonts not supported by the product, or tree-like navigation structures that expand/collapse upon click. Interestingly, some vendors require custom CSS scripting to replace a logo, since most customers want to customize the placement and appearance.

3. **Extensibility.** Extensibility is important when customers want to add new functionality to an analytics tool. A good tool provides one or more mechanisms for extending the functionality of a product (e.g., adding new charts, utilities, or calculations). Ideally, the mechanism, such as an API, plug-in library, or software development kit, will minimize the amount of coding and create a marketplace of extensions that are accessible within the tool, such as third-party visualizations inside a chart library.

   A litmus test of extensibility is the quality of the vendor's API documentation. It should conform to industry standards, clearly describe objects, methods, and properties, and show code samples in a variety of languages.

   Ideally, an embedded product should support the following extensions:

   a. **Custom visualizations.** Customize existing visualizations, create new ones from scratch, or insert third-party visualizations into the tool's charting library.

   b. **Data connectors.** Create new data connectors for unsupported data sources, applications, files, and external data sets.

   c. **Application components.** New navigation panes, menus, filters, selectors, tables, buttons, and so on with corresponding actions.

   d. **Utilities and functions.** Customize existing utilities, such as schedulers, backup mechanisms, export formats, printing, alerts, and dashboard layouts.

   e. **Transformations.** Custom functions to clean, parse, concatenate, or integrate data.

   f. **Formulas.** Custom formulas for calculating metrics.

   g. **Other.** Whatever people can dream up, they can build and integrate.

4. **Data architecture.** The data architecture supplies data to the analytics tool and determines its scalability, performance, and flexibility to address new questions. Ideally, a data architecture is flexible, allowing customers to tailor the architecture to support unique requirements. For instance, many products let customers select whether to query data at its source or import it into an in-memory database to ensure consistent, fast query performance.

   Here are some questions to ask:

   a. **Data sources.** What data sources and applications can the tool query out of the box? Does it support relational, Hadoop, NoSQL, OLAP cubes, cloud applications, and external data sets? Does it support generic connectors (e.g., ODBC, JDBC, OLE DB) or native connectors and application calls?

   b. **Query architecture.** Does the tool query data sources directly or extract and store data locally? Direct queries give real-time data, but might overwhelm the host application. The extract-and-store approach enables customers to pre-integrate and model the data and delivers faster, more consistent performance. Tools that provide both offer the greatest flexibility.

   c. **Semantic layer.** Does the tool require the creation of a semantic layer or business model that users query? Does it create it automatically, or does it let users query the source tables directly? Can a semantic layer query and join data from multiple sources (e.g., query federation)? Semantic layers simplify ad hoc queries, but add overhead during design and runtime.

   d. **Data storage.** Does the tool store data locally? Does it come with its own database or let customers select a database of their choice, both for content data and metadata? Does it offer an in-memory database or a persistent one, or both? Can other applications or services query the database? A separate database improves scalability and offloads queries from the host application, but it must be managed separately. A database from a BI provider might not be as scalable as one from a database specialist.

   e. **Transformations.** Can administrators clean, integrate, and transform data to facilitate queries? Can administrators create aggregates to speed query performance?

5. **Process integration.** This characteristic defines how closely the BI tool is linked to its host application. (See “Types of Embedding” above.) For instance,

the hallmark of fused analytics is that it supports bidirectional communication between the host application and the analytics tool. Most BI tools use iFrames to embed analytic content inside host applications. Traditionally, iFrames are independent windows inside a webpage that have minimal interaction with that page.

Here are questions to ask:

a. **Embedded mode.** What mode(s) of embedding does the product support? Linked, inline, or fused?

b. **Write-backs.** Can the analytics tool update the host application through its API or update its underlying data store?

c. **Webhooks.** Can the analytics tool trigger an action in the host application? For example, if a metric exceeds a threshold, can it issue an alert in the host application?

d. **Navigation.** Can users click on a navigation pane in the host application to change what is displayed by the analytics tool?

6. **Security.** Security governs who can access analytics functionality and data through authentication and access controls. Security can be a major impediment to embedding commercial analytics products.

   Here are some questions to ask:

   a. **Authentication.** Does the tool support single sign-on? Or does it require a separate log-on and security framework? If it has its own framework, is it required or optional? If required, how easy is it to map between security systems? Does it support authentication standards such as SAML, oAuth, and so on?

   b. **Authorization.** Are roles and rights established in the parent application passed to the analytics application to ensure end users are granted the appropriate levels of access?

   c. **Application security.** How robust is the tool's permissions structure? Do administrators have fine-grain control over what content users can see and what functions they can apply?

   d. **Security.** Does the tool control access to data sources, tables, columns, and rows?

7. **Multi-tenancy.** Multi-tenancy enables companies to deploy unique content (e.g., data) and analytic functionality for different customers (e.g., tenants). Although multi-tenancy is a key issue for cloud software providers, it is also becoming a

critical concern for enterprises that want to deploy unique analytic content and capabilities to different internal groups.

Here are some questions to ask:

a. **Tokenization.** Can multiple users or tenants access different views of the same report, dashboard, or data model? Does a change in the master propagate to each tenant so changes don't have to be made individually? Tokenization eliminates the need to create and maintain separate reports or data marts for each tenant.

b. **Tenant isolation.** Can each tenant view its own data, whether it's stored in the same database, separate databases, or separate servers?

c. **Tenant administration.** Can each tenant administer its own instance and users? Can each tenant add, edit, and delete users and groups? Can they assign permissions for content and functionality to each user and group? This is particularly important if the tenant offers tiered pricing for analytics functionality.

d. **Global administration.** Can the host monitor and manage all tenant instances in a single view? Can the host upgrade the analytics tool without breaking customizations applied by each tenant?

8. **Administration.** An embedded application creates an administrative headache, since it is often administered separately from the host application. Administrative complexity is compounded in a multi-tenant environment because companies often want to provide each tenant with the ability to administer different parts of the analytics application—usually the ability to control access to analytic data and functionality.

   Here are some questions to ask:

   a. **Dev/ops.** Does the product support check in/out, version control, and capture audit trails of all activity? How easy is it to migrate code from development to test to production? Can it roll back to a prior version easily? Does it sync with the host product's dev/ops processes and tools?

   b. **User management.** How do administrators manage users and/or tenants? Can they manage them through the same console as the host application?

   c. **Systems monitoring.** How do administrators handle errors, timeouts, non-responsive dashboards, and so on? Can they manage them through the same console as the host application?

d. **Systems management.** How do administrators configure and manage load balancing, backup, restore, and clusters? Can they manage them through the same console as the host application?

e. **Cloud provisioning.** How do administrators configure the product for private or public clouds and elastically scale the product? Can they manage them through the same console as the host application?

f. **Localization.** What languages, currencies, symbols, date formats, and graphical icons does the product support out of the box? Can the localization be customized to support new languages?

9. **Software architecture.** Increasingly, companies want embedded products that conform to the software architecture of their applications. Modern application architecture is Web-based, with a JavaScript front end connected to a Java or .NET back end using a standard API such as REST and JavaScript.

   Here are some questions to ask:

   a. **API design.** Was the tool built using its own API? Does it have a separate front end that uses the API to call functions on the back end?

   b. **Front end.** Does it use JavaScript frameworks rather than pure HTML with JavaScript or iFrames?

   c. **Back end.** Does it generate JSON and respond to REST calls?

10. **Systems architecture.** It's also important to understand the systems footprint of the analytics tool. For instance, does it need a separate server? Database? Metadata repository?

    Here are some questions to ask:

    a. **Systems requirements.** How much RAM does the tool require? What types of hardware are needed to power it? How much disk space is needed for the tool itself, as well as its data and metadata? What kind of hardware is required to power the tool for your workloads?

    b. **Application server.** Does the product require its own proprietary application server? Or can it run as JAR files or DLLs on the host application server or on a separate (non-proprietary) application server, if required for scaling?

    c. **Database server.** Does the product require the installation of a proprietary database? Does it require a database to store metadata and/or application data? Can it work without its own database?

d. **Desktop tool.** Does the product require a desktop tool to author reports and dashboards, design data models, or execute data loads?

e. **Mobile app.** Does it require users to download a mobile app to view content via a smartphone or tablet device?

f. **IDE.** Does the product come with its own integrated development environment, specify a third-party IDE (e.g., Microsoft Visual Studio), or can developers use their own and access the tool's functionality through a plug-in or extension?

g. **Language frameworks.** Does the tool require the use of a language-specific framework, such as .NET, to perform custom development?

11. **Vendor.** It is critical to evaluate a vendor's ability to support its embedded analytics customers. Each BI vendor has a different level of experience in the embedded space. Some BI vendors only sell embedded analytics products; others sell general-purpose BI products; and others sell many products in addition to BI tools. Most BI vendors see the embedded space as a key growth area and are adding functionality, adapting pricing, and fine-tuning support to meet the unique needs of embedded analytics customers.

    Here are some questions to ask:

    a. **Ratio.** What is the vendor's ratio of original equipment manufacturer (OEM) customers to regular customers? How many OEM customers in total?

    b. **Viability.** How financially stable is the vendor? What are its annual revenues? How many customers does it have? How many years has it been in the BI market? What are its cash reserves and investment history?

    c. **Pricing.** For ISVs, usage-based pricing based on number of users, queries, or CPU cycles is hard to estimate and could generate large, unplanned expenditures. The vendor needs to offer flexible pricing and work with each customer based on their requirements. Ideally, customers can choose between perpetual and subscription, and per-core, per-application, or unlimited usage. Some also negotiate a fee for every application license sold.

    d. **Support.** Does the vendor offer services and/or guidelines for how to sell, market, price, support, roll out the product in a commercial setting? Whether you simply need to augment your staff with a consultant or require a whole team to complete a large scope of work, evaluate the range of professional services offered and the extent of the partner network. Some things to consider include:

i. **Onboarding.** Look for a quick-start program that quickly ramps up your team on the solution, helps you identify and coordinate internal resources to support a go-to-market strategy, and sets milestones for completing each phase in the project plan.

ii. **Account management.** Vendors should assign account managers who are vested in your success, update you on the latest product news and trends, and coordinate vendor resources to address your issues and requests.

iii. **Support.** A combination of live and self-service support options backed by experienced professionals should be available to help you work through any technical question. Service-level agreements should clearly set expectations for response times.

iv. **Community.** An active user community can lend peer support and share valuable best practices so you can benefit from the experience of others.

12. **Analytics.** Finally, it's important to know the type of analytics the product offers. This is the functionality that most customers examine in detail, skipping many of the above characteristics, although they too are critical to the success of an embedded environment. It's best to examine the vendor's heritage to understand its strengths and weaknesses as an analytical tool.

    Vendors tend to be strong in one of the following categories:

    a. **Standard reporting.** Page-oriented, printable reports.

    b. **Dashboards.** Interactive dashboards with strong visualizations.

    c. **Ad hoc reports.** A self-service environment that enables users to create or edit reports using a business semantic layer.

    d. **Visual analytics.** A self-service environment where users can connect to data sources, blend and normalize data sets, visualize and analyze the results, and share with others.

    e. **Advanced analytics.** These workbenches and libraries enable data scientists to find and prepare data, create analytical models, and share them.

Conclusion: Success Factors

Embedding another product into an application is not easy. There's a lot that can go wrong, and the technology is the easy part. The hard part is corralling the people and establishing the processes required to deliver value to customers.

Here are key success factors to keep at the forefront of your mind during an embedded analytics project.

1. **Know your strategy.** If you don't know where you're going, you'll end up somewhere you don't want to be. Define the goal for the project and keep that front and center during design, product selection, and implementation.
2. **Know your users.** Pushing dashboards to customers for the sake of delivering information will not add value. Dashboards, whether embedded or not, need to serve an immediate need for a specific target group. Identify and address information pain points and you'll succeed.
3. **Identify product requirements.** It's hard to tell the difference between embedded analytics tools. Use the criteria defined in this report and our companion spreadsheet to find the best product for your needs. It may not be one you already know!
4. **Define a go-to-market strategy.** Here's where the wheels can fall off the bus. Before you get too far, assemble a team that will define and execute a go-to-market strategy. Especially if you are an ISV, get your sales, marketing, pricing, support, training, and legal teams together at the outset. Keep them informed every step of the way. Make sure they provide input on your plan.

Following these key principles will help ensure the success of your embedded analytics project.

About Eckerson Group

Wayne Eckerson, a globally known author, speaker, and advisor, formed Eckerson Group to provide data-driven leaders like yourself a cocoon of support during every step of your journey toward data and analytics excellence.

Today, Eckerson Group has three main divisions:

- **Eckerson Research** publishes insights so you and your team can stay abreast of the latest tools, techniques, and technologies in the field.
- **Eckerson Consulting** provides strategy, design, and implementation assistance to meet your organization's current and future needs.
- **Eckerson Education** keeps your data analytics team current on the latest developments in the field through three- and six-hour workshops and public seminars.

Unlike other firms, Eckerson Group focuses solely on data analytics. Our veteran practitioners each have more than 25 years of experience in the field. They specialize in every facet of data analytics—from data architecture and data governance to business intelligence and artificial intelligence. Their primary mission is to share their hard-won lessons with you.

Our clients say we are hard-working, insightful, and humble. We take the compliment! It all stems from our love of data and desire to serve—we see ourselves as a bunch of continuous learners, interpreting the world of data for you and others.

Accelerate your data journey. Put an expert on your side. Give us a call and learn what Eckerson Group can do for you!

About Yellowfin

Yellowfin is the only enterprise analytics suite that enables organizations to extract transformational value from their data because we combine action based dashboards, automated data discovery, and data storytelling into a single, integrated platform.

By delivering the best analytical experience, Yellowfin provides your users with unique ways to engage with and act on their data, and addresses the needs of data analysts, business users, customers and developers who want to build, deploy or use amazing analytical experiences.

Analytics for software companies

Integrate and embed analytics with a difference into your app

- Replace legacy or home grown reporting tools
- Embed a modern self-service analytics suite
- Deliver an exceptional customer experience

Analytics for enterprise

Get more value from your data in new and innovative ways

- Migrate from spreadsheets to a modern analytics platform
- Replace legacy BI applications
- Embed analytics into operational workflows

Analytical Application Builders

Leverage your domain expertise to create data products that delight

- Create unique data driven applications
- Close the loop on analytics
- Deliver insights as a service

www.yellowfinbi.com