



BUILD VS. BUY

7 Key Considerations When
Adding Report Functionality to
Your Application

INDEX	pg. 2	Technology & Compatibility
	pg. 3	Features
	pg. 4	Performance
	pg. 5	Design
	pg. 6	Documentation, Training & Support
	pg. 7	Speed to Market
	pg. 8	Cost & ROI
	pg. 9	Weighing the Key Considerations

To Build or To Buy...

7 Key Considerations When Adding Report Functionality to Your Application

If you're planning a new application, or in the middle of building one, you should be actively considering your reporting needs and development strategy. If your application is already deployed, you should be evaluating your user's satisfaction with its reporting capabilities and making plans to raise the bar.

Most applications need some level of reporting functionality, and getting it right reduces operational costs, increase productivity, improve user satisfaction or even provide product differentiation.

So, regardless of the stage or situation, you know that you've got important decisions to make that will involve weighing the pros and cons of building or buying your reporting solution.

In this White Paper we'll look at the major factors affecting your decision including: questions, considerations, options and the advantages to each approach. Use the guide to help choose whether 'Build' or 'Buy' wins in each of the 7 Key Considerations, rank their weight in your decision, and use the final score as a recommended course of action.

Keep in mind, there is no right or wrong answer—what is right for one company may not work for the other. And in the end, it will probably depend on how you prioritize these 7 Key Considerations.

Let's take a look at the major factors and considerations you will face while making your decision. →

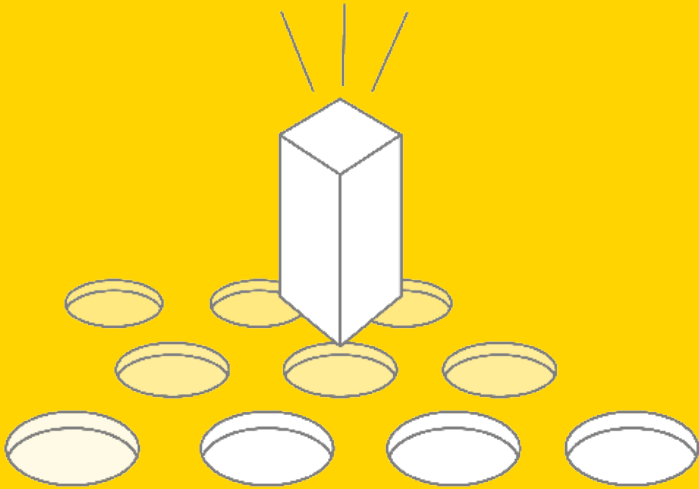
1. Technology & Compatability

Square Pegs & Round Holes

Integration compatibility can be a deal-breaker when deciding on your report solution. Some solutions are just unfit for your application—like the saying “square pegs don’t fit in round holes”.

Among third-party software apps, ease of integration varies widely. However, many packaged software solutions make it possible to perform a proof of concept (POC) before investing in a solution.

If you choose to build your solution, compatibility is not likely to be an issue. However you’ll still need to consider how your solution will connect to it’s potentially disparate data sources.



Questions to Ask Yourself:

- What is our technology platform?**
- What are our data source(s)? And are they accessible and available?**
- What are our output format requirements (i.e. print, display, PDF)? And how will they change over time?**
- What is the level of integration needed? Fully Integrated? Connected? Stand Alone?**
- What packaged solutions are available? What platforms are they compatible with?**

ADVANTAGES

BUILD	BUY
<ul style="list-style-type: none"> • Built in compatability • Full access to code • No API required 	<ul style="list-style-type: none"> • Pre-tested Integration • Ability to create a rapid prototype or Proof of Concept (POC) • Packaged integration/connectors
Which is best for you?	
<input type="checkbox"/>	<input type="checkbox"/>

2. Features

The Must Have's, Should Have's, Could Have's, & Won't Have's

Some organizations use the MoSCoW method to prioritize feature requirements.

Must Have- Critical for success or otherwise failure.

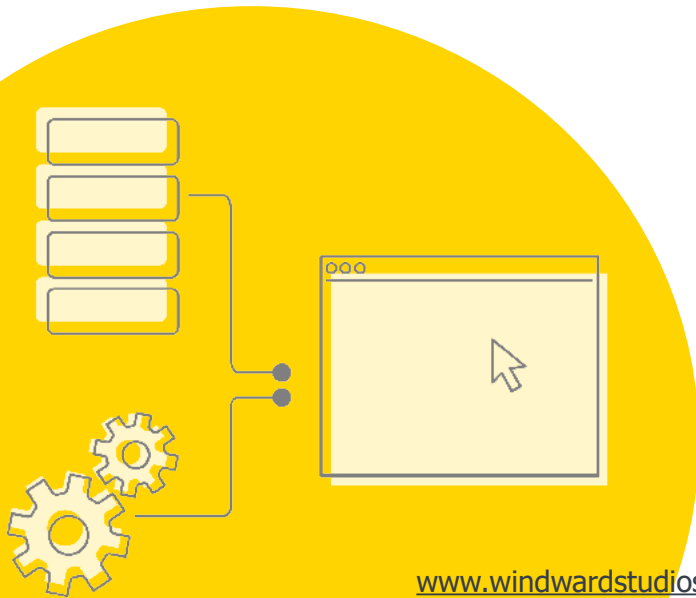
Should Have- Important but not necessary or critical.

Could Have- Desirable but not necessary. Or if time and resources permit.

Won't Have- Least critical or for a later time.

It's important to understand your end-users' needs and how they are driving your prioritization—know what is non-negotiable and what is noise.

When considering packaged solutions it's also important to look beyond what the software claims it can do, to what it does best. For example, report solutions that are strong in BI analytics often have little to no document generation capability. You may find that two separate packages is an ideal way to offer a best-in-class total solution.



Questions to Ask Yourself:

What will the output be used for: executive dashboards or interactive analysis, print or presentation, ad-hoc or scheduled?

Do we have clear prioritization from 'Must Have' to 'Won't Have'?

Will feature demands increase with growth of the application or company?

As users become more sophisticated, will they need new or better features?

ADVANTAGES

BUILD	BUY
<ul style="list-style-type: none"> • Control to build only what you need and nothing you don't need. • Control the prioritization of feature requests. 	<ul style="list-style-type: none"> • Packaged solutions typically model best practices. • Bugs and feature requests are not your developers problem.
Which is best for you?	
<input type="checkbox"/>	<input type="checkbox"/>

3. Performance

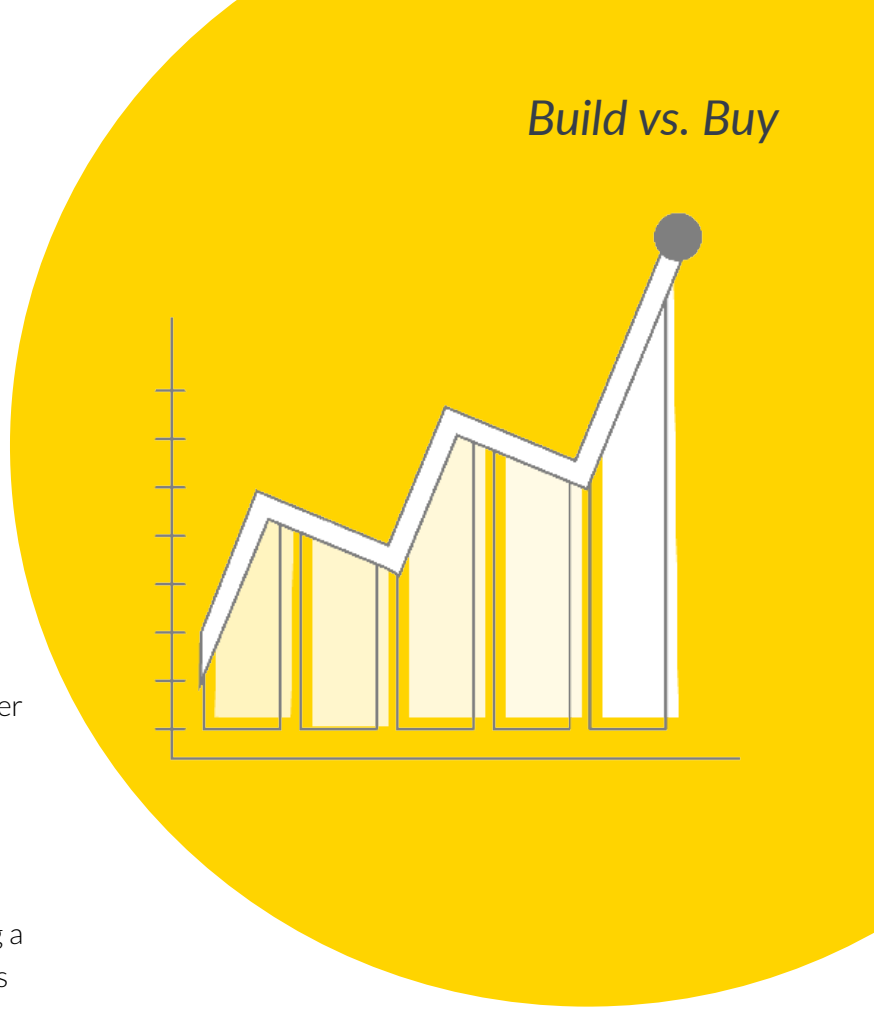
The Need for Speed

The most elegant solution is only as good as its ability to meet performance requirements.

Not only must you establish your current performance needs, but you must look into the future to determine how your needs may change over time, and how you will address them.

Within any solution, there are countless potential points of failure and performance bottlenecks.

You wouldn't show up to a Formula One race driving a school bus. You need to be certain your equipment is up to the task.



Questions to Ask Yourself:

What are our output volume requirements: ad hoc, on demand or hundreds-of-thousands of documents each day?

How will our needs change over time? Will our customer base change? Will our reporting needs become more complex? Can scale be addressed with hardware, software or both?

How will we test, measure and assure performance?

If downtime occurs, what is our exposure and risk?

Where might our performance bottleneck exist: connection speed, network speed, database performance, processing, etc...?

ADVANTAGES

BUILD	BUY
<ul style="list-style-type: none"> • Development control over architecture and performance—for better or worse. • Ability to update software if performance issues arise. 	<ul style="list-style-type: none"> • Tested configurations and established benchmarks. • Ability to establish a Service Level Agreement (SLA).
Which is best for you?	
<input type="checkbox"/>	<input type="checkbox"/>

4. Design

It's All About the Experience

Will you have a developer designing or a designer developing your reports? If the prior is true you'll need some beautiful pre-built templates. If the latter is true, you'll need a very intuitive interface. Of course, having both would be best for everyone.

Any well-designed report is able to communicate the right information quickly and clearly. But customer facing documents have the added responsibility of reinforcing a company's brand, style and professionalism. You'll definitely need some amount of design flexibility but in the case of branded documents you'll likely need very fine free-form design control.

You can learn more about creating better reports, in our White Paper: [Design Tips for Beautiful Reports](#).



Who are our users: customers, developers, internal business users?

Will the designer and developer be working closely together or will there be multiple designers working more autonomously?

How important is the user experience?

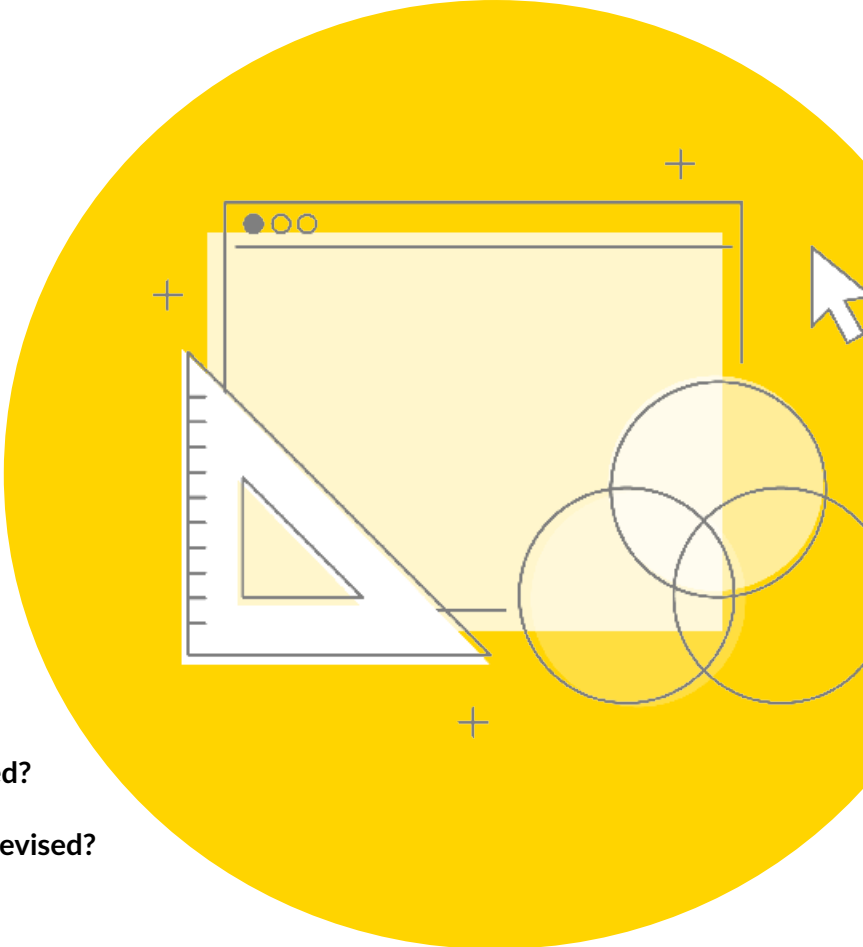
Are canned report templates sufficient, or will our users need design capabilities?

How much design flexibility and control are needed?

How often will report documents be created and revised?

ADVANTAGES

BUILD	BUY
	<ul style="list-style-type: none"> Standards and best practices are built in. Packaged solutions are available with a wide range of design flexibility and control.
Which is best for you?	
<input type="checkbox"/>	<input type="checkbox"/>



5. Documentation, Training & Support

Forget Something?

The success of your reporting solution depends on your user’s ability to learn the tools and get the results they want.

Reporting solutions tend to be complex. At least at some level, working with data can be confusing or tedious even for experienced analysts. Most rely on a collection of tools to help reduce errors and increase their productivity, and each of those tools is likely to approach things a little differently and have unique user interfaces.

Documentation, training and support can make a huge difference in the success of your reporting solution. Unfortunately, they are often an afterthought, or under-developed components of the complete solution. Developing materials and providing ongoing support can easily consume as much, or more, resources than writing the original code.

You will want to carefully weigh the need and effort as you’re making your build/buy decision.

How intuitive and how complex is the solution we’ve designed or purchased? Will we need in-depth documentation, training, or just “quick-start” level material?

How often will documentation need to be updated due to feature and version changes?

How will ongoing support be handled? And who will train the support team?

What is the initial cost of material development? And what is the cost of ongoing support?

ADVANTAGES

BUILD	BUY
<ul style="list-style-type: none"> • Documentation and training materials can be developed around the specific needs of the end users. • Control over support prioritization and response. 	<ul style="list-style-type: none"> • Documentation, training and support are typically included. • Solution expertise and support are provided by the vendor. • Periodic updates are typically provided by the vendor.
Which is best for you?	
<input type="checkbox"/>	<input type="checkbox"/>

6. Speed to Market

React Quickly, or Better Yet—Anticipate

Bringing your reporting solution to market sooner than later has numerous advantages—provided that you don’t cut corners to do it.

- Happier end users and customers
- Lower development costs due to less context switching and less scope creep
- Competitive differentiation (in the case of commercial applications)
- Less rogue IT software (in the case of internal enterprise solutions)
- Faster ROI because productivity efficiency is realized sooner

It’s a rare case that fast and first isn’t a good thing where software is concerned. If someone (like your customers or your CFO/COO) is asking for something, it’s likely that they’re already experiencing some level of un-met need or discomfort. You don’t want to wait for them to become unhappy or seek alternatives before taking action.



ADVANTAGES

BUILD	BUY
	<ul style="list-style-type: none"> • Input, design, QC development, and documentation are done—drastically increasing your speed to market. • Advanced features are likely already built into a mature package.
Which is best for you?	
<input type="checkbox"/>	<input type="checkbox"/>

Questions to Ask Yourself:

What or who is driving the timeline?

What are the potential consequences of delivering later versus sooner?

Have we considered all of the components of a complete delivered solution in the project estimate: input, design, POC, development, integration, QC, documentation, training, etc.?

What factors may affect our ability to deliver a solution on a timely schedule? To what degree do we have control over those factors?

7. Cost & ROI

Justification

ROI is the bottom-line look at your 'Build' vs. 'Buy' decision. The beauty of an ROI analysis is that you'll have a black-and-white number to look at and share with other stakeholders. The downside is that the number will only be as good as your assumptions and thoroughness. This document should help identify the primary costs and payback opportunities. But you should definitely do some digging to uncover the less obvious factors that may be unique to your organization.

One of the biggest variables is estimating development costs. There are volumes written on how to do this, yet few would claim that it can be done with certainty. Most would agree that various sizing, scoping, prototyping or POC exercises will improve the accuracy of the estimates but those can add significant cost themselves.

It's important to have a realistic view of the margin of error in your build estimates and your level of confidence in a potential vendor's ability to deliver as promised.



Questions to Ask Yourself:

Have we accounted for costs & payback opportunities?

Have we associated timelines to both the costs and payback opportunities? Have we considered how the speed to market could accelerate your ROI timeline?

Have we considered the ongoing costs of design and feature revisions?

Have we also considered potential lost opportunities due to resource allocation?

What is my margin of error?

What is my level of confidence in the vendor's implementation/integration claims and their ability to deliver software and services as promised?

ADVANTAGES

BUILD	BUY
<ul style="list-style-type: none"> • Expenditure may be spread out over a longer time period. • Periodic subscription and service fees can be avoided. • Lower development costs as they are spread across many customers. 	<ul style="list-style-type: none"> • Cost estimates are more reliable as they are fixed by contract. • Development costs are spread out over many customers and typically far less than one-off building costs. • Speed to market will significantly accelerate and shorten the ROI timeline.
Which is best for you?	
<input type="checkbox"/>	<input type="checkbox"/>

7 Key Considerations Scoring		Build	Buy
Mark an 'X' for chosen 'Build' or 'Buy' and rank the importance 1-3			
Technology/Compatibility →	<input type="checkbox"/> 1 2 3	<input type="checkbox"/> 1 2 3	
Features →	<input type="checkbox"/> 1 2 3	<input type="checkbox"/> 1 2 3	
Performance →	<input type="checkbox"/> 1 2 3	<input type="checkbox"/> 1 2 3	
Design →	<input type="checkbox"/> 1 2 3	<input type="checkbox"/> 1 2 3	
Documentation, Training & Support →	<input type="checkbox"/> 1 2 3	<input type="checkbox"/> 1 2 3	
Speed to Market →	<input type="checkbox"/> 1 2 3	<input type="checkbox"/> 1 2 3	
Cost/ROI →	<input type="checkbox"/> 1 2 3	<input type="checkbox"/> 1 2 3	
Tally each column for your results			

Weighing the Key Considerations

There's one last important step in your decision-making process: adding weight to your decision criteria. It's likely that as you've considered each of the points in this worksheet, 'Build' wins in some areas while 'Buy' wins in others. But not all criteria are of equal importance, and not all people will place the same amount of weight on each point. So, take a minute to consider not only the winner in each area, but also the amount of importance you would assign to each point. Then a simple tally should lead you to the right conclusion!

About Windward Studios

Virtually every one of our customers has faced the 'Build' vs. 'Buy' decision when it came to adding report and document generation capabilities to their applications. Those that choose Windward recognize that our approach to the question is to offer the best of both options by providing a robust mature packaged solution that's designed to be embedded or called from within their existing application. This approach drastically reduces speed to market, shortens the ROI timeline, frees up internal resources, eliminates documentation and support concerns, and delivers a rich and scalable feature set—everybody wins!

Windward Studios makes document generation software that allows companies to connect document templates to multiple data sources for high-volume output in a variety of formats. Our design tools offer unmatched flexibility, ease of use and control, and our output engines can be embedded in .NET, Java or RESTful applications.

Visit www.windwardstudios.com for more information or [Start a 7-day trial](#).

© 2018 Windward Studios