



# Best Practices for Choosing a Web Application Development Tool for the System i and Beyond

By Heather Gately

**R**esearching software solutions can be challenging, particularly when you are entering the Web application arena. Before diving into a product evaluation, you want to make sure the vendors who are courting you can give you what you actually need. Many products appear under the heading of application development, but they are a wide and varied lot, so asking the right questions can be crucial to your long-term happiness. Unfortunately, sometimes knowing exactly what to ask is the hard part.

So, today we'll cover some of the best practices for selecting an application development tool for creating data-driven IBM® System i Web applications, and provide a checklist of the most important questions to ask your potential vendors. Here's what you need to consider:



## Ease of Use and Training

Who is going to be using this tool?

- What skill sets are required to be productive with the solution? Will your users be required to learn a language in order to use the tool? What kind of learning curve can you expect?
- What kind of training is required to get up to speed, and how much does this training cost?
- Can this tool be useful to, or used by, departments outside IT? (Where can you get the most bang for your buck? How extensive are its capabilities?)

Many tools require highly technical users to learn their proprietary development language, like a 4GL (fourth-generation language) with learning curves upwards of six weeks to become productive. Other tools, like IDEs, require users to learn a development

language such as PHP or Java. This language training can take 6–24 months to reach productivity. However, there are also solutions that don't require any language training that can allow a wide range of users at all skill-levels to be productive and begin producing Web applications right away.

**Real World Example:** Take the case of two rival businesses in a race to develop e-commerce sites. Company A selects a 4GL-based application development tool. Company B selects a Web application development tool that does not require any language training. In two weeks, with a week of training, Company B has developed their e-commerce site, securely tied to their enterprise system, and begins rolling it out to select customers, bringing in scores of new sales their first week. Company A's employees, at this point, still have three more weeks of 4GL language training to go before they can even start developing, giving Company B the definitive advantage.

**Best Practice:** Take training times and learning curves into account when looking at overall timeframes and productivity.



### Capabilities

If the basic version doesn't meet your needs and you must purchase an add-on module, how much will these additional add-on modules cost in the long-run?

- Does the tool you are evaluating allow you to integrate with your business's packaged solutions, such as ERPs, CRMs, MRPs and the like?
- Can you incorporate tried and true business code and logic or complex calculations/algorithms into your new Web applications? How does the tool you are evaluating allow you to accomplish this?
- Do you have any applications written in a third-party reporting software or reports hand-coded in an older language like RPG or COBOL? Will you be stuck starting these over from scratch?

Complex learning curves don't necessarily mean comprehensive solutions. Make sure your needs are covered in the long term.

**Real World Example:** Company A wanted to modernize their reports, most of which were written in Query/400. The vendor they were speaking with was equipped to automatically, and seemingly inexpensively, bring them to the Web. But, on further review Company A also had a number of critical reports that had been written in third-party software. Their new vendor couldn't modernize those reports. Those would all need to be re-written from scratch requiring a large-scale expensive consulting project. Company A discovered this "good deal" wasn't a good fit after all. Unfortunately, it was after they had paid large sums to upgrade their hardware and purchased the tool.



IBM® System i™

**Best Practice:** Make a laundry list of what you want to accomplish now, and include those development projects that seem far off on the horizon. Ask potential vendors how you might practically accomplish these long-term goals with each solution because all tools are not created equal. Making sure you ask the right questions up front

can save you a lot of headaches (and money) in the end.



### Technical Requirements

Where will the application development tool live? Does it have to live on your System i, or could it run elsewhere? What about the applications you create with this tool?

- Do you have data stored in any non-DB2 databases? If so, you often need to purchase additional adapters. How much will those cost?



# Replacing Query/400?

## WATCH!

### Act 1 - "The Elevator"

starring

## Randall Munson

visit [www.NGSI.com](http://www.NGSI.com)

Read NGS' Query/400 Replacement Plan and Things You Should Know About DB2 Web Query at [www.NGSI.com](http://www.NGSI.com).

Then, request a web demonstration of the NGS query, reporting, and business intelligence solution that's right for you.

[www.RandallMunson.com](http://www.RandallMunson.com)



**NGS**™  
NewGeneration  
Software, Inc.

800.824.1220 • [www.NGSI.com](http://www.NGSI.com)

- What kind of processor upgrade or memory upgrade combinations will your system require to run this solution? How much will *those* cost?
- What is the application development tool's architecture? What about the applications it produces? What language are the applications coded in?

Your users need applications that work, and work better than the ones they had before. Performance is key, and the choices you make can greatly affect that. Whether they are accessing the application from their cell phone, a dial-up, or a T1, they expect sub-second response time, and the choices you make based on your solution's technical requirements can greatly affect both performance and cost.

**Real World Example:** Company A selects a seemingly inexpensive Web application development tool that can only run on their System i. In order to run the tool, Company A will need upgrades to their System i processor and memory that add up well into five figures. Company B selects a tool with a higher price tag that can be served from a wide variety of hardware including their System i™. Company B has similar hardware restrictions to Company A, but avoids added hardware costs by serving their tool from a less expensive, more efficient setup on a Linux server, effectively saving them tens of thousands of dollars, and providing a faster solution for a lower overall cost.

**Best Practice:** Make sure you factor in the technical requirements before you make your final decision. Some application development tools, even seemingly simple report-writer or query tools, can require large demands on your current hardware and infrastructure.



### Deployment

How do you feel about exposing your System i to the Web?

- Can you deploy and serve your new Web applications from another server?
- Are there costs associated with deploying to another server? If you need to move your Web application to a new server, will you need to re-develop or re-compile your application? Will you need to pay run-time fees?

If you are planning on serving your applications from the System i, you need to consider whether you are comfortable exposing it to the Web. Many tools require you to do this. If that is not something you are comfortable with, you need to look at the solution vendors that allow you to deploy and serve your applications elsewhere, whether it's a second IBM e-Server, or a Windows or Linux server.

**Real World Example:** Company A has decided to create a system of Web applications to allow their suppliers to access



invoices, pay bills, and order products online, but they don't want to expose their System i to the Web, so they decide they want to serve their applications from another server. The vendor they have chosen charges run-time fees to deploy their applications to another platform, making their new venture cost-prohibitive, and Company A unable to affordably provide this solution.

**Best Practice:** Look for platform flexibility if you are looking for a long-term solution, and select a tool that doesn't require run-time fees to keep your long-term costs down.



### Maintenance

Once my applications are built, what do I need to do to maintain them? Will I require help from the software vendor to maintain them long-term?

- How are upgrades handled? Will upgrades to my solution cause me to have difficulties supporting the applications I've already built?
- If my vendor stopped supporting my application development tool, or went out of business, how would I maintain my applications built with their solution?

Selecting a software tool that creates stand-alone applications in an open and accessible software language is an important consideration when addressing long-term maintenance.

**Real World Example:** Company A purchases a Web application development tool, and develops an entire system of Web applications over the course of five years. Their custom applications, crucial to their daily business, are written in code that was proprietary to their tool vendor. When Company A's vendor stops supporting their tool, Company A has no way to update, modify, or maintain their slew of applications because they are written in their vendor's unsupported language. They are left in the dust. At the same time, Company B's tool vendor goes out of business. The difference is, Company B's tool wrote applications in industry-standard code, so their applications are modifiable and maintainable long-term, by any outside developer, so their solutions live on well into the future.

**Best Practice:** Choosing an application development tool that creates business applications in supportable languages (without proprietary ties) means that in the long-term an independent developer can come in to maintain, or modify these applications, no matter what happens to the vendor or tool along the way.



### Pricing

How does the pricing work? Is the pricing per-seat, or a pricing model that is simplified to include unlimited users? How does this affect the price, when taking

into account the number of employees I have?

- What is the cost of the most basic software, and what additional modules might I have to pay for either now or in the future? How do additional modules affect hardware and performance considerations?
- What is the annual maintenance fee, and what does that include? Does it include support? Does it include unlimited upgrades?

Some tools require additional modules to tie into your ERP, schedule or distribute reports, query database management systems, create datamarts, etc. How much will the additional components you require cost?

**Real World Example:** Companies A and B are both concerned about their budgets. Company A looks at the price tags of their vendor options, and goes with the one with the obviously lower price tag. But, Company A requires

“plug-in” modules and their vendor has required upfront maintenance charges for each module, which raises the price by tens of thousands of dollars (annually). Suddenly a price tag that looked like a few thousand dollars has ballooned to a six-figure gotcha. Company B, on the other hand, selects a vendor with a higher initial price tag, but because their vendor has unlimited users and all-inclusive capabilities, with optional maintenance costs, their initial price that seemed steeper prices out for a much lower total cost of ownership.

**Best Practice:** Carefully examine pricing structures. Software vendors are not in business to give away their software for free, and they find many ways to line their pockets. Per-seat or per-developer licensing, particularly when coupled with modules, can turn what once looked like a cost-effective solution of a few thousand dollars, into a six-figure monster. Asking the right questions up front can save your budgets long-term.



## Support and Track Record

Who is the vendor? How long have they been in business? What kind of support do they offer? Hours of service?

- Can you talk to their current customers and find out if they are happy with the solution?
- If the basic version doesn't meet your needs and you must purchase an add-on module, is the module from a partner, or the company... and who will provide support?

One way to test a company's support mettle is in their support of software trials. How are the support questions handled there? Are your questions responded to with timely and complete answers? How seriously does the company take their support?

**Best Practice:** Good vendors can show you indicators of how quickly they handle support calls, and keep metrics of their support record. Ask to see their dashboard of recent support call records, and how they were handled.

Booth 534

# Where has your data been? Where is your data going? Where is your data now? Who is watching it?

Secure manage collect and control your data

PATRICK TOWNSEND SECURITY SOLUTIONS [www.patowndsend.com](http://www.patowndsend.com)

The best kept secret



## Primary Goals and Other Important Long-Term Considerations

Are you interested in one or two applications, or a long-term solution? Does the tool vendor offer consulting or mentoring to work with your long-term goals?

- How do you feel about being tied to a proprietary solution? Would you be more comfortable with an open solution?
- With this tool can you give administrators the proper security or the ability to limit data views based on authorization?
- Are users required to run their applications from the tool's development environment, or are their new solutions stand-alone Web applications they can access from their browser?
- What about scalability? As your company grows, both in users, and in applications, what kind of stress will that put on your system?

**Best Practice:** Look for the application development tool that isn't purchased for your business today, but purchased for your business tomorrow, meaning that it not only fits your needs, but allows for growth and usability beyond your initial project.



## Closing Thoughts

When choosing your application development tool, it is important to remember, the only constant is change. The best case scenario is growth, whether that's customers, end-users, or development seats. How will this affect your needs and bottom line long term? Mergers and acquisitions and new decision makers can quickly change the scope of platforms and technology. How would the solution you've chosen be affected if you moved off your current platform a year from now? What about the applica-

tions you created? Could you just point them to the new platform, or would they need re-development?

Lastly, perhaps the most important best practice for selecting a Web application development tool is to have a clear goal in mind before embarking on your journey. The more you clarify your own needs, both short-term and long-term, the more application development vendors and consultants can provide you with the right solution fit for your business, both today and well into the future. 

## About the Author Heather Gately

lives in Chicago and works at mrc ([www.mrc-productivity.com](http://www.mrc-productivity.com)). She is a regular contributor to IT industry publications, newsletters, industry blogs and System i Trend articles. mrc is the producer of the Web application development software tool: m-Power ([www.mrc-productivity.com/products/](http://www.mrc-productivity.com/products/)).



**COMMON**  
A USERS GROUP  
COMMUNITY - EDUCATION - ADVOCACY

Readers of COMMON.CONNECT are more than just readers... they're our engaged members and your qualified prospects.



For advertising rate information contact:

*Daniel Meek*  
Sales Associate  
[daniel\\_meek@common.org](mailto:daniel_meek@common.org)  
312.279.0218

*Manzoor Siddiqui*  
Marketing Communications Manager  
[manzoor\\_siddiqui@common.org](mailto:manzoor_siddiqui@common.org)  
312.279.0253