

Top ten checklist for evaluating modernization tools

"Give us the tools, and we will finish the job." –Winston Churchill, February 9, 1941

While application modernization isn't as daunting as the challenges Churchill faced during World War II, developers and IT managers may feel his words perfectly express their sentiments as they face a mountain of 'legacy' code that needs updating.

Some developers may wish for the kinds of bombs and artillery Churchill wanted, so they can blow the legacy code to smithereens and start afresh without all the headaches of gnarly old code. Dropping a bomb on an enterprise's code portfolio might be gratifying to the IT team, but it would likely destroy the enterprise along with the code.

When more sensible views prevail, development teams often start looking for the right application modernization tools to help them do the job.

To help in that search, here's my top ten checklist for finding the right tool.

1) Get a good handle on what 'modernization' means so you know what tool capabilities you need.

You may need the ability to create a variety of user interfaces – Web, mobile, etc., but 'refacing' isn't where modernization ends. Go deeper and look at the full range of modernization issues that need to be addressed.

2) Kick a few tires to get a real world feel for the variety of tools that are available.

I'm not talking about lengthy test drives, that comes later. Watch some videos on

vendor Web sites, visit trade show and conference expo booths, and invite some vendors to present their demos on-site. Keep written notes of what looks promising, as well as concerns over shortcomings in the products you look at.

3) Get advice from your peers in the IBM i social network.

Hearing about other IBM i developers' experiences with modernization tools is invaluable. You wouldn't buy a car without plugging into the discussions on www.edmunds.com would you? To use your time productively, I'd suggest getting customer references from product vendors and using that 'legacy device', the telephone, to call. For some reason, serious IBM i application developers don't seem to hang around Facebook to share their insights.

4) Sketch your Use Scenario and your Enterprise Application Architecture.

Once you had some exposure to what's really out there in the product world, it is time to shut your office door and draft a concise description of where you are and where you want to go.

A Use Scenario is a collection of use cases



Paul Conte
President PCES, and a
leading Application Strategist

and other information that describes your situation and needs. For example, are you focused on making your applications run on platforms in addition to the IBM i? Is support for mobile devices a priority? And so on.

An Enterprise Application Architecture describes how a set of building blocks and principles should be used to design, implement and adapt applications that fulfill the enterprise's business objectives. A well-thought-out application architecture not only provides a conceptual structure to guide developers. It also lays the foundation for a corresponding framework and toolset that can automate much of the application development effort, an essential prerequisite for greater agility, productivity and reliability.

You can learn more about these two subjects in my eBook trilogy "Transforming IBM i Applications" www.beyondmodernization.com

5) For most IT organizations: Go with an integrated product.

In theory, your organization can assemble disparate tools to implement your architecture. In practice, this may require additional staffing to support this type of infrastructure.

Modernization isn't a one-time facelift or code conversion. Modernization is a transformation of your application development strategies, so you can efficiently

Transforming IBM i Applications

Your Journey Beyond "Modernization"

An eBook Trilogy by Paul Conte

Book 1: Prepare for Your Journey — Know your goals & chart a course

Book 2: Be a Savvy Traveler — Plan your trip, step-by-step

Book 3: Embark with Confidence — Be well-equipped & follow this guide

Download your copy from: www.beyondmodernization.com

deliver the right information and capabilities to the right people at the right time, as determined by changing business needs and initiatives.

For most IBM i organizations, purchasing an integrated application generator is the most practical way to deploy the tools necessary to fulfill this objective.

6) You're going to need a tool that provides an 'application repository'.

An application repository provides persistent and structured storage for all the information that defines applications. The repository may also contain information related to requirements, standards, environment and other aspects of application development and deployment.

The pivotal advantage of a repository is that shared application items can be stored, managed and re-used effectively. Items like data element definitions, business calculations, user interface templates, code snippets and so forth, can be defined once and used across many applications and developers in a consistent manner. Having one definition for a shared repository item yields application consistency, and greatly simplifies application maintenance.

7) Take any candidate product for a hard 'test drive'.

One of the most common mistakes I've seen when an IT group evaluates a modernization tool, is that their 'test drives' usually are comparable to making sure the wheels roll and the brakes work during a new car test. Not very demanding.

What you need to do is what I did when I tested the sport sedan that I later bought. I went to a large, vacant parking lot, accelerated to 40 mph and then jammed down hard on the brakes while whipping the steering wheel sharply to the left. Impressively, the car rolled

right through the sharp turn with no skid at all. That's when I knew the car measured up to its promotional literature.

So, throw hard cases at any prospective tool to see how well it handles under pressure.

8) Spend a lot of evaluation effort and reference checking time on how well a tool enables you to exploit the functionality of your existing code.

For many IBM i development teams, the biggest hurdle to modernization is the insurmountable cost required, not to mention the risk, to 'rip and replace' legacy RPG code. Consequently, the ability of a tool, such as an application generator, to incorporate existing application functionality is critical and varies widely among available products that support the IBM i.

9) Evaluate risk realistically.

Humans are notoriously bad at correctly assessing risk, and IT professionals aren't immune to this limitation. Management is likely to be uneasy about the risk of acquiring tools for application development that aren't as widely used in the industry as Java or SQL.

The reality is that no comprehensive application modernization tool for the IBM i will ever reach the level of use enjoyed by these two languages or some of the popular code-centric Integrated Development Environments (IDEs), such as Visual Studio and Eclipse. But here's another reality: No version of RPG and its associated IDE will ever have much of a market presence either.

There are tools and application generators that produce industry standard code, freeing up developers from low-level coding complexities and allowing them to focus on business functions, a wider range of user interfaces and better integration with internal and trading partner applications.

Although I don't have concerns about the IBM i or RPG compiler going away, corporate mergers and acquisitions that force IT to support a different platform (as well as, or instead of the IBM i), are just one of the 'risks' that IT must also consider in choosing a development strategy. Application generators that target additional platforms, as well as the IBM i, can mitigate this category of risk.

There aren't any canned answers to balancing risks, so give this critical dimension of your modernization strategy very careful consideration.

10) Beware of 'shelf ware', budget adequately for training and mentoring.

With any powerful tool or toolset, your organization will benefit tenfold by having adequate initial training and mentoring.

While good tools can automate and standardize much of the 'glue and plumbing' code required in modern applications, the same tools provide such wide latitude to developers, that you want to get off on the right foot using 'best practices' learned from training and mentorship.

More Wisdom

There's another bit of wisdom that Churchill passed along, which seems appropriate to IBM i developers and managers facing the challenges of application modernization: "A pessimist sees the difficulty in every opportunity; an optimist sees the opportunity in every difficulty".

I certainly think there's lots of opportunity in a well-planned IBM i modernization strategy.

Paul is a software developer, consultant, widely read author in the computer software field, and a former senior technical editor for System iNetwork. Paul is a guest blogger at <http://blog.lansa.com/> ■

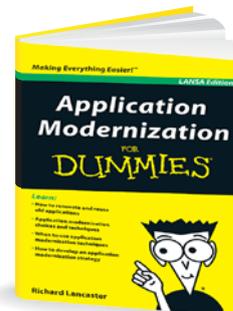


Application Modernization for Dummies

Learn:

- How to renovate and re-use old applications
- Application modernization choices and techniques
- When to use application modernization techniques
- How to develop an application modernization strategy

Download your copy from: www.modernizationfordummies.com



For Dummies, the Dummies Man logo, and related trademarks and trade dress are trademarks of Wiley Publishing, Inc.