Why Upgrade Your ERP Solution?

By Sean Solberg

NAV Practice Director, Stoneridge Software

Stoneridge software

Why Upgrade Your ERP Solution?

Contents

ERP Upgrade Overview

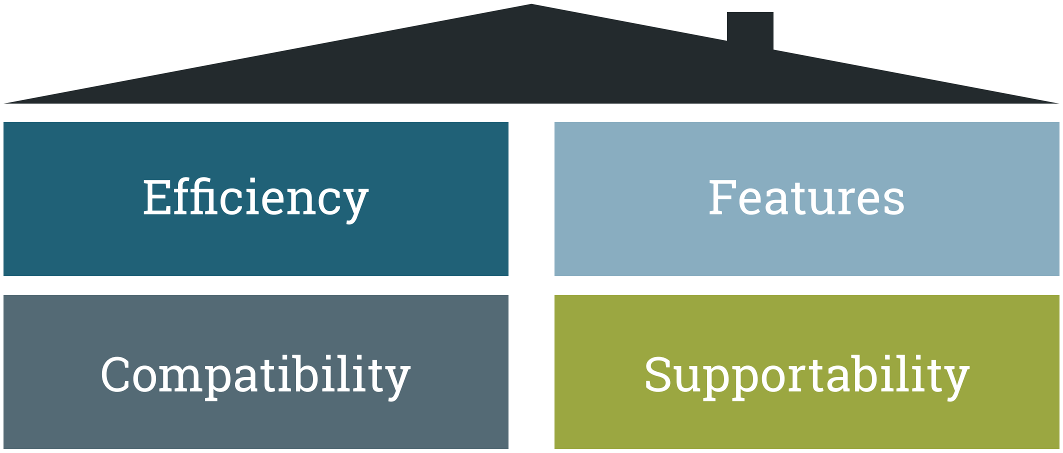
In this white paper, we will explore the benefits of an upgrade, what to consider as you decide, what to expect with your upgrade, and how to make the most of it.

Why Upgrade Your Microsoft Dynamics ERP Solution?

The decision to upgrade your ERP solution isn’t exactly a spontaneous, impulse choice. A thoughtful and responsible upgrade decision requires care, consideration, and pre-planning.

In the Enterprise Planning (ERP) arena, you can think of building a house as an analogy – start with a strong foundation, add options, and build it to serve the needs of those who live there. ERP upgrades would correspond to adding higher efficiency windows or appliances, installing infrastructure for “smart home” features, addressing building code changes, or minimizing future service costs.

The key benefits of a Dynamics AX upgrade or a Dynamics NAV upgrade are:



Efficiency

Features

Compatibility

Supportability

EFFICIENCY

Increase performance and refresh the user experience.

New windows and furnaces are more efficient today than when your home was first built. The same holds true for computer technology. Technology is continually evolving, this is a given. Not too many years ago, we wouldn't have imagined the feasibility of cloud-deployed solutions, smart-phones, tablet adoption, and the general speed at which we are now able to do business.

Processors are faster, RAM is cheaper, and solid-state drives are more prevalent. And network improvements allow for remotely deployed solutions (cloud/Azure, hosted) – removing much of the IT responsibility from your organization. On the software side of the equation, Windows and SQL improve regularly to take advantage of both those hardware changes and other shifting tools.

> **"If you are still using an ERP solution that was architected years ago, you aren't leveraging significant recent advancements in technology.."**

If you are still using an ERP solution that was architected years ago, you aren't leveraging significant recent advancements in technology. As a result, processes begin to feel "sluggish" when compared to other software. Productivity ultimately suffers due not only to actual software performance, but is compounded by reluctance of end users to engage the tool.

FEATURES

Introduce new features and functionality.

The Internet of Things (IoT) is all the buzz right now. In the home, this may mean that your refrigerator can generate shopping lists or even place orders at the grocery store. Applications like Nest allow smarter heating and cooling regulation, lighting control, plant monitoring and many other comforts. Likewise, newer ERP solutions also provide significantly upgraded features not available in prior versions.

Every ERP solution evolves, and this is for a variety of reasons; to improve the user experience, add upgraded functionality, and for the software publisher to stay competitive in the market. The net effect of these evolutions is good for users, and can include critical infrastructure improvements, or new and improved functionality, such as more streamlined, role-tailored workflows, new functional areas and better interface options.

COMPATIBILITY

Your business eco-system will be in harmony.

One of the more serious, and arduous, aspects of the house-building analogy is the concept of ensuring the building is up to code as new information emerges and regulations adapt. Likewise, the assumptions that your ERP solution was originally written against have changed. New Windows versions alone can account for significant compatibility issues if you continue to try to run older software programs on them. External programs that your ERP solution talked to previously may no longer be able to communicate, once updated. The bottom line is that new business tools are continually introduced and older tools are constantly evolving. Some examples of these include smart phones, tablets, VOIP, web stores, credit card integration, business intelligence (BI) applications, shipping software, and Microsoft Office.

> **“New business tools are continually introduced and older tools are constantly evolving.”**

Upgrading your ERP solution can be essential to your business systems staying connected.

SUPPORTABILITY

Hold down ongoing support costs.

Just as an experienced plumber or electrician can work with outdated fixtures and piping, a good ERP solution provider can usually provide support for older versions of your software. But they are typically not as expert or efficient in the outdated older environment. Or, worse yet, your solution provider may have discontinued support of any kind for the environment.

For example, if you are on an ERP solution from ten years ago, you may well find that your ERP partner has a limited set of human resources who have direct experience with your version. In most cases, support will be available – it just may have become more scarce and expensive. (Basic supply and demand principles apply here.) Because your partner is busy innovating, implementing new systems and extending workflows, outputs and configurations in different and exciting ways, they will have fewer applicable tools for your legacy version. If, however, your company goes with the upgraded version of the solution, the software and your partner can offer significant additional value to the investment, like custom micro-solutions, reports, add-on tools, and deep expertise on how to best utilize the improvements in your business.

What To Expect When Upgrading Your ERP Solution

In a word: disruption – at least a little.

For most companies, direct involvement by your preferred partner is required to ensure that the configurations and customizations made to your existing solution work as needed in the upgraded solution.

Specifically, you and your partner will need to address potential conflicts between:

1) What was done to customize the earlier solution and data to meet business needs and;
2) What the software vendor has added to the out-of-the-box solution in the more recent version(s).

Whether your ERP solution handles those customizations directly in the business logic, as added layers, or completely externally to the system, conflicts will inevitably occur.

Your partner will help address those conflicts by enhancing data upgrade tools to include your structural changes, merging your custom business logic into the new solution, possibly removing customizations no longer needed, addressing any security changes needed for the new version, and generally testing the system to ensure consistency.

> **“An upgrade project does not actually impact your live environment until the release/go-live date.”**

Your team will be impacted to a moderate degree – in testing iterations, possibly some joint-process-(re)design sessions and, in some cases, formal end-user acceptance testing. Likewise, the release of the upgraded application and environment will probably cause efficiencies to decrease slightly in the short-term as everyone gains familiarity with changed interfaces, new functionality and altered workflows.

On the bright side, and unlike remodeling a house (using the analogy from Part 1 of this series), an upgrade project does not actually impact your live environment until the release/go-live date. Therefore, during the project, you won't be trying to work in a kitchen without counters or a room without flooring. Your use of the existing production ERP solution should not be limited in any way until you finally execute the upgrade at go-live.

UPGRADE PHASES

A typical upgrade includes the following phases *(order may vary depending upon project approach and business requirements):*

1. **Technical Upgrade:** Your partner merges code from the current version of the application to the new version of the application
2. **Server / Environment:** You partner or your internal IT staff builds or updates the server and infrastructure to support new application
3. **Test Data Upgrade 1:** A test instance of the application is loaded onto your network, generally including an initial pass at upgrading your data
4. **Integrations:** External systems with integrations or touch-points to the application are duplicated in a test environment (or redirected as needed for testing)
5. **Key Training:** Key users are trained on the new version of application
6. **Test Workflows:** Key users test the new version of application
7. **Iterative Changes:** Any reported issues, workflow conflicts, or security conflicts are reported to the partner and are addressed
8. **Test Data Upgrade 2:** [Optional] A second test run of the upgrade is performed – typically if some data conversion issues were discovered in testing
9. **End User Training:** Key users train end users
10. **User Acceptance Testing:** End users test their respective processes – for workflows, outputs and permissions
11. **Go/No-Go Decision:** Go-live / conversion date is confirmed (with conversion activity typically targeted for a weekend)
12. **Release:** Go-live / release of upgraded solution
13. **Support:** Post release partner support and issue resolution

INVESTMENT LEVELS

There are two ends of the investment spectrum when we consider the approach to testing:

Low Investment

On this end, no testing is performed and any conflicts or issues with the code and data migrations are addressed after go-live. This approach is often the less expensive route, even when post-release costs are included. However, many companies avoid this approach so they can bypass user frustration and adoption limitations.

High Investment

Some companies can actually over-rotate here. The argument here is for spending whatever it takes, in terms of hard dollars and internal resource time, to ensure that not a single surprise surfaces after go-live. While this route is much more expensive, it can actually "burn out" the users on the product and their own processes. On the other hand, deployments at the highest investment level are comprehensive and relatively worry-free.

For most companies, there is a sweet spot in the middle, based on budget, user preparation for expected post-go-live pain, and the time-sensitivity of your processes and data. If your go-live is fully pain-free, then you:

1) may have over-invested in testing,
2) had no conflicts in your customizations or,
3) got very, very lucky.

For the highest investment impact, aim for full testing of critical processes and light-to-moderate testing of non-critical processes. If you are able to then absorb the incidental issues arising after go-live within the first few weeks, congratulations! You have probably struck the perfect balance.

Getting the Most from Your ERP Upgrade

You’ve made the decision to upgrade your ERP solution. You have prepared yourself and your organization for some level of disruption. Now you’re ready to engage on the project. You have one final decision to make, regarding your deeper involvement in the project.

UPGRADE VS. RE-IMPLEMENTATION

The two ends of the upgrade project spectrum are often labeled as “upgrade” versus “re-implement.” Even the most aggressive “re-implementation” is still an upgrade at some level, and most projects fall somewhere in the middle. With that distinction in mind, it’s preferable to consider the following:

1. Treat the project as a pure upgrade.

Ask your partner to carry forward any and all customizations as they were originally written and introduce no new changes outside of what is available in the newer version. Aside from the changes introduced by the software publisher, the application, configuration and your workflows will be unchanged.

2. Use the project as an opportunity to make additional changes that will help the business.

Engage users, promoting more thorough solution ownership throughout the company. Engage with various department and process “owners” within your organization to understand:

- What new pain-points have arisen since you initially implemented?
- Do newer business processes call for different approaches within your ERP solution than you are using today?

- Are there any gaps that were not addressed in the original implementation now demand more attention?
- What customizations and unique application configurations (and reports) can now be discarded?

The first option, a pure upgrade, allows your team to mostly remain outside of the project until the testing phase. This approach works well if the application is lightly customized, internal constraints prohibit active user involvement in the project, or if the current largely solution does what you need – minus some features that are already included in the newer version. This route also tends to be much cheaper, provided the newer version of the software works more or less the same as today.

The second option requires a more formal project approach and significant involvement from your team. This approach works better if current customizations can be left behind (i.e. cleanup the database), if new functionality or workflows are justified, or if users don’t already “love” the current version of the solution as implemented. Analysis is often needed, either formally as a distinct project phase, or through ongoing dialogue as functional areas or customizations are considered for inclusion. This option feels more like a highly managed implementation. And an upgrade is generally less threatening than an implementation as no one necessarily feels that their job or role is being threatened through ROI pressures.

REVIEW THE SOFTWARE ITSELF

One final factor that may dictate this decision for you is the software itself. If the software publisher has reinvented the solution, partially or completely, it almost always makes sense to take the second approach. Going back to the house-building analogy, if you ask a service technician into your home to upgrade an appliance, you probably don’t need to participate much beyond the initial purchase. The new appliance should do exactly what the old appliance did, but possibly with a few extra buttons

On the other hand, you probably wouldn’t ask a contractor to remodel an entire room in your home without some deeper discussion of the various options now available or without your continued participation in the project as it progresses. When the software publisher rewrites part of the ERP solution, your old workflows, configuration decisions and customizations will likely not apply the same way or produce the exact same results as today. Further, there may be significant, entirely new areas of the application not previously available. You should be involved to ensure that you aren't carrying forward antiquated or invalid workflows and that you are taking advantage of any possibilities afforded by the newer version.

If you go with the second approach, strongly consider how much you should be involved and plan accordingly. Include other areas of the application that may be available today, but haven't been put to use in the solution. Because everyone will be re-focused on the software, the project can also be used to shore up some training and revisit individual workflows for lost efficiencies. Discuss your thoughts with your partner and go deep into the dialogue. You should both walk away from that discussion with a clear understanding of immediate goals and concerns, impact of the publisher's changes, cost of carrying forward any previous customizations, resource availability, and longer-term potential changes in business.

Your partner will always know more about the software, but less about your business. Your involvement, where justified, will help ensure the two are again in harmony when the project is complete.

If you have further questions regarding the ERP software and upgrade process please reach out to our resources at solutions@stoneridgesoftware.com or call 612-354-4966.