



# App Topics

Email: [info@boru-consulting.com](mailto:info@boru-consulting.com)  
Web: <http://www.boru-consulting.com>  
Phone: +353-1-633 5161

Vol. 1, Issue 3, September 1999

## Managing an upgrade

### Introduction

Have you recently spent large amounts of money and scarce resources implementing ERP applications that will be Y2k compliant? If your implementation involved the Oracle Applications 10.7 release (character, smart client or NCA versions), then you are most likely considering the benefits or necessity of upgrading to release 11 or 11i. This article deals with the upgrade process right from the decision on whether to upgrade, through to sustaining the applications once the upgrade is complete. Although the focus is on the Oracle applications, the principles apply to any ERP applications upgrade.

### Why Should I Upgrade?

Getting an organization through an implementation of ERP applications can be difficult. Upgrading those applications can be just as difficult. It is necessary to carry out some preliminary analysis to determine whether an upgrade is right for your organization at this time

and if so, which upgrade path is the correct one to choose.

#### Chasing Versions

Some of this analysis will be easy. If you are currently using a release of the Oracle Applications prior to 10.7, then it is recommended to upgrade to 10.7 to maintain an upgrade path to release 11 and beyond using Oracle migration tools.

Of course, if you are really determined, you can approach this type of upgrade as a “new” implementation and skip the 10.7 release by going directly to one of the 11.x releases. This will involve converting your data to fit with release 11 table structures. You will also need to review any customizations to ensure compatibility and their relevance in the new release.

Upgrading from 10.7 (character, smart client and NCA) to release 11 is supported with Oracle supplied migration tools. If you are using 10.7 NCA or smart client releases, then you have the option of upgrading to release 11 right now. If you are using 10.7 character version, then you

## App Topics

may wish to wait until later in 2000 until a stable version of release 11i is available. The additional functionality may justify postponing any upgrade plans until then.

In any case, perhaps the most significant external factor in making the upgrade decision is continued support of your current release. It would be foolish to continue to operate your business on a release that is not being supported. Oracle has recently announced continued support for the 10.7 release until 2001. This should allow adequate time for most organizations to plan for an upgrade well before 10.7 is de-supported.

### Cost-Benefit Considerations

De-support of your current applications release will probably make an upgrade decision easy. Upgrading to a new release on the merit of new or improved functionality alone is a more difficult decision. Like any other business decision, it should be possible to analyze the benefits to your organization and the costs that will be involved in upgrading.

In looking at new functionality, it is important to keep in mind the strategy and goals of your business. Do new features support your current goals or provide a strategic advantage for the future? If you have carried out any **business process re-engineering** (BPR) activity, then you should have information on your current business processes that can be used to quickly assess the relevance of any new application functionality.

When you are considering the costs involved in upgrading your applications, you should look at:

- Cost to support the current release
- Cost to upgrade to the new release
- Ongoing costs to support the new release

Obtaining information for the current support costs and the upgrade costs should not be too difficult. The ongoing support costs for a new release may be more difficult to assess, but will be an important part of the analysis. You should consider:

- Elimination or continuation of customized forms, interfaces, reports etc. If you succeed in eliminating customizations through an upgrade, this can have a significant impact on the upgrade cost and your continuing support costs.
- Current system architecture vs. architecture to support new release. Moving from a smart client environment to an NCA environment, should have a significant impact on desktop support.

## Gathering Your Resources

Now that you have the information required to justify an upgrade, you need to obtain the approval of your management team to proceed. Make sure that there is a valid business case to support the upgrade. It should include all the costs, the benefits and clear statements of how the upgrade will support the objectives of your business. At this time, you should also identify and present the deliverables for the upgrade and put together a preliminary upgrade project schedule with an estimate of the financial and people resources that will be required.

An upgrade can be as demanding of resources as a new implementation. It will require a project manager who has the experience necessary to guide a team of technical and functional experts through the upgrade process. You should also consider enlisting the support of **external consulting resources** who can bring experience,

expertise and a different perspective to the upgrade.

If you are fortunate to have experienced technical resources, your management team should be prepared to commit these resources to the upgrade project. Don't underestimate the importance of functional expertise throughout the upgrade. You should identify **KEY** functional users who can be assigned to the upgrade project for the duration. If you have managed to retain implementation team members in your organization, they may be ideal candidates for this role.

## The Upgrade Project

Your upgrade project can be broken into a number of distinct phases:

- Justification/Upgrade Decision
- Analysis and Planning
- Configuration and Test
- Upgrade
- On-going Support

The process of justifying the upgrade and arriving at a decision has been covered in the previous section.

### Analysis and Planning

In this phase of your upgrade project, you need to carry out a detailed analysis of your current applications environment, your existing business processes and the functionality provided in the new applications release.

For this phase, you should ensure that the functional and technical experts on your upgrade team have received training on the new release. A full understanding of the system architecture

required and the functionality will be important in making decisions during this phase.

You should also consider bringing in some **external consulting expertise** to support your team. This will supplement the knowledge that your team members are gaining and should provide some additional insight into how the applications may work with your business processes.

Take the time to analyze your current business processes and determine how the new release of the applications will impact them. Will there be changes required to the existing processes or will completely new processes be required? Make sure that any changed or new processes are mapped.

Have the team look at customizations that have been carried out on the installed applications. This should include forms, interfaces, reports etc. The assessment should include a comparison of the requirement for a customization with the functionality that exists in the new release. Is there an opportunity to eliminate a customization in favor some new functionality? What changes would be required to existing business processes to support this? What would it cost to make these changes? If a customization **MUST** be carried forward in the upgrade, remember that this will require time and resources in your upgrade project to validate the customization and make any required modifications to work with the new application release. If at all possible, you should consider modifying a business process to use new functionality rather than carry forward customizations. Using standard application functionality reduces the ongoing support burden for your technical resources and makes future upgrades less painful.

The technical members of your team need to review the system resources and architecture

## App Topics

required for the new release. This is particularly important if you are upgrading from a character-based or smart client release. The Oracle “NCA” system architecture is quite different to that required for earlier releases.

At the end of this phase, you should have a detailed project plan, a definition of your system architecture and a full understanding of the resources required to make the upgrade successful.

### Configuration and Test

Initially, you can install the demo database to help the members of the team fully understand the application functionality.

For the serious configuration and testing work, you will need to create a test environment. Copy your production database to the test environment and perform the upgrade there. This will provide your technical team members with some first hand experience of the upgrade process and provide the functional team with some “real world” data to work with. The functional team need to configure the applications to work with your business processes.

The technical members of the team also need to start working on any required customizations.

The functional team should carry out a Conference Room Pilot using the upgraded applications. The objective of the pilot should be to validate that the existing or modified business processes work with the upgraded applications and that any potential issues are identified and dealt with. Have a full review of the Conference Room Pilot with the team and ensure that action plans are put in place to resolve any open issues.

You should schedule any custom development work to complete to allow time for testing with the applications. Depending on the complexity of

your environment, you may opt to test individual applications separately.

The true test of the upgraded applications environment should be an integrated test. You should produce a “test script” to use here. This is something that your **consulting resources** can assist with. Consider refreshing the test database with current production data at this time.

To make testing more relevant, you can have the functional team run some key reports in the production environment that can be used to validate the results found during testing. This is an opportunity for your team to test the complete environment including any customizations. For each step in the test script, you should identify the expected result, what actually happened and any modifications or corrections that may be required. The team can sign off on each test that successfully completes.

Review the results of testing with the team. Were there any unexpected or undesired results? How did the customizations perform? Put together an action plan to address any shortcomings.

At the end of this phase, your team should also prepare any training materials that may be required for user training. If you are upgrading from a character based release of the applications, remember that the user interface in release 11 and beyond will be significantly different and affect all users (including casual users of the applications). Wherever possible, your training should be by job function rather than by application. Typically, users will move between a number of applications in the course of their job and will use them in different ways depending on the business process involved. If you do not have adequate resources in your organization to develop training materials, consider using an **external resource**.

## Upgrade

Before the actual upgrade can begin, you should verify the following:

- All testing and action items are complete
- All required hardware and software for the upgrade is in place
- All required user training materials are available

Installation of the hardware and software can probably be scheduled to happen towards the end of the configuration and test phase. Configuration of the production environment is best left until all testing has finished and any action items are complete.

Schedule user training to happen as close to the actual upgrade as possible.

You should produce a very detailed schedule for the actual upgrade paying special attention to any task dependencies. Most organizations are careful to schedule an upgrade a time when the risk of interrupting normal business activities is reduced. The end of a fiscal quarter should be avoided if possible. You may also be fortunate to schedule your upgrade to happen over a three day weekend. Even if this is not possible, you should ensure that the detailed upgrade schedule allows adequate contingency time should some disaster occur.

As in the testing phase, you should have the functional team members run some key reports in the production environment to be used to validate the upgraded environment **BEFORE** any transactions occur there.

Once your upgraded applications are “live”, there should be a final acceptance test that involves the operation of the key business processes.

## On-going Support

When the applications are “live”, you can expect a surge in the volume of support requests from users. Carefully prepared user training can help minimize the number of support requests, but unanticipated and untested scenarios will arise.

The key to reducing the support burden for your technical resources is ensuring that there is adequate user ownership of the upgraded applications. One way to do this is to identify a user “champion” within each functional area. This champion should become an expert in the applications as they relate to a particular business function. The functional champion should be the first point of contact when a user in their business function needs support. In this way you can:

- Filter support requests to ensure that only critical requests require technical support
- Build expertise and knowledge within the business function

To supplement the technical support provided by Oracle, you may also choose to use **consulting support** from time to time, particularly when functional issues arise.

## Conclusion

The decision on whether or when to upgrade can be difficult. Some organizations will be forced to make the jump as Oracle de-support older releases of the applications. With some careful analysis and planning, it is possible to work through an upgrade and leverage new functionality to support your business objectives.