



White paper

ORACLE E-BUSINESS SUITE UPGRADES

Jan-Willem Phylipsen & Ron Bruijn



mcx.nl

Abstract — IT landscapes are very much dominated by upgrades in all sorts of areas. This stems from the fact that upgrades are part of the regular software life cycle. Upgrades used to be slow-paced processes, especially in the Enterprise Resource Planning (ERP) application arena. Moreover, nowadays upgrades are much more frequent than ever before [1]. This is caused by two important factors:

- **Changing attack vectors:** applications are more and more connected to the Internet, making them vulnerable to outside attacks. To counteract this, frequent security updates are needed to keep applications and the information they process safe;
- **Increased market demand:** new application functionalities are needed at an ever-accelerated pace to adapt systems to new market demands resulting in necessary upgrades.

With each upgrade, the testing effort is needed to check whether all required and desired functionality is still working properly. These testing activities can be quite intense and stressful at times. This white paper will not make the lives of the people testing an application easier. However, the authors strongly believe that by shedding light on the Oracle E-Business Suite ERP application upgrade process, a level of understanding of the process will result in the acceptance of the consequences such an upgrade has in terms of possible downtime and effort.

Introduction

This white paper focuses on the upgrade process of an Oracle E-Business Suite environment. An Oracle E-Business Suite upgrade is a major release that in general will be supported by Oracle for many years to come.

An upgrade from 12.1 to 12.2 seems small in terms of version numbers. However, such an upgrade is not a bundle of patches but a complete new built-up of the application software stack, the underlying database including a new data model. Once on release 12.2, it is possible to separate the technology stack (Oracle RDBMS and Middleware) updates from applications code updates, allowing to use the latest available technology while retaining the current applications code level without a major upgrade. This is called the Continuous Innovation release model.

Before the actual production upgrade can be executed, several test upgrades, called iterations, must be performed. These iterations are required to create a functional and technical upgrade 'run book'. These cookbooks have a fixed part and a bespoke part. It is the latter part that is the primary reason for doing several iterations as part of an upgrade project. MCX has been doing upgrades since 1999 and has built extensive experience in this field of expertise. Based on this experience MCX has developed an upgrade methodology that delivers production upgrades that can be timed to the minute. Being able to accurately predict the duration of the actual production upgrade is crucial for planning the production upgrade. The downtime of the production environment should be kept at an absolute minimum.

This white paper starts with a description of the problem that arises in an Oracle E-Business Suite upgrade followed by an overview of the MCX upgrade methodology that is the solution to the problems that usually occur in an Oracle E-Business Suite upgrade. This paper finalizes with a conclusion and an overview of possible areas for future research.

II. Challenge

Oracle E-Business Suite upgrades are complex, intense, and time-consuming. This is partly caused by the fact that an ERP application like the Oracle E-Business Suite provides a lot of business functionality that is or can be used throughout an entire organisation. An average Oracle E-Business Suite upgrade, therefore, takes on average at least six months. Depending on organisation complexity this can even be extended to more than a year.

In the experience of MCX, two aspects are important during a project. On the one hand there the financial budget which is closely related to time. On the other hand, there is the quality of the end product meaning does the upgraded application work as expected.

A. Keep control of time and budget

The 'time and money' problem can be subdivided into several other problems. The authors in no way try to be exhaustive in that respect. The years of experience have revealed the following problems during an upgrade.

1) Prevent known issues from reoccurring

In every upgrade, small and large, issues will occur. Issues will be solved by patches, scripts, or modifications to the application setup. Each solution to an issue needs to be carefully recorded so it can be re-executed during any successive iteration.

2) Work in advance

When doing project management, one could argue that all the work that has been done in advance does not have to be done at a later stage. This approach gives 'working space' at the end of a project and can compensate for any unexpected delays. Keeping this approach in mind, experience has shown that in an Oracle E-Business Suite upgrade some steps can be done in advance or parallel to other activities. One of these activities is the execution of the 12.2 upgrade readiness report. This report shows whether all pre-upgrade steps have been completed successfully resulting in a smooth technical upgrade.

Oracle E-Business Suite Release 12.2 introduces a new set of compliance, code, and patching standards that customizations must adhere to. One of the important steps is to compile a comprehensive inventory of your existing customizations. Enhancements delivered with Release 12.2 may make some of these customizations unnecessary. There are also 12.2. standard violations that need fixing.

3) Keep production downtime to a minimum

The downtime of the production environment should be kept at an absolute minimum. This will depend on factors unique to your situation like the size of your Oracle E-Business Suite footprint, the number of customizations and integration, other IT changes you combine with the upgrade, and complexity of topology.

II. Challenge

B. Quality and Control

Next to budgetary constraints, quality is of great importance once you start planning the Oracle E-Business Suite upgrade. But how is the upgrade kept under control in terms of quality? With quality, MCX aims at a predictable upgraded environment based on the runbook that is created during the various project phases which are mentioned in II.B. Years of experience with these types of upgrades allow us to share information on how MCX deals with this quality and control problem. Again, this area is subdivided into several categories that are discussed below.

1) Plan the upgrades

Though this may sound obvious, planning your upgrade is not only vital for the first challenge as defined in A but also contributes to quality. Extensive experience in the field of upgrades has resulted in an optimised upgrade plan and a set of tools that simplifies the upgrade process. All this together results in an upgraded environment that delivers the required quality.

2) Production “State” Upgrade Iteration

Converting your software accurately and efficiently depends on the quality and makeup of the software itself. Therefore, each upgrade iteration is based on a current copy (clone) of the production environment. This will not only give valuable information about how the testing process should be structured and how long it will take to complete the upgrade but also ensures the software, including customisations, is always up to date at the start of each upgrade iteration.

3) Interim quality check

To validate the quality of a product, one or more checks can be done during the project phase forcing the product to adhere to the required quality requirements. In the same line of thinking an Oracle E-Business Suite upgrade should be treated in the same manner. The method used for this is the User Acceptance Test (UAT). This test is the final validation of system functionality, business processes, and data, and, more importantly, serves as a point of reference for the users to accept the new system. MCX defines this phase early in the project, establishing success/acceptance criteria for system performance which are validated during the UAT phase of the project.

III. Solution

To get an accurate estimate for the duration of the production upgrade, one should perform the upgrade. Though this sounds logical it is also paradoxical. This paragraph shows how an upgrade can be done, even multiple times, and be part of the iteration process.

A. Pre-upgrade tasks

Before the technical upgrade starts, it is recommended to fix custom code violations in your current Oracle E-Business Suite environments as much as possible. To identify those fixes, we run the 12.2 Compliance Reports which are delivered by an Oracle patch that has to be installed in the current 12.1 environments.

III. Solution

This helps the developers to fix custom code violations. It is important to rerun these Compliance Reports in waves to see how many violations have been reduced over time. As can be seen in Figure 1 this can be done in the current test environments and once agreed with the outcome, the new code can be

deployed into the current production environment. This is an optional step and can also be applied during the upgrade iterations into the newly upgraded Oracle E-Business Suite release. MCX strongly advises executing this step in the current production environment.

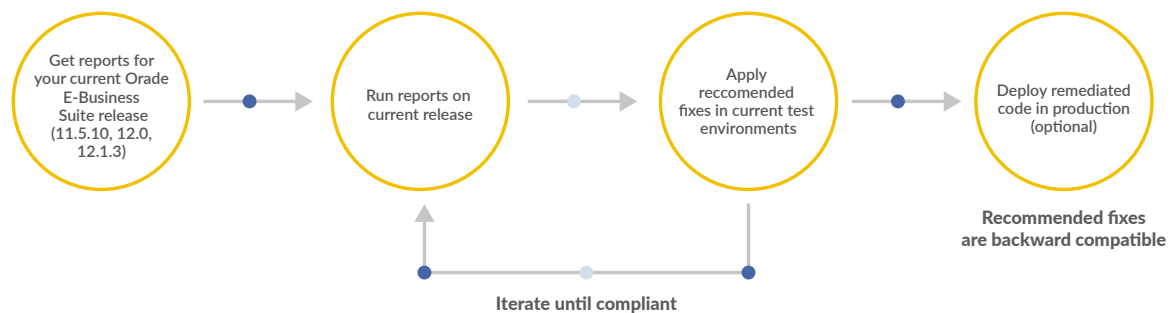


Figure 1 - Fix custom code violations

An upgrade to Oracle E-Business Suite Release 12.2 may require the upgrade of the operating system and hardware. In preparation for the Oracle E-Business Suite upgrade, MCX investigates if these changes are required and act in case this is needed. This is, of course, specific to each situation, and at the start of the project, a decision needs to be made whether an operating system and or hardware upgrade is needed. Because an Oracle E-Business Suite upgrade is for many years to come, MCX advises upgrading the underlying operating system and hardware at this stage to the most recent certified level.

In preparation for the upgrade, external integrations must be upgraded to releases or products that are certified with Oracle E-Business Suite 12.2. During the pre-upgrade tasks, MCX advises starting an investigation to which extent external integrations products like OAM, OID, OBIEE, etc. are used. The presence of these products might require them to be upgraded as well.

B. The upgrade process

Over the years, new features have been added to Rapid Install (the installer of the Oracle E-Business Suite) and AdPatch (the Oracle E-Business Suite patching tool) to ease installation and periodic maintenance. Rapid Install provides the latest certified versions of the Oracle E-Business Suite products, including technology stack components. During an upgrade, the installer creates a new file system for the application (middle) tier components and a new file system for the database. After the upgrade, Rapid

Install will be run again to configure the application tier. An upgrade also includes various manual steps like the execution of scripts or the application of patches. AdPatch will be used to apply the Oracle E-Business Suite patches, including the unified driver which upgrades the Oracle E-Business Suite Database objects to Release 12.2.0. When the application is at the 12.2.0 level, adop (new Oracle E-Business Suite patching tool) is used to apply all Oracle E-Business Suite patches and bring the application to the required level (12.2.10 as of writing).

III. Solution

C. Iterative test upgrades

As mentioned in the introduction several test upgrades must be performed to finally make the production upgrade a breeze. MCX creates a detailed plan containing all different tasks as described in the latest Oracle upgrade documentation. Based on this documentation the plan is amended to reflect the needed resources required for the execution of the upgrade. The result of this is a detailed timing of the upgrade that can serve as an input for planning the production upgrade. To speed up an upgrade as much as possible and to reduce failures the advice is to apply automation during an iteration as much as possible.

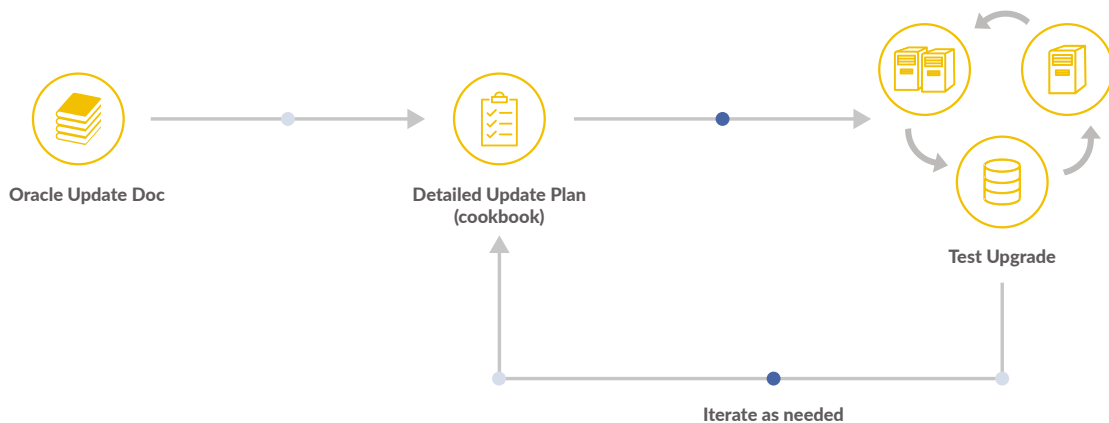


Figure 2 - Use Iterative Test Upgrades

The goal is to run the final production upgrade iteration within a weekend to keep production downtime at a minimum from a business perspective.

D. Inside the upgrade

The concept of iterations is used to gradually migrate to a stage where the actual production upgrade runs as smoothly as possible. But what happens inside an iteration? Before we come to that point it is important to know that each iteration starts with a full clone of the current production environment. This step is needed to be able to predict the timings more accurately and to determine the impact of any changes that are made in the current production environment and keep production downtime to a minimum as was identified in challenge 3. A high-level overview of this process is given in the picture below:

Interaction 1	Interaction 2	Interaction 3	Interaction 4
Initial upgrade environment	System intergration Testing	User Accepting Testing	Production Go-live Friday 5 PM to Sunday 3 PM

Figure 3 - Iteration based upgrade

Experience has learned that several iterations need to be completed before all problems and bugs have been fixed. Since the list of bugs and problems can be quite extensive, MCX advises customers to do at least four iterations during an entire upgrade project including the production upgrade itself. To get a global overview of how this iteration process works, a short description is given of what is done in a four iterations-based project.

III. Solution

1) Iteration 1: Initial Upgrade

The initial upgrade is the longest of all iterations. Everything needs to be sorted out in terms of software, hardware, and operating system. Upgrades of subcomponents like the database deserve special attention and need to be incorporated into the technical upgrade plan. When upgrading to release 12.2, one should consider an upgrade duration of at least one month and maybe even longer. The reason for this is that the technology stack in version 12.2 is completely changed accounting for most of the duration of this initial iteration. At the end of this iteration, an upgrade plan is created that describes all required steps including fixes to any technical problems that have been tackled along the way.

The fixes can range from applying additional patches, running scripts, or manually modifying files. When the technical part of the iteration is completed, it is time to investigate the custom-code and functional changes. This means that functional tests, repairing of the customisations, adjustments to the reports, and so on are executed. In this part problems or errors can occur that need a fix. This fix can be a patch, a script, or an updated piece of customised software that must be added to the upgrade plan to prevent the error from reoccurring in the next iteration. Close interaction between functional and technical project members is required to optimise the upgrade plan and not to miss any vital steps in the process.

2) Iteration 2: System Integration Testing

With all fixes incorporated in the upgrade plan, a new iteration is started. In this iteration the technical upgrade process, the custom-code fixes, functionality, and interfacing are tested. It is important to start with interfacing as soon as possible with other systems so that it can be investigated at an early stage if and where any issues may occur. A complete iteration test is completed during this phase and any findings are documented in the upgrade plan.

3) Iteration 3: User Accepting Test

Before the production upgrade is started a user acceptance environment is upgraded. This iteration is executed to allow a larger group of end-users to check the application for correct working. Next to this, this iteration is also used to time the upgrade from a technical point of view. The goal here is to have the upgrade completed, including all customer-specific fixes (the tailor-made part). The iteration tries to come as close as possible to the duration of the actual production upgrade. Also, this environment is integrated with other systems to test the interfacing.

Because more people are allowed in the test, more settings are checked which can reveal new problems. For these new problems, new fixes need to be implemented which in turn need to be added to the upgrade plan. If the number of additional fixes is quite extensive, the project members can choose to do an additional iteration. The additional iteration can be used to accurately determine the timings and test any negative interference between the newly applied patches and previously made changes for example. If this decision is not made, then the production upgrade can be planned using the timings of this iteration.

III. Solution

4) Iteration 4: Production go-live

All fixes have now been recorded in the technical upgrade plan and the actual production upgrade for the Oracle E-Business Suite environment can be executed. Since this upgrade is quite a technical upgrade, one needs to keep in mind that such an upgrade can take about 30 to 48 hours to complete. Normally the production upgrade including testing can be done on an (extended) weekend. Timing during the previous iteration will give a definite indication.

During the iterations, time is also spent on changing and testing the cloning process. This is important because the cloning method will slightly change because of the new technology that will be used in release 12.2.

IV. Post upgrade support

Although the aim is to fix all issues during a project, some issues are addressed after the actual production upgrade is completed. This choice is sometimes made to meet deadlines. The new cloning method that is required for the Oracle E-Business Suite release 12.2 is tested during the upgrade project but often non-production environments like test and development need to be in sync with

production immediately after go-live. At this stage, these environments are created by the new cloning method. All activities for operational administration are incorporated, like monitoring, backup, and restore, and must be officially transferred to the regular system administrators based on the Daily Agreed Procedures (DAP).

V. Conclusion

Upgrading to a new release as mentioned in this paper is a lengthy and intense process. By using the iteration based upgrade method an organisation can keep track of the project budget to start with. This method also allows for full quality control throughout the entire upgrade process. In our opinion, these two important aspects of an Oracle E-Business Suite upgrade should always be addressed properly. This method does just exactly that.

When an organisation has completed the upgrade to release 12.2, the organisation can benefit from the Oracle Continuous Innovation release model delivering ongoing applications and underlying technology stack updates without a major upgrade.

References

- [1] Bjelland, E., & Haddara, M. (2018). Evolution of ERP Systems in the Cloud: A Study on System Updates. *Systems*, 6(2), Systems, Jun 2018, Vol.6(2)
- [2] Note:1577661.1 Developing and Deploying Customizations in Oracle E-Business Suite Release 12.2
- [3] <https://www.oracle.com/a/ocom/docs/ebs-upgrade-to-122-resource-guide.pdf>

Jan-Willem Phylipsen. The first author has been working with MCX since 2000. Starting as a consultant/administrator over time he grew into the role of Account Manager. His secondary job description is Security Manager, where he oversees and manages all security related aspects of MCX. Before MCX, he started his working career at Oracle. He holds a master's degree in mathematics. When not working, Jan-Willem likes to spend time with his family, reading (scientific) books, walking the dog and do road cycling.

Ron Bruijn. The second author has been working with MCX since 2014. His career started as system integrator for the geographic information industry and during that period he came into contact with Oracle technologies. Nowadays he is part of a team of experts for the technical management of Oracle environments. His experience is in the field of Oracle applications and architecture on different platforms and industries. His role as project manager is to manage implementation, administration, upgrades and transitions projects for various Oracle products. In his free time, he is a fanatic diver, loves skiing in the Alps and trail running.



ENGINEERED FOR IT

mcx.nl