

Software Upgrade Tips and Tricks

1. Determine the importance of the software being upgraded. How long can the business do without the application if the production environment fails. How expensive in terms of labor, lost revenue, customer dissatisfaction, etc. would it be if the application is not available? The more people in your business and the business itself depend on the software, the more important it is to assure that there is no unexpected outage time. Full software upgrades are full of pitfalls and negative impacts. It is important to have a clear assessment to assure readiness for the event.
2. The more business-critical your application is, the more important it is to have a testing environment that mimics the production environment as closely as possible in order to test the upgrade before it goes live. The test environment should not only be on the same operational platform as the production environment, but have the same amount of processing power, the same processing space, the same connectivity to other applications, or tools that exist on the production environment.
3. Make sure that the application running on test is at the same version level, and the same patch level as your production environment.
4. Restore a recent backup of your production data to your test environment.
5. Make sure you have whatever you need to restore the test environment back to the pre-upgrade level if the upgrade fails. This includes data, database configuration, operating system and patch level, other software running on the test server, etc.
6. Make sure no one is currently using the test environment during the upgrade window, and allow a soak time on the test box before moving to upgrade the production environment.
7. The software salesperson probably said that the upgrade is relatively foolproof, but here are the things to look for once the upgrade has succeeded in your test environment. Complete testing is essential to the success of the upgrade.
 - a. Will the upgrade require end-user re-training and new documentation?
 - b. Is the hardware adequate for your company's use of the upgraded software? Do you have the capability to load-test the upgraded application?
 - c. Did all of the software configuration changes made in the previous version all come over successfully, or do many of them have to be reconfigured?

- d. Are field lengths and field functions all working as expected?
 - e. Are all of your reports functioning as before, or do they need to be re-written?
 - f. Is there a change in the way the user interfaces with the application? Do individual PC's need to be configured properly in order to use the new version.
 - g. If the upgraded application is to reside on a shared server, does the upgraded software maintain the same neutral relationships with other software running on the same server? Are all applications on the server functioning properly?
 - h. Are all upstream applications (those that may feed data into your application) compatible with the new release? Are those connections still functioning?
 - i. Are all downstream applications (those that receive data from the upgraded application) compatible with the new release? Are those connections still functions.
 - j. If your production is load balanced or clustered, is the upgrade compatible and functioning properly with the software that manages the environment?
 - k. If there are listening routines, cron jobs, backup routines, monitoring software, or scripts running in the production environment, do they still interact with the upgrade properly?
 - l. Will the upgraded application result in a significant increase in load on the networks connected to the server?
8. Once the software upgrade has been installed and thoroughly tested in the test environment, the implementation plan to install the upgrade should be finalized and contain at a minimum the following elements:
- a. A generous upgrade maintenance window that includes last minute backups of the application, database, and data, time to upgrade the software and the data if appropriate, time to function test and sanity test the application, and time to back the whole thing out and restore to the prior version if necessary. This window must be scheduled with the support of the customers using the software.
 - b. Who is doing the work, and a time frame for the work to be done
 - c. All specialized experts who must attend or be on call for support. These should include technical experts for the platform and operating system, database administrator, knowledgeable experts for the other software applications

- sharing the server, and customer representatives to test the upgrade before the start of business to assure no unexpected surprises.
- d. Step by step instructions for upgrading the software.
 - e. Step by step instructions for testing the various aspects of the software and its relationship to other systems and applications.
 - f. Step by step instructions for backing out the upgrade and restoring the application to full functioning at its previous version level.
9. Those businesses that run their production application in a clustered environment should drop one server out of the cluster and upgrade the application on the dropped server first and then fail over to the upgraded application. Run the application on the upgraded cluster for a predetermined period before upgrading the remaining servers in the cluster. If anything should fail on the upgrade application, it is easy to fail back to the servers that have not been upgraded in order to restore service.
 10. For complicated upgrades with multiple experts and customers involved, open a conference line for the duration of the upgrade so that the coordination of activities can be facilitated.

Happy Upgrading!!