



Constructing and Executing an Upgrade Plan

How American Bank of New Jersey smoothly upgraded their enterprise level document management system.

Software is never static. We all know that the software we depend on needs to be upgraded periodically in order to ensure that it continues to meet our business needs. When end users think of an upgrade, their thoughts tend to turn toward the idea of inserting a disk, clicking on the “setup” icon, and letting the upgrade run automatically. While that’s a perfectly standard procedure for Desktop software, Enterprise level document management systems (and indeed, enterprise-level software in general) demands far more preparation and consideration than standalone Desktop applications. Any key component of the business on which employees depend must be given the proper care during an upgrade, and that means proper planning. Even if a system is advertised as being easy to upgrade, resellers and service providers would be wise to draw on their documented experiences with the software in question in order to formulate an effective upgrade plan. This article is, in part, a case study that draws on CHAMPION’s recent experience upgrading the TokOpen document management system at one of our clients, the American Bank of New Jersey (ABNJ). However, the principles are universal and can readily be applied to many different software implementations in a variety of different industries. In CHAMPION’s case, the upgrade consisted of a single step up three minor version numbers of TokOpen (from 2.0 to 2.3).

When you are performing an organization-wide upgrade that will affect the majority of users (or all users, as is typically the case), an upgrade plan is an absolute necessity. The upgrade plan should take the form of a single document that both the service provider and the client, will use during all phases of the upgrade, starting with testing. Client participation is critical to a smooth upgrade. During the planning process, the service provider is responsible for setting expectations with regard to both the schedule, the necessary participation on the part of the client, and the results. It is highly preferable not to keep the client in the dark about any phase of the upgrade; hence the sharing of information in the form of an upgrade plan.

Your plan should take a chronological approach, beginning with the testing phases and ending with the actual execution of the upgrade. The parties

CHAMPION WORKFLOW SYSTEMS INC.
39 Route 46 East — Unit 803
PO Box 362
Pine Brook, NJ 07058
800-WORK-911



responsible for performing each task should be clearly outlined. In CHAMPION's case, we knew based on our previous experiences upgrading the software that we needed to thoroughly test the client's database in the lab against the new version. While CHAMPION was capable of running through most of the tests in a fairly short timeframe, it was still necessary for testing to occur against the client's environment. There is no substitute for being able to validate the new software's operation in your client's production environment. While you may not always have this opportunity, in this case we were able to install the new version of the software on the ABNJ test server and then run an upgraded version of the TokOpen database on the client's production server. Because we established this as a possibility during the planning phase, we constructed the testing plan so that while CHAMPION was testing the software in its lab, the client could perform their own testing in their test environment in concurrent fashion. The take-home point here is that certain items in the plan, if they are not dependencies, can be performed simultaneously if you have your client's cooperation. This is again another reason to encourage client participation. In CHAMPION's case, this approach proved fruitful because the client was able to identify potential issues in their environment that we were able to address before the upgrade took place.

Our testing also allowed us to confirm that the software's older PC client would be compatible with the new database, allowing the bank to take a piece-by-piece approach to upgrading its branches instead of being forced to undertake a costly, in-place upgrade of all clients at once. Be advised, however, that taking this approach will require you to conduct testing on the older version of your software as well as the new version. Both the lab and site testing that was done in CHAMPION's case included both versions of the client software. Never assume anything about your client's environment, or their existing software. Even cases where a provider performed the initial installation of the software, that provider cannot guarantee that his or her client's environment and infrastructure has not undergone changes as the result of completely unrelated actions on the part of the client's IT department. Again, client participation in the planning phase is essential to prevent nasty surprises from cropping up when you are on site, doing the upgrade. Success definitely requires a team approach.

Your upgrade, if planned correctly, should consist of at least 99% "boredom" and not more than 1% "terror". The actual execution of the upgrade, where the software is brought up to the current version and put into production, can be made to occur in as short a period of time as possible with an effective plan in place. In our case, since we had already performed an upgrade of the client's database in the lab, we were able to develop a set of SQL scripts that automated the beginning



phases of the upgrade. This freed up time onsite to address more critical aspects of the upgrade, and was one of the many benefits to us and our client that we were able to realize through effective planning. Overall, the upgrade went very smoothly due to the planning effort involved. While upgrading complex software is never easy, with an effective upgrade plan, the service provider and the client working as a team can make it look easy to the end user.