

TITLE: 9 words

Get a Winning Oracle Upgrade Season Using the Quarterback Approach!

ABSTRACT: 318 char

Upgrades, upgrades...too much customer down time. Find out how we shrunk our production upgrade schedule 40% from our estimate of 10 days 12 hours to 6 days 2 hours using the Quarterback Approach. So your upgrade is not that complex, come anyway. This approach is scalable to any size project and will be extremely valuable.

EXECUTIVE SUMMARY:

When it came time to advise our executive management of the production downtime required to upgrade JPL's Oracle 10.7 version to 11i, it was not a pretty picture. Our initial implementation schedule was going to take our production environment down for 10 days and 12 hours. While that time did include two weekends, it still brought our production environment to a halt for one full week! Totally unacceptable when you are preparing for future launches to Mars, (let alone trying to pay your employees.)

Oh, did I forget to mention, this schedule was working 24 hours a day, 7 days a week.

So we were tasked with developing a methodology that would do it for less, much less! Thus, the Quarterback Approach evolved.

The Who: Our teams consisted of our Sponsor, Management, Quarterbacks, Database Administrators, System Administrators, Data Security, Migrators, Developers, and Functional Users.

The Planning: We met with all the teams; looked at their detailed schedules; reviewed the Category 1-6 steps, integrated the schedules; identified tasks that could be completed in parallel; evaluated long running jobs; streamlined the timeline using Microsoft Projects indicating dependencies; and documented, documented, documented. Once all the planning was completed, we reviewed again individually with all the teams, made adjustments as we continued to flush out the minute tasks. Then we fine-tuned it with a dress rehearsal.

The Executing: To make the entire schedule work we used a Quarterback. This approach had one person on-site managing the schedule during the actual upgrade. That person was the primary contact of all information coming in and going out, and for orchestrating the activities. Everything was funneled through this person. We planned our resources in great detail, but we also used the "just-in-time" philosophy and had all of our teams staged and ready to adjust the work schedule on two hours notice. We had several early warning indicators that allowed us to contact the next team "up" to come in several hours early when we went ahead of schedule. This allowed for no lapse of productivity.

The Communicating: With a large group of people to implement or support the project, it was extremely important to have a detailed communication plan with various contingencies to reach team members and alternates on a 24 hour basis. (Oh, I forgot to mention, we implemented the week before Christmas.) With the use of various electronic tools and some old fashion persistence, we established Ground Rules for all team members to follow. This set the expectations for all participants.

The Contingencies: Planning for an Oracle Upgrade is very similar to Disaster Recovery Planning. The primary focus was pro-active planning, risk assessment, and contingencies. This

too was a critical part of the Quarterback approach. We were prepared for anything, even a snowstorm in Southern California.

The Tools: Some of the tools we incorporated to make our project a success included:

- Microsoft Projects Timeline Schedule
- Detailed Work Schedules with Alternates
- Contact Lists
- Pagers, Cell Phones, Home Phones
- Voice Mail & E-mail
- Call in Status Line
- Ground Rules incorporating the hand-offs, communication plan, when and where to check in, how to check in, along with other valuable information
- Housing Arrangements for those living out of the local area
- Stocked Pantry
- Quarterback & Duty Manager Shifts
- On-site Personnel Only
- Coordinated support with Oracle Support
- And others

Come and learn the Quarterback Approach we utilized to shave off 40% of the projected implementation time.

So your upgrade is not that complex, come anyway. This approach is scalable to any size project and will be extremely valuable for all teams and future upgrades...