Staying Current on RMS Version 10 – The Longs Drug Story

Steve Dorfman, RMS Development Team Lead, Longs Drug Stores
Amy Leibowitz, RMS Development Team Lead, Longs Drug Stores

Introduction

Longs Drug Stores (Longs) operates a retail drug store chain on the West Coast of the United States and in Hawaii, with over 520 stores. Four years ago, Longs partnered with Oracle Retail (Retek), beginning a company wide Supply Chain Initiative (SCI) to modernize and improve supply chain processes. To date, Longs has implemented Oracle Retail solutions in four phases starting with Warehouse Management using Oracle Retail Warehouse Management System (RDM). In the second phase, Longs implemented Foundation Data and Pricing in the Retail Merchandising System (RMS). The third phase brought PO Management into RMS as well as Allocations using a custom front end for the Oracle Retail Allocation Calculation engine.

For the fourth phase, Longs is in the process of rolling out a store solution integrated with RMS. Longs will embark on two more phases of this initiative after the store solution is rolled out to all locations.

Throughout the four year phased implementation, Longs Drugs has completed three major upgrades and one minor upgrade within the RMS version 10 application set. Along the way, Longs has developed a successful methodology to handle such upgrades. This process supports business needs, prioritizes functional changes, tracks modifications and allows for other projects part of the supply chain initiative to continue during the upgrade.

Longs' Upgrade Process

Longs challenged its team to stay no more than three versions behind the current version of RMS while continuing the aggressive SCI project. This process follows the standard software development life cycle.

To begin the development process, Longs creates a merged spreadsheet of all of the defects supplied by Oracle for each release included in the upgrade. This document is used to track all research, notes, and customizations for each fix. Next, the process of applying the code change begins with developers applying each patch to each different code type (for example form changes, package changes, batch changes).

The developers review each piece of code to determine if there are any custom modifications, if the changes will impact Longs' business users, if there will be a performance impact because of this change, and if the fix resolves the defect. If there are no custom modifications, then the code is checked into the code repository. If customizations exist, further impact analysis is performed to ensure that the new code will not impact the enhancements made by Longs. If necessary, additional code changes are completed to ensure the new code will work properly with the existing code.

While the developers are applying the code, systems analysts review the SIR documents from Oracle to determine if the changes have any impact on Longs' business processes. If there is an impact, the analyst creates a business impact write up that describes the problem, the changes, and how it will affect Longs' business users.

Additionally, the systems analyst indicates any new test cases that should be executed during testing. This document is then turned over to our business and quality assurance (QA) teams in order to communicate the changes and new testing requirements.

Once development is complete, the QA process begins. Longs first level of testing is unit testing, which is performed by the development groups before deploying to a test environment. After completing unit testing, the code moves to a testing environment. At this point, Longs' QA team executes test scripts to system test RMS. These test scripts cover all business uses of RMS at Longs. The QA group also executes similar regression test scripts on all applications that send or receive data from RMS to ensure no there are no problems in these systems due to the upgrade.

Performance testing occurs once regression testing begins. Longs tests any batches or forms flagged by developers or systems analysts during development or unit testing. Additionally, Longs tests processes where testers experienced an undesired run time. This testing is completed in an environment similar to Longs production setup so that an accurate production run time can be deduced.

The final testing steps are user acceptance testing (UAT) and mock go live testing. Business analysts assigned to Longs' SCI project execute tests which mimic real life business processes and data setup as closely as possible. Mock go live testing executes the implementation steps in a test environment. This ensures that each task in the process is correct. Additionally, this allows the team to estimate the duration of the actual production implementation. This enables accurate outage information to be disseminated to all effected groups.

Longs uses a T-minus plan to coordinate the actual implementation and the tasks required for the days leading up to and following the install date. This plan includes every task required to complete the installation of the upgrade as well as the day and time it must be completed. Additionally, this plan tracks any dependencies between various tasks so as to ensure that no steps are completed out of sequence. During the implementation, a coordinator follows the plan and contacts the owner of each task for progress and communicates when the next step can begin.

There are many benefits from upgrading RMS frequently. First, Longs is able to leverage Oracle's expertise and development resources, thus freeing Longs' team members for other projects. With a more up to date code base, it is easier to apply a critical fix from Oracle as less analysis and testing is required. Additionally, Longs spends less time analyzing and maintaining service requests (SRs) with Oracle.

Lessons Learned

In Longs' experiences upgrading, there have been many lessons learned. Overall, Longs has found that putting off an upgrade only complicates the development, testing, and installation process. Additionally, business teams need to be involved early and often – to analyze business impacts and communicate these changes to the end users. Furthermore, documentation above and beyond what is provided by Oracle in terms of technical documentation is extremely helpful for testing and long term support. In particular this is useful for extract, upload and download batch processes.

In planning each upgrade, Longs estimated that each resource will be available to work on the tasks approximately 70 percent of each day, with the rest of that time being spent on other problems or tasks that are unplanned.   Allotting only 30 percent for other projects that may occur after the upgrade project is planned might not be sufficient for the changing information technology demands of a retailer.

In our most recent upgrade, we found that there were at least five new projects assigned to our team after the upgrade was planned that delayed our work on the upgrade. Additionally, in planning future upgrades, Longs will also take into account the number of versions to be applied by the upgrade – if it is more than three, the duration of development needs to be increased.

In our next upgrade, Longs will try to minimize the big bang approach by applying more functionality individually. For example, Longs will break up the upgrade into releases by functionality (cost change, price change, physical inventory etc.). We will also investigate other ways to minimize our outage duration as more and more stores demand RMS to be available as much as possible. Lastly, we will plan for more time to handle the details, as our version of RMS has more and more customizations that require analysis and careful application of the upgraded code.