The Power of Process, People, and Tools When Testing a Complex Integration Landscape for a Very Large Initial Retail ERP Implementation

Nicki Nicolo
Nike, Inc.
Nicki.Nicolo@nike.com

Abstract

Anyone who has ever led the integration testing effort for a large project with a complex integration landscape has had a difficult time finding guidance on what is necessary to succeed in this type of enterprise. There is scant information available that provides any meaningful assistance. This paper presents a real life project as a case study on how the nexus of process, people, and tools is the foundation for enabling a high quality delivery.

The Nike Retail Enterprise Resource Planning (ERP) team created a number of tools including complex data and transaction trackers to allow communication across partner systems and business functional tracks. Additionally, there was a well-planned test schedule. The Retail ERP team utilized existing off the shelf tools such as Quality Center for test requirements, test cases, test execution, and defect management.

The implementation for this case study involved the initial implementation of a Retail ERP system with 26 partner systems, approximately 150 interfaces, 5 integration tools, and systems belonging to other corporations.

The paper will close with a look at ideas for improvement based on the issues the Retail ERP team faced and what the Retail ERP team learned along the way.

Biography

Nicki Nicolo is a lead software QA engineer at Nike, Inc. in Beaverton, OR. Nicki has worked at Nike for 26 years starting out as an MVS systems programmer and also working as a project manager for 6 years. She has 33 years of experience working with both software and hardware. Nicki has a Bachelor of Arts degree from Mount Holyoke College.

Copyright Nicki Nicolo 2012
1. **Introduction**

This paper is a case study of integration testing, from soup to nuts, for a “greenfield” Merchandise (Merch) Retail Enterprise Resource Planning (ERP) implementation at Nike, Inc. We will explore the nexus of process, people, and tools as the foundations for enabling a high quality delivery of the integration portion of this implementation. The paper will close with a look at ideas for improvement based on the issues the Merch Retail ERP team faced and other key learnings along the way.

The main functions of this Merch Retail ERP system are financial and product planning for stores, procurement and distribution of product for stores, sales and other operational reporting for stores, inventory management for stores, product movement between stores, pricing changes for product in stores, nightly financial reconciliation processing, processing invoicing and product returns with vendors, and providing general ledger information to Nike’s financial system of record.

When I was asked to take the integration Quality Assurance (QA) lead position of the Retail ERP implementation, I spent some time in research looking for information on how to test integration in a large and complex environment. I was looking for both test strategies and case studies that would reveal the types of issues our team might face. I found nothing. I also attended a flash session at the Pacific Northwest Software Quality Conference (PNSQC) that year and asked for assistance. People talked to me about the issues I could expect to face. However, no one pointed me to any literature in this area. It’s my hope that this paper will be the beginning of a greater body of work in this area.

2. **Background**

The project for this case study involved the initial implementation of a Retail ERP system with 26 integration partner systems, approximately 150 interfaces, five integration/middleware tools, and some partner systems belonging to third party vendors. The Retail ERP system was designed and constructed along the lines of eight business functional tracks (Master Data, Finance, Purchasing and Distribution, Inventory Management, Sales Analysis, Financial Planning, Assortment Planning, and Reporting). Testing was structured to follow the business processes under each of those tracks.

Eight of the integration partner systems were considered core partners. The core partners are other Nike systems such as global store systems (brick and mortar), digital stores, our wholesale ERP system, our Warehouse Management Systems (WMS), digital store WMS, our sales tax system, and our markdown optimization system. Retail ERP test efforts needed to be tightly coupled with those of the core partner systems because any severe issues with these integration points could result in major business impacts.

The program management team determined that there would be four testing cycles for the entire implementation. Quality Assurance Streamlined Test (QAST) was the early testing phase and was scheduled to last five weeks. The purpose of QAST was to ensure that the system and interfaces were ready for end-to-end testing. The integration test effort during QAST utilized grey box techniques to validate the interfaces functioned according to specifications. System Integration Test (SIT) 1 and SIT 2 were the two end-to-end testing cycles. Again, testing was based on grey box techniques. SIT 1 was scheduled for six weeks, while SIT 2 was scheduled for seven weeks. SIT 2 actually continued through User Acceptance Test (UAT) due to late code deliveries and major defects that needed to be resolved. UAT was the user acceptance cycle and lasted for three weeks. Our focus was to ensure the middleware transformations provided the proper mapping of data fields between integration points and created any derived data correctly. We also validated error handling.
At Nike, unit testing is performed by the application engineering teams prior to the code being delivered to the formal QA test environments.

The QA and implementation teams were comprised of onsite and offshore consultants, and Nike employees. A Nike employee led the effort for integration testing and created the strategy and plan for integration testing. The functional track leads were consultants and were responsible to make sure that the Merch Retail ERP tests were designed and executed correctly for their respective tracks. Our test partners in other organizations were responsible for determining and executing any additional testing they deemed necessary. They were also responsible for determining the impact of defects to their applications.

Since the majority of this implementation’s integration relied on batch processing, we combined the test efforts for integration and batch. The batch system consists of scripts that trigger the running of various processes. The batch system holds the scripts and determines when they should be executed, either based on a schedule (day and time) or some other trigger such as the completion of another batch job that is a predecessor.

Combining integration and batch testing allowed us to work efficiently. However, this meant that the development teams had to tightly coordinate the delivery of interfaces and the batch jobs to run those interfaces. The test team was trained on the use of the batch tool so they could run all jobs independently. In general test teams at Nike must request the batch team to run the jobs for testing. However, testing is not the batch team's priority.

3. The Power of Process, People, and Tools

Here are the major points revolving around people, processes, and tools that were key to the success of the Nike Retail ERP QA team’s efforts:

- We implemented a single point of contact process that ensured cooperation and accountability with our partner systems, both for testing and for defect remediation. It also helped because some of the teams involved are very large and project members outside of that team may not know who to contact.
- We tested the interfaces as early as possible so that we had a solid platform for end-to-end business process testing. Initial integration testing used black box techniques to find transformation mapping errors.
- We created a test data tracker that allowed us to have clear visibility to test data as it crossed system boundaries and also ensured that testers were not stepping on each other as test data was coordinated in the tracker.
- We managed all code promotion tightly so we were testing on known platforms.
- We had a shared test calendar so test activities were well coordinated and well known across all of the teams involved.
- We had a dedicated team room for formal and informal meetings and triage work with representatives from all the involved teams.
- We had a well run defect management process. This allowed us to triage quickly and prioritize fixes.
- We coordinated closely with our key partner system test teams via the daily stand-up meetings where participation was mandatory. This allowed us to change and discuss daily plans, outstanding issues, and key fixes that needed to be tested.
- We clearly defined test and defect responsibilities across the various teams so there were no grey areas.
4. People and Roles

Early on in the project we utilized the **Single Point of Contact (SPOC)** concept with our partner system test teams and partner system development teams. This was useful as some of those organizations are very large and our team had limited knowledge of who we needed to contact for various tasks and activities. As the project progressed we lowered the level of SPOC touch points to the various system functional levels. This allowed us to operate efficiently, in the face of complexity that would have otherwise been very difficult to overcome.

Our project team was trained on our processes and tools. The team also gave frequent feedback that led to ongoing improvements in our processes and tools.

**Merch Retail QA Manager** - Overall responsibility for Merch Retail Quality Assurance.

**Merch Retail QA Project Manager** – Maintain QA project plan and facilitate resolution of issues and mitigation of risks.

**Merch Retail QA Test Manager** – Manage and track the execution calendars and provide weekly status reporting.

**Merch Retail QA Leads (Functional Lead, Integration/Batch Lead, and Reporting Lead)** - Manage the execution of tests, coordinate with partner system test teams, provide inputs to Weekly Status reports, verify test results and defects, manage issue resolution and follow up with partner system QA, and track daily progress and raise issues.

**Defect Triage and QA Build Mgmt Leads** - Schedule and lead Defect Triage Meetings and approve transports/builds prior to move to QA environments to ensure a stable test platform.

**Partner System QA Leads** – Manage the execution of partner testing, coordinate with the Merch QA Integration Lead, and provide Weekly Status reports to Merch QA Integration Lead. Some of Nike’s technical teams do not have separate QA roles so this role was taken by a business system analyst with testing responsibilities for that team.

**Testers** - Execute test cases, record results, report defects, and validate fixes. Some of Nike’s technical teams do not have separate QA roles so this role was taken by a business system analyst with testing responsibilities for that team.

**Business Systems Analysts** – Consults with QA Leads and participates in defect triage meetings. These team members can clarify issues/defects and drive their resolution with the AE teams.

**Subject Matter Experts** – Business team members who consult with QA Test Leads and participate in defect triage meetings. These team members provide the business perspective as to the criticality of defects.

**Application Engineers (Batch, Environment, Integration)** – Consult with QA Testers, participate in defect triage meetings, fix defects, and prepare for code migration. Their primary job is to fix the defects and ask questions if they don’t understand the defects.

5. Tools

Given the complexity of this project and the large number of people who worked on the project from a number of teams, most notably Merchandising Retail, Wholesale, Global Store Systems, Digital,
Warehousing systems, Markdown Optimization systems, and the Retail tax system, our toolset enabled both our people and our processes to help deliver a successful project.

QA Contact List

This list resided on the program Sharepoint site. The contact list names spanned all Nike technical teams across departments and all external companies involved in the integration testing effort so anyone on the QA team could easily find the correct person and contact information.

Data Tracker

This was a tool the Merch Retail QA team developed to allow us to track test data and transactions as they passed through all of the systems involved in integration. When I was seeking insights on what major issues this kind of integration testing would face, test data management was by far the most common point.

Given that data attributes are referred to differently by different Nike teams from different departments and business areas because of system differences, this allowed us to speak a common language and to ensure transformations were functioning properly. This tool started as a very simple spreadsheet to track product data as it flowed through various systems. However, the team refined the tool and morphed it into a very complex and complete tool. This provided great details during root cause analysis for defect resolution.

The Data Tracker was used to track fourteen critical data areas ranging from master data for retail stores, warehouses, finance, and product, to specific inventory to be used for end-to-end testing over all the wholesale warehouses and vendors to order transactions, to track orders from placement to receipt to invoicing and exception situations.

The Data Tracker was designed and built around the data flowing through the various business systems involved in the overall integration. Some of the systems involved were created by commercial software vendors. Others were created by Nike application engineering teams. And another group of systems was created by other external businesses. The interfaces involved were either created by Nike’s middleware engineering team or in a few cases used the vendor provided functionality of certain systems.

Test Management

The team used an off the shelf software package to create test scripts, track test execution, and manage defect resolution. We trained all of the project teams on how to use this tool and created specialized views of defects relevant to various development and business analyst groups. This tool enabled our defect management process and our metrics and statusing processes.

Test Calendar

The Merch Retail QA team manually created calendars showing detailed, daily test execution plans for each test phase (QAST, SIT 1, and SIT 2). This tool helped all the test teams stay coordinated and let us understand how pre-requisite testing factored into the overall schedule. After the initial release, we enhanced the test calendar creation so that it can be created using the planned end and start date fields for the tests in the test management tool.
Batch Execution

The Merch Retail QA team was trained so they could manually execute batch jobs using the batch management system. This enabled the test team to speed up test execution as they did not have to wait on hand-offs to and from another team. There were also periods when the batch management system ran on the automated schedule to ensure the schedule was set-up correctly. Normally, the batch support team must run batch for testing. However, production support is batch support’s first priority.

QA Team Room

The Merch Retail QA team had the use of a large, private, conference room which was used for both formal and informal meetings. Defect triage meetings and stand-up meetings were held in the room. It was also available for SWAT teams comprised of application engineers, business systems analysts, and QA engineers from multiple teams working to understand defects and find solutions quickly. Public conference rooms are often not available on short notice so that aided our efforts to reduce resolution time.

6. Processes

These key processes enabled the project team members to deliver on quality and keep the entire team apprised on the state of the project in terms of quality. For a number of reasons, QA processes within the technology teams at Nike are not standardized. Because this project spanned a number of Nike technology teams clearly defining, communicating, and utilizing this consistent set of processes helped everyone work together.

Design Reviews

The Merch Retail QA team was involved in review meetings for the integration high level design documents, the functional specifications, and the technical specifications. This provided invaluable assistance in providing the critical information for creating our test cases as well as ensuring that those documents provided clearly testable functionality.

Test Execution Management

We used our test execution management process to coordinate testing across teams and to provide daily metrics reports. We also asked our partner system test teams to provide similar reports. By and large, integration test management was done as part of the larger testing effort. We did track batch testing separately to ensure each job was tested at least once before the implementation went live.

Defect Management

We had a robust defect management process and we asked our test partners to report any critical defects in our system as well as using whatever process that team normally used. We kept a whiteboard of all very high and high defects that needed to be resolved in order to move into the next phase of testing or finally into business implementation. We made sure that the integration project manager was given immediate visibility to any defects that needed a quick resolution.
Code Promotion Management

We had a code promotion management process to ensure that code deployments and batch changes were done in a controlled manner on our test systems. Nothing could be moved without approval from a QA Lead. This ensured that our code base was always at a known state and alerted us when defect fixes needed to be tested. We also utilized soft and hard code freezes to provide a stable code base for “go-live”.

Daily Integration Stand-up Meeting

The Merch Retail Integration/Batch QA Lead ran a daily stand-up meeting for the core partner systems so everyone could be on the same page about what testing would occur on a daily basis and any issues could be discussed and escalated as necessary. This allowed all core partner systems to be aware of any issues with data conversion and system functionality.

7. Software Quality Strategy – Objectives, Activities, and Phases

Integration Test Objectives

- Ensure all necessary configurations and platform set-ups for integration functionality are correct.
- Ensure the outputs from the Merch Retail ERP integration links are compatible with partner systems.
- Ensure all data transformations meet business, technical, and legal requirements. Data must exit the source application/system in the expected format and arrive at the target application in the expected format and load into the target system correctly.
- Validate set-up of batch jobs.
- Validate batch scheduling set-up.
- Validate batch security set-up for batch logins/users for:
  - Retail ERP
  - Partner systems
  - FTP drop boxes

Planning Phase – Test scenarios

The functional track test teams created the signed-off test scenarios based on Merch Retail business processes and variations of these processes. These test scenarios were signed off by the Business System Analyst, Subject Matter Expert (SME), and Partner Systems QA. The test process scenarios were used to build the interface, as well as functional (system level), test cases for the project.

Development Phase – Test Case, Test Set Creation, and Interface Test Environment Validation

The QA team wrote test cases based on test scenarios, referring to the Business Process Documents and Functional and Technical Specification documents. Each application testing team (Global Store Systems, Markdown Optimization, wholesale, etc.) was responsible for identifying, gathering, and preparing test cases and test sets for their respective areas. Merch Retail QA worked with our partner QA teams to identify the integration points.

After all test cases for an integration development phase were created, test sets were created.
The Business Systems Analysts reviewed the test cases and provided the final sign-off for the test sets.

The Application Engineers specified input/output and unit test scenarios in unit test scripts which were a part of each functional technical design document. Unit tests only tested the functionalities/requirements specified in the business process and functional specifications. All unit tests were stored in the appropriate repositories.

QA team did not participate in unit testing. However QA team did refer to the unit test cases to refine test cases (if necessary).

Execution Phase

Retail Interface Testing – Early Test Phase (QAST)

All of the interfaces were new. Our chief concern early on was the accuracy of any transformations. We decided to use grey box testing for this phase as it met our needs. The goal of QAST was to enter SIT knowing that the critical interfaces were functioning properly prior to starting end-to-end testing.

Objectives

- Validate that business critical integration functioned according to business and technical specifications prior to the start of full systems integration testing to allow SIT testing to focus on end-to-end business process testing.
- Test integrations, including connectivity, between Retail ERP components and the partner systems that are immediately adjacent to Retail ERP.
- Validate scripts and security for batch jobs was set-up properly.
Retail Interface Testing – System Integration Testing Phases (1 and 2)

Objectives

- Ensure all business integration was functioning as expected in support of end-to-end business process testing.
- Greybox level testing.
- Validate scripts and security for batch jobs was set-up properly for untested jobs from early phase.
- Validate batch scheduling via turning on the automated scheduler functionality for a period that simulates week in the life.

8. Overall Results of Nike Retail ERS Integration Testing

- Our metrics techniques for QAST did not allow us to separate integration testing from other testing done during that phase. Metrics became more important and were refined during the SIT test phases.
- 923 integration test cases were executed during the SIT 1 test phase. These covered batch testing as well.
- 818 integration test cases were executed during the SIT 2 test phase. These covered batch testing as well.
- Most of the integration and batch issues detected post “go-live” were minor defects or were, in fact, requirements changes.

9. Issues Encountered During Integration Test Execution

- One of our critical issues was late code delivery. The last code delivery occurred during the SIT 2 Test Phase. This did not leave us with much time to get all of our testing accomplished.
- There were issues with core ERP functionality that led most integration testing to be blocked for a significant period. The team faced significant issues with product-related functionality until late into the SIT 2 Test Phase. Again this led us into a compressed schedule. The daily stand-up meetings were invaluable for allowing us to understand what testing to prioritize based on what code and what fixes were available to us as well as the business priorities of the partner systems in terms of various scenario executions.
- There were data quality issues with converted data that forced the team to used mocked up data for testing. This was not optimal as we were not able to encounter data-related issues that existed in the system. This led to a few defects not being discovered until post “go-live”.
- There was test resource turnover with some of the core partner systems that meant we had to go through repeated ramp-ups with those teams and created the risk that regression testing that might be required by those systems would not be identified and executed.
- The test team faced issues with the availability of core partner test system environments as they were not always in sync with Retail needs. Again we had to be flexible and nimble with our execution scheduling.
- The test team sometimes faced slow turn around of test tasks with some partner test teams. We did our best to work on relationships with those teams and escalate if necessary.
• The test environment ended up being shared by the training team and business analyst teams working on issues. This meant that QA faced testing in an uncontrolled environment. As much as we could, we tried to keep the test data separate to ensure that we understood which work caused the issues and to ensure that test data was not overlaid by teams who were not performing work for the test track.

10. Ideas for Improvement

• Data proved to be one of our biggest issues throughout this exercise. The systems involved are highly sensitive to different data permutations. Since this project finished, I have been exploring the use of combinatorial testing to provide deeper coverage in the areas affected by data sensitivity.
• Do not succumb to the temptation to test with mocked-up data because eventually the issues masked by using mocked-up data will surface. It's far better to find these issues earlier rather than later.
• Implement better change management processes for test scenarios and test scripts. This was difficult given the need to balance getting testing completed within tight windows and ensuring changes were reflected. We implemented a clean-up effort after the implementation went live so that our test scripts would be usable in the future.
• There was one major issue with performance in one set of interfaces post “go-live”. In the future, integration should be more tightly coupled with the performance test track.
• When we started testing we did not understand how order processing testing worked with Nike’s wholesale and warehouse management teams. We badly underestimated the time it would take for orders to be placed and fulfilled by those teams. In the future, we should engage our test partners to understand their test processes better.

11. Acknowledgements

I would like to acknowledge the Retail ERP test manager and Retail ERP functional test leads who took my concept of a test data tracker and created a tool beyond anything I initially imagined. This tool was innovative and strategic.

I’d also like to acknowledge the leads and project managers from our core partner systems for their time and efforts above and beyond, and their cooperation to ensure the success of this portion of the project. Without them, our final results would never have been this good.