

Viewpoint

Building a Winning WMS Testing Strategy

There are several factors that play a part in a successful Warehouse Management System (WMS) installation. Among those factors are package selection, design documentation, configuration, testing, and training. Testing is one key piece that is often underestimated and poorly planned. Crafting a winning test strategy requires an intentional commitment to early scheduling, thorough planning, and disciplined execution.

Scheduling

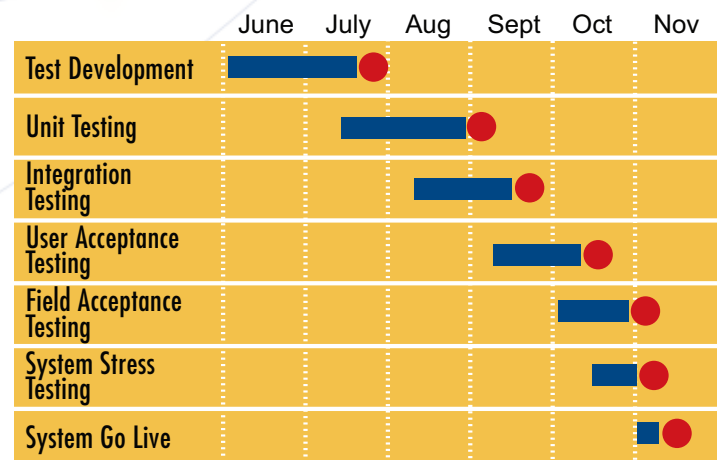
Developing a test schedule is an exercise that should happen in the infancy of a WMS project. The following steps should be followed during this part of the project:

- Identify the specific individuals who will have a role in each part of the testing. An inability to assign resources is a good indication that some stakeholders may not be fully committed to the success of the project.
- Develop contingencies in the schedule in the event that issues are discovered that could delay the go live date. It is naïve to think that testing will run smoothly without discovering bugs big enough to incur delays. It is also important to receive commitments from the WMS vendor and host interface team on how long issues will be addressed well in advance of the start of testing.
- Identify the type of hardware needed to carry out each phase of testing. This could include printers, RF devices, wedge scanners, and bar code labels.
- Plan for the test material that will be needed on-site for the field testing. It is not difficult to set up dummy inventory, but it can be very time consuming.

- Plan for integrated partnerships with the functional design, configuration, and training teams. Testing should not occur in a vacuum. For example, plan to use the functional design document as a guide for creating test scenarios. Also, work in close proximity to the configuration team in order to stay abreast of issues and discussions. Finally, the testing team should plan to provide their step-by-step test scripts to the training team so that operating procedures can be more easily established. The intent of the integrated partnerships is to shorten the time required for testing and training.
- Hold a kickoff meeting where testing terms and guidelines are defined, the testing approach is presented, and the schedule is published. It is important that this meeting include all the resources identified as having a role in the testing.

The following schedule with overlap between test phases provides a high level illustration of a testing timetable:

Test-phase Overlap



Test Plan Preparation

Preparing test scenarios or cycles can be an exhaustive process. For that reason it is important to break the test development tasks into manageable parts. Additionally, there are usually aspects such as data prep that cannot be prepared as early as the actual scenarios. The following points offer a road map for the test plan preparation:

- Determine what type of testing will be required. Typically, there is unit testing of the WMS functions, integration testing of the WMS with host and material handling systems, user acceptance testing of complete inbound/outbound processes, field acceptance testing, and volume/stress testing.
- Determine what system environments are available for testing. It is not uncommon for multiple environments to be available and used for different phases of testing. Obviously, any stress testing should occur in a system environment that mirrors production as closely as possible.
- Determine the migration plan from one system environment to the next. For example, configuration may be carried out in a development system and migrated to a testing or QA environment for actual testing.
- Start with building simple test cases. These test cases should represent basic functions such as pallet building, cart picking, or trailer loading. It is important to use the functional design documents and any modification specification documents to identify what functions and exceptions are relevant.
- Once test cases are defined, then it is possible to string them together to form test scenarios or cycles that are more process related. For example, an inbound scenario may start with a test case for scheduling an appointment and continue through to a test case for pallet putaway. It is a good idea to have test cases and scenarios defined as much as possible prior to the start of unit testing.

- After test cases are compiled into test scenarios, then it is necessary to document or “script” the step-by-step keystrokes required to carry them out. This is a tedious task, but it is extremely important because it clearly defines how the tests are carried out. A new user with relatively little training should be able to pick up a completed script and carry out a test scenario with no assistance from the test author. As mentioned earlier, these scripted steps are also invaluable to the training team for their preparation.
- The final step in developing the test scenarios is preparing the data that will be used in each of the scenarios. Real SKUs, ASNs, and Orders should be used in the test environment. This means that the data should originate in the host system and interface as it will in production to the WMS.

Test Plan Execution

It is important to maintain organization and discipline during the execution of the test scenarios. Establish rules and adhere to them. The following points will help maintain the needed structure:

- Develop an issue list to track questions, configuration problems, and bugs. The issue list at a minimum should record the nature of each problem, its severity, the date it was discovered, and its current status. Establish rules for who can edit the issue list, what is eligible to be on the list, and how it will be managed.



- Establish regular status meetings to review issue status and overall testing progress. These meetings are especially important if multiple vendors are involved in the project.
- Keep track of what test cases and test scenarios have passed and failed. It is not uncommon to use color coded filing boxes to distinguish which test scenarios are available for first time testing, are in failed status, or are complete/passed.
- Keep track of configuration and code changes as they occur during testing. It is important that these changes are applied properly to the training and eventual production environments.
- Develop ground rules for how and when changes are implemented during testing. This is especially important if multiple users are involved in carrying out test plans.
- Determine what types of issues are severe enough to prevent go-live and which ones can have process workarounds. Ensure that both kinds of issues are communicated thoroughly to the project team.

In summary, WMS testing cannot be approached as an afterthought or something to be squeezed in during the last two weeks prior to go live. Careful scheduling that takes place at the beginning of the project is necessary to properly plan for required time and resources. Subsequently, test development can be carried out in an integrated fashion with the design and training teams. In turn, this collaboration naturally leads to a well structured and disciplined test execution.

Dustin Vaughn is a Senior Consultant with enVista. He can be reached at dvaughn@envistacorp.com

The following is one example of an issue list template:

| Issue No. | Functional Area | Process | Description | Notice Date | Age (Days) | User | Owner | Severity | Status | Date Resolved | Resolution Note |
|-----------|-----------------|---------|-------------|-------------|------------|------|-------|----------|--------|---------------|-----------------|
| | | | | | | | | | | | |
| | | | | | | | | | | | |