Testing Overview for a Multi-Application Implementation

Kelly Bishop, VCS Principal Consultant, PMO Practice

In response to the Meaningful Use (MU) initiative in the American Recovery and Reinvestment Act (ARRA), many healthcare organizations are currently implementing or planning to implement an Electronic Health Record (EHR). An EHR is defined as a computerized record of the medical care a patient receives in a hospital and/or physician’s office. Implementing an EHR is a complex process that involves many different applications and may include multiple vendors.

It’s common knowledge that a successful EHR implementation requires extensive testing regardless of whether a full clinical system suite is being implemented, a subset of clinical applications is being added, or one or more ancillary applications are being implemented to enhance the patient record. Unknown to the common citizen is the amount of effort required to assure that testing is adequate. This paper attempts to outline the time, effort, resources, and types of testing necessary to support a successful EHR implementation.

Why is all this testing necessary?

It’s often been asked why it’s necessary to test, then re-test, and then test some more. These applications are installed and running successfully at many different hospitals, doctor’s offices, and integrated delivery networks, so the software must work, but why is it necessary to test so extensively?

There are four main reasons that testing has to be meticulous since patient safety is at stake:

1. You are testing to determine if you encounter any software deficiencies. While this same software may be running successfully in other medical facilities, it’s doubtful that there will be another organization with the same mix of various software applications, on the same software version, configured the same way, with the exact interfaces, exact hardware and mobile devices, and supporting the exact same workflow as your organization. One variation may uncover an issue that has gone undiscovered and, if so, you want to be aware of it before Go-Live.

2. Thorough testing will also help validate the completeness of your build.

3. Thorough testing will help you identify workflows that may have been missed, and verify accuracy of others. Changes, if any, can be addressed prior to training and Go-Live.
4. Lastly, and possibly most important, testing will help solidify your knowledge of the application(s). Generally, there are many people on the project team that have no previous experience with the applications being implemented and it’s impossible to gain complete knowledge during vendor training. Becoming comfortable with the new software and learning much of the general troubleshooting occurs during testing. Testing helps assure that what occurs during each scenario is expected; if not, and the system is working as designed, then you will have time to modify the build or the process before training the end-users.

The length and depth of testing will vary depending on whether you’re implementing a complete EHR, several integrated clinical applications, clinical ancillaries, financial management applications, or standalone modules. The following next steps will go a long way in assuring a successful Go-Live.

Start at the Beginning
To be ready to perform testing when scheduled in the implementation project plan, it’s necessary to begin planning for the test period(s) at the beginning of the project. It’s recommended that a testing project manager (PM), or at least a testing coordinator be identified at the project outset. The testing PM will then create a detailed testing project plan which aligns with the dates identified in the master implementation plan.

Decisions that need to be made and activities completed by the testing PM include:

♦ Estimate the resource requirements and identify the resources necessary to complete testing in the timeframes designated in the master plan.
♦ Assure that all interfaces with third party vendors will be ready for testing during the appropriate timeframes. Resources from these vendors will also need to be available to assure connectivity and assist in troubleshooting any issues that may arise. It may be necessary to purchase new or upgraded interfaces or services from these vendors. For example, a new pharmacy or lab system may require an updated interface version from a third party cabinet, robotics, or instrument vendor (interface and/or resource fees may be incurred for these services).
♦ Assure that test environments are available not only for all applications being implemented, but also for all applications that those systems will be sending or receiving data via interfaces. Third party applications are often forgotten until the last minute and it takes time to have a test environment created or a test machine shipped. For example, pharmacy medication cabinets often have no test environment and testing will require additional equipment to be shipped to the facility. If a new pharmacy application is being implemented or upgraded, it’s necessary to test these cabinets and interfaces between the two. If you’re implementing a new radiology information system but will continue to use a separate voice recognition system, how will you test it with the new Radiology Information System (RIS)? How will you validate the Picture Archiving and Communications System (PACS)?
♦ Determine where testing issues will be logged. In an electronic database or on paper? If issues will be logged electronically, who will be able to enter and update the issues? It may be necessary to conduct
training and add User IDs to the issues database tracking system. Decisions regarding classifications of issues (i.e., critical, medium, low), what the classification means, and expectations regarding updates and resolutions need to be made and communicated so that issues are managed the same way for each application.

♦ Facilitate creation of the testing scenarios for application, interface, and integrated testing. After integrated testing scenarios are written, they will need to be approved as appropriate and complete by each department/nursing area.

♦ Determine the space and equipment requirements necessary to perform integrated testing. How many resources are required to complete the testing scripts each morning and afternoon? Is there a telephone conference line in the room? How many workstations are needed? Are there enough network connections? What about wireless connectivity? How and what types of printers are needed? Do the printers need to be located in the testing room? What type of reports and labels will need to be available and how much of each? It's recommended that at least one of each type of workstation and printer to be used in production be utilized during integrated testing (and used to perform the functions and print the type of labels or reports that it will be used for in production). Likewise, any mobile or peripheral devices to be used should be identified and available as well.

♦ Assure that documentation of the integrated testing results occurs during each testing round. Results should be validated and approved (as pass/fail) on each round of testing by the same group that approved the initial scenarios. The testing PM should also assure that all issues are entered in the issues database and resolved as necessary.

Testing Phases
There are several different types of testing that occur throughout the implementation period. Each subsequent type of testing requires that the previous testing be complete. The more thorough a previous testing phase, the more efficient the next one will be. With that being said, there will be some overlap of the testing phases as information is gleaned from each type of testing.

1. Unit Testing
Unit testing is also known as build validation or build testing. It is typically done by super users building the various files. The purpose is to verify the way the data is being built and how it will look and function. For example, you don’t want to build a 5,000 item formulary file only to realize that after completion, you should have created your mnemonics or descriptions differently (because you didn’t check to validate the way the items would appear during a formulary item search, during med order entry or review, or in the electronic medication administration record (eMAR) or computerized physician order entry (CPOE) applications). It’s recommended that after completing approximately 10%, 25%, and 50% of the build that setup be validated by skeletal testing in the application.
2. Application Testing

Application testing is the detailed testing of a particular application via your department’s workflows. The entire application and all internal workflows should be thoroughly tested to validate the system.

Ask the vendor to provide an initial checklist and/or test plan for your application. You will want to revise the plan to assure that your particular build and workflow scenarios are incorporated fully. All patient types and order types should be tested. Coordination is key so you’ll know what has been tested, when, and by whom. Don’t forget to track the results through an audit trail.

♦ For pharmacy application testing, make sure to include full patient scenarios, such as manually admitting a patient to ICU, entering orders, and transferring the patient to a med floor. Are the orders handled appropriately? What happens if the patient is readmitted? What happens to a patient’s orders if he is admitted through the ED as an observation patient and then becomes an inpatient?

♦ For nursing clinical documentation, make sure to validate the flow on each type of unit with each type of patient.

♦ For radiology application testing, test all modalities, link exams, and the attending physician versus resident workflows.

♦ For lab application testing, make sure to test all areas and instrumentation in each – microbiology, chemistry, hematology, blood bank, and pathology. There are often CAP and/or FDA testing requirements that the lab must comply with when a new information system is installed; assure that all test plans and results documentation are stored appropriately.

♦ Include negative testing. If an error occurs, it should be noted as such. Do reports capture errors correctly?

♦ Each different user security type should be tested in each application. Different roles allow access to different functions and may also require verification.

♦ When you test reports or label batches, don’t just check to make sure something prints. Does the correct report print? Is it formatted properly? Does it contain all the information it should and only the information it should? Is there an error report?

♦ Test charges and credits. If you have billing configured to net the charges and credits in any given billing run, make certain to have scenarios that will net credits for validation. Validate not only your billing report, but the billing file that is generated for your revenue system to make sure they match.

♦ Check each different type of printer to be used in your department and print the information that will generally be printed on that printer. For example, don’t have your labels pointed to a report printer and never test the printing of labels on label stock in your label printer.

♦ Test each type of mobile device.

All issues should be tracked and resolutions retested for validation.

Various areas may choose to conduct parallel testing for a period of time (usually 3 to 7 days). This involves selecting several different types of actual patients on different floors, copying their orders (removing identifiable information) and entering their orders, results, and documentation on a test patient. This helps capture additional workflows that may have been missed in the checklist/scenario creation.
3. Interface testing

Interface testing includes establishing connectivity and communications between interfaced systems. After this is complete, test transactions are sent to receiving systems. This is generally done system-by-system in preparation for integrated testing.

Interface resources for third-party systems need to be available to facilitate the initial communications testing. It’s best to plan and communicate the room/bed numbers where the interface transactions will be occurring. In the event that your department is still conducting application testing, this will prevent any of your test patients/scenarios from being disturbed.

At the very minimum, at least one of each different type of transaction for each interface should be tested with each system. Errors should be addressed and those transactions resent upon correction.

Take this time to learn how to troubleshoot your interfaces. Do you know how to tell if the interfaces are running? Do you know how to stop and start your interfaces? Do you know where to see the error log?

It should be noted that some applications are so integral to the function of another that much more testing should occur among them than the official integration testing period allows. These applications may be integrated or interfaced. A good example is closed loop medication management – pharmacy, eMar, barcode medication administration, and CPOE. Ongoing testing between such applications occurs after the initial application testing is complete and after the interface connectivity is established (if interfaced). Scenarios should be developed and results documented in preparation for the full integrated testing cycles.

4. Integrated Testing

Integrated testing is the final testing phase before Go-Live. This testing involves all applications being implemented, in addition to those systems that will send or receive data from these applications. It validates cross-functional workflows, such as: following different patient types, from registration through the entire patient visit or hospital stay, through the discharge and billing. Successful completion of integrated testing means that your application(s) are ready for production. If serious defects are found and aren’t resolved by the end of integrated testing, it may be necessary to either delay your Go-Live or create and train personnel on a work-around solution.

The testing PM will identify a team representing each department impacted by the impending implementation, and will work with them to jointly identify test scenarios that will adequately validate cross-departmental workflows. Each representative must assure that all key functions in their respective areas are addressed in the scenarios. Upon completion, the project team and sponsors must approve the scenarios that will be used for integrated testing.

The number of test scenarios will vary based on the size of the organization and the number of applications being implemented. There should be a minimum of 12 scenarios, but larger networks may have several hundred. Three rounds of integrated testing are usually planned; one week of testing, followed by a week to resolve issues. This is repeated through two additional rounds in hopes that the last cycle will be error-free.
Scenarios should include:

♦ Detailed cross-departmental workflows of all patient types
♦ Workflows that incorporate errors/mistakes to address negative testing
♦ Different test user IDs with varying security roles (per scenarios)
♦ Interface testing (via scenarios)
♦ All end-to-end testing even if the hand-off is verbal or via label or report
♦ Orders/results testing including testing of orders sent via printer routing
♦ Validate patient and/or physician portals
♦ Test mobile devices
♦ Test bar code scanners. Make sure bedside administration is tested via scanner as well as manually tested
♦ Charge/credit, insurance, finance testing
♦ Report generation and validation

In preparation for integrated testing, the test PM will assemble the testing team and discuss expectations, instructions, timeframes, roles, and documentation requirements. Generally, a room is secured for integrated testing so the testers will be together to assist each other with any questions, problems and/or reporting of issues. Vendor resources for the primary applications should be on-site and third-party vendors should be available by telephone.

Each scenario may take a day or more to complete as many will require daily processes to run before completing the next steps. Therefore, it’s critical that problems be addressed as quickly as possible so that the scenario can begin again as appropriate.

The testing PM will be available to oversee the process and assure that issues are documented and reported as required. Conference calls are generally held at least once daily to review status. The project team and/or sponsors must review and approve the final test results.

Upon completion of the final integrated testing round, it will be determined whether the system is ready for production.
5. **Miscellaneous Testing**

Miscellaneous testing incorporates the additional items listed below and may be done during application testing, integration testing, or just before going live in production:

- All workstation and PC access to appropriate applications
- All barcode scanners and readers
- All printers, print routing, and alignment
- All devices or equipment interfaced to the systems such as monitoring equipment, IV pumps, lab instrumentation
- Any data conversions
- Backup and recovery processes
- Wireless connectivity
- Volume testing

**System Freeze**

Once testing is complete and the system is approved for production, there should be no patches, configuration changes, or upgrades applied. Modifying any system code invalidates previous testing.

Assure that the production system contains all code and/or configuration changes made during the final testing phases and then lock down until after Go-Live.

While testing takes time, successful completion of the full testing process should help assure a successful Go-Live.

For more information about integrated testing or other solutions provided by VCS, please contact us at 610.444.1233, vcs@getvitalized.com, or visit getvitalized.com.