White Paper

Lessons Learned from User Acceptance Testing

M. Glenn Newkirk
Helen A. Sims

June 15, 2002

InfoSENTRY Services, Inc.
www.infosentry.com
919.838.8570

InfoSENTRY's Lessons Learned from User Acceptance Tests

M. Glenn Newkirk
Helen A. Sims

InfoSENTRY Services

"Get this project on the fast track." "Cut to the chase." "Skip detailed user acceptance testing. We (the vendors) have it covered." "Testing is an unnecessary roadblock to getting on with the real work." If one (or more) of these statements reflects your organization's philosophy toward independently administered user acceptance testing, get ready for the system implementation crash that is coming. After the crash you will realize it's better to be on the RIGHT track than the FAST track. These lessons from InfoSENTRY on user acceptance testing will help ensure you are on the RIGHT Track for project success.

Our Experience

InfoSENTRY has designed and administered user acceptance tests relating to implementation of various systems. Our acceptance test clients include State agencies, not-for-profit organizations, State Boards of Elections, county government offices, and state legislatures. One of our staff members led a team effort that resulted in acceptance and certification of technical and user documentation for one of the largest telecommunications switches on the market.

Some of our software tests involved a limited number of test items. Those tests concluded in several days. Other user acceptance tests for financial systems, membership management systems, document management systems, document imaging systems, vote tabulation systems, and voter registration systems involved hundreds of test items and lasted several weeks.

Our Process

InfoSENTRY has developed a project management database system called SLIM™. It stands for System Lifecycle Information Management.

SLIM™ Process Stages
Needs assessment and requirements analysis
Software requirements specifications
System design specifications
Request for proposals
Proposal evaluations
Contract with vendors
Final Acceptance testing
Implementation and cutover

SLIM™ provides step-by-step IT project management guidelines and documentation of a project's major processes. We use it to assist in managing software tests for our clients. We document our clients' system requirements and develop specific test scripts to determine if the installed solution meets those requirements.

We recommend that our clients' managers and staff actually carry out the tests, documenting the results. We collect and organize the documentation

Rev: 1.0: 15-June-2002

for the each test item, assisting our clients to determine **independently from the vendor tests** if the installed system passes or fails the test items. (Our software has various test status categories to reflect what typically happens in tests: “pass,” “pass, conditional,” “fail,” “pending,” “scheduled,” “suspended,” and “deleted.”)

We have the ability to post all results in a password controlled web site to make the results available to our clients, software developers, and vendors as soon as they are completed. SLIM™ provides a variety of test status reports on demand.

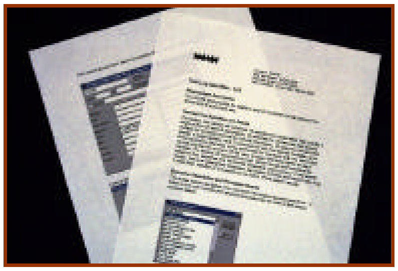
In forming a test team with our clients, we provide **InfoSENTRY** consultants who have detailed training and experience in software quality assurance reviews, software test design, and test administration. Typically, we provide a Certified Quality Engineer as one of our test team members.

We often videotape all or portions of the tests. These videotapes provide an excellent source of documentation in addition to the volumes of reports and files that normally result from our tests. We compile all of the test results, providing a comprehensive collection of documentation on CD-ROM(s).

After each one of our acceptance tests, we carry out an “after action review” to capture lessons learned. These “AARs” are part of our own continuous improvement process at **InfoSENTRY**. We hope the following compilation of test tips helps as you prepare to embark on your system implementation.

Acceptance Testing Lessons Learned

1. **Be prepared for acceptance tests to take longer than you think.** Plan ample time for the tests. Here are some thoughts on the time required for testing. Assume that you have 120 test items. (A test item could be to produce a specific report, scan 10 batches of 30 documents each into a document imaging system, or generate an export file with 600,000 records.) Further assume that it will take you an average of three hours to set up each test, carry out the test, and document the results. That means the entire test cycle will require 360 hours (120 test items x 3 hours each). That is just over two labor months’ effort. Of course, that all assumes that the system or software solution passes each test on the first past—which almost never happens. It also assumes that each test item takes only three hours.



   You might have some test items that take up to one or two days each. Also, remember that some tests will require multiple system re-boots. So, do not underestimate the time and resources that will be required to complete a full test plan.

2. **Start planning your acceptance tests as you prepare your business case, needs assessment, and requirements analysis.** There is no better time to start preparing test scripts than when you are looking most intensely at your business processes and requirements. Early preparation of test scripts will help you include script details that you might otherwise forget. Sometimes, preparation of a detailed test script

will give you ideas for re-engineering the way you do things—and maybe even help you decide that you do not need to carry out some functions at all.

3. **Write into your RFP and (most importantly!) your contract with your vendor the expectation and clear right for you to prepare and conduct the acceptance test.** You might think that the worst mistake would be to forget to put acceptance testing in a contract with a system vendor. Actually, we have seen much worse. In the case of some enterprise resource planning systems (ERP), we have seen language that gave the right of the client to test the software "if they choose to do so"! Adding insult to injury, the contract stated that if the client found any errors during the testing, the client could ask to have the vendor correct the error AT THE CLIENT'S EXPENSE!

   We strongly suggest that you state in your RFP that you intend to design and conduct an acceptance test—and that a specified portion of payment will be made based upon successful completion of the testing to your satisfaction.

4. **Do not confuse the system developer's unit testing, integration testing, or system testing with your user acceptance documentation.** Vendors often indicate that they have already tested the systems, upgrades, or "bug fixes" they are delivering. This kind of testing refers typically to tests their programmers or (more rarely) their separate test staffs have made to determine if the software modules or "units" work. Occasionally, vendors will refer to more comprehensive system or integration that they have undertaken. At a minimum, you should ask to see the set of test documentation the vendors produced internally. However, this type of testing is not user acceptance testing. Usually it means that the software or system meets the vendors' requirements and has passed the vendors' internal processes. If often means that you, the customer, are being used as what are known as "alpha" or "beta" testers—working the bugs out. It does **not** mean that the software or system meets your requirements and standards.

5. **If possible, provide the vendor with the acceptance test in advance.** Sometimes, this step simply will not be possible. The vendor might not deliver full documentation to you in time for final completion of the test prior to test time. At a minimum, it is a good idea to provide the vendor with samples of acceptance test items to let them know that you plan to engage in a serious system test. User acceptance testing is a joint endeavor between testers, clients, and vendors in an effort to implement a product that meets the client's needs. To the greatest degree possible, the test should not be a pop quiz. It is an open-book, take-home, pass-fail effort to achieve a desired outcome.

6. **Set an agreed upon time with the vendor for the test beginning and end.** If you do not set firm dates, there is a danger the test will never end as the developer or vendor tries to "tweek" the software to make it meet all the requirements. In this circumstance, you are actually becoming a part of the vendor's development team. The entire effort can drag on indefinitely. The vendor does not get paid and you do not get your system installed. Eventually, someone will run out of time, money,

or patience. Either the vendor will walk away from the project or you will have to accept a largely untested system solution. By this time, you are often frantic to get the system installed and will settle for a system that is untested and undocumented. A firm date for test completion sets realistic expectations for when the test must be completed and you will be ready to put the system into production…or get ready to use your existing process for some time to come.

7. **Communicate to managers, testers, and the vendor the test progress (a "status scorecard") at least weekly.** During intense periods with large numbers of tests, daily might be even more useful. The test administrator or test manager should be responsible for preparing and distributing the test status reports. The status report should contain a clear statement of where each test item stands. Typical status steps for each test item will be pass, pass-conditional, fail, in process, pending, scheduled, and deleted. A good test management software package will allow you to add other test status indicators to meet your individual needs.

   However, you do not want to add status steps just to give someone a warm and fuzzy feeling of accomplishment.

8. **A website is a useful way to post test results.** It helps the test manager organize and communicate test results. It helps foster a sense of "best test practices" among the testers. It lets the vendor know where the test plan stands and that everyone is taking the test process seriously. Frequently, the testers and the vendors' representatives will be in different locations. The website allows this virtual test team to remain in the testing loop.

9. **Include staff and representatives from all user groups in the acceptance testing.** They often know best when a system works and when it does not. Their participation also is a great source for enhancement suggestions. Through their participation, they will learn the system and how it fits into their business operations

10. **Acceptance testing can be a highly valuable team building experience among staff.** Why pay to send your staff on a winter survival training expedition to build team *esprit de corps*?

    Why have them go through encounter sessions to get in touch with their inner selves? Have them do something more challenging together—test the system they will be using. It will help everyone learn about how other staff members work and relate to each other.

11. **A thorough test process can be an excellent, detailed training tool for system users.** It can provide a great time for already skilled people to become more proficient in their jobs. It also provides an excellent opportunity to cross train your staff. By participating in testing, your staff can become "super users" or "trainer assistants" for other staff…if the system passes the test and you adopt it. If the system fails the test, staff that participated in testing will become great resources as you start to define your requirements and

begin the procurement process anew.

12. **Consider videotaping major components of your acceptance tests.** As a general rule, programmers and testers do not like to document their work. Also, during testing there are often very important comments and observations that go unnoticed or unwritten. Sometimes there are phone discussions with vendors, the results of which are undocumented. Videotaping major test activities can provide a very easy way to capture test activities in the context of what is happening on the video monitor or printer.

The videotapes also become very useful resources for training new staff and refreshing the memories of staff involved in the initial training.

Videotaped test segments also help the vendor see more clearly the problems you ran into during testing.

13. **Comprehensively documented test results are among the best ways to introduce accountability into the contract situation with the vendor.** Documented test results demonstrate due diligence on the customer's part in demonstrating that they are getting what they contracted and paid for. Good user acceptance testing and comprehensive test documentation are measures of due diligence shown to the corporation's board in the private sector or to governing bodies in the public sector. Good user documentation ***for each completed test item*** will include the name of the test, the original functional or technical requirement upon which the test is based, a planned test script, the name of the lead tester, the names of other testers, the start date and end date, a narrative of the steps and procedures undertaken during the test, copies of all files and documents printed during the test, and a statement of the "pass/fail" status of the test. The Institute for Electrical and Electronics Engineers (IEEE) has an excellent set of standards and guidelines for how to structure your test documentation.

While all of this documentation formerly might occupy a very thick set of notebooks for even a limited number of tests, it is a great idea to convert all of the documentation into a standard format (such as PDF). Then copy it to a CD or DVD. Massive amounts of documentation will fit on a single platter, making it convenient for distribution and frequent use.

14. **Thorough test documentation will serve as a benchmark for future system support and enhancement.** Vendors get frustrated when clients say in about a year after the test that "…it has never done this before." Users get frustrated when vendors say, "No, the system has never been able to produce that report" after an upgrade that wiped out existing capabilities. As you maintain and enhance your software, you can refer back to your acceptance tests

to check for what are called "regression" errors that might creep in while you are trying to improve your system. Clear test documentation can assist you to go back to examine system characteristics at earlier stages. Good test documentation has use well beyond the date of acceptance.

Using **InfoSENTRY**'s "lessons learned" for user acceptance testing can help you be on the RIGHT track for project success. Our clients have been pleased with how our user acceptance test process has helped them accomplish their system implementation goals. If you want help with your user acceptance testing, please contact Helen Sims at helen_sims@infosentry.com or call Helen at 919.838.8570.

For an example of a single, completed test item, click or visit http://www.infosentry.com/uatp_sample_test_item.pdf .