



Lucitech, Inc.
P O Box 1169
Ramona CA 92065
(760) 789-5753
www.lucitechinc.com
tracyengel@lucitechinc.com

10 STEPS TO TESTING YOUR AIM 6.X VERSION **by Tracy Engel, President of Lucitech, Inc.**

One client has asked that Lucitech manage the project to upgrade their current AIM 5.20.x version to AIM 6.3.x and we are *almost* ready to push into production. We have been in testing and elected to continue testing the 6.3.x version that we have nearly approved rather than go straight to 6.4 now that 6.4 is released. The testing process is definitely a process. Our Vertafore Super-Support Tech has been on-the-spot with every fix and it has not been terribly painful, just a process. It would be a somewhat unique process at each agency, presumably, as different operations focus on or use different features of AIM and have different custom enhancements.

Now that 6.4 is released, several Lucitech clients have mentioned that Vertafore has installed the 6.4 version on the client's test server and they are about to begin the process of testing. Thought it would be helpful to post some notes regarding the process we adopted and the lessons learned.

It has been most helpful for Lucitech to keep our own list of any issues we discovered and the resolution, outside of MyVertafore. Not to keep track of the issue resolution, necessarily, but because a number of fixes need to be re-applied in-house at the time of the production push either at the database level or the user machine level and we wanted our own notes. We are getting ready to push to production and now witness our gratitude that we kept those notes so organized and complete.

Another item to note is more on the human level. There is an overall 'error-mentality' to overcome with user testing. Users are somewhat accustomed to access violations and other errors in older versions of AIM as part of the course of doing business. The error response is almost automatic in regular users of older versions, to the point the error is closed before they've even stopped to think about it. However, in testing, it is better to take the attitude that 'every error is reportable.' Users, left alone with the 'error-mentality' might sit and suffer with something that is very fix-able. Worse, we've witnessed an immediate switch to 'work-around' mode when it would be preferable in testing to report everything.

Lucitech takes a two-pronged approach to defend against this user-mentality. First, we do as much pre-testing of AIM underwriting processes as practical prior to releasing the version to user testing. We experienced, reported and resolved a great deal prior to interrupting users for testing so their experience would be as smooth as possible. The second prong is to sit with each user or a user group and share their desktop for the first hour of testing. This hour opens with a loaded question: "Please tell me all of your AIM woes. What bothers you most about AIM? Let's test it." Then we log into the test server and tick off their issues, reported or unreported, one-by-one. That way we know that for the first hour we will be targeting specific issues. When they see some of their vexing issues are fixed, they are willing to "buy-in" to the testing process. All through the hour, we stress how important it is for them to report anything and how they can reach us to report things directly. At least in this office, so far, knock-on-wood, the users haven't gotten any access violations with this version. In fact, with my pre-testing process, the users haven't had many errors at all and so far all are pleased with the version we are testing.

Again, there is no one-size-fits-all testing or upgrade approach. However, in an effort to start a dialogue, here is our overall test plan, cleaned up and revised based on lessons learned so far:

Nothing in this document is to be construed as advice on your unique implementation. This list is not purported to be all-encompassing or to fully test all the AIM features but only hits on features used frequently at one office in one setting. Experience in one office may vary from experience in another office.

1. Open MyVertafore and test every open issue for this office on new implementation. Close any that need closing and re-open or refresh any issues that are not resolved in the new implementation.
2. Open DMU as SA and delete and re-activate myself as a user to begin testing
 - a. There were a number of errors here
 - b. Some were machine specific
 - c. All were easily resolved.
3. Run every report at least once
 - a. There were occasional errors due to script customizations
 - b. Each error was easily resolved
 - c. If there was an error, it occurred upon opening the reports. I did a couple reconciliations of key reports such as the primary system production reports and these checked out as 'better' on the new implementation in this particular office.
4. Docuparser Integration
 - a. Encountered an error based on the dll on the users machine
 - b. this is a machine specific error and must be resolved on each machine when you roll out into production.
 - c. Default forms are now treated as required forms and user is unable to de-select this. Vertafore's initial response was that this is by design. Once I demonstrated that this was clearly not the design, have not received back yet a response on a fix.
5. Bordereaux
 - a. These are all custom scripts
 - b. Didn't appear to be any changes to the 'contract participation' processes
 - c. One script customization was over-written by the upgrade. We restored the script and the custom bordereaux began working properly again.
6. Accounting
 - a. Testing in accounting has been good.
 - b. Accounting department is generally pleased
 - c. Direct bill upload process
 - i. This had problems with new 'truncation errors' that never took place in the past.
 - ii. There was a work-around to modify the excel file prior to uploading
 - iii. Got the distinct impression that any fix on the upload process would be considered custom work and left it at an easy work around prior to uploading the file (change zip code to a 5 digit zip is the fix)
 - iv. There were more than just that. We have the file working for the month I was using for testing.
 - v. Users are assigned to test additional months to be sure there is nothing unique to the one month I had that makes it pass but another one might fail.
 - vi. Later in testing it was learned that the whole upload process was hitting the wrong report. As of this writing we are still waiting for a fix and will be unable to move forward without.
7. Rackley Integration
 - a. Early testing indicated that it 'works' in the broad sense that the premium is coming over and looks nice.
 - b. User testing has reported no issues to date

Nothing in this document is to be construed as advice on your unique implementation. This list is not purported to be all-encompassing or to fully test all the AIM features but only hits on features used frequently at one office in one setting. Experience in one office may vary from experience in another office.

8. Quoting/Binding/Invoicing
 - a. Installment Bill
 - i. There were a number of issues with how installments were processed in this office's previous build.
 - ii. As part of item #1 above, I gave installments their own item in my plan as there were many things to test about installments.
 - iii. Virtually all issues with regard to installments that this office had previously experienced are fixed in this version.
 - b. Agency Bill
 - c. Direct Bill
 - d. Courtesy Filings
 - e. KY Taxes
 - i. The Municipal Tax module was not helpful to our office with the KY tax setup.
 - ii. It applies municipal tax once and only once to each policy on the "Quote" level, not the "location" level.
 - iii. We disabled it and went back to the old way of doing KY taxes.
 - f. Documents
 - i. Originally went through and started doing like I did with reports and running every AIM document at least once.
 - ii. This quickly got tedious and wasn't running into problems so I stopped.
 - iii. AIM Documents seem to be OK
 - iv. Any 'word-merge' or anything fancy isn't tested here because this office doesn't have anything fancy here. If there were anything fancy here, I'd be testing it vigorously.
 - g. Tricky Transactions
 - i. Users are instructed to get creative on their own testing time (which they are assigned a given amount of independent testing time) and try any tricky transactions and try to 'Stump The Upgrade'
 - ii. When you stump it, report it. There was at least one case for us where there was new development that required DMU setup before it would work.
 - iii. To date all user reported issues have not been show-stoppers at all, and did come out during the 'tricky-transaction' push.
 - h. Sundry quote processes
 - i. Under this bullet point you can put submitting quotes to brokerage, declining business, closing files, and so forth.
 - ii. These bullet points are mostly directed by each individual user in order to target the processes this office uses most – what do they do each day?
9. Endorsements/Cancellations/Other Underwriting Transactions
 - a. To date, there are no major issues with endorsements or other supporting processes

Nothing in this document is to be construed as advice on your unique implementation. This list is not purported to be all-encompassing or to fully test all the AIM features but only hits on features used frequently at one office in one setting. Experience in one office may vary from experience in another office.

10. Custom Processes

- a. This agency has a number of custom processes from they way their due dates on invoices are processed in AIM to custom upload processes and reports.
- b. It turned out that one custom CIS process failed, but only on one day of the month was the failure apparent. No lie. It was lucky I was testing that day.
- c. I have a list of all custom processes and a set of targeted test cases for each process.
- d. When I don't have test cases for a process, I turn it back to user testing and have the user demonstrate the process to me and talk to me about any "pain-points" with the process.
- e. Any unresolved "pain-points" are documented for CIS support. Many of these unique types of 'old invisible user-issues' are being referred to development for future versions, but the cases are being documented very completely in accordance to Lucitech documentation guidelines and the cases are open now in MyVertafore and being tracked appropriately, rather than sitting under a metaphorical band-aid at the users desk.

WHO IS LUCITECH?

Lucitech is a private consulting firm that develops and delivers customized insurance software solutions through project design, software programming and project management, improving workflow efficiencies.

Lucitech specializes in the MGA market and has extensive client expertise in commonly deployed Vertifore AIM software. Lead by Tracy Engel, who has over 20 years of experience in insurance and IT, Lucitech understands insurance workflows and develops solutions that directly impact productivity and profitability for insurance companies and MGA's. Lucitech assumes the role of both an insurance analyst and system architect, ensuring detailed documentation on processes and workflows, delivering on-target budgets, schedules and timelines, for project implementation success.

Contact Lucitech if you would like assistance with your AIM 6.4 Upgrade!

www.lucitechinc.com.

Nothing in this document is to be construed as advice on your unique implementation. This list is not purported to be all-encompassing or to fully test all the AIM features but only hits on features used frequently at one office in one setting. Experience in one office may vary from experience in another office.