By Scott Cooley

Can You Really **Try Before You Buy?**

It's a problem almost everyone in our industry has either experienced first-hand or had a close business associate who has. A mortgage company buys a software solution and it quickly deteriorates into a boondoggle. The staff frustrations mount, productivity grinds to a standstill, management is up in arms and everyone's pointing figures.

The company is then faced with a difficult choice: It can either toss out the software, resulting in significant loss, or keep it and force a reluctant staff to use something they would rather not. Could this frequent problem be prevented?

With some proper planning, there are ways of preventing catastrophic technology implementations. However, not all of the pre-planning is common sense. Let's look at some common traps to avoid. Hopefully this will help ensure that your next software implementation is smooth and efficiency-enhancing.

The most important decision is making the right technology choice from the start. In my eMortgage column in the November issue of *Mortgage Banking*, I outlined how to choose the right software product/vendor. What I will cover in this column (my first in Mortgage Banking's new quarterly technology section) is how to take advantage of trial periods and some of the issues that are critical to a proper technology implementation.

First, ask yourself would you really want to go through a trial period using a new system. Most likely, the answer is entirely dependent upon your subjective viewpoint. These are some things you should consider:

- If the software vendor is fairly new and without many proven customers, it's far more important to try the company's solution before jumping in and buying.
- If there is a lot at risk by making the wrong software selection, you should be more likely to try it before buying.
- If the technology niche is newer, such as workflow versus basic loan processing, then a trial session is more pertinent.
- How strongly do you feel about the solution you have chosen? If your confidence is very low, then a trial can help.

The problem with trial periods is they take up a lot of resources. So, you want to make sure one is really needed. Assuming that proper procedures were followed for selecting the right software to begin with, most companies can get by without a trial period.

Software vendors hate trial periods, but it's not for the reasons you might think. Sure, software companies want to collect their revenues quickly; but the primary reason for not liking trial periods is that customers aren't "all-in" in terms of their commitment until they actually buy.

If you feel your software implementation was frustrating, then don't be too quick to blame the vendor.

When I ran Contour Software, I experimented with different free trials for years, but I could never get them to work. I always found that mortgage companies would put into the implementation what they had risked to get there. A company that got a CD for free would often stick it in their desk for good, or play with it for a matter of minutes. On the other hand, a company that just spent thousands to buy software would darn well make sure the staff was using it.

Thus, trial periods often fail as managers find too many obstacles to get past. When the staff knows it's on a trial period, their goal typically becomes a game of finding all the holes in the new system.

In November, Irvine, California–based Portellus Inc. opened its "Customer Readiness Lab"—an offering to prospective clients to ease their nerves about making a major software purchase. The Customer Readiness Lab gives Portellus the ability to control the testing environment and ensure users are well taken care of. Users can play with the software without having to install it on their own systems. It's a reasonable way to play with the software, even though a lab located at a software company doesn't allow for an extensive trial in a real-world office.

There are other problems with trial periods as well. We like to think that loan production is about processing a commoditized product that is mostly the same. However, it's anything but—every loan is different in some fashion. It's very difficult to see how a system will operate under all the different types of loans, borrowers, products, properties and so forth.

Without a full pipeline loaded up, we can't see how well the management-reporting systems operate. We also can't go through crucial processes such as end-of-year accounting, regulatory reporting and the like.

The trial period often doesn't fully engage the group psychology issues that kick up in a real situation where new technology is being adopted. There's a reason why good software implementation plans include the nebulous "staff buy-in." With a trial period, it's not real—and everyone knows it.

Another common problem is that often just the top few employees are chosen to participate in the trial periods. However, I'd also want to bring in the most average and even difficult employees to the table. They can help identify just how complicated the software really is.

Finally, how often do you see stopwatches pulled out at a time trial? Yet isn't the primary reason you're switching systems is to become more productive? Thus, we must time-trial all the core areas of any software application and compare them with the existing solution.

I've often found bleeding-edge applications to run slower than tried-and-true systems. Today, the more advanced Web-based solutions are often slower than the older client/server products, based on my experience.

One final note, if you feel your software implementation was frustrating, then don't be too quick to blame the vendor. I've found that most of the mistakes actually reside with the mortgage company. Perhaps the company made a poor choice to begin with, or, more often, it lacked experience in making sure the software installation was properly managed.

Basic ground rules are often violated during implementation, such as not using the software vendor's implementation team (or at least its assistance), not obtaining staff buy-in, not putting forth the level of required resources needed for that product, buying a software solution that was too high-end or too low-end, or not doing enough training before implementation.

No software vendor wants a bad reference, and some go to great lengths to get the customer to do a proper implementation. I've found that if the vendor has a long history and its products are well used by firms similar to yours, mostly likely your problems stem from within.

Still, software vendors can also make some common mistakes, which include the following:

- Overselling the product and its features (more of a problem with newer vendors than older ones).
- Claiming their software can work great for your company even though you are much larger or smaller than their typical client.
- Overemphasizing how easy it is to install and use the software.
- Not adding in enough on-site training and support in order to keep the final price lower or to meet a budget.
- Promising their end-to-end solution is everything needed, when often it lacks key components.

Every software implementation will have bumps along the road, and rarely does everything go right. However, with proper expectations and some good planning, any mortgage company can swap out systems and be happy in the end—with or without a trial period.

Scott Cooley is an independent mortgage technology consultant, analyst and author based in Los Gatos, California. He can be reached at scooley@scooley.com.