



Will You Ever Need an Exit Strategy?

Henry Schaffer, *North Carolina State University*

A good decision maker, when adopting a new software solution, considers the initial cost, continuing costs, the return on investment, compatibility with existing software and personnel skills, and so on—evaluating all of the start-up and projected continuing costs. Today, possible software solutions must include cloud computing and, more generally, software as a service (SaaS) or “x” as a service (XaaS¹). Too often, however, decision makers fail to consider a crucial aspect (which I left off of my list... did you notice?), because it isn’t visible at the start of a project. Beware of this blind spot... it can come back to bite you in a painful way.

I’m referring to the exit strategy, which is often neglected because few want to consider the demise of what is, at the moment, a seemingly wonderful solution, being adopted and implemented with great effort and expectation. This is a time of hope, and extensive analysis has shown a rosy view of the future. It would be rude to introduce the topic of planning for the failure of the software or service (or of the company providing it). You don’t need to plan for a failure; instead, you can simply plan for a potential parting of ways. This brings to mind P.T. Barnum’s famous sign, “This Way to the Egress,” which was posted to lure lingering visitors out of his museum (many visitors, not knowing that “egress” means “exit,” followed the sign

under the assumption that it led to another exhibit). Sometimes, we don’t want to exit, but we really do need to consider a safe and affordable way out.

Costly Lack of Foresight

Why do I bring up this unpleasant topic? It all started with a recent discussion with a colleague who is dealing with the failure of a software company on which she depends.

Out of Business

This colleague of mine has been involved in a hobby for the past 40 years, and when personal computers became available, she decided to store her data records on a computer. So, well over 20 years ago, she purchased a program to handle and display the records, and she typed in the data from her paper records. Since then, she has been entering all of her data into the program and keeping up with successive software releases. She’s not a computer-savvy person, but she was aware of the need to keep backups, and she tried to address all the risks she could imagine. She was happy with the program—and with everything associated with it—until the company went out of business.

She had heard about the company closing, but she didn’t worry about it, because she had her data backed up and had the CD of the most recent version of the program. Did she ever wonder whether another program could

read these precious data files? Apparently not—at least not until she realized that her Windows XP computer was growing old, at which point she learned that the program wouldn’t run on Windows 7. That’s when she called me.

“Don’t worry,” I advised her. “Worst case, you can display all of the output in printed form or screen captures, and then key it into a new program.” I had to hold the phone several inches from my ear as she explained that there were, perhaps, 1,800 records, and each was up to 1,000 characters long.

After calming her down, and pointing out that optical character recognition (OCR) could ease the task should she require this last resort, I advised her to go online to the different forums and groups of people who shared her hobby. I told her to visit the discussion groups for the companies supplying software to this community to search for information and ask questions. I hope this worked—I haven’t heard from her since. Yet her issue brought back memories of a rather large problem in which I was involved many years ago.

Proprietary Data Format

A long time ago, a state agency wanted to computerize a fairly important function, so the staff wrote specifications and requested bids. A contract was awarded, and the installed system worked well, becoming an important and widely used system, affecting essentially

everybody in the state. The contract was renewed several times.

However, after a decade or so, the state agency thought it might be a good idea to re-bid for the system to see if it could lower the expenses and perhaps gain some added features. The agency described the bid plans to the software company. In response, the company advised the agency that if the contract went to another company, then the databases, which were in a proprietary format, would no longer be accessible. The data could be extracted from the databases and delivered to the agency in another format, but there would be a rather substantial charge for this, which would exceed many years of any possible savings. Needless to say, the contract was renewed, albeit with some changes.

Let's go back to the initial contract specifications and planning—and the failure to include the need for an exit strategy. There didn't seem to be such a need, because everything was going well. Perhaps the state employees felt too uncomfortable to ask, "How do we get our data out of your databases if the need, in our opinion, ever arises?" The state agency and its "owners" (the taxpayers) paid heavily for this lack of foresight. (On the other hand, if you're a software solutions vendor, this might remind you of how important it is to lock in your customers and exploit their lack of foresight.)

Avoiding "Down the Road" Catastrophes

Now, back to the present. Cloud computing and SaaS offerings are becoming widespread, and the economics of adoption are often compelling. Once again, there are questions that must be answered and understood before signing a contract, including

- Who owns the data we enter?
- How do we get our data out, in a timely and affordable manner and in a format we can use?
- How will we store, manage, and use our data in the cloud?

Recently Nirvanix, a cloud storage company, announced that it might be closing and gave customers two weeks to retrieve their stored data.² Note that some customers had 10 to 20 petabytes of data! (Extra credit question: given a 1 gigabit/second Internet connection, how long would it take to download 20 petabytes of data?)

Most of us are in the user/purchaser community, so how might we avoid this type of "down the road" catastrophe? It's not complicated, if you continue to do everything you're already doing but also make sure to include an exit strategy.

Now that you're convinced of this need and ready to invest some resources, you'll find that there are some relatively easy ways to proceed. They aren't as easy as doing nothing, and they might not be perfect, but they go a long way toward protecting against the risks you face.

Data Transfer

A good example of protecting against cloud storage exit problems is to be aware of vendors other than the one you're using and to learn how to transfer stored data from one vendor to another. Transfer can be technically demanding—for example, using `rsync` (<http://rsync.samba.org>) for incremental file transfer. This is an effective method, but not everyone is set up to do this, and not all storage formats easily lend themselves to this.

It would be better for us if data transfer was somewhat less technically demanding, and the cloud storage community is aware of

this need. This has led to the development of a standard called the REST (REpresentational State Transfer) API, which aims to ease storage transfer between vendors who have adopted the standard. Investigation is the key to picking a method that meets your needs.

Software Escrow

Software unavailability issues depend on multiple factors, and although there's no guarantee you can avoid such issues, you can address the factors to lessen the risks.

For custom software or software from a small provider, you can be left high and dry if the provider goes out of business. Although you might be able to continue using your existing copy, there are many situations in which some updates are required—such as a change in the operating system. A strategy for this situation is "third party source code escrow," in which the source code and documentation are kept by a third party to be turned over to you if the provider ceases operations (see www.softwareescrowguide.com for a fuller description).³ Several companies provide this type of escrow service.

This solution can keep you in operation, but again, it comes with costs. You will have the cost of the escrow service, and then, in the unlikely event you need to avail yourself of this escrowed code, you'll need skilled technical personnel to use it. Even if the software vendor isn't small, a company can stay in business but discontinue the software that's critical to your operations. The vendor might provide an alternative product that doesn't meet your specific needs. Again, provisions for a graceful change should be made, and the best time to do that is before signing the original contract.

Provisions for Decreased Use

Exit strategies for SaaS and platform as a service (PaaS) are even more complicated, because both storage and software are generally involved. In some cases, the two are separable, as in infrastructure as a service (IaaS) and PaaS, and a straightforward combination of the two types of strategies can be used. But SaaS integrates the software, platform (computing resources), and storage, and the typical usage-based license adds a new dimension of risk.

In addition to the vendor ceasing operations, you must also be prepared for a decrease in your own SaaS needs—because of business cycles or a shift in business emphasis. It's thus important to be able to decrease payment levels. Provisions for this must be included in the initial license negotiations in addition to such

factors as software escrow and data retrieval.

Cloud and XaaS have given us new capabilities and also new risks. Yet the old risks are still with us, including media and format obsolescence. These are serious risks and require attention, but many people, companies, and agencies have been bitten by ignoring these. My colleague's distress was caused by exactly this data format problem.

Another interesting area to consider is the needs of libraries, which are repeatedly faced with this type of obsolescence. Do you remember when, for example, many technical books came with an accompanying 5 ¼ inch floppy disk? After a while, personal computers no longer came with the capability to read such disks, as the industry

moved to the 3 ½ inch “floppy” disks. (I still have an external drive for these, but don't think I've used it in the past five years. I'm not even certain it still works!) Reading either of these media formats is a serious problem today—even if the media has stayed readable over the years. (The same is true for seven-track tapes.)

Rather than catalog all of the media that has been available and when each has become obsolete, it's more productive to have a data preservation process in place, as do most university libraries. This process takes into account the usable lifetime of the digital media, the equipment needed to read it, the optimum times to move to new media, and which media should be used. This investment comes at a price, but it's important to decide whether the data must be preserved and, if so, to ensure the data can be affordably retrieved when needed.

The key concept is insurance against risks of future costs and damage to yourself and your organization. Having an exit strategy requires good planning and keeping up with IT trends and changes—but that's just what you'd expect from an IT professional. 

OVERVIEW:

hibu, Inc. is seeking a Systems Engineer II position for its King of Prussia, PA office.

RESPONSIBILITIES:

- Provide hands on system administration and support of various SAP systems within hibu.
- Participate in business critical projects completing complicated build work and supervise junior team members.
- Complete detailed project plans and architectural drawings and drive best practices in the implementation and execution of support processes.
- Act as subject matter expert in the SAP BASIS ECC6 space, advising management in the areas of SAP architecture and service improvement.
- Analyze business requirements for SAP integration and identify and plan opportunities for consolidating production systems as part of any upgrade or redesign of systems.
- Responsible for correction and transport management (TMS/CTS/CTS+).
- Subject to a biweekly rotation to cover 24/7 SAP system support.

REQUIREMENTS:

- Bachelor's Degree in Engineering, Computer Science, or Mathematics
- 5 years of SAP BASIS experience with an emphasis in configuration, analysis, and troubleshooting in Netweaver 2004S systems.
- Experience with any of the following SAP systems: ECC/BI/PI/EP/SolMan.
- Experience in installation, configuration, and maintenance with SRM/MDM in Windows/SQL required.
- Must have demonstrated knowledge of ECC 6.0.

Please apply at the following link: <https://jobs-hibu.icims.com/jobs/5652/job>

References

1. H.E. Schaffer, “X as a Service, Cloud Computing, and the Need for Good Judgment,” *IT Professional*, vol. 11, no. 5, 2009, pp. 4–5.
2. K. Finley, “The Dark Side of the Cloud: IBM Partner Gives Folks Two Weeks to Move Data,” *Wired*, 9 Sept. 2013; www.wired.com/wiredenterprise/2013/09/nirvanix.
3. M.R. Overly and M.A. Karlyn, “Source Code Escrow Agreements,” *A Guide to IT Contracting: Checklists, Tools, and Techniques*, CRC Press, 2012.

Henry Schaffer is a Professor Emeritus and Coordinator of Special IT Projects at North Carolina State University. Contact him at hes@ncsu.edu.