

The Loch Ness Monster And Other Myths:

Implementing Application Systems Solutions Successfully

By Warren S. Reid, ©Copyright, 1996, All Rights Reserved

The huge beast moved quickly and silently beneath the surface of the water with the slightest move of its tail. Above it were the legs and torsos of swimmers playing in the sea, their vacation going perfectly, just as planned.

They heard about the beast, had read about it, but would not believe they would actually see it. No...this could never happen to them! The beast opened its huge jaws and, with the force of a juggernaut, was upon them...

The jaws of many computer systems implementations are also wide open and ravenous. The firm did everything right: used a consultant, identified its business requirements first, prepared a request for proposal, reviewed and analyzed responses, checked references attended demonstrations, and, with the assistance of their attorneys, signed the "perfect" contract. Why, then, are they suddenly interrupted by "Nessie," the Loch Ness Monster? (Note: Nessie is many times bigger than a simple computer bug.)

They have heard and read about failing installations at other firms. They know about vendors successfully suing users and users successfully suing vendors. But they have done everything right. Why are they now being visited with frustration, disruption, missed deadlines, low morale, and inadequate systems performance?

The answers are not simple. My experience shows that several myths still exist regarding systems implementation, and these myths can cause even the best systems to fail. While having committed and supportive personnel is an absolute necessity, it is not enough. Planning and project management continue to be essential. Yet many myths cause these essentials to be ignored, especially where software application packages are involved, because people somehow continue to believe that tested/proven application packages allow for all sorts of shortcuts during implementation.

These myths appear rational and "convenient," which causes them to be perpetuated. Building a system on these mythical precepts, however, is like building your house on a watertrap. In fact, *if architects built buildings the way most companies implement systems the first woodpecker to come along would destroy civilization.*

As more and more attorneys are being invited to give their input during and regarding the systems acquisition process, on contracting for such systems, and on representing clients for "failed systems installations," *understanding and busting these myths* becomes a paramount instrument in the attorney's symphony of services.

When Is an Application Package Like a Custom-Designed System?

Contrary to popular wisdom, implementation of an application package requires most of the same steps as does a custom designed and developed system. While in the former case, there is much less of a need to design, code, and unit/string test than for a custom system, all the other steps are still required. Because of the limited dialogue between the designers and users (*i.e.*, your specific clients) during the development of the application packages, there is more likelihood that specific features and functions have been

implemented for mass consumption, and in ways not conceived or understood by your client.

Such nuances, misunderstandings, and legitimate multiple interpretations of the same functionality can wreak havoc on these otherwise innocent parties and cause a system to be deemed a failure. Systems developed from scratch for a specific client situation usually involve months of designers closely working together with users to develop specific requirements. The whole concept of "iterative prototyping" of developing accurate user requirements acknowledges that users themselves oftentimes don't really understand or can't define what they want or need until they have had a chance to work with a product. The simple act of having a new system will make the user "smarter" about his or her needs and thus bring about the need for even newer and still more improved systems. In other words, implementing and using a new system changes your needs as soon as you begin using it.

Myth No.1: There Is No Need for a Detailed Implementation Plan

Given the similarities and differences of application package installations to custom-designed installations, the following steps must be planned for:

- Planning all the segments of the installation phase including dollars, computing resources, staff resources, special resources, etc.;
- Implementing a systems steering committee and getting the company committed to the success of the project-not just involved in the project;
- Identifying and planning a hardware site, appropriate air conditioning, line conditioning, uninterrupted power supply, lighting, sound-proofing, special wiring, security, etc.;
- Changing company procedures to integrate and support the new system(s) (this includes re-designing and re-engineering new company policies and procedures and ways of doing things to get the most advantage of the introduction of the new system; documenting the new "methods/ways");
- Assuring the new system(s) interface and integrate with all other new and continuing systems(s);
- Planning sequencing and obtaining appropriate training for all affected individuals;
- Designing, ordering, and implementing new forms;
- Cleaning up data or preparing new data and loading all of your data onto the new system;
- Testing and accepting the new system/application package in your environment including: the development or approval of all test plans; setting up the appropriate test environment(s); overseeing the execution of the plans and assuring all the appropriate resources are available and trained; monitoring the test/correction iterative processes (configuration management and regression testing; and signing off on the systems and acceptance tests;
- Changing over to the new system and transferring to routine operational status; and
- Monitoring new production; identifying and documenting future enhancements and changes and corrections.

These steps and relevant substeps should be identified in a detailed project implementation plan. Start with the vendor's plan, challenge it, and prepare a plan that makes sense for your firm in terms of timing and content. Revise it as necessary as the project progresses.

While the vendor may have vast industry and application specific expertise, each user site is somewhat unique. Even if you contractually offload responsibility for successful installation/implementation, as some contracts provide for, your client must do all that he or she can to help manage the project to assure its success.

After the definition and approval of the implementation plan work steps and deliverables are complete, assign responsibility and start-and-completion dates, and identify hours to complete each task.

The major segments (and their objectives) of the systems installation phase of work are summarized in the Appendix, and are taken from an actual client situation, whose name has been changed.

Myth No.2: Your Staff Will Love It (or, a Project Team That Includes Users Is Not Necessary)

Nothing dooms a project for failure like a system forced upon those who must use it. Avoid this problem by assembling a project team consisting of users from all affected departments and functions, vendor representatives, and members of data processing (where applicable). Select a project manager. The project manager need not be a data processing technician, but should have the following characteristics to keep morale high and the project on target:

- Enthusiasm and high energy level (a cheer- leader, a yea-sayer);
- Leadership capability, communication and project management skills;
- Problem-solving ability;
- Knowledge of the firm as a whole and the interrelations of departmental functions; and
- A builder esprit-de-corps (the common feeling of purpose, pride, loyalty, and responsibility that unites a disciplined group).

While much has changed at lightning speed since the advent of the computer revolution (these past 45 plus years), some things that have persisted thousands of years still persist-and come up and bite (byte?) you. One such thing is peoples' resistance to change in a world of change. There is oftentimes very strong resistance to a system that will help the organization and the individual. There are several reasons, some unconscious, why this occurs, but fear is most often the culprit:

- Users feel insecure about the changes-they feel a threat to power, ego, and job security;
- Users feel uncertain about how the system will effect them-this fear is often based on lack of information regarding the new system;
- Users fear their job will become more complex and that they will be rendered incompetent or unneeded-especially in these years of downsizing, right-sizing, and outsourcing;
- Users fear change in the organizational relationship-with the new system affecting job content, areas of responsibility, and patterns of personal interaction; and
- Users fear job rigidity or time pressures-brought about by new schedules or by management of the new system itself.

A fairly new word in the dictionary is "cyberphobia" (some 15 years new) which means "fear of new technology." New technologies result in change. Even if the old ways are not working, at least they were or are familiar and comfortable. Plus, there is a common fear of the computer itself. Mitch Kapor, founder of Lotus, Inc., recently said that one of the most offensive frauds perpetrated on the American public was that computers and systems were being developed that were "user friendly." New systems for the most part are simply *not* user friendly-until you have had a lot of experience with them and until you realize how the system interprets and achieves your specific commands and requirements.

It is in these areas of fear and change where the project manager can and must help the most. By soliciting user concerns and ideas, and communicating through open discussion about project plans, progress, and impact of the system on employees, many of these fear barriers can be broken down. The project team consisting of energetic user opinion leaders must meet regularly to perform the detailed tasks on the time-phased implementation plan. At project team meetings, the project manager should lead the review of progress made, assist in identifying and resolving questions, fears, problems and concerns, and identify tasks to be accomplished in the following period. The project manager should have access to a steering committee

of top corporate executives to resolve higher level problems, policy issues, and organization conflicts, and to report overall progress. In today's Fortune 1000 companies, many boards, executive committees, and board subcommittees are being populated by chief information officers and vice presidents of management information systems.

Myth No.3: Top-Level Leadership Is Not Necessary; The Project Will Drive Itself

Below is an excerpt from an actual memo delivered to a client (names changed) regarding the organization and administration of the steering committee. It shows the importance and role of this top-level committee.

Key operating managers supervisors from various departments should be asked to participate in the Systems Steering Committee. The Systems Steering Committee will meet regularly to monitor the progress of the project and resolve major policy issues. The presence of the Steering Committee also makes management's support of the new systems apparent and helps assure the confidence and support of personnel at lower levels. Such support is imperative to the successful definition, selection, implementation, and continuing operation of the new systems. Because the new systems will be addressing applications such as laboratory, production, marketing, order entry and distribution, quality control, purchasing, and financial applications, this Systems Steering Committee provides the decision-making mechanism for allocating limited resources *across these multiple departments and interests*.

It is WSR Consulting Group's experience that only through joint interdepartmental and interdisciplinary participation and commitment will Russ Chemical Corporation accomplish an *integrated* production and accounting system with flexibility for future growth.

Systems Steering Committee members at Russ Chemical Corporation should be authorized to commit departments to using new systems and be able to speak for individual department policies and practices. Members will establish project priorities and provide policy direction (with the approval of the Senior Executive)-not direct management control-over individual project(s). Specified project teams (the sales team, the accounting group, production managers, etc.) will be responsible for the day-to-day carrying out of the detailed project plans in their respective areas as approved by the Systems Steering Committee.

The Systems Steering Committee will oversee data processing operations and evaluate equipment acquisitions during the next 1-2 years.

Members of the Systems Steering Committee, in our opinion, should include the Chief Executive Officer, relevant department heads, the Director of Finance, new Plant Manager, Director of Human Resources, Sales Manager, Director of Engineering, Director of Information Services, etc., and an outside consultant with appropriate skills. Vendor personnel will be invited when required/desired with the prior approval of the head of the Systems Steering Committee.

Having personally co-chaired Information Technology Steering Committees with some of the most successful, busy, and well known entrepreneurs and CEOs in the country to help steer large mission-critical or breakthrough technologies/systems, I know that a poor installation can bankrupt a whole organization at worst, or cause it to lose its market share (remember Bank of America's trust department system? or the Atlanta IRS office tax-payment processing system? or the AT&T telephone switching disaster? or what about Jurassic Park's system?). Getting top management committed through participation on a steering committee is simply indispensable management and necessary business! There must be a reason why the successful companies do it. Make sure your client operates/organizes that way too.

Myth No.4: The Vendor Will Implement the System for You

Most vendors are in business to develop and sell a product. Few are organized structurally or financially to attract and maintain appropriate support staff for all their installations. Those that have larger staffs charge \$75 to \$175 per hour for implementation services above a typically small number of "free" hours. The implementation process can be an outstanding forum for users to learn and become comfortable with the system, experiment with it and identify improvements and simplifications during the less threatening period before the system becomes "live." Involving users early in the implementation process typically permits staff to develop a positive attitude and success mentality regarding the change. Because a company usually can control the time of its staff better than it can control the vendor's staff, the user/vendor implementation team approach insures more successful implementations within the required time frame.

Myth No.5: If a Package Doesn't Fit, Modify It

Nothing in life is perfect, especially application software. Application software is typically much less expensive than custom-developed software and almost always can be implemented in a much shorter time frame than a custom system. The 20 percent to 40 percent concession you must make on certain nice-to-have features that would be available in a custom system is more than offset by the cost, convenience, and ongoing vendor support, the 85 percent to 95 percent or more required features' availability and reliability advantages of the application package.

I usually tell clients, "If you feel the urge to modify a package, lie down . . . until that feeling goes away." Most vendors will not warrant or support their package if you modify it. In addition, all future vendor enhancements and releases must also be similarly modified and tested by you to keep your system current.

If you modify, use an approach that will not compromise or corrupt the integrity of the application package. Either develop programs that process files after the application package is through processing them, or work with pre-processors, i.e., programs that process data before the data is entered into the application package. Or, develop semi-detached, standalone modules that interface with the package at a single or very few points. This will make it easier to upgrade your system, isolate errors, and perhaps even allow vendor support for the changes in the future.

Myth No.6: Since the Package Is Already Written, Nothing Can Go Wrong...Go Wrong...Go Wrong

Application packages are traditionally designed to meet requirements of many companies within a particular industry or application area. Accordingly, they are developed using a series of parameters, codes, switches, and tables which are initialized for your particular firm. This initialization process represents a tailoring or "customizing" for the system for your specific client environment. Such parameters can include:

- Report formats;
- Audit trails, control reports, and levels;
- Validation rules;
- Reporting responsibilities, levels, and other relationships;
- Classification schemes;
- Module integration requirements;
- Processing sequences;
- Security; and
- Backup procedures.

In addition, every computer system is part of a *larger firm system definition* that includes policies, procedures, people, and company objectives. An application package that works for your client's competitor

may not work properly or well for your client. The act of creating and setting up the parameters and tables, etc., noted above actually "customizes" the application for you and makes each implementation of the system unique - *either uniquely right for you or uniquely wrong for you.*

Also, putting up your new files is a major effort that can introduce major errors into the new system. You may have to write new procedures manually which integrate the new applications with the functions that take place before data is captured/entered into the system and discuss how the information generated by the system should be used.

Myth No.7: Because the System Is Used At Many Other Locations, a Systems Test Is Not Required

This is one of the most frightening myths because it appears very sound on the surface. Often a user will select a system on the basis of a vendor demonstration and a viewing of the application at a live reference site. However, as noted above, once all the system's codes, tables, and parameters have been set, and once the other system elements are considered (*e.g.*, people, policies, procedures, and objectives), the application is unique. An appropriate systems test is needed to check and assure that *the system is operating as the users expect and the users are operating as the system expects.*

Communications regarding systems functions and capabilities are frequently misunderstood by vendors and users. It is only during a rigorous test, with carefully planned test data and results that "flex the system," that the users will appreciate the nuances and functionality of the system (*i.e.*, how a negative number in inventory quantity affects the turnover ratio, exactly how back-orders will be allocated, how post-closing adjustments will be handled/allowed, etc.).

Myth No.8: Putting Your Files on the New System Is a Snap

All firms have a significant amount of data to be entered into the new system before it is ready for use. Such data include client, inventory, vendor records, charts of accounts and account balances, and other items unique to your firm.

Putting up your new files or reformatting your old files (called "converting") is a time-consuming task that must be planned and executed very carefully and quickly. Appropriate cut-offs, reconciliations, and balancing routines are necessary to assure all data is entered accurately, completely, and in synchronization with other related data.

This is the time to analyze your parts/inventory, customer/client, vendor/supplier, and account coding/numbering schemes to assure that they provide the system with the ability to give you the reports and analyses you need. Correcting/changing a bad numbering/coding system later will take great effort at best, or may never be corrected at worst - and limit the value of your system.

Schedule overtime and part-time help, if necessary, to get the data in quickly. If you don't enter your data quickly, it may take months to learn whether the system can or cannot do what you want. Every weekly delay reduces your chances of getting a no-charge resolution.

Myth No.9: An Acceptance Test Is Not Needed Once a Systems Test Has Been Performed with Vendor Data

Vendor data typically is developed to show a system in its very best light, including how quickly a system responds to inquiries. Once all the parameters are set up, it is critical for users to *develop their own*

acceptance test data and situations to assure that all key features of the system are operating as anticipated and required by each user. The purpose of this test is not to say that the system is fine, but to find any problems or bugs in the system before going live with production and before fully paying the vendor.

Myth No.10: Vendor Training Is of Limited Benefit

Many firms wrongfully assume that people who have experience with accounts receivable or scheduling or order entry systems on a manual or previously automated basis can automatically bring those talents into using, controlling, and managing a new system without additional or specialized training. Others believe that the documentation provided by the vendor will supply all the training that is necessary.

My experience shows that training is a critical factor, *required* to allow individuals to learn and *experiment* with the system in a supportive environment where questions can be asked and mistakes can be made. The confidence and team building developed during training sessions, as well as the relationship developed between the user and the vendor, can be critical to the ongoing success of the installation. In addition, most users' manuals and documentation are developed for reference purposes and not as a substitute for training.

You should keep several concepts in mind when scheduling and planning your training:

- Be sure that everyone who needs training receives it. This will prevent having to pay extra later. More importantly, this can stop untrained users from making costly mistakes.
- Try to have the training off-site (preferably at the vendor's site) or at a specially developed user training lab. This should, hopefully, stop everyday business disruption and allow personnel to fully concentrate on learning the new system. If you must conduct training at your site, try to hold all phone calls and minimize disruption.
- Don't let the trainer leave until the training is satisfactorily completed. You've paid for training and are entitled to a complete and competent job and functioning staff.
- Perform the training in phases. It is difficult for people to learn everything in one concentrated period. By having one session now and a second one in, say, 15 to 30 days, questions can be accumulated and bad habits can be corrected before they are too deeply ingrained.

Myth No.11: Don't Look Back!

Those who do not learn from the mistakes of the past are doomed to repeat them! This could not be more true than for systems implementations - whether mainframe, mini, micro, networked, client-server, or mixed computer configurations. A critical segment of work that is oftentimes omitted is the conduct of the post-installation review.

This segment is conducted after the new systems have been implemented so that management may assess the success and shortcomings of the systems efforts. Since the objective is assessment, timing is important. It is impractical to perform this segment until the system has operated long enough to produce measurable results (usually two months after the systems have been converted and accepted). The systems steering committee is responsible for authorizing the commencement of this segment and receiving and interpreting the results of the review(s) prepared by the project team(s) and the quality assurance group (if one exists). The review(s) should include: comparing/analyzing actual versus planned performance, costs versus benefits, anticipated versus actual resources required, business and systems lessons learned, guidelines for future systems installations and upgrades, pitfalls to avoid in the future, and estimating guidelines. These measures and lessons can be turned into standards which should be published and re-used and can be trusted over time.

Myth No.12: If It Doesn't Work, I'll See You in Court!

Litigation is no substitute for proper planning, contracting, and the development of escalating dispute resolution procedures. Time spent litigating systems matters is not really time well spent when you take into account the objectives and business reasons for developing, buying, or licensing in the first place. While those objectives and benefits are not being met, your staff will be spending untold hours reviewing documents and mistakes of the past. There are two sides to every story, and telling your three to five year old story to a judge or jury who many times has little understanding of the technical and business issues involved is a very expensive, time-consuming, gut-wrenching and *unpredictable* exercise. Use litigation only as a last resort - and then again, only after mediation and arbitration before an appropriate mediator or arbitrator (or panel) with *business, technical, and legal expertise*. These persons do exist and can make the best out of a bad situation.

Conclusion

Each of these myths is a monster - a Nessie - not merely a bug that can be swatted away. Belief in these myths can result in huge time and dollar overruns, and disappointing results. If these myths are understood and dealt with effectively, there is no reason why an otherwise properly selected system should not go "live" successfully.

Knowing where the "Nessies" lurk and learning how to spot them will help protect any firm swimming in the ocean of new systems. "Just when you thought it was safe to go into the water again, it is! Come in, the water's fine."

APPENDIX

Systems Installation Phase

It is within this multi-segmented phase that the systems are actually installed. The segments are intended to accomplish the following major objectives:

- *Pre-automation* is the segment in which new procedures for controlling data processing will be implemented at Russ Chemical Corporation. Such standards within the data processing department will include evaluation of user personnel, testing standards and systems and user training, program documentation, standard backup/recovery/restart, audit trails and controls, operating standards, etc.
- *Install hardware and software*. The computer area will be physically readied to install the new system. Supplies will be procured, and the actual hardware and software will be tested using a series of vendor and operator tests.
- *Conversion planning* for order entry/production management/financial systems. The project team will identify the specific Files, tables, history, and parameters that must be set to operate and control the new system. A specific plan for converting data from the old system to the new system, and "proving" that the data has been transferred properly will be approved. A new chart of accounts, responsibility reporting relationships, and reporting summaries may also be defined.
- *Systems and acceptance testing and full conversion*. The project team will install the systems modules in an agreed-to sequence. Integration tests (to prove that each module works and all modules interface properly together) and user acceptance tests (that assure that the system can be operated effectively in Russ Chemical Corporation's environment by Russ Chemical Corporation people using newly developed policies and procedures and documentation) will be accomplished.
- *Monitor production*. Check that the system is operating as specified. Identify problems and areas for improvement to discuss with vendor to obtain timely resolution. Assure that staff is utilizing system as intended.

WSR CONSULTING GROUP, LLC

Management, Technology and Litigation Consulting

(818) 986-8842 Fax: (818) 986-7955

E-mail: consult@wsrcg.com

Encino, California

**Previous
Page**