The Information System Lifecycle

It's Tough
When There's No Can Opener

By Mark L. Symonds
Innovative Information Systems, Inc.
63 Nahatan Street
Norwood, Massachusetts 02062

In years past I had been an economist. I heard a lot of jokes about that profession in those days. The most memorable of which involved three people stranded on that old familiar desert island. One was a chemist, one was a hockey player and the other was an economist. They had nothing but one large can containing food and supplies.

While pondering their dilemma, the chemist suggested they let the container soak in seawater until it rusted through. That way they could get at the food. The others scoffed at him pointing out that they would waste away from hunger long before getting their first meal. The hockey player insisted they should heave the can against the palm tree to break it open. The economist heaved a condescending sigh and said "There's an obvious solution to this problem." The others quickly leaned forward, listening intently for the answer. The economist began:
"First, assume we have a can opener ..."

Information systems development poses similar problems. If one assumes we have huge budgets, a large, experienced staff, well-defined requirements and flexible deadlines then anyone can put together beautiful systems ... in theory. When we assume we don't have a can opener, things get interesting.

Despite what some software vendors say, systems do not come in cans. It is not just a matter of "open and serve." We all know that implementing systems requires careful planning, hard work and diligent execution. Indeed, defining, developing and maintaining information systems is a complex undertaking that requires many different kinds of skills and management techniques.

My purpose here is to outline the steps, issues and challenges in each phase of the information systems lifecycle. I will also illustrate some practical techniques for coping with these challenges and for avoiding some of the pitfalls inherent in each phase of work.

There are a great many temptations, blind alleys and wrong turns on the road to a successful systems implementation. There is always a scarecrow pointing down the road of custom programming saying "there aren't any packages to fit our business." There are the packages that seem to have six bedrooms and 4 bathrooms as well as a pool and tennis court, but turn out to be only a nice facade. It is important, then, to have a good road map and to plan the trip. It is equally important to be a diligent and careful driver.

Information Systems Lifecycle

The Information Systems Lifecycle is a conceptual framework that will provide the basis for effective and efficient systems development projects. There are four major phases in the lifecycle:

I. Information Planning
II. Information Design
III. Systems Implementation
IV. Support and Maintenance

Each phase has its own set of challenges, pitfalls and rewards. Each in turn requires different skills on the part of MIS management and staff as well as that of top management. Before exploring those details let's look at what is usually entailed in each of the four phases of the cycle.

Information Planning

In general terms, information planning is the process by which an organization determines what information it will need over the next three to five years to remain competitive in its industry or to gain a comparative edge. It also looks at how this information should be gathered, stored and distributed so as to make most efficient use of the data and other resources such as hardware and people.

The product of the information planning process should be a set of goals and a detailed plan for achieving those objectives. These goals are set only after careful study of how the organization operates now, how it should operate and of what competitors might be doing. The action plan will define various distinct projects to be undertaken, prioritize those projects and estimate the costs, benefits and impact of each.

Information Design

As the projects identified in the information plan are undertaken, they enter the Information Design phase. Here, the detailed requirements of information content and flow are defined. Analysts assist users in determining the types of data needed, the sources of the information and the required timing and presentation.

In most cases a software package will meet a high percentage of the requirements for an application. It is during this phase that analysts evaluate alternative packages against their list of requirements to find the one that is the best "fit." It is here also that any required modifications to the system are identified and designed. If a package does not make sense, all the details of a custom system are designed at this time.

Other important steps in this phase are planning the conversion of data to the new system and sizing the hardware needed to support the application.

Systems Implementation

This is where proper planning and design pays off. Lack of it shows up very clearly as well. Tasks include detailed design and programming of package modifications or custom programs, user training, procedures development, comprehensive testing and conversion of data.

It is at this stage of the systems lifecycle that decisions and details can no longer be put off. A well-planned project will require a minimum of redesign and unforeseen effort during this phase. The potential for problems is well-known. One of Murphy's Laws is that the first half of a project takes ninety percent of the money and time allotted. The other half takes the other ninety percent.

Maintenance and Support

User requirements are not static. New ways of looking at the data emerge; new data need to be captured; new technology allows more efficient processing.

This phase involves ongoing support of users for questions and problems. It also entails bug diagnosis and correction and development of enhancements. This work is very different from the other phases because it is ongoing and sometimes repetitive.

Eventually it comes time to reevaluate the existing information architecture. The current systems are mature and have been modified and enhanced for some time. Often, an individual system can be studied and replaced without affecting the others. Periodically however, it is necessary to take a fresh look at the overall approach to information processing. A new Strategic Information Plan may confirm some approaches and highlight bottlenecks and inadequacies in other areas. So the process begins again to ensure that the organization has the complete, accurate and timely information it needs to maintain its competitive edge.

CHALLENGES AND TECHNIQUES

I. Information Planning

The strategic planning phase often poses the most difficult problems for MIS management. The concept of a Chief Information Officer has only recently gained a foothold in the Fortune 500. It is a very rare phenomenon for small to medium sized organizations to recognize the competitive and strategic importance of information planning.

A. Selling the Idea to Top Management

I've seen many projects fail due to infighting and lack of cooperation among departments. Commitment from senior management is essential to ensure that each user area do what it can to help attain the timely and successful completion of the project.

The key to getting the top people "on board" is speaking their language. Top management is accustomed to getting formal reports from other areas such as finance and operations.

Formal, informative reporting in terms they understand will earn respect and confidence. To the extent possible, estimates of cost savings, both tangible and intangible, should be provided. Over time, this practice will bring more attention to the importance of information processing.

B. Making it Pervasive

There is a great temptation to study carefully the areas that are most interesting. There is a tendency to gloss over ones that may yield large benefits despite the lack of glamour. Voice response order entry may get a lot of attention even though the system cannot allocate inventory properly and loses backorders.

It takes discipline to do a thorough job of information planning. It is essential, though, to touch every box on the organization chart.

C. Keeping Focused

An information may be well-planned and designed to study all important areas of the organization and still run into problems. It's a challenge to remain focused on the scope of the project and not to get distracted by fun details that should be studied later.

II. Information Design

Since the products of the design phase are less tangible it often does not get the attention it deserves. It is the thought of many that the design phase ends when the budget runs out. The major issues here are completeness and organization.

A. Complete Design Before Coding

The biggest temptation here is to start coding and seeing some results before the whole system is scoped out. Management and users are eager to see results and the staff is excited and anxious to get to "the fun part."

The fun part becomes a disaster, though, when the house is half-built and the bathroom and kitchen have to get moved to the other end.

B. Get Good User Input

Don't skimp on user interaction with the design staff. After the initial requirements definition users must remain available for questions and clarification. This is more easily accommodated when analysts batch their questions instead of interrupting users constantly. They typically have full-time responsibilities in addition to the development project.

Analysts should understand the business functions under review. A good analyst will lead the user through the universe of features and functions and not just take notes. An experienced business systems analyst can make this a two-way conversation drawing upon experience at other businesses or other areas. S/he can help the user define which ones are essential for doing his/her job effectively.

C. Prioritize

User requirement lists must not be taken as gospel. A knowledgeable user will include wish list items that would make his or her job easier and provide better information. The analyst should help the user rank each item as 1) Required, 2) Very Helpful or 3) Nice to Have. The nice-to-haves are often difficult and expensive to implement... especially in comparison to their benefits.

It is often the case that the users are not prepared to take full advantage of all the features on day one of the new system. The core pieces can be put in place first as long as appropriate "hooks" are provided to accommodate future functionality. In any case, generalized hooks should be built in to reduce the impact of unforeseen future requirements.

D. Be Creative

Too often we are constrained by past standards and perceived limitations. Creativity is the mother (or at least the mother-in-law) of greater efficiency and productivity.

III. Systems Implementation

Whereas the products of the design phase can be somewhat nebulous, the implementation segment has very concrete deliverables. It is here that poor execution of the previous phases really shows up.

If the planning and design are carried out diligently, the implementation will be much smoother, but there are still pitfalls.

A. Supervision and Resource Management

There is a paradox in most management situations. You want the most qualified person for a given task but you also want the individual to grow and learn. A good manager must balance these opposing ideals.

The key here is to delegate and then supervise carefully. The individual should have to reach to accomplish the tasks but not be so lost as to get discouraged. Proper training is an important element. Some specialized education will give a staffperson a sense of worth as well as additional tools to perform the job.

Making deadlines and expectations clear must be coupled with providing the means of accomplishing them. In this way project personnel will have more control over the outcome of their piece of the job.

B. Testing

Testing is not given its due often enough. The testing process should begin in the detailed design phase. It is there that the programmer/analyst defines the conditional logic and should identify test cycles and conditions. Rigorous unit testing will make the all-important integration test go more smoothly.

Package acceptance testing is also an oft neglected procedure. Few packages are 100% bug-free. I have also known there to be occasional errors when installing such software. Diligent acceptance testing not only helps shake out any problems but gives the in-house support staff a much better understanding of the programs and data structures.

C. Change Control

How often is it that users sign off on the design and are not heard from again? Changes to design during this phase cannot be made willy-nilly. The pressures of an implementation project usually result in such changes being thrown in without adequate forethought.

Some larger shops will not process any modification requests until after the system, as designed, is implemented. Of course, the tighter the design the easier it is to enforce such a rule. Again, proper design should reduce the call for last-minute changes.

IV. Ongoing Support

Maintenance programming is often thought of as the mailroom of MIS: not much excitement and not much growth potential. It doesn't have to be that way. One can distinguish himself here by providing consistency, creativity and organization.

A. Organization

Users need an effective means of reporting their problems or enhancement requests. There is room for creativity in defining the methods to be used.

A good approach will involve meaningful change request or problem report forms as well as meetings with analysts if more explanation is needed. These are complemented well by "user group" meetings which serve as an open forum to discuss system use and and proposed enhancements.

B. Feedback

Another valuable aspect of user group meetings is to recap and explain the status of bug reports and new modifications to the system. My former boss once told me: "If you're giving someone something make sure they know about it." Users have to be aware of the changes made for them.

C. Prioritization

Communication is a two-way street. Users must also make their priorities clear. When there is a difference in opinion management must be prepared to step in to moderate. Establishing and reevaluating relative priorities helps assure that scarce resources are put to the best use.

The information systems lifecycle is fraught with perils, pitfalls and temptations. Many can be avoided or at least minimized with diligence, proper planning and discipline.

The potential rewards of well-defined and carefully installed systems are enormous. A sound information architecture will provide a solid foundation for creating or maintaining a company's competitive edge.

Not even canned systems come with can openers and simplifying assumptions won't fly in the boardroom. Organization, hard work and resourcefulness are the next best bets.