EDP ANALYZER

©1977 by Canning Publications, Inc.

DECEMBER, 1977
VOL. 15, NO. 12

PROGRESS IN PROJECT MANAGEMENT

Work continues on new methodology for managing application system development projects. The goal being sought is a means to complete projects on schedule and on budget, and to have users satisfied with the resulting systems. While this goal does not seem unreasonable, particularly to user management, it can be a very difficult goal to achieve in some environments. In this report, we discuss user experience with two new methodologies aimed at achieving this goal.

The City of Tacoma, Washington, with a population of 157,000, is one of the major cities of the Pacific Northwest. The city government is performed by an elected city council, supported by an appointed city manager (who administers the city departments) and an appointed public utilities board and a public utilities director (who administer city-owned utilities). These utilities include electric power, water, and a small railroad.

The City's data processing department serves both the public utility departments and the city administrative departments, in a service bureau type of operation. The data processing department has an IBM 370/135 that operates under DOS/VS. The system is scheduled 24 hours a day, five days a week. It serves 23 CRT terminals and two remote job entry terminals. The City has a staff of 22 analysts and programmers.

In 1975, data processing management was unhappy with the department's system development performance. Good systems were being developed to meet user needs—but the projects were too often behind schedule and over cost. So management started to seek a better way to plan and control the system development process.

In October 1975, a department representative attended a seminar given by J. Toellner & Associates, of Los Angeles, California, on the company's new SPECTRUM-1 project management system. SPECTRUM-1 looked like what the City was seeking. After an investigation of the system, permission to purchase the system was obtained in December 1975. SPECTRUM-1 was installed in January 1976 and the first project was planned under it the following month.

What has happened to their system development process in the intervening months has pleased both the data processing personnel and the user departments. We saw a project status board which displayed 21 projects controlled by SPECTRUM and 6 old projects not under SPECTRUM. The 21 projects under SPECTRUM were essentially on schedule and on budget. The 6 old projects, which had been so far along in development that they were not put under SPECTRUM, were not in as good a condition. All were behind schedule and two had significant cost overruns—up to several times the original budgets. While SPECTRUM-1 has not eliminated all cost and time variances, it has greatly reduced them, say the people at the City of Tacoma.

What is SPECTRUM-1? It is a system development methodology for handling both large and small development projects, as well as enhancement and maintenance projects. The elements of SPECTRUM-1 include the following: (1) a method for getting top management direction and user department management involvement in data processing planning and control; (2) a structured approach to system development that involves three major phases divided into 13 sub-phases, plus defined quality review check points; (3) 15 to 20 tasks per sub-phase, with defined methods to use and work products to be produced; and (4) estimating guidelines for determining the resource requirements for each phase. These estimating guidelines take into account such factors as team size, team experience on this application, number of users, user availability, and so on. There is one manual for each of the 13 sub-phases. Each analyst, programmer, project manager, and supervisor requires only a single book specially written for that person's job function. Additional books of reference standards are available for staff use as needed. Documentation standards are provided for all project documentation. Separate manuals are available for the administration of software contracts and the control of purchased packages. For more information on SPECTRUM-1, see Reference 1.

"Our first project under SPECTRUM-1 almost drove people up the wall," said the Tacoma systems and programming manager. "It forces a discipline that is not always well received. But then there is a pleasant surprise when you find, at the end of a phase, that the work has in fact been completed, that there are no loose ends, and that schedule and budget are being met. An average user organization might be in awe of the apparent overhead of the methodology—but the results seem to justify that overhead."

The direction for the City of Tacoma's whole data processing program is provided by a data processing advisory board. This board consists of the assistant city manager, the assistant director of public utilities, two department managers, and the data processing manager. The board meets at least once a month, to review project requests and project status. It is the approving agency for all projects.

A screening function for project requests is performed by a data processing coordinating committee. This committee consists of representatives from the departments. It reviews project proposals and assigns suggested priorities to those projects.

On a small project, of up to one man-year in length, the project leader selects the appropriate tasks from an overall list of tasks. An initial estimate of times and costs is prepared at this time. By the end of the third sub-phase, much more accurate time and cost figures are developed. These more accurate figures are the basis for the "contract" price for the project which the user department must approve. At the end of the sixth sub-phase, which deals with the detailed design, another review of times and costs is performed.

On a large project, the project leader may identify some 250 tasks that will have to be performed. The initial estimate of times and costs is performed in much the same way as it is for small projects. But the City of Tacoma holds five reviews of the times and costs for the large projects, as opposed to three for small projects.

Each task is described in considerable detail in the SPECTRUM-1 documentation. Several pages of narrative discussion tell what is to be done in the task. References are given to any detailed descriptions of how to perform a task, such as how to perform interviews or how to do structured programming. A planning checklist is provided for each task, to help insure that all necessary things are done. Also, the work products to be produced by the task are defined, such as the forms that must be completed and the other documentation that must be created.

SPECTRUM-1 provides estimating guidelines for all tasks. Task worksheets list the different activities that must be performed. For each activity, time standards are shown—for minimum, average, and maximum complexity cases. The project leader selects the time standard he or she deems most appropriate for the activity. All of these times are then summed for a sub-phase, to give the first estimate of the man-hours required for that sub-phase.

The next step in time estimating is to adjust this first estimate by means of environmental weighting factors. Eighteen weighting factors are considered. These factors include team experience with the application area involved, user understanding of the requirements, and so on. These factors are both additive and subtractive. For instance, for a team of one person, the team factor is -0.1, while for a team of three or more persons, the factor is + 0.2. The appropriate values are selected for the factors, the values are then added, and the total is used to multiply the first estimate of man-hours. The result is the estimated manhours for the sub-phase.

If the total hours for the project as a whole, or for one or more sub-phases, looks unacceptable, analysis is made to determine just what can be changed. Toellner (Reference 3) provides a discussion of this time estimating procedure and the things to be cognizant of when using the procedure.

The City of Tacoma found that the major difficulties they had with their previous methods in meeting time and cost estimates were due to overlooking things. Whole tasks might be overlooked, as well as environmental factors that affect how quickly tasks can be performed. Now, with the extensive checklists provided by SPECTRUM-1, the City of Tacoma finds that these oversights are much less likely. There are still user-requested changes in the requirements in the middle of a project and these do affect schedule and budget. But the users recognize these are changes and that they will impact the times and costs.

The City of Tacoma sometimes contracts for software to be developed by outside organizations. The City now requests that SPECTRUM-1 be used for planning and controlling such projects and that SPECTRUM-1 documentation be produced. The outside suppliers sign non-disclosure agreements on the use of SPECTRUM-1 and then use the system for developing and documenting the City's application.

So the City of Tacoma has found SPECTRUM-1 to be most useful for managing their application system development projects.

Airborne Freight Corporation

Airborne Freight Corporation, with headquarters in Seattle, Washington, is the second largest air freight forwarder in the U.S. The company forwards air freight by using commercial air lines, air taxis, and commuter air lines. It has offices in 77 locations in the U.S. and in 10 foreign locations. Annual revenues are over $132 million and the company employs some 2,000 people. For its data processing, Airborne uses an IBM 370/148 batch system and a 370/145 on-line system tied to over 200 terminals. There is a development staff of 16 people for system development, maintenance, and production support.

In 1975, a major consulting firm performed a study of Airborne's data processing function. One of the recommendations made by the consultants was that Airborne install a project accounting and management system, as well as a system development methodology. These were in addition to a system standards and procedures manual covering many other development and production functions. For the project accounting and management system, Airborne considered both inhouse development and purchasing a package. In October 1975, they chose PC/70, marketed by Atlantic Software, Inc. of Philadelphia, Pennsylvania. (We discussed PC/70 in our September 1976 report.)

Airborne initially chose to construct their own project management system, basing it on the approach and methodology used by the consulting organization. After three man-months of effort, this approach was discontinued. Several project management packages were evaluated and in January 1976, SDM/70 was selected. It, too, is marketed by Atlantic Software.

SDM/70 uses a phased approach to project implementation, with nine phases. The first three phases are concerned with determining requirements, considering alternative solutions, and developing the system specifications, in a language understandable by users. The next three phases are concerned with system design, program development, and testing. The final three phases deal with conversion, operation, and post-implementation review.

Six types of guidelines are provided. Methods guidelines describe in detail how each task is to be performed and give tutorial material where needed for understanding underlying concepts. Documentation guidelines tell how to prepare all documentation for each phase. Forms guidelines tell how to complete the required forms. Estimating guidelines provide estimating parameters and tell how to estimate costs and schedules. Administrative guidelines tell how to administer and control project activities. Management guidelines describe the roles and responsibilities of user and executive managers during the development process. In addition, special guidelines are provided for short duration projects and for on-going maintenance.

SDM/70 also provides for user-oriented documentation for the early phases of a project, when user involvement is so essential for identifying user needs and problems. Some of the other concepts upon which SDM/70 is based are the following. Each task has a fully specified end product, checklists of pertinent design considerations, and pre-formatted forms with required information clearly indicated. Commitments in terms of both costs and schedules are limited to the next phase. Testing is also phased. And SDM/70 provides a "release" approach for maintenance and enhancements. The guidelines also describe the use of improved productivity techniques (HIPO, structured programming, etc.) that we discussed last month, as well as technical quality review, training of users, and operating steering committees. For more information on SDM/70, see Reference 2.

After obtaining SDM/70, Airborne discarded the backlog of major projects that was then in existence. They restructured all of these existing projects into about eight major projects and then performed a phase one study (requirements definition) on the four most attractive projects. They then began work on the two most suitable projects.

Airborne now divides their project portfolio as follows. When a user department submits a request for service, the request is reviewed by a systems department manager. If the request clearly involves less than five man-days of time and is considered worthy, it is immediately referred to the production support function. The work is done without the use of SDM/70 methods but is documented according to SDM/70 standards.

If the work will involve over five man-days of time, the service request is given to a methods analyst. The analyst can spend up to four hours in investigating the request. The analyst talks to both users and technical people, and develops a concise statement of the problem, the possible benefits to be realized, and the size and scope of the project. If the project is less than two man-months in size, the vice president, systems, can assign the priority. Larger projects are referred to the company's steering committee.

The steering committee consists of the three top vice presidents; also, the vice president, systems, and the systems manager attend all the meetings. The committee meets monthly to review project progress and assign priorities for new projects. On a quarterly basis, the committee reviews the allocation of all development resources.

For each development project, Airborne sets up a user review group, consisting of upper management or staff of the affected user departments. These groups meet during the projects, particularly at the end of the phases, to review the documentation that has been produced and to compare progress against plan.

After more than one year of use of SDM/70, Airborne has found the following benefits. The phased approach has been of great help in communicating with user department management and the steering committee. It has also helped to establish credibility in the eyes of both top management and user department management. User management no longer expects end-to-end cost commitments, and recognizes that the early estimates will later be refined into commitments. SDM/70 provides for the uniform tasking of projects while still allowing a good amount of flexibility in project conduct. The methodology forces the planning of a project, which helps to prevent false starts and implementing the first solution that comes to mind. And the documentation gives uniform, comprehensive results and provides effective communication with users and top management and between systems groups.

Managing system development

System development is a complicated function. It is much more complex than it might first appear to be. One of the reasons for this deceptive simplicity is that, with manual systems, the people involved with the system take corrective actions in order to make the systems work. To a certain extent, manual systems are self-correcting. So the system building process might appear to be simpler than it really is.

To illustrate the breadth and depth of the subject, here are some of the aspects involved in managing system development. (The dates in parentheses indicate our previous reports in which we discussed these subjects.)

TABLE 1

MANAGING SYSTEM DEVELOPMENT

Senior management guidance and control

Use of steering committees (May 1975)
Long range plans for data processing (August 1975)
Formal project initiation and funding (August 1975)
Sub-divide large projects into small ones (October 1970)
Management reviews with "creeping commitment" (May 1973)

Project management

Phased approach to project development (May 1973)
Comprehensive task checklist
Time and resource estimating procedure
Defined work products for each task (May 1973)
Schedule preparation procedure (September 1976)
Project progress control systems (September 1976)

Design and construction

Requirements determination methods (July 1977)
Methods for handling complexity (July 1977)
Installation performance standards (August 1975)
Quality review procedures (July 1977)

Staff development

Career paths for staff members (August 1971, July 1975)
Training programs (August 1972)

Why all of this complexity? There are lots of things that have been and are being developed in the world that do not seem to involve this variety of considerations. Why are information systems different?

Our position is that information systems really are *not* so different. Consider, for instance, the following two situations of building a house.

Situation one. A single house is built on a site isolated from other houses. It is built by a very experienced craftsman. This craftsman has built many similar houses before. He is able to lay the foundation, do the rough and finish carpentry, the plumbing, the electrical work, the painting, and so on. The customer may have to give this craftsman only a rough sketch of what is desired, plus some constraints such as a target maximum cost figure, in order to get a satisfactory house. Further, this method can work for houses ranging from very modest to quite outstanding in quality.

Situation two. Many houses are being built in an urban area. The houses are being built by a large labor force, consisting of specialists in a wide range of skills. One group of people lay the foundations, another group does the rough carpentry, still another group does the plumbing, and so on. But in addition, generations of experience with this type of construction has shown that other things must be considered. The urban area needs zoning laws, so that the new houses are compatible with existing construction. The urban government must have a process of requiring detailed drawings and specifications on the houses, plus a set of building codes (standards). The drawings and specifications must be checked and approved and building permits issued before construction can start. Further, these drawings and specifications must be prepared in a manner to be useful by the different building skill groups involved. The urban government must also perform an inspection process during construction, to make sure that the construction codes are being met. Finally, the houses must be built according to the plans; any changes made during construction can result in delays and extra costs.

As a matter of fact, experience with construction has shown that even more "overhead" is required. When the remodelling of a house is contemplated, again the plans must be prepared and approved before the work can start—to make sure that construction codes are followed. And even with maintenance, some types of maintenance work must be done by state-licensed craftsmen, as in the case of electrical work.

Our point is that the construction of a tract of houses in an urban area has many of the aspects of complexity as does the building of multiple, integrated application systems in a multi-programming environment.

It is when one moves away from the individual, stand-alone system built by one or more very skilled craftsmen that the true complexity of the process becomes apparent. The "overhead" needed to properly manage this complexity begins to stand out when numerous, inter-related systems are being built by "average" craftsmen.

In our May 1973 report, we discussed user experiences with a methodology developed by Touche Ross & Co., embodying a phased approach to system development. And in our December 1974 issue, we discussed some user experiences with Pride, a widely used packaged methodology for system development. In the interim, two other methodologies—Spectrum-1 and SDM/70—have reached the marketplace which incorporate somewhat different points of view. We are thus returning to this subject area in order to give a somewhat broader perspective of it. We believe that this is an area in which data processing management will continue to be interested.

A comment on terminology. In our September 1976 report, we discussed "project management systems" for project planning and control. These were mechanized systems for helping to plan the allocation and scheduling of development staff members and for keeping track of actual work versus plan. While the term "project management systems" is widely applied to such systems, perhaps the term "project control systems" or even "project planning and control systems" would be more appropriate. These project control systems represent only a small portion of the overall project management subject area.

We talked to numerous people about project management systems, in the preparation of this report. We would particularly like to acknowledge the contributions of John Toellner, whose company developed SPECTRUM-1, of David Katch who developed SDM/70, and of Michael Mulcahy, Director of Data Processing for the Municipality of Metropolitan Seattle, Washington, plus, of course, the people at the City of Tacoma and Airborne Freight.

What is wrong with older methods?

A still-all-too-familiar problem is the case where an organization's application system development projects are behind schedule, over cost, and end up not really satisfying the users. The data processing field is making progress toward solving this problem, but difficulties still exist.

What are some of the shortcomings with today's "conventional" methods of system development? First, there is usually a voluminous flow of requests for service from user departments into data processing departments, asking for new systems, enhancements to existing systems, and maintenance on existing systems. These requests are so voluminous that the data processing staff is kept constantly under pressure. Much of the time of the development staff is spent on the maintenance and small enhancement projects, to the point where it is hard to find time to do larger, more meaningful projects. Many of these maintenance and enhancement requests are due to the fact that the application systems were not developed properly in the first place.

Perhaps even worse, when the larger, more meaningful projects *are* undertaken, the man-hour requirements are seriously under-estimated. Tasks are overlooked, programmers jump into programming before design is completed, false starts occur and rework must be done, time estimates are made on the basis that the best people will be doing all of the work, and so on. The natural result is that projects fall behind schedule and go over budget. As such a situation develops, there is a bad psychological effect on the team working on the project. The team sees that the time and cost estimates are unrealistic and that those estimates cannot possibly be met. The team members feel that they have been put in an untenable position and get depressed.

One person that we talked to gave the following reasons for time and cost overruns on projects. About 70% of the time, he said, the cause for the time and cost overrun was just poor estimating in the first place. In another 10% of the cases, the cause was changes in the requirements requested by the users. Another 10% of the time, the development staff members were not as productive as expected. And the final 10% of the time, data processing management was at fault. Of course, for any given project, the troubles can be caused by two or more of these reasons.

If poor estimating is the main cause of the troubles, it is probably due to two main reasons. For one thing, whole tasks or activities can be completely overlooked. The other main reason is that the environment in which the work is to be done is not fully considered. Yes, it may be realized that new, inexperienced staff members may have to be used on a project. But it may not be fully appreciated that a large team size will slow down the work of each team member, or that the knowledgeable people in the user departments will not always be available when needed. And it may be overlooked that the team members will be interrupted in order to perform "fire fighting" maintenance work on other systems.

Another person we contacted gave his ideas on overruns. One reason is using poor estimating techniques and not adjusting the estimates at key milestones when the work is better defined. Another is the lack of an effective freeze process which helps keep a project from ballooning. A third reason is the mixing of development and maintenance work, where maintenance keeps taking time away from development.

These are only a few of the shortcomings of the older, "conventional" system development methods. None of these shortcomings is particularly difficult to understand. The problem is that there are so many shortcomings. And they tend to be overlooked unless a systematic means is used to consider each one and ask, "is this a potential problem on this project?" The upshot is, then, that better methods are needed for developing application systems.

Today's project management systems

Today's project management systems (PMS) attack the above problems on several fronts. Some of the principles that these systems use are the following.

Doing the right things. Top management guidance of the whole data processing program is an essential part of the new PMS. The vehicle most commonly used for obtaining this participation is the steering committee, consisting of members of executive management. The methodologies make such a strong point of the need for this participation that top management finds it hard to refuse. Further, the methodologies describe the means by which these steering committees can perform effectively.

Doing things right. The modern methodologies stress the need for carefully determining user requirements, so that the problem is properly identified and defined. They urge that a variety of alternative solutions to the problem be considered, not all of which need involve the computer. If the best solution is selected and carefully implemented, then the need for subsequent enhancements and maintenance is greatly reduced. Further, the methodologies provide good documentation—for users, for computer operations, and for any subsequent enhancement and maintenance needs.

Establishing credibility. By providing a mechanism for making more accurate time and cost estimates, the new methodologies tend to build up the credibility of the data processing department's promises. Credibility comes from doing what was promised—meeting schedules, keeping costs under control, and producing an efficient system that meets user needs.

To give a more comprehensive picture of these new methodologies, here is a list of the characteristics of today's project management systems. Note, however, that not all characteristics need apply to any one PMS. Each system seems to have its strong and its weaker points.

TABLE 2
CHARACTERISTICS OF PMS

Overall direction and administration
- Senior management guidance
- Long range planning for data processing
- Careful determination of user requirements
- Resource usage planning and control
- Phase-limited commitment
- Post-implementation reviews

Phased approach
- Defined phases (and perhaps sub-phases)
- Defined tasks for each phase
- Defined work products for each task
- Phased testing

Detailed guidelines and checklists
- For:
  - Methods
  - Documentation and forms
  - Time and cost estimates
  - Maintenance, small projects, enhancements

Design and construction methods
- General description of how to perform each task
- Brief description of applicable technology
- Management of the data resource

Control methods
- Quality and progress review points, at phase ends
- Approval to continue only through next phase
- Tie-in with project control systems
- Control of changes to design, after freeze points
- Administration of software contracts

Training and consulting assistance
- Rewrite of manuals to fit company's standards
- Retention of desired forms, procedures, standards
- Assistance in setting initial project priorities
- Post-implementation review of the PMS itself
- How to use the methodology, conduct interviews
- How to capture user requirements

As we said, not all of the new PMS have all of the above characteristics. But the list gives reasonable idea of the features of these new methodologies.

How do PMS control system development?

The new project management systems seek to get system builders to do the right things, then to do those things right, and thereby to build up credibility.

Users tend to want to get their new systems, enhancements, and changes onto the computer just as fast as possible. The new methodologies help to make sure that the problem is identified and defined correctly and that the best of several alternative solutions has been selected. These steps may seem to slow down the projects, in the eyes of users, but they make ultimate satisfaction much more likely.

By providing extensive checklists, which apply to a large variety of types and sizes of projects, the new project management systems make it less likely that tasks will be overlooked.

The new PMS provide detailed task definitions, including the work to be done and the documentation to be produced. Standard forms provide consistency and make sure that all necessary tasks have been considered and performed.

Time estimating guidelines may be provided, to help insure that all relevant factors are considered when the estimates are prepared.

Quality and progress review checkpoints, at the end of each phase, help insure that each phase has been performed completely and correctly. In theory, work does not commence on the next phase until the previous phase has passed its checkpoint. In practice, if everything seems to be going well, project management often will take a gamble and start work on the next phase before the previous checkpoint has been passed. However, project management recognizes that this *is* a gamble, and generally tries to get the review performed as soon as possible.

Finally, the new PMS impose a control over all changes after preliminary design has been accomplished. This principle applies at both the system level and at the program level. Formal change control helps insure that new ideas cannot be "sneaked in" to the system plans. And the methodology requires that any change which *is* considered be properly analyzed before it is incorporated and properly documented after it is in.

These, then, are some of the ways in which the new PMS control system development so as to greatly reduce problems that have plagued the earlier methods.

Is all this overhead necessary?

The use of a modern PMS appears to involve a lot of overhead. "Is all of this overhead really necessary?" you might ask. "Our users get edgy if too much time is taken up with planning a project, instead of getting on with the job. Do we have to use all of the formality of these PMS?"

The answer seems to be, as is so often the case, both Yes and No. Let us consider the No side of the answer first. If you are building mostly independent, stand-alone systems with highly skilled people, and if there are just some aspects of your system development process that you are unhappy with, then you may not need the formality of a full PMS. You could study the characteristics of a full PMS and select just those features designed to give the improvements you seek. (This point is controversial, as we will discuss shortly.)

The Yes answer applies to those situations where a number of inter-related application systems are being built and used, a data base may also be in use, and there is a mixture of novice, junior, and experienced staff members. In such situations, the likelihood is very great that the formality of a full PMS is needed.

Of course, there is a wide range of situations between these two extremes. Our opinion is, though, that a good fraction of the computer-using organizations would benefit from the use of a full PMS. For instance, we have seen cases where an on-line time sharing service could well have used the small-project version of a full PMS for developing a stand-alone application system for a small business. We have seen other cases where an independent software firm could have used a PMS to advantage for develpoing a set of applications for a small business. All in all, we believe that the answer "Yes, the overhead of a full PMS is necessary" is true more often than it is not.

"Well, how about this top management involvement?" you might ask. "Top management has many demands on its time—and after all, the data processing budget only represents one or two percent of the organization's total budget. How can top management's attention be justified under these circumstances?"

A reasonable answer is that data processing is much like the central nervous system of the human body. It carries much of the information on which the organization depends. The central nervous system may represent only a small percentage of the total weight of the human body. But its importance far exceeds its relative weight.

In short, if application systems are not developed properly, at the very least substantial resources may be wasted. But even worse, the systems that are poorly developed may not serve the user departments well. And this in turn can affect the efficient performance of those user departments. Since the effects of good or poor system development are so wide reaching, top management *can* afford to spend some reasonable amount of time in the guidance and top level control of this function.

Just how much overhead is involved in the use of a full PMS? We tried to find out. It is difficult information to obtain. Perhaps 10% of the total man-hours in a project might be spent in PMS-type activities. Not all of this is extra time, of course. For instance, some documentation would have to be developed anyway. Then, too, when the elimination of false starts and rework are considered, it is quite possible that a PMS-controlled project uses less time and that future maintenance and enhancement times are also reduced.

What are the options?

But the question remains, does a user organization have to take all or nothing of a full PMS? The answer seems to be No. The *formality* of a full PMS might be desirable for most computer-using organizations, but there are still some options open. Let us consider what some of those options are.

"Voluntary" versus "mandatory" use. Some organizations that we have talked to have decided not to impose the PMS as a mandatory standard. Instead, they leave it up to each project leader to decide how much or how little of the PMS will be used on a given project. They hope that in time, the PMS will be accepted for all projects. Other organizations, however, have decided that the only way they are going to get their system development under control quickly is to impose the PMS as a mandatory standard on all projects. This is one option open to users.

"Enhance present system" versus "install new methodology." The user organization may already have an extensive set of installation standards that it does not wish to replace. In such a case, it may only seek to select certain parts of a full PMS and then modify those to fit in with its standards. At the other extreme are those organizations that feel their present methods are quite inadequate and would prefer to replace them with the new methodologies.

If the decision is made to enhance the present installation standards, then there is a series of options available as to which enhancements to make. These options include the following: (1) which phases of the PMS to adopt, (2) the number of standard tasks to prescribe for each phase, (3) the number of defined work products to be produced by each task, (4) how each task is to be performed, and (5) the degree of detail used in making the man-day time estimates for each phase.

One representative of a consulting firm that we talked to said that his firm advocated this "enhancement" approach. His firm had developed an almost complete PMS methodology. On consulting assignments, the firm's consultants would recommend that the client install only those standards that were missing from the client's set of installation standards, and that were considered desirable. This consultant claimed that the enhancement approach made it easier for users to get the benefits of a PMS, as opposed to installing a complete, new PMS.

There are, of course, counter arguments to this approach. One such argument is that the approach leads to fragmented solutions to specific problems, rather than to integrated solutions. Another argument is that when these fragmented solutions are developed, it is then quite difficult to develop an integrated solution.

"Full PMS" versus "abbreviated version." Even users of a full PMS may have the option of using an abbreviated version for smaller projects. One might argue that the abbreviated version could be used on the larger projects, too, in order to avoid some of the overhead of the full PMS. We did not encounter anyone who advocated this approach, but it is a possibility.

Another option is whether or not a project planning and control system is to be used in conjunction with the PMS. We discussed project planning and control systems in our September 1976 report. Given a task list and the resources required to accomplish the tasks (which could come from the PMS), the PP&C system can help plan and schedule the project, and can provide periodic progress reports showing actual progress versus plan.

It seems to us that if data processing management is reasonably well satisfied with the present application system development process that is used, then "enhancement" could be selected instead of installing completely new methodology. But if data processing management and/or user department management are reasonably dis-

·atisfied with the present methods, then the installation of a full PMS should be considered.

Installing a PMS

As we have pointed out in numerous past issues, many software products should be considered as tools, not solutions. That is, if the computer-using organization is in a "crisis" situation, the software package will probably not cure the crisis and may, in fact, only make it worse. It seems to us that a full PMS of the type we have been discussing, on the other hand, comes much closer to being a solution. The installation of a full PMS might alleviate many of the problems in a crisis situation for system development.

There are a number of problems associated with the installation of a full PMS, however. One organization that we talked to said that the user ought to allow about one full year for getting the system in and under proper use. Others agreed that there were problems in installing a PMS but felt that one year was a longer period than they required to get the methodology in full operation. In fact, significant parts of a PMS can be installed in from one to two months.

Following are some of the problems that are involved in installing a PMS. The PMS manuals may have to be modified so as to be compatible with the company's existing systems and procedures. Company-standard names and terminology, as well as certain installation standards and operating methods, which are just as good as those used in the PMS, should be retained. The rewriting of the manuals to incorporate such changes can either be done by the supplier (at extra cost) or by the company. This is a substantial activity.

Another problem has to do with setting up the manuals of the methodology so that they are suitable for use by the existing and near-future staff. Novice system analysts, for instance, will desire a good deal of narrative discussion of each task that they are to perform. Experienced system analysts, who understand the tasks but who are just becoming acquainted with the PMS, may want guidelines but not extensive narrative. And experienced system analysts who have used the PMS often may just want checklists. If the PMS has not been set up to meet such needs in the first place, then such required changes will involve a non-trivial task.

The first project that is planned under a full PMS may cause shock waves to go through the data processing staff and the affected user departments. The amount of the planning activities may seem excessive. And the man-days that are estimated to be required may seem unreasonable. Users may be suspicious that the new methodology just pads the times and costs, and may feel that the whole project should move faster. Experience will show whether or not the PMS activities and estimates are reasonable and realistic.

Most PMS use a phase limited commitment philosophy. This philosophy requires that each phase be completed, reviewed, and its work products approved before work begins on the next phase. As we indicated earlier in this report, in practice the "creeping commitment" philosophy is not followed exactly. If the previous phase has gone reasonably well and the project is close to schedule and budget, the project leader may start work going on the next phase before the previous phase is reviewed. The reason, of course, is to keep the project team members busy. But it should be recognized that this action is a gamble on the part of the project leader.

It was emphasized to us by one user that a PMS never replaces a good analyst. But a PMS does help a good junior analyst perform like an experienced analyst. It compensates somewhat for a lack of experience. It does not, however, make a good analyst out of an average one.

Some of today's PMS prescribe just how each task should be performed. Others provide a good deal of flexibility on task conduct. For instance, for program design and construction, some PMS may prescribe a "standard" method to be used for all programs. In contrast, others allow the programmers to choose among several methodologies, such as modular programming, structured programming, Warnier's method, and so on. Data processing management would have to decide just how much flexibility is desired in the use of the PMS.

Conclusion

As we indicated in Table 1 above, the subject of managing system development is a complex one. We have treated various aspects of the subject in numerous previous issues. Because the subject continues to challenge data processing management, we expect to return to some of the aspects of it in future issues.

As opposed to some software systems—such as data base management systems and project planning and control systems—which should be considered tools more than solutions, we think that project management systems can be viewed as solutions. Thus a data base management system should be installed only after an organization has its data pretty well under control; it should *not* be installed to get that data under control. A project management system, however, *can* be installed in order to get the system development process under control.

In Table 2, we gave the main characteristics of today's PMS. As we pointed out, not all PMS have all of these characteristics, but the table can be useful in helping to judge the scope of any particular PMS.

With full project management systems, we believe that the computer field is making real progress toward the goal of getting application systems done on time, within budget, and of satisfaction to the users.

REFERENCES

1. For more information on SPECTRUM-1, write J. Toellner and Associates, 4311 Wilshire Boulevard, Los Angeles, California 90010.
2. For more information on SDM/70, write Atlantic Software, Inc., 5th and Chestnut Streets, Philadelphia, Penna. 19106.
3. Toellner, John, "Project estimating," *Journal of Systems Management* (24587 Bagley Road, Cleveland, Ohio 44138), May 1977, p. 6-9; price $1.75.

One of the major computer field trends during the 1970s has been toward the use of the shared-data base that serves multiple applications. Numerous benefits are being obtained from data bases—but at the same time, they have brought into much sharper focus the need for assigned responsibility and control of the organization's data resources. The data administrator function is being set up and given this responsibility. And data dictionaries are being installed as tolls for the data administrators. Next month, we will describe some user experiences in "installing a data dictionary."

EDP ANALYZER published monthly and Copyright© 1977 by Canning Publications, Inc., 925 Anza Avenue, Vista, Calif. 92083. All rights reserved. While the contents of each report are based on the best information available to us, we cannot guarantee them. This report may not be reproduced in whole or in part, including photocopy reproduction, without the written permission of the publisher. Richard G. Canning, Editor and Publisher. Subscription rates and back issue prices on last page. Please report non-receipt of an issue within one month of normal receiving date. Missing issues requested after this time will be supplied at regular rate.

SUBJECTS COVERED BY EDP ANALYZER IN PRIOR YEARS

1974 (Volume 12)

Number

1. Protecting Valuable Data—Part 2
2. The Current Status of Data Management
3. Problem Areas in Data Management
4. Issues in Programming Management
5. The Search for Software Reliability
6. The Advent of Structured Programming
7. Charging for Computer Services
8. Structures for Future Systems
9. The Upgrading of Computer Operators
10. What's Happening with CODASYL-type DBMS?
11. The Data Dictionary/Directory Function
12. Improve the System Building Process

1975 (Volume 13)

Number

1. Progress Toward International Data Networks
2. Soon: Public Packet Switched Networks
3. The Internal Auditor and the Computer
4. Improvements in Man/Machine Interfacing
5. "Are We Doing the Right Things?"
6. "Are We Doing Things Right?"
7. "Do We Have the Right Resources?"
8. The Benefits of Standard Practices
9. Progress Toward Easier Programming
10. The New Interactive Search Systems
11. The Debate on Information Privacy: Part 1
12. The Debate on Information Privacy: Part 2

1976 (Volume 14)

Number

1. Planning for Multi-national Data Processing
2. Staff Training on the Multi-national Scene
3. Professionalism: Coming or Not?
4. Integrity and Security of Personal Data
5. APL and Decision Support Systems
6. Distributed Data Systems
7. Network Structures for Distributed Systems
8. Bringing Women into Computing Management
9. Project Management Systems
10. Distributed Systems and the End User
11. Recovery in Data Base Systems
12. Toward the Better Management of Data

1977 (Volume 15)

Number

1. The Arrival of Common Systems
2. Word Processing: Part 1
3. Word Processing: Part 2
4. Computer Message Systems
5. Computer Services for Small Sites
6. The Importance of EDP Audit and Control
7. Getting the Requirements Right
8. Managing Staff Retention and Turnover
9. Making Use of Remote Computing Services
10. The Impact of Corporate EFT
11. Using Some New Programming Techniques
12. Progress in Project Management

(List of subjects prior to 1974 sent upon request)

PRICE SCHEDULE

The annual subscription price for EDP ANALYZER is $48. The two year price is $88 and the three year price is $120; postpaid surface delivery to the U.S., Canada, and Mexico. (Optional air mail delivery to Canada and Mexico available at extra cost.)

Subscriptions to other countries are: One year $60, two years, $112, and three years $156. These prices include AIR MAIL postage. All prices in U.S. dollars.

Attractive binders for holding 12 issues of EDP ANALYZER are available at $6.25. Californians please add 38¢ sales tax.

Because of the continuing demand for back issues, all previous reports are available. Price: $6 each (for U.S., Canada, and Mexico), and $7 elsewhere; includes air mail postage.

Reduced rates are in effect for multiple subscriptions and for multiple copies of back issues. Please write for rates.

Subscription agency orders limited to single copy, one-, two-, and three-year subscriptions only.

Send your order and check to:
EDP ANALYZER
Subscription Office
925 Anza Avenue
Vista, California 92083
Phone: (714) 724-3233

Send editorial correspondence to:
EDP ANALYZER
Editorial Office
925 Anza Avenue
Vista, California 92083
Phone: (714) 724-5900

Name____

Company______________________________________

Address_______________________________________

City, State, ZIP Code____________________________

____