accident one of the traces was left on for 18 hours and took 17 cylinder of 3350 space. GTF itself seemed to take less than 5 minutes CPU time for that period. I used DYL280 to sort and summarize the records since we don't have SAS and don't have GTFPARS. The results were used to determine which COBOL modules to put into the LPA. I also removed some TSO modules from the LPA library based on the results. The information also suggested that even more TSO modules could be removed. Unfortunately GTF does not pick up SVC calls and certain other module use information when just the load, link and XCTL records are specified. The data gathered also suggested that some reduction of system overhead could be gained by the combining of modules.

The next step was to have our SE verify that I could safely remove JES2 and MICR modules. A couple of weeks later he got the confirmation from the Dallas systems center. I removed the FMID's by applying and accepting a USERMOD which deleted EJE1102 and EMI1102. This removed all modules from the systems and distribution libraries and all related PTF's from the PTS. A side discovery in this process is that when you delete a FMID, you also should delete the PTS entry for the FMID or you will continue to get maintenance. I went through my PTS intending to delete JES2 maintenance and instead ended up deleting TCAM 10 maintenance.

The surprises started coming when I deleted EMI1102 which looks like MICR but might better be called the obscure device FMID. I found we also lost support for OCR, tape cartridge readers, and diskette readers. Fortunately we don't have any intentions of having any of these devices on our system. Then I did a sysgen to apply MVS SP1.3.2 and found that the three sysgen macros owned by EMI1102 were unconditionally required by sysgen. Even more surprising was the fact that for SYSGEN purposes, a 3505 and a 3525 are in this obscure device category because Rochester owned them. We HAVE a 3505 and a 3525. Research with a very helpful person at level 2 sysgen verified that all of the 3505 and 3525 modules were owned by EDM1102 so that I hadn't deleted any of them and that the SYSGEN macros in question had last changed with release 1.3 of VS1. Thus I felt confident in restoring them to a special library from a backup tape and redoing the sysgen pointing to that library in the concatenation. Elimination of EMI1102 bought about 10K and of JES2 35k so that the savings were trivial. However the experience was useful and the data gathered showed that there are potential savings of 300k to 350k. If we remove the capability of running free VTAM2 in emergencies and on weekends, we can save 300k. Removal of some more TSO unused modules is the other area of simple reduction. If I could get rid of MSS from the target, or running libraries without doing a full sysgen and leave it in the DLIB's and continue to get maintenance I would do it. It seems theoretically feasible and clean to just apply a deleting user mod and not accept that mod. If and when we get rid of ISAM in our shop, those modules represent 40-50K of LPA that can be removed. Even now, if we were really tight on space we would experiment with moving them to SYS1.SVCLIB and see if the system would pull them into the address space.

In conclusion, while removing modules is the slow way to virtual storage contraint relief, it does have benefits and the excercise may tell various things that will help your system run better. Because there are a number of modules it is desirable to have in the LPA, it would be useful to easily eliminate unused modules from our running libraries and keep them maintained only in our distribution libraries so that we have them if we ever need them. This study also suggested architectural changes that could give at least some relief and this might be a good topic for follow-on discussions.

SESSION REPORT

SHARE

61	M560	A MANAGEMENT PERSPECTIVE OF ~~PROJECT MANAGEMENT~~	130
SHARE NO.	SESSION NO.	SESSION TITLE	ATTENDANCE
ADM		Carolyn Andersen	USF
PROJECT		SESSION CHAIRMAN	INST. CODE

A MANAGEMENT PERSPECTIVE

OF

PROJECT MANAGEMENT

Dr. Doris F. Bernardini

IBM Information Systems Management Institute
909 Third Avenue
New York, New York 10022

Session No. M560

Project ADM

INTRODUCTION

Our topic this morning is "A Management Perspective of Project Management." We're going to be focusing on three components of project management:

- Planning Projects
  - Using Project Life-Cycles
  - Establishing Phases and Milestones
- Controlling Project Performance
  - Estimating Resources
  - Controlling Change
- Managing People
  - Motivating Project Participants
  - Building Teamwork
  - Increasing Personal Effectiveness

I'll be making some remarks on these topics first. When I have finished, I'll invite you to ask questions or to give us the benefit of your own experiences relating to project management.

PLANNING PROJECTS

Planning: Achieving project results on purpose rather than by accident. We can contrast a planned project and a "happening" by looking at two different ways of building your dream house.

- One way is to hire a builder and, after a brief conversation, put him to work. He makes the decisions and orders the supplies as he needs them. Periodically you check to see how your dream house is coming along. Occasionally some problems will arise due to breakdowns in communications and some work will be redone. And then, too, some problems won't surface until so late in the process that the cost to change them would be exorbitant, so you must make the decision to live with the mistake or pay for a very costly change. Eventually your dream house becomes a compromise -- and you no longer consider the builder to be your friend.

- A second way to build your dream house is to start out with some detailed planning and then proceed with the construction. This should result in a document that at least helps reduce the misunderstandings and cuts down on the actual construction time. And hopefully the end result will be something like the dream house you had in mind when you started. One more thing: Planning -- an additional activitiy in this version of building a house -- did cost in time and money, but saves you many times over what you would have spent in rework, idle time, and in aggravation.

The analogy between building a house and building a data processing system is helpful to keep in mind, and particularly that of building a house with or without plans.

Every project manager plans to some extent, but not all of us plan in a formal way. Formal planning means:

- A written procedure
- A procedure that is uniformly and regularly applied

. A procedure that calls for written output

According to Philip Metzger, "too many programming projects are treated like mystery novels. You're left hanging by your fingernails down to the last climactic moment when it's suddenly clear which manager was the villain."

Surveys of project managers and programmers from many companies and government agencies have shown reasons for failure of a project to look something like this:

. Poor planning
. Lack of a project methodology
. Poor planning
. Lack of project management experience
. Poor planning
. Poor problem definition
. Poor planning
. Lack of change management procedure
. Poor planning
. Political pressure
. Poor planning

The list goes on and on. But notice that ever present is poor planning. This could be interpreted to mean there is no plan at all -- which means none has been put in writing. Or it could be interpreted to mean that some parts of the project have been overlooked. How, then, does one go about planning a project?

Planning is a process that starts with a mission, sets the goals and objectives to be reached, develops plans and procedures for achieving them, and assigns responsibility and accountability to see that it is done. It is an iterative process constantly cycling between goals and resources -- between what you want to do and what you are able to do.

The output of the planning process is a plan -- not the plan. As work is being performed, you as the project manager are monitoring it and comparing the progress to the plan. When your control system signals a problem, it usually means adjustments. So planning is a continuous process. It isn't just a one-time effort.

Planning is hard work. You need to decide what should be done -- and what should not be done. The choices are endless and often difficult -- which means the right person or persons need to be involved. It's a very complex mental task. Planning is a major function for any manager at any level. And it is becoming even more important when we consider the huge impact on the organization that many projects have. It behooves us to make every effort to have a successful project.

When then don't we plan better? We wouldn't be here discussing this topic today if everyone were doing a superb job of planning. Some of the reasons for not planning are these:

. "It takes too much time." We're too busy with a crisis to spend time on planning.
. "Planning is too expensive." We need our people on our major project activities.

- Planning doesn't allow flexibility." Our situation changes too fast; we can't get locked in.
- "Our project is unique."
- "Planning stifles creativity." Design and programming are more art than science.

For the most part, statements against planning are really statements against bad planning or overplanning. Some plans are too expensive, too inflexible, too stifling, and take too much time to create. But we need to understand why we plan. For example, one purpose of planning is to bridle creativity and to direct it toward the goals of the project. A programmer might like to use a creative way to accomplish a function, but is this really a requirement? So if planning helps to prevent this kind of activity, then it is serving a useful purpose.

If it's done carefully, planning provides many benefits. For example, it:

- Makes it easier to achieve specific results
- Helps management
- Assures higher quality project products

But remember that too little planning is not good -- and too much planning also creates problems.

Let's take a look then at some of what goes into the project manager's planning for a project.

I. PLANNING PROJECTS

a. Using Project Life Cycles

A project manager's first responsibility is to figure out how he can make his project orderly, well managed, and successful. A project can be just that. It can meet the definition of a successful project: On time, within budget, according to specifications.

The major reason so many projects fail to meet these criteria is lack of control on the part of the project manager -- lack of control because things are not kept visible enough.

- The needs, requirements, and opportunities are often not visible -- or at least not clear -- because we don't take time to make them explicit.
- Design and code are often invisible because if they do exist, they are still in people's heads, or perhaps on scraps of paper, or in private notebooks or files.
- The project itself is frequently an amorphous collection of people, documents, and activities -- not something seen with a beginning and an end.

To make a project manageable, it needs to be visible. It needs to be divided into pieces you can get your arms around. It needs to be made modular.

One of the more successful ways of making a project modular is

by providing a structure -- a framework -- called a project life cycle, or a development cycle, in which a group of people can use their various skills and experiences to create a system to fulfill a business need.

Once your organization has agreed to use this sort of framework, then break the cycle into a series of modules. We call them phases, or stages. Use whatever number of phases you want as long as they enable you to see your project and to exert some control over it. Some organizations divide the project life cycle into four phases, some use six, some use seven, some even use twelve. The important thing is that each phase has a very clear set of objectives and definable outputs. Then each person the project manager works with can understand the planned project life cycle.

The project life cycle concept makes a project manageable in these ways:

- Managerial attention and technical efforts focus on completing a limited specified set of tasks at a time -- one phase.
- Each phase has well defined end products -- "milestones" -- that can be reviewed and evaluated by management.
- Each phase culminates in a decision to continue into the next phase -- either now or later -- or to terminate the project. Yes, terminate. That is always one of the options. Just because the organization has spent $7 million thus far on a project doesn't necessarily mean it should spend another $7 million if the situation has changed and the project product is no longer useful.
- Planning is heavily emphasized, particularly in the phases before programming begins.
- The project has a start date and an established completion date. And breaking down the cycle into manageable pieces means setting start and completion dates for each phase.

Consider that DP projects involve writing specifications, designing, coding, testing, documenting, training, and other activities directly related to building the project product. But there are other activities that are part of the process, too: planning, controlling, estimating, hiring, scheduling, assigning tasks, and so forth. They are part of the process -- the methodology, if you will, used to build the end product. They are the formal collection of explicit policies, methods, procedures, practices, and definitions that describe how something is to be accomplished. and how they interface with other groups and systems within the organization. In order to be considered a formal DP project methodology, it must be established, written, and distributed within the organization.

A growing number of organizations, both large and small, are

establishing methodologies for their DP projects. Whatever they consist of, the benefits are similar. An established methodology:

- Aids in managing and controlling projects -- by making the parts visible.
- Improves estimating by making it possible to compare similar phases in earlier projects.
- Involves users -- in a formal way.
- Improves communication by standardizing procedures and terminology.
- Involves management in major decisions by specifying approval points.

And so on. Productivity and product quality increase in varying degrees when an organization puts a formal methodology in place.

b. Establishing Phases and Milestones

Phases

Graphically the phases we mentioned earlier are displayed as vertical slices of time. This implies that at a point in time one phase ends and the other begins. Is this realistic? Even if the phases do overlap somewhat, the goal should be to start a phase only when the preceding one has been completed satisfactorily.

And how do we know when a phase has been completed? The results -- the deliverables, or the written outputs -- must be reviewed by both the project manager and the customer to see whether they are acceptable. These deliverables are agreed upon by the customer at the time the project is defined. Keeping the user involved in the project life cycle process increases the probability of the end user's acceptance of these documents as they are delivered.

What kinds of documents are we talking about? Let's take a brief look at what is expected outcome for each phase -- written, of course.

- Phase I: Identification/Definition/Objectives/Feasibility
  - A written definition of the customer's problem -- with his involvement and agreement, of course. Essentially we are defining the problem for which a solution is needed. And we're also looking at the feasibility and desirability of the project. Very little resource is to be used -- perhaps three people for five days.
- Phase II: Requirements
  - A clear, detailed, written specification of the customer's requirements. It's similar to but much more involved than Phase I. The emphasis here is on completeness and correct-

ness since they can be correct here much cheaper than if they are discovered in the design or a later stage.

- Phase III: Design
  - A design document that includes design of the system, design of the test plan, design of the conversion and installation plans, and so on. And again the customer and the project manager agree.
- Phase IV: Programming/Coding
  - Documented system programs
  - Documented test package
  - Written customer procedures
- Phase V: System Test
  - Completion and acceptance of a set of tests in a "live" environment which includes hardware, software, and procedures.
- Phase VI: Acceptance
  - Customer demonstration
  - Acceptance by operations, audit, security, and any other
- Phase VII: Installation and Operation
  - Complete all activities for an operational system, including conversion of data, programs, and customer procedures, and the training of customer personnel.

Although this shows a typical breakdown of project time among the phases, it isn't necessarily the right way for every project. Two parts of the project life cycle that so often get short-changed are the front end and the rear end. The planning done up front is often haphazard,

- "Let's start writing programs."

analysis is weak,

- "We all understand the customer's problem."

and baseline design is nonexistent. On the rear end of the project, system testing is sometimes not even included in the project plan:

- "There's no time left."
- "The programmer's integration test does the same job anyway."

Resources are committed gradually. Substantial numbers should be held to a minimum during the planning phases (prior to the beginning of coding). Only after management has evaluated the feasibility, the risk, and the value of the proposed system are substantial resources committed.

Left-to-right scheduling is a must. Occasionally it does happen that you must live with a mandated target date and schedule right-to-left, but when it isn't a legislative requirement, propose an alternative such as:

- Reduce the scope of the project
- Increase the resources
- Some combination of these two

Learn to say no in a positive way: "I can have this much ready by X date but the next part will be delivered on Y date."

Milestones

Webster defines a milestone as a "significant point in development." The key word here is significant. The project manager needs to look for points in the schedule (1) when something significant should be completed, and (2) when a decision must be made, such as a GO/NO GO decision, a decision to get more resources, or the like.

A milestone should be stated in measurable terms so the project manager knows when he gets there. Avoid such milestones as:

- Coding is 50% complete.

Does this mean lines of code? Modeules? The easy ones? Or the difficult ones?

- Module C data is entered.

Is this significant? The end of each task should not be considered a milestone. Think "significant."

Instead, consider the following as possible milestones that can be modified for your own situation:

- Problem definition is written and accepted by the customer.
- Project plan is completed.
- Design specification is written.
- Design phase review is completed.
- Customer training is completed.

Milestones are few enough but important enough that project people will put forth enough effort to meet a milestone that is in jeopardy. Too many milestones may mean too many crises. The project manager should avoid crying "Wolf!" too often. Space the milestones at least two weeks apart.

One more thing we might consider: How does the phased approach as an effective management process relate to large projects? It really is ideal for long, large projects because you are dividing a project into segments you can control. Each one can be planned and estimated. The end of each phase can be the mile-

stone where management makes the GO/NO GO decision. And remember money: The phased approach helps you track and control your spending. Problems can be identified earlier through project reviews. And change management is essential on a large, complex project.

How does the phased approach fit small projects? Actually this approach may be even more crucial with a small project. You have less time to recover from problems and more emphasis on meeting your deadlines. So the tracking, project reviews, and change management are critical. However, you may want to consider combining development phases. That includes combining documents, too, in order to save time. But don't forget that all the same major tasks still need to be done and the same information needs to be documented.

II. CONTROLLING PROJECT PERFORMANCE

a. Estimating Resources

Estimating the size of the job to be done is probably one of the toughest tasks a project manager faces. Thus far no one has succeeded in coming up with a cookbook to make estimating a mechanical process. But people still look for one -- or a course to take -- that will solve their problem. Estimating is just plain hard work. It's a complex process influenced by dozens of variables, some of which are subjective and impossible to quanitify.

However, certain principles do hold true across organizations, but they do need to be tailored to the specific organization. Estimating has to take into consideration an organization's people, procedures, and history. And as these change, so should the estimating procedures and techniques.

But first, what is an estimate? It is your best judgment of what a job will cost in terms of man-months, calendar time, computer time, and other resources. When translated into money and calendar time, and adopted for your project, the estimate becomes your budget.

Your project plan must evolve and change as conditions change and as you learn more about your project. The same is true about your estimate. If a change requires more resources, you must re-estimate and change the budget. And that is the key. An estimate will always need refining as the project moves forward. So the project manager's contract should be written to allow for re-estimating at the end of each phase. Otherwise it will never happen!

The first step toward improving estimates is to standardize the project process so we can learn from our experience. We need to keep histories on our projects. Set up an outline for a project history or borrow one from another organization. Then use the histories in planning later projects.

This is where the project life cycle comes in handy again. Adopt

one and use it for each project. And make the estimating procedures formal and documented too. Then when you list man-hours or computer time costs for module test or system test, the terms will mean the same thing on the new job you're estimating as they did on the ones for which you kept the histories. Otherwise you're comparing apples to oranges. As a rule of thumb, break down each task to no more than on man-week. Then estimate on a task-by-task basis. The total estimate becomes the sum of the task estimates. At the end of the project, feed the actual data back into the documented procedures to update the standards.

Most people now agree that although the project product may be unique, the project process is basically the same for all projects. Look for relevant past experiences.

Even with a formal methodology in place, one can overlook whole sets of activities. People usually estimate functions like design, coding, and testing, but they omit other activities such as training, planning, and documentation. And on a large project, project managers tend to overlook the effort required for interaction and communication. One way to make certain that tasks aren't overlooked is to develop your list of tasks from a list that is standard for any project. Then add to it.

The phased approach to managing projects makes it clear from the beginning that initial estimates are not very precise nor reliable. We learn as we go along, and we need the opportunity to

update our estimates when we have learned more. Remember that at all times we are looking for useful estimates rather than accurate ones.

b. Controlling Change

Project managers generally seem to agree that change is one of the major factors that can make or break a project. Change here is defined as "any event, action, or edict which may affect the scope of a project, the schedule of a project, or the resources planned for the project." The methodology in your organization spells out the kinds of changes to be controlled and the procedures for handling those changes.

We all know that change during a project is inevitable. No matter how good the methodology, and no matter how hard one tries to do everything right the first time, changes will still occur. In fact, we want change to happen if the application is to be current and relevant when it's finally installed. Some changes will be mandated -- like a nine-digit ZIP code -- or an organizational policy is changed -- like management has slashed the budget across the board. Whatever the source, change is disruptive and often costly in time, in planning, in money -- even though it may be welcome in terms of its benefits to the organization. But changes doesn't need to have an adverse effect on schedules, costs, productivity, rework, and morale. Change can be controlled.

A good methodology can reduce the impact of change in two ways:

- It reduces the number of changes that are requested by reducing the need for changes. Instead, work is done in a logical order that has been well thought out.
- It provides a procedure for evaluating proposed changes and for implementing those that are approved. Hopefully the number of nonsensical changes is reduced too.

That procedure is frequently referred to as change control. But since the word control seems to imply preventing or inhibiting change, a better term to use is change procedure, or change management. Since change can be desirable, it should be encouraged. The question is how. Many projects have turned into disasters by running over schedule and budget without notice until very late in the project life cycle. The usual reason is that the requirements changed, or the people changed, or the requirements were misunderstood which led to misinterpretation. So naturally the schedule and budget that accompanied the original plan were not changed accordingly.

Using a standard procedure for handling project changes helps too when a change involves a commitment of resources or an impact on the schedule. Now the change becomes larger than just a project change because other areas of the business are affected too.

Change can usually be initiated by anyone who sees a need. And the mechanism to use is a Project Change Request form. This acts as a control document, not a technical document. Included on the form should be such information as:

- A statement of the proposed change
- A log number
- The date the change should be made
- Management approval
- Supporting reasons for the change (justification)

Phil Braverman has categorized changes as necessary, nice to have, and nonsense. Presenting these on a Change Request form to the change evaluation/review committee for approval should reduce the number in the nonsense category. The committee then looks at the time dimension: Should the change be implemented? Now or in a later version of the system? How important is it? And what effect would it have on the work schedule and the project resources?

The project manager is usually the one who manages changes. It's a good idea to hold every member of the project team responsible for keeping alert to and feeding back any problems they feel may impact the schedule, the scope of work, or the resource requirements. It's important to create an environment in which each team member understands the reason for controlling change so he will be motivated to use the established change

management procedure. They need to be thoroughly familiar with the items being managed -- like the requirements and design documents, estimates, assumptions, and so forth -- in order to catch the subtleties of the changes.

And you need to be sure the changes get communicated to the project team too. It's easy to get the word around when there are only three people involved. Three people means three interactions. But when ten people are involved, the interactions increase to 45 using the following formula:

$$\text{Interactions} = \frac{\text{People (People - 1)}}{2}$$

The potential for not getting the word around increases significantly as the number of people increases. So prepare to deal with interactions effectively.

A procedure for managing change is important for any project, but particularly for small projects where the effects of a change can be felt more violently than on larger projects. All the same things need to be done -- or you may eventually find your small project has become a large project.

What specific benefits do we gain with a change management procedure?

- The right product is produced. How often do we see a project continued in one direction even though the need now shows a different course should be taken! How important is it to come in on time, within budget, and according to specs -- when the requirements have changed and the product doesn't meet the organization's need?
- Changes are installed efficiently with minimum disruption and cost.
- The procedure provides an audit trail.
- Product quality improves.
- Project team morale improves because of better direction and efficiency.

Bear in mind that the Chinese symbol for change is a combination of their symbols for danger and risk and for opportunity. Employ a change procedure to better understand and evaluate change and to reduce the danger and risk. Then you can capitalize on the opportunity to meet your project objectives.

III. MANAGING PEOPLE

In any organization, the key to a project manager's success is the effectiveness with which he manages his people. And an organization's most valuable asset is its people. The employees who make up your project team bring their skills, their initiative, and their willingness to work. You as the project manager must assess these and build on the strengths of your team members. The Meld Group uses the term People-Ware as an add-on to hardware and software. It's

knowing how to deal with your People-Ware effectively that makes the difference. I think most of you will agree that this isn't a skill we're born with, but rather one that we need to develop.

What can project managers do to deal with people effectively? Let's start with motivating them.

a. Motivating Project Participants

How does a project manager go about motivating his project team members? Perhaps we should first look at a definition for the word motivation.

Saul Gellerman says motivation is "Any action or event which has the effect of changing another person's behavior; something which results in another person acting in a different way than he otherwise might have." I like another definition too: Motivation is the drive from within a person. What we see then is the project manager providing an environment so that a person will respond from within.

You might ask the question "Why bother?" Just remember that all behavior is motivated, but some adds to the organization's costs. Costly behavior includes:

- Absent
- Careless
- Uncooperative
- Restricted output
- Strikes

- Sabotage
- Turnover

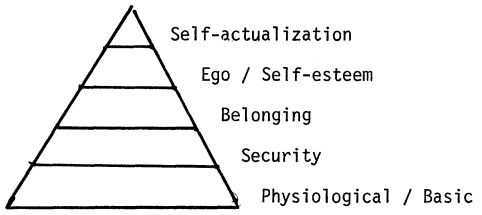
Instead, you want your project team to contribute to the organization's profits through such behavior as:

- Sustained effort
- Growth of competence
- Communicating ideas and information
- Responsibility
- Cooperative

We need to periodically ask ourselves what it costs when a person leaves our organization. Or even worse, consider the costs when a person no longer appears interested in getting involved. He is no longer motivated to perform at his best. He no longer has a loyalty to you as the project manager or to the organization as a whole. He has, in effect, taken on-the-job retirement.

Each person on your project team is motivated by a combination of factors. It's your challenge to determine what is important to each individual on your team.

Abraham Maslow, in what was probably the first general theory on human motivation, describes a hierarchy of needs:



He builds on the premise that man always wants and then wants more. As a need at one level is reasonably fulfilled, a higher level need emerges. The levels are not rigid; there can be some overlap. But once a need is basically satisfied, it no longer motivates.

What are the implications to management? Each member of your project team is striving to satisfy needs. As a manager, you control the extent to which the needs can be satisfied. You can reward a person for behavior -- which is then more apt to be repeated -- or withhold a reward. The important point here is for you to know what motivates each team member. Think about where he fits on the hierarchy of needs. Then capitalize on that need in creating an environment where he will respond with that "drive from within." Watch, too, for changes in a person's needs.

One survey which is repeated from time to time over the years asks a worker what he wants from his job. Workers responded recently with these:

1 - Interesting work
2 - Full appreciation of work done
3 - Feeling of being in on things
4 - Job security
5 - Good pay

But isn't it interesting to see that managers of these same people <u>think</u> the workers want:

1 - Good pay
2 - Job security
3 - Promotion and growth
4 - Good work conditions
5 - Interesting work

Managers in this study saw only one item related to the job itself as a motivator for his people. Notice also how our perspective can change when we become managers!

Eugene Jennings suggests we look at a person's self-image and ask how that compares with our role in life. Think about this as it relates to our work environment. The problem is one of balance.

- Does the self-image of a recent MBA graduate of a well known business school match his role as a member of your project team? Or does he see himself in your job as project manager?

- Think about the person who performed so well on your last project that your promoted him to project manager of a new project. But is he happy now -- or overwhelmed in his new role?

Look for a match between self-image and role. And consider that "Self-image may well be the most important force in the world today and is related to our role in life." (Eugene Jennings)

We've looked at the individual member of the project team -- the one we interact with on an individual basis. But how can you as a project manager build teamwork among your group! Data processing people may respect you for your technical knowledge but you'll succeed only if you can apply that knowledge in interpersonal situations.

b. Building Teamwork

A unique characteristic of people in data processing is their low social need relative to the level of social need of professionals in other fields. Cougar and Zawacki confirmed this in their recent study where they looked at people's need to interact with others. They found in looking at the DP side -- programmers, analysts, and three levels of management -- that professionals in other parts of the company looked for frequent interaction with other people -- their peers, their employees, and their managers.

Why do DP people have this low social need? One factor seems to be this: People entering the DP profession begin by learning programming. And they get unusually good feedback on how well they perform -- and from the computer, not from the instructor. The good programming student doesn't need someone else's help, and actually prefers to work alone. He is bright, logical, independent -- not necessarily antisocial, but he doesn't need much skill in verbal communication.

But what is the typical career path for a programmer? Up through programming, analysis, and up the managerial ladder. So the low social needs become a prime factor in some of the problems that many project managers face.

What are the implications to a project manager? Analysts are often grouped on teams and the Chief-Programmer-Team concept is popular. But you may want to consider some of the following as a means of dealing with the project team members:

- Continue the emphasis on the project team but limit the frequency and the duration of team meetings. (A frequently quoted guideline for structured walkthroughs is no more than one hour per session.)

- Let a team member select his own office-mate -- even though he may be assigned to work with someone else.

- Concentrate on developing your own communication and behavioral skills. You need them in order to be effective as a manager. Look for some formal education. Remember that the typical customer you need to deal with in your job has a much higher social need and tends to call more meetings, want more communication and feedback.

- Learn negotiating skills for dealing with the user manager and even the technical manager.

. Learn to recognize feelings. As a project manager, you need to develop an ability to interpret the mood of each team member. Then you can adjust what you are saying to help reach the person who is not with you.

. Assign responsibility for both the work product and the communication of it to other team members.

c. Increasing Personal Effectiveness

What we have talked about in team building also very much ties into increasing your personal effectiveness. But in addition, you might ask yourself what you can do in terms of better:

1) Managing your time. Take time to think about ways of saving time. For example:

. Look for new techniques
. Examine old habits
. Revise your goals periodically
. Schedule time off
. Concentrate on one thing at a time
. Run a time log on yourself
. Establish a "to do" list
. Build on your successes

2) Decision making. Peter Drucker suggests the following:

. Ask for opinions. They give meaning to facts.
. Build a case "against" as well as "for." Usually the

first "decision" is nothing but a kind of instinctive response to something that bothered us.
Think through why this should not be the right decision.

. Spend a good deal of time thinking through alternatives. If there is no alternative, the decision is always a bad one. One alternative to always look at is to do nothing.

. The worst decision is the right decision on the wrong problem. Dissenting opinions help you to understand what the decision is all about.

. Don't worry about who is right; think through what is right. A decision is not a popularity contest.

2) Contributing to the organization

. Focus on contribution, not on work. We in data processing are very process-oriented so we need to see action. Instead, we should be thinking about "What contribution can I make that would really make a difference to the performance and the results of this organization?"

. Communicate with those who need to know what you're doing. Let them know what you're trying to contribute.

. Ask what contribution you can make in your job as project manager that would really make a difference. It's

always something new -- and it's something that's not in your job description.

CONCLUSION

You've had an opportunity here this morning to stand back and get another perspective on your role as a project manager, both from a project and from a management viewpoint. There is much written on any one of these topics so we're really just skimmed the surface. There are no doubt many other ways you can operate and be effective. But I've tried to mention here some things that have been successful for a large number of project managers in a large number of organizations. If you find you're already doing many of these things, then you may feel good in knowing others are finding success in managing their projects the same way.

Hopefully, too, you've gotten an idea or two about something you would like to try doing differently.

I wish you success as a project manager, and would love to hear from you about your problems or successes.

BIBLIOGRAPHY

Burrill, Claude W. and Leon W. Ellsworth. Modern Project Management: Foundations for Quality and Productivity. Tenafly, NJ: Burrill-Ellsworth Associates, Inc., 1980.

Cougar, J. and Robert A. Zawacki. Motivating and Managing Computer Personnel. New York: John Wiley and Sons, 1980.

Drucker, Peter F. The Effective Executive. New York: Harper and Row, 1967.

Drucker, Peter F. The Practice of Management. New York: Harper and Row, 1954.

Dyer, William G. Team Building: Issues and Alternatives. Reading, MA: Addison-Wesley Publishing Company, 1977.

Gordon, R. L. and J. C. Lamb. "A Close Look at Brooks' Law," Datamation, 1976.

"How to Make Groups Work for You." Newington, CT: The Meld Publishing Company, 1983.

Metzger, Philip W. Managing a Programming Project, 2nd edition. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1981.

Winters, Raymond J. It's Different When You Manage. Lexington, MA: Lexington Books, 1983.