Introduction of the project management discipline in a software development organization

by T. Raz

An approach to the introduction of the project management discipline in a software development organization is presented, with emphasis on the aspects that can be generalized and adopted by other organizations under similar circumstances. The presentation includes the key elements of the approach taken, the maturity scale used to guide the introduction effort, a short description of the education program developed specifically for this case, a methodology for developing project models, and the staff and support structure put in place. The paper concludes by reporting the initial experience and noting directions for future development.

Managing a project requires specific skills and techniques to deal with problems related to the following areas: planning and scheduling tasks and activities; recruiting, motivating, and developing project staff members; assigning tasks to individuals and appraising their performance; negotiating with groups internal and external to the organization; assuring quality; dealing with risk and uncertainty; and other human relations and business subjects. This body of knowledge, which is distinct from the technical expertise required to actually execute the project, constitutes the basis for the project management profession.

However, the term "project management" has a more specific and limited meaning, restricted to the planning and control aspects of managing projects. In fact, the *Dictionary of Computing*[1] provides the following definition for project management and two related terms:

> *project management:* "[In system development, t]he activities concerned with project planning and project control."
> *project planning:* "The activities concerned with the specification of the components, timing, resources, and procedures of a project."
> *project control:* "The activities concerned with monitoring the progress of a project, its direction, quality, and resource utilization, as compared with project plans."

The discipline of project management has been around for several decades. Its formal beginnings are traced back to the precedence network diagramming techniques developed for the Polaris submarine project in the early 1960s. Today, virtually all construction and logistics projects and a good portion of research, development, and engineering projects in most areas use formal project management techniques. In fact, academic-level coursework on project management is part of most, if not all, college curricula in civil, chemical, and industrial engineering, as well as operations management and business administration.

The need for a disciplined approach to managing projects is particularly marked in the software development industry in light of some of its characteristics, such as the tremendous rate of technological change that induces risk and uncertainty; scarcity of productive resources, mainly skilled labor; intense competition in the marketplace; high rate of product obsolescence; and frequent requirements changes.

The situation, however, is less than desirable due to a combination of factors, starting with the lack of proven techniques for activity analysis and for duration and resource estimation, and ending with the notion that software development is an art and as such is not subject to rigorous engineering methodologies. However, a consensus has emerged in the last few years, according to which software development efforts must incorporate project management techniques to maintain reasonable levels of quality and productivity.[2-6] This point is clearly made by Humphrey[7] who explicitly mentions project management as a key requirement for having a software development organization emerge from its initial (chaotic) stage and move up the maturity scale adopted by the Software Engineering Institute. In addition, studies of the software development process, such as those of Radice et al.[8,9] and Hoffnagle and Beregi,[10] clearly suggest the need for formal procedures for planning and tracking the various activities that constitute the process and for controlling its execution.

The purpose of this paper is to give an example of a successful effort to introduce the project management discipline to a medium-sized software development organization. The presentation highlights those aspects that can be generalized and adopted by other organizations under similar circumstances.

Background information

The subject organization is a medium-sized IBM software development laboratory, which had been in existence in its current form for about three years at the time the implementation effort began. The mission of the organization was to execute software development projects. At any point in time, about six major projects, each involving more than 50 professionals for 18 months or longer, about ten medium-sized projects, each involving 15–50 professionals for 12–18 months, and an undetermined number of minor projects, involving fewer than 15 professionals for periods shorter than 12 months, were active.

A substantial percentage of the software development professionals either had been with the company for less than three years or had transferred from other areas, mainly customer support. Thus, the amount of collective expertise and experience in software development in general and in software engineering methods and techniques in particular was quite limited.

Although subject to frequent changes, the organizational structure could be described as a hybrid between a project-oriented structure and a matrix structure. Development groups were organized by project, with project managers reporting to product managers, who in turn reported to the laboratory director. Each product manager was responsible for developing software products for a specific combination of operating systems and hardware. Development support groups, such as information development, build and integration, system test, and performance test, were organized in a roughly parallel structure through a different management chain that also reported to the laboratory director. Finally, certain support groups, such as usability testing, business planning, and early customer support, were divided among several separate management chains also reporting to the director of the laboratory.

Thus, the manager of a typical project had direct control over the development staff, and sometimes also over the planning and design groups, but had to go outside the product organization to a support organization to obtain build, integration, information development, and national language support services, and to yet other organizations for usability testing and business and financial planning assistance.

The situation is further complicated by the fact that some of the development groups are in remote locations across the United States or even in Europe or Asia, and some of the development work is done by other companies working under contract with IBM.

This rather complex organizational structure generated considerable planning, coordination, and control problems that adversely affected the ability to meet schedule and resource commitments and to respond to market-driven changes.

It was against this background that several organizational assessments, both internal and external, were conducted. These assessments unequivocally identified a need for more rigorous practices in managing projects throughout the organization. Concurrent with the assessment activity, a major program was launched to promote grassroots involvement in quality improvement by forming quality action teams. Several of these teams, which were mainly composed of technical staff rather than management, articulated the need for better methodologies and tools to support project management. Building on the support from both the management and technical communities, an internal consulting group took it upon itself to introduce the project management discipline to the organization.

Staffing

The availability of qualified support personnel is a key factor in the introduction of any new methodology or technology. Accordingly, management funded four full-time positions to enable the introduction of project management. These positions were:

- Program manager—responsible for overall coordination, compilation of standards, and enlisting the cooperation of the various groups targeted for the introduction. This position was filled with an individual with extensive experience in software development and management, including several years of project management in a development environment.
- Technical consultant—responsible for developing project management methods and procedures to meet local needs and providing education and consultation services to the user community. This position was filled with an individual with an extensive background and academic education in industrial engineering and operations management, including project management, and with teaching and consulting experience.
- Tool support—responsible for ensuring the installation and maintenance of the computer-based tool and developing and supporting computerized procedures to meet local needs. Two positions were required and were filled with experienced systems programmers with backgrounds in tool development.

Approach used

The introduction of new technologies and methodologies almost invariably requires the potential users to change the way in which they think and act. Understandably, such changes in a work culture are quite often met with suspicion and some measure of apprehension, sometimes bordering on hostility. Besides, any changes are bound to cause a certain amount of disruption and confusion in the operation of the target organization and entail the risk of misuse and possible rejection of the proposed innovation. Accordingly, to reduce the likelihood and extent of adverse effects, a strategy was carefully formulated at the beginning of the introduction effort and adhered to throughout the effort. The introduction strategy was based on several key premises, which are described next in detail.

Common tool. The need for a common tool to support project management in all areas of the organization became apparent at the beginning of the planning for the introduction of the discipline. By having all project groups use the same tool, better communication across groups and reduced training costs would be realized. The tool selected was the IBM Application System (AS), program number 5688-108, available at the time in Version 2 Release 2. Although it requires several days of training, AS provides many advantages that overall make it an alternative superior to any other project management software tool. These advantages are: a virtually unlimited number of activities, relationships, and resources; superior modeling capabilities (four types of relationships, multiple calendars, four types of required dates, hammocks and milestones, multiple projects, segmentation, priority scheduling, resource scheduling based on fixed time or fixed resources, etc.); outstanding flexibility in report and chart generation; easy integration with other major IBM products, such as Structured Query Language (SQL) and the REXX programming language, and with other AS features; compatibility with AD/Cycle*, the strategic IBM application development environment;

extensive installed base and user community within the corporation; and well-developed formal and informal user support mechanisms.

Project orientation. The complexity of the organization suggested that the only way in which to have complete and accurate project data was to have all of the groups in the organization, including the various support groups, apply the discipline and the tool consistently. Rather than approaching it all at once, and having to bridge between the different needs of development groups that function like project organizations and the development support groups that function to some extent like job shops, the focus was primarily directed at those groups that had a specific project mission. The rationale was that these groups would benefit most from the project management discipline. Further, it was (correctly) anticipated that the project groups would exert pressure on the support groups to provide them with accurate and compatible dates, durations, and other project data, and that this pressure would encourage the support groups to eventually develop their own project management models.

Management orientation. Project management was presented as a discipline that allows managers to better plan and control their projects. The focus was on the decision-making needs of project managers, rather than on routine tracking and reporting. This focus was achieved by emphasizing the flexibility of the discipline and its supporting tool, and their capability to support the specific needs of managers at various levels in the project organization, along with de-emphasizing the clerical aspect of periodic data reporting in predefined formats. Accordingly, only managers and technical professionals were invited to participate in the various stages of the introduction; the involvement of administrative and nonprofessional staff was discouraged.

Maximum flexibility. One possible approach was to form a central department to gather, process, and distribute all the data related to project management and to try to enforce standard procedures and formats throughout the organization. Instead, we chose to grant to the user groups the maximum freedom possible in defining the types of data they would gather, in defining the file structures, coding conventions, etc., that they would use, and in defining the types, formats, and frequencies of the various reports and charts that would be generated. This approach was inspired by the current philosophy of empowering employees to manage and execute their work without unnecessary external interference. This approach in effect allowed project teams to utilize project management to the best of their own specific needs. We anticipated that it would increase significantly the likelihood of having the resulting plans accepted by those who were expected to execute them, since they were involved in their definition and development. It was left up to the core support group to formulate guidelines and to create standards and common procedures that would be flexible enough to accommodate the needs of all of the users.

Staged introduction. The route from a lack of formal project management discipline to its complete and correct use was divided into several distinct sequential stages, and the various groups in the organizations were guided and assisted in their progress through the steps. A report card for the main project areas was compiled and distributed periodically, allowing the various groups to monitor their progress, compare themselves with one other, and identify inhibiting factors. The following five stages were summarized according to a maturity scale:

1. Education: Managers and technical leaders of the development and development support groups attend an education program devoted to the discipline of project management and to the parts of AS that support it. This three-day program is described in detail in the next section.
2. Model development: Following a methodology presented in the education program, a designated individual in each group coordinates the compilation of the data needed to create a project model in AS (activities, relationships, calendars). The data are entered in AS, and basic charts and reports are produced.
3. Initial usage (production): The model created in the previous stage is used to generate periodic reports presented at project status meetings. During this stage substantial revisions to the model are expected as the users find by experimentation the appropriate level of detail and scope of coverage of their models.
4. Resource analysis: The model that emerged from several weeks of initial usage is augmented by estimates of the resource requirements of each activity for each type of re-

source that the project manager needs to control. This analysis may require other methodologies and tools for sizing and estimation. The resulting estimates are entered in AS, along with the planned resource availability over time by resource type. The AS model is now capable of generating schedules while balancing the requirements for resources against their availability, and of producing charts and reports that describe resource utilization patterns over time.

5. Integration across project models (roll up): Data, mainly dates (actual, planned, committed), are shared across the individual project models developed and used by the various groups within a project. There are two basic modes of integration across models:

    a. Roll up of activity dates: Selected activities along with their dates and other pertinent attributes are extracted from the individual models and combined into a single model that provides an aggregate view to higher management. In this mode, the combined model does not include relationships or resources, and consequently no rescheduling at that level is possible. Rolling up of dates serves to monitor the combined status of several groups rather than being used for aggregate planning.
    b. Internetworking: Dependencies between activities in different models are added to the project models of the various groups, and procedures are implemented to ensure that changes in the dates or duration of an activity are communicated to the activities that depend on it and reside in separate models.

Figure 1 presents an example of the report which is a chart showing the progress of the eight major project areas along the five-step maturity scale. This chart was updated monthly and distributed to management.

Education program

Education is the first and possibly the most critical stage in progressing through the project management maturity scale. Consequently, a significant amount of effort was invested in planning and preparing the education program. The following were key considerations in its design: The education should be aimed at allowing project team members to progress to the next stage in the maturity scale (model development) rather than providing coverage of all aspects of managing projects. The duration of the program should be as short as possible to allow participation without disrupting the actual work on the projects. The classes should be taught by local staff, who would remain available on site to provide continuing support and follow-up. All individuals who may have to contribute data to the development of the project model should be invited to attend the education program. The class exercises and interaction should be designed to promote cooperation and team spirit within project members. The managers of the various groups and teams should attend the education program and become involved in the exercises along with their lead technical staff members.

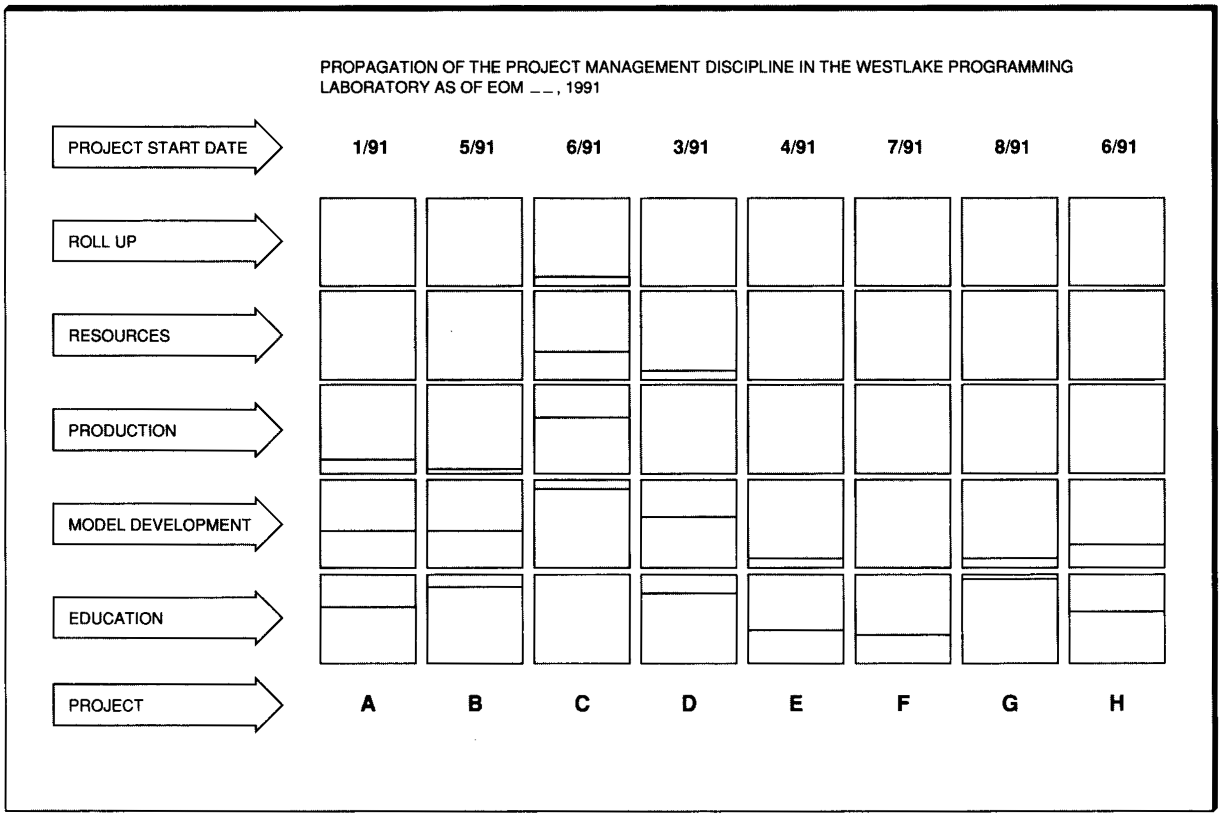
Class materials from various project management courses were examined, and eventually an education program consisting of two classes taught consecutively to groups of 20 to 30 project team members was offered on a monthly basis. The contents of the two courses are described next.

Project management theory. This is a two-day basic course on the planning and control aspects of the project management discipline. Lecture and exercise sessions alternate throughout the course, with students working on the exercises in groups of five or six and presenting their results to the rest of the class.

The first day is devoted to the terminology of project management (activities and their attributes, different types of relationships, dates and their various types, calendars, different types of resources) and to the algorithms used to calculate dates (early, late, and float) based on durations, relationships, and resource requirements. This part includes three exercises based on data given to the students: network construction, time analysis, and resource analysis.

The second day is devoted to a methodology for developing project models and to the use of such models for decision support in project planning and control. The seven-step methodology is described in detail in the next section. Four exercise sessions follow the steps of the methodology. Each group of students selects the project that will be the subject of the exercises, which should be related to the students' current job responsi-

Figure 1 Propagation of project management along the maturity scale



bilities. In this way, the results of the exercises become a starting point for the actual model that the students need to develop.

Project management tool. This one-day class provides a hands-on guided tour of the facilities of AS that support project management. Students work individually or in pairs in front of terminals and execute a series of exercises on AS. The exercises cover model definition, data entry, network construction, time analysis, resource analysis, report generation, chart generation, and the procedures developed specifically for local use.

Development of the project model

A model is an abstraction of reality created to better understand reality. In the context of project management, reality is the actual project, which has not yet completed, or perhaps has not even started. The model consists of the collection of data that is used to develop schedules for the activities in the project. This collection includes data about activities, relationships, dates, calendars, resources, product and process structures, and so on. In addition, the model includes the procedures used to process these data to support project management decisions. The methodology consists of the following seven steps:

1. Product structure
2. Activities
3. Relationships
4. External dependencies
5. Process structure
6. Resources and calendars definition
7. Resource requirements

The completeness and validity of the project model needs to be verified by the members of the project team and representatives of the various organizations with which the project has external dependencies. Verification, which consists of examining the data in the model and checking the accuracy and consistency of the data, should be done concurrently with the development of the model, after major changes are introduced, and on an ongoing basis at regular periods of time throughout the life of the project.

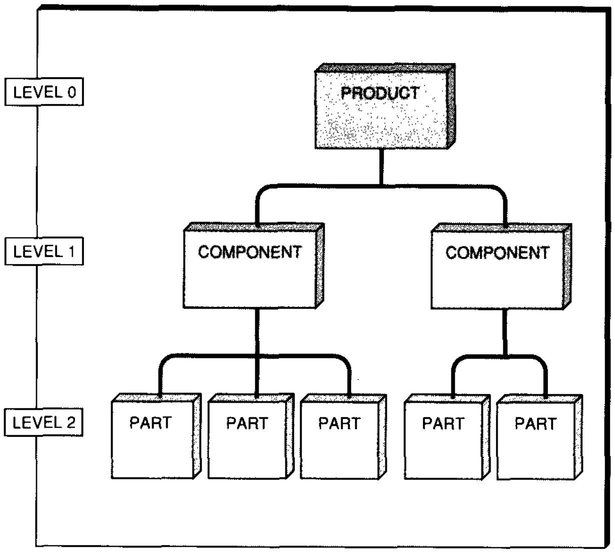
Product structure. The product structure is obtained by repeatedly decomposing the product into its major components, assemblies, subassemblies and basic parts. Each step in the decomposition yields an exhaustive set of mutually exclusive elements so that their aggregate is equivalent to the original element being decomposed. Each of the elements in the product structure is a product by itself, meaning that it is the tangible result of a project or process. The product structure is represented as a tree diagram with the root node representing the product as a whole. The nodes represent the elements of the product and are arranged in levels, beginning with the highest (level 0) for the root. The aggregate of all of the nodes at a given level is equivalent to the entire product. Edges are drawn between two elements in two successive levels and show that the element in the lower level is part of the element in the higher level. A generic example of a product structure is shown in Figure 2.

Usually it is preferable to limit the number of product elements in the product structure tree. The decomposition should be carried out until each bottom-level element has internal unity and cohesiveness of:

- Ownership—A single individual is responsible for the production of the element.
- Time—The activities required to produce the product element are normally executed continuously in time.
- Resource utilization—The activities required to produce the product element normally use resources at a constant rate.

Activities. This step consists of determining the activities that are required to design and produce all the elements that constitute the product. The analysis is guided by the product structure generated in the previous step and typically consists of two passes: a top-down pass for the design activities and a bottom-up pass for the production and integration activities.

Figure 2 Generic product structure

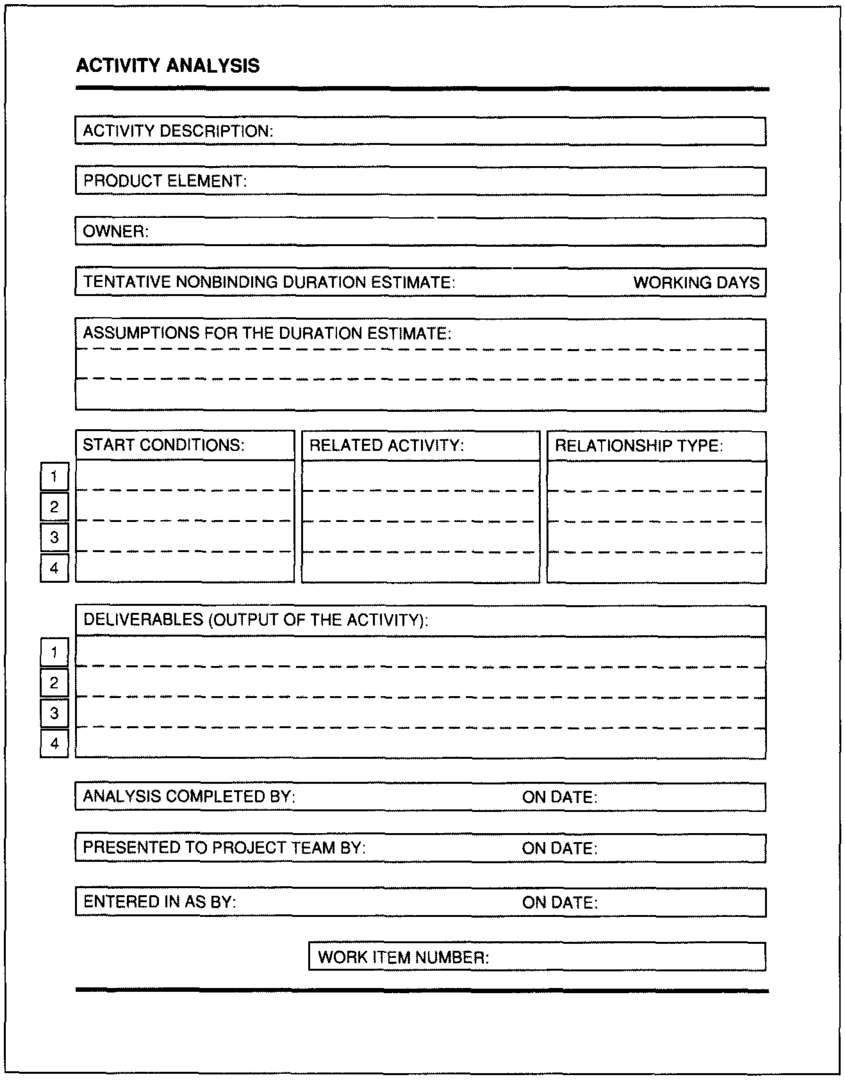


Starting at the highest level and proceeding down through the product structure, we determine the activities required to design each product element. Some examples of design activities are: writing the business plan and gathering requirements at the product level, high-level design at the component level, low-level design at the module level.

Once all the design activities down to the lowest product elements have been identified, we determine the activities required to produce the elements at the bottom of the product structure. Next, the activities required to assemble or integrate the product elements at the second lowest level in the product structure from their respective constituent elements are determined. This is repeated for all levels of the product structure until all of the activities required at all levels are listed. Some examples of production or integration activities are: code and unit test at the module level, build and component test at the component level, and system test and publications at the product level.

Identifiers and descriptors are assigned to each activity, durations are estimated, and entry and

Figure 3 Activity analysis form



ACTIVITY ANALYSIS

ACTIVITY DESCRIPTION:

PRODUCT ELEMENT:

OWNER:

TENTATIVE NONBINDING DURATION ESTIMATE: WORKING DAYS

ASSUMPTIONS FOR THE DURATION ESTIMATE:

	START CONDITIONS:	RELATED ACTIVITY:	RELATIONSHIP TYPE:
1			
2			
3			
4			

DELIVERABLES (OUTPUT OF THE ACTIVITY):

1

2

3

4

ANALYSIS COMPLETED BY: ON DATE:

PRESENTED TO PROJECT TEAM BY: ON DATE:

ENTERED IN AS BY: ON DATE:

WORK ITEM NUMBER:

exit criteria are defined based on the nature of the activities and their inputs and outputs. Finally, an owner is assigned to each activity. An example of a form designed to capture information about project activities appears in Figure 3.

Obviously, any process can be decomposed into activities in many different ways with varying degrees of detail. Greater detail provides more precise control but requires more effort in model development and maintenance and may require more management involvement. It is up to the developers and users of the project model to find the appropriate balance point between detail and effort. It is a good rule of thumb to define as few activities as possible, provided that each activity fulfills the same criteria used for bottom-level product elements: single owner, normally executed continuously, and constant resource utilization rate.

Relationship analysis. A relationship is a time dependency between two activities. Relationships are identified by examining the entry and exit criteria of each activity and determining prerequisites and corequisites that appear as entry or exit criteria in other activities in the project. The lags of the relationships (required amount of elapsed time between the two activities) should reflect only technological or administrative constraints rather than effort expenditure or availability of resources.

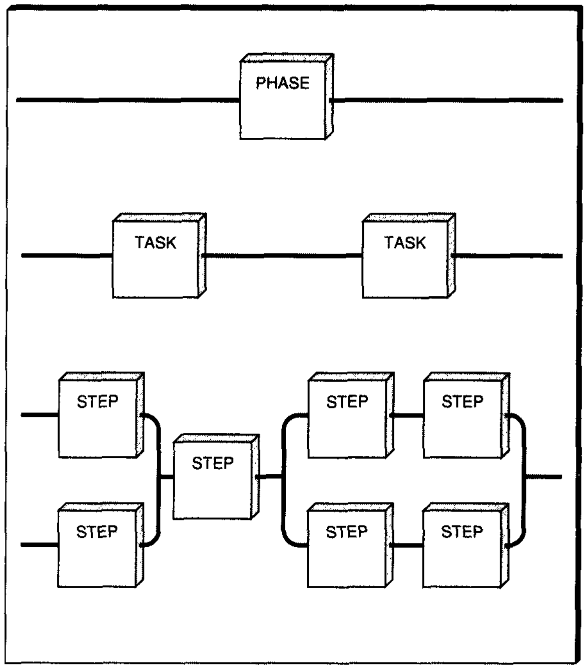
In fact, no resources at all should be consumed by any lag period. For example, a time dependency between the completion of an inspection and the start of a review meeting may include a lag for the distribution of the inspection report. That lag should reflect the amount of time required to deliver mail to the participants of the review meeting but should not include the time required to prepare the inspection report. That effort should either be included as part of the inspection activity or defined as a separate activity.

External dependencies. External dependencies are dependencies on activities that are not included in the project model, mainly because they are outside the control of the project manager. In fact, there is no point in burdening the model by including activities that are owned by individuals or organizations outside the authority of the project manager or that have schedules or resource utilization patterns that the project manager cannot affect. External dependencies may involve deliverables from project activities to external activities, or receivables from external activities to project activities. These dependencies are accounted for by defining checkpoint activities that represent the transmittal of deliverables and receivables, and by relationships between these checkpoints and project activities.

Required dates are imposed on the project schedule on the basis of considerations external to the time dependencies and resource utilization patterns of the project activities. They may be mandatory, or they may be upper or lower limits on either start or finish dates, and they should be taken into account by the scheduling algorithm. Since required dates come from external sources, they should be identified and recorded along with the external dependencies.

Process structure. At this stage in the development of the model it becomes significant to define the process structure for the entire project. The process structure provides several levels of aggregation that allow analysis and output generation to be done at various levels of detail to support different decision-making needs. The activities identified so far correspond to the basic steps of the process. Steps are grouped into tasks, and tasks into phases, with the possibility of defining other levels of aggregation as needed. The process structure may be developed with some accepted process model as the base, such as the IBM Checkpoint Process or IBM Programming Process Architecture, or on some common attribute or combination of attributes, such as owner, type of input, resources used, time proximity, location, and so on. Figure 4 shows a generic process structure. It is important to note that tasks and phases are defined as hammock activities, so that their durations are calculated as the sum of the durations of their constituent activities.

Figure 4 Generic process structure



Resources and calendars definition. At this point it is appropriate to identify the resources that the project manager may want to include in the project model, along with their respective calendars. Only resources over which the project manager has scheduling authority or utilization accountability should be included. These resources usually

Table 1 Change in level of detail of the project plan over time

Plan Version	First Quarter	Second Quarter	Third Quarter	Fourth Quarter
Initial plan at beginning of project	High	Medium	Low	Low
Revision at end of first quarter	History	High	Medium	Low
Revision at end of second quarter	History	History	High	Medium
Revision at end of third quarter	History	History	History	High

Note:

- High detail: most of the activity durations are less than 2 weeks.
- Medium detail: most of the activity durations are between 4 and 6 weeks.
- Low detail: most of the activity durations are more than 6 weeks.
- History: part of the plan that has already been executed.

include personnel, major equipment and funds, as well as other resources for which the cost and criticality justify the added model maintenance effort.

Resource requirements. Resource requirements constitute the last major set of data needed for the model. This step involves estimating the amounts of each type of resource required to complete each activity in the project plan within its assigned duration. The accurate estimation of resource requirements demands substantial experience in similar projects, along with the ability to project the impact that new technologies may have on the planned activities.

Incremental planning. Planning is a continuous process that needs to be done throughout the life of the project. As the project unfolds and remote activities become closer in time, more information becomes available, and additional details may be added to the model. Thus, it is advisable to postpone the detailed analysis of future activities and to focus on the detail of those activities that are likely to be scheduled in the near future. In this manner we avoid spending too much effort planning future events that may change significantly, and we reduce the initial complexity of the model. This approach applies to both activity and resource data: Using gross estimates for the durations and resource requirements of activities in the distant future and refining them as the activities become closer in time simplifies the maintenance of the project model and improves its ease of use. The approach is illustrated in Table 1, which shows the level of plan detail as the project unfolds. One should note that at all times the plan covers the entire project, and nothing is left out. What varies is the level of detail, which matches the need for management control.

Support structure

A major component of the introduction effort consisted of establishing a support structure to allow the user population to effectively use the project management discipline and its supporting tool, and to enable continuous growth and propagation. Besides the core staff groups previously mentioned (eventually augmented by two individuals on part-time assignment), the support structure included a project management council and an electronic questions and answers database. These two support mechanisms are described next.

Project management council. The project management council was established to promote two-way communication between the core support staff (program manager, technical consultant, and tool support) and the user community. The council evolved from a quality implementation team that played an advisory role during the early phase of the introduction. The council is chaired by the program manager, and its members represent the various projects in the organization. The representatives are typically individuals who were designated to coordinate the project management activities in their respective areas or functions. Typical agenda items for the council, which meets biweekly, include: availability of enhancements to the AS tool, such as procedures or add-on products; publication of model development standards (file structures, labeling schemes, etc.); announcements of education offerings; and requests for additional support from the user community.

Questions and answers database. An electronic questions and answers library (EQUAL) was installed on the various computer systems that are accessed by the project management users. The objective was to ease the collection, processing, and dissemination of requests for information about the tool. Questions are submitted electronically by the users from their terminals to one of the tool support staff. Based on the subject, the question is assigned to the appropriate core staff member, who does the research, prepares the answer, and sends it to the submitter. Also, if the subject appears to be of wide interest, the respondent has the option of adding the question and its answer to a repository. With the same EQUAL tool, users can search the repository before submitting their questions to see if a similar or equal question has already been submitted and answered. In such cases, the answer in the library becomes available to the user, eliminating the submit and respond steps of the cycle. The EQUAL tool uses a sophisticated search algorithm that scans the text of all questions and answers on the basis of partial selection criteria. Some of the typical subjects of questions were related to printing AS output on the mainframe; exporting and importing project models; formatting charts and reports; and similar subjects that cannot always be covered in the education program and surface only when the tool is actually used.

Evaluation

Within one year of launching the introduction effort, about one third of the development staff and managers had participated in the education program, and all of the major project areas in the organization had full-time personnel responsible for project management. About one third of the departments and groups throughout the organization were already using or developing project models following the methodology described earlier, and most of the remaining groups had plans to this effect. The project management council was meeting regularly, and various follow-on activities, such as advanced classes, a symposium, and publication of technical reports, were under way.

These results suggest that project management had been accepted as an integral component of software development and, consequently, that the introduction effort had been successful. Prior to this effort, there were at least two attempts to introduce project management discipline supported by AS to the same organization. Both failed and were abandoned after several months of effort. Thus, it is appropriate to try to evaluate the effectiveness of the approach taken here in order to learn which of the factors that contributed to its success can be generalized and applied to similar situations.

Ideally, to evaluate a program such as the one described here, we would like to have two situations that are very similar in terms of organizational structure, personnel experience and background, project mix, etc. We would apply the approach under evaluation to one situation and leave the second situation as it is as a control, while assuring that the two stay mutually isolated. At the end of a significant period of time we would compare the differences along some predefined, measurable, and significant criteria, and draw our conclusions.

Clearly, this is not feasible in cases that involve changes in the culture and mode of operation of an organization. However, rather than giving up completely on the notion of evaluation, we will provide a qualitative assessment based on informal opinions of the members of the organization and on our own experience. These are divided into two sections: The first section presents the perceived benefits of the project management discipline as that discipline was presented and introduced to the organization. The second section discusses some of the key factors that contributed to the success of the introduction effort.

Benefits of project management. At the beginning of the introduction effort some developers and managers expressed the concern that a disciplined approach to project management could hinder the creativity and ingenuity of the designers and programmers. They were also concerned that the need to report progress in order to keep the model up to date may adversely affect their productivity. Besides, there were doubts about the ability of the organization to adapt to and comply with a more structured way of planning and controlling work. A year later, these concerns disappeared completely, and project management is perceived as part of the toolkit that developers use in their work.

None of the groups and project teams abandoned project management and reverted to the previous way of planning and controlling project work, although the rates of assimilation vary among groups and over time. Periodically managers and project management staff are invited to share their experiences. Some of the benefits derived from the use of project management that they mention include:

1. Better understanding of the relationships and dependencies among the various groups that contribute to the project, and between the project team and external organizations
2. Better assessment of the time and resources required to achieve the project objectives
3. A logical and consistent way to calculate dates and schedules based on the analysis of dependencies, durations, and resource requirements
4. An objective basis for evaluating the impact of changes on the original schedule and for selecting among alternative courses of action in response to deviations from the plan
5. A systematic process for tracking actual progress against the plan and for identifying potential problems before they evolve into critical situations

Although it is difficult to quantify the value of these benefits, the following example provides some indication of their magnitude. The team of a medium-sized project decided to implement the project management methodology in midstream. Once the project model was created in AS, the analysis showed that any feasible completion date would be too late to achieve the market penetration that will generate sufficient revenue to justify the cost of the project. Consequently, the project was terminated, and its team reassigned to other projects, resulting in savings of several millions of dollars. It is possible that the same decision would have been made without formal project management methods, but it is not likely that it would have been made in the same timely and objective manner.

Success factors. Two basic prerequisites to the success of an attempt to introduce new methods and to change the operational paradigm of an organization are: a perceived need for change among the target population, and firm and visible commitment from all levels of management. In the case described here, these two prerequisites existed to a fairly large degree and permitted achieving the results noted in a short period of time. As the effort unfolded, several other factors emerged as key contributors to its success. They are:

1. Adequate size and composition of the core support team. The responsibilities of this team were to gain the cooperation of all the parties who had to apply the project management discipline, to educate them in the theory of project management, and to support the tool. Consequently, it was critical that the team members be not only technically competent but also that their personal communications skills with all levels of personnel be very good and that they would be capable of earning the attention and respect of their peers and their managers. These objectives were achieved by selecting senior, highly experienced individuals with strong track records and with education and leadership credentials. In addition, by allowing these individuals to devote their full time and attention to the task at hand, they become capable of responding in a very timely manner to requests for assistance and to any problems or issues as they surface.
2. Appropriate focus of the education program. Most project management education programs offered by training organizations are generic in nature and cover a wide array of topics. In contrast, the education program described earlier in this paper was designed for the specific objective of teaching a fairly homogenous population how to develop, build, and use a project model in AS for planning and controlling their own projects. This design reduced the length of the classes, minimizing the disruption of ongoing work. It also allowed the use of the local jargon and of local examples, all of which enhanced the communications between the instructors and the students. The exercises on the second day of the theory class were based on the students' own current projects, which gave them a head start on the development of their models once they returned to their project teams.
3. Comprehensive support structure. Past efforts at introducing project management were limited to bringing in consultants for training and making motivational speeches, leaving the implementers on their own. In our case, the promoters of the new discipline were also the instructors and served as consultants after the training sessions were over. Further, they were part of the same organization that was expected to apply project management, so that the commitment and ownership were evident. In addition to the availability of the instructors as consultants, the fact that the project man-

agement council was chaired by the program manager, who was one of the instructors, contributed to the continuity of the support offered to the various groups that were implementing project management.

Concluding remarks

Any change in the culture and the operations of an organization entails some degree of risk. If the change is successfully introduced, the organization will reach new levels of productivity. If the introduction meets with resistance and mistrust, not only are productivity gains forfeited and the resources allocated to the introduction effort wasted, but the likelihood of success of a later attempt at change is jeopardized. In this paper we described in some detail an effort undertaken to introduce such a change in the area of project management and discussed the factors that contributed to its success. More work lies ahead of the software industry before it reaches the extent of project management typical of other, more mature industries. We believe that the essence of the experience reported here can be adapted to and applied by other software development organizations intent on improving the quality and integrity of their project plans and schedules.

Acknowledgment

The work described here is the result of the combined efforts of many individuals in the Westlake Programming Laboratory and in the IBM Corporation. This includes the members of the project management Core Support Team (Kimberly Bauer, Ralph Gomez, Bob Marshall, Byron McLaughlin, Paul Nanson, Alan Yaung); members of the original Quality Implementation Team (Roger Miller, Wanda Sarti, Susan Tanner, Peggy Whitham); and the two managers directly involved in this effort, Wayne Gerlach and Gaila McLaughlin. In addition, the contributions made by the many project management users are also acknowledged.

*Trademark or registered trademark of International Business Machines Corporation.

Cited references

1. *Dictionary of Computing*, SC20-1699-08, IBM Corporation (1991); available through IBM branch offices.
2. J. Blaney, "Managing Software Development Projects," *Project Management Institute Symposium*, Atlanta, GA (October 1989), pp. 410–417.
3. E. W. Larson and D. H. Gobeli, "Significance of Project Management Structure on Development Success," *IEEE Transactions on Engineering Management* **36**, No. 2, 119–125 (May 1989).
4. W. H. Roetzheim, *Structured Computer Project Management,* Prentice-Hall, Inc., Englewood Cliffs, NJ (1988).
5. R. B. Rowen, "Software Project Management Under Incomplete and Ambiguous Specifications," *IEEE Transactions on Engineering Management* **37**, No. 1, 10–21 (February 1990).
6. N. Whitten, *Managing Software Development Projects*, John Wiley & Sons, Inc., New York (1990).
7. W. S. Humphrey, "Characterizing the Software Process: A Maturity Framework," *IEEE Software* **5**, No. 2, 73–79 (March 1989).
8. R. A. Radice, N. K. Roth, A. C. O'Hara, and W. A. Ciarfella, "A Programming Process Architecture," *IBM Systems Journal* **24**, No. 2, 79–90 (1985).
9. R. A. Radice, J. T. Harding, P. E. Munnis, and R. W. Phillips, "A Programming Process Study," *IBM Systems Journal* **24**, No. 2, 91–101 (1985).
10. G. F. Hoffnagle and W. E. Beregi, "Automating the Software Development Process," *IBM Systems Journal* **24**, No. 2, 101–120 (1985).

Accepted for publication October 1, 1992.

Tzvi Raz *IBM Programming Systems, Westlake Programming Laboratory, 5 West Kirkwood Boulevard, Roanoke, Texas 76299.* Dr. Raz received a B.Sc. degree from the Technion, Israel, in 1977, an M.A.Sc. from the University of Toronto in 1979, and a Ph.D. degree from the University of Missouri in 1982, all in industrial and management engineering. He worked as an analyst in a military computing center from 1979 to 1980 and as a management engineer in a Veterans Administration hospital from 1981 to 1982. He was an Associate Professor of Industrial Engineering at the University of Iowa from 1982 to 1989 and at the University of Arizona in 1988, and a Senior Lecturer at Ben Gurion University in Israel in 1989 and 1990. He taught and conducted research in the areas of computer and information systems, quality planning and control, production and operations management, and quantitative methods. Dr. Raz has authored or coauthored over 30 technical papers published in refereed journals, edited two books, and written a number of book chapters, conference papers, and technical publications of various types. He joined IBM in 1990 as an Advisory Development Analyst with the Westlake Programming Laboratory, working as a technical consultant on project management.

Reprint Order No. G321-5513.