EFFECTIVE MANAGEMENT OF SOFTWARE DEVELOPMENT

AT&T TECHNICAL JOURNAL

Lenard B. Robertson and Glenn A. Secor

Lenard B. Robertson *and* ***Glenn A. Secor*** *are supervisors in the Software Methodology and Tools Department at AT&T Bell Laboratories in Liberty Corner, New Jersey, where they are responsible for the Software Project Management Workshop. Mr. Robertson joined the company in 1982. He has a B.A. degree in business administration from Upper Iowa University, an M.B.A. in management and finance from Southern Illinois University, and a Ph.D. in management from American Western University. Mr. Secor joined Bell Laboratories in 1970. His degrees, B.S., M.S., and Ph.D., are all in engineering science, from the University of California.*

The goal of most projects is to produce a product that meets market and customer needs, on a schedule responsive to the market, at an appropriate cost, and with appropriate quality. Attainment of that goal depends on how effectively the project is managed. This article presents an effective management process that is designed for software projects but is applicable to other types of projects as well. The article discusses four principles of effective management: planning—detailing every phase of what is to be done, who will do it, with what resources, in what time period—then organizing, monitoring, and controlling. Above all, effective management considers the people involved in the project.

Criteria for Success

Success in building software products means meeting customer needs, on a schedule responsive to the market, at the right cost, and with high quality. Often success has been low, primarily because of poor management. An effective management process involves planning, organizing, monitoring, and controlling the project. It includes estimating product and project size, and—most importantly—dealing with people.

Although this article deals specifically with effective management of software projects, most of the principles can be applied to any type of project.

Planning the Project

Every project should have a series of plans: an overall project plan and several supplementary plans, which are described in this article. A plan is defined here as a document that states goals, presents approaches to follow, states constraints, identifies activities, identifies inputs to and outputs from activities, identifies mileposts, assigns responsibility for activities, specifies resources, sets schedules, and provides for progress monitoring.

There are two major planning guidelines. First, each project must establish a realistic objective, setting forth the goals in detail. Second, the project must identify all activities (and their inputs and outputs) required to develop the product and determine their sequential relationships, defining what will be done in terms of *observable quantities* and dependencies.

Project Plan. The project plan is unique for each project and covers all aspects of the project, including responsibilities of other organizations. It identifies what is to be done, who will do it, how long it will take, resources required, and strategies needed to guide the project. Specifically, it must state how many people, with what skills, are needed at what times, and must address all the equipment needed to build, test, deliver, and maintain the product.

The project plan identifies key mileposts that will prove progress toward the objective, and all tasks that must be completed to reach each milepost. Each task must be assigned to a specific individual or organization to eliminate ambiguity of responsibility. The responsible party must provide time and cost estimates for each task. The plan must also contain *one schedule*. Responsible parties must agree to this schedule, as well as to the complete plan.

A workable project plan contains an executive summary, management approach, and supplementary plans for development, facilities, organization, budget and resources, documentation, test, qualification, maintenance, deployment, training, and quality assurance. A start-up plan is also recommended for the definition and architecture phases of development. All these plans are not described in detail, since the definition given earlier applies to each; but pitfalls, reasons, and recommendations are given in the sections that follow.

Start-up Plan. The goal of the start-up plan is to provide information needed for a "go-nogo" decision about the proposed project. It should cover the following topics:

- Problem description
- Feasibility, economic, market, and risk analyses
- Prospectus (business case)
- Budget, facility, and staff requirements
- Preliminary project plan
- Functional requirements
- Architecture definition
- Final project plan
- Assumption resolution.

A small team should plan the start-up project and execute it. Work should be scheduled in one-week increments, with no activity lasting longer than two weeks. Planning should provide for frequent communication among people doing diverse tasks. At least two standard milepost reviews (described in the section on Monitoring the Project) are necessary, one to review the preliminary project plan, and a second to review the final project plan, to decide whether to continue.

The output of this phase of the project is critical, as it defines the product. Inadequate requirements or an unwieldy architecture definition will cause serious problems downstream, as will inadequate planning.

Executive Summary. This document abstracts the project plan, prospectus, and problem description. It must be short, but present significant information, enough to provide the basis for a decision about the project's viability. To justify continuation of a project, the executive summary must show that

- The problem is understood.
- The solution (product) is understood.
- The project is feasible.
- The project benefits the company.
- A market exists for the product.
- The risks and assumptions are reasonable.
- The planning is satisfactory.

Finally, the executive summary must clearly state the resources and commitments required to produce the product, it must summarize the case, and must recommend action.

Management Approach. This document, prepared by the project manager, gives strategies to guide the project, clearly stating expectations of people and groups. Although it reflects the project manager's view, other project members must "buy into" it. It should cover development, management, project, and global issues.

Development Issues. A release strategy and a development strategy are required. Release strategy deals with issuing the product in one release or in multiple phased releases. Large projects usually have more success by spreading releases over time. Development strategy addresses the same issue internally: one delivery to the test organization or several incremental deliveries. Projects in which the interval from design through unit test is longer than four to six weeks should use incremental development.

A project should have a single development method, chosen by qualified members of the staff and adopted as its standard.

The project also needs a strategy and methods for testing and, finally, a maintenance policy. (The plans for these are described later.)

Management Issues. The management approach document should specify when to use audits, standard milepost reviews, and inspections, and how to obtain training in their use. At least one audit shoũld be scheduled for each project, and a standard milepost review scheduled for each major milepost. Every major document and feature should be inspected.

Project Issues. Control mechanisms should be specified, encompassing, as a minimum, baselining (defining the point at which changes require formal approval), change control, configuration control (inventory of the product piece parts), and a software manufacture process (assembly of the product). In addition, quality goals and quality-improvement goals should be stated. The goal statements become the basis for the quality assurance plan (discussed later).

Global Issues. Since deploying the product requires setting aside resources for maintenance, a deployment strategy is needed. It includes identification of key customers as well as identification of affiliate organizations and the type of support required from each.

Development Plan. This plan defines the feature packages, the activities to produce, deliver, and support them, and the schedule for all activities.

Although the plan can be for a single release of a product, phased releases are usually more appropriate. Phased releases deliver many features to the customer every 12 months or so, and provide early feedback about the development process and product quality. (Phased releases incur additional costs, however, because of overlapping maintenance and development.) Each phase should demonstrate visibly that it gives what the customer wants. The first release includes design of the architecture and basic system structures. All releases require design decomposition to the unit level (work that can be done by one person in two to four weeks).

The product architecture should be simple enough that it is easy to understand how work flows through the system. Allocation of major features, data bases, and work centers to subsystems must be clear, and both external and internal interfaces must be included. Planning must define steps to verify and protect the architecture, and to ensure agreement by all interested qualified parties.

Each product release follows a common pattern. A detailed schedule for all aspects of the first release can be a template, which can be used for all releases with few changes, provided technical detail is separately maintained in tables and matrices.

To provide for the unexpected, the development plan should include a contingency plan, which may involve having increments only partially full, or an extra increment following a risky increment.

Facilities Plan. This plan deals with the hardware and software needed for product development. It is often preferable to contract for facilities, since this incurs least effort. If the project owns facilities, planning must address choice, acquisition, installation, operations (methods and staff), and maintenance. In either case, training is required. Even with project-owned facilities, such as a computer center, it is preferable for it to be operated by a contract staff to avoid personnel problems; that is, so that operators can be evaluated with their peers rather than with project staff.

Inadequate facilities and tools will severely limit productivity. One reason for this is the effect of "virtual machine volatility" (computing facility stability); see Reference 1.

Organization Plan. The management structure should be as close to a project organization as possible—one project manager controlling all resources. This may be difficult, but a well-organized project can even cross organizational boundaries, using internal and external resources effectively. In addition, the development organization should match the product architecture; that is, there must be a group of people for each product unit, and responsibilities and lines of communication must be clear. As the organization is being reviewed, the manager should be able to see an unrestricted work and product flow and find product and document ownership well defined by the structure. The management of each person or unit participating in the project should endorse this plan in writing.

Budget and Resource Plan. This plan presents requirements for resources: people, facilities, and money. If a resource is not available, plans must be made to acquire it.

Staff resources should be presented in a matrix showing numbers of staff by job as a function of time. Other costs should be presented by line item as a function of time. Simple graphs showing totals as a function of time should summarize costs. (A more detailed "work breakdown structure" is desirable; see Reference 1.)

Software cost estimation is also covered extensively in Reference 1, but some elaboration is appropriate here. Estimation is probably done best by estimating size (related to *what* is built), and then resources (related to *how* it is built). Thus, the effects of cost drivers, for example environment, become more obvious, and steps can be taken to reduce their influence.

Even in algorithmic methods, estimating size or cost depends on the similarity of what will be built to what has been built. It is essential to document the estimation process and also expenditure of resources. Demarco (Reference 2, Chapter 5) outlines ways to improve data collection.

Documentation Plan. This plan identifies required documents and fixes responsibility for their preparation, inspection, review, and approval. A schedule shows the planned, estimated, re-estimated, and completion dates for each document's draft, inspection, revision, approval, reproduction, and distribution. This is best shown in a table with major mileposts indicated on the project schedule.

Test Plan. According to the IEEE Standard for Software Test Documentation,[3] the test plan describes the scope, approach, resources, and schedule of intended testing activities. It includes test items, features to be tested, testing task, person or organization to do each test task, and any risks requiring contingency planning.

Since the purpose of testing is to find defects, design of the process should reflect that. In particular, compressing the test interval results in defects in the field; sharing test and development responsibilities leads to the neglect of one or the other; and rewards to the testing staff should be based on individual job success, not project success.

Testing should start during the requirements phase and should use an independent system test group, test inspections, and frequent demonstrations.

Qualification Plan. The qualification (trial) plan specifies requirements for tests conducted by an independent organization to qualify a product for general release. The plan must address

- Customer and site selection
- Execution of a letter of agreement with the site
- Site preparation—training, installation, conversion
- Conduct of qualification—support, monitoring, data collection
- Evaluation—data requirements and analysis, acceptance criteria
- Rework.

Maintenance Plan. The maintenance of a system release is a project in its own right. This section addresses issues that are particularly significant when the product is in the customer's hands.

A maintenance policy should define project commitment, including when changes will be made, time required, and frequency. A process should be defined for receiving data, evaluating data, deciding whether to accept or reject the customer's requests, implementing the accepted requests, and notifying the customer about modification request status. The maintenance plan should explain how the customer issues a modification request and how quickly it is evaluated.

The planning process must allocate personnel for maintenance, recognizing that maintenance depletes resources. Maintenance training may well be required. It is important to ensure that personnel are rewarded commensurate with their performance *in maintenance.*

A formal change control procedure is required for every project output, whether during maintenance or development. The modification request process is one aspect of change control. The other is configuration control—knowing what is a valid configuration of any version of the system at any time. A change control plan provides for developing and using these processes throughout the life of the project. Delivery of changes should be considered—who is responsible, the process, convenience to the customer, advance notification, etc.

Customer support should be integrated into the modification request process. In fact, the support staff may well be the primary source of modification requests because of their contacts. Customer support should follow certain principles:

- Availability of contacts
- Knowledgeability of contacts
- Responsiveness of contacts
- Presence (site support must be available).

Deployment Plan. Deployment of a system requires consideration of several issues: key customers, rate of deployment, delivery and installation, customer support, and system maintenance. The deployment plan makes decisions about the first two consistent with the resources available for the last three.

Key customers are those whose acceptance will lead others to purchase the product, or justify building the product without more customers. Explicit activities are needed to identify key customers early, usually during product definition. It is necessary to define the order in which they are approached, trading off deployment in sites where early maintenance will be needed against deployment where the system will not be stressed as heavily. Project management must take steps to ensure that the marketing organization follows the strategy.

The deployment rate must be controlled to be consistent with maintenance resources and also with delivery resources and (perhaps) hardware availability.

Training Plan. Any training plan must provide for the following:

- Determination of needs
- Determination of source—vendor if possible
- Preparation of original material—creation, review, rewrite, and trial
- Identification of trainers by name
- Schedule and delivery—considering audience availability.

A matrix best presents the data, showing for each course the person responsible for preparation; dates for draft, inspection, trial, and revision; person responsible for delivery; date for trial of course; revision; schedule for presentation to classes.

Quality Assurance Plan. Poston[4] discusses the writing of quality assurance plans. This section highlights some issues.

Customer-related quality criteria are needed at each step, from concept to customer use. They must be appropriate for the product, and observable. There must be a way to derive objective measures of the criteria, which can be used to improve quality. The desired level of quality must be stated and targets set to meet the objective measures.

Procedures must be defined for measurement, staff allocated to perform those procedures, schedules set, and reports produced and brought to the attention of management. In addition, procedures must be defined for determining and eliminating causes of inadequate quality.

Organizing the Project

Organization has been mentioned earlier, but this section discusses the role of the project manager and expands some points.

Paramount in importance is appointment of a full-time competent project manager whose administrative, technical, and political skills are equal to the task. This person must have enough authority to bring needed operations and support into the project system. The project manager's role and responsibilities must be clearly defined. He or she must be willing to take reasonable risks and assume the initiative when problems loom.

Each project must be organized to keep abreast of events, to ensure that information is disseminated, and to coordinate interdisciplinary activities. Appropriate communications links must exist among all elements of the company that play a role in the project's success. Organization must be such that structures and interfaces are clear, there is a single point of responsibility for the achievement of each activity and product, and product and documentation ownership are known throughout the project's life cycle.

The project needs the proper mix of technical and administrative skills. Organizing this effort requires identifying valid skill mixes, a plan for recruiting, and a plan for training.

Monitoring the Project

Project monitoring is based on the principle that there is a *single* plan (and supporting schedule) against which the progress is monitored. Metrics must be defined to track progress (see Tracking below).

Evaluation. The three basic monitoring processes are audits, standard milepost reviews, and inspections.

Audits. An audit is a formal examination, by an external team, of the project development and management processes. There should be at least one audit for any project, and it should occur sometime after completion of the final project plan. An audit begins with selection of a leader, who assembles a team. The team reviews project documents—project plan, requirements, architecture, etc.—then conducts in-depth interviews of the project manager and staff, key members of the project, and others as appropriate, including customers. The team reviews the baselining process (that is, the process for stopping unapproved changes), the change control system, and crisis management, and pays particular attention to management of people and over-commitment. Finally, the team makes a formal report and follows up with a visit to see how the project implements the team's suggestions.

Standard Milepost Reviews. A standard milepost review is a formal examination, by management, of official project mileposts, to approve output and to authorize continuing or stopping a project. Reviews are usually conducted after completion of feasibility analysis, product requirements analysis, system integration and test, and qualification.

Inspections. An inspection is a formal examination by a local peer group of some particular deliverable part of the project, for example, design documents or code. The process follows these steps:

- Planning: Moderator schedules and distributes material.
- Overview: Originator educates participants.
- Preparation: Participants study.
- Inspection: Participants search for errors.
- Rework: Originator corrects errors.
- Follow-up: Moderator ensures corrections are made.

No work should be delivered to another organization until it has passed inspection.

Project Review Meetings. A project review meeting is a formal, periodic meeting of key project staff to review goals, progress, status, and problems. These meetings follow a fixed written agenda and are held on a regular basis (every two to four weeks). They are held at a fixed time, on a fixed day, and at a fixed place. Attendance is mandatory for named participants.

The attendees review goals, mileposts, and current status for the previous period, current period, and next period, emphasizing activities that are now critical or are becoming critical. Progress is reported in detail against the plan for each task under review. The project manager must encourage forthright reporting of problems. When problems are encountered, an action item should be written and the meeting should move forward. Existing action items should be reviewed. The meeting should be used to present statistics about development, reliability, performance, and quality, and to review status of modification requests.

The recorder prepares a synopsis of the meeting and attaches a copy of handouts, visuals, and the action item list. This package constitutes the minutes, which are distributed promptly to all members of the project.

Tracking. It is the project manager's responsibility to track all project activities, in particular schedule, budget, and product and project performance.

Product Tracking. One quality metric is the number of failures per thousand lines of noncommented source code delivered to the customer. Performance is tracked in terms of capacity and response time. Reliability is tracked by mean time to fail, to recover, and to fix problems.

Project Tracking. This includes tracking budget and resource usage and tracking the schedule against time by milepost completion. The project manager must encourage revised estimates when difficulties arise so the schedule remains realistic. Functionality is tracked by date of demonstration. Development statistics must be tracked to show how resources are being expended and how products are being produced. Action items must be tracked to ensure closure in a timely manner. Any reported failures are tracked, paying particular concern to time to clear the report. Tracking market penetration is necessary to keep the project personnel aware of product acceptance.

Demonstrations. At the end of each project review meeting, supervision should see a demonstration of completed increments. Demonstrations, more than any other approach, make mileposts visible.

Controlling the Project

The sections under Planning the Project have already mentioned some control mechanisms, such as phased release and incremental development, and change control processes. Another control mechanism is software manufacture, which means that all parts of the product are assembled by a small group of experts. Production details are thus hidden from the people who do not need to know them. This helps to realize the other control mechanisms, especially since software manufacturing often carries much of the configuration control responsibility.

Restoring Control. In the event that control mechanisms break down, provision must be made to deal with crises or troubles. A crisis is more than minor slippages or loss of staff. A crisis exists when something is so wrong that a significant project failure is expected. The best way to handle a crisis is to avoid it by establishing a "crisis alert" process for identification of a possible crisis *before* it occurs. This requires that

- Criteria for identifying crises are defined beforehand.
- Values of the criteria are tracked and reported.
- A standard report mechanism exists (and is used).
- The reporter of an impending crisis survives the report.

Issuing a crisis alert should be a part of standard project routine. The project manager must create an environment in which the reporter knows that reporting an impending crisis carries no stigma, because the report gives time to avoid the crisis.

Because a crisis is rare in a well-managed project, things must be done in a crisis that are foreign to the normal way of doing business:

- Track status in detail, that is, by day.
- Try unusual methods, for example, two people on a "one-person" job.
- Concentrate authority with one person.
- Waste resources—money, staff, computer time.

Total commitment is critical. Those who can help must; those who cannot help must stay out of the way.

A project is in trouble when there is no way to meet the original schedule at the cost target with the required features and quality. It is safe to say that this comes about because of ineffective management:

- Over-commitment
- No planning, or planning at too high a level
- Inadequate understanding of required function
- No mileposts, or mileposts too fuzzy or too far apart
- No risk management
- No clear definition of responsibility.

The remedy is resolution of management problems, either with existing or new management personnel. Poor habits, such as missing mileposts, must be changed. Clear plans must be formulated and followed. Above all, the failure habit must be eliminated, often by helping staff produce concrete outputs on schedules they set.

Summary

The basic principles of effective project management are few and simple, namely:

- Plan: Identify activities; assign responsibilities; provide resources; define crisp mileposts; set schedules.
- Organize: Structure the organization like the product, because the product will look like the organization. Make responsibilities and interfaces clear.
- Monitor: Track mileposts, performance, and budget.
- Control: Install mechanisms at project inception and use them throughout the project. Take control when needed.

The most important aspect of effective management, from inception to completion of the project, is consideration of the people involved.

References

1. Barry W. Boehm, *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, N.J., 1981.
2. Tom Demarco, *Controlling Software Projects*, Yourdon Press, New York, 1982.
3. "IEEE Standard for Software Test Documentation," IEEE Standard 829-1983. (Order from IEEE Service Center, 445 Hoes Lane, Piscataway, N.J. 08854.)
4. Robert W. Poston, "IEEE 730: A Guide to Writing Successful SQA Plans," *IEEE Transactions on Software Engineering*, March 1985, pp. 86-88.

(Manuscript received January 27, 1986)