Journal of International Information Management

Volume 1 | Issue 1 Article 5

1992

Software project management - there is never enough time to do it right!

Brian McNamara
California State University, Bakersfield

James Uigen
California State University, Bakersfield

John Tarjan
California State University, Bakersfield

Software project management - there is never enough time to do it right!

Brian McNamara
James Uigen
John Tarjan
California State University, Bakersfield

ABSTRACT

Efficiency in software development projects is examined. Project management tools—GANTT charts, program evaluation and review technique (PERT), and critical path method (CPM) are defined and discussed. Of these productivity tools GANTT charts and PERT are the most relevant. Based on an analysis by Shelmerdine (1989), six steps that provide guidelines for software development are presented. These steps provide the project manager with an opportunity to produce efficiency and deliver a software development project on time. Selected project management software packages are presented.

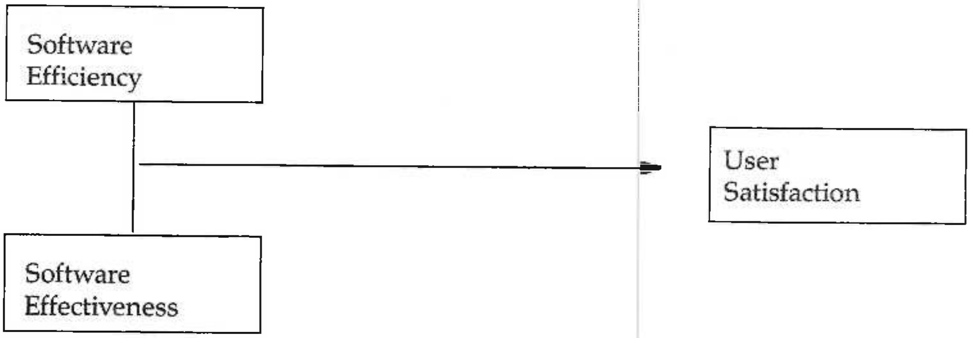
INTRODUCTION

Two major problems facing management information systems (MIS) managers are the tracking of multiple projects and planning and scheduling of individual projects. In terms of tracking multiple projects, MIS managers are being asked to develop plans and budgets for these projects and then be accountable for ensuring that they are met. In many cases the manager does not have control of budgets or schedules and is being asked to operate the MIS function as a profit center (McCusker, 1989). On the other hand, project management relating to planning and scheduling of individual software development projects continues to be pervasive. Murphy's Laws such as "There is never time to do it right but there is always time to do it over," continues to represent the mode of operation in many MIS departments. This is a reflection of fact that methodologies used in the software development cycle are better understood than how to apply these same methodologies in real world software development projects.

Within the context of software development there are two general areas to consider when discussing development of software: the efficiency with which the software is produced and the effectiveness of the software produced. These relationships are reflected in Figure 1.

There are several ways in which authors have differentiated between effectiveness and efficiency issues in software development. Boehm (1981) dichotomized software engineering goals into product and process goals which can have qualitative and quantitative measures. Process issues deal with the production of a piece of software while product issues refer to the attribules of the final product (Boehm, 1981). Thus, process goals relate to efficiency issues while product goals relate to effectiveness measures.

Figure 1



Due to the vastness of the subject matter involved in software development, further discussion will be restricted to efficiency measures. Efficiency is concerned with the ability to coordinate and control software development utilizing the organization's resources required to produce and deliver a software product. Efficiency is also the product of good project management skills since all projects are dependent on the planning, control, and leadership qualities of the personnel involved in the project.

Unfortunately, project management and software development are treated as mutually exclusive terms in much of the literature. When the use of project management is considered, especially in textbooks, the subject is normally relegated to an appendix. This situation is counter to the importance of project management in software development. This importance is supported by research findings. For example, Thayer, Pyster, and Wood (1980) found that 90% of an extensive list of problems commonly encountered in software development were either totally or partially managerial in nature (Thayer et al., 1980). Bruggere (1978) reports three key factors which determine the success or failure of a software development project. These are: (1) project management, (2) personnel selection, and (3) development methodology (Bruggere, 1979).

Project Management

Project management productivity tools can be a factor in ensuring successful software development. In particular, the three most popular productivity tools for project planning and scheduling are: Gantt Charts, Program Evaluation and Review Technique (PERT), and Critical Path Method (CPM). These methods have been available for over thirty years and are information producing techniques that provide a practical and structured method to plan, schedule, monitor, revise, and control the progress of a project.

Gantt Charts are a bar charting technique which shows planned progress for a number of activities, that have to be completed, against a horizontal time scale. One weakness of the Gantt Chart is its inability to depict the interdependency or interrelationships between activities. PERT and CPM charts show the same information but incorporate precedence relationships into the chart. Both PERT and CPM are graphic network techniques that depict the flow of activities through the network in terms of the sequence in which they must be completed. PERT is oriented to stochastic activity time estimates and uses stochastic inference in determining the probability that the project will be completed on time or within a particular time frame of reference. CPM assumes that activity times are stable or deterministic and uses these activity times to control both time and cost elements of the project. These estimates are used to control both time and cost elements of the project. CPM is generally used for projects in which there is a defined relationship between resources allocated to an activity and the completion time associated with the activity. Within software development projects, this relationship may be difficult to establish since the resources devoted to activities are almost exclusively human (programmers) in nature. However, there may be instances where the addition of resources may lead to a reduction in activity times and in these instances, CPM could be used. PERT is used in situations where there is likely to be variances within the activity completion times and the relationship between the addition of resources and reduction in activity times is not well established. This situation is compatible with the environment of software development projects. As stated by Powers, Adams, and Mills (1984) "In systems development, it is difficult to relate the allocation of resources to the time required to complete any task.

Advantages of using PERT/CPM have been identified by Wolf and Hauck (1985) and they include:

1. It forces a thorough pre-planning of each task.
2. A better coordination of the work to be performed is achieved.
3. Problems are resolved on paper before they occur.
4. It focuses management's attention on to the critical path activities rather than on non-critical path activities.
5. Thorough pre-planning reduces the chance of omission of a task.
6. A network diagram is a working model which can be followed with little explanation.
7. The scope of the entire project can be readily seen on a summary network.
8. The added cost of crashing (reducing the activity time of critical activities) can be determined.

Hence, this network technique forces a software developer to focus on all aspects of the project. Continuing and further expounding upon this theme, Shelmerdine (1989) identified six planning steps which he feels must be adhered to for successful project management. These steps also overcome one of the shortcomings usually attributed to scheduling with PERT/CPM models in that they do not address resource utilization and availability. The six steps include:

1. Identificaton of each function (activity) that must be undertaken to successfully complete the project. What are the objectives of the project and the activities that will meet these objectives.

2. Identification of the specific tasks involved in each of the activities.

3. A complete description, in writing, of what is to be produced by each activity. That is, the "deliverables" (documents) from each task. A description of what constitutes successful completion of the tasks and activities.

4. Development of the PERT Chart for each activity.

5. An estimate of the effort to be expended in completing the activities. PERT activity times are stated in worker-hours. The average for each activity is based on three time estimates—most optimistic, most likely, and most pessimistic. In software development projects these estimates are not generally made by the individual who will perform the work associated with activity tasks. They are made by managers and supervisors and their accuracy and reliability will be predicted on the knowledge of the individual making the estimate. For example, will the resource (worker) assigned to the task be of a high skill level, average worker, or a new trainee? If the resources (workers) get committed to the project, the uncertainty in the time estimates is reduced since these resources will not be assigned to another project. Once resource (worker) names are assigned to an activity, a determination must be made as to when the resource will be abailable. This is necessary because if equipment, material, etc., are not available at the appropriate time, the individual cannot execute the planned tasks. Also, in refining the time estimates, other planned and unplanned activities must be considered. Unplanned activities such as turn-around time, structured walkthroughs, and project status reviews must be considered. Other planned activities would include vacations, sick leaves, holidays, education/training, and travel times. All these factors must be considered if the activity times are to be realistic and provide management with the ability to control the project.

6. With the five steps completed, the project manager is in a position to use PERT and track performance and hold individuals accountable for their performance (Shelmerdine, 1989).

The six steps provide guidelines for software development projects. Whether the project is simple and straightforward or complex and detailed, the basic six steps still apply. The result following these steps will be an effective software product delivered on time.

Each individual step by itself may seem very time consuming and cumbersome but, through the widespread availability of project management software, the work involved in each step can be drastically reduced. The benefits derived from the combination of project management techniques and software which underscores these skills can dramatically increase the productivity and control of a project manager. Ultimately, time and cost overruns will be reduced.

Project Management Software

The vertical software market has not been slow to pick up on the topic of project management. Project management software generally falls into three categories—planning, planning and updating, and automatic resource scheduling. Most software falls into the first two groups of planning and planing and updating. The third category which relates to automatic resource scheduling, is relatively expensive, and operates on the high end of hardware configurations (Davis & Martin, 1985). Software, specifically related to managing individual software development projects and with the ability to track multiple projects, are identified in an article by McCusker (1989). These software include: MICROMANII by Poc-It Management Services Inc., Santa Monica, California; MULTITRACK (Multitrack Software Development Corporation); PROJECT WORKBENCH (Applied Technology Corporation); and SUPERPROJECT EXPERT (Computer Associates International). The cost of these packages varies from PROJECT WORKBENCH at $1,200 to a mainframe version of MULTITRACK at $120,000.

Also on the market are a number of project management software packages that are very modest in terms of acquisition cost. Reviews of these inexpensive microcomputer project management software packages as presented in an article by Vigen, Rudd, McNamara and Ahamadi (1989) and also may be found in articles by Bermant (1986), Davis & Martin (1985), and Poor & Brown (1986). These reviews do an excellent job of detailing the basic and advanced functions offered by project management software costing more than $250 such as SUPER-PROJECT (Computer Associates), HARVARD TOTAL PROJECT MANAGER (Software Publishing Corporation), and PROJECT SCHEDULER (Scitor Corporation). In the $200 and less category, Bermant includes ADVANCED PRO-PATH 6 (Softcorp Incorporated), SCHEDULING AND CONTROL (Softext Publishing Company), MILESTONE (Digital Marketing Corporation), EASYGANTT (Morgan Computering Company, Incorporated), PROJECT MANAGEMENT (International Machines Corporation), and others. In these reviews, the price of the packages and the computer systems requirements necessary to effectively use the software is documented. However, due to rapid change within the software industry, many of these packages have been updated and many more products have entered the market. Moreover, there was no one project management software package declared as a clear winner in this literature. However, the reviews indicated that the key to software performance is the user's skill and judgement rather than any special features in the software itself.

The above list and references to project management software is not exhaustive but will give the project manager a place to start the process of selecting the software packages which best reflects their needs. Suitable selection of a package will pay for itself in a very short time due to increased control of projects and productivity increases.

CONCLUSION

With the effective project management tools described in this article, a software development leader has the capability to create an effective plan which can be implemented and monitored. Through systematic monitoring, corrective action may be taken as the situation warrants and the primary goal, of delivering the project on time, can be attained.

It is well known that software development mismanagement can destroy the best of analysis and design methods; hence, the authors have dealt with the efficiency side of software development projects. In many instances when one hears that a software development project is well behind schedule, the assumption is that major calamities have been the cause of the delay. The reality, however, is that the delay is generally caused by day-to-day problems that make the project late one day at a time (Brooks, 1974). Because of the efficiency and control created by project management tools, the manager has a built-in early warning system against getting behind schedule one day at a time (Brooks, 1982).

REFERENCES

Boehm, B. W. (1981). *Software Engineering Economics.* Englewood Cliffs, NJ: Prentice-Hall.

Brooks, F. P. (1974, December). The Mythical Man Month. *Datamation.*

Brooks, F. P. (1982). *The Mythical Man Month.* Addison-Wesley Publishing Company.

Bruggere, T. H. (1979). Software Engineering: Management, Personnel and Methodology. *Proceedings, Fourth International Conference on Software Engineering*, 361-368.

Davis, E. W. & Martin, R. (1985). Project Management Software for the Personal Computer: An Evaluation. *Industrial Management, 27,* 1-21.

McCusker, T. (1989, October 15). Project Planning Made Easy. *Datamation*, 49-50.

Poor, A. & Brown, B. (1986). Project Management Software: The Top Sellers. *PC Magazine, 5,* 155-164.

Powers, M. J., Adams, D. R. & Mills, H. D. (1984). *Computer Information Systems Development: Analysis and Design*. Southwest Publishing Company, Cincinnati, Ohio, 618-619.

Shelmerdine, E. K. (1989, January). Planning For Project Management. *Journal of Systems Management*, 16-20.

Thayer, R. H., Pyster, A. & Wood, R. C. (1980). The Challenge of Software Engineering Project Management. *Computer, 13*(8), 51-59.

Vigen, J. W., Rudd, H. F., McNamara, B. & Ahmadi, M. (1989). Project Planning and Control Software for Small Manufacturing Firms. *International Journal of Management, 3,* 282-287.

Well, Bermant, C. (1986). Project management under $200. *PC Magazine, 5,* 191-199.

Wolf, G. L. & Hauck, W. C. (1985). PERT/CPM: A Tool for Managing Projects. *Industrial Management, 27,* 22-25.