# PLANNING AND SCHEDULING SOFTWARE DEVELOPMENT PROJECTS

DALE H. FAWCETT
Simulator Systems Engineering Department
Goodyear Aerospace Corporation
Akron, Ohio

## INTRODUCTION

The planning and scheduling of the resources for a project is one of the most critical activities for the success of a project. It is necessary for management to spend considerable time on this activity for a large software development project. This time is well spent as good planning and scheduling is a requirement for a successful project. The schedule, together with a reporting scheme, ensures that the project is progressing as expected or quickly informs management if it isn't. The schedule provides a primary communication channel between the technical and managerial personnel. The schedule must, therefore, be correct and meaningful to both the technical and managerial teams. The monitoring of project progress when compared to the schedule informs management of the project status. If problems develop, management can be quickly informed so that corrective action can be initiated. The plan and schedule provides a specific guideline for the technical personnel to assist in visualizing the relationship of specific tasks to the total project.

To accomplish effective software planning, a set of three schedules must be generated. These schedules are an activity network of tasks, a Gantt chart of personnel resources utilization and a table of computer resource utilization. The project planning and scheduling is accomplished in the following steps:

Define and estimate tasks

Generate task activity network and schedule

Generate personnel and computer resources schedules

Update activity network with resource utilization data

Verify the interactions of the three schedules.

Many projects have had a good plan and schedule of tasks to be accomplished and have failed to be completed on time. One of the causes of this failure is the lack of a resources utilization schedule which is interactive with the task schedule. The limited personnel and computer resources must be carefully accounted for in the overall project plan. The planning and scheduling method must, therefore, account for the characteristics of software development and the interaction of limited resources upon the task schedule.

The plan must also be flexible to accommodate schedule adjustments during the project caused by inaccurate estimates or unexpected events.

The planning of a software development project begins with defining the tasks to be accomplished. Then estimates of the resources for each task are made. Finally the interdependencies of the tasks must be described. The scheduling of the project can then be accomplished. The schedule must account for the tasks and utilization of the resources. As the project progresses, the plan and schedule must be modified if necessary to reflect the project status and any new developments.

## TASK DEFINITION

Based on the functional and design requirements of the system, the software design can be established. Using a "Top Down Technique," the software is decomposed into hierarchical tasks each of which can be clearly defined. The decomposition process consists of subdividing each task into the next level of detail. Each task at the lowest level of detail consists of specific actions such as design, code, debug/test or integrate for each distinct program entity. The level of decomposition is of course variable, dependent upon the overall project. This level must be

detailed enough to provide a capability of timely monitoring. If the task breakdown is at too high a level, the project could be significantly behind schedule before management is informed. If the level of decomposition is too low, the planning, scheduling and tracking of progress becomes an unmanageable task because of the excessive paper work. Basically, the level of decomposition should provide visibility in the time frame of the schedule for progress reporting to management. For example, if management should be made aware of progress on a bi-weekly basis, then specific tasks should have a duration of from one to four weeks. This level of decomposition should not be a burden of record keeping and should be adequate for management to see and react to problems so that corrective action can be taken in a timely fashion. In a large project with several levels of management, different schedules oriented toward each hierarchical level of tasks may be required.

During the planning stage, all tasks must be anticipated because errors of omission can be at least as costly as errors of commission. It becomes very difficult and costly to have to add unplanned for tasks after the project has progressed downstream. A careful top-down analysis of the problem should define all necessary tasks at the beginning. Extra attention must be given to tasks and programs which are required to support the main effort. These secondary tasks are frequently omitted from the original plans for projects, but they must however be performed, and will require both personnel and computer resources. Because of the dependence of main programs on the support programs, support programs must be completed prior to the main programs. Therefore, accounting for the support programs at the first stages of design is very important.

## ESTIMATING

Each of the tasks should be identified and clearly defined so that a valid estimate of the required personnel and computer resources can be made. With inaccurate estimates, rescheduling will be done frequently as the project continues until valid estimates are made. This activity prevents management from clearly seeing what is happening and from making rational decisions. Estimates are not perfect, but they must be reasonably good for a plan and schedule to be effective. Methods of estimating software are quite varied. Estimating methods range from the use of accurate records on previous projects to uneducated guesses. Specific techniques for estimating are beyond the scope of this paper, but it is assumed that the estimates are derived from the best available technique.

The estimating should be made in a bottom-up fashion. The lowest level tasks are estimated first because they have the most specifically defined details. These estimates are then summed to provide the estimates for higher level tasks and the total project. The bottom-up estimate should be verified by comparison with another estimate such as a top-down estimate. A top-down estimate is generated by evaluating the total project effort in size and complexity by comparison with previous projects. Each specific task should be estimated in person days or weeks and computer resources in hours required per week. The personnel resources estimate must include the number and type of personnel required and account for training or familiarization time needed. The task estimate will indicate the time required for each person and the total task duration. The computer resources estimate must include CPU time, terminal use time and any other significant resources.

Once these estimates are made, they should not be modified unless the modification can be justified by new data affecting the estimate. It is assumed that the most reliable method and information available was used in the generation of the estimate. It, therefore, does not make sense to shorten the time required to accomplish a task even if a rescheduling causes it to start late and someone wants it to complete on time. This activity of trimming a valid estimate has led to many panics because a project appears to be on schedule until the end and suddenly is far behind. However, if new valid information affects the estimate and the change is justified, it should certainly be made. Deliberately ignoring valuable information which causes a change to original estimates seldom leads to success. For a plan and schedule to be a useful tool, it must be maintained as accurately as possible.

## NETWORKING

Programming is an activity oriented process, consisting of activities such as design, code, debug/test, and integration. There is not necessarily a unique event occurring at the termination of each activity. Thus, it is more appropriate to consider the activities than the events such as completion of documents. The task schedule must, therefore, be an activity oriented schedule.

Another important characteristic of software development is the interdependency of the activities. The tasks for each module must be performed in a designated sequence and one task cannot be started until another is completed. The various design levels, coding, test/debug, and system integration tasks must be accomplished in a chronological sequence. There is also an inter-module dependency. Because of the top-down design and testing technique, certain modules must be completed and integrated before others can be integrated into the system. This characteristic or interdependency requires that the scheduling technique must be a network type so that the interdependencies of the tasks and the effects of one event on successive tasks and the overall project can be visualized.

The critical path method (CPM) is ideally suited for software planning and scheduling. CPM is an activity oriented network technique. This enables CPM to show the interdependencies of the software tasks. CPM provides the calculation of float times in the paths and determination of the critical path, the one with zero float time. This capability provides management with the rapid view of the critical tasks which must be monitored more closely.

After all the tasks have been defined, the next step is to lay out the complete network of tasks based on the interdependencies using CPM. This network plan must also indicate the effects of tasks external to the software. The availability of hardware or software from other activities must be indicated. Similarly, if a software task affects other activities, it should be indicated on the network. This helps to ensure that proper communication between major activities of a project is maintained. If changes must be made to a project's schedule, the effects of this on other activities of the project must be considered and the manager of those activities informed. Figure 1 presents a sample of a software network.

The generation of a network chart should be clearly labeled, easy to read and follow basic guidelines presented by O'Brien[6] to ensure its usability. If it fails to conform to these guidelines, it will be quickly discarded as useless.

## RESOURCES UTILIZATION

Another characteristic of a software project which complicates the schedule and control activities is the interactions of the finite resources with the task schedule. The two primary resources are the personnel and computer time. The failure to realize the interactions of finite resources with the task schedule has led to severe difficulties in many projects.

Because of the nature of software development, personnel cannot be randomly reassigned to tasks on the schedule. There is a training and familiarization time required to accomplish the task correctly if a new person is assigned to a new task. It is also often more difficult for several individuals to perform the same function on the same task simultaneously. Because of the intercommunication requirements, three people cannot accomplish a task in one-third of the time required for one person. Another effect of limited personnel is that when one programmer is delayed on a task, the successive tasks for which he has been scheduled will be delayed, possibly affecting the overall project schedule. This effect is not directly visible from the schedule of tasks only.

The computer time resource can seriously affect a project schedule. If the computer becomes overburdened, the turnaround time is adversely affected which can lead to delays in task completions. All of the computer resources must be accounted for in the scheduling including terminals and keypunching facilities because any inadequate resources can produce a delay. In systems such as simulators, in addition to program development,

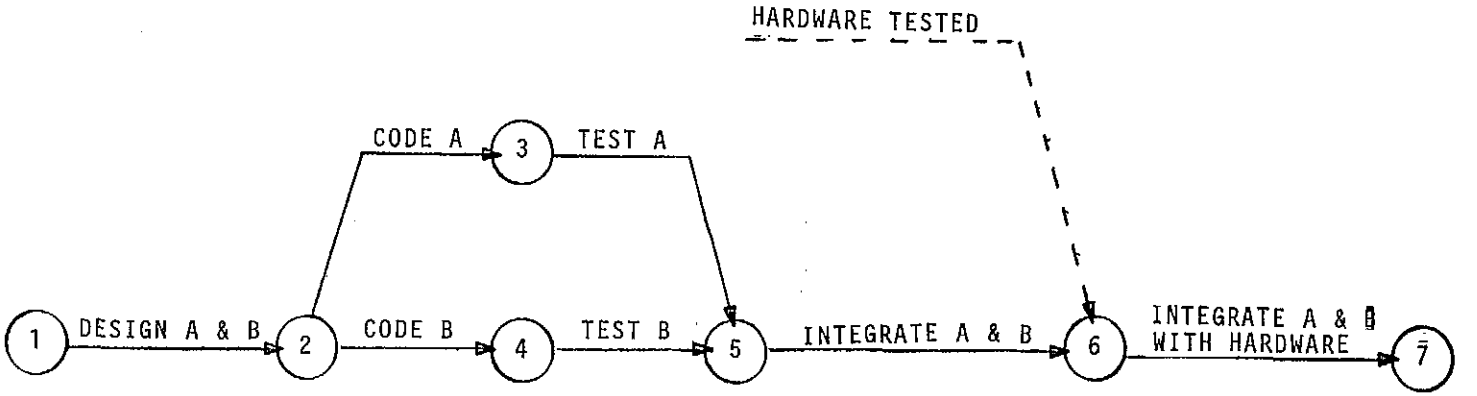


Figure 1. Software Network

the computer test-bed time to integrate the software and hardware is extremely important in scheduling. Projects have reached the final stages before it was discovered that 48 hours of computer time was required per day to complete the project as scheduled. This led to serious delays from which there was no possible recovery. It is, therefore, necessary to ensure that the task schedule does not cause an overrun of the available computer resources. Thus, the scheduling technique must initially account for the utilization of resources and their interactions with the tasks.

## SCHEDULING

Having networked the tasks, we are ready to generate the schedules for the project. The task time estimates are added to the network and the path times are computed. These estimating times should be in whole units of days or weeks as smaller units are not significant for large software projects. The estimated task duration times are used to compute start and end times for each task, and from that, the float times of each one. The path with zero float time is the critical path and the length of this path is the minimum time to complete the project. The critical path is compared with the required completion time of the project. If a conflict exists with the time required being greater than allowed, it must be resolved at this time. Either the project completion time must be increased or the network plan must be modified. After this conflict is resolved, the resources schedules can be generated. The personnel resources schedule consists of a series of Gantt charts or a table of times indicating each person, their assigned tasks, and the times during which they are working on these tasks. This information can then be fed back into the network to schedule the tasks and float times on the noncritical paths. The utilization of specific personnel for selected tasks may affect the overall task schedule. Indications of this effect should be entered on the network in the form of dashed lines with appropriate labels indicating that certain tasks cannot begin until the personnel have completed other tasks and are available. This additional information may aid in the quick assessment of the effects of possible changes to the schedule. The effects of personnel scheduling may even change the critical path of the project.

This phase of the process may reveal that more personnel are required during certain periods or that some personnel have periods with no designated tasks. If it is found that these schedules are incompatible, then adjustments must be made either in the task personnel assignments or the task network. Several iterations of adjustments to the network and personnel schedules may be required to obtain the desired schedules. Figure 2 presents a sample personnel schedule.

Once the task and personnel schedules have been satisfied, the computer resources schedule can be generated. This operation can be performed simultaneously with the personnel schedule, but it becomes more complicated if done manually. This schedule will indicate computer resource requirements by task on a weekly basis. This schedule must account for the type of system to be used. For example, if on-line terminals are to be used, the terminal usage times must be evaluated in addition to total computer time. Based on these inputs, the total computer resources can be computed for each week. This schedule can then be evaluated to determine if there are sufficient terminals available, or how many shifts the computer must be run and if there is sufficient computer power available. If the schedule shows a requirement for more than 168 hours of computer usage per week for the last few weeks, then either a second computer is required or the schedule requires adjustment. Failure to plan for computer resources has been the downfall of many projects. Suddenly at the end of the project, everyone wants computer time and there isn't sufficient time available. The project which was on schedule is suddenly delayed significantly just before completion. This tragic happening can be avoided through proper planning of resources at the beginning of the project. Figure 3 presents a sample computer resources schedule.

This scheduling process can be automated by using a computer program. The program can compute the critical path and verify that the resources are utilized effectively. The task assignment and network plan must, however, be performed manually because of the subjective nature of the decisions. The program cannot know task interrelationships and which person can effectively perform which tasks; and random task assignment is not advised

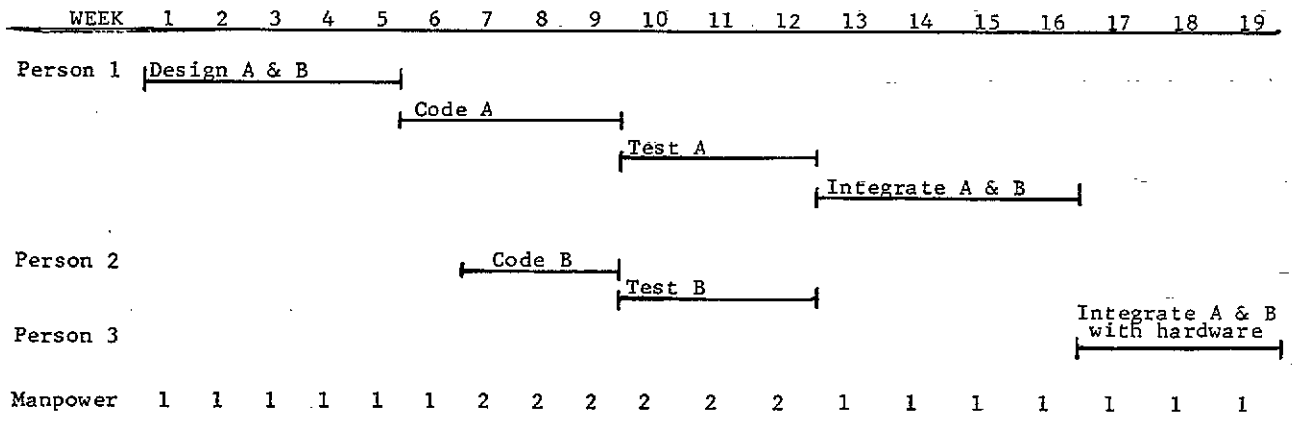


Figure 2. Personnel Resources Schedule

| WEEK | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Design A & B | | | | | | | | | | | | | | |
| Code A | 5 | 10 | 15 | 5 | | | | | | | | | | |
| Code B | | 5 | 5 | 10 | | | | | | | | | | |
| Test A | | | | | 5 | 10 | 10 | | | | | | | |
| Test B | | | | | 5 | 5 | 10 | | | | | | | |
| Integrate A & B | | | | | | | | 5 | 10 | 15 | 15 | | | |
| Integrate A & B with hardware | | | | | | | | | | | | 10 | 10 | 10 |
| Total Time | 5 | 15 | 20 | 15 | 10 | 15 | 20 | 5 | 10 | 15 | 15 | 10 | 10 | 10 |

Figure 3. Computer Resources Schedule

because all personnel are not equal in ability and experience.

The network schedule can be easily oriented toward calendar time. This revised schedule can account for vacation, holiday, and weekend times and provide task completion calendar dates. After the initial schedule has been generated, the calendar start date and time off days can be factored in resulting in a calendar oriented schedule such as in Figure 4. This information can be entered into the network chart resulting in a chart as presented in Figure 5.

On a large project, there may be several levels of managers for the software project. These individuals have different levels of interest in the progress of the project. Therefore, different levels of the task network can be generated corresponding to the top-down hierarchy of tasks. This capability provides each manager with a reasonable schedule for monitoring project progress without unnecessary details, applicable only to the lower level manager.

## PROGRESS MONITORING

Once the task network and resources schedules have been completed, they can be used to monitor the progress of the project. It is important to remember that the plan and schedules are based on estimates. The schedule serves as a guide but is not perfect. As the project proceeds, changes may be required in the schedule and plan. Therefore, they must be flexible, and it must be possible to implement changes. Very small deviations may be ignored because the effort involved in documenting the schedule change is significant. A task which deviates from schedule by one week out of 10 or 20 should not cause a schedule update as the estimate may be off by 5 or 10% which could nullify the deviation. Of course, the decision to regenerate the schedule is subjective and no hard and fast rule can be made. Major changes affecting schedule must lead to a reschedule process. If the changes affect the resource estimates, new estimates must be generated and used to update schedules. The task network, personnel and computer resources schedules must all be updated and verified for compatibility. The schedules can then be used by management to track the project progress. When significant deviations occur, management can quickly react to accommodate the changes. As long as no significant schedule deviations exist, management can devote its energies to other projects which require assistance. This management-by-exception is highly dependent upon the belief that the schedule is accurate and that the reporting of progress is true.

Progress reporting should be highly objective and not subjective. Although the project is activity oriented, each activity must be clearly defined so that it can be easily seen when it begins and ends. This is essential so that progress reports do not indicate 90% completion for 50% of the time. The person reporting the status must be able to produce visible evidence of task completion such as a listing or document which can be easily evaluated. The lowest level tasks are considered to be zero percent complete until they are 100% completed. This policy avoids subjective and often meaningless status reports. The short duration of these tasks accomplished through the heirarchical decomposition makes this method of status reporting possible.

Accurate records of the progress and exact resource expenditures should be maintained. The data collection of actual personnel and computer resource expenditures can be part of the progress monitoring activity. If a rescheduling occurs, the old schedule should be filed and not discarded. When the project is complete, this data base can then be used for evaluating the overall project accomplishments and for future estimating and planning activities. Whether the project follows the original schedule or not, the collection of this data is very useful for future reference.

## CONCLUSION

A large software project is a very complex task. The problems of managing such a project can be greatly reduced through proper planning and scheduling of the tasks, personnel and computer resources. One of the keys to successful scheduling is to consider all of these items simultaneously to account for their interactions, and the interdependencies of the various software tasks. Using a network for task planning and scheduling, Gantt charts for personnel resources and a table of computer resources

| ACTIVITY | DURATION | TASK DESCRIPTION | RELATIVE DATES | | | | | | | CALENDAR DATES | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | EARLY START | EARLY FINISH | LATE START | LATE FINISH | POSSIBLE FLOAT | SCHEDULED START | SCHEDULED FINISH | SCHEDULED START | SCHEDULED FINISH |
| 1-2 | 5 | Design A&B | 0 | 5 | 0 | 5 | 0 | 0 | 5 | 1-10-77 | 2-14-77 |
| 2-3 | 4 | Code A | 5 | 9 | 5 | 9 | 0 | 5 | 9 | 2-14-77 | 3-14-77 |
| 2-4 | 3 | Code B | 5 | 8 | 6 | 9 | 1 | 6 | 9 | 2-21-77 | 3-14-77 |
| 3-5 | 3 | Test A | 9 | 12 | 9 | 12 | 0 | 9 | 12 | 3-14-77 | 4-4-77 |
| 4-5 | 3 | Test B | 8 | 11 | 9 | 12 | 1 | 9 | 12 | 3-14-77 | 4-4-77 |
| 5-6 | 4 | Integrate A & B | 12 | 16 | 12 | 16 | 0 | 12 | 16 | 4-4-77 | 5-2-77 |
| 6-7 | 3 | Integrate A & B with hardware | 16 | 19 | 16 | 19 | 0 | 16 | 19 | 5-2-77 | 5-23-77 |

Figure 4. Calendar Schedule

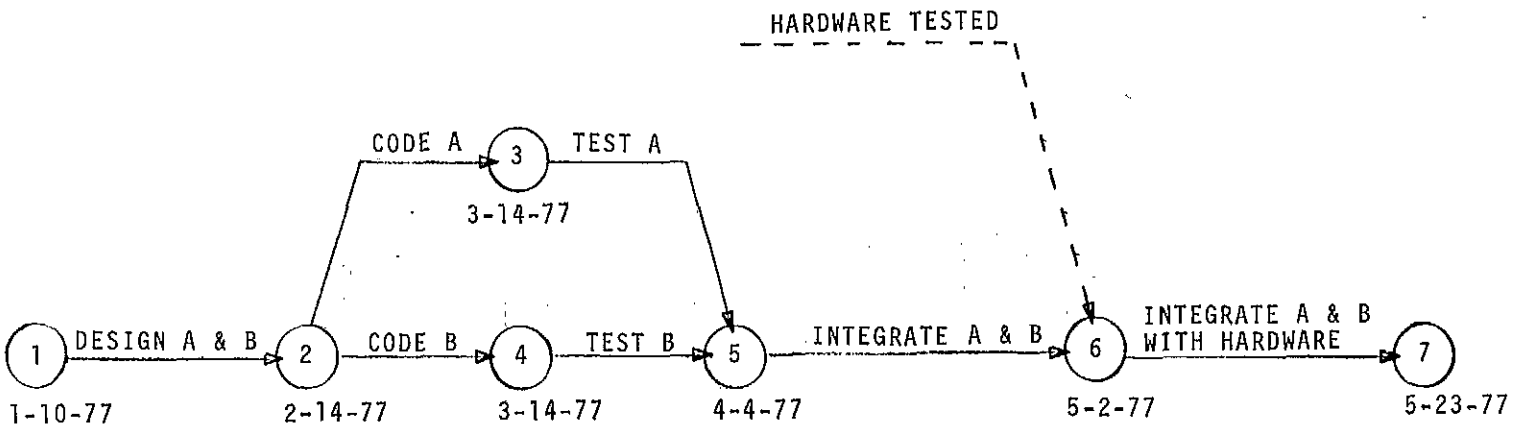


Figure 5. Software Network With Dates

utilization, the problem can be handled in an effective manner.

REFERENCES


1. Davis, E.J. and Jackson, H.M., "Management Overview" Bell System Technical Journal, 1975, Safeguard Supplement pp. 223-229.

2. Donelson, William, "Project Planning and Control," Datamation, Vol. 22, No. 6, June 1976, pp. 73-80.

3. Kuone, J.P., "A Means to Effective Computer Resource Utilization", Bell System Technical Journal, 1975, Safeguard Supplement, pp. 191-196.

4. Metzger, Phillip W., "Managing a Programming Project." Prentice-Hall, Inc., 1973.

5. Musa, J.D. and Woomer, F.N., Jr. "Software Project Management," Bell System Technical Journal, 1975, Safeguard Supplement, pp. 245-259.

6. O'Brien, James J., "CPM in Construction Management," McGraw-Hill, Inc., 1965.

7. Weinwurm, George F., "On the Management of Computer Programming," Princeton-Auerbach Pub., 1970.


ABOUT THE AUTHOR


*MR. DALE H. FAWCETT is a Senior Scientific Analyst Programmer in the Simulator Systems Engineering Department of Goodyear Aerospace Corporation. He joined Goodyear Aerospace in 1976 and is responsible for proposal work and International Research and Development (IR&D) projects. He has been working on a project to improve the software capability through better production and management techniques. His previous experience includes lead programmer for development of real-time airborne reconnaissance system. He has also worked extensively on Electronic Intelligence (ELINT) data reduction systems and with missile system engineering simulations. He received a B.S. in mathematics from Bucknell University and an M.S. in systems management from the University of Southern California.*