Chapter 9

Monitoring and control

OBJECTIVES

When you have completed this chapter you will be able to:

- monitor the progress of projects;
- assess the risk of slippage;
- visualize and assess the state of a project;
- revise targets to correct or counteract drift;
- control changes to a project's requirements.

9.1 Introduction

Once work schedules have been published and the project is under way, attention must be focused on ensuring progress. This requires monitoring of what is happening, comparison of actual achievement against the schedule and, where necessary, revision of plans and schedules to bring the project as far as possible back on target.

In earlier chapters we have stressed the importance of producing plans that can be monitored – for example, ensuring that activities have clearly defined and visible completion points. We will discuss how information about project progress is gathered and what actions must be taken to ensure a project meets its targets.

The final part of this chapter discusses how we can deal with changes that are imposed from outside – namely, changes in requirements.

9.2 Creating the framework

Exercising control over a project and ensuring that targets are met is a matter of regular monitoring, finding out what is happening, and comparing it with current targets. If there is a mismatch between the planned outcomes and the actual ones then either replanning is needed to bring the project back on target or the target will have to be revised. Figure 9.1 illustrates a model of the project control cycle and shows how, once the initial project plan has been published, project control is a continual process of monitoring progress against that plan and, where necessary, revising the plan to take account of deviations. It also illustrates the important steps that must be taken after completion of the project so that the experienced gained in any one project can feed into the planning stages of future projects, thus allowing us to learn from past mistakes.

See Chapter 11 for a discussion of software quality.

In practice we are normally concerned with departures from the plan in four dimensions – delays in meeting target dates, shortfalls in quality, inadequate functionality, and costs going over target. In this chapter we are mainly concerned with the first and last of these.

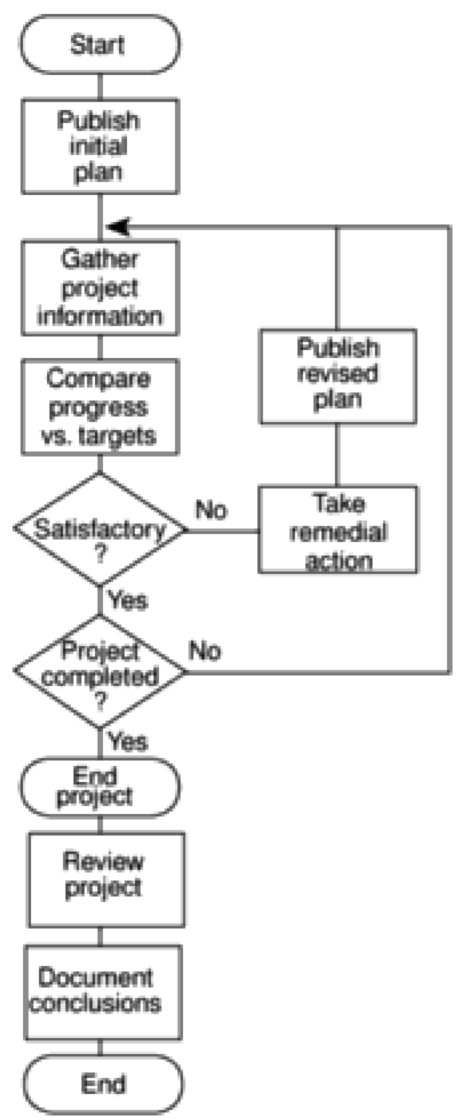


Figure 9.1 *The project control cycle.*

Responsibility

The overall responsibility for ensuring satisfactory progress on a project is often the role of the *project steering committee* or *Project Board*. Day-to-day responsibility will rest with the project manager and, in all but the smallest of projects, aspects of this can be delegated to team leaders.

Figure 9.2 illustrates the typical reporting structure found with medium and large projects. With small projects (employing around half a dozen or fewer staff) individual team members usually report directly to the project manager, but in most cases team leaders will collate reports on their section's progress and forward summaries to the project manager. These, in turn, will be incorporated into project-level reports for the steering committee and, via them or directly, progress reports for the client.

The concept of a reporting hierarchy was introduced in Chapter 1.

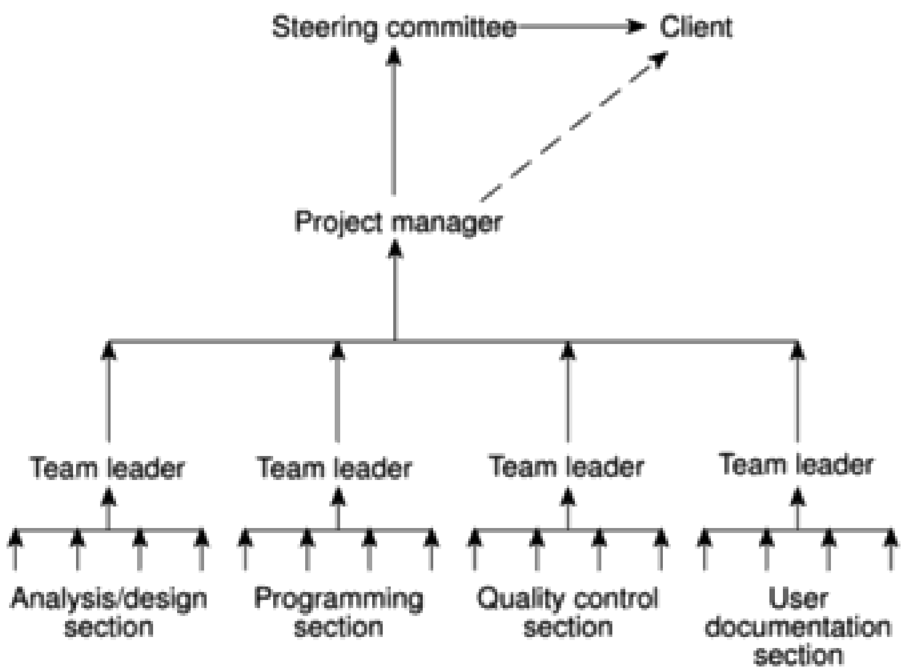


Figure 9.2 *Project reporting structures.*

In a PRINCE 2 environment, there is a Project Assurance function reporting to the Project Board and independent of the Project Manager.

Reporting may be oral or written, formal or informal, or regular or ad hoc and some examples of each type are given in Table 9.1. While any effective team leader or project manager will be in touch with team members and available to discuss problems, any such informal reporting of project progress must be complemented by formal reporting procedures – and it is those we are concerned with in this chapter.

Assessing progress

Progress assessment will normally be made on the basis of information collected and collated at regular intervals or when specific events occur. Wherever possible, this information will be objective and tangible – whether or not a particular report has been delivered, for example. However, such end-of-activity deliverables might

Table 9.1 *Categories of reporting*

Report type	*Examples*	*Comment*
Oral formal regular	weekly or monthly progress meetings	while reports may be oral formal written minutes should be kept
Oral formal ad hoc	end-of-stage review meetings	while largely oral, likely to receive and generate written reports
Written formal regular	job sheets, progress reports	normally weekly using forms
Written formal ad hoc	exception reports, change reports	
Oral informal ad hoc	canteen discussion, social interaction	often provides early warning; must be backed up by formal reporting

not occur sufficiently frequently throughout the life of the project. Here progress assessment will have to rely on the judgement of the team members who are carrying out the project activities.

Setting checkpoints

The PRINCE 2 standard described in Appendix A has its own terminology.

It is essential to set a series of checkpoints in the initial activity plan. Checkpoints may be:

- regular (monthly, for example);
- tied to specific events such as the production of a report or other deliverable.

Taking snap-shots

The frequency with which the a manager needs to receive information about progress will depend upon the size and degree of risk of the project or that part of the project under their control. Team leaders, for example, need to assess progress daily (particularly when employing inexperienced staff) whereas project managers may find weekly or monthly reporting appropriate. In general, the higher the level, the less frequent and less detailed the reporting needs to be.

Short, Monday morning team progress meetings are a common way of motivating staff to meet short term targets.

There are, however, strong arguments in favour of formal weekly collection of information from staff carrying out activities. Collecting data at the end of each week ensures that information is provided while memories are still relatively fresh and provides a mechanism for individuals to review and reflect upon their progress during the past few days.

Major, or project-level, progress reviews will generally take place at particular points during the life of a project – commonly known as *review points* or *control points*. PRINCE 2, for example, designates a series of checkpoints where the status of work in a project or for a team is reviewed. At the end of each project Stage, PRINCE 2 provides for an End Stage Assessment where an assessment of the project and consideration of its future are undertaken.

9.3 Collecting the data

As a rule, managers will try to break down long activities into more controllable tasks of one or two weeks duration. However, it will still be necessary to gather information about partially completed activities and, in particular, forecasts of how much work is left to be completed. It can be difficult to make such forecasts accurately.

Exercise 9.1

A software developer working on Amanda's project has written the first 250 lines of a Cobol program that is estimated to require 500 lines of code. Explain why it would be unreasonable to assume that the programming task is 50% complete.

How might you make a reasonable estimate of how near completion it might be?

Where there is a series of products, partial completion of activities is easier to estimate. Counting the number of record specifications or screen layouts produced, for example, can provide a reasonable measure of progress.

In some cases, intermediate products can be used as in-activity milestones. The first successful compilation of a Cobol program, for example, might be considered a milestone even though it is not the final product of the activity code and test.

Partial completion reporting

Many organizations use standard accounting systems with weekly time sheets to charge staff time to individual jobs. The staff time booked to a project indicates the work carried out and the charges to the project. It does not, however, tell the project manager what has been produced or whether tasks are on schedule.

It is therefore common to adapt or enhance existing accounting data collection systems to meet the needs of project control. Weekly time sheets, for example, are frequently adapted by breaking jobs down to activity level and requiring information about work done in addition to time spent. Figure 9.3 shows a typical example of such a report form, in this case requesting information about likely slippage of completion dates as well as estimates of completeness.

Asking for estimated completion times can be criticized on the grounds that frequent invitations to reconsider completion dates deflects attention away from the importance of the originally scheduled targets and can generate an ethos that it is acceptable for completion dates to slip.

Weekly timesheets are a valuable source of information about resources used.

They are often used to provide information about what has been achieved. However, requesting partial completion estimates where they cannot be obtained from objective measures encourages the 99% complete syndrome – tasks are reported as on time until 99% complete, and then stay at 99% complete until finished.

Time Sheet

Staff John Smith **Week ending** 26/3/99

Rechargeable hours

Project	Activity code	Description	Hours this week	% Complete	Scheduled completion	Estimated completion
P21	A243	Code mod A3	12	30	24/4/99	24/4/99
P34	B771	Document take-on	20	90	1/4/99	29/3/99

Total recharged hours	32

Non-rechargeable hours

Code	Description	Hours	Comment & authorization
Z99	day in lieu	8	Authorized by RB

Total non-rechargeable hours	8

Figure 9.3 *A weekly time sheet and progress review form.*

Risk reporting

One popular way of overcoming the objections to partial completion reporting is to avoid asking for estimated completion dates, but to ask instead for the team members' estimates of the likelihood of meeting the planned target date.

One way of doing this is the traffic-light method. This consists of the following steps:

There are a number of variations on the traffic-light technique. The version described here is in use in IBM and is described in Down, Coleman and Absolon, *Risk Management for Software Projects*, McGraw-Hill, 1994.

- identify the key (first level) elements for assessment in a piece of work;
- break these key elements into constituent elements (second level);
- assess each of the second level elements on the scale *green* for 'on target', *amber* for 'not on target but recoverable', and *red* for 'not on target and recoverable only with difficulty';
- review all the second level assessments to arrive at first level assessments;
- review first and second level assessments to produce an overall assessment.

For example, Amanda decides to use a version of the traffic-light method for reviewing activities on the IOE project. She breaks each activity into a number of component parts (deciding, in this case, that a further breakdown is unnecessary) and gets the team members to complete a return at the end of each week. Figure 9.4 illustrates Justin's completed assessment at the end of week 16.

Note that this form refers only to uncompleted activities. Justin would still need to report activity completions and the time spent on activities.

Activity Assessment Sheet

Staff Justin

Ref: IoE/P/13 **Activity: Code & test module C**

Week number	13	14	15	16	17	18	
Activity Summary	G	A	A	R			

Component							Comments
Screen handling procedures	G	A	A	G			
File update procedures	G	G	R	A			
Housekeeping procedures	G	G	G	A			
Compilation	G	G	G	R			
Test data runs	G	G	G	A			
Program documentation	G	G	A	R			

Figure 9.4 *A traffic-light assessment of IoE/P/13.*

Traffic-light assessment highlights only risk of non-achievement; it is not an attempt to estimate work done or to quantify expected delays.

Following completion of assessment forms for all activities, the project manager uses these as a basis for evaluating the overall status of the project. Any critical activity classified as amber or red will require further consideration and often leads to a revision of the project schedule. Non-critical activities are likely to be considered as a problem if they are classified as red, especially if all their float is likely to be consumed.

9.4 Visualizing progress

Having collected data about project progress, a manager needs some way of presenting that data to greatest effect. In this section, we look at some methods of presenting a picture of the project and its future. Some of these methods (such as Gantt charts) provide a static picture, a single snap-shot, whereas others (such as time-line charts) try to show how the project has progressed and changed through time.

The Gantt chart

One of the simplest and oldest techniques for tracking project progress is the Gantt chart. This is essentially an activity bar chart indicating scheduled activity dates and durations frequently augmented with activity floats. Reported progress is recorded on the chart (normally by shading activity bars) and a 'today cursor' provides an immediate visual indication of which activities are ahead or behind schedule. Figure 9.5 shows part of Amanda's Gantt chart as at the end of Tuesday of week 17. *Code & test module D* has been completed ahead of schedule and *code & test module A* appears also to be ahead of schedule. The coding and testing of the other two modules are behind schedule.

Henry Gantt (1861–1919) was an industrial engineer interested in the efficient organization of work.

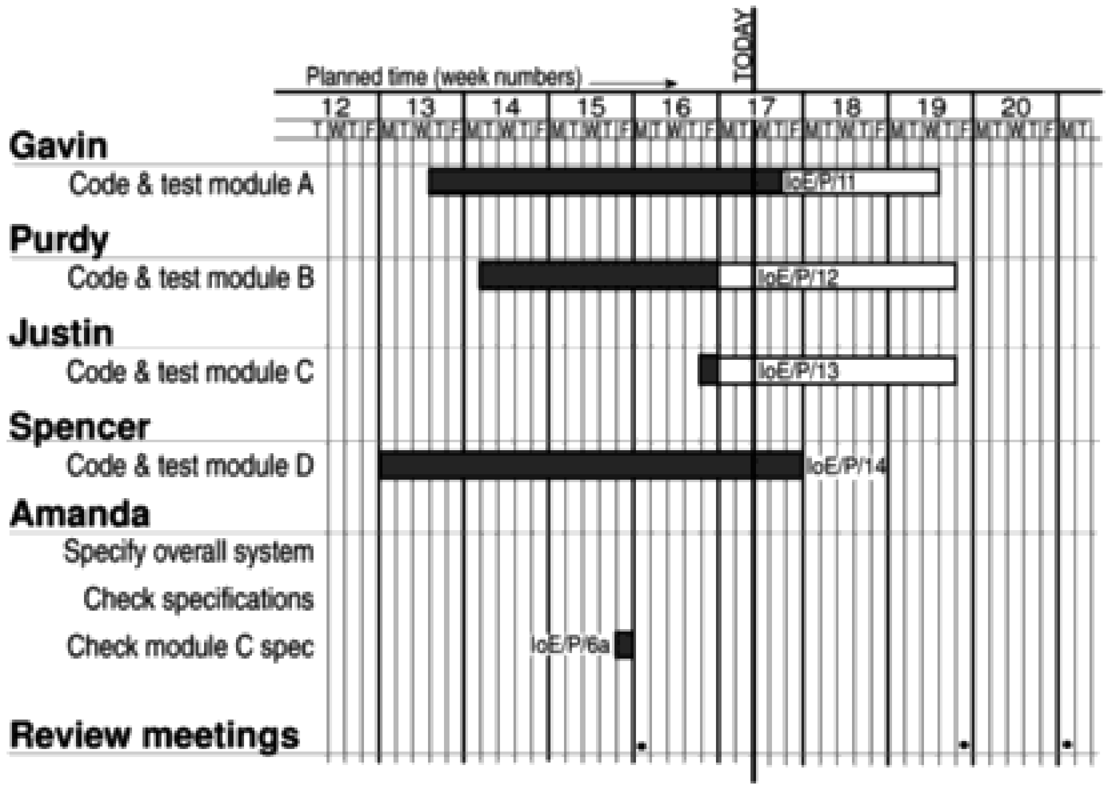


Figure 9.5 *Part of Amanda's Gantt chart with the 'today cursor' in week 17.*

The slip chart

A slip chart (Figure 9.6) is a very similar alternative favoured by some project managers who believe it provides a more striking visual indication of those activities that are not progressing to schedule – the more the slip line bends, the greater the variation from the plan. Additional slip lines are added at intervals and, as they build up, the project manager will gain an idea as to whether the project is improving (subsequent slip lines bend less) or not. A very jagged slip line indicates a need for rescheduling.

Ball charts

A somewhat more striking way of showing whether or not targets have been met is to use a ball chart as in Figure 9.7. In this version of the ball chart, the circles indicate start and completion points for activities. The circles initially contain the original scheduled dates. Whenever revisions are produced these are added as second dates in the appropriate circle until an activity is actually started or

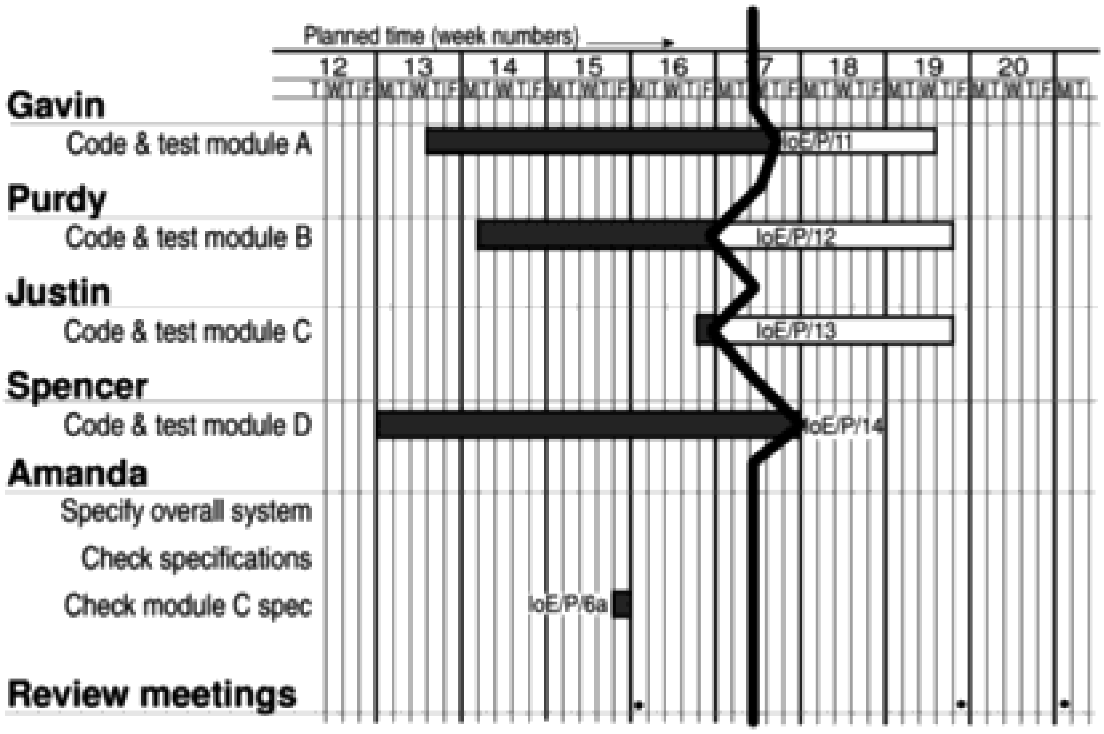


Figure 9.6 *The slip chart emphasizes the relative position of each activity.*

completed when the relevant date replaces the revised estimate (in bold italic in Figure 9.7). Circles will therefore contain only two dates, the original and most recent target dates, or the original and actual dates.

Where the actual start or finish date for an activity is later than the target date, the circle is coloured red (dark grey in Figure 9.7) – where an actual date is on time or earlier than the target then the circle is coloured green (light grey in Figure 9.7).

Such charts are frequently placed in a prominent position and the colour coded balls provide a constant reminder to the project team. Where more than one team is working in close proximity, such a highly visible record of achievement can encourage competitiveness between teams.

David Youll in *Making Software Development Visible*, John Wiley & Sons, 1990, describes a version of the ball chart using three sets of dates and part-coloured balls.

Another advantage of ball charts over Gantt and slip charts is that they are relatively easy to keep up to date – only the dates and possibly colours need to be changed, whereas the others need to be redrawn each time target dates are revised.

The timeline

One disadvantage of the charts described so far is that they do not show clearly the slippage of the project completion date through the life of the project. Knowing the current state of a project helps in revising plans to bring it back on target, but analysing and understanding trends helps to avoid slippage in future projects.

The timeline chart is a method of recording and displaying the way in which targets have changed throughout the duration of the project.

Figure 9.8 shows a timeline chart for Brigette's project at the end of the sixth week. Planned time is plotted along the horizontal axis and elapsed time down the vertical axis. The lines meandering down the chart represent scheduled activity

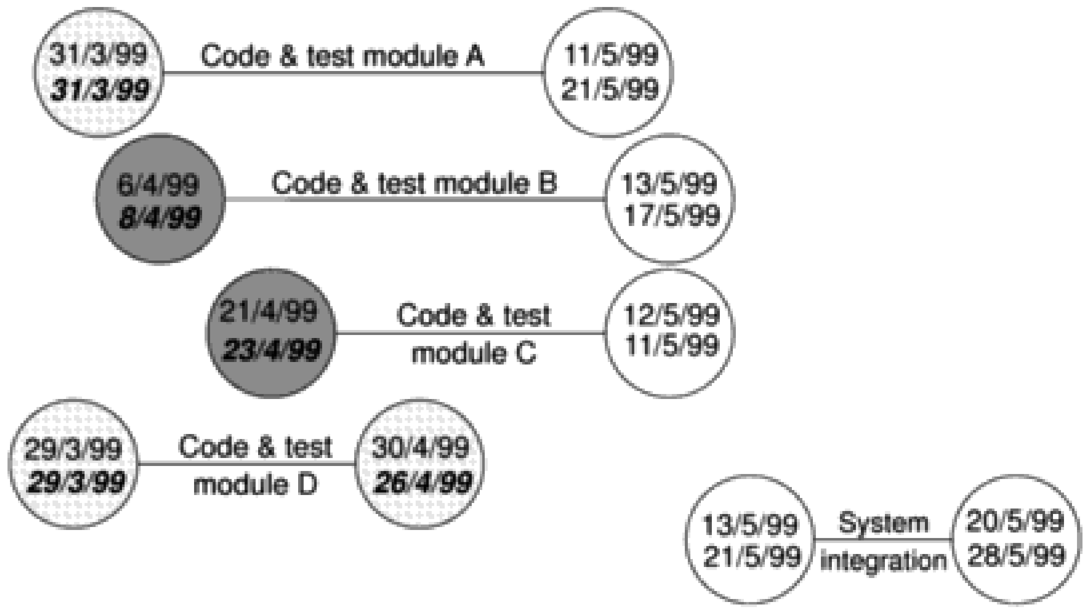


Figure 9.7 *The ball wall chart provides an incentive for meeting targets.*

completion dates – at the start of the project *analyse existing system* is scheduled to be completed by the Tuesday of week 3, *obtain user requirements* by Thursday of week 5, *issue tender*, the final activity, by Tuesday of week 9, and so on.

At the end of the first week Brigette reviews these target dates and leaves them as they are – lines are therefore drawn vertically downwards from the target dates to the end of week one on the actual time axis.

At the end of week two, Brigette decides that *obtain user requirements* will not be completed until Tuesday of week six – she therefore extends that activity line diagonally to reflect this. The other activity completion targets are also delayed correspondingly.

By the Tuesday of week three, *analyse existing system* is completed and Brigette puts a blob on the diagonal timeline to indicate that this has happened. At the end of week three she decides to keep to the existing targets.

At the end of week four she adds another three days to *draft tender* and *issue tender*.

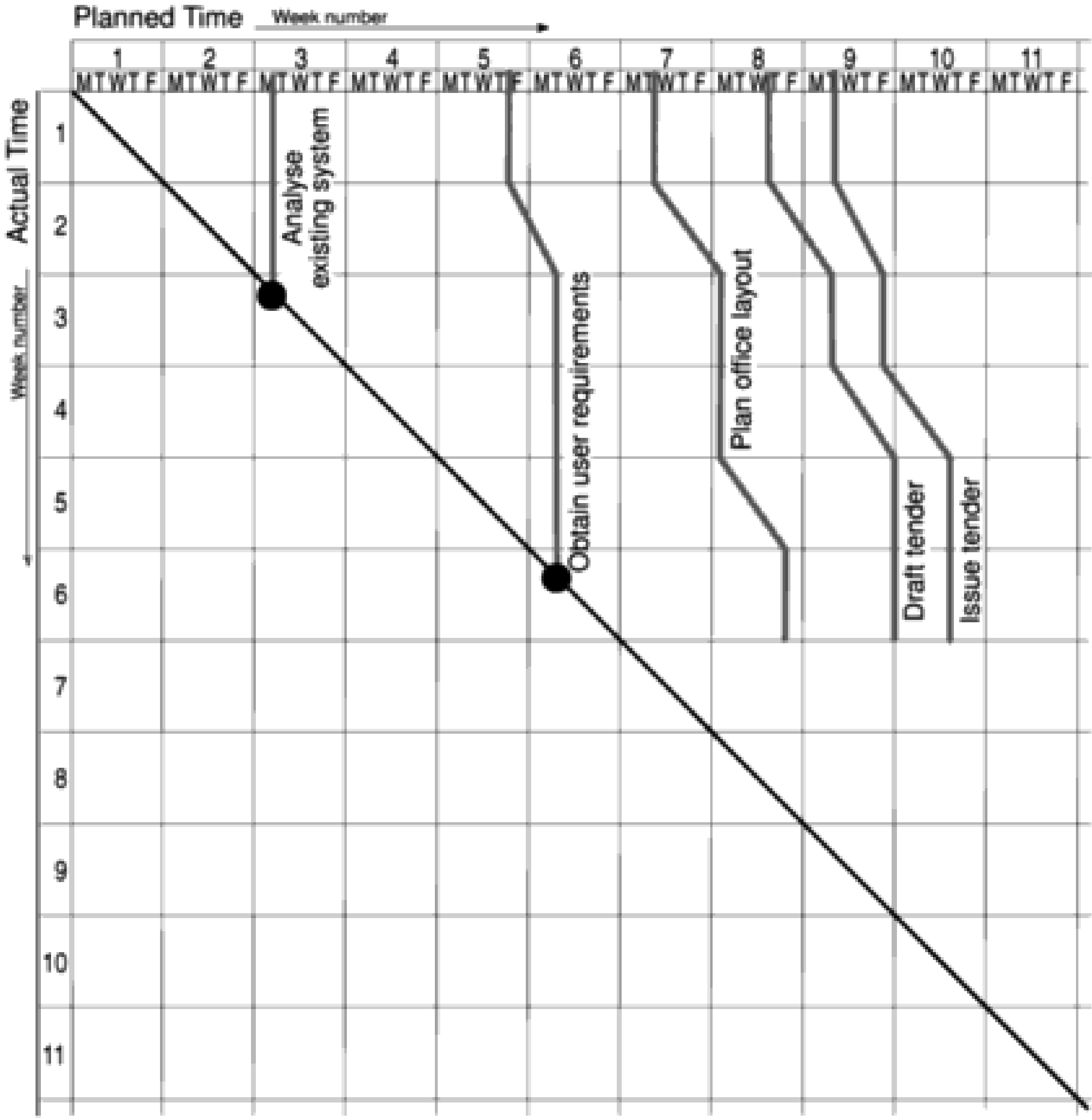
Note that, by the end of week six, two activities have been completed and three are still unfinished. Up to this point she has revised target dates on three occasions and the project as a whole is running seven days late.

Exercise 9.2

By the end of week 8 Brigette has completed planning the office layout but finds that drafting the tender is going to take one week longer that originally anticipated.

What will Brigette's timeline chart look like at the end of week 8?

If the rest of the project goes according to plan, what will Brigette's timeline chart look like when the project is completed?



Brigette's timeline chart contains only the critical activities for her project; ● indicates actual completion of an activity.

For the sake of clarity, the number of activities on a timeline chart must be limited. Using colour helps to distinguish activities, particularly where lines cross.

Figure 9.8 *Brigette's timeline chart at the end of week six.*

The timeline chart is useful both during the execution of a project and as part of the post-implementation review. Analysis of the timeline chart, and the reasons for the changes, can indicate failures in the estimation process or other errors that might, with that knowledge, be avoided in future.

9.5 Cost monitoring

Expenditure monitoring is an important component of project control. Not only in itself, but also because it provides an indication of the effort that has gone into (or at least been charged to) a project. A project might be on time but only because more money has been spent on activities than originally budgeted. A cumulative expenditure chart such as that shown in Figure 9.9 provides a simple method of comparing actual and planned expenditure. By itself it is not particularly meaningful – Figure 9.9 could, for example, illustrate a project that is running late or one that is on time but has shown substantial costs savings! We need to take account of the current status of the project activities before attempting to interpret the meaning of recorded expenditure.

Project costs may be monitored by a company's accounting system. By themselves, they provide little information about project status.

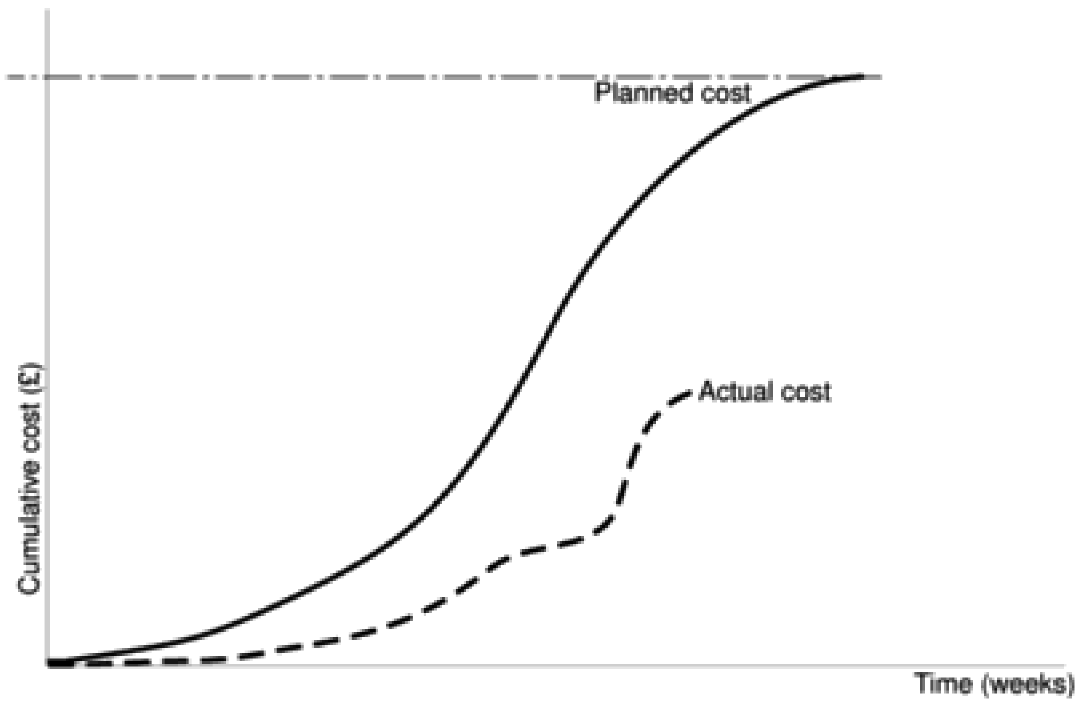


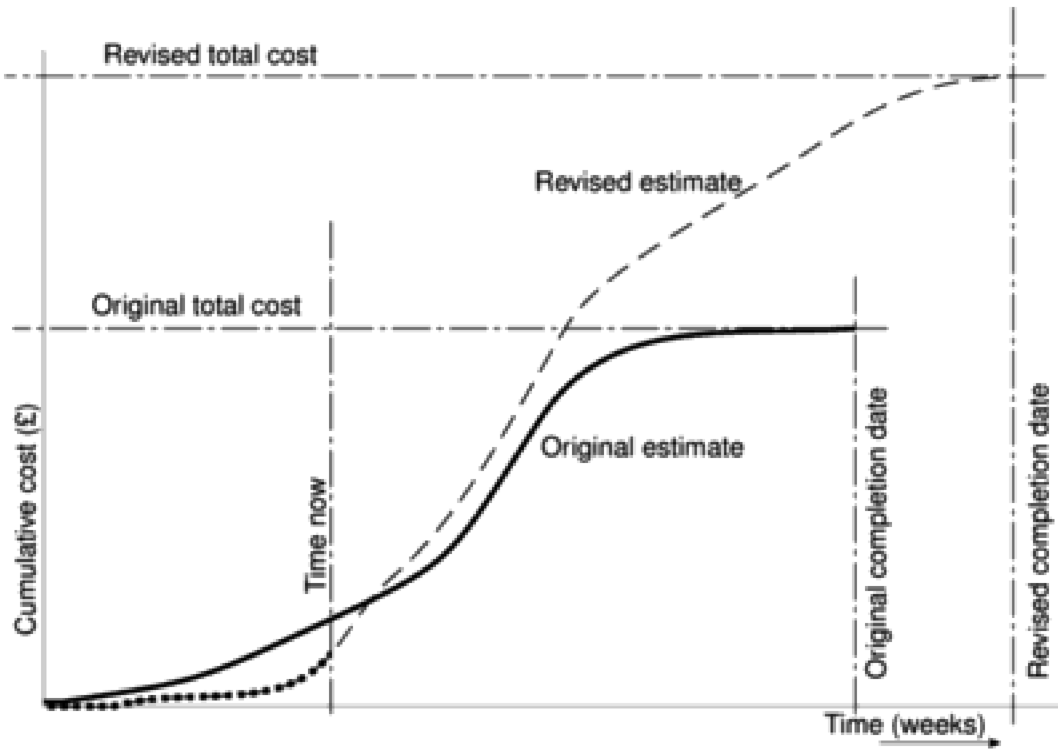
Figure 9.9 *Tracking cumulative expenditure.*

Cost charts become much more useful if we add projected future costs calculated by adding the estimated costs of uncompleted work to the costs already incurred. Where a computer-based planning tool is used, revision of cost schedules is generally provided automatically once actual expenditure has been recorded. Figure 9.10 illustrates the additional information available once the revised cost schedule is included – in this case it is apparent that the project is behind schedule and over budget.

9.6 Earned Value

Earned Value Analysis, also known as Budgeted Cost of Work Performed, is recommended by a number of agencies including the US and Australian departments of defence. It is also recommended in BS 6079.

Earned Value Analysis has gained in popularity in recent years and may be seen as a refinement of the cost monitoring discussed in the previous section. Earned Value Analysis is based on assigning a 'value' to each task or work package (as identified in the WBS) based on the original expenditure forecasts. The assigned value is the original budgeted cost for the item and is known as the *baseline budget* or *budgeted cost of work scheduled* (BCWS). A task that has not started is assigned the value zero and when it has been completed, it, and hence the project, is credited with the value of the task. The total value credited to a project at any point is known as the *earned value* or *budgeted cost of work performed* (BCWP) and this can be represented as a value or as a percentage of the BCWS.



Project costs augmented by project monitoring can be used to generate forecasts of future costs.

Figure 9.10 *The cumulative expenditure chart can also show revised estimates of cost and completion date.*

Where tasks have been started but are not yet complete, some consistent method of assigning an earned value must be applied. Common methods in software projects are:

- **the 0/100 technique** Where a task is assigned a value of zero until such time that it is completed when it is given a value of 100% of the budgeted value;
- **the 50/50 technique** Where a task is assigned a value of 50% of its value as soon as it is started and then given a value of 100% once it is complete;
- **the milestone technique** Where a task is given a value based on the achievement of milestones that have been assigned values as part of the original budget plan.

Of these, we prefer the 0/100 technique. The 50/50 technique can give a false sense of security by over-valuing the reporting of activity starts. The milestone technique might be appropriate for activities with a long duration estimate but, in such cases, it is better to break that activity into a number of smaller ones.

The baseline budget

The first stage in setting up an earned value analysis is to create the *baseline budget*. The baseline budget is based on the project plan and shows the forecast growth in earned value through time. Earned value may be measured in monetary values but, in the case of staff-intensive projects such as software development, it is common to measure earned value in person-hours or workdays. Amanda's baseline budget, based on the schedule shown in Figure 8.7, is shown in Table 9.2 and diagrammatically in Figure 9.11. Notice that she has based her baseline budget on workdays and is using the 0/100 technique for crediting earned value to the project.

Table 9.2 *Amanda's baseline budget calculation*

Task	*Budgeted workdays*	*Scheduled completion*	*Cumulative workdays*	*% cumulative earned value*
Specify overall system	34	34	34	14.35
Specify module B	15	49	64	27.00
Specify module D	15	49		
Specify module A	20	54	84	35.44
Check specifications	2	56	86	36.28
Design module D	4	60	90	37.97
Design module A	7	63	97	40.93
Design module B	6	66	103	43.46
Check module C spec	1	70	104	43.88
Specify module C	25	74	129	54.43
Design module C	4	79	133	56.12
Code & test module D	25	85	158	66.67
Code & test module A	30	93	188	79.32
Code & test module B	28	94	231	97.47
Code & test module C	15	94		
System integration	6	100	237	100.00

Amanda's project is not expected to be credited with any earned value until day 34, when the activity *specify overall system* is to be completed. This activity was forecast to consume 34 person-days and it will therefore be credited with 34 person-days earned value when it has been completed. The other steps in the baseline budget chart coincide with the scheduled completion dates of other activities.

Monitoring earned value

Having created the baseline budget, the next task is to monitor earned value as the project progresses. This is done by monitoring the completion of tasks (or activity starts and milestone achievements in the case of the other crediting techniques).

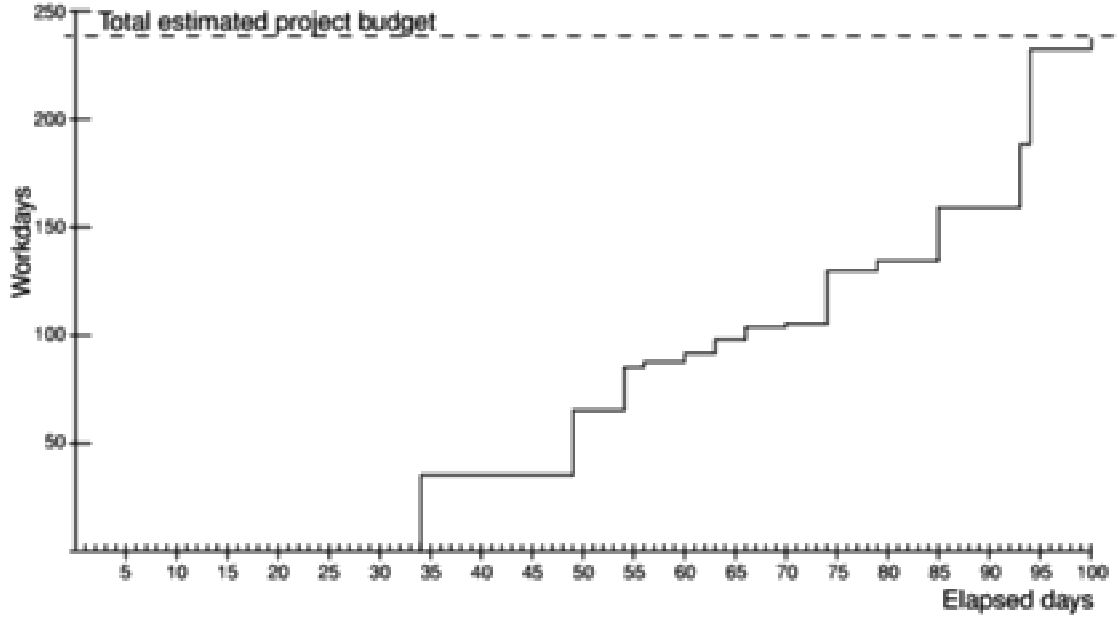


Figure 9.11 *Amanda's baseline budget.*

Exercise 9.3

Figure 9.12 shows Amanda's earned value analysis at the start of week 12 of the project. The earned value (BCWP) is clearly lagging behind the baseline budget, indicating that the project is behind schedule.

By studying Figure 9.12, can you tell exactly what has gone wrong with her project and what the consequences might be?

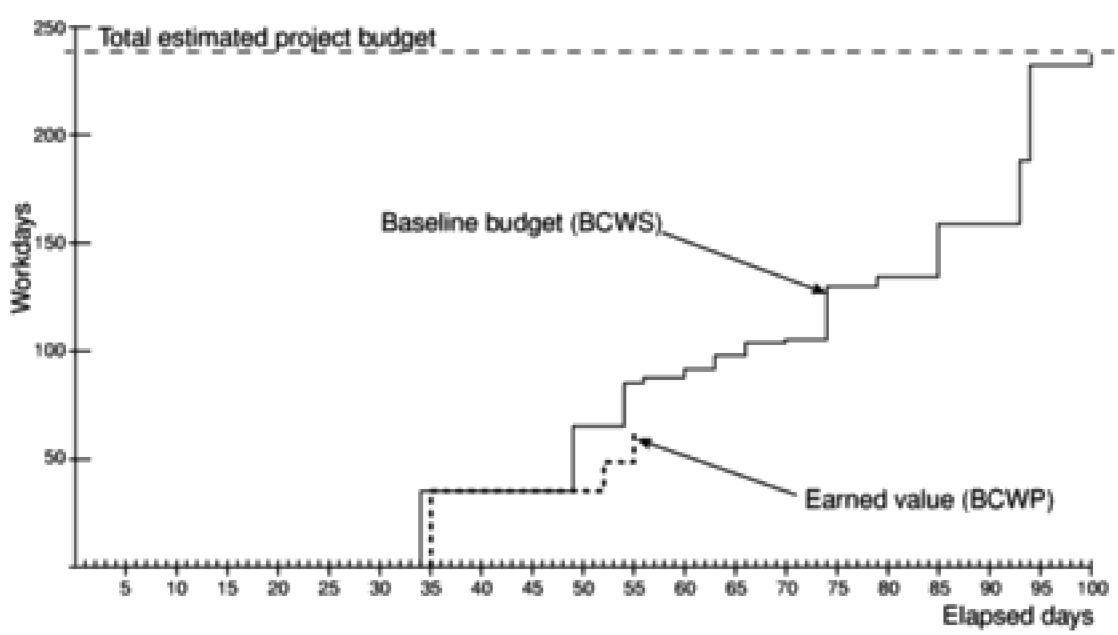


Figure 9.12 *Amanda's earned value analysis at week 12.*

As well as recording BCWP, the actual cost of each task can be collected as *actual cost of work performed*, ACWP. This is shown in Figure 9.13, which, in this case, records the values as percentages of the total budgeted cost.

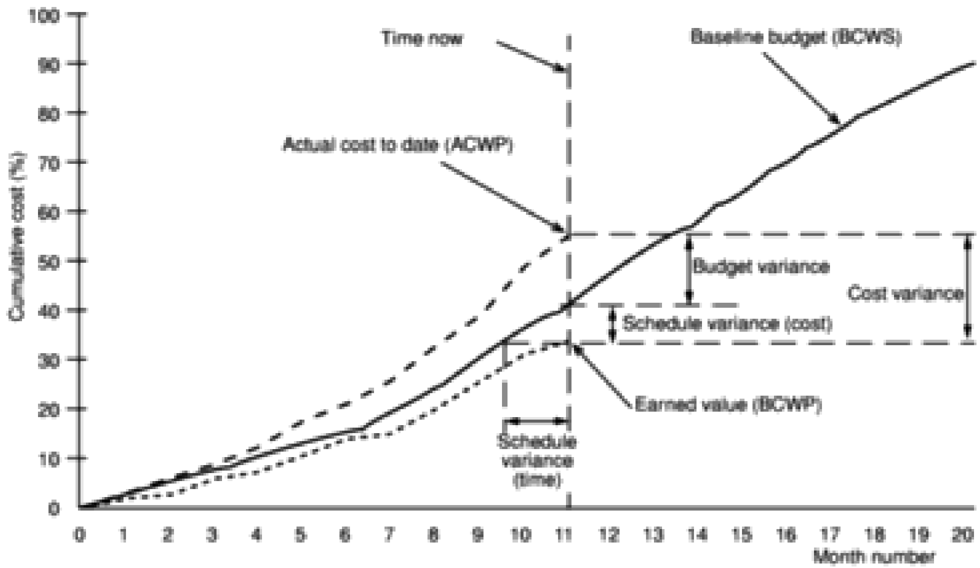


Figure 9.13 *An earned value tracking chart.*

Figure 9.13 also illustrates the following performance statistics, which can be shown directly or derived from the earned value chart.

Budget variance This can be calculated as ACWP – BCWS and indicates the degree to which actual costs differ from those planned.

Schedule variance The schedule variance is measured in cost terms as BCWP – BCWS and indicates the degree to which the value of completed work differs from that planned. Figure 9.13 also indicates the schedule variance in time, which indicates the degree to which the project is behind schedule.

Cost variance This is calculated as BCWP – ACWP and indicates the difference between the budgeted cost and the actual cost of completed work. It is also an indicator of the accuracy of the original cost estimates.

Performance ratios Two ratios are commonly tracked: the *cost performance index* (CPI = BCWP/ACWP) and the *schedule performance index* (SPI = BCWP/BCWS). They can be thought of as a 'value-for-money' indices. A value greater than one indicates that work is being completed better than planned whereas a value of less than one means that work is costing more than and/or proceeding more slowly than planned.

In the same way that the expenditure analysis in Figure 9.9 was augmented to show revised expenditure forecasts, we can augment the simple Earned Value tracking chart with forecasts as illustrated in Figure 9.14.

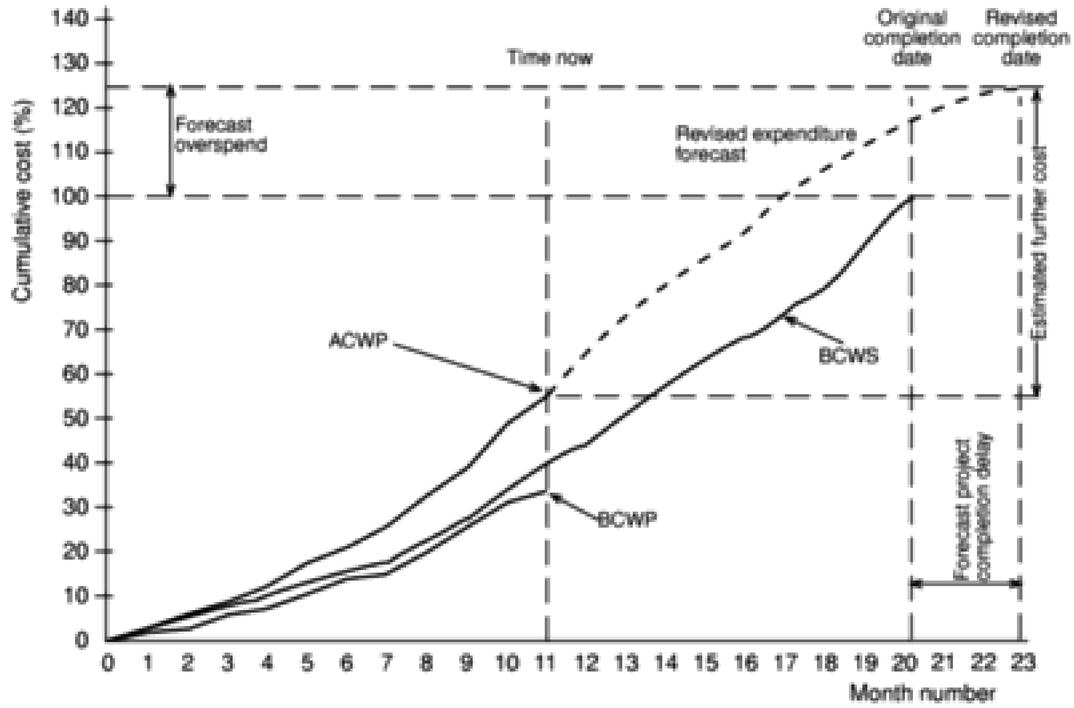


Figure 9.14 *An Earned Value chart with revised forecasts.*

Earned value analysis has not yet gained universal acceptance for use with software development projects, perhaps largely because of the attitude that, whereas a half-built house has a value reflected by the labour and materials that have been used, a half-completed software project has virtually no value at all. This is to misunderstand the purpose of earned value analysis, which, as we have seen, is a method for tracking what has been achieved on a project – measured in terms of the budgeted costs of completed tasks or products.

9.7 Prioritizing monitoring

So far we have assumed that all aspects of a project will receive equal treatment in terms of the degree of monitoring applied. We must not forget, however, that monitoring takes time and uses resources that might sometimes be put to better use!

In this section we list the priorities we might apply in deciding levels of monitoring.

- **Critical path activities** Any delay in an activity on the critical path will cause a delay in the completion date for the project. Critical path activities are therefore likely to have a very high priority for close monitoring.
- **Activities with no free float** A delay in any activity with no free float will delay at least some subsequent activities even though, if the delay is less than the total float, it might not delay the project completion date. These subsequent delays can have serious effects on our resource schedule as a delay in a

Free float is the amount of time an activity may be delayed without affecting any subsequent activity.

subsequent activity could mean that the resources for that activity will become unavailable before that activity is completed because they are committed elsewhere.

- **Activities with less than a specified float** If any activity has very little float it might use up this float before the regular activity monitoring brings the problem to the project manager's attention. It is common practice to monitor closely those activities with less than, say, one week free float.

PERT and the significance of activity duration variance was described in Chapter 7.

- **High risk activities** A set of high risk activities should have been identified as part of the initial risk profiling exercise. If we are using the PERT three-estimate approach we will designate as high risk those activities that have a high estimated duration variance. These activities will be given close attention because they are most likely to overrun or overspend.

- **Activities using critical resources** Activities can be critical because they are very expensive (as in the case of specialized contract programmers). Staff or other resources might be available only for a limited period, especially if they are controlled outside the project team. In any event, an activity that demands a critical resource requires a high level of monitoring.

9.8 Getting the project back to target

A contingency plan should, of course, already exist as a result of the risk analysis methods described in Chapter 7.

Almost any project will, at one time or another, be subject to delays and unexpected events. One of the tasks of the project manager is to recognize when this is happening (or, if possible, about to happen) and, with the minimum delay and disruption to the project team, attempt to mitigate the effects of the problem. In most cases, the project manager tries to ensure that the scheduled project end date remains unaffected. This can be done by shortening remaining activity durations or shortening the overall duration of the remaining project in the ways described in the next section

The schedule is not sacrosanct – it is a plan that should be adhered to so long as it is relevant and cost-effective.

It should be remembered, however, that this might not always be the most appropriate response to disruptions to a plan. There is little point in spending considerable sums in overtime payments in order to speed up a project if the customer is not overly concerned with the delivery date and there is no other valuable work for the team members once this project is completed.

There are two main strategies to consider when drawing up plans to bring a project back on target – shortening the critical path or altering the activity precedence requirements.

Shorten the critical path

The overall duration of a project is determined by the current critical path, so speeding up non-critical path activities will not bring forward a project completion date.

Extolling staff to 'work harder' might have some effect, although frequently a more positive form of action is required, such as increasing the resources available for some critical activity. Fact-finding, for example, might be speeded up by allocating an additional analyst to interviewing users. It is unlikely, however, that the coding of a small module would be shortened by allocating an additional programmer – indeed, it might be counterproductive because of the additional time needed organizing and allocating tasks and communicating.

Resource levels can be increased by making them available for longer. Thus, staff might be asked to work overtime for the duration of an activity and computing resources might be made available at times (such as evenings and week-ends) when they might otherwise be inaccessible.

Where these do not provide a sufficient solution, the project manager might consider allocating more efficient resources to activities on the critical path or swapping resources between critical and non-critical activities. This will be particularly appropriate with staff – an experienced programmer should be significantly more productive than a more junior member of the team.

Time/cost trade-off: there is a general rule that timescales can be shortened by buying more (or more expensive) resources; sometimes this is true.

By such means we can attempt to shorten the timescale for critical activities until such time as either we have brought the project back to schedule or further efforts prove unproductive or not cost-effective. Remember, however, that shortening a critical path often causes some other path, or paths, to become critical (see Section 6.16).

Reconsider the precedence requirements

If attempting to shorten critical activities proves insufficient, the next step is to consider the constraints by which some activities have to be deferred pending completion of others. The original project network would most probably have been drawn up assuming 'ideal' conditions and 'normal' working practices. It might be that, to avoid the project delivering late, it is now worth questioning whether as yet unstarted activities really do have to await the completion of others. It might, in a particular organization, be 'normal' to complete system testing before commencing user training. In order to avoid late completion of a project it might, however, be considered acceptable to alter 'normal' practice and start training earlier.

One way to overcome precedence constraints is to subdivide an activity into a component that can start immediately and one that is still constrained as before. For example, a user handbook can be drawn up in a draft form from the system specification and then be revised later to take account of subsequent changes.

If we do decide to alter the precedence requirements in such a way, it is clearly important to be aware that quality might be compromised and to make a considered decision to compromise quality where needed. It is equally important to assess the degree to which changes in work practices increase risk. It is possible, for example, to start coding a module before its design has been completed. It would normally, however, be considered foolhardy to do so since, as well as compromising quality, it would increase the risk of having to redo some of the coding once the final design had been completed and thus delay the project even further.

9.9 Change control

So far in this chapter, we have assumed that the nature of the tasks to be carried out has not changed. A project leader like Amanda or Brigette might find, however, that requirements are modified because of changing circumstances or because the users get a clearer idea of what is really needed. The payroll system that Brigette is implementing might, for instance, need to be adjusted if the staffing structure at the college is reorganized.

Other, internal, changes will crop up. Amanda might find that there are inconsistencies in the program specifications that become apparent only when the programs are coded, and these would result in amendments to the specifications.

Careful control of these changes is needed because an alteration in one document often implies changes to other documents and the system products based on that document. The Product Flow Diagrams that have been explained in Chapter 2 indicate relationships between the products of a project where this is the case.

Exercise 9.4

A change in a program specification will normally be carried through into changes to the program design and then changed code. What other products might need to be modified?

Configuration librarian's role

BS EN ISO 9001:1994 (formerly BS 5750) requires that a formal change control procedure be in place.

Control of changes and documentation ought to be the responsibility of someone who may variously be named the Configuration Librarian, the Configuration Manager or Project Librarian. Among this person's duties would be:

- the identification of all items that are subject to change control;
- the establishment and maintenance of a central repository of the master copies of all project documentation and software products;
- the setting up and running of a formal set of procedures to deal with changes;
- the maintenance of records of who has access to which library items and the status of each library item (e.g. whether under development, under test or released).

It will be recalled that it was suggested that the setting up of change control procedures might be one of the first things the Brigette might want to do at Brightmouth College.

Change control procedures

A simple change control procedure for operational systems might have the following steps.

1. One or more users might perceive a need for a modification to a system and ask for a change request to be passed to the development staff.
2. The user management consider the change request and if they approve it pass it to the development management.
3. The development management delegate a member of staff to look at the request and to report on the practicality and cost of carrying out the change. They would, as part of this, assess the products that would be affected by the change.
4. The development management report back to the user management on the findings and the user management decide whether, in view of the cost quoted, they wish to go ahead.
5. One or more developers are authorized to take copies of the master products that are to be modified.
6. The copies are modified. In the case of software components this would involve modifying the code and recompiling and testing it.
7. When the development of new versions of the product has been completed the user management will be notified and copies of the software will be released for user acceptance testing.
8. When the user is satisfied that the products are adequate they will authorize their operational release. The master copies of configuration items will be replaced.

Exercise 9.5

The above steps relate to changes to operational systems. How could they be modified to deal with systems under development?

Changes in scope of a system

A common occurrence with IS development projects is for the size of the system gradually to increase. One cause of this is changes to requirements that are requested by users.

This is sometimes called scope creep.

Exercise 9.6

Think of other reasons why there is a tendency for scope creep.

The scope of a project needs to be carefully monitored and controlled. One way is to re-estimate the system size in terms of SLOC or function points at key milestones.

9.10 Conclusions

In this chapter we have discussed the requirements for the continual monitoring of projects and the need for making progress visible. Among the important points to emerge were:

- planning is pointless unless the execution of the plan is monitored;
- activities that are too long need to be subdivided to make them more controllable;
- ideally, progress should be measured through the delivery of project products;
- progress needs to be shown in a visually striking way, such as through ball charts, in order to communicate information effectively;
- costs need to be monitored as well as elapsed time;
- delayed projects can often be brought back on track by shortening activity times on the critical path or by relaxing some of the precedence constraints.

9.11 Further exercises

1. Take a look at Amanda's project schedule shown in Figure 8.7. Identify those activities scheduled to last more than three weeks and describe how she might monitor progress on each of them on a fortnightly or weekly basis.
2. Amanda's Gantt chart at the end of week 17 (Figure 9.5) indicates that two activities are running late. What effect might this have on the rest of the project? How might Amanda mitigate the effects of this delay?
3. Table 9.2 illustrates Amanda's earned value calculations based on workdays. Revise the table using monetary values based on the cost figures that you used in Exercise 8.5. Think carefully about how to handle the costs of Amanda as project manager and the recovered overheads and justify your decisions about how you treat them.
4. If you have access to project planning software, investigate the extent to which it offers support for earned value analysis. If it does not do so directly, investigate ways in which it would help you to generate a baseline budget (BCWS) and track the earned value (BCWP).
5. Describe a set of change control procedures that would be appropriate for Brigette to implement at Brightmouth College.