CHAPTER 3

An overview of project planning

OBJECTIVES

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

- approach project planning in an organized step-by-step manner;
- see where the techniques described in other chapters fit into an overall planning approach;
- repeat the planning process in more detail for sets of activities within a project as the time comes to execute them.

3.1 Introduction to Step Wise project planning

This chapter describes a framework of basic steps in project planning upon which the following chapters build. Many different techniques can be used in project planning and this chapter gives an overview of the points at which these techniques can be applied during project planning. Chapter 4 will illustrate how different projects may need different technical approaches, but the overall framework should always apply to the planning process.

The OGC was previously the CCTA (Central Computing and Telecommunications Agency).

The framework described is called the Step Wise method to help to distinguish it from other methods such as PRINCE2. PRINCE2 is a set of project management standards that were originally sponsored by what is now the Office of Government Commerce (OGC) for use on British government ICT and business change projects. The standards are now also widely used on non-government projects in the United Kingdom. Step Wise

Appendix A adds some further details about the PRINCE2 approach.

should be compatible with PRINCE2. It should be noted, however, that Step Wise covers only the planning stages of a project and not monitoring and control.

In order to illustrate the Step Wise approach and how it might have to be adapted to deal with different circumstances, two parallel examples are used. Let us assume that there are two former Computing and Information Systems students who now have several years of software development experience under their belts.

CASE STUDY EXAMPLE A: A BRIGHTMOUTH COLLEGE PAYROLL

Brigette has been working for the Management Services department of a local authority when she sees an advertisement for the position of Information Systems Development Officer at Brightmouth College. She is attracted to the idea of being her own boss, working in a relatively small organization and helping them to set up appropriate information systems from scratch. She applies for the job and gets it. One of the first tasks that confronts her is the implementation of independent payroll processing. (This scenario has already been used as the basis of some examples in Chapter 1.)

CASE STUDY EXAMPLE B: INTERNATIONAL OFFICE EQUIPMENT ANNUAL MAINTENANCE CONTRACTS

Amanda works for International Office Equipment (IOE), which assembles, supplies, installs and services various items of high-technology office equipment. An expanding area of their work is the maintenance of ICT equipment. They have now started to undertake maintenance of equipment of which they were not the original suppliers. An existing application built by the in-house ICT department allows sales staff to input and generate invoices for completed work. A large organization might have to call out IOE several times during a month to deal with problems with equipment. Each month a batch run of the system generates monthly statements for customers so that only one payment a month needs to be made. The management of IOE would like to provide a service where for a single annual payment customers would get free servicing and problem resolution for a pre-specified set of equipment. Amanda has been given her first project management role, the task of implementing this extension to the IOE maintenance jobs billing system.

The enhanced application will need a means of recording the details of the items of equipment to be covered by a customer's annual maintenance contract. The annual fee will depend on the numbers of each type of equipment item that is to be covered. Even though the jobs done under this contract will not be charged for, the work will be recorded to allow for an analysis of costs and the profitability of each customer and each type of equipment. This will provide information which will allow IOE to set future contract prices at an optimally profitable level. At the moment, job details are only recorded after job completion so that invoices can be generated. The new system will allow a central coordinator to allocate jobs to engineers and the system to notify engineers of urgent jobs automatically via their mobile phones.

In Table 3.1 we outline the general approach that might be taken to planning these projects. Figure 3.1 provides an outline of the main planning activities. Steps 1 and 2 'Identify project scope and objectives' and 'Identify project infrastructure' could be tackled in parallel in some cases. Steps 5 and 6 will need to be repeated for each activity in the project.

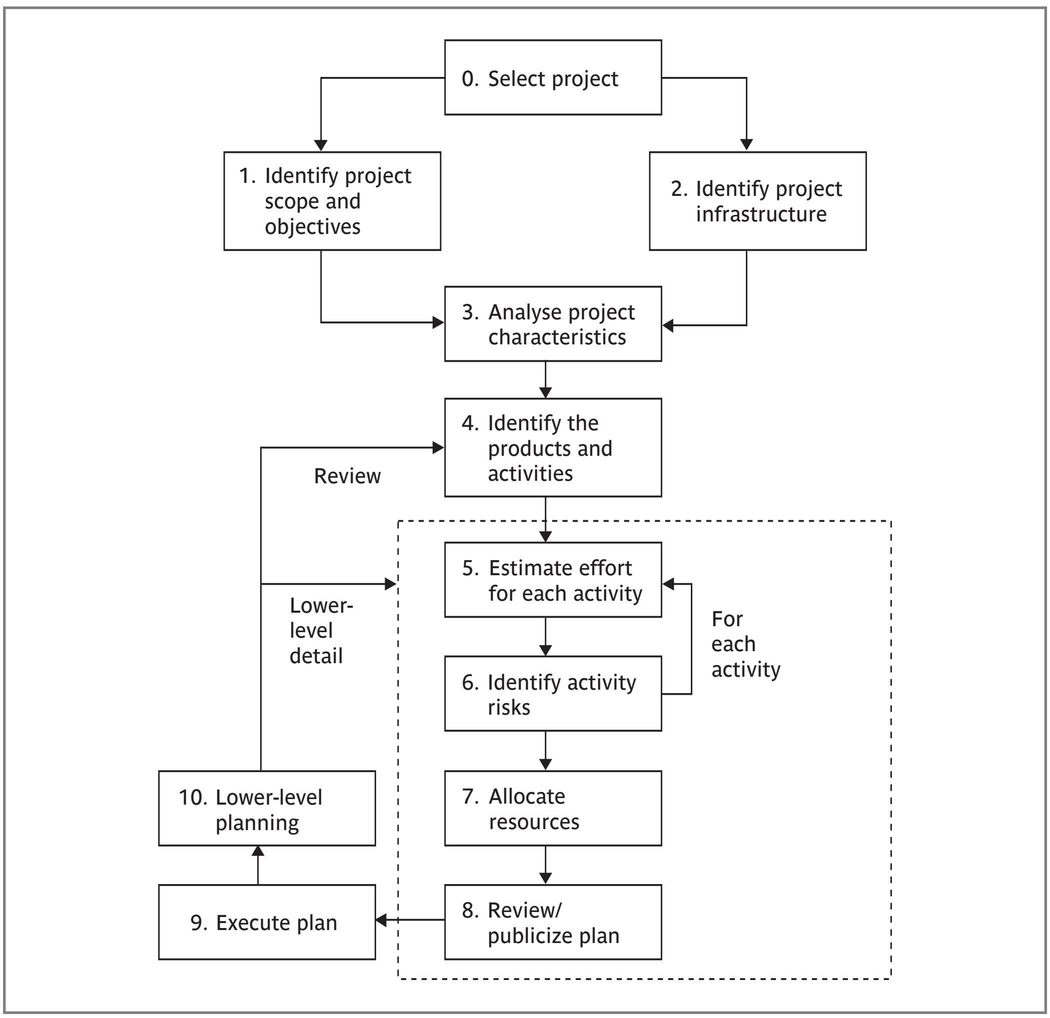

FIGURE 3.1 An overview of Step Wise

Step	Activities within step
0	Select project
1	Identify project scope and objectives
	1.1 Identify objectives and measures of effectiveness in meeting them
	1.2 Establish a project authority
	1.3 Identify stakeholders
	1.4 Modify objectives in the light of stakeholder analysis
	1.5 Establish methods of communication with all parties
2	Identify project infrastructure
	2.1 Establish relationship between project and strategic planning
	2.2 Identify installation standards and procedures
	2.3 Identify project team organization
3	Analyse project characteristics
	3.1 Distinguish the project as either objective- or product-driven
	3.2 Analyse other project characteristics
	3.3 Identify high-level project risks
	3.4 Take into account user requirements concerning implementation
	3.5 Select general life-cycle approach
	3.6 Review overall resource estimates
4	Identify project products and activities
	4.1 Identify and describe project products (including quality criteria)
	4.2 Document generic product flows
	4.3 Recognize product instances
	4.4 Produce ideal activity network
	4.5 Modify ideal to take into account need for stages and checkpoints
5	Estimate effort for each activity
	5.1 Carry out bottom-up estimates
	5.2 Revise plan to create controllable activities
6	Identify activity risks
	6.1 Identify and quantify activity-based risks
	6.2 Plan risk reduction and contingency measures where appropriate
	6.3 Adjust plans and estimates to take account of risks
7	Allocate resources
	7.1 Identify and allocate resources
	7.2 Revise plans and estimates to take account of resource constraints
8	Review/publicize plan
	8.1 Review quality aspects of project plan
	8.2 Document plans and obtain agreement
9/10	Execute plan/lower levels of planning
	This may require the reiteration of the planning process at a lower level

TABLE 3.1 An outline of Step Wise planning activities

A major principle of project planning is to plan in outline first and then in more detail as the time to carry out an activity approaches. Hence the lists of products and activities that are the result of Step 4 will be reviewed when the tasks connected with a particular phase of a project are considered in more detail. This will be followed by a more detailed iteration of Steps 5 to 8 for the phase under consideration.

3.2 Step 0: Select project

Chapter 2 has already discussed these issues in some detail.

This is called Step 0 because in a way it is outside the main project planning process. Proposed projects do not appear out of thin air – some process must decide to initiate this project rather than some other. While a feasibility study might suggest that there is a business case for the project, it would still need to be established that it should have priority over other projects. This evaluation of the merits of projects could be part of project portfolio management.

3.3 Step 1: Identify project scope and objectives

The activities in this step ensure that all the parties to the project agree on the objectives and are committed to the success of the project. We have already looked at the importance of the correct definition of objectives in Chapter 1.

Step 1.1: Identify objectives and practical measures of the effectiveness in meeting those objectives

CASE STUDY EXAMPLES: PROJECT OBJECTIVES

The project objectives for the Brightmouth College payroll project have already been discussed in Exercise 1.8.

Amanda at IOE has the objectives clearly laid down for her in the recommendations of a business case report which have been accepted by IOE management. The main objectives are to allow:

- details of annual maintenance contracts to be recorded;
- details of maintenance work covered by these contracts to be recorded;
- analysis of costs to be carried out so that the optimal level of maintenance contract fees may be identified;
- recording of job requests and notification of jobs to engineers via mobile phones.

Other objectives are laid down that refer to expected timescales and the resources that might be used.

Step 1.2: Establish a project authority

We have already noted in Chapter 1 that a single overall project authority needs to be established so that there is unity of purpose among all those concerned.

Step 1.3: Stakeholder analysis – identify all stakeholders in the project and their interests

Recall that this was the basis of a discussion in Chapter 1. Essentially all the parties who have an interest in the project need to be identified. In that chapter we listed as an example the stakeholders in the Brightmouth College payroll project.

EXERCISE 3.1

What important stakeholders outside the IOE organization might be considered in the case of the IOE annual maintenance contracts system?

CASE STUDY EXAMPLES: PROJECT AUTHORITIES

Throughout the text we use capitalized initial letters to indicate a term that has a precise meaning in the PRINCE2 standards, e.g. Project Board.

Amanda finds that her manager and the main user management have already set up a Project Board which will have overall direction of the project. She is a little concerned as the equipment maintenance staff are organized with different sections dealing with different types of equipment. This means that a customer could have work done by several different sections. Not all the sections are represented on the Project Board and Amanda is aware that there are some differences of opinion between some sections. It is left to the user representatives on the board to resolve those differences and to present an agreed policy to the systems developers.

Brigette finds that effectively she has two different clients for the payroll system: the finance and human resources departments. To help resolve conflicts, it is agreed that the managers of both departments should attend a monthly meeting with the vice-principal which Brigette has arranged in order to steer the project.

Step 1.4: Modify objectives in the light of stakeholder analysis

Compare this with the 'Theory W' of Boehm and Ross mentioned in Chapter 1.

In order to gain the full cooperation of all concerned, it might be necessary to modify the project objectives. This could mean adding new features to the system which give a benefit to some stakeholders as a means of assuring their commitment to the project. This is potentially dangerous as the system size may be increased and the original objectives obscured. Because of these dangers, it is suggested that this process be done consciously and in a controlled manner.

CASE STUDY EXAMPLES: MODIFIED PROJECT OBJECTIVES

The IOE maintenance staff are to be given the extra task of entering data about completed jobs. As no customer charges are generated by visits under annual maintenance contracts, engineers may feel that completing cost details is unnecessary bureaucracy, and start to do this in a careless and inaccurate manner. To give some benefit to the engineers, the system is to be extended to reorder spare parts automatically when required. It will also automatically capture timesheet details which previously had to be completed by hand.

At Brightmouth College, the human resources department has a lot of work preparing payroll details for finance. It would be tactful to agree to produce some management information reports for human resources from the payroll details held on the computer.

Step 1.5: Establish methods of communication with all parties

For internal staff this should be fairly straightforward, but a project leader implementing a payroll system would need to find a contact point with BACS (Bankers Automated Clearing Scheme), for instance. This step could lead to the first draft of a *communications plan* – to read more about these, see Chapter 12.

3.4 Step 2: Identify project infrastructure

Projects are never carried out in a vacuum. There is usually some kind of existing infrastructure into which the project must fit. Where project managers are new to the organization, they must find out the precise nature of this infrastructure. This could be the case where the project manager works for an outside organization carrying out the work for a client.

B. Iyer and R. Gottlieb (2004) 'The Four-Domain Architecture: an approach to support enterprise architecture design' *IBM Systems Journal* 43(3) 587–97 provides a good introduction to enterprise architecture concepts.

Step 2.1: Identify relationship between the project and strategic planning

We saw in Chapter 2 how project portfolio management supported the selection of the projects to be carried out by an organization. Also, how programame management can ensure that a group of projects contribute to a common organizational strategy. There is also a technical framework within which the proposed new systems are to fit. Hardware and software standards, for example, are needed so that various systems can communicate with each other. These technical strategic decisions should be documented as part of an *enterprise architecture* process. Compliance with the enterprise architecture should ensure that successive ICT projects create software and other components compatible with those created by previous projects and also with the existing hardware and software platforms.

CASE STUDY EXAMPLES: ROLE OF EXISTING STRATEGIC PLANS

Amanda finds at IOE that there is a well-defined rolling strategic plan which has identified her annual maintenance contracts subsystem as an important required development. Because it is an extension of an existing system, the hardware and software platforms upon which the application are to run are dictated.

Enterprise Resource Planning (ERP) systems are integrated software applications usually acquired as off-the-shelf packages that require considerable customization. They integrate all the standard financial and trading applications common to most businesses.

Brigette at Brightmouth College finds that there is an overall College strategic plan which describes new courses to be developed, and so on, and mentions in passing the need for 'appropriate administrative procedures' to be in place. There is a recommendation in a consultant's report concerning the implications of financial autonomy that independent payroll processing be implemented as just one module in an ERP system which would cover all the college's financial processing needs. Although the college has quite a lot of ICT equipment for teaching purposes, there is no machine set aside for payroll processing and the intention is that the hardware to run the payroll will be acquired at the same time as the software.

Step 2.2: Identify installation standards and procedures

See discussion of the ISO/IEC 12207 standard in Chapter 1.

Any organization that develops software should define their development procedures. As a minimum, the normal stages in the software life cycle to be carried out should be documented along with the products created at each stage.

Change control and *configuration management* standards should be in place to ensure that changes to requirements are implemented in a safe and orderly way.

See Chapter 9 on monitoring and control.

The procedural standards may lay down the quality checks that need to be done at each point of the project life cycle or these may be documented in a separate *quality standards and procedures* manual.

The organization, as part of its monitoring and control policy, may have a *measurement programme* in place which dictates that certain statistics have to be collected at various stages of a project.

Finally the project manager should be aware of any *project planning and control standards*. These will relate to how the project is controlled: for example, the way that the hours spent by team members on individual tasks are recorded on timesheets.

CASE STUDY EXAMPLES: IDENTIFYING STANDARDS

Amanda at IOE finds that there is a very weighty volume of development standards which, among other things, specifies that a specific structured systems analysis and design method be used. She finds that a separate document has been prepared which lays down quality procedures. This specifies when the reviews of work will be carried out and describes detailed procedures governing how the reviews are to be done. Amanda also finds a set of project management guidelines modelled on PRINCE2.

Brigette finds no documents of the nature that Amanda found at IOE except for some handouts for students that have been produced by different lecturers at different times and which seem to contradict each other.

As a stop-gap measure, Brigette writes a brief document which states what the main stages of a 'project' (perhaps 'job for the user' would be a better term in this context) should be. This happens to be very similar to the list given in Chapter 1. She stresses that:

- no job of work to change a system or implement a new one is to be done without there being a detailed specification first;
- the users must record agreement to each specification in writing before the work is carried out.

She draws up a simple procedure for recording all changes to user requirements.

Brigette, of course, has no organizational quality procedures, but she dictates that each person in the group (including herself) has to get someone else to check through their work when they finish a major task and that, before any new or amended software is handed over to the users, someone other than the original developer should test it. She sets up a simple system to record errors found in system testing and their resolution. She also creates a log file of reported user problems with operational systems.

Brigette does not worry about timesheets but arranges an informal meeting with her colleagues each Monday morning to discuss how things are going and also arranges to see the vice-principal, who is her official boss, and the heads of the finance and human resources sections each month to review progress in general terms.

Step 2.3: Identify project team organization

Some of these issues will be discussed in Chapter 12 on working in teams.

Project leaders, especially in the case of large projects, might have some control over the way that their project team is to be organized. Often, though, the organizational structure will be dictated to them. For example, a high-level managerial decision might have been taken that software developers and business analysts will be in different groups, or that the development of business-to-consumer web applications will be done within a separate group from that responsible for 'traditional' database applications.

If the project leader does have some control over the project team organization then this would best be considered at a later stage (see Step 7: Allocate resources).

CASE STUDY EXAMPLES: PROJECT ORGANIZATION

At IOE, there are groups of business analysts set up as teams which deal with individual user departments. Hence the users always know whom they should contact within the information systems department if they have a problem. Software developers, however, work in a 'pool' and are allocated to specific projects on an ad hoc basis.

At Brightmouth College, a software developer has been seconded to Brigette from the technicians supporting the computing courses in the college. She is also allowed to recruit a trainee analyst/programmer. She is not unduly worried about the organizational structure needed.

3.5 Step 3: Analyse project characteristics

Chapter 4 elaborates on the process of analysing project characteristics.

The general purpose of this part of the planning operation is to ensure that the appropriate methods are used for the project.

Step 3.1: Distinguish the project as either objective- or product-driven

This has already been discussed in the first chapter. As development of a system advances it tends to become more product-driven, although the underlying objectives always remain and must be respected.

Step 3.2: Analyse other project characteristics (including quality-based ones)

For example, is an information system to be developed or a process control system, or will there be elements of both? Will the system be safety critical, where human life could be threatened by a malfunction?

Step 3.3: Identify high-level project risks

Consideration must be given to the risks that threaten the successful outcome of the project. Generally speaking, most risks can be attributed to the operational or development environment, the technical nature of the project or the type of product being created.

CASE STUDY EXAMPLES: HIGH-LEVEL RISKS

We have already noted that Amanda has raised concerns about the possibility that engineers lack the motivation to complete with due care and attention the cost details for jobs done under annual contracts. Another risk relates to the software functionality which will produce cost analysis reports used for the future pricing of annual contracts. If the analysis is incorrect IOE could suffer financially. Amanda decides therefore that the analysis functionality will be produced using an iterative approach where an IOE marketing analyst will look at versions of the reports produced and suggest improvements to the methods of calculation and presentation before the system is finally made operational.

Brigette at Brightmouth College considers the application area to be very well defined. There is a risk, however, that there may be no package on the market that caters for the way that things are done at the moment. Brigette, therefore, decides that an early task in the project is to obtain information about the features of the main payroll packages that are available.

Step 3.4: Take into account user requirements concerning implementation

The clients may have their own procedural requirements. For example, an organization might mandate the use of a particular development method.

Step 3.5: Select development methodology and life-cycle approach

The development methodology and project life cycle to be used for the project will be influenced by the issues raised above. The idea of a methodology, that is, the group of methods to be used in a project, was discussed in Chapter 1. For many software developers, the choice of methods will seem obvious: they will use the ones that they have always used in the past. In Chapter 4 we recommend caution in assuming that the current project is really similar to previous ones.

Chapter 4 discusses life cycles in more detail.

As well as the methods to be used, there are generic ways of structuring projects, such as the use of the waterfall life cycle outlined in Chapter 4, that need to be considered. While the setting of objectives involves identifying the problems to be solved, this part of planning is working out the ways in which these problems are to be solved. For a project that is novel to the planner, some research into the methods typically used in the problem domain is worthwhile. For example, sometimes, as part of a project, a questionnaire survey has to be conducted. There are lots of books on the techniques used in such surveys and a wise move would be to look at one or two of them at the planning stage.

Chapter 5 goes into more detail on this topic. Function points are an attempt to measure system size without using lines of code.

Step 3.6: Review overall resource estimates

Once the major risks have been identified and the broad project approach has been decided upon, this would be a good point at which to re-estimate the effort and other resources required to implement the project. Where enough information is available an estimate based on function points might be appropriate.

3.6 Step 4: Identify project products and activities

The more detailed planning of the individual activities now takes place. The longer-term planning is broad and in outline, while the more immediate tasks are planned in some detail.

Step 4.1: Identify and describe project products (or deliverables)

In general, there can be no project products that do not have activities that create them. Wherever possible, we ought also to ensure the reverse: that there are no activities that do not produce a tangible product. Identifying all the things the project is to create helps us to ensure that all the activities we need to carry out are accounted for. Some of these products will be handed over to the client at the end of the project – these are *deliverables*. Other products might not be in the final configuration, but are needed as *intermediate* products used in the process of creating the deliverables.

These products will include a large number of *technical* products, such as training material and operating instructions. There will also be products to do with the *management* and the *quality* of the project. Planning documents would, for example, be management products.

The products will form a hierarchy. The main products will have sets of component products which in turn may have sub-component products and so on. These relationships can be documented in a Product Breakdown Structure (PBS) – see Figure 3.2. In this example the products have been grouped into those relating to the system as a whole, and those related to individual modules. A third 'group', which happens to have only one product, is called 'management products' and consists of progress reports. The asterisk in the progress reports indicates that there will be new instances of the entity 'progress report' created repeatedly throughout the project.

PRINCE2 suggests that the PBS be presented as a hierarchy diagram. In practice it may be more convenient to produce a structured list.

Note that in Figure 3.2 the only boxes that represent tangible products are those at the bottom of the hierarchy that are not further subdivided. Thus there are only six individual product types shown in the diagram. The boxes that are higher up – for example 'module products' – are simply the names of groups of items.

Some products are created from scratch, for example new software components. A product could quite easily be a document, such as a software design document. It might be a modified version of something that already exists, such as an amended piece of code. A product could even be

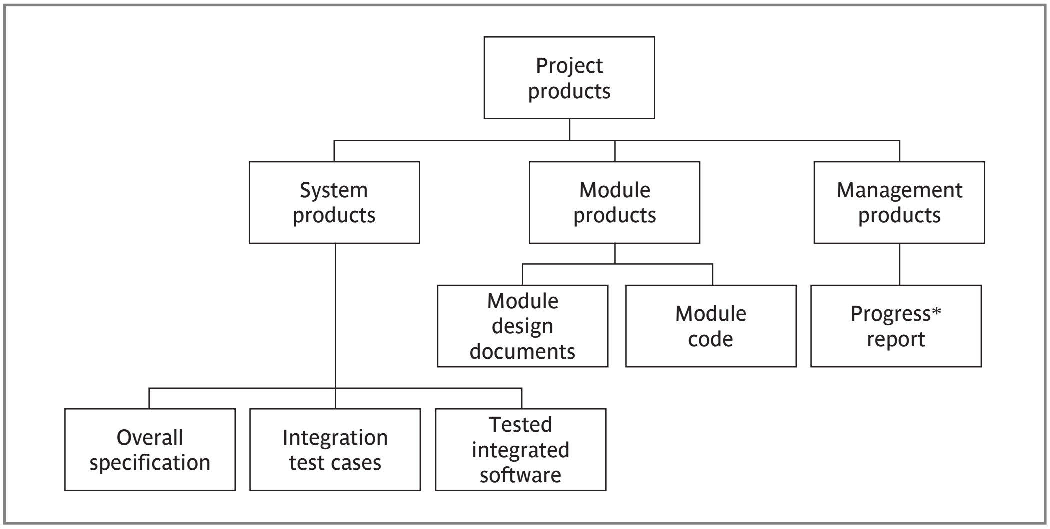

FIGURE 3.2 A fragment of a Product Breakdown Structure for a system development task
* indicates that further progress reports can be added during the course of the project.

a person, such as a 'trained user', a product of the process of training. Always remember that a product is the result of an activity. A common error is to identify as products things that are really activities, such as 'training', 'design' and 'testing'. Specifying 'documentation' as a product should also be avoided – by itself this term is just too vague.

This part of the planning process draws heavily on the standards laid down in PRINCE2. These specify that products at the bottom of the PBS should be documented by *Product Descriptions* which contain:

- the *name/identity* of the product;
- the *purpose* of the product;
- the *derivation* of the product (that is, the other products from which it is derived);
- the *composition* of the product;
- the *form* of the product;
- the relevant *standards*;
- the *quality* criteria that define whether the product is acceptable.

EXERCISE 3.2

At Brightmouth College, Brigette has decided that the finance department at the college should carry out acceptance testing of the new payroll system. This type of testing ensures that the application has been set up in a way that allows the users to carry out their jobs accurately using the new system. As the finance department staff are not sure what test case documents should look like, Brigette draws up a product description of a test case. Write the content for this product description.

CASE STUDY EXAMPLES: PRODUCT BREAKDOWN STRUCTURES

At IOE, Amanda finds that there is a standard PBS that she can use as a checklist for her own project.

Brigette at Brightmouth College has no installation standard PBS, although she can, of course, refer to various books for standard checklists. She decides that one part of the PBS should contain the products needed to help select the appropriate hardware and software for the payroll application (Figure 3.3).

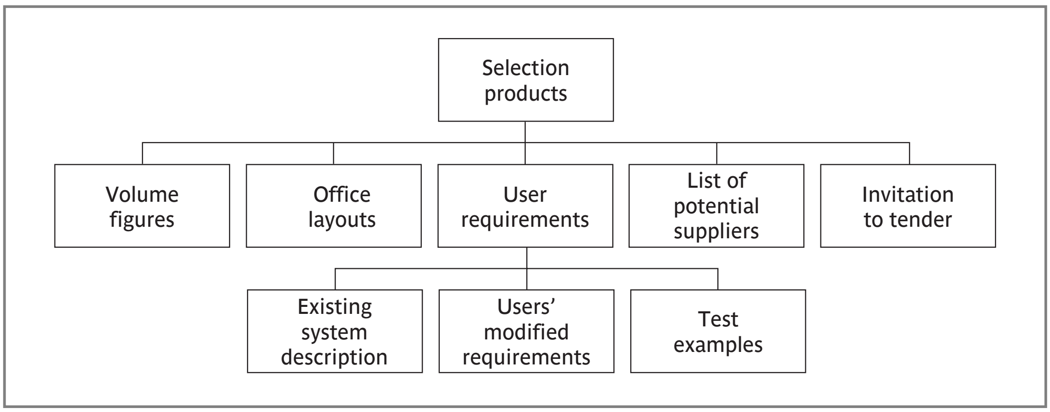

FIGURE 3.3 A Product Breakdown Structure (PBS) for the products needed to produce an invitation to tender (ITT)

Step 4.2: Document generic product flows

The PFD effectively documents, in outline, the method (see Chapter 1) for the project.

Some products will need one or more other products to exist first before they can be created. For example, a program design must be created before the program can be written and the program specification must exist before the design can be commenced. These relationships can be portrayed in a *Product Flow Diagram* (PFD). Figure 3.4 gives an example. Note that the 'flow' in the diagram is assumed to be from top to bottom and left to right. In the example in Figure 3.4, 'user requirements' is in an oval which means that it is used by the project but is not created by it. It is often convenient to identify an overall product at the bottom of the diagram, in this case 'integrated/tested software', into which all the other products feed.

PFDs should not have links between products which loop back iteratively. This is emphatically *not* because iterations are not recognized. On the contrary, the PFD allows for looping back at any point. For example, in the PFD shown in Figure 3.4, say that during integration testing it was found that a user requirement had been missed in the overall system specification. If we go back to overall system specification and change it we can see from the PFD that all the products that follow it might need to be reworked. A new

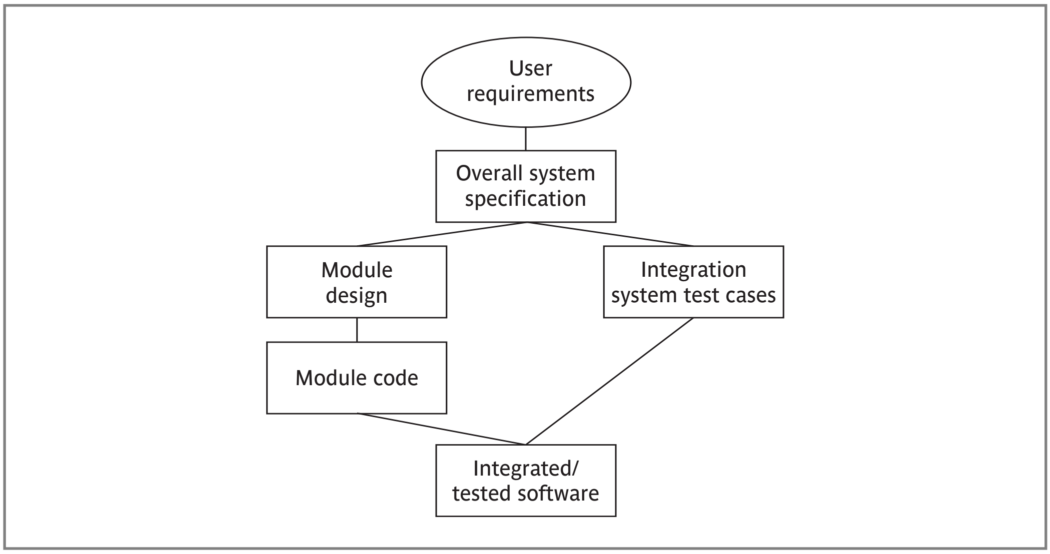

FIGURE 3.4 A fragment of a Product Flow Diagram (PFD) for a software development task

module might need to be designed and coded, test cases would need to be added to check that the new requirements had been successfully incorporated, and the integration testing would need to be repeated.

The form that a PFD takes will depend on assumptions and decisions about how the project is to be carried out. These decisions may not be obvious from the PFD and so a textual description explaining the reasons for the structure can be helpful.

CASE STUDY EXAMPLES: IOE HAS STANDARD PFD

At IOE, Amanda has an installation standard PFD for software development projects. This is because a recognized software development method is used which lays down a sequence of documents that have to be produced. This sequence of products can be straightforwardly documented as a PFD.

EXERCISE 3.3

Draw up a possible Product Flow Diagram (PFD) based on the Product Breakdown Structure (PBS) shown in Figure 3.3. This identifies some of the products of the Brightmouth payroll project, particularly those generated when gathering information to be presented to potential suppliers of the hardware as part of an 'invitation to tender'. The volume figures are such things as the number of employees for whom records will have to be maintained.

This may be delayed to later in the project when more information is known.

Step 4.3: Recognize product instances

Where the same generic PFD fragment relates to more than one instance of a particular type of product, an attempt should be made to identify each of those instances. In the example in Figure 3.2, it could be that in fact there are just two component software modules in the software to be built.

Step 4.4: Produce ideal activity network

In order to generate one product from another there must be one or more activities that carry out the transformation. By identifying these activities we can create an activity network which shows the tasks that have to be carried out and the order in which they have to be executed.

CASE STUDY EXAMPLES: ACTIVITY NETWORK FOR IOE MAINTENANCE ACCOUNTS

Part of the initial activity network developed from the PFD in Figure 3.4 for the software development task might look like Figure 3.5.

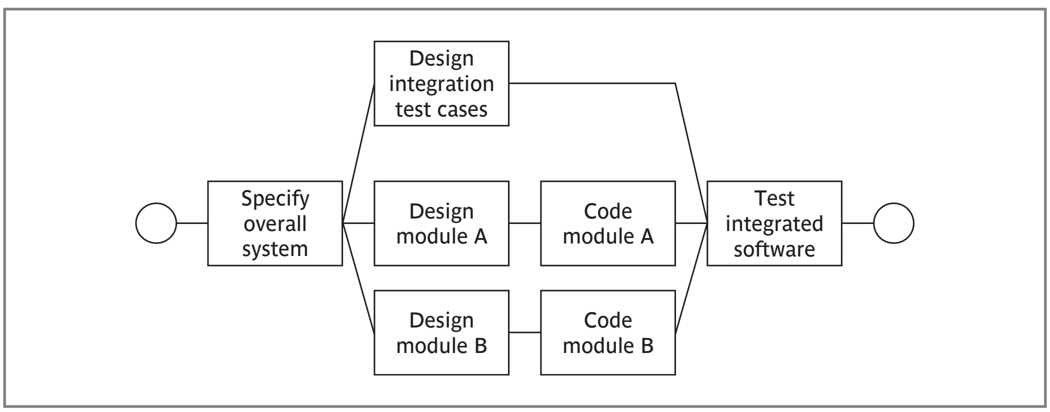

FIGURE 3.5 An example of an activity network

EXERCISE 3.4

Draw up an activity network for the Product Flow Diagram that you created in Exercise 3.3 (or the PFD given in the solution if you prefer!).

The activity networks are 'ideal' in the sense that no account has been taken of resource constraints. For example, in Figure 3.5, it is assumed that resources are available for both software modules to be developed in parallel. A good rule is that activity networks are never amended to take account of resource constraints.

Step 4.5: Modify the ideal to take into account need for stages and checkpoints

The approach to sequencing activities described above encourages the formulation of a plan which will minimize the overall duration, or 'elapsed time', for the project. It assumes that an activity will start as soon as the preceding ones upon which it depends have been completed.

Strictly, a milestone is a dummy activity with no duration that indicates the start or end of a group of activities. The milestone would therefore be *after* the checkpoint activity.

There might, however, be a need to modify this by dividing the project into stages and introducing checkpoint activities. These are activities which draw together the products of preceding activities to check that they are compatible. This could potentially delay work on some elements of the project – there has to be a trade-off between efficiency and quality.

The people to whom the project manager reports could decide to leave the routine monitoring of activities to the project manager. However, there could be some key activities, or *milestones*, which represent the completion of important stages of the project of which they would want to take particular note. Checkpoint activities are often useful milestones.

EXERCISE 3.5

In the example in Figure 3.5, it has been decided that the designs for modules A and B are to be checked for consistency by 'dry-running' them against the integration test cases before committing staff to software coding. Redraw the activity network to reflect this.

3.7 Step 5: Estimate effort for each activity

Step 5.1: Carry out bottom-up estimates

Some overall estimates of effort, cost and duration will already have been done (see Step 3.6).

Chapter 5 on software effort estimation deals with this topic in more detail.

At this point, estimates of the staff effort required, the probable elapsed time and the non-staff resources needed for each activity will need to be produced. The method of arriving at each of these estimates will vary depending on the type of activity.

The difference between *elapsed time* and *effort* should be noted. Effort is the amount of work that needs to be done. If a task requires three members of staff to work for two full days each, the effort expended is six days. Elapsed time is the time between the start and end of a task. In our example above, if the three members of staff start and finish at the same time then the elapsed time for the activity would be two days.

The individual activity estimates of effort should be summed to get an overall bottom-up estimate which can be reconciled with the previous top-down estimate.

The activities on the activity network can be annotated with their elapsed times so that the overall duration of the project can be calculated.

Step 5.2: Revise plan to create controllable activities

The estimates for individual activities could reveal that some are going to take quite a long time. Long activities make a project difficult to control. If an activity involving system testing is to take 12 weeks, it would be difficult after six weeks to judge accurately whether 50 per cent of the work is completed. It would be better to break this down into a series of smaller subtasks.

CASE STUDY EXAMPLES: IOE ANNUAL MAINTENANCE CONTRACTS – CARRY OUT BOTTOM-UP ESTIMATES

At IOE, Amanda has to estimate the lines of code for each of the software modules. She looks at programs that have been coded for similar types of application at IOE in the past to get some idea of the size of the new modules. She then refers to some conversion tables that the information systems development department at IOE have produced which convert the lines of code into estimates of effort. Other tables allow her to allocate the estimated effort to the various stages of the project.

Although Brigette is aware that some additional programs might have to be written to deal with local requirements, the main software is to be obtained ‘off the shelf’ and so estimating based on lines of code would clearly be inappropriate. Instead, she looks at each individual task and allocates a time. She realizes that in many cases these represent ‘targets’ as she is uncertain at the moment how long these tasks will really take (see Step 6 below).

There might be a number of activities that are important, but individually take up very little time. For a training course, there might be a need to book rooms and equipment, notify those attending, register students on the training system, order refreshments, copy training materials and so on. In a situation like this it would be easier to bundle the activities into a single merged activity ‘make training course arrangements’ which could be supplemented with a checklist.

In general, try to make activities about the length of the reporting period used for monitoring and controlling the project. If you have a progress meeting every two weeks, then it would convenient to have activities of two weeks’ duration on average, so that progress meetings would normally be made aware of completed tasks each time they are held.

3.8 Step 6: Identify activity risks

Step 6.1: Identify and quantify activity-based risks

Chapter 7 on risk touches on this topic in more detail.

Risks inherent in the overall nature of the project have already been considered in Step 3. We now want to look at each activity in turn and assess the risks to its successful outcome. Any plan is always based on certain assumptions. Say the design of a component is planned to take five days. This is based on the assumption that the client's requirement is clear and unambiguous. If it is not then additional effort to clarify the requirement would be needed. The possibility that an assumption upon which a plan is based is incorrect constitutes a risk. In this example, one way of expressing the uncertainty would be to express the estimate of effort as a range of values.

As will be seen in Chapter 7, a simple way of dealing with uncertainty is to have a 'most likely' estimate for where everything works with no problems (such as users changing their requirements) and a second estimate that includes a safety margin so that it has an estimated 95 per cent chance of being met.

A project plan will be based on a huge number of assumptions, and so some way of picking out the risks that are most important is needed. The damage that each risk could cause and the likelihood of it occurring have to be gauged. This assessment can draw attention to the most serious risks. The usual effect if a problem materializes is to make the task longer or more costly.

Step 6.2: Plan risk reduction and contingency measures where appropriate

It may be possible to avoid or at least reduce some of the identified risks. On the other hand, *contingency plans* specify action that is to be taken if a risk materializes. For example, a contingency plan could be to use contract staff if a member of the project team is unavailable at a key time because of serious illness.

Step 6.3: Adjust overall plans and estimates to take account of risks

We may change our plans, perhaps by adding new activities which reduce risks. For example, a new programming language might mean we schedule training courses and time for the programmers to practise their new programming skills on some non-essential work.

CASE STUDY EXAMPLES: IDENTIFYING RISKS

As well as the new software modules that will have to be written, Amanda has identified several existing modules that will need to be amended. The ease with which the modules can be amended will depend upon the way that they were originally written. There is therefore a risk that they may take longer than expected to modify.

Amanda takes no risk reduction measures as such but notes a pessimistic elapsed time for the amendment activity.

Brigette identifies as a risk the possible absence of key staff when investigating the user requirements, as this activity will take place over the holiday period. To reduce this risk, she adds a new activity, 'arrange user interviews', at the beginning of the project. This will give her advance notice of any likely problems of this nature.

3.9 Step 7: Allocate resources

Chapter 8 on resource allocation covers this topic in more detail.

Step 7.1: Identify and allocate resources

The type of staff needed for each activity is recorded. The staff available for the project are identified and are provisionally allocated to tasks.

Step 7.2: Revise plans and estimates to take into account resource constraints

Some staff may be needed for more than one task at the same time and, in this case, an order of priority is established. The decisions made here may have an effect on the overall duration of the project when some tasks are delayed while waiting for staff to become free.

Gantt charts are named after Henry Gantt and 'Gantt' should therefore not be written in capital letters as if it stood for something!

Ensuring someone is available to start work on an activity as soon as the preceding activities have been completed might mean that they are idle while waiting for the job to start and are therefore used inefficiently.

The product of Steps 7.1 and 7.2 would typically be a Gantt chart – see Figure 3.6. The Gantt chart gives a clear picture of when activities will actually take place and highlights which ones will be executed at the same time. Activity networks can be misleading in this respect.

CASE STUDY EXAMPLES: TAKING RESOURCE CONSTRAINTS INTO ACCOUNT

Amanda has now identified three new major software modules plus an existing software module that will need extensive amendment. At IOE the specification of modules is carried out by the lead systems analyst for the project (who in this case is Amanda) assisted by junior analyst/designers. Four analyst/programmers are available to carry out the design, coding and unit testing of the individual modules. After careful consideration and discussion with her manager, Amanda decides to use only three analyst/programmers so as to minimize the risk of staff waiting between tasks and thus reduce staff costs. It is accepted that this decision, while reducing the cost of the project, will delay its end.

Brigette finds that she herself will have to carry out many important activities. She can reduce the workload on herself by delegating some work to her two colleagues, but she realizes that she will have to devote more time to specifying exactly what they will have to do and to checking their work. She adjusts her plan accordingly.

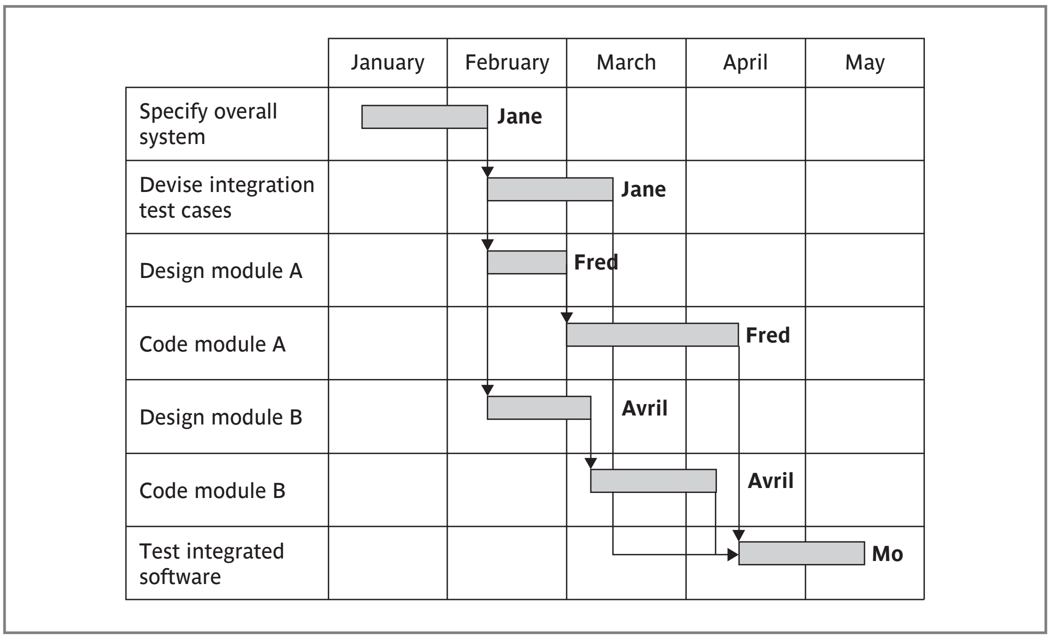

FIGURE 3.6 Gantt chart showing when staff will be carrying out tasks

3.10 Step 8: Review/publicize plan

Step 8.1: Review quality aspects of the project plan

A danger when controlling any project is that an activity can reveal that an earlier activity was not properly completed and needs to be reworked. This, at a stroke, can transform a project that appears to be progressing satisfactorily into one that is badly out of control. It is important to know that when a task is reported as completed, it really is – hence the importance of quality reviews. Each task should have *quality criteria*. These are quality checks that have to be passed before the activity can be 'signed off' as completed.

CASE STUDY EXAMPLES: IOE EXISTING QUALITY STANDARDS

Amanda finds that at IOE, the Quality Standards and Procedures Manual lays down quality criteria for each type of task. For example, all module design documentation for any group of modules that interact with one another has to be reviewed by a group of colleagues before the coding can commence. This is to reduce the likelihood of integration problems when the components are finally executed together. Amanda adds an activity to her plan to deal with this.

EXERCISE 3.6

Brigette has no installation standards to help her apart from the minimal ones she has written herself. What quality checks might Brigette introduce to ensure that she has understood the users' requirements properly?

Step 8.2: Document plans and obtain agreement

It is important that the plans be carefully documented and that all the parties to the project understand and agree to the commitments required of them in the plan. This may sound obvious, but it is amazing how often this is not done. Chapter 12 describes the use of a *communications plan* to ensure appropriate communications between stakeholders at the right points in the project.

EXERCISE 3.7

At the end of Chapter 1 the main sections of a project plan document were listed. Draw up a table showing which Step Wise activities provide material for which sections of the project plan.

3.11 Steps 9 and 10: Execute plan/lower levels of planning

Once the project is under way, plans will need to be drawn up in greater detail for each activity as it becomes due. Detailed planning of the later stages will need to be delayed because more information will be available nearer the start of the stage. Of course, it is necessary to make provisional plans for the more distant tasks, because thinking about what needs to be done can help unearth potential problems, but sight should not be lost of the fact that these plans are provisional.

CASE STUDY EXAMPLES: LOWER-LEVEL PLANNING

While work is going on with the specification of the individual modules, Amanda has some time to start planning the integration tests in some detail. She finds that one of the modules – the one that deals with recording job requests – does not actually communicate directly with the other new modules and can therefore be reviewed independently of the others. She schedules an earlier review of this module as this allows coding of the module to be started earlier.

When Brigette comes to consider the activity 'draft invitation to tender', she has to familiarize herself with the detailed institutional rules and procedures that govern this process. She finds that in order to draft this document she will need to obtain some additional pieces of information from the users.

3.12 Conclusion

This chapter has presented a framework into which the techniques described in the other parts of the book should slot. It is suggested that any planning approach should have the following elements:

- the establishment of project objectives;
- the analysis of the characteristics of the project;
- the establishment of an infrastructure consisting of an appropriate organization and set of standards, methods and tools;
- the identification of the products of the project and the activities needed to generate those products;
- the allocation of resources to activities;
- the establishment of quality controls.

Project planning is an iterative process. As the time approaches for particular activities to be carried out they should be replanned in more detail.