dangerous, since the system size might be increased and the original objectives obscured. Because of these dangers, this process must be done consciously and in a controlled manner.

Case Study Examples: The IOE maintenance staff are to be given the extra task of entering data about Modified project completed jobs. They do not benefit from this additional work. To give some objectives benefit, the system is to be extended to reorder spare parts automatically when required.

At Brightmouth College, the personnel department has a lot of work preparing payroll details for finance. It will be tactful to agree to produce some management information reports for personnel 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.

Identifying Project Infrastructure:

Projects are rarely initiated in a vacuum. There is usually some kind of existing infrastructure into which the project can fit. The project leader who does not already know about this structure needs to find out its precise nature.

Some of the issues of strategic planning are addressed m Chapter 3.

Step 2. 7 : Identify relationship between the project and strategic planning As well as identifying projects to be carried out, an organization needs to decide the order in which these projects are to be carried out. It also needs to establish the framework within which the proposed new systems are to fit. Hardware and software standards, for example, are needed so that various systems can communicate w ith each other. These strategic decisions must be documented in a strategic business plan or in an information technology plan that is developed from the business plan.

Case Study Examples: Amanda finds at IOE that there is a well-defined rolling strategic plan that has Role of existing strategic identified her group accounts subsystem as an important required dev elopment, plans Because it is an extension of an existing system, the hardware and software platforms upon which the application are to run are dictated.

Brigettc at Brightmouth College finds that there is an overall College strategic plan that describes new courses to be developed, and so on, and mentions in passing the need for 'appropriate administrative procedures* to be in place. In a short section in a consultant's report from an accountancy firm concerning the implications of financial autonomy, there is a recommendation that independent payroll processing be undertaken. Although the college has quite a lot of IT 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 Any organization that develops software should define its 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 See Chapter 9 on ensure that changes to requirements are implemented in a safe and orderly way. 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 andprocedures manual.

The organization, as part of its monitoring and control policy must have in place a measurement programme that 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 the way that the project is controlled: for example, the way that the hours spent by team members on indiv idual tasks are recorded on time-sheets

Amanda at IOE finds that there is a very weighty manual of development Case Study Examples:

standards, which, among other things, specifies that SSADM will be the analysis Identifying standards and design method used. She finds that a separate document has been prepared, laying down quality procedures. This specifies when the reviews of work will be carried out and describes detailed procedures about how the reviews are to be done. Amanda also finds a set of project management guidelines modelled closely on PRINCH 2.

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 that 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 I. She stresses that:

• no job of work to change a system or implement a new one is to be done w ithout there being a detailed specification first;

• the users must agree to, or 'sign off', 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 his or her work at the end of a major task and that, before any new or amended software is handed over to the users, someone other than the original producer 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 time sheets but arranges an informal meeting w ith her colleagues each Monday morning to discuss how things are going and also arranges to see the Vice-Principal, w ho is her official boss, and the heads of the finance and personnel sections each month to rev iew progress in general terms.

Stop 2.3: Identify project team organization

Some of these issues will be discussed in Chapter 11 - Managing people and organizing teams.

Project leaders, especially in the case of large projects, will often have some control over the organizational structure of the project team. More often, though, the organizational structure will be dictated to them. For example, there might have been a high level managerial decision that code developers and systems analysts will be in different groups, or that the development of PC applications will not be done w ithin the same group as that responsible for legacy' main-frame 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: At IOE. there are groups of systems analysts set up as teams that deal with Project organization individual user departments. Hence the users always know whom they should contact within the information systems department if they have a problem. Code developers, however, work in a 4pool* and are allocated to specific projects on an ad hoc basis.

At Brightmouth College, Brigette has seconded to her a software developer who has been acting as a technician 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 that is needed.

Analyzing Project Charactertics:

The general purpose of this part of the planning operation is to ensure that the Chapter 4 elaborates on appropriate methods are used for the project. the process of analysing project characteristics.

Step 3. /: Distinguish the project as either objective- or product-driven This has already been discussed in the first chapter. A general point to note is that as system development 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 this an information system that is being developed or a process control system, or does it have elements of both? Is it a safety-critical system, that is, 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.

At IOE, Amanda identifies the danger of there being resistance to the new system Case Study Example: by maintenance engineers, especially as a new centralized group accounts office High level risks is to be set up. Amanda decides therefore that additional efforts are needed to consult all sections involved and that the new procedures should be introduced in small increments to accustom staff to them gradually.

Brigette at Brightmouth College considers the application area to be very well-defined. There is a risk, however, that there might be no application 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 applications that are available.

Step 3.4: Take into account user requirements concerning implementation The clients will usually have their own procedural requirements. For example, work for government departments usually requires the use of SSADM.

Step 3.5: Select general lifecycle approach in the light of the above

The project life cycle to be used for the project will be influenced by the issues Chapter 4 discusses lite raised above. For example, a prototyping approach might be used where the user cycles in more detail, requirements are not clear.

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

Chapter 7 goes into this in more detail.

PRINCE 2 suggests that the PBS be presented as a hierarchy diagram. In practice it may be more convenient to produce a structured list.

Stop 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.

2.6 Step 4: Identify project products and activities

The more detailed planning of the individual activities that will be needed 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. Making sure we have identified all the things the project is to create helps us to ensure that all the activities we need to carry out are accounted for.

These products will include a large number of technical products including training material and operating instructions, but also 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 might have sub-component products and soon. These relationships can be documented in a Product Breakdown Structure (PBS).

This part of the planning process draws heavily on the standards laid dow n in PRINCE 2. 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 relevant standards;

• the quality criteria that should apply to it.

Case Study Example: PBS

At IOE. Amanda finds that there is a standard PBS that she can use as a check-list for her own project.

Brigette at Brightmouth College has no installation standard PBS. although she can, of course, refer to various books for standard check-lists. 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 2.2).

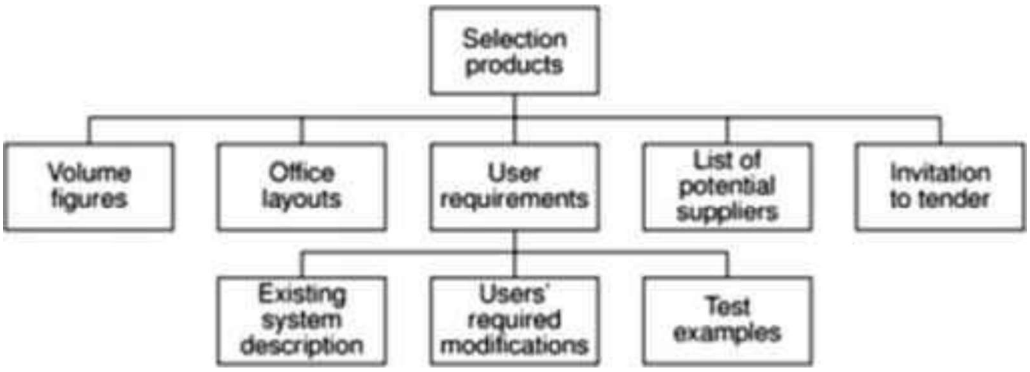


Figure 2.2 A fragment of the PBS for the Brightmouth College Payroll Project.

Stop 4.2: Document generic product flows

Some of the products will need some other product 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 Mow Diagram (PFD). Figure 2.3 gives an example.

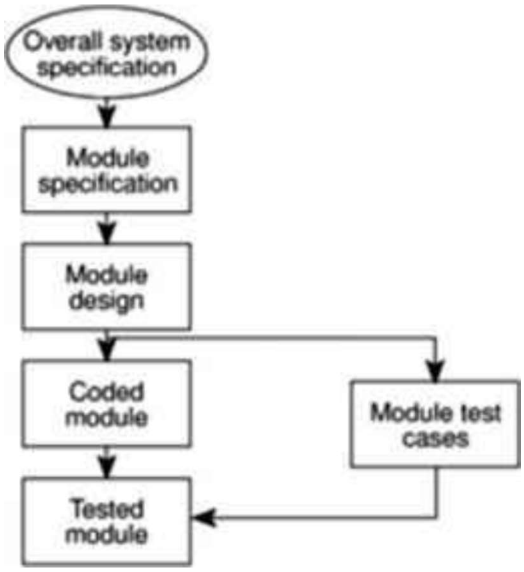


Figure 23 A fragment of a PFD.

Case Study Example: At IOE, Amanda has a standard installation PFD. Many of the products that will IOE has standard PFD make up Amanda's application will be of the same type: hence the same generic

PFD will apply to each instance. It is pointless in these circumstances to draw up a separate PIT) for each instance of the product.

Exercise 2.2 Draw up a possible Product Flow Diagram (PFD) based on the Product

Breakdown Structure (PBS) shown in Figure 2.2. This represents the products that will be produced when gathering information to be presented to potential suppliers of the hardware. The volume figures are for 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.

Case Study Example: identifying product instances

Amanda decides that there are likely to be four major software modules needed in her application for which the PFD fragment in Figure 2.3 would be appropriate The products that Brigette can identify at the present all have a single instance.

Step 4.4: Produce ideal activity network

In order to generate one product from another there must be one or more activ ities 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 Example: Part of the initial activity network for the IOE Maintenance (¡roup Accounts Activity network for IOE project might look like Figure 2.4. Maintenance Accounts

Exercise 2.3 Draw up an Activity Network for the Product Flow Diagram that you created in

Exercise 2.2 (or the PFD given in the solution if you prefer!).

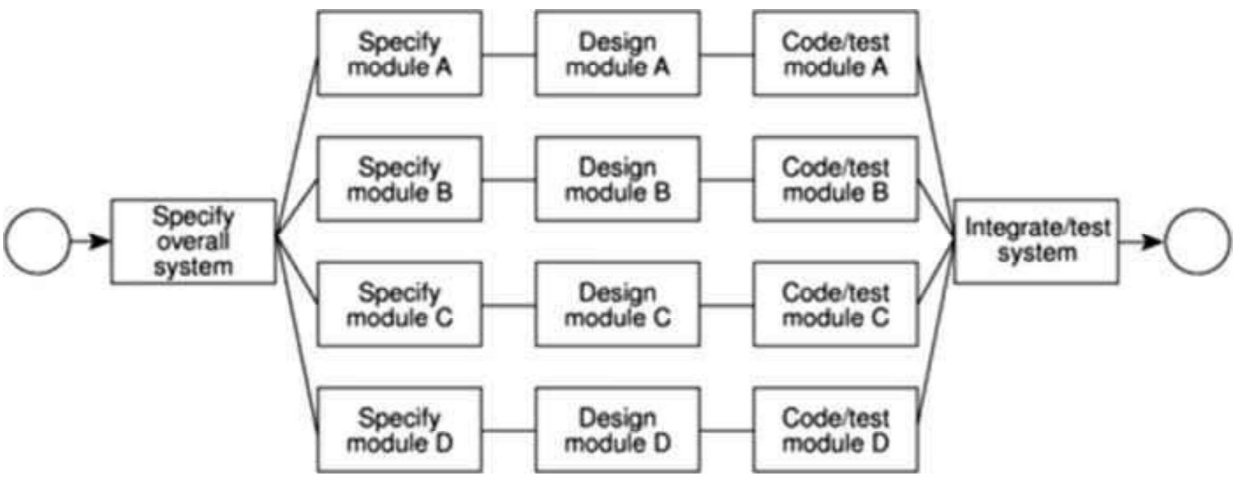
The activity networks are * ideal* in the sense that no account has been taken of resource constraints. For example in Figure 2.4, it is assumed that resources are available for all four software modules to be developed in parallel.

Step 4.5: Modify the ideal to take into account need for stages and check-points

The approach to sequencing activities described above encourages the formulation of a plan that 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.

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

Amanda decides that after the four modules have been specified, the four Exercise 2.4 specifications need to be carefully checked to see that they are consistent and compatible. Redraw the activity network in Figure 2.4 to reflect this.



Identifying project products and activities:

The more detailed planning of the individual activities that will be needed 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. Making sure we have identified all the things the project is to create helps us to ensure that all the activities we need to carry out are accounted for.

These products will include a large number of technical products including training material and operating instructions, but also 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 might have sub-component products and so on. These relationships can be documented in a Product Breakdown Structure (PBS).

This part of the planning process draws heavily on the standards laid down in PRINCE 2. 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 relevant standards;

the quality criteria that should apply to it.

Case Study Example: PBS

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

Brigette at Brightmouth College has no installation standard PBS, although she can, of course, refer to various books for standard check-lists. 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 2.2).

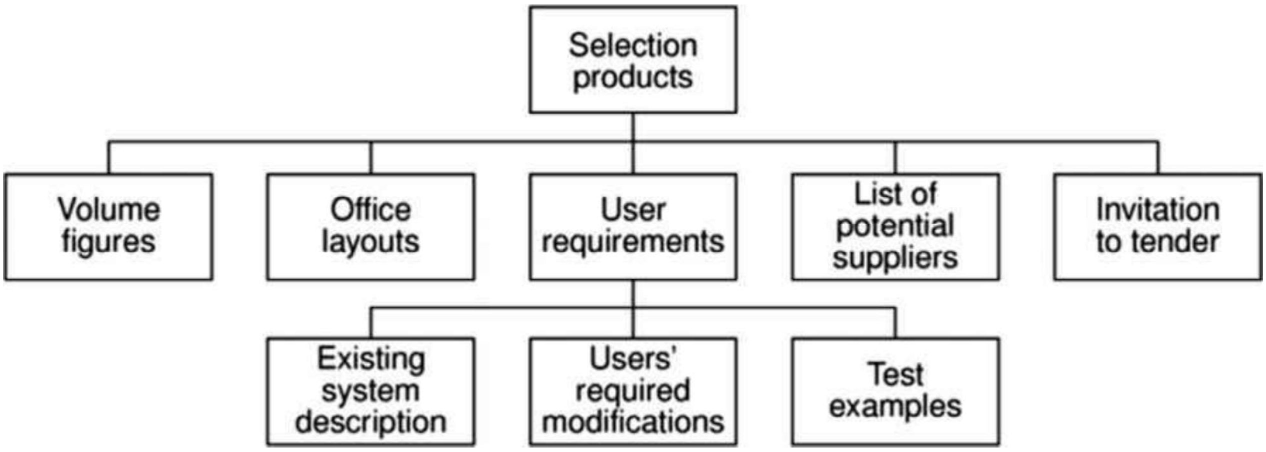


Figure 2.2 A fragment of the PBS for the Brightmouth College Payroll Project.

Step 4.2: Document generic product flows

Some of the products will need some other product 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 2.3 gives an example.

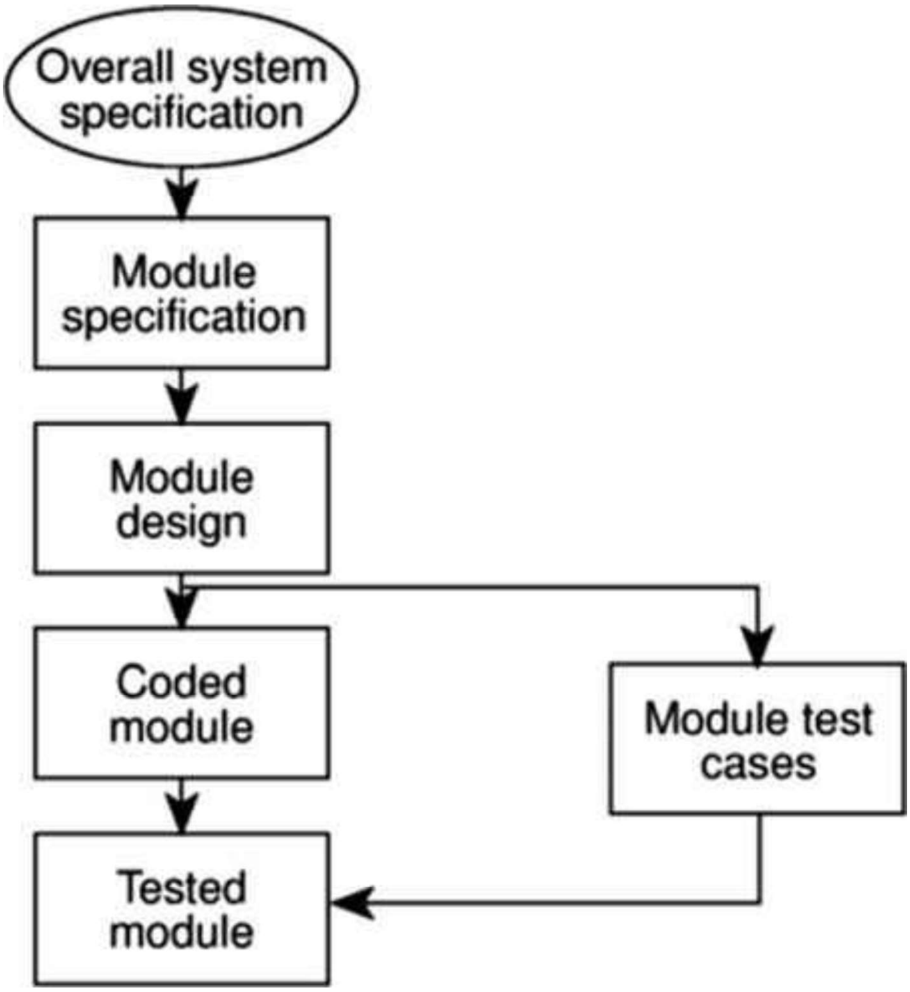


Figure 2.3 A fragment of a PFD.

Case Study Example: At IOE, Amanda has a standard installation PFD. Many of the products that will IOE has standard PFD make up Amanda's application will be of the same type: hence the same generic

PFD will apply to each instance. It is pointless in these circumstances to draw up a separate PFD for each instance of the product.

Exercise 2.2 Draw up a possible Product Flow Diagram (PFD) based on the Product

Breakdown Structure (PBS) shown in Figure 2.2. This represents the products that will be produced when gathering information to be presented to potential suppliers of the hardware. The volume figures are for 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.

Case Study Example: identifying product instances

Amanda decides that there are likely to be four major software modules needed in her application for which the PFD fragment in Figure 2.3 would be appropriate

The products that Brigette can identify at the present all have a single instance.

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 Example: Part of the initial activity network for the IOE Maintenance Group Accounts Activity network for IOE project might look like Figure 2.4. Maintenance Accounts

Exercise 2.3 Draw up an Activity Network for the Product Flow Diagram that you created in

Exercise 2.2 (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 2.4, it is assumed that resources are available for all four software modules to be developed in parallel.

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 that 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.

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

Amanda decides that after the four modules have been specified, the four Exercise 2.4 specifications need to be carefully checked to see that they are consistent and compatible. Redraw the activity network in Figure 2.4 to reflect this.

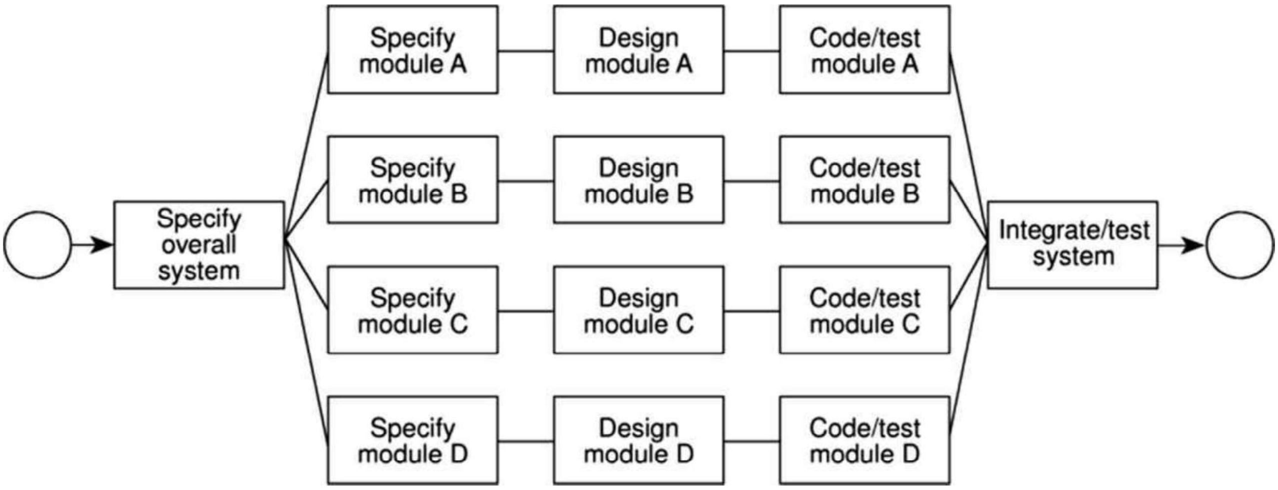


Figure 2.4 An activity network fragment for the IOE Maintenance Group

Estimate Efforts Each Activity:

Step 5.1: Carry out bottom-up estimates

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

Chapter 5 on Software At this point, estimates of the staff effort and other resources required, and the

Estimation deals with this probable elapsed time needed for each activity will need to be produced. The topic in more detail. method of arriving at each of these estimates will vary depending on the type of activity.

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 might reveal that some are going to take quite a long time. Long activities often make a project difficult to control. If an activity involving system testing is to take 12 weeks, it might be difficult after six weeks to judge accurately whether 50% of the work is completed. It would be better to break this down into a series of smaller sub-tasks.