



Appendix B: Answer pointers

Chapter 1

1.1 Examples of projects

The order you put these projects in is, of course, to a large degree subjective. Here is one example of a possible ordering.

- 1 Putting a robot vehicle on Mars to search for signs of life** Almost everybody puts this one first. The huge scale of the task, the relative novelty of the project, all the different specialisms involved and the international nature of the project make it special. Note that the successful achievement of the project from the engineering point of view is the safe landing of the robot, not the discovery of signs of life.
- 2 Writing an operating system** This is a prime example of a software development project.
- 3 Amending a financial system to deal with a common European currency** This project is modifying an existing system rather than creating a new one from scratch. Many software projects have this characteristic and it does not make them any less a software project.
- 4 Installing a new version of a word processing package in an organization** Although no software is being produced or modified, many of the stages that are associated with software projects will be involved and the techniques of software project management would be appropriate.
- 5 Investigation into the reasons why a user has a problem with a computer system** This will have many of the stages common to software projects, although the precise nature of the end result is uncertain at the outset. It could be that the user needs some simple remedial training. On the other hand, it could turn out to be quite a considerable software modification task.
- 6 Getting married** There should be lots of arguments about this one! Some will be reluctant to give a high rating to this because of its personal nature. The degree to which this is 'project-like' will depend very much upon the cultural milieu in which it takes place. Very often it requires a high degree of planning, involves lots of different people and, for most people, is a non-routine operation.
- 7 A research project into what makes a good human-computer interface** Compared to some of the projects above, the objectives of the research project are more open-ended and the idea of a specific client for the end-product may be less well defined. Research projects are in some ways special cases and the approach to their planning needs a rather different approach, which is outside the scope of this book.

- 8 Producing an edition of a newspaper** In some ways this has all the characteristics of a project. There are lots of different people with lots of different specialisms whose work needs to be coordinated in order to produce an end-product under very tight time constraints. What argues against this as a typical project is that it is repeated. After a while, everyone knows what he or she needs to do and most of the problems that arise are familiar and the procedures to deal with them are well defined.
- 9 A second-year programming assignment for a computing student** This is not being done for a customer, although it could be argued that the tutor responsible for setting and assessing the assignment is, in effect, a surrogate client. Not all the stages of a normal project will be gone through.

1.2 Brightmouth College payroll: Stages of a project

- 1 Project evaluation** All the costs that would be incurred by the college if it were to carry out its own payroll processing would need to be carefully examined to ensure that it would be more cost-effective than letting the local authority carry on providing the service.
- 2 Planning** The way that the transfer to local processing is to be carried out needs to be carefully planned with the participation of all those concerned. Some detailed planning would need to be deferred until more information was available, for example which payroll package was to be used.
- 3 Requirements elicitation and analysis** This is finding out what the users need from the system. To a large extent it will often consist of finding out what the current system does, as it may be assumed that in general the new system is to provide the same functions as the old. The users might have additional requirements, however, or there might even be facilities that are no longer needed.
- 4 Specification** This involves documenting what the new system is to be able to do.
- 5 Design/coding** As an 'off-the-shelf' package is envisaged, these stages will be replaced by a package evaluation and selection activity.
- 6 Verification and validation** Tests will need to be carried out to ensure that the selected package will actually do what is required. This task might well involve parallel running of the old and new systems and a comparison of the output from them both to check for any inconsistencies.
- 7 Implementation** This would involve such things as installing the software, setting system parameters such as the salary scales, and setting up details of employees.
- 8 Maintenance/support** This will include dealing with users' queries, liaising with the package supplier and taking account of new payroll requirements.

1.3 Estimation of the height of the building you are in

We will not spoil the fun by suggesting a particular method.

1.4 The nature of an operating system

Many large organizations that are committed to computer-based information systems have specialists responsible for the maintenance of operating systems. However, as an operating system is primarily concerned with driving the hardware, it is argued that it has more in common with what we have described as embedded systems.

1.5 Brightmouth College payroll: objectives-driven vs. product-driven

This project is really driven by objectives. If in-house payroll processing turns out not to be cost-effective, then the project should not try to implement such a solution. Other ways of meeting the objectives set could be considered: for example, it might be possible to contract out the processing to some organization other than the local authority at a lower cost.

1.6 Brightmouth college payroll: stakeholders

Major stakeholders would include:

- the finance department;
- the human resources department, who would need to supply most of the employee details needed;
- heads of departments, who would need to submit details of hours worked for part-time staff;
- staff, who would naturally be concerned that they are paid correctly;
- site management: the new arrangements may mean that the office layout has to be rearranged physically;
- software and hardware vendors.

One group of stakeholders that might not be readily identified at first is the local government authority and its staff. It might seem strange to list the people who used to do the job, but who are no longer required. The project manager's job will be made a lot easier if their cooperation and help can be obtained. The project manager would do well to sound out tactfully how the local authority staff feel about losing this work. It could be that they are pleased to be shot of the workload and hassle involved! Arrangements that take into account existing local authority staff might be possible. For example, if the college needs to recruit new staff to deal with payroll, it might smooth things to give the job to a member of the local authority staff who already deals with this area.

1.7 Defining objectives

Among the comments and queries that could be made in each case are:

- (i) Have the actual time and the amount of the budget been specified somewhere? Deadline and budget constraints normally have to be set against the scope and the quality of the functions to be delivered. For example, if the deadline were not achieved, would the customer rather have the full set of functionality at a later date, or an essential sub-set of the functionality on the deadline date?

- (ii) 'The fewest possible software errors' is not precise. Removing errors requires effort and hence money. Can developers spend as much money and time as they like if this reduces errors?
- (iii) What does 'user-friendly' really mean? How is it measured? Normally 'ease of use' is measured by the time it takes for a beginner to become proficient in carrying out standard operations.
- (iv) What does 'full documentation' mean? A list of the types of document to be produced, perhaps with an indication of the content layout, would be more useful.

1.8 Brightmouth college payroll: objectives, goals and measures of effectiveness

The original objective might have been formulated as: '*To carry out payroll processing at less cost while maintaining the current scope and quality of services*'.

In order to achieve this, sub-objectives or goals will usually have been identified, for example:

- to transfer payroll processing to the college by 1 April;
- to implement in the new system those facilities that exist in the current system less those identified in the initial report as not being required;
- to carry out the implementation of the payroll processing capability within the financial constraints identified in the initial report.

It should be noted that the objectives listed above do not explicitly mention such things as putting into place ongoing arrangements to deal with hardware and software maintenance, security arrangements and so on. By discussing and trying to agree objectives with the various people involved, the true requirements of the project can be clarified.

Measures of effectiveness for the sub-objectives listed above might include the following:

- *Date of implementation* Was the new system being used operationally by the agreed date?
- *Facilities* In parallel runs, were all the outputs produced by the old system and still required also produced by the new system?
- *Costs* How did the actual costs incurred compare with the budgeted costs?

1.9 A day in the life of a project manager

Planning:

- staffing requirements for the next year.

Representing the section:

- at the group meeting;
- when communicating with the human resources manager about replacement staff;
- when explaining about the delay to users.

Controlling, innovating, directing:

- deciding what needs to be done to make good the progress that will be lost through temporarily losing a member of staff.

Staffing:

- deciding which member of staff is to do what;
- discussion with human resources about the requirement for temporary staff;
- planning staffing for the next year.

Note: the same activity can involve many different roles.

1.10 Collecting control data

The project seems to have two major components: training and document transfer. If trainers were expected to tour offices giving training then one would expect there to be a schedule indicating when each office was to receive training. The following information about the progress of the information might therefore be collected:

- the number of offices that had received training – this could be compared with the schedule;
- the number of staff who had received training – to ensure that all staff were attending;
- feedback from staff on the perceived quality of training – for example, by post-training evaluation forms.

For the document transfer aspect, the following might be usefully collected for each office on a regular basis during the transfer process:

- number of documents transferred;
- estimated number of documents still needing to be transferred;
- number of staff-hours spent on transferring documents – to monitor the budget and transfer productivity;
- number of staff involved in the transfer.

When all documents have been transferred, performance tests to check response times might be required.

Chapter 2

2.1 Costs and benefits for the Brightmouth College payroll system

Table B.1 lists costs and benefits for the proposed Brightmouth HE College payroll system. It is not comprehensive but illustrates some of the types of items that you should have listed.

Category	Cost/benefit
Development costs	Software purchase – software cost plus selection and purchasing cost Project team employment costs
Setup costs	Training includes costs of trainers and operational staff time lost while training Staff recruitment Computer hardware and other equipment which might have a residual value at end of projected life Accommodation – any new/refurbished accommodation and furniture required to house new system Initial systems supplies – purchase of stationery, disks and other consumables
Operational costs	Operations staff – full employment costs Stationery – purchase and storage Maintenance and standby – contract or estimation of occurrence costs Accommodation, including heating, power, insurance etc.
Quantified and valued	Saving on local authority fees Later payment – increase interest income through paying salaries later in the month
Quantified but not valued	Improved accuracy – the number of errors needing to be corrected each month
Identified but not easily valued	Improved management information – this should lead to improved decision making but it is very difficult to quantify the potential benefits

TABLE B.1 Costs and benefits for the Brightmouth College payroll system

2.2 Ranking project cash flows

Obviously you will have your own views about which have the best and worst cash flows. You should, however, have considered the following points: project 2 requires a very large investment compared to its gain – in fact we could obtain £100,000 by undertaking both projects 1 and 3 for a lower cost than project 2. Both projects 1 and 4 produce the bulk of their incomes relatively late in their lives compared with project 3, which produces a steady income over its life.

2.3 Calculating payback periods

The payback periods for each of the projects will occur during the year indicated: project 1, year 5; project 2, year 5; project 3, year 4 and project 4, at the end of year 4.

We would therefore favour project 3 or 4 over the other two. Note that, in reality, with relatively short-term projects such as these we would produce a monthly (or at least

quarterly) cash flow forecast and it is therefore likely that project 3 would be seen more clearly to have a shorter payback period than project 4.

2.4 Calculating the return on investment

The return on investments for each of the projects is: project 1: 10%, project 2: 2%, project 3: 10% and project 4: 12.5%. Project 4 therefore stands out as being the most beneficial as it earns the highest return.

2.5 Calculating the net present value

The net present value for each of the projects is calculated as in Table B.2. On the basis of net present value, project 4 clearly provides the greatest return and project 2 is clearly not worth considering.

Year	Discount factor	Discounted cash flow (£)		
		Project 2	Project 3	Project 4
0	1.00	-1,000,000	-100,000	-120,000
1	0.90	181,820	27,273	27,273
2	0.82	165,280	24,792	24,792
3	0.75	150,260	22,539	22,539
4	0.68	136,600	20,490	20,490
5	0.62	186,270	18,627	46,568
NPV		-179,770	13,721	21,662

TABLE B.2 Calculating the net present value of projects 2, 3 and 4

2.6 Calculating the effect of discount rates on NPV

Table B.3 illustrates the effect of varying discount rates on the NPV. In each case the 'best' project is indicated in bold. In this somewhat artificial example, which project is best is very sensitive to the chosen discount rate. In such a case we must either have a very strong reason to use a particular discount rate or take other criteria into account when choosing among the projects.

2.7 Project evaluation using cost-benefit analysis

Expected sales of £500,000 per year over four years would generate an expected net income of £1.2m (after allowing for annual costs of £200,000), which, by almost any criteria, would provide a good return on an investment of £750,000. However, if sales

are low, and there is a 30% chance of this happening, the company will lose money – it is unlikely that any company would wish to take such a risk knowingly.

This example illustrates one of the basic objections to using this approach for one-off decisions. Were we to repeat the project a large number of times we would expect, *on average*, an income of £500,000 per annum. However, the company is developing this package only once – they can't keep trying in the hope of, *on average*, generating a respectable income. Indeed, a severe loss on this project could mean it is the last project they are able to undertake.

Year	Cash flow values (£)		
	Project A	Project B	Project C
0	-8,000	-8,000	-10,000
1	4,000	1,000	2,000
2	4,000	2,000	2,000
3	2,000	4,000	6,000
4	1,000	3,000	2,000
5	500	9,000	2,000
6	500	-6,000	2,000
Net Profit	£4,000	£5,000	£6,000
NPV @ 8%	£2,111	£2,365	£2,421
NPV @ 10%	£1,720	£1,818	£1,716
NPV @ 12%	£1,356	£1,308	£1,070

TABLE B.3 The effect on net present value of varying the discount rate

Chapter 3

3.1 External stakeholders in IOE accounts system

The main stakeholders who need to be considered are the IOE customers. It will be worth consulting some representative customers about the attractiveness of the new annual maintenance contract scheme. IOE might have a partnership arrangement with the manufacturers of the equipment it maintains whereby it is recognized as approved to carry out repairs. It might therefore need to consult the equipment providers about the new scheme. The suppliers might, for example, be willing to promote the scheme on a commission basis. It is possible with annual maintenance contract schemes of this nature to outsource their financing to an insurance company. Essentially, in return for an annual premium, the insurance company would pay IOE every time a maintenance job is carried out under this scheme.

3.2 Product description for acceptance test cases

An example of a possible product description for acceptance test cases is shown below:

Name/identity	Acceptance test case
Purpose	To record the individual tests that will be carried out during the acceptance testing. It will ensure that testing is comprehensive i.e. that all user requirements are tested.
Derivation	The user requirements report
Composition	For each test case the following will be recorded: (i) cross-reference to user requirements; (ii) preconditions – including items that would need to be set up on the database before the test can be executed; (iii) input data; (iv) expected results.
Form	A word-processed document created using a template
Quality criteria	Independently reviewed against the requirements document to ensure that all requirements are covered. Internal consistency checked, including whether pre-conditions are complete and expected results correctly calculated.

Note: Other products – such as a testing plan – would also have to be created in order to document the acceptance testing phase.

3.3 Creating an invitation to tender (ITT) – Product Flow Diagram

Figure B.1 illustrates a Product Flow Diagram for the products needed to create an invitation to tender for Brightmouth College payroll.

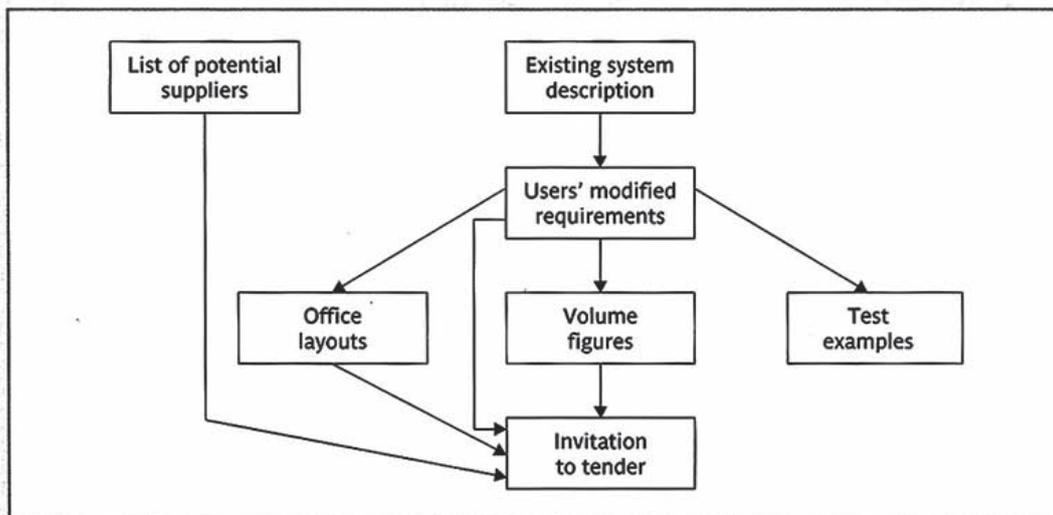


FIGURE B.1 Product Flow Diagram for the creation of an 'invitation to tender'

Different PFDs could be produced depending on the policy decisions made about how the process is to be carried out. This is one way in which it could be done. A person acting as an analyst investigates the current way of doing payroll in order to find out the basic functions that the new system must have. This document may prompt the user management to come up with some new functions that would let them do things with the new system that they could not do before. Once it is known, for example, what types of record the new system will hold, and the functions the new system will have, the size of the database and the number and size of transactions to be carried out can then be estimated. This will indicate the size and power needed for the hardware platform on which the application will run. The hardware will need to be housed within a particular physical layout governed by the office space available at the college and contractors may need to take this into account. The invitations to tender (ITTs) will need to be sent to suitable potential suppliers and some research will be needed to decide who these suppliers will be. The documented requirements are the basis for a set of procedures to evaluate the proposals, including some test cases.

3.4 Invitation to tender activity network

Figure B.2 illustrates the activity network, showing the activities needed to create an invitation to tender for the Brightmouth College payroll.

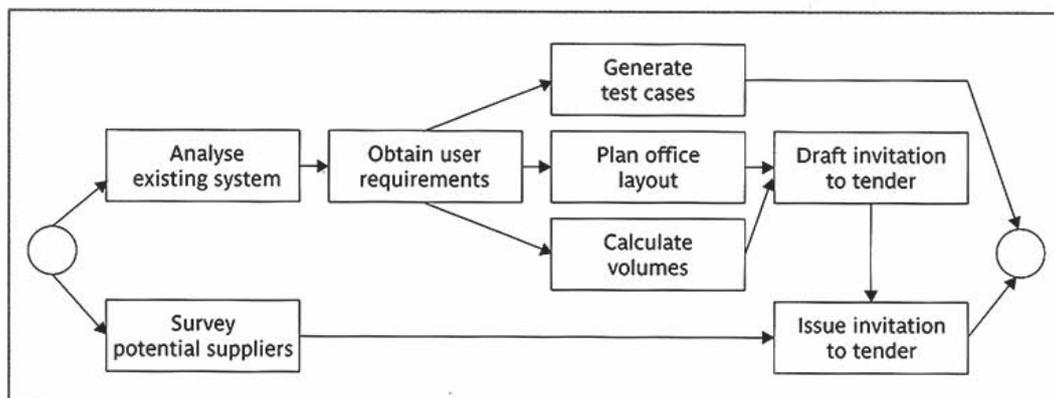


FIGURE B.2 Brightmouth College payroll project activity network fragment

3.5 Including a checkpoint

Figure B.3 illustrates the inclusion of a checkpoint in Amanda's activity network.

3.6 Quality checks on user requirements

The users will need at least to read and approve the system specification. This might be rather late to make major changes, so user approval of earlier documents such as interview notes would be helpful.

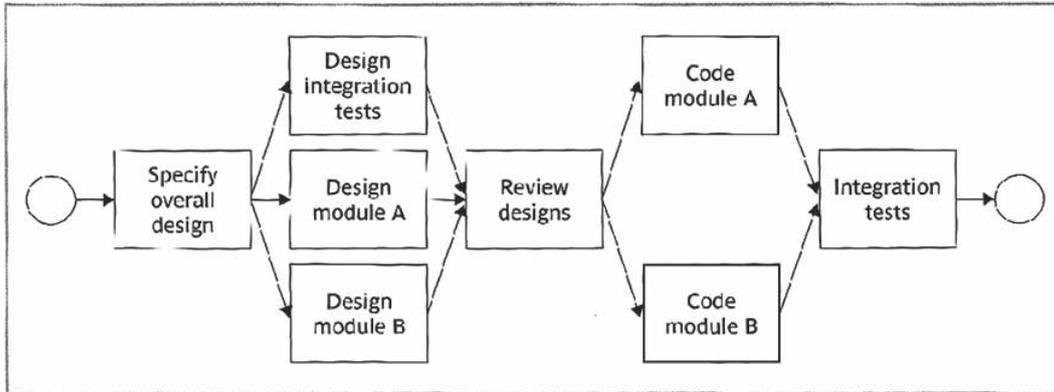


FIGURE B.3 An activity network for a software development task, modified to include a checkpoint

3.7 Cross-reference of a planning document to Step Wise activities

Table B.4 suggests the Step Wise activities needed to create the different sections of a plan.

Section of plan	Step Wise activities
Introduction	
Background	1.3 Identify stakeholders 2.1 Establish relationship between project and strategic planning
Project objectives	1.1 Identify objectives and measures of effectiveness 1.4 Modify objectives in the light of stakeholder analysis
Constraints	1.1 Identify objectives and measures of effectiveness 2.2 Identify installation standards and procedures
Methods	3. Analyse project characteristics 4.2 Document generic product flows (this could help establish a methodology)
Project products	4.1 Identify and describe project products
Activities to be carried out	4.4 Produce ideal activity network 4.5 Modify ideal activity network
Resources to be used	3.6 Review overall resource estimates 5.1 Carry out bottom-up estimates 7. Allocate resources
Risks to the project	3.3 Identify high-level project risks 6. Identify activity risks
Management of the project	1.2 Establish project authority 1.5 Establish methods of communication with all parties 2.3 Identify project team organization

TABLE B.4 Sections of a plan cross-referenced to Step Wise activities

Chapter 4

4.1 Classification of systems

- (a) A payroll system is a data-oriented or information system that is application specific.
- (b) The bottling plant system is a process control or industrial system which contains embedded software.
- (c) This looks like an information system that may make use of computer graphics. The plant itself might use control software which might be safety critical but this is not the subject of the project under consideration.
- (d) Project management software tools are often categorized as general packages. There would be a considerable information systems element to them.
- (e) This could use an information retrieval package that is a general software package. It is also a strong candidate for a knowledge-based system.

4.2 Identification of risks

The user staff could, arguably, be regarded as a project resource. The writers' view is that it is useful to add a fourth category of risk – those belonging to the *environment* in which the system is to be implemented.

Among the risks that might be identified at Brightmouth College are:

- conflict of views between the finance and personnel departments;
- lack of staff acceptance for the system, especially among personnel staff;
- lack of cooperation by the local authority that used to carry out payroll work;
- lack of experience with running payroll at the college, leading to errors and delays in processing;
- lack of administrative computing expertise at the college;
- possible inadequacy of the chosen hardware;
- changes to the payroll requirements.

4.3 Selection of project approaches

- (a) This would appear to be a knowledge-based system that is also safety critical. Techniques associated with knowledge-based systems could be used for constructing the system. Testing would need to be very carefully conducted. A lengthy parallel run where the system is used to shadow the human decisions made in real cases and the results compared could be considered. Another approach would be to develop two or more systems in parallel so that the advice offered could be cross-checked.
- (b) This is an information system that will be on a relatively large scale. A structured approach designed for information systems applications, such as SSADM, would be justified. When student loans were first introduced there was no existing system and so there might have been some scope for a prototype.

- (c) This is an embedded system that is safety-critical. Measures that might ensure the reliability of the system include:
- use of mathematics-based specification languages to avoid ambiguity;
 - developing parallel versions of the same software so that they can be cross-checked;
 - statistical control of software testing to allow for the estimation of the reliability of the software.

4.4 Stages of a project where a prototype can be appropriate

- (a) A prototype could be useful as part of the feasibility study. A mock-up of an executive information system loaded with current management information could be set up manually and then be tried out by the managers to see how easy and useful they found it.
- (b) A prototype could be used to assist in the design of the user dialogues. Structured approaches like SSADM often allow for prototypes for this purpose as part of requirement specification.
- (c) A prototype of the most response-critical transactions could be made at the physical design stage to see whether Microsoft Access could produce software that gave a satisfactory performance.

Chapter 5

5.1 Calculating productivity rates and using productivity rates to project effort

Tables B.5 and B.6 illustrate productivity rates and estimated project effort.

Project	Work-months	SLOC	Productivity (SLOC/month)
a	16.7	6,050	362
b	22.6	8,363	370
c	32.2	13,334	414
d	3.9	5,942	1,524
e	17.3	3,315	192
f	67.7	38,988	576
g	10.1	38,614	3,823
h	19.3	12,762	661
i	59.5	26,500	445
Overall	249.3	153,868	617

TABLE B.5 Productivity rates

Project	Estimated work-months	Actual	Difference
a	$6050/617 = 9.80$	16.7	6.90
d	$5942/617 = 9.63$	3.9	-5.73

TABLE B.6 Estimated effort

There would be an under-estimate of 6.9 work-months for project a and an over-estimate of 5.7 for project d.

5.2 Agile methods and the problems of estimating

Points that might be discussed include the following.

- *Diseconomies of scale with larger projects.* It is recommended that the programming team does not contain more than ten people in order assist easy team communication.
- *Threats to quality of tight deadlines.* Time-boxing can help here. There are four sets of project outcomes that can be traded off: scope of the functionality, quality, project duration and cost. The XP approach argues that quality, project duration, and cost can be controlled by the business management, but scope must be controlled by the development team. If the project comes under time pressure, some low-priority deliverables may need to be held over to the next release, but something will still be released on time, and the quality of this will not have been compromised.
- *Substandard work not being apparent until late in the project.* Testing is done as an integral part of the design/code process and is not put off as a task to be done by another group, such as a system testing group, at a later stage in the project.

5.3 Course staff costs program – activities required

A list of activities might include:

- obtaining user requirements;
- analysis of the structure of the data already held;
- design of the report layout;
- writing the user proposal;
- planning test cases;
- technical specification;
- design of the software structure;
- software coding;
- testing software;
- writing the operating instruction;
- acceptance testing.

The most difficult tasks to estimate are often those that are most sensitive to the size and the complexity of the software to be produced, in this case the design, writing and testing of the software. Writing the technical specification can also be difficult because of this, but estimating problems tend to be concealed here as deadlines can be met by omitting detail that can be added later when deficiencies are found.

The duration of activities that are to be carried out by users may also present problems, as this might depend upon their sense of priorities.

5.4 SLOC estimate for customer insertion program

Figure B.4 gives an outline program structure using a Jackson structured diagram. The numbers in circles are our estimates of the lines of 'generic' code needed to implement each sub-process in the program. They should add up to 95 SLOC.

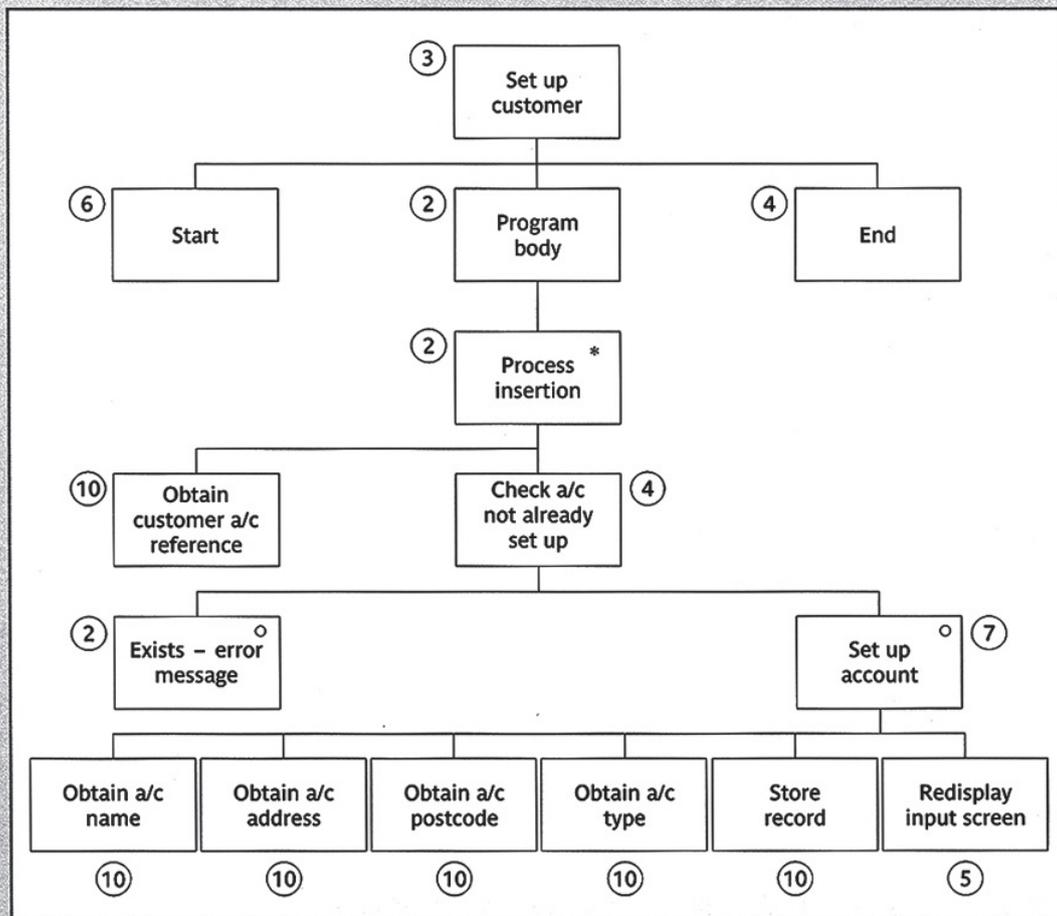


FIGURE B.4 Outline program structure for 'set up customer' transaction
 * means the process is repeated; o means the processes are alternatives.

5.5 Effort drivers for a student assignment

The most obvious effort driver would seem to be the number of words required. Difficulty factors might include:

- *availability of material*, for example in the library;
- *familiarity* of the student with the topic;
- *breadth/depth required*, that is, a broad survey of a wide field or an in-depth study of a narrow area;
- *technical difficulty* – some topics are easier to explain than others.

It could be argued that time available is the constraint. The student just does what can be done in the time available (see 'design to cost').

5.6 Calculating Euclidean distance

The Euclidean distance between project B and the target case is the square root of $((7 - 5)^2 + (15 - 10)^2)$, that is, 5.39. Project A is therefore a closer analogy.

5.7 Albrecht function points

External input types	none
External output types	the report, that is, 1
Logical internal file types	none
External interface file types	payroll file, staff file (timetabling), courses file (timetabling), that is, 3
External inquiry types	none

The function point counts are as follows:

External enquiry types	none
External output types	$1 \times 7 = 7$
Logical internal file types	none
External interface types	$3 \times 7 = 21$
External inquiry types	none
Total	28

5.8 Calculation of SLOC from Albrecht function points

The estimated lines of Java = $28 \times 53 = 1484$. With a productivity rate of 50 SLOC per day, this gives an estimated effort of $1484/50$, that is, approximately 30 days.

5.9 Mark II function points

The function types are:

Input data types	6
Entities accessed	1
Output data types	1

Unadjusted function points = $(0.58 \times 6) + (1.66 \times 1) + (0.26 \times 1) = 5.4$

5.10 Data movements

Data movement	Type
Incoming vehicle sensed	E
Access vehicle count	R
Signal barrier to be lifted	X
Increment vehicle count	W
Outgoing vehicle sensed	E
Decrement vehicle count	W
New maximum input	E
Set new maximum	W
Adjust current vehicle count	E
Record adjusted vehicle count	W

Note that different interpretations of the requirements could lead to different counts. The description in the exercise does not, for example, specify that the system should output a message that the car park is full or has spaces, although this might be expected.

5.11 COCOMO – calculating the exponent scale factors

Table B.7 shows scale factors for the example.

Factor	Rating	Value
PREC	nominal	3.72
FLEX	high	2.03
RESL	very low	7.07
TEAM	very high	1.10
PMAT	low	6.24

TABLE B.7 Assessing the scale factors

- (i) The overall scale factor would be $0.91 + 0.01 \times (3.72 + 2.03 + 7.07 + 1.10 + 6.24)$
 $= 0.91 + 0.01 \times 20.16$
 $= 1.112$
- (ii) The estimated effort would be $2.94 \times 2^{1.112} = 6.35$ staff-months

5.12 COCOMO II Applying effort multipliers

Factor	Description	Rating	Effort multiplier
RCPX	product reliability and complexity	vh	1.91
RUSE	reuse	vh	1.15
PDIF	platform difficulty	l	0.87
PERS	personnel capability	vh	0.63
PREX	personnel experience	nominal	1.00
FCIL	facilities	nominal	1.00
SCED	required development schedule	nominal	1.00

TABLE B.8 Effort multipliers

The combined effort modifier would be

$$(1.91 \times 1.15 \times 0.87 \times 0.63 \times 1.00 \times 1.00 \times 1.00) = 1.20$$

The modified estimate would be $200 \times 1.20 = 240$ staff months

Chapter 6

6.1 Drawing a CPM network

A solution is given in Figure 6.14. If your solution is not exactly the same as this, do not worry. Just check that it is *logically* the same and that it follows the precedence network conventions of layout and labelling etc.

6.2 The precedence network

Figure B.5 illustrates a precedence network for Amanda's project, showing an earliest completion date of day 104.

6.3 Calculating activity floats

Free float and interfering float for each of the activities are shown in Table B.9. Note that activity A has no free float since any delay in its completion will delay the start of activity C. Float must be regularly monitored as a project progresses since a delay in any activity beyond its free float allowance will eat into the float of subsequent activities.

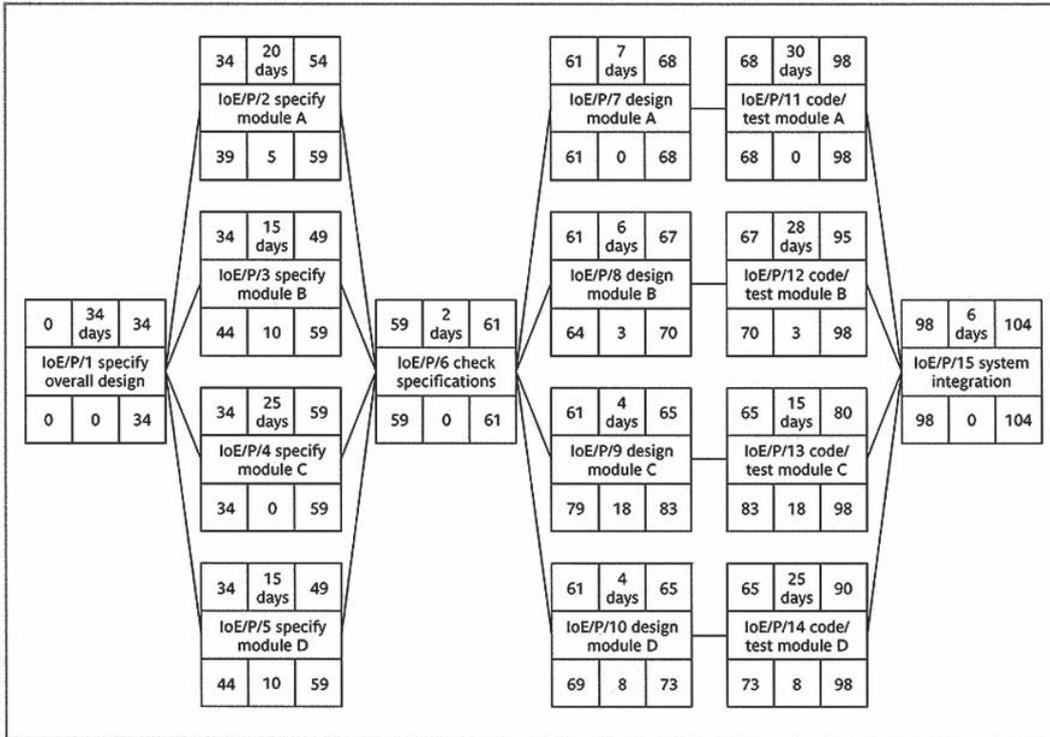


FIGURE B.5 Amanda's precedence network

Activity	Total float	Free float	Interfering float
A	2	0	2
B	3	0	3
C	2	0	2
D	3	1	2
E	3	3	0
F	0	0	0
G	0	0	0
H	2	2	0

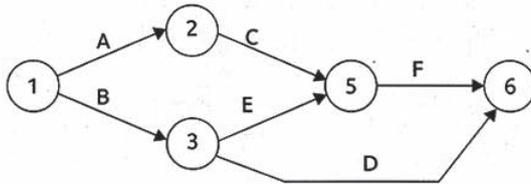
TABLE B.9 Activity floats

6.4 Shortening a project duration

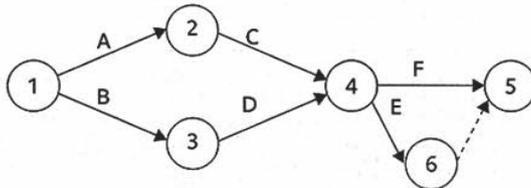
Shortening activity F to 8 weeks will bring the project completion date forward to week 11 – that is, it will save 2 weeks on the duration of the project. However, there are now two critical paths, start–F–G–finish and start–A–C–H–finish, so that reducing the duration of activity F any further will not shorten the project duration any further. If we wish to complete the project earlier than week 11 we must save time on both of these critical paths.

6.5 Errors drawing activity networks

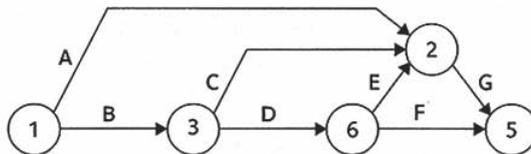
- (a) Activity D dangles, giving the project two 'end events'. This network should be drawn as below. To aid comparison with the original, the nodes have not been renumbered, although we would normally do so.



- (b) Once again, this network has two end nodes, but in this case the solution is slightly different since we should introduce a dummy activity if we are to follow the standard CPM conventions.



- (c) Either this one has a dangle (although, because of the way it is drawn, it is less obvious) or activity E has its arrow pointing in the wrong direction. We need a bit more information before we can redraw this one correctly.
- (d) Strictly speaking, there is nothing wrong with this one – it is just badly drawn and the nodes are not numbered according to the standard conventions. It should be redrawn as in the following example.



In this diagram the nodes have retained their original numbers (to aid identification), although they should of course be renumbered sequentially from left to right.

- (e) This one contains a loop – F cannot start before G has finished, G cannot start before E has finished and E cannot start before G has finished. One of the arrows is wrong! It is probably activity F that is wrong but we cannot be sure without further information.

6.6 Drawing Brigitte's activity network as a CPM network

Brigitte's payroll CPM network should look like the diagram shown in Figure B.6. If your diagram is not exactly the same as this, check that it is logically the same.

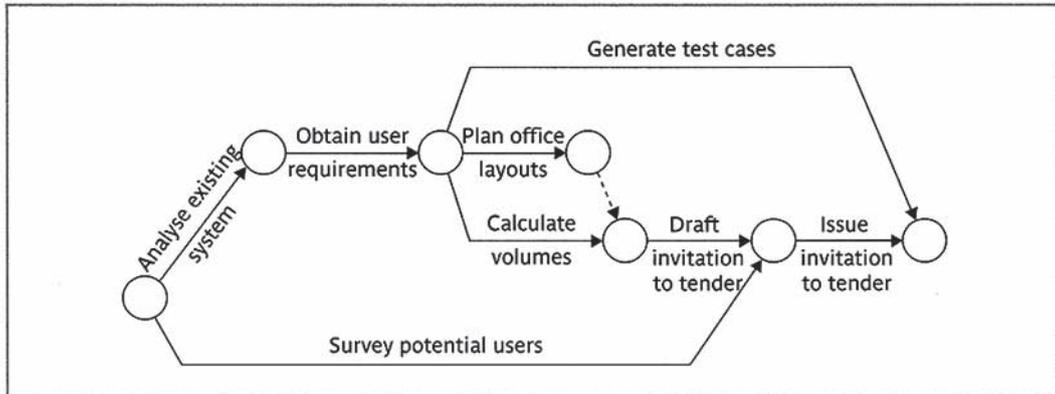


FIGURE B.6 Brigitte's CPM network

Chapter 7

7.1 Matching causes and effects

There is no one correct answer to this. An example of a possible answer is provided below.

- (a) i and ii. Staff inexperience leads to code that has many errors in it and which therefore needs additional testing time. Inexperienced staff will take longer to carry out development in any case.
- (b) iv. If top management do not have a strong commitment to the project they will not act with a sense of urgency.
- (c) ii. New technology takes time to get used to.
- (d) iii. If users are uncertain of their requirements then they are likely to identify new requirements as the project progresses.

Exercise 7.2 Identifying risks

Once again the answer below can only be indicative – there is no one correct answer.

Domain	IOE	Brightmouth payroll
Actors	Possible user resistance – see the Section 3.5 case study example	Lack of experience running payroll – see answer to Exercise 4.2
Structure	Not all stakeholders are represented on the Project Board – see Section 3.3	Lack of cooperation between the local authority and the college
Tasks	Uncertainty about time needed to change existing software – see Section 3.8	Evaluation of packages – software may be difficult to access to carry out evaluation testing
Technology	Existing hardware is not adequate to deal with new application	Existing hardware is not adequate to deal with new application

Exercise 7.3 Conditions needed for successful pooling arrangement

Among the conditions would be the following.

- The chance of fire is precisely 1 in 1000. As this is only an estimate of an average, this could not be guaranteed. If a fire happened at a second location, the pool would already have been exhausted. Having a larger number of contributors and a larger pool would reduce, but not eliminate, this risk.
- The sites would have to be at completely different locations so that a fire at one site does not affect the others.
- Each site has the same chance of fire. If the people at a site were aware that the chance of fire was a lot less in their location, they might object to having to effectively subsidize other sites.
- The amount of damage caused is always the same.

Exercise 7.4 Pre-conditions to facilitate contingency actions

Staff illness is just one of several reasons why you might need to transfer staff between job roles in the middle of an activity. Such transfers would be made easier if:

- there was a standard methodology for the way that the work was carried out;
- intermediate steps were well documented;
- other staff were involved in reviewing products at regular intervals;
- job descriptions were flexible.

It is interesting to note that in an extreme programming environment, the recommended approach of pair programming should provide an alternative way of dealing with this problem.

The factors to be taken account of could include costs and human factors. The structured approach to development that the bullet points above imply would involve costs in selecting the right methodology, training and other aspects of implementation and management of the process to ensure that staff adhere to the requirements of the methodology. Very flexible staffing arrangements where staff could be switched between jobs at short notice could have implications for morale (which might be positive as well as negative).

7.5 Calculating expected activity durations

Table B.10 shows the activity duration estimates from Table 7.6 along with the calculated expected durations, t_e .

7.6 The forward pass to calculate expected completion date

The expected duration and the expected dates for the other project events are shown in Figure 7.6. An expected duration of 13.5 weeks means that we expect the project to be completed halfway through week 14, although since this is only an expected value it could finish earlier or later.

Activity	Activity durations (weeks)			
	Optimistic (<i>a</i>)	Most likely (<i>m</i>)	Pessimistic (<i>b</i>)	Expected (<i>t_e</i>)
A	5	6	8	6.17
B	3	4	5	4.00
C	2	3	3	2.83
D	3.5	4	5	4.08
E	1	3	4	2.83
F	8	10	15	10.50
G	2	3	4	3.00
H	2	2	2.5	2.08

TABLE B.10 Calculating expected activity durations

7.7 Calculating standard deviations

The correct values are shown in Figure 7.7. Brief calculations for events 4 and 6 are given here.

Event 4: Path A + C has a standard deviation of $\sqrt{(0.50^2 + 0.17^2)} = 0.53$

Path B + D has a standard deviation of $\sqrt{(0.33^2 + 0.25^2)} = 0.41$

Node 4 therefore has a standard deviation of 0.53.

Event 6: Path 4 + H has a standard deviation of $\sqrt{(0.53^2 + 0.08^2)} = 0.54$

Path 5 + G has a standard deviation of $\sqrt{(1.17^2 + 0.33^2)} = 1.22$

Node 6 therefore has a standard deviation of 1.22.

7.8 Calculating z values

The z value for event 5 is $\frac{10 - 10.5}{1.17} = -0.43$, for event 6 it is $\frac{15 - 13.5}{1.22} = 1.23$.

7.9 Obtaining probabilities

Event 4: The z value is 1.89 which equates to a probability of approximately 3%. There is therefore only a 3% chance that we will not achieve this event by the target date of the end of week 10.

Event 5: The z value is -0.43 which equates to a probability of approximately 67%. There is therefore a 67% chance that we will not achieve this event by the target date of the end of week 10.

To calculate the probability of completing the project by week 14 we need to calculate a new z value for event 6 using a target date of 14. This new z value is

$$z = \frac{14 - 13.5}{1.22} = 0.41$$

This equates to a probability of approximately 35%. This is the probability of not meeting the target date. The probability of meeting the target date is therefore 65% (100% – 35%).

Chapter 8

8.1 Smoothing resource demand

Smoothing analyst/designer demand for stage 4 is reasonably easy. The design of module D could be scheduled after the design of module C. Stage 2 is more problematic as scheduling the specification of module D to start after the completion of B would delay the project. Amanda might consider doing this if whoever is specifying module A could also be allocated to module D for the last six days – although she may well decide that drafting an extra person into a specification activity is unsatisfactory.

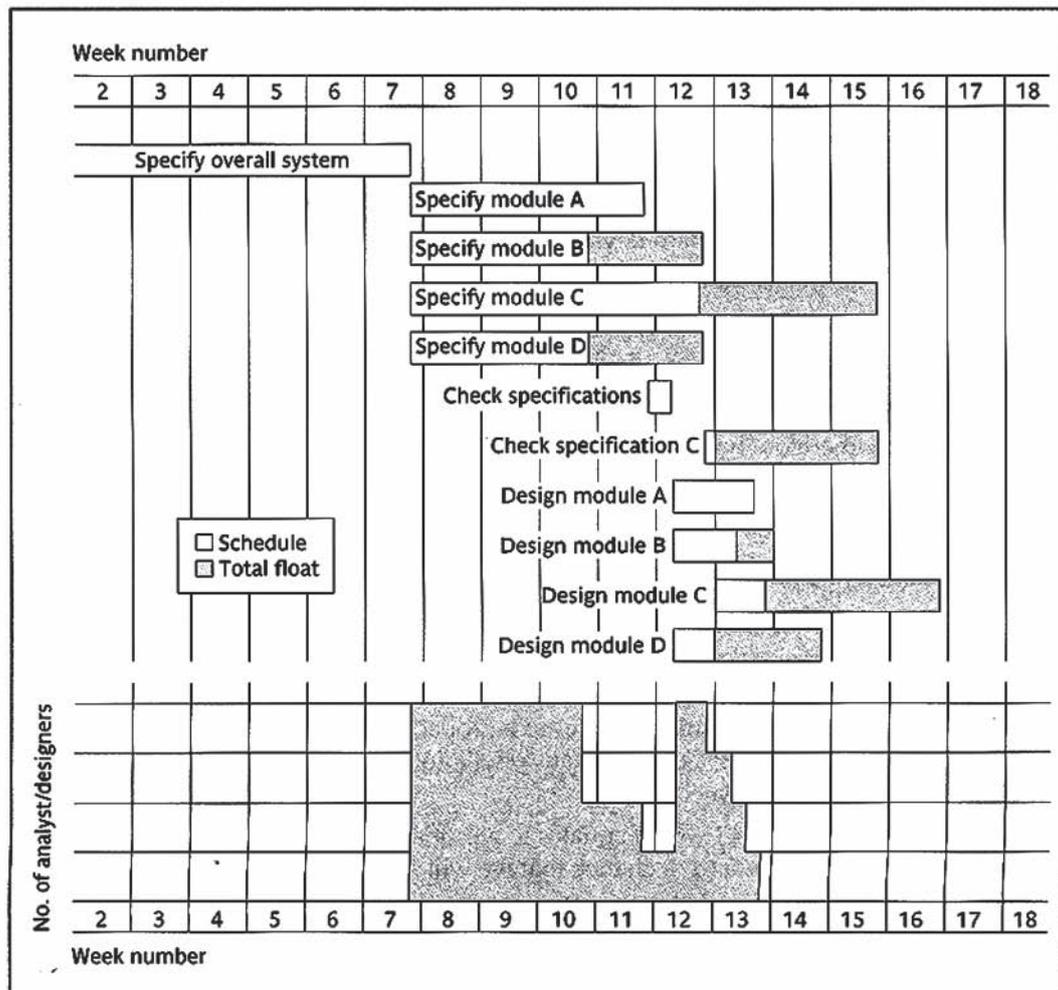


FIGURE B.7 Amanda's revised bar chart and resource histogram

8.2 Drawing a revised resource histogram

In Figure B.7 activities start at their earliest dates. The shaded area of each bar is the activity's total float. If an activity starts later than its earliest date, part of its float is 'used up'. In Figure B.8 this is shown by the shaded area of some bars moving to the left.

If the activities are scheduled at the earliest dates, then the plan still calls for four analyst/designers as shown in Figure B.7. By delaying the start of some activities, however, Amanda is able to ensure that using three analyst/designers is sufficient except for a single day. This is shown in Figure B.8.

Note that if the specification of module C were to be delayed for a further day, the project could be completed with only three analyst/designers, although its completion day would, of course, be delayed.

with only three analyst/designers, although its completion day would, of course, be delayed.

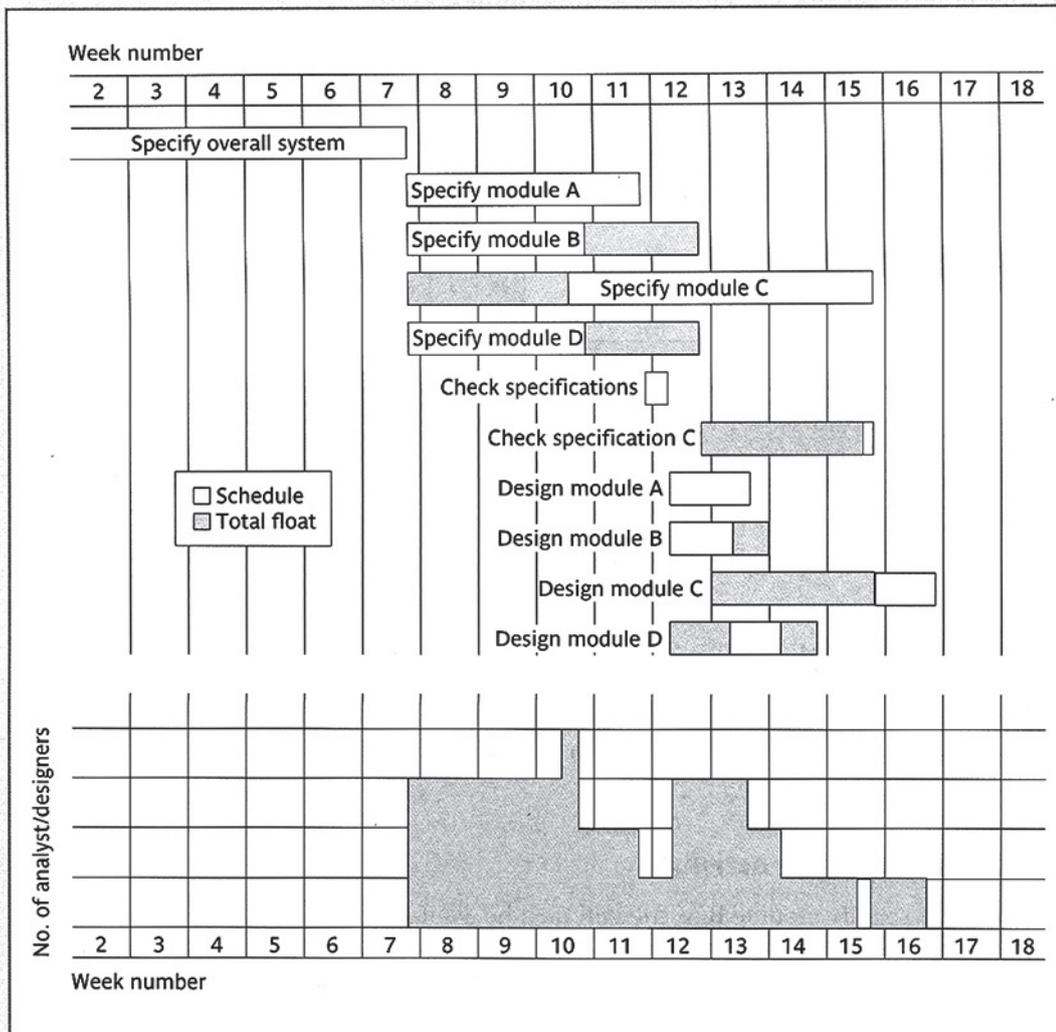


FIGURE B.8 The effect of delaying some activity starts

8.3 Identifying critical activities

The critical path is now as shown in Figure B.9. Note the lag of 15 days against activity IoE/P/4, ensuring that its start is delayed until an analyst/designer is expected to be available.

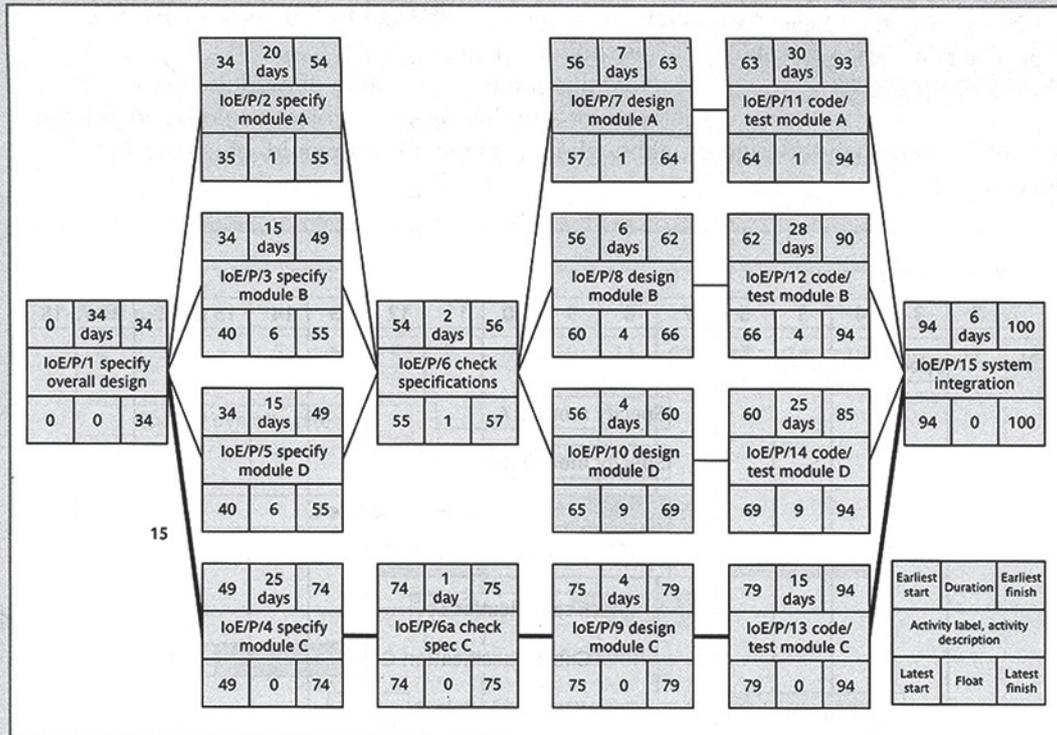


FIGURE B.9 The critical activities after delaying the start of module C

However, the availability of an analyst/designer for IoE/P/4 is dependent upon IoE/P/3 or IoE/P/5 being completed on time – these two activities are therefore also now critical in the sense that a delay in both of them would delay IoE/P/4, which is on the normal critical path. These two activities, although not on the critical path, are, in that sense, critical.

8.4 Assigning staff to activities

Belinda must specify module B as she will then be available in time to start the specification of module C. This leaves Daisy for the specification and design of module A. Belinda cannot do the design of module B as she will still be working on the module C specification when this needs to be done (6 days between days 56 and 66). This will have to be left to Tom, as he should be free on day 60.

Can you think of any other way in which she might have allocated the three team members to these activities?

8.5 Calculating project costs

The easiest way to calculate the total cost is to set up a table similar to Table B.11.

Calculating the distribution of costs over the life of the project is best done as a per week or per month figure rather than as daily costs. The expenditure per week for Amanda's project is shown as a chart in Figure 8.9.

Analyst	Daily cost (£)	Days required	Cost (£)
Amanda	300	110 ²	33,000
Belinda	250	50	12,500
Tom	175	25	4,375
Daisy	225	27	6,075
Gavin	150	30	4,500
Purdy	150	28	4,200
Justin	150	15	2,250
Spencer	150	25	3,750
Daily on-cost	200	100	20,000
Total			90,650

TABLE B.11 Calculating the cost of Amanda's project

² This includes 10 days for pre-project planning and post-project review.

Chapter 9

9.1 Lines of code as a partial task completion indicator

There are many reasons why the proportion of lines coded is not a good indicator of completeness. In particular, you should have considered the following:

- the estimated total number of lines of code might be inaccurate;
- the lines of code so far might have been easier, or harder, than those to follow – for example, reuse of existing components might speed up development;
- a program is not generally considered complete until it has been tested – testing and debugging the code could take considerable time once the code has been written.

With more knowledge of what has been done and what is left to complete it might be possible to make a reasonable estimate of completeness. Breaking the development task into smaller sub-tasks such as software design, coding and unit testing might be of some assistance here.

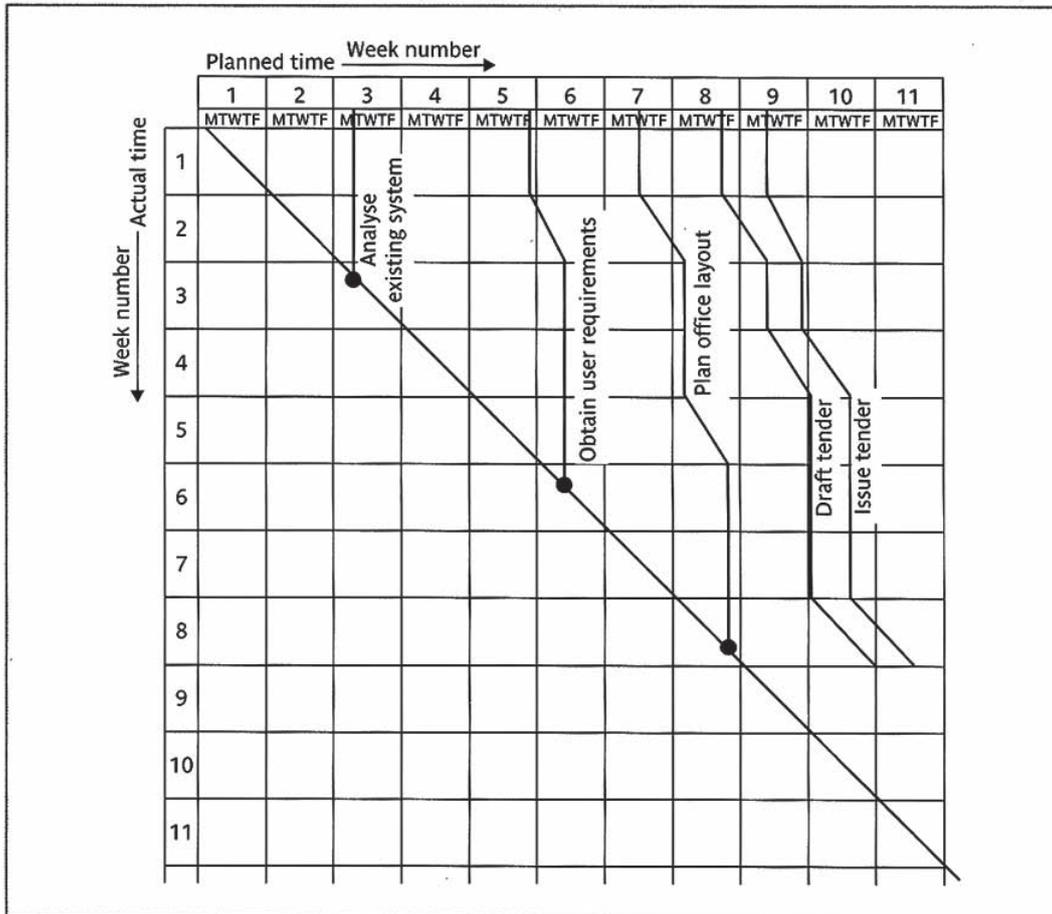


FIGURE B.10 The revised timeline chart

9.2 Revising the timeline chart

At the end of week 8, the scheduled completion dates for drafting and issuing the tender need to be revised – note that both need to be changed since they are both on the critical path (Figure B.10).

Subsequently, Brigitte needs to show only the completion of each of these two remaining activities on the timeline chart – the project being completed by the Thursday of week 11 (Figure B.11).

9.3 Amanda's earned value analysis

It should be apparent from Figure 9.11 that the initial activity, 'specify overall system', has slipped by one day. It may not be quite so obvious from Figure 9.11 alone what else has happened to her project – inspection of Figure 9.11 and Table 9.2 should, however,

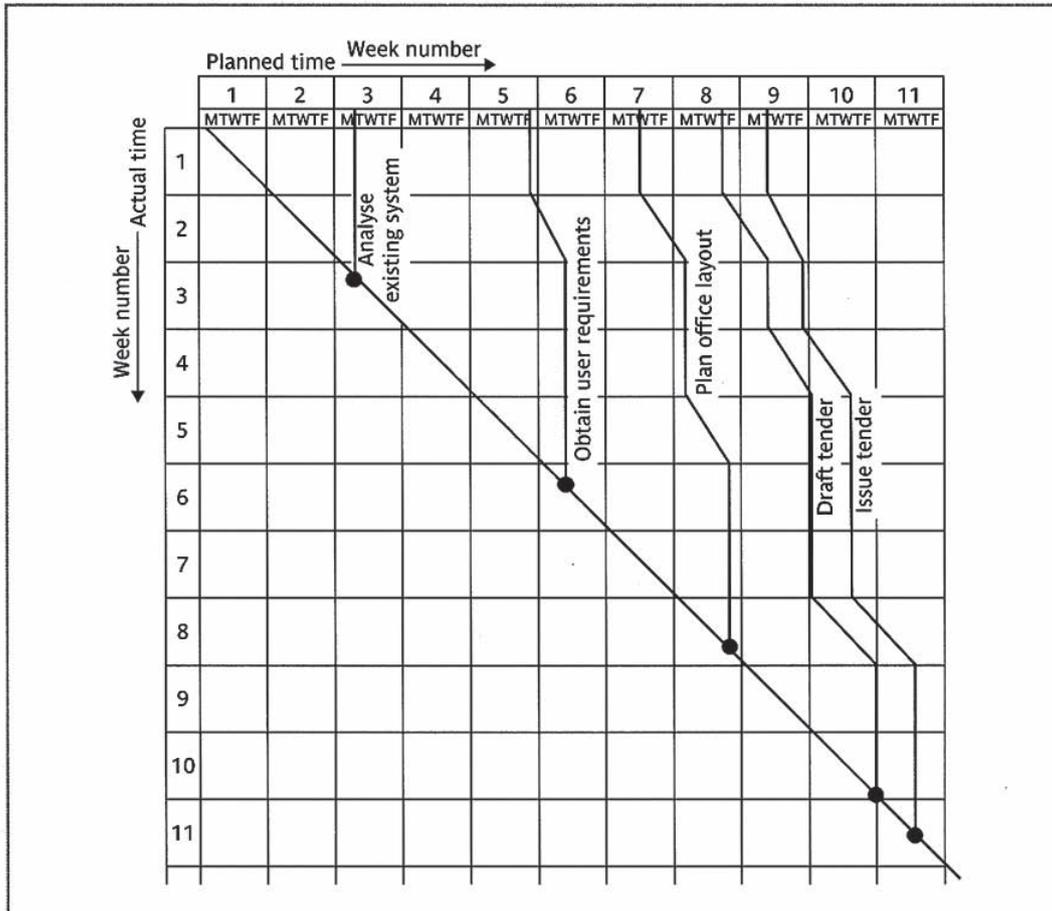


FIGURE B.11 The completed timeline chart

make it possible to deduce that one of the activities 'specify module B' and 'specify module D' has taken 2 days longer than forecast and the other has taken 5 days longer. In addition, 'specify module A' should have been completed by day 54 but has not. Thus, the project has earned 34 workdays by day 35, 49 workdays by day 52 and 64 workdays by day 55.

From Figure 9.11 it is not possible to deduce the underlying causes of the slippage or to forecast the consequences for the project. The use of earned value analysis for forecasting is described later in Section 9.6.

9.4 The effects of specification changes

Among the items most likely to be affected by the change are test data, expected results and the user handbooks.

9.5 Control procedures for development systems

Stages 1 to 6 will be basically the same except that an estimate on the effect of the project's timescale will need to be included in steps 3 and 4. Step 7 might not be required as system acceptance might not have taken place yet and acceptance testing of the changes will be included in that.

The release of software in step 8 will not be needed if the system is not yet operational, although master copies of products will need to be updated.

9.6 Reasons for scope creep

As well as user requests for extra features, developers will find that additional code may be needed to deal with exceptional circumstances that become apparent during detailed design. Additional functionality could also occur because of the need to coordinate components.

Chapter 10

10.1 Choice of type of package at IOE

The problem for Amanda at IOE would be that the new annual maintenance contracts subsystem would essentially be an extension to and enhancement of the existing maintenance accounting system, so that the interfacing of an off-the-shelf package might involve quite a few difficulties. This seems to indicate that bespoke development is needed. An alternative approach might be to consider replacing the whole of the maintenance accounting system with a new off-the-shelf application.

10.2 Calculation of charges for a project

For the first 2000 FPs	$\$967 \times 2,000$	= \$1,934,000
For the next 500 FPs	$\$1,019 \times 500$	= \$509,500
For the next 500 FPs	$\$1,058 \times 500$	= \$529,000
For the last 200 FPs	$\$1,094 \times 200$	= \$218,800
Charge for all 3,200 FPs		\$3,191,300

10.3 Calculating the cost of additional functionality

For changed FPs	$500 \times 600 \times (150/100)$	= \$450,000
For additional FPs	200×600	= \$120,000
Total charge		\$570,000

10.4 Advantage to customer of variable cost charges

The supplier will need to quote a price that will include a margin to cater for possible increases in equipment prices. It might turn out that actual prices do not increase as much as was estimated – in the case of ICT equipment some prices are likely to go down – but the customer would still have to pay the additional margin. If the contract specifies a fixed charge plus the actual cost of materials and equipment, then the customer in this case would be better off.

10.5 Calculating value for money

System X savings would be $£20 \times 20 \text{ hours} \times 4 \text{ years} = £1,600$, for the automatic scale point adjustment facility, and $£20 \times 12 \text{ hours} \times 2 \text{ times a year} \times 4 \text{ years} = £1,920$, for the bar-chart production facility. In total the saving for system X would be £3,520.

For system Y, the saving would be $£300 \times 0.5$ (to take account of the probability of change). That is, £150.

Even though system X costs £500 more, it will still give better value for money.

Note that discounted cash flow calculations could be applied to these figures.

10.6 Evaluation methods

- i.** The usability of an existing system could be evaluated by such means as the examination of user handbooks, the observation of demonstrations and practical user trails.
- ii.** This is clearly tricky. One would have to evaluate the methods that the developers intend to use to see whether they adhere to good interface design practice. One might also examine any interface standards that are in use by the supplier.
- iii.** Note that the question focuses on the costs of maintenance, rather than that of reliability. The cost of unexpected maintenance could be reduced, at least for a short time, by passing this risk to the supplier if there is a comprehensive warranty. The warranties provided by suppliers would therefore need to be scrutinized. Discussion with reference sites might also be helpful.
- iv.** Once again guarantees could be put in place by suppliers concerning this. The nature of these guarantees could be examined. Discussion with reference sites could once again be helpful.
- v.** Training materials could be examined. The training staff could be interviewed and their CVs examined. Reference sites that have already used the supplier's training services could be approached for their views.

Chapter 11

11.1 Tasks and responsibilities of an analyst/programmer

Analyst/programmers are expected to be able to carry out both analysis and programming tasks. It is likely, however, that the kinds of analysis tasks undertaken will be restricted. They may, for example, do the analysis work for enhancements to existing systems but not of completely new applications. Making this broad assumption, a list of tasks and responsibilities might be as follows:

- carry out detailed investigations of new requirements for existing computer applications;
- analyse the results of investigations and review the solutions to problems experienced, including the estimation of relevant costs;
- prepare systems specifications in accordance with organizational standards;
- conduct appropriate systems testing;
- prepare functional module specifications;
- produce and modify module structure diagrams;
- code and amend software modules;
- carry out appropriate unit testing;
- produce and amend user documentation;
- liaise with users, carrying out appropriate training in the use of computer applications where required.

11.2 Rewarding reuse

A problem here is that the software developers who make most use of reused components will, as a consequence, be producing less code themselves. You also want to encourage programmers to produce software components that other people can use: this might help the productivity of the organization but not that of the current project that they are working on!

You need to have a method, like function point analysis, which measures the functions and features actually delivered to the user. You also need to have some way of measuring the code used in the application that has been taken from elsewhere. Percentage targets of the amount of reused code to new code could be set and staff rewarded if the targets are met. As an alternative, the savings made by reuse could be measured and a profit-sharing scheme could be operated.

Programmers could be encouraged to produce and publish reusable components by a system of royalties for each time a software component is reused.

11.3 Financial incentives for top executives

This exercise was designed to be thought-provoking. Some thoughts that have come out of discussion on this topic are given below.

- To some extent, material wants and, therefore, the motivation to obtain more money to satisfy these wants can be generated through the marketing and advertising of new types of goods and services – but how likely is this to be at the very top?
- Large salaries are associated with status, esteem and success. It could be that these are the real reward.
- Historically, wealth has been associated with power, such as the ownership of land.

The essential point is that for many people money is not just a means of satisfying material wants.

11.4 High and low motivational incidents

This will obviously depend on individual experiences.

11.5 Possible objections to the stockholder ethical model

The purpose of this exercise is to stimulate debate, but some possible discussion points could be:

- The model implies that employees and customers exist simply to maximize the profits of the stockholders/shareholders. This suggests that the whole purpose of the business is this generation of profits. But as has been shown in the recent global financial crisis, businesses carry out functions that are important for society as a whole, for example, when it appeared that some high street banks might cease to operate, the UK and other governments stepped in and took over ownership in order to support the economy as a whole. This suggests that the capital provided by shareholders can be seen as a means of enabling the main business of the enterprise rather than the other way around.
- If commercial organizations are amoral in that their only concern is the generation of profits to the possible detriment of other stakeholders, then this is an argument for the public ownership of organizations such as energy suppliers upon whom society depends.
- If we expect individuals as individuals to be socially responsible and ethical, then when these individuals are also shareholders we might expect them to be socially responsible in that role as well.
- Paradoxically, acting in a socially responsible way may be a way of fostering goodwill in the community, winning new business and contributing eventually to shareholder value.

Chapter 12

12.1 Social loafing

Among other ideas, the effects of social loafing can be reduced by:

- making the work of each performer individually identifiable;
- involving and interesting group members in the outcome of group efforts;

- rewarding individuals for their contributions to the group effort (rather like sports teams who pick out a 'club player of the year').

12.2 Effect of ICT on the Delphi technique

Developments in ICT that assist cooperative working, especially the advent of electronic mail and groupware such as Lotus Notes, will cut quite considerably the communication delays involved in the Delphi technique.

12.3 Modes of communication

Once again, there is no one correct set of answers. Discussion points might be:

- (a) The developer might not be familiar with the context of the application domain – for example the terminology employed by the users. Ideally, clarification should be via same time/same place or same time/different place (e.g. telephone), as the business analyst and developer could go through a cycle of questions and answers quickly, and follow-up questions could be posed where answers were not completely clear.
- (b) To start with, same time/different place (e.g. a call to a help desk) might be most appropriate as the problem might be a simple misunderstanding. If a fault in the software was actually found then a different time/different place mode of communication would be needed to record the fault report so that it could be dealt with by the software maintenance team.
- (c) The requirement could be very complex and need considerable analysis, so initially different time/different place communication involving studying and writing documents might be best. Different options might need to be considered by the business and for this a same time/same place meeting might be expedient.
- (d) There might be good reasons why the second developer is late. For example, he might have been on sick leave. It might also be that he is unaware of the urgency of the task. Informal communication, ideally face to face, might be appropriate, at least initially.

12.4 Classification of types of power

More than one type of power can be involved in each case.

- i. Some *expert* power is involved here, but for those who are subject to the audit, the main type of power is *connection* power as the auditor will produce a report that will go to higher management. External auditors often have *coercive* power.
- ii. Here, power will mainly be *expert-* and *information-based*, but as the consultant will report to higher management, *connection* power also exists.
- iii. This sounds pretty *coercive*.
- iv. Brigitte has some *connection* power. The technical expertise that is involved in her job means she has some *expert* power. She has little or no *coercive* power as she is not the manager of the staff involved. She might be able to exert some *reward* power as she can satisfy some of the staff's need for ICT support.

- v. Amanda is unlikely to have direct *coercive* power although she might be able to institute disciplinary procedures. Through the system of annual reviews common to many organizations, she might have some *reward* power. *Connection* power, through her access to higher management, is also present. Her access to users means she has *information* power. If she brings specific expertise to the project (such as analysis skills) she might have some *expert* power. By acting as a role model that other project team members might want to emulate she may even be displaying *referent* power.

12.5 Appropriate management styles

- i. The clerk will know much more than anyone else about the practical details of the work. Heavy *task-oriented* supervision would therefore not be appropriate. As the clerk is working in a new environment and forging new relationships, a considerable amount of people-oriented supervision/support might be needed initially.
- ii. Both task-oriented and people-oriented management would be needed with the trainee.
- iii. The experienced maintenance programmer has probably had considerable autonomy in the past. The extensions to the systems could have a considerable, detailed, impact on this person's work. A very carefully judged increase in task-oriented management will be required for a short time.

Chapter 13

13.1 Selection of payroll package for college

- (a) Carry out an investigation to find out what the users' requirements really are. This might uncover that there are different sets of requirements for different groups of users.
- (b) Organize the requirements into groups relating to individual qualities and attributes. These might be, for example, functionality (the range of features that the software has), price, usability, capacity, efficiency, flexibility and reliability.
- (c) Some of these requirements will be of an absolute nature. For example, an application will have to hold records for up to a certain maximum number of employees. If it cannot, it will have to be immediately eliminated from further consideration.
- (d) In other cases the requirement is relative. Some of the relative requirements are more important than others. A low price is desirable but more expensive software cannot be ruled out straightaway. This can be reflected by giving each of the requirements a rating, a score out of 10, say, for importance.
- (e) A range of possible candidate packages needs to be identified. If there are lots of possibilities, an initial screening, for instance, by price, can be applied to reduce the contenders to a manageable shortlist.
- (f) Practical ways of measuring the desired qualities in the software have to be devised. In some cases, for example with price and capacity, sales literature or a technical specification can be consulted. In other cases, efficiency for instance, practical trials could be conducted, while in yet other cases a survey of existing users might provide the information required.

- (g) It is likely that some software is going to be deficient in some ways, but that this will be compensated by other qualities. A simple way of combining the findings on different qualities is to give a mark out of 10 for the relative presence/absence of the quality. Each of these scores can be multiplied by a score out of 10 for the importance of the quality (see (d)) and the results of all these multiplications can be summed to give an overall score for the software.

13.2 Possible quality specifications for word processing software

There are many that could be defined and just two examples are given below. One point that may emerge is that the software might be best broken down into a number of different functional areas, each of which can be evaluated separately, such as document preparation, presentation, mail merging and so on. For example:

- *quality*: ease of learning;
- *definition*: the time taken, by a novice, to learn how to operate the package to produce a standard document;
- *scale*: hours;
- *test*: interview novices to ascertain their previous experience of word processing. Supply them with a machine, the software, a training manual and a standard document to set up. Time how long it takes them to learn how to set the document up;
- *minimally acceptable*: > 2.5 to 4 hours;
- *target range*: 1 to 2.5 hours;
- *now*: 3 hours;

or

- *quality*: ease of use;
- *definition*: the time taken for an experienced user to produce a standard document;
- *scale*: minutes;
- *test*: time user who has experience of package to produce the standard document;
- *minimally acceptable*: 40 to 45 minutes;
- *target range*: 30 to 40 minutes;
- *current*: 45 minutes.

This topic of evaluation is an extensive one and the pointers above leave all sorts of unanswered questions in the air. Readers who wish to explore this area should read one of the more specialist books on the topic.

13.3 Availability and mean time between failures

Each day the system should be available from 8 a.m. to 6 p.m., that is 10 hours.

Over four weeks that should be $10 \times 5 \times 4$ hours = 200 hours.

It was unavailable for one day, i.e. 10 hours.

It was unavailable until 10.00 on two other days = 4 hours.

The hours available were therefore $200 - 10 - 4 = 186$ hours.

Availability would therefore be $186/200 \times 100 = 93\%$.

Assuming that three failures are counted, mean time between failures would be $186/3 = 62$ hours.

13.4 Entry requirements for an activity different from the exit requirements for another activity that immediately precedes it

It is possible for one activity to start before the immediately preceding activity has been completely finished. In this case, the entry requirement for the following activity has been satisfied, even though the exit requirement of the preceding activity has not. For example, software modules could be used for performance testing of the hardware platform even though there are some residual defects concerning screen layouts.

Another situation where the entry requirements could vary from the preceding exit requirements is where a particular resource needs to be available in order for the new activity to start.

13.5 Entry and exit requirements

- *Entry requirements* A program design must have been produced that has been reviewed and any rework required by the review must have been carried out and been inspected by the chair of the review group.
- *Exit requirements* A program must have been produced that has been compiled and is free of compilation errors; the code must have been reviewed and any rework required by the review must have been carried out and been inspected by the chair of the review group.

It should be noted that the review group could use checklists for each type of product reviewed and these could be regarded as further entry/exit requirements.

13.6 Application of BS EN ISO 9001 standards to system testing

There would be a need for a documented procedure that governs system testing.

The quality objective for system testing might be defined as ensuring that the software conforms to the requirements laid down in the user specification.

Processes to ensure this could include documented cross-references from test cases to sections of the specification.

The results of executing test cases would need to be recorded and the subsequent remedying of any discrepancies would also need to be recorded.

13.7 Precautionary steps when work is contracted out

The project manager could check who actually carried out the certification. They could also discover the scope of the BS EN ISO 9001 certification that was awarded. For

example, it could be that certification only applied to the processes that created some products and not others.

Perhaps the most important point is that the project manager will need to be reassured that the *specification* to which the contractors will be working is an adequate reflection of the requirements of the client organization.

13.8 Information flows for staff allocation

When the *architecture design* process which creates work packages is taking place, there could be a further output, namely the effort estimation for each software component. These could be passed to a management process which allocates staff to the *develop software* process. The develop software process would need to pass back information about the actual effort being used as this would allow adjustments to resource allocations to be made as necessary.

13.9 Comparison of peer review and pair programming

Here are some ways in which they might be contrasted:

Pair programming	Peer review
Works on the principle that two heads are better than one	Peer review groups could be made up of more people.
Driver and navigator are jointly responsible for producing the software product	Developer solely responsible for the initial creation of the product which is then reviewed.
Discussion of the rationale for the design as it is being produced	Reviewers see the final product only, not the reasoning behind it unless design documentation provides the rationale.
Real-time interaction between participants	Batch orientation with focus on the documents.
Development effort doubled	The time of several members of staff needed to study documents and then attend the review meeting.

There could be further discussion of the respective advantages and disadvantages of each approach.

13.10 The important differences between a quality circle and a review group

The quality circle would be looking at the process in general while the review group would look at a particular instance of a product. The use of review groups alone could be inefficient because they could be removing the same type of defect again and again rather than addressing, as the quality circle does, the task of stopping the defects at their source.