PROJECT-MANAGEMENT FOR IT-RELATED PROJECTS

Third edition

Bob Hughes (editor)

bcs
The Chartered Institute for IT

Table of Contents

PROJECT MANAGEMENT FOR IT-RELATED PROJECTS

BCS, THE CHARTERED INSTITUTE FOR IT

BCS, The Chartered Institute for IT, is committed to making IT good for society. We use the power of our network to bring about positive, tangible change. We champion the global IT profession and the interests of individuals, engaged in that profession, for the benefit of all.

Exchanging IT expertise and knowledge

The Institute fosters links between experts from industry, academia and business to promote new thinking, education and knowledge sharing.

Supporting practitioners

Through continuing professional development and a series of respected IT qualifications, the Institute seeks to promote professional practice tuned to the demands of business. It provides practical support and information services to its ...

PROJECT MANAGEMENT FOR IT-RELATED PROJECTS

Third edition

Bob Hughes (editor)
Roger Ireland, Brian West, Norman Smith,
David I. Shepherd

bcs
The Chartered Institute for IT

CONTENTS

LIST OF FIGURES AND TABLES

USEFUL WEBSITES

IT PROJECT MANAGEMENT QUALIFICATIONS AND SYLLABUSES

https://www.bcs.org/get-qualified/certifications-for-professionals/project-programme-management-and-consultancy-certifications/bcs-foundation-certificate-in-is-project-management/
BCS Professional certification: Foundation Certificate in IS Project Management

https://www.bcs.org/get-qualified/certifications-for-professionals/project-programme-management-and-consultancy-certifications/bcs-essentials-certificate-in-programme-and-project-support-office/
BCS Professional certification: Foundation Certificate: Programme and Project Support Office Essentials

https://bcs.org/upload/pdf/dippmsyll.pdf
BCS Higher Education Qualifications: Diploma in IT: IT Project Management Syllabus. An 'academic' examination at university 2nd-year level, popular with overseas candidates

AGILE PROJECT MANAGEMENT APPROACH

https://www.agilebusiness.org/
Agile Business Consortium. This group was formerly the DSDM Consortium. It developed the DSDM Agile project management framework (which was for a short time branded as Atern)

https://www.scrumalliance.org/
Scrum Alliance has a set of resources supporting the Scrum Agile framework

Some professional bodies – APM and PMI have their own qualifications:

https://www.apm.org.uk/
Association for Project Management: the UK professional body for generic project management (rather than just IT)

https://www.pmi.org/
Project Management Institute: US-based professional body. A UK chapter of PMI exists **https://pmi.org.uk/** and there is also a PMI-run magazine and community website for project managers at **https://projectmanagement.com**

https://www.ipma.world/
International Project Management Association: A global umbrella association to which most national project management professional bodies are affiliated

PLANNING TOOLS

https://microsoft.com/project/en-us/project-management.aspx
Microsoft Project: probably the most widely used project planning tool

https://www.oracle.com/uk/industries/construction-engineering/primavera-products/
Oracle Primavera: another, perhaps more industrial, project planning tool (and much else)

https://trello.com
Trello: an excellent example of a modern software tool that supports collaborative working

https://asana.com/
Asana: another tool to support project planning and execution

https://www.smartsheet.com/
Smartsheet is an easy to use tool for small one-off projects where there is a need to do things quickly and simply. It can interface with more 'traditional' spreadsheets and Gantt charts

QUALITY

https://www.tickitplus.org/en/
TickIT: UK initiative to apply ISO9001 to IT development

https://cmmiinstitute.com/
Details of the SEI Capability Maturity Model (CMMI)

ESTIMATION AND MEASUREMENT

https://cosmic-sizing.org/
All you need to know about COSMIC function points

PROJECT ORGANISATION

https://axelos.com/best-practice-solutions/prince2
PRINCE2, the UK government-sponsored standard for project management procedures

GENERAL KEEPING UP-TO-DATE

https://www.bcs.org/membership/member-communities/project-management-specialist-group-proms-g/
PROMS-G: the BCS Project Management Specialist Group

www.pmtoday.co.uk/
Project Management Today, trade magazine

https://blog.practicingitpm.com/
The Practicing IT Project Manager, news, articles and a weekly round-up

https://girlsguidetopm.com/welcome-to-the-resource-library
20 project templates

ACKNOWLEDGEMENTS

This book has arrived at its present state via a process of development through three editions that have involved a large number of people. For a start, the following people contributed the original material for the text in the first edition:

Norman Smith: Chapters 1 and 4

Bob Hughes: Chapters 2 and 6

Roger Ireland: Chapters 3 and part of 8

Brian West: Chapters 5 and part of 8

David I. Shepherd: Chapter 7

Although the text has since gone through many changes in terms of updating and general tinkering, the original material on which these have been built has been an enduring foundation. Sue McNaughton and Elaine Boyes at BCS drove the publication project for the first edition. The original development of the Foundation Certificate ...

PREFACE

This book aims to give a practical introduction to fundamental IT project management principles and techniques. The first edition was written with the specific purpose of providing learning material to candidates for the BCS Foundation Certificate in IS Project Management. The second edition still supported this qualification, but updated some of the material and broadened its practical application.

Taking this qualification is not itself a daunting challenge. It consists of an hour-long 40-question multiple choice examination. However, the intention was never just to help cram for an examination. While there might be an immediate concern to pass a test, for most people the more important motivation was to gain guidance on planning and managing an IT project. The text was designed to help those from an IT practitioner background who were beginning to take on project management responsibilities. It might also give IT users some insights into IT project management issues. The book therefore goes beyond simply helping people to tick the right boxes in a

test and aspires to support novice IT project leaders in their place of work.

With any new topic, a good starting point is a text which provides a simple explanation of the basics. Having grasped the basics, you can then go on and explore more advanced concepts. A measure of the success of the previous editions was that they started to be used for purposes for which they were not primarily designed. One example of this was the BCS Higher Education Qualification Diploma in IT Project Management (an 'academic' BCS qualification comparable to a UK university award and taken mainly by overseas candidates).

However, the focus still remains on the foundations – there is only so much you can cram into a three-day course – but care has been taken here to provide links to other, more detailed project management material. Wherever possible, alternatives to the terminology we have used are provided for techniques and concepts to allow easier cross-reference to other bodies of knowledge. For example, 'steering committee', 'project board' and 'project management board' all refer to largely the same concept in project management.

We have put in links to further material using a  symbol for those who want to explore a topic more deeply. Some material in the basic text goes beyond what is needed for the BCS Foundation syllabus and these have been marked with a  symbol to indicate an 'advanced topic'.

It may be heretical to say this in a project management book, but successful projects depend on more than good project management and some of the links provided are to material on complementary disciplines that can assist positive project outcomes. (The BCS International Diploma in Business Analysis, to which the Foundation Certificate in IS Project Management can contribute, supports this view.)

The BCS Foundation Diploma syllabus has been very stable in recent years and there have been no massive changes in content in this (third) edition. The immediate motivation was to update outdated references, particularly to standards, so that, for example, we refer to the ISO 25000 series of standards on software quality requirements rather than ISO 9126. However, we have taken more care to acknowledge that IT projects increasingly involve implementing existing functionality provided by vendors and writing business software from scratch is less prevalent. Where software is developed, Agile approaches are now common. The main principles of project agility – such as the focus on iterations and increments in project delivery – had been well-established before the term 'agile' was adopted in the context of software development, so it has been easy to signpost those elements of our approach that dealt with them.

In this edition we have attempted to reduce reliance on PRINCE2 concepts and terminology. PRINCE2 is a UK government-sponsored set of procedures for managing major projects that is administered by Axelos, a venture jointly owned

by Capita and the UK government. In our view, it effectively describes an information system for a project that allows it to be run in a controlled and efficient manner. Although PRINCE2 is really an administrative standard that will tell you what decisions need to be taken and when, it does not claim to be a set of project management principles and techniques.

There is an understandable tendency for IT project management to lose the 'IT' and be treated as just project management. We want to resist this. It is certainly true that IT projects are often parts of broader business change programmes. It is frequently argued that technical expertise is unnecessary for professional project managers who can move successfully between different industry and business sectors, and it is certainly true that great software engineers are frequently unsuited to or uninterested in project management. But project management is not just a matter of persuasive communication. A glib personality can sell courses of action which are just plain wrong. Leadership includes providing guidance on the best ways of meeting the challenges of applying technologies to meet organisational objectives based on sound evidence. Successful IT leaders cannot know everything about IT, but they need a solid understanding of the professional practices involved in IT. They need to know how to develop and exploit the skills and expertise of a range of people from different disciplines.

We hope this book will help you to plan and manage your IT and software projects. By a happy coincidence, the last day of

my work on the new edition has coincided with the 50th anniversary of the Apollo 11 landing on the moon. Individuals can do good, but to achieve true greatness we need to work together.

1 PROJECTS AND PROJECT WORK

LEARNING OUTCOMES

When you have completed this chapter you should be able to demonstrate an understanding of the following:

- the definition of a project;
- the purpose of project planning and control;
- the typical activities in a system development life cycle;
- system and project life cycles;
- variations on the conventional project life cycle;
- transition strategies;
- the purpose and content of the business case;
- types of planning documents;
- post-implementation reviews.

1.1 PROJECTS

A **project** may be defined as a **group of related activities carried out to achieve a specific objective**. Examples of projects include building a bridge, making a film and re-organising a company. The outcome of a project is usually ...

2 PROJECT PLANNING

LEARNING OUTCOMES

When you have completed this chapter you should be able to demonstrate an understanding of the following:

- project deliverables and intermediate products;
- work and product breakdowns;
- product definitions (including the identification of ‘derived from’ and ‘component of’ relationships between products);
- relationship between products and activities in a project;
- checkpoints and milestones;
- elapsed time and effort required for activities;
- activity networks (using the ‘activity on node’ notation);
- calculation of earliest and latest start and end dates of activities and the resulting float;
- identification and significance of critical paths;

- resource allocation, smoothing and levelling, including the use of resource histograms;
- work schedules and Gantt charts.

2.1 INTRODUCTION

Having described the context in which IT projects exist, we are now going to explore the steps in producing a plan for a project. You will recall from Chapter 1 that before the detailed planning of a project starts, the **business case** for the project is set out. This, among other things, lays out a technical strategy or 'solution' identifying the practical steps needed to achieve the project's objectives. It also gives an idea of the costs of developing the solution. You need to show that the value of the proposed IT application's benefits will outweigh the costs of developing and managing it. The overall **objectives** of the project, which define the successful outcomes of the project as agreed by the main participants in the project, also need to be identified.

2.2 APPROACHES TO PLANNING

There are two approaches to identifying the components of a project: **product-based** and **work-** or **activity-based**.

2.2.1 PRODUCT-BASED PLANNING

With the product-based approach, detailed planning usually begins with identifying the **project deliverables**; that is, the products that will be created by the project and delivered to the

client. A product must be in some way tangible; perhaps a software component, a document, a piece of equipment, or even a person (for example, a trained user) or a new version of some existing product, as with a modified version of a software component.

In the case of the Water Holiday Company integration project, the deliverables include:

- common software functionality, which enables members of the public to book online boating holidays provided by the previously separate Canal Dreams and Minotours companies via the merged website;
- enhanced network systems that can cope with the expected additional traffic to the Water Holiday Company website;
- a new merged network support and customer service centre to support 24-hour/seven days a week activity rather than the two separate sites for the two former trading entities.

Once the deliverables have been defined, **intermediate products** can be identified. These are products created during the course of the project, but which may not actually be delivered to the client at the end of the project. In the case of the Water Holiday Company integration project, intermediate products include (among other things):

- consolidated business process models;
- interface design documents which take account of the new corporate identity, such as a corporate website style guide,

site maps, ‘wireframe’ designs;

- an enhanced IT infrastructure architecture plan;
- software specifications;
- acceptance test plan;
- progress reports.

Some products, such as progress reports, will relate to the management or quality control of the project.

The deliverable and intermediate products can be written simply as a list of products, but sometimes they are shown in a **product breakdown structure** diagram (PBS); see Figure 2.1 for an example.

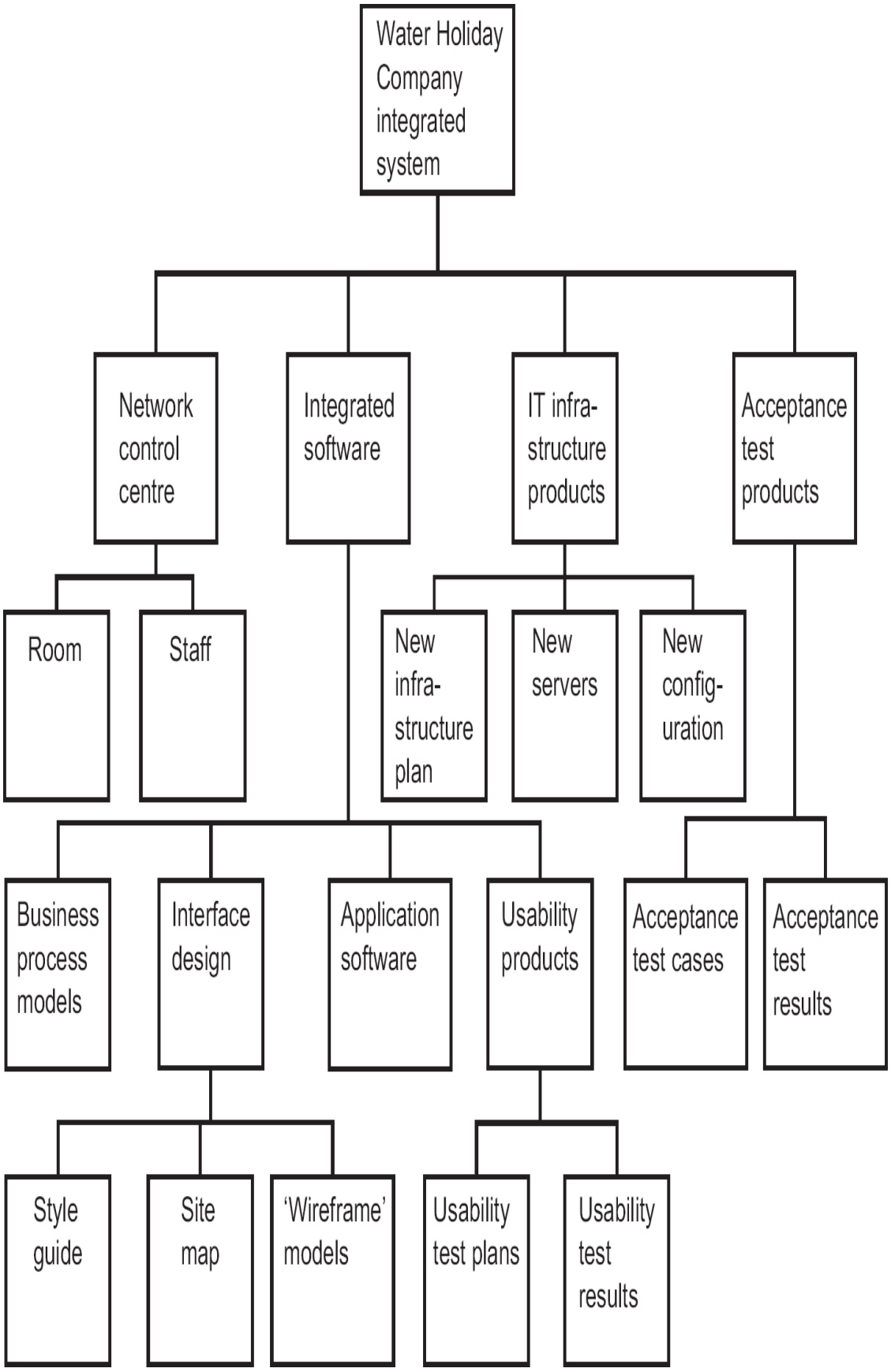
Some of the stakeholders in the project may find there are some products with which they are unfamiliar. Some users, for example, may be unsure of what is meant by an ‘acceptance test plan’. To remedy this, planning should include drawing up **product definitions**. For each product, the following should be documented:

- The **identity** of the product – for example ‘acceptance test plan’.
- A **description** of the product – for example ‘a plan of the test cases and the results that users expect the application to produce’.
- The product or products that have to exist before this one can be created; that is, those it is **derived from** – for

example, the acceptance test plan is stated to be derived from the requirements specification, which describes the main transactions of the application.

- The **components** that make up the product – in the case of an acceptance test plan, the main sections in the document.
- The **format** of the product – for example, that it is a word-processed document or a spreadsheet or a piece of software.
- The **quality criteria** that explain how the product will be judged as satisfactory – for example, the acceptance test plan being reviewed against the requirements specification. These are also known as **acceptance criteria**.

Figure 2.1 A product breakdown structure diagram


Water Holiday Company integrated system
Network control centre
Integrated software
IT infra-structure products
Acceptance test products
Room
Staff
New infra-structure plan
New servers
New config-uration
Business process models
Interface design
Application software
Usability products
Acceptance test cases
Acceptance test results
Style guide
Site map
'Wireframe' models
Usability test plans
Usability test results

2.2.2 WORK AND PRODUCT BREAKDOWN STRUCTURES

An alternative method of planning is the work- or activity-based approach, which identifies the required work activities or tasks in a **work breakdown structure** (**WBS**). In this case, the intermediate products listed on the previous page relating to setting up the Water Holiday Company integration project would be replaced by activities such as:

- analysing, merging and redesigning business processes;
- designing web interfaces;
- redesigning unified network architecture;
- specifying software;
- acceptance testing;
- reporting progress.

As nearly all activities will generate a product – or else why do them? – and all products will need to have some activities that give birth to them, there may not really be much difference between the two approaches in practice.

ACTIVITY 2.1

Which products are created by each of the following?

a. testing

b. training

c. network installation

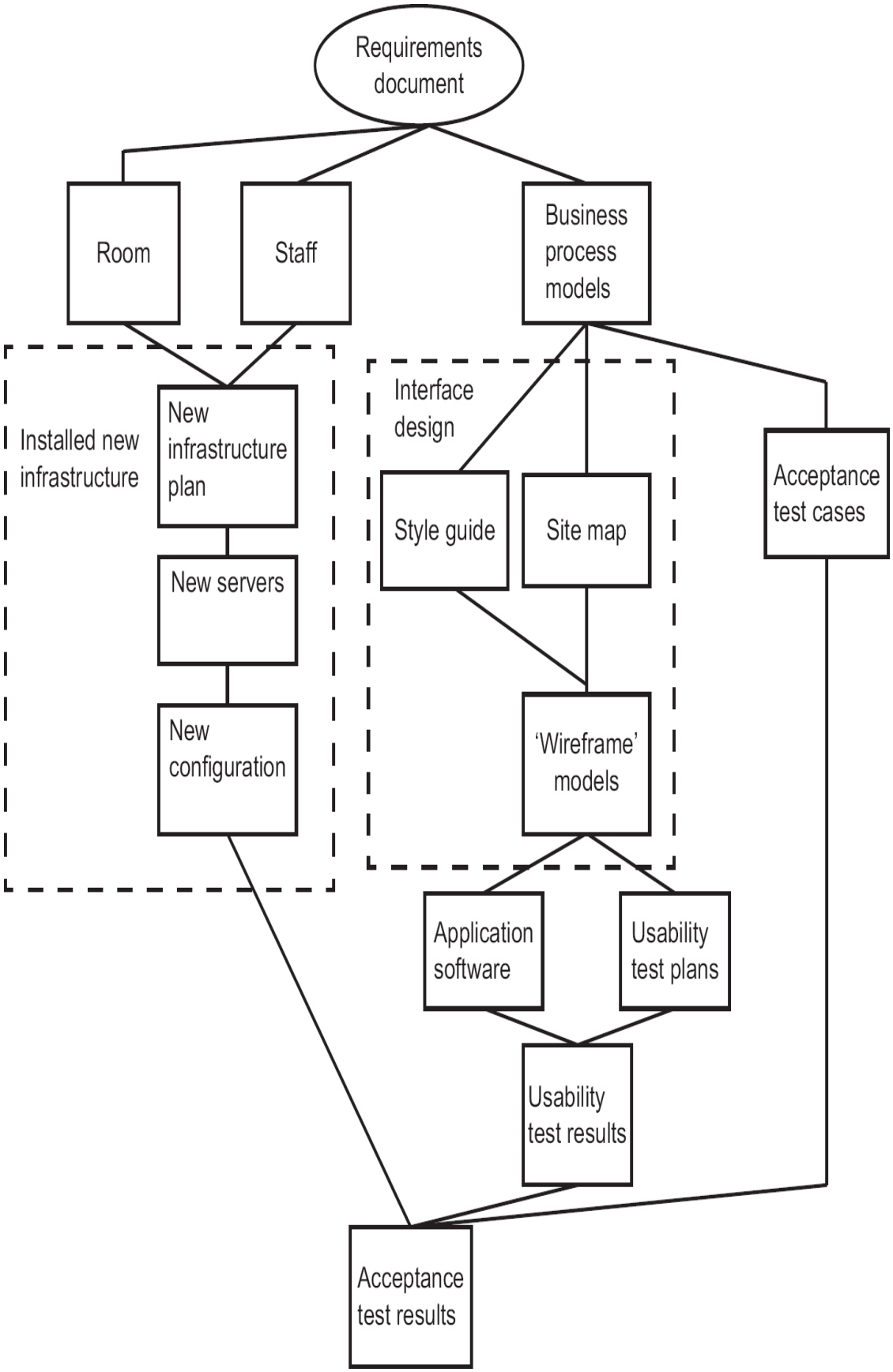
d. a project progress meeting

2.3 PRODUCT FLOW DIAGRAM

If you have adopted a product-driven approach, it is possible to draw up a **product flow diagram** (**PFD**) showing the order in which the products have to be created. This should be relatively easy to draft if you have already produced product definitions that specify from which other products each product is derived. Figure 2.2 gives an example fragment of a product flow diagram.

Note the oval with 'requirements document' in it. This refers to a product that already exists and that will be used to generate one or more of the products in the PFD. There is no one correct PFD – its structure depends on the technical solution adopted to achieve the project objectives. For instance, in Figure 2.2 the assumption is that the business process model will provide most of the information needed to design the sequence of screens for both the website and the mobile phone app for the merged Water Holiday Company. A unified corporate image is needed for the company business entity and it was felt that the business models would give the designers of the style guide a good idea of the business environment. The structure needed for the interfaces and the style guide would both be taken account of in the final interface design.

Figure 2.2 A product flow diagram


Requirements document
Room
Staff
Business process models
Installed new infrastructure
New infrastructure plan
New servers
New configuration
Interface design
Style guide
Site map
'Wireframe' models
Acceptance test cases
Application software
Usability test plans
Usability test results
Acceptance test results

The flow of a PFD is normally from top to bottom and then left to right. Looping back is not allowed – not because this cannot happen in real life, but because it is almost always possible **technically** to go back and re-work a product previously thought to be completed. In this case all the products depending on the reworked one might also need some re-working.

In two places in Figure 2.2 we have put boxes of broken lines around a sequence of products. This is not part of the official PFD notation. We wanted to show that a group of components (in one case, site maps, style guide and 'wireframe' models) will be treated as one large product (the interface design).

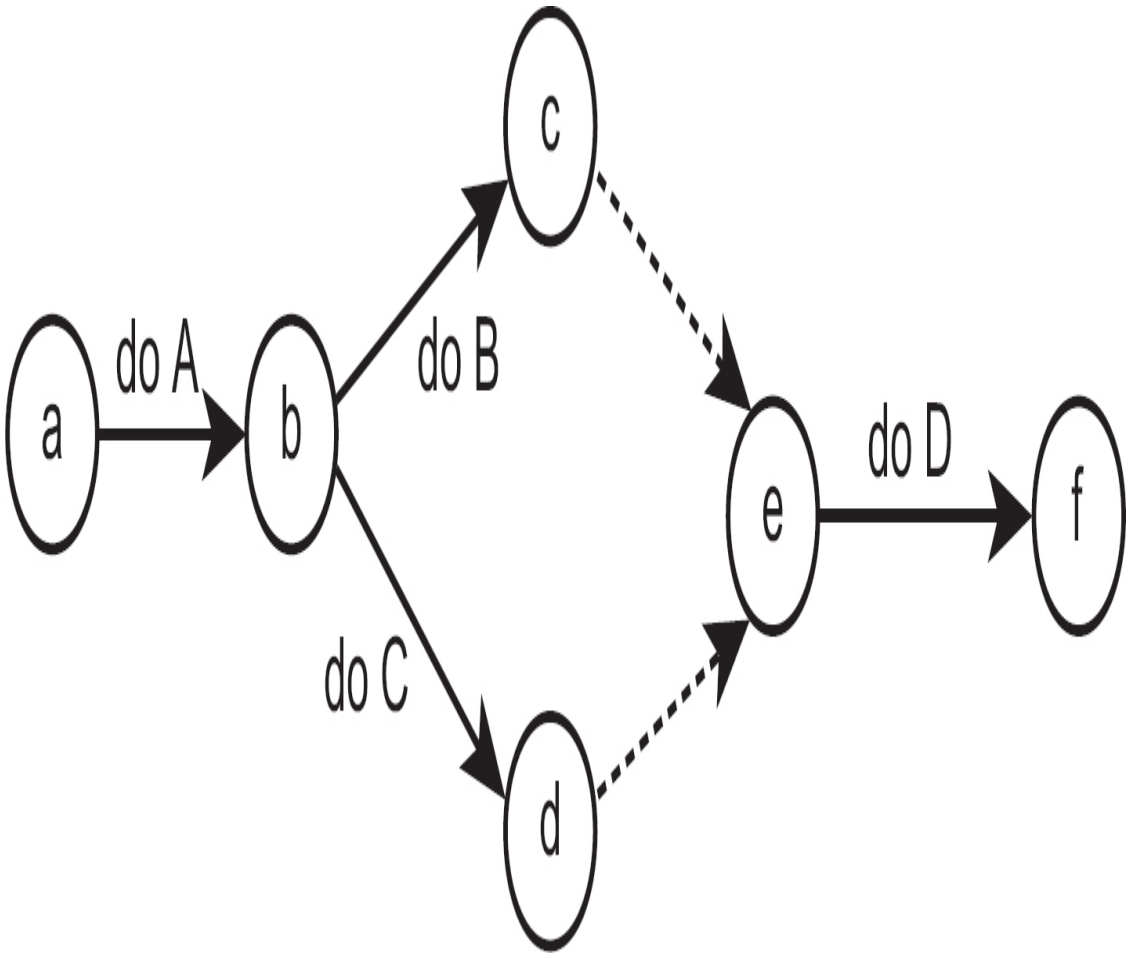
2.4 ACTIVITY PLANNING

Whether a product flow diagram has been drawn up or whether the planner has simply drawn up a list of activities, the next step is to draw up an **activity network**. This shows the activities needed for the project and the order in which they are to be carried out.

2.4.1 ACTIVITY NETWORK DIAGRAM

There are two sets of conventions for drawing up activity networks: **activity on node** and **activity on arrow**. Figure 2.3 shows an example of an activity on arrow diagram.

Figure 2.3 Activity on arrow network



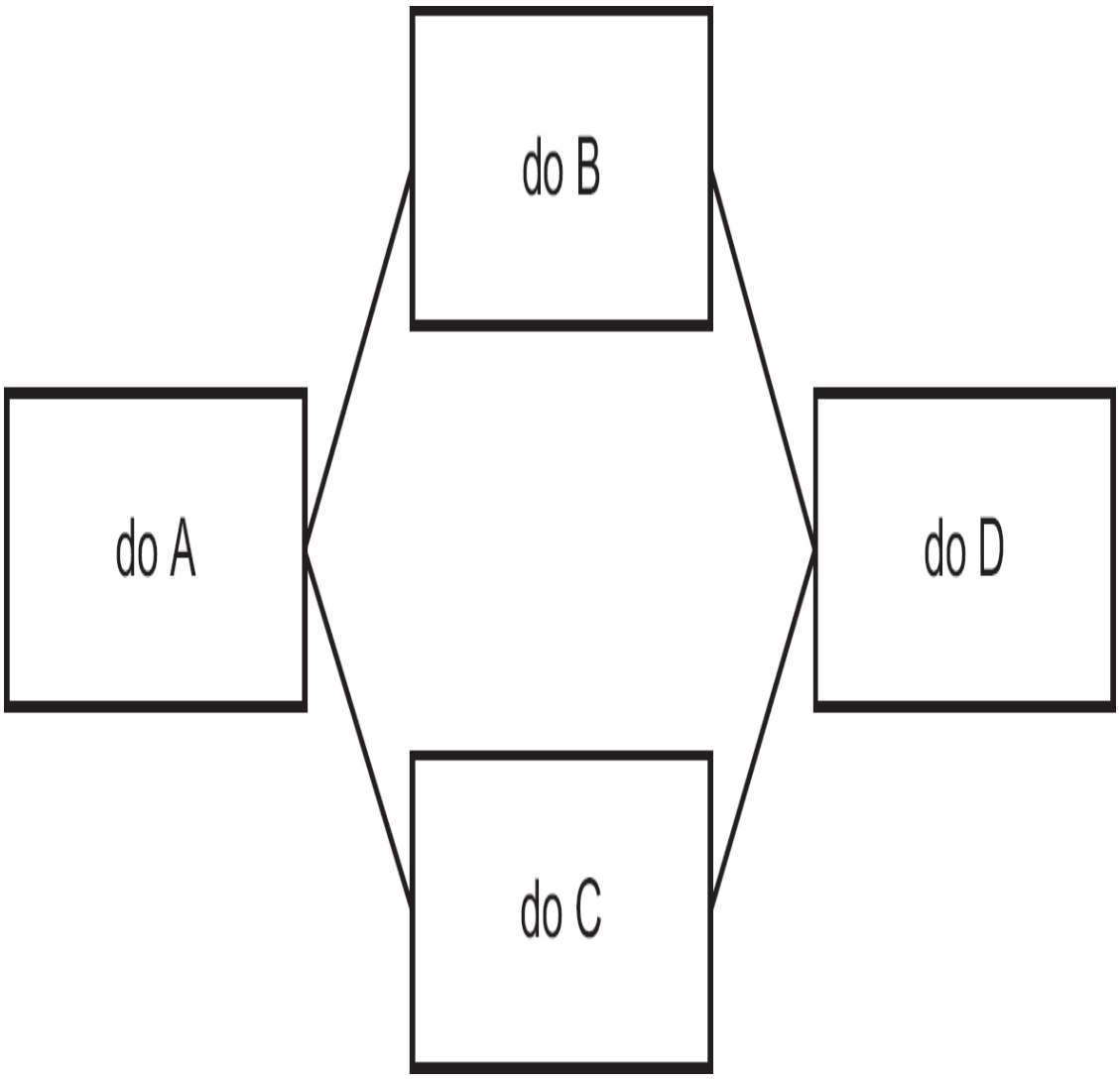
As the name implies, the arrows in an activity on arrow diagram represent activities, while the circles that link the arrows (that is, the **nodes**) represent the ends of some activities and the starts of others. The arrows with broken lines indicate 'dummy activities', which simply show a dependency between two of the event nodes – for example c, the end of 'do B', and e, the start of 'do D'.

We will use a different set of conventions, **activity on node**, which is used by most modern project planning tools, including Microsoft Project.

Figure 2.4 shows the same activities as Figure 2.3, but using activity on node notation. Here the boxes (which are the 'nodes' in this case) represent activities, while the lines between the boxes show where the start of one activity depends on the completion of some other activity. Note that at this stage the constraints may be technical or external. A technical constraint normally means that a product has to be created by one activity so that another can use it. An example of an external constraint is a system that can only be tested out of office hours, so it has been agreed contractually that testing will take place at weekends only.

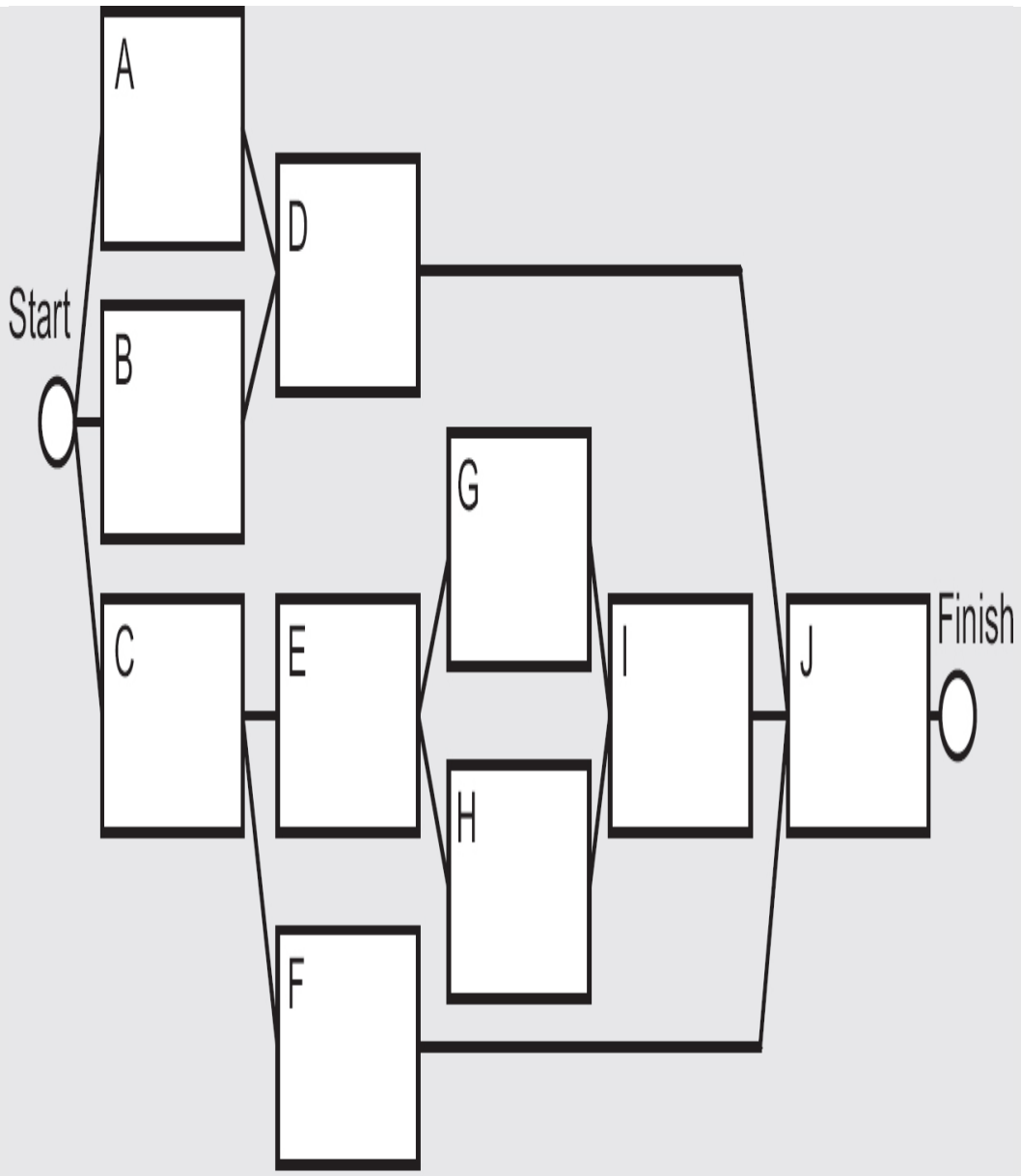
What are not taken into account at this point are **resource constraints** – for example, that a person will not be able to start on one task because he or she will not have finished another. These considerations are deferred because we do not know all the competing demands on a particular resource until all the activities in a project have been set out.

Figure 2.4 Activity on node network



ACTIVITY 2.2

In this activity network, match the activities with the boxes so that the activity network is compatible with the product flow diagram in Figure 2.2.



... allocate room
... analyse business processes
... carry out acceptance tests
... carry out usability tests
... devise usability tests
... design interface
... draft acceptance test cases
... install infrastructure
... recruit staff
... write software

In Activity 2.2 we added a ‘start’ and ‘finish’. These are important points of time (or ‘**events**’) in the life of the project, but they will not actually take up any time. If, for example, the finish of the project was marked by a celebration that took up several hours, then the ‘event’ would become an activity in its own right. We call these important events **milestones**. Milestones can also be located in the middle of a project, for example at the end of one important phase and the start of the next. Always remember, though, that milestones do not take up time. Sometimes an important point in a project may be marked by a meeting to check that everything planned has been completed successfully before the next part of the project starts. This checkpoint would be an activity in its own right.

2.4.2 ESTIMATING ELAPSED TIME

Having identified the activities and the order in which they have to be worked on, we now need to estimate how long we think each activity is likely to take. Note that we are concerned here about **elapsed times**. This is the time from the start of an activity to the finish. This is not the same as the **effort** spent on an activity. Effort could be more than elapsed time – for example, where we have three people working together on a job for two days, the elapsed time would be two days, but the effort would be six staff days. Effort could also be less than elapsed time – for example, where someone works only afternoons. Estimates of effort become important when

projects are being costed. Estimating is covered in more detail in Chapter 6.

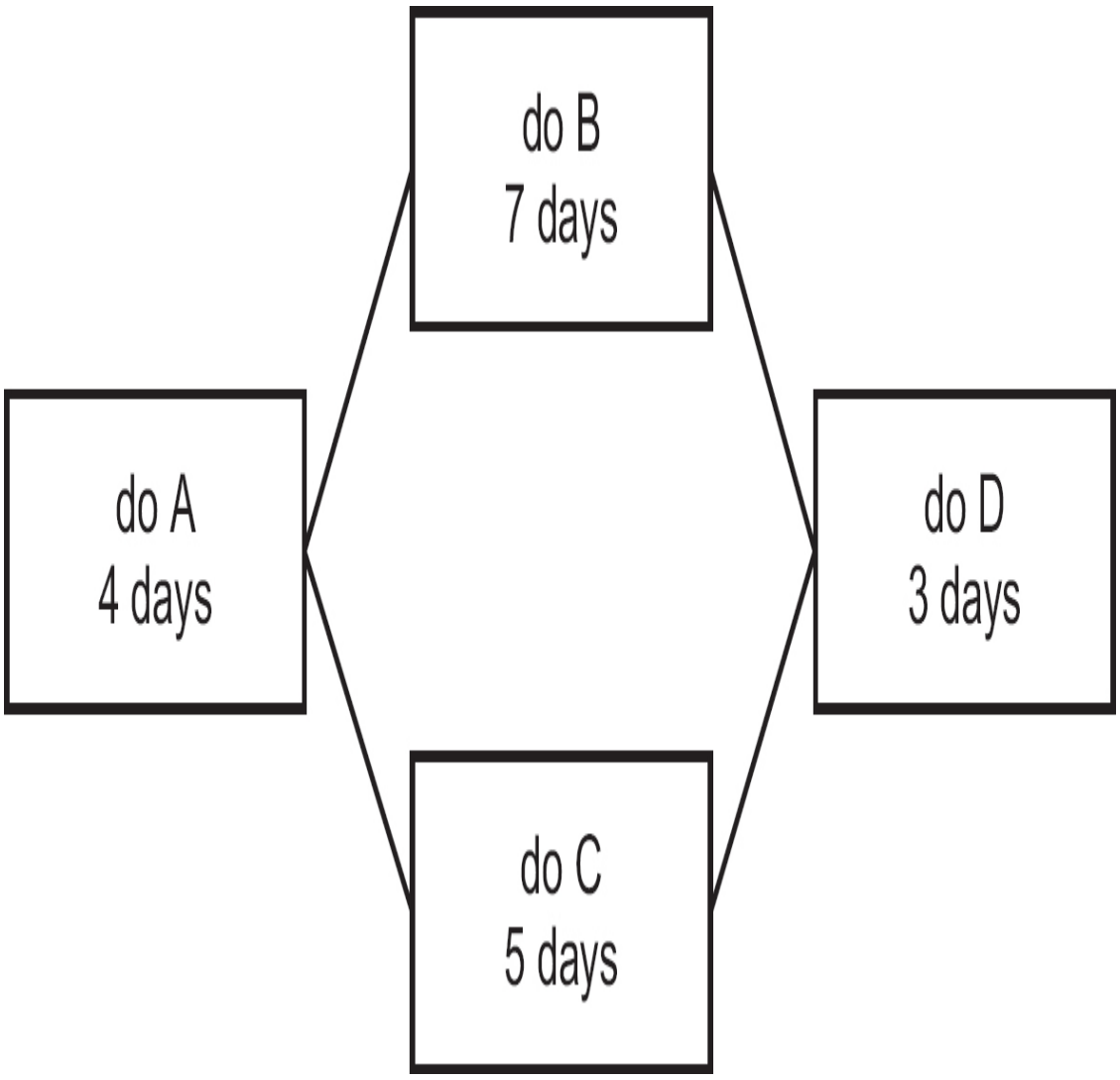
Let's assume that we can allocate estimated durations to the activities in the activity network of Figure 2.4 (see Figure 2.5).

We want to know the earliest day that each activity can start. Rather than worry about taking account of weekends and public holidays at this point, we simply allocate each working day a sequence number, starting with day 0. (Technically, day 0 means 'the end of day 0', which means the start of day 1, as explained below.)

The **earliest start date** for 'do A' is day 0 by definition, because it is the first activity in the network. The earliest time at which the activity can finish is day 0 plus the duration of the activity: that is, the end of day 4.

earliest finish date = (earliest start date + activity duration)

Figure 2.5 A network activity fragment with activity durations



The earliest start dates for the two activities ‘do B’ and ‘do C’ are governed by the earliest finish date of the preceding activity, ‘do A’. In fact, we can say that in this case the earliest start dates for ‘do B’ and ‘do C’ are the same as the earliest finish date for ‘do A’, that is, day 4.

You may wonder why it is not the **following** day. Well, the convention is that when we say ‘do A’ finishes on day 4, we mean at the **end** of day 4. When we say ‘do B’ and ‘do C’ start on day 4 we really mean at the **end** of day 4, which of course

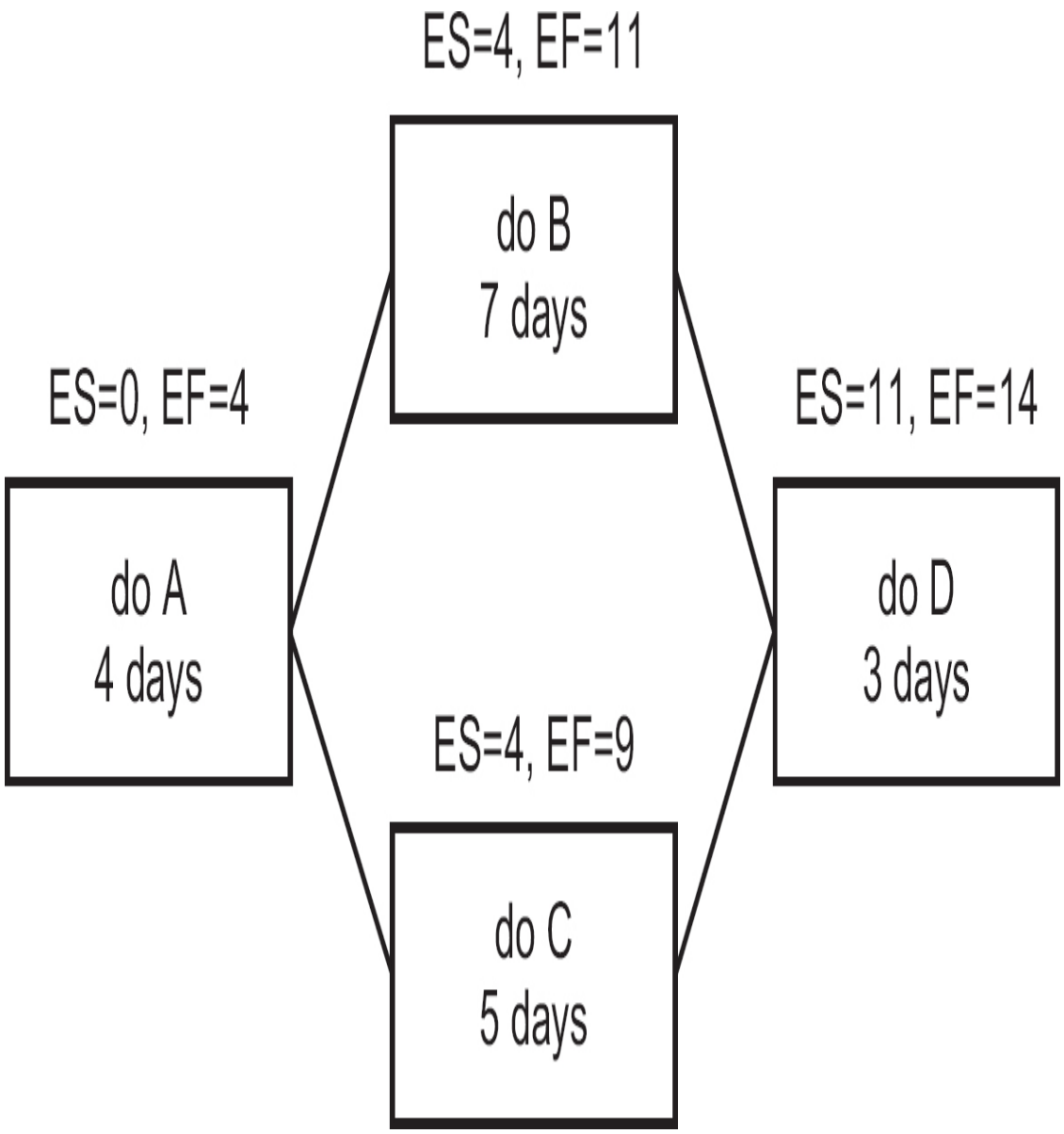
really means the start of day 5. It is best just to accept this as the convention. It saves problems arising where activities do not take whole numbers of days, for example 5.5 days.

We can now work out the earliest finish days of ‘do B’ and ‘do C’ as day 11 and day 9, respectively. What about the earliest start date for ‘do D’? We have two preceding earliest finish dates, so we take the one which is later: that is, day 11.

> earliest start date = the latest of the earliest finish dates of the preceding activities upon which the current activity is dependent

We end up with the day numbers shown in Figure 2.6, where ES means the earliest start date and EF means the earliest finish date.

Figure 2.6 Earliest start (ES) and finish (EF) days



It is possible for some activities to start or finish late without the project as a whole being delayed. To see where this is the case, the **latest finish** and **latest start** dates for each activity are calculated. We will assume that we want the project as a whole to take the shortest time possible: that is, to finish on day 14. Day 14 becomes the latest finish date for the activity 'do D'. The latest start day for this activity is calculated by

subtracting the duration from the latest finish: that is, 14 – 3 = Day 11.

> latest start date = latest finish date of current activity – duration

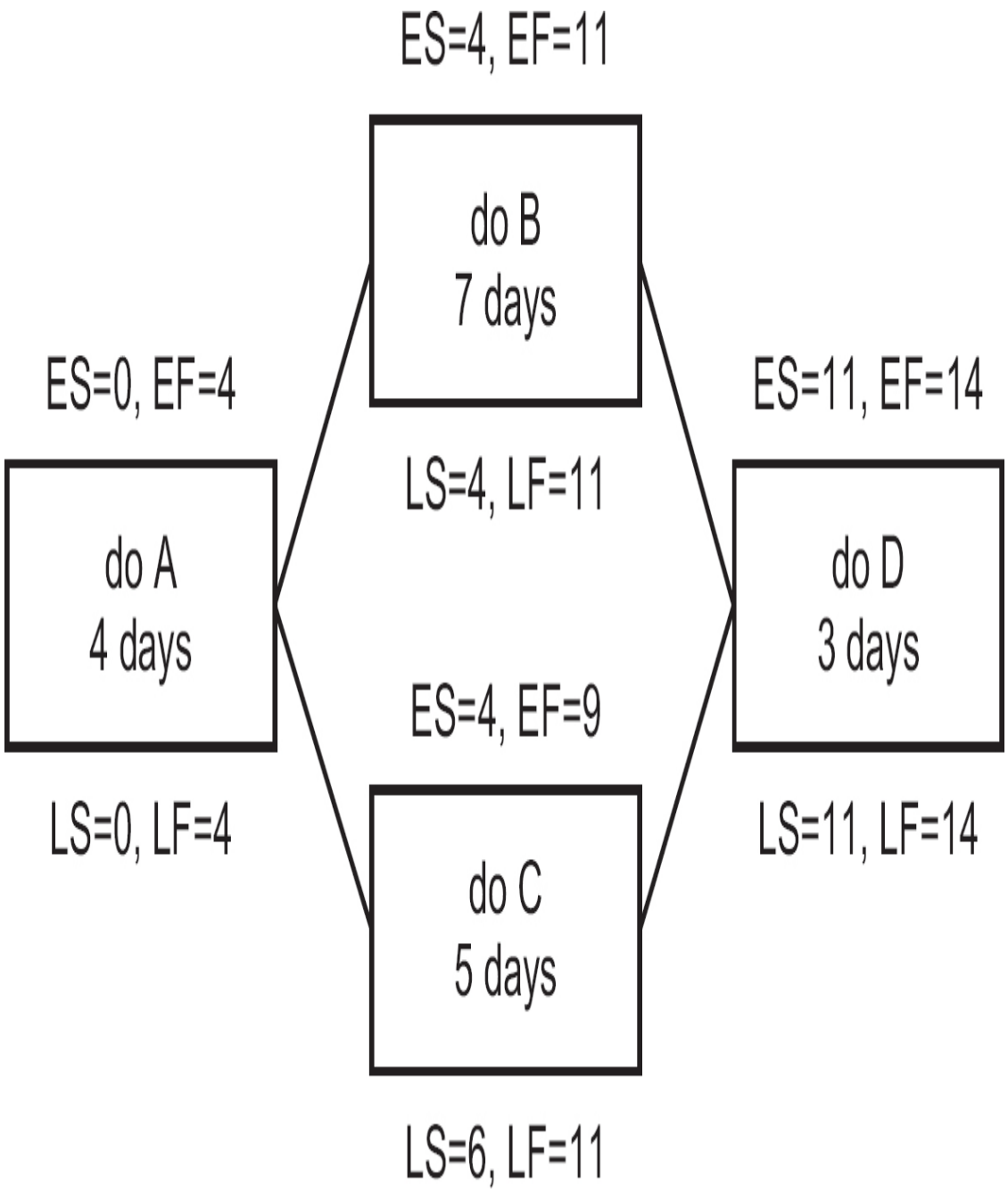
We now work backwards. The latest start day for the activity ‘do D’ becomes the latest finish day for ‘do B’ and ‘do C’. By subtracting the durations for these activities from their latest finish days we get their latest start days: that is, 11 – 7 = Day 4 for ‘do B’ and 11 – 5 = Day 6 for ‘do C’.

In the case of ‘do A’, we have to decide whether to base the latest finish on the latest start date of ‘do B’ or ‘do C’. The earlier of the two is taken as the latest finish time for ‘do A’: that is, day 4, which comes from ‘do B’.

> latest finish date = the earliest of the latest start dates of the activities that are dependent on the current activity

We now have the situation shown in Figure 2.7, where LS means latest start and LF means latest finish.

Figure 2.7 Latest start (LS) and finish (LF) dates



It can be seen from Figure 2.7 that for all the activities except 'do C', the earliest and latest **start** days are the same, as are the earliest and latest **finish** days. This means that if these activities are late, the project as a whole will be delayed. In the case of 'do C', if you look at the day numbers you can see there is a difference of two days between the earliest and latest day

numbers. This means that ‘do C’ could be one or two days late and the duration of the project as a whole would not be affected.

This leeway is called the **float** and can be defined as:

float = latest finish date – earliest start date – duration

A quick way of calculating this is by subtracting the earliest start from the latest start (or the earliest finish from the latest finish).

In Figure 2.7, ‘do A’, ‘do B’ and ‘do D’ all have zero float. They form a small chain of three activities from the beginning to the end of the activity network. This chain is the **critical path** (**CP**). If any activity on this path is delayed, then the whole project will be delayed.

The details for each activity can be displayed more clearly if the boxes on the activity diagram are divided up as shown in Figure 2.8.

Thus, for the activity ‘do C’, the activity box in the activity network could be drawn up as in Figure 2.9.

The **activity span** is the total period during which the activity could take place, and is defined as:

activity span = latest finish day – earliest start day

In this case it is 11 – 4: that is, 7 days. Where an activity has float, there is a ‘window of opportunity’, reflected by the

activity span (see Figure 2.10), within which the activity can start and be completed.

Figure 2.8 Layout of an activity box

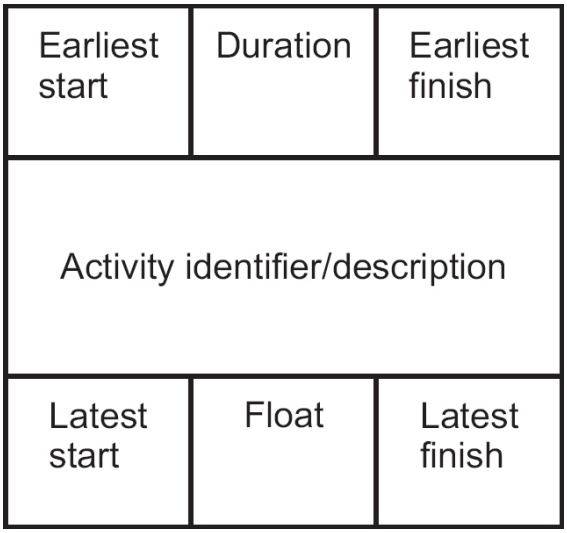


Figure 2.9 Activity box for ‘do C’

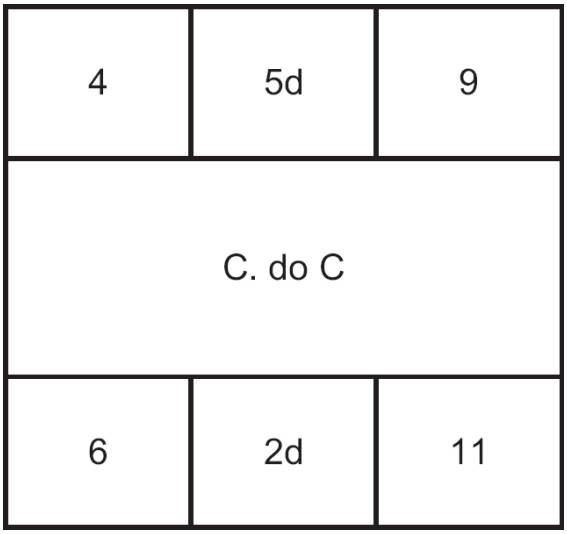
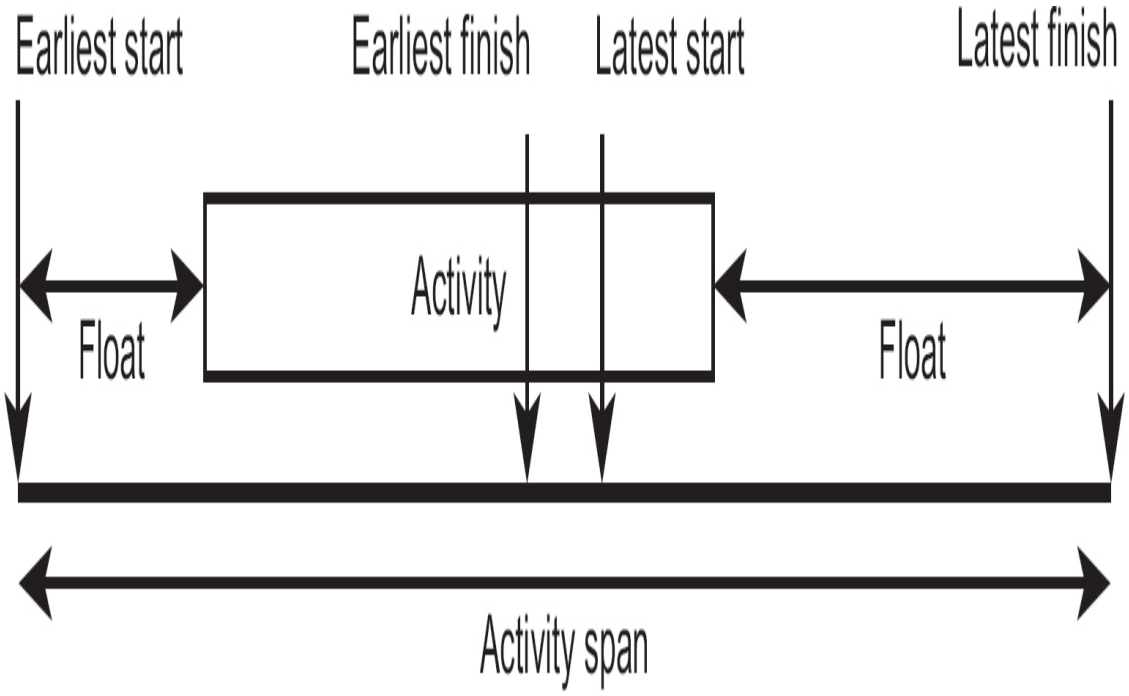


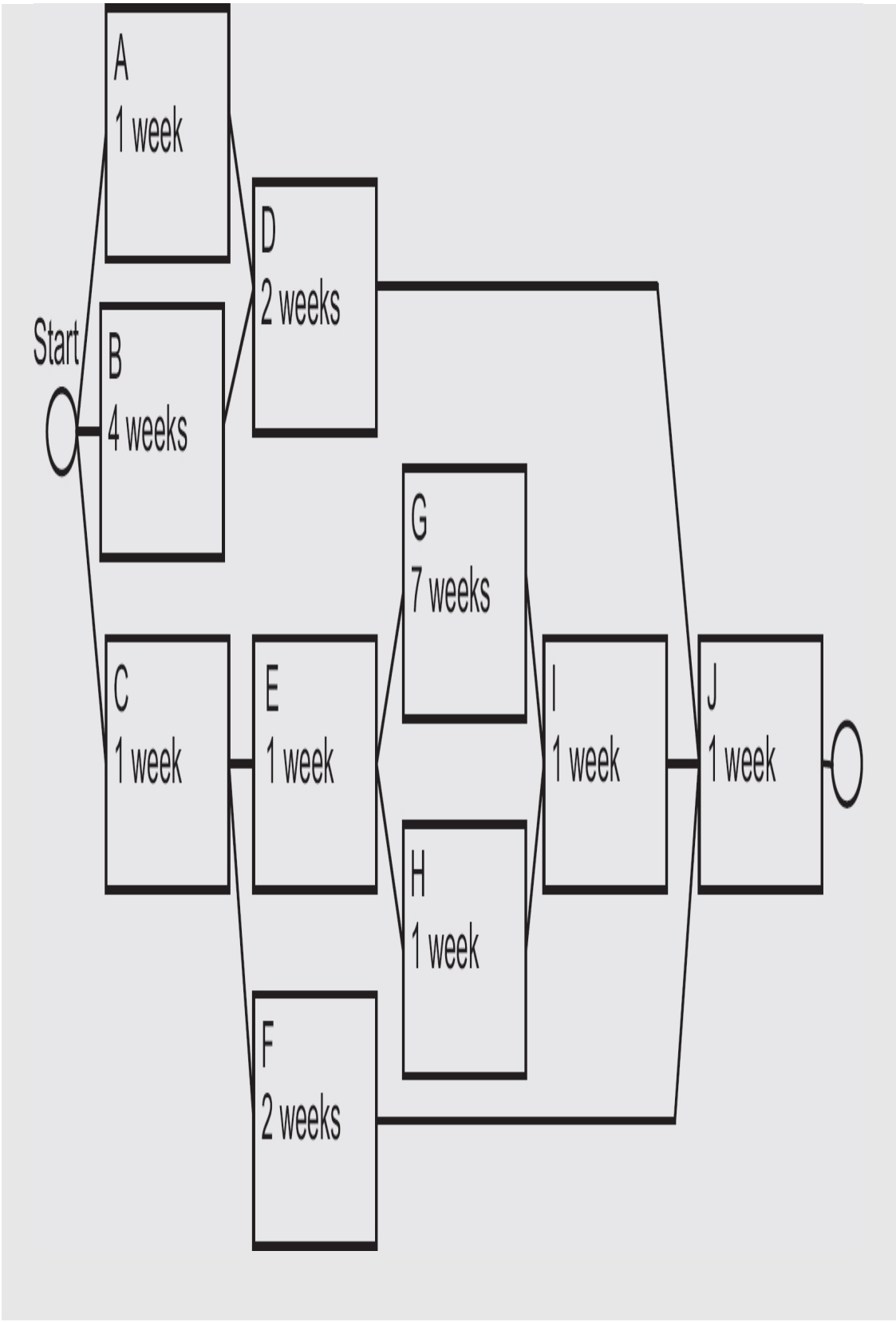
Figure 2.10 The activity span



The activity has to take place within the activity span. Because in Figure 2.10 there is some float, there is some freedom about when it can take place within that period. However, the start must be in the period between the earliest and the latest start. If it is not, it will not be completed within the activity span – unless its duration can be shortened in some way.

ACTIVITY 2.3

Calculate the earliest and latest start and finish weeks and floats for each of the activities in the activity network below. Use the results to identify the critical path.


Start
A
1 week
B
4 weeks
C
1 week
D
2 weeks
E
1 week
F
2 weeks
G
7 weeks
H
1 week
I
1 week
J
1 week

In Section 1.7.3 the iterative model was introduced where one or more activities could be repeated, with each loop creating a new version of the products of the process. Activity networks assume that all activities are executed just once. Clearly, some – such as usability tests – can be carried out more than once on revised versions of the software. We deal with this by concealing the iteration within an activity. Usability tests may be carried out a number of times on revised versions of the software, but all the iterations are together expected to take no longer than one week in the example.

2.5 RESOURCE ALLOCATION

So far we have taken no account of the availability of the resources needed to carry out any task. It is assumed that they will be available when they are needed. The resources that now need to be considered include raw materials, staffing and equipment. Usually with IT projects, the main concern is staffing, although sometimes equipment can also cause problems. For example, when conducting acceptance testing for a modified IT application, there is often a need for some testing of the whole operational system. This may need to be done on a public holiday or at the weekend, when no normal operational use is being made of the system. Here, we focus on staff resourcing.

For each activity, the **resource types** needed are identified. A resource type is a group of people of which any member could carry out a particular task. For example, if a software

component needs to be written in Java, identifying Ali as the needed resource would be too precise; Jane may be equally proficient in Java. Identifying the resource simply as a software developer, on the other hand, may be too vague: Alfred is a software developer but, as a Cobol programmer, he has no knowledge of Java. Identifying the required resource as a Java programmer may be just about right.

ACTIVITY 2.4

Match the following resource types and activities. More than one resource type might be needed for an activity.

Activities	Resource types
allocate room	business analyst
analyse business processes	human resources manager
carry out acceptance tests	interface designer
carry out usability tests	IT infrastructure support
devise usability tests	premises manager

design interface software developers

draft acceptance test cases users

install infrastructure

recruit staff

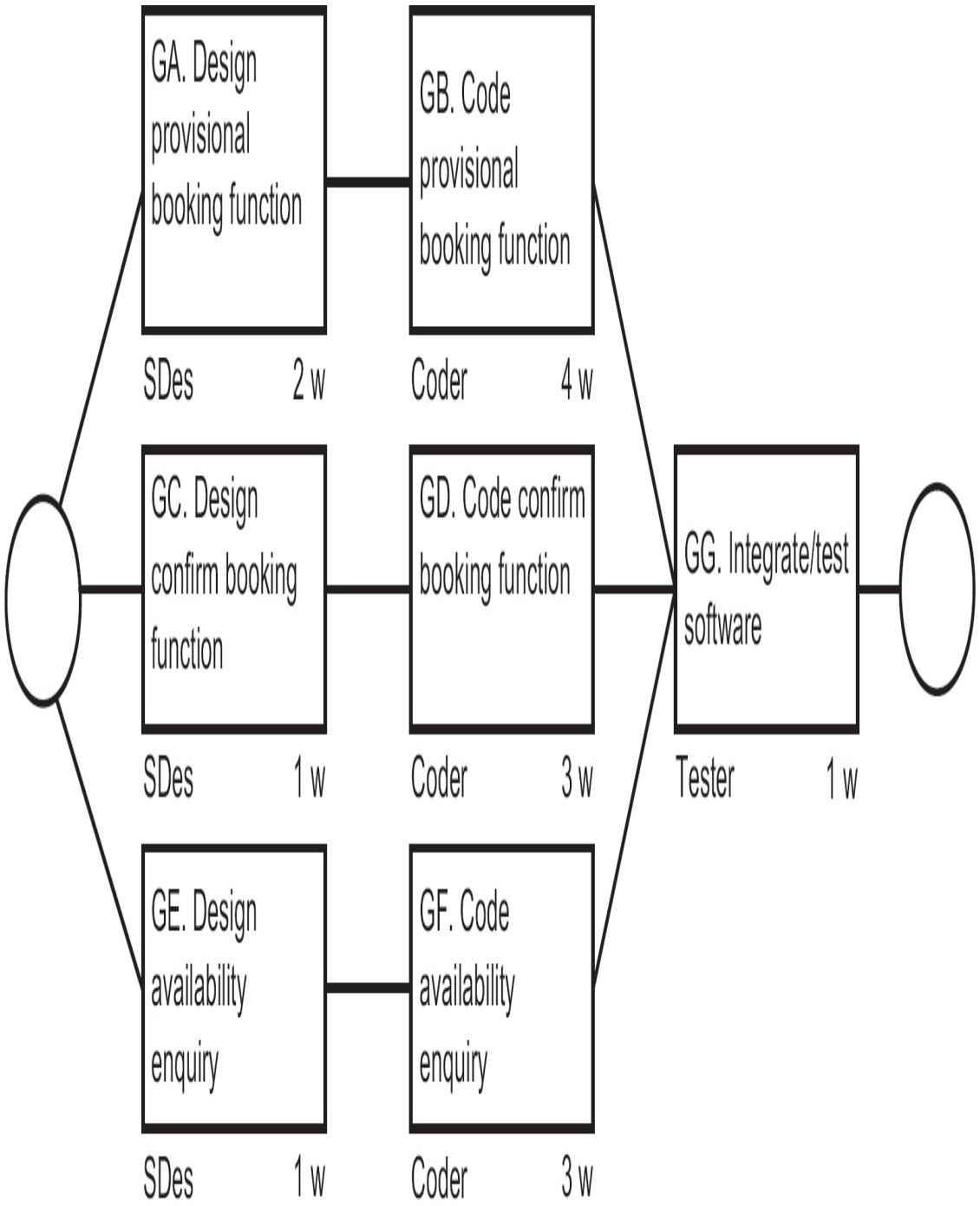
write/test software

In order to illustrate the process of resource levelling and smoothing, we break down one of the activities in our Water Holiday Company integration project, 'G. Write software', into more detailed tasks (see Figure 2.11).

Having allocated resource types to activities, we now go through the activity network and note the resources needed for each unit of calendar time (in this case each week) if the activity is to start at its earliest start date. In the top part of Table 2.1 we have, for example, identified the different resource types and allocated particular activities to them by putting the alphabetic characters we used in Figure 2.11 to identify each activity (for example, 'GA' for 'design provisional booking function') into the relevant week cells. Where the same type of resource is needed for different activities in the same week, we

identify different instances, for example SDes 1, SDes 2 and so on and allocate them to the different parallel activities where they are needed. When we have finished this allocation, we can count the number of each type of staff needed in each week. We can also depict this information as a **resource histogram** (see Figure 2.12).

Figure 2.11 Water Holiday Company project: 'G. Write software' activity

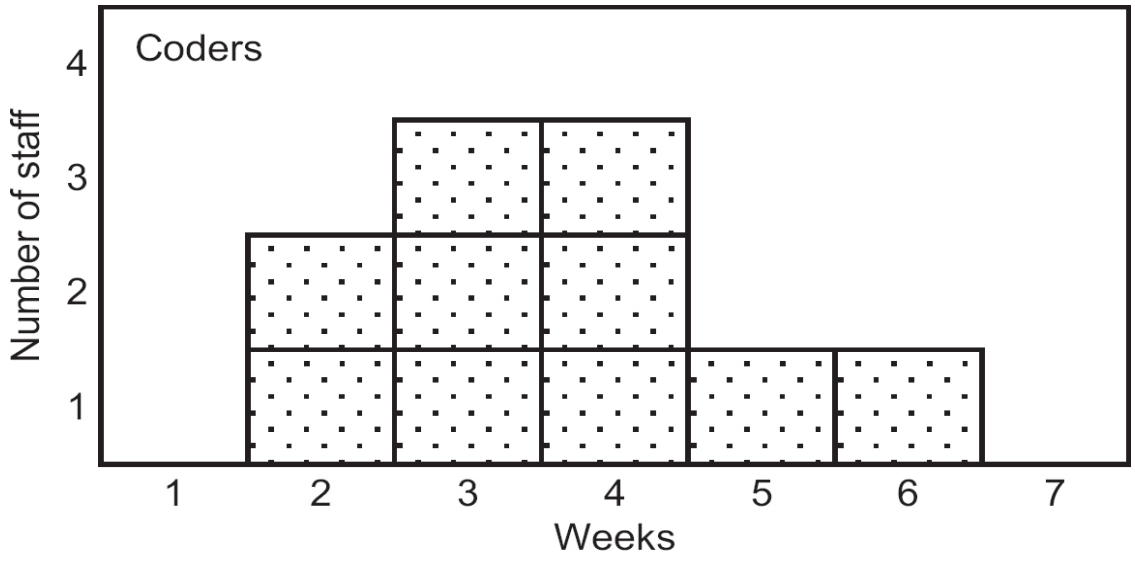


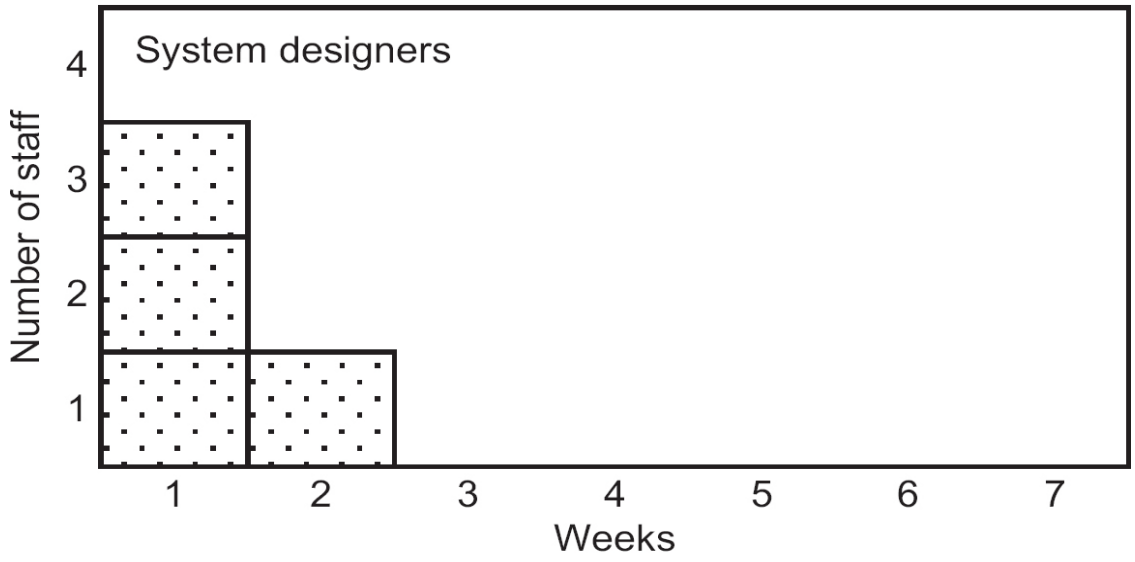
(SDes = System designer, w = week)

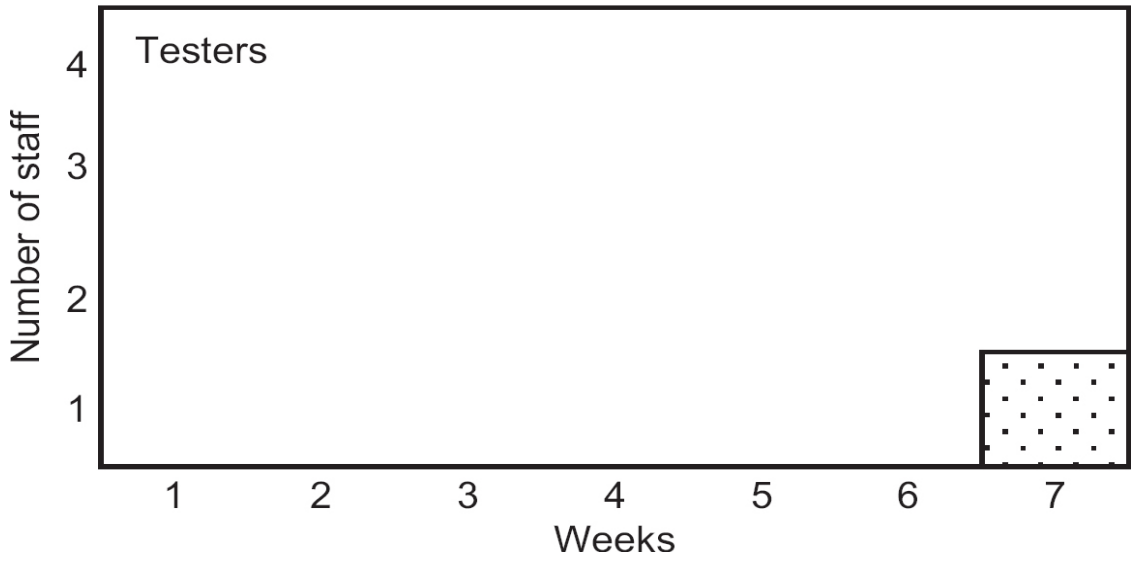
Table 2.1 Numbers of each resource type needed in each week

Week >	1	2	3	4	5	6	7
Resource	Activity they have been allocated to in each week						
System designers							
SDes 1	GA	GA					
SDes 2	GC						
SDes 3	GE						
Coders							
Coder 1			GB	GB	GB	GB	
Coder 2		GD	GD	GD			
Coder 3		GF	GF	GF			
Testers							
Tester 1							GG
Number of each resource type needed in each week							
System designers	3	1	0	0	0	0	0
Coders	0	2	3	3	1	1	0
Testers	0	0	0	0	0	0	1

Figure 2.12 A resource histogram for each resource type


Coders
Number of staff
4
3
2
1
1
2
3
4
5
6
7
Weeks


System designers
Number of staff
4
3
2
1
1
2
3
4
5
6
7
Weeks


Testers
Number of staff
4
3
2
1
1
2
3
4
5
6
7
Weeks

In some cases, the project plan needs more staff of a certain type at a particular time than we have available. Also, where temporary staff need to be employed on expensive contracts, we try to avoid having short periods of intense activity alternating with periods when nothing much is happening and staff are underemployed. Delaying certain activities may allow these peaks and troughs to be 'smoothed', which will lead to more economical staff costs and more productive work distribution.

For example, in Figure 2.12, three system designers are needed in week 1 and only one in week 2. If we have only two system designers on the staff, then we could employ a third system designer, possibly on an expensive temporary contract, for the first week. However, if we calculate the float for each activity in the activity network (Figure 2.11), we can see that two of the activities needing system designers have two weeks' float. If we delay starting one of these activities until week 2, this part of the project as a whole will not be delayed, but we will need only two system designers.

When there are not enough staff of a particular resource type to carry out the activities due to take place at a certain time, there is a **resource clash**. Even if there are no resource clashes, a planner should try to arrange activities so that the use of a resource type is as stable as possible.

ACTIVITY 2.5

Redraw Table 2.1 to take account of a delay of one week to the activity ‘design availability enquiry’.

Where there is a resource clash or the demand for a resource is very uneven, the following options may be considered:

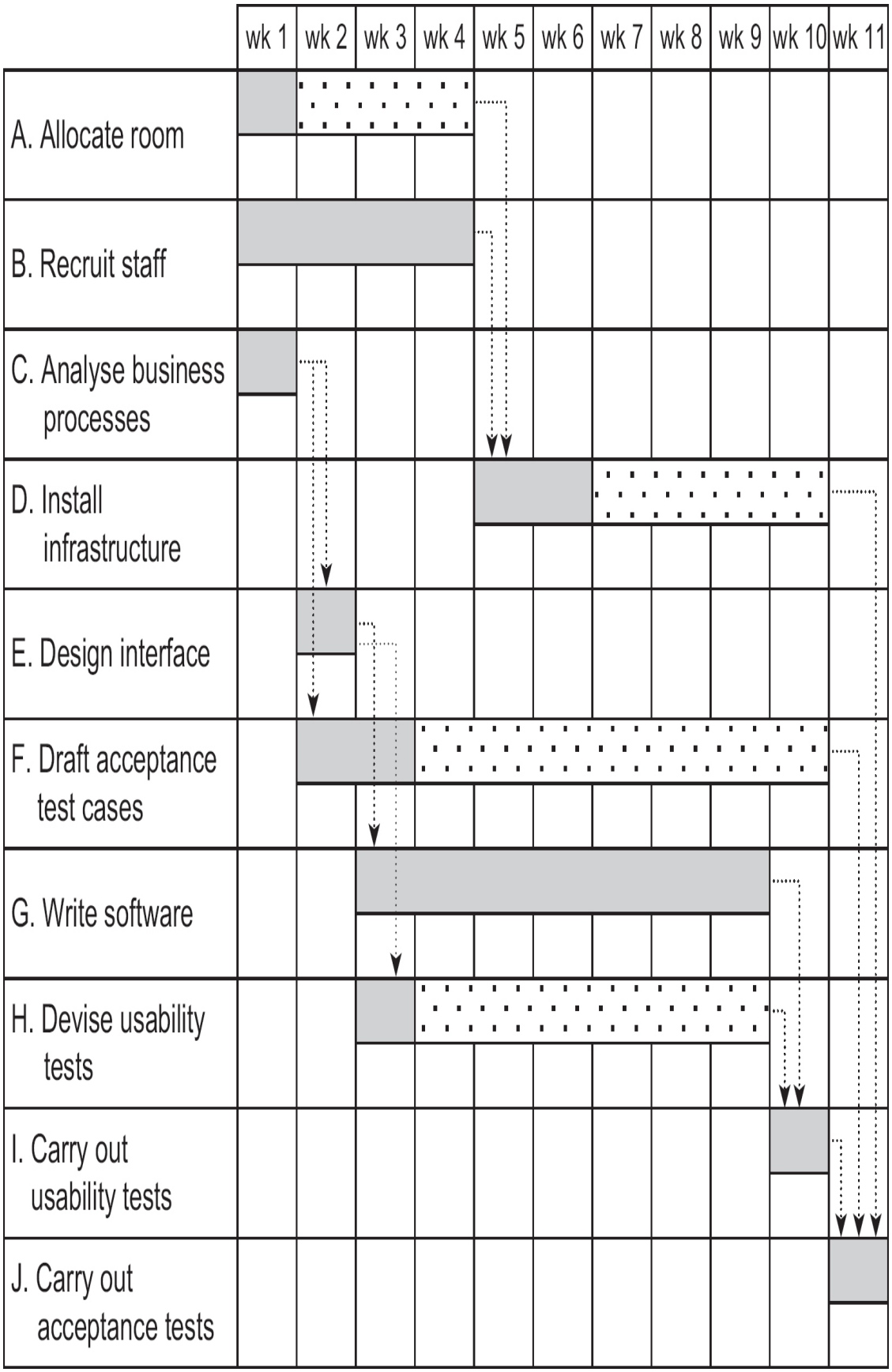
- Use the float of an activity to delay the start of some work until the required staff member is available (as in the example above).
- Delay the start of an activity even though the float has been used up. This will delay the overall completion date of the project, but that may be preferable to the extra cost of employing more staff.
- Buy in additional staff to cover the staff deficiency. This will normally increase the cost of the project.
- Split an activity into sub-activities. For example, it may be possible to split the provisional booking function into two component sub-functions, each requiring a week of design and two weeks of coding. This could allow the demand for systems design to be spread more evenly.

ACTIVITY 2.6

Redraw Table 2.1 to reflect the demand for the different types of resource if the provisional booking function is split into two equal-sized software components, each needing two weeks of coding, and a management decision is made to employ only one systems designer.

We are now in a position to put our plan into a form that will be easily understood by all those who are going to be carrying it out. The most common format used is a Gantt chart (see Figure 2.13). The activities are listed down the left-hand side and the calendar units (in this case, weeks) are shown along the top. In the body of the diagram there are blocks for each activity, showing when the activity will be carried out. In the diagram, the **free float** related to each activity is indicated by the lighter blocks that extend the base period for an activity.

Figure 2.13 Gantt chart


wk 1
wk 2
wk 3
wk 4
wk 5
wk 6
wk 7
wk 8
wk 9
wk 10
wk 11
A. Allocate room
B. Recruit staff
C. Analyse business processes
D. Install infrastructure
E. Design interface
F. Draft acceptance test cases
G. Write software
H. Devise usability tests
I. Carry out usability tests
J. Carry out acceptance tests

Free float is the amount of time that an activity can be late without any other activity being late. For activity A (Allocate room), this is 3 weeks. If activity A is later than 3 weeks, the start of activity D (Install infrastructure) will be delayed and activity D's free float will be reduced. However, activity A can be up to 7 weeks late and, as long as activity D does not take longer than planned, the project as a whole will not be delayed. The 7 weeks is known as activity A's **total float**.

For a smaller project, an alternative layout is to list each member of staff down the left and show what they are doing in each time period in the body of the diagram. This is not dissimilar to the holiday planning charts on the walls of many offices showing when staff will be away. A disadvantage of this format is that activities involving more than one team member have to be duplicated.

If an activity has to be delayed until a member of staff becomes available upon the completion of some other activity, this should **not** be indicated by a dependency link between the two activities on the activity network. Rather, the start date of the waiting activity should simply be amended on the Gantt chart. If other staff were to be released earlier than foreseen, they could be used to expedite the waiting activity. If the waiting task had a dependency link to another task, it would imply a technical reason for waiting until the other was complete and would mask the opportunity to use other staff.

ACTIVITY 2.7

What does GANTT stand for in the name ‘Gantt chart’?

2.6 USING SOFTWARE TOOLS FOR PLANNING

In this chapter it has been assumed that all the planning, and calculation of the consequences of particular planning decisions, will be done by hand with no computer assistance. It will be a relief for most people to know that there are software packages that will carry out most of the calculations for you. Examples of these are Microsoft Project and Oracle Primavera.

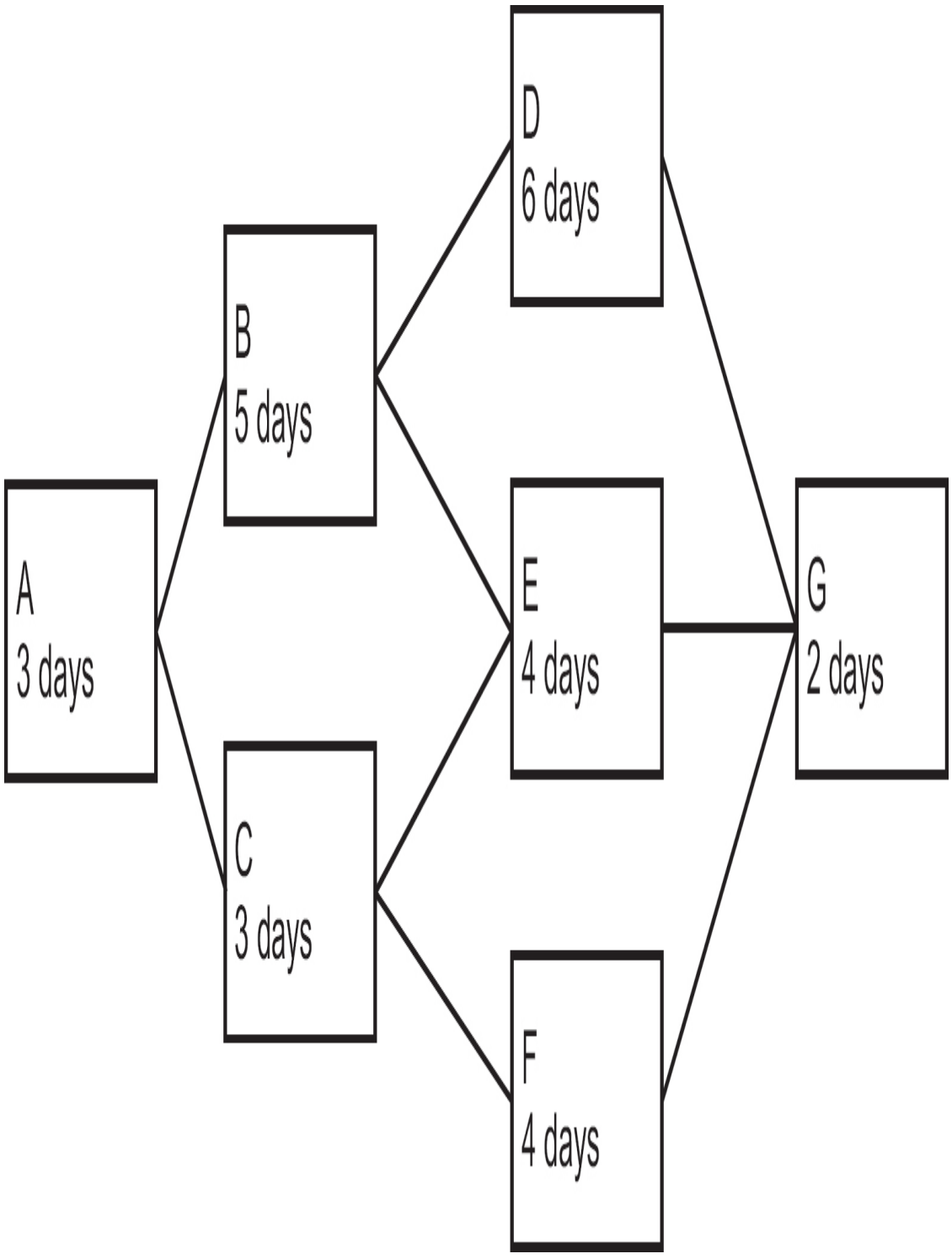
There are numerous alternatives available – just search on the internet for ‘project management software’. With many of these, the creation of a Gantt chart is just one function among a range of facilities to support large multiuser projects. Some provide the opportunity for a free limited trial period and there are also open source planning tools available.

In most cases, for each activity you will input the activity name, the duration of the activity, the activities upon which it depends and the resources that it will use. Given this information, the software then produces activity networks, resource histograms, Gantt charts and other useful reports. Some packages suggest ways of resolving resource clashes, but users need to check that the results are what they really want. Where activity networks and Gantt charts are produced, the planner often has to tweak them to make them easy for others

to understand. For example, a Gantt chart often spreads over several pages, many of which are blank. The advantage of using a software tool is that it is easy to make changes to the plan and see what the consequences of the changes will be.

SAMPLE QUESTIONS

Questions 1, 2 and 3 are about the diagram below. The number of days in each box show the duration of the activity.



1. Which of the following is the critical path?

a. A, B, D, G

b. A, B, E, G

c. A, C, E, G

d. A, C, F, G

2. What is the float for activity F?

a. 0 days

b. 2 days

c. 4 days

d. 6 days

3. Activities B and C have to be completed by the same person. What is the delay to the finish time of the project?

a. 3 days

b. 5 days

c. 4 days

d. 1 day

4. Which of the following does not take account of the dependencies between activities?

a. Gantt chart

b. activity network

c. work breakdown structure

d. resource histogram

POINTERS FOR ACTIVITIES

ACTIVITY 2.1

Among the products that may be created for each activity are:

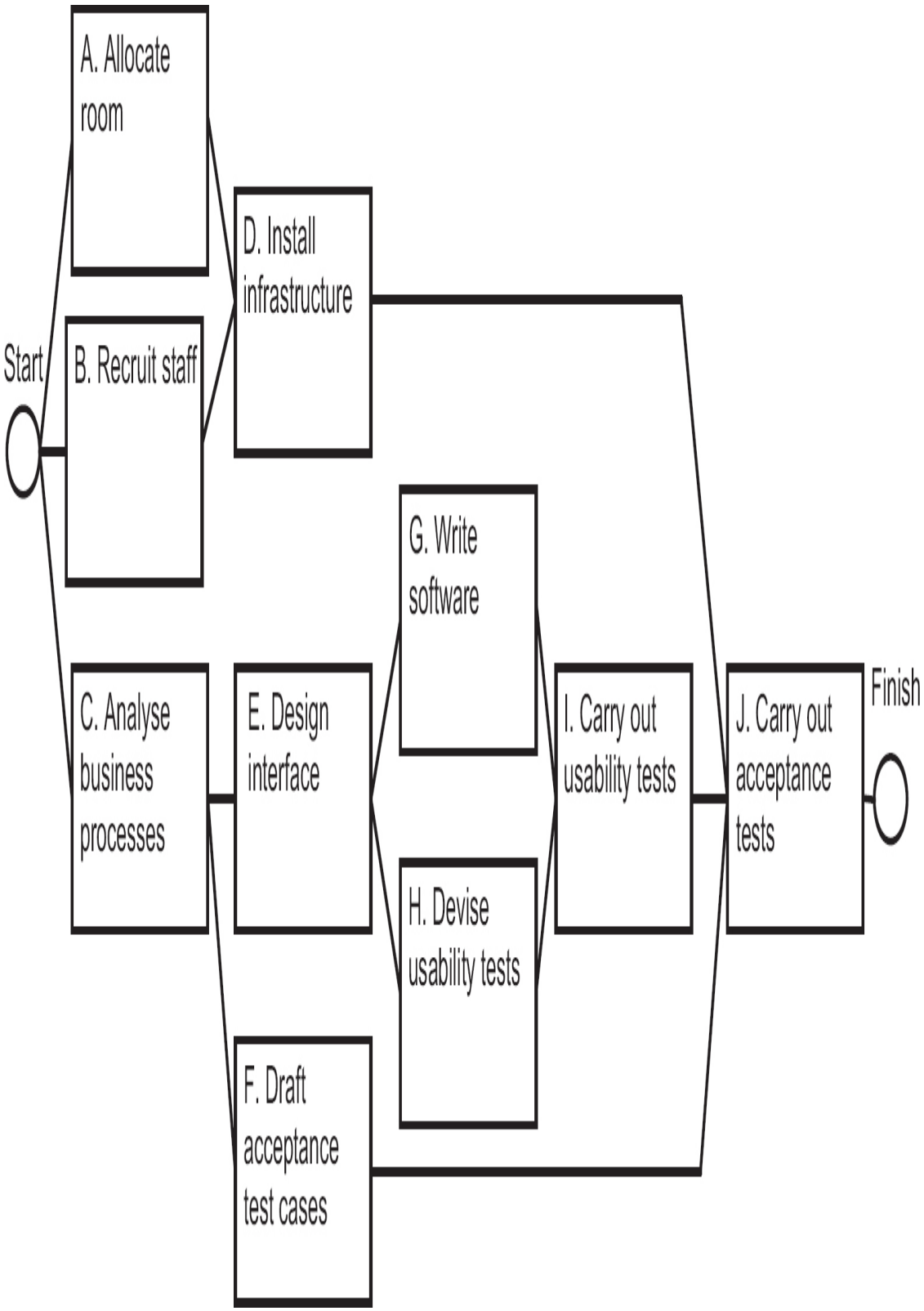
a. test results, error reports

b. trained users

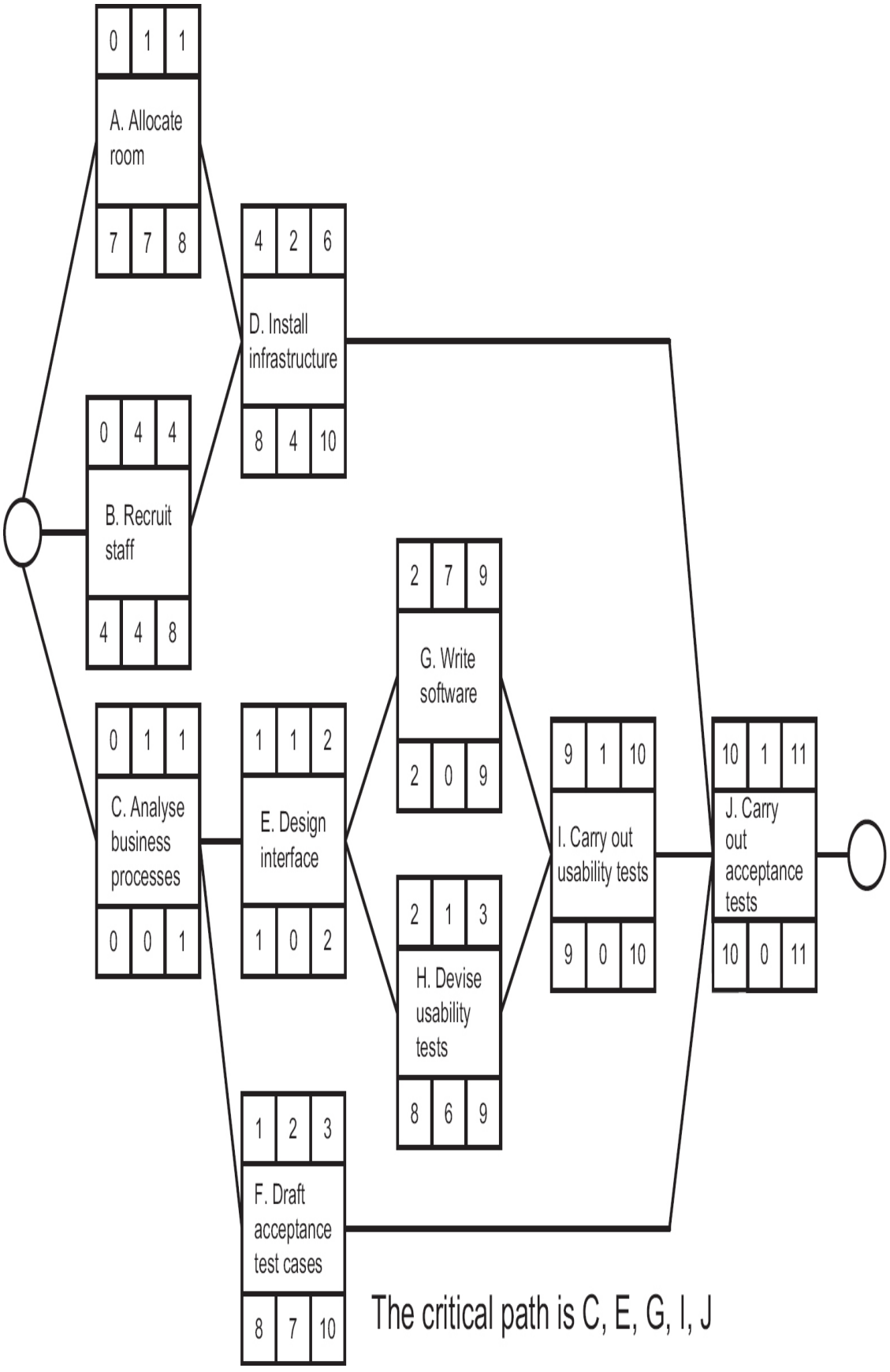
c. a new network

d. meeting minutes, to-do lists, updated plans

ACTIVITY 2.2


A. Allocate room
B. Recruit staff
C. Analyse business processes
D. Install infrastructure
E. Design interface
F. Draft acceptance test cases
G. Write software
H. Devise usability tests
I. Carry out usability tests
J. Carry out acceptance tests
Start
Finish

ACTIVITY 2.3


0 1 1
A. Allocate room
7 7 8
4 2 6
D. Install infrastructure
8 4 10
0 4 4
B. Recruit staff
4 4 8
0 1 1
C. Analyse business processes
0 0 1
1 1 2
E. Design interface
1 0 2
2 7 9
G. Write software
2 0 9
2 1 3
H. Devise usability tests
8 6 9
1 2 3
F. Draft acceptance test cases
8 7 10
9 1 10
I. Carry out usability tests
9 0 10
10 1 11
J. Carry out acceptance tests
10 0 11
The critical path is C, E, G, I, J

ACTIVITY 2.4

One way of allocating resources would be:

Activity	Resource type
allocate room	premises manager
analyse business processes	business analyst, users
carry out acceptance tests	users, business analyst
carry out usability tests	interface designer, users
devise usability tests	interface designer
design interface	interface designer, users
draft acceptance test cases	business analyst, users
install infrastructure	IT infrastructure support
recruit staff	human resources manager
write software	software developers

ACTIVITY 2.5

Week >	1	2	3	4	5	6	7
Resource	Activity they have been allocated to in each week						
System designers							
SDes 1	GA	GA					
SDes 2	GC	GE					
Programmers							
Coder 1			GB	GB	GB	GB	
Coder 2		GD	GD	GD			
Coder 3			GF	GF	GF		
Testers							
Tester 1							GG
Number of each resource type needed in each week							
System designers	2	2	0	0	0	0	0
Coders	0	1	3	3	2	1	0
Testers	0	0	0	0	0	0	1

ACTIVITY 2.6

Week >	1	2	3	4	5	6	7
Resource	Activity they have been allocated to in each week						
System designers							
SDes 1	GC	GE	GA(i)	GA(ii)			
Programmers							
Coder 1		GD	GD	GD	GB(ii)	GB(ii)	
Coder 2			GF	GF	GF		
Coder 3				GB(i)	GB(i)		
Testers							
Tester 1							GG
Number of each resource type needed in each week							
System designers	1	1	1	1	0	0	0
Coders	0	1	2	3	3	1	0
Testers	0	0	0	0	0	0	1

ACTIVITY 2.7

GANTT does not stand for anything. Gantt charts are named after their inventor, Henry Gantt. Gantt should, therefore, not be written in capitals!

3 MONITORING AND CONTROL

LEARNING OUTCOMES

When you have completed this chapter you should be able to demonstrate an understanding of the following:

- the project control cycle, including planning, monitoring achievement, identifying variances and taking corrective action;
- the nature of and purposes for which project information is gathered;
- how to collect and present progress information;
- the reporting cycle;
- how to take corrective action.

3.1 INTRODUCTION

Chapter 1 described the typical stages of a project that implements an information system. There we stressed the

importance of controlling the project to ensure that it conforms to the plan. In Chapter 2, we explained the way in which the plan for a particular project is created. ...

4 CHANGE CONTROL AND CONFIGURATION MANAGEMENT

LEARNING OUTCOMES

When you have completed this chapter you should be able to demonstrate an understanding of the following:

- the reasons for change control and configuration management;
- change control procedures:
  - the role of the change control board;
  - the generation, evaluation and authorisation of change requests;
- configuration management:
  - purpose and procedures;
  - the identification of configuration items;
  - product baselines;

- the content and use of configuration management databases.

4.1 INTRODUCTION

Business changes have many implications outside the narrow confines of IT development, including their impact on an organisation's staffing levels, skills and responsibilities. **Change management** is concerned with the smooth transition of the organisation to the new system that a project delivers. The Water Holiday Company integration project will create new software, change existing software and expand IT infrastructure. These are technical changes, but the integration project will also require changes to the way the merged business will be run.

The particular focus in this chapter, however, is on the events during a project that result in alterations to project requirements. The project manager's skill lies in controlling those changes so that minimal disruption is caused to the planned objectives of time, cost and quality. **Change control** ensures that modifications to any aspect of the project are only accepted after a formal review of their impact upon the project and its environment.

The procedures for managing change should be established at the beginning of the project. The planned functionality to be implemented may be modified during the project as the needs of the users and the business are clarified. These changes need to be tracked through a change control system. If outside

contractors are carrying out some of the work, the need for a change control system is even greater.

New IT systems usually need to be integrated with existing operational systems, and this may require modifications to these legacy systems so that they can communicate with the new functionality created by the project. The changes to these existing systems need particular care to ensure existing operations are not degraded.

Allowances need to be made within the project plans for the possibility of additional work caused by requested changes. These allowances could be part of the tolerances delegated to the project manager (see Chapter 3). The project manager ensures that these allowances are not exceeded. Where they are exceeded, a new project plan (the **exception plan** mentioned in Section 3.6) would need to be drafted and approved by the project sponsors.

Change control, at some level, should be applied to all changes, whether they arise from a change to user requirements or are due to design modifications. It requires participation by the users and the developers, guided by the project manager. The users must judge whether a change is essential, desirable or optional. This decision takes account of the possible impact of the change as well as of **not** making it. The project manager must assess the technical feasibility of the change, and identify its impact on costs, time and quality. Once account has been taken of users' and developers' advice, if it is decided to go

ahead with the change, it is the responsibility of the project manager to implement the change.

4.2 DEFINITION OF CHANGE

In Chapters 2 and 3, we introduced the idea of the project as a sequence of activities, each of which takes certain inputs and uses them to produce outputs. The outputs from one process may be inputs to others. Thus, a specification could be an input to a process that creates a system design (that is, the format of the user interfaces with the system) which will then be implemented as code. During the process, the proposed interface designs may change rapidly as the designers try out different designs and the users give feedback on them. This does not need a formal change process. At some point, however, a decision needs to be taken that the design is now in a satisfactory state for constructing the system. At this point, the design will need to be frozen; in other words, the design is **baselined**. If changes are needed to the design after it has been baselined, then rigorous change control is needed because of their potential impact on the processes that use the design.

The idea of baselines was introduced in Chapter 3 with regard to project plans. A **baselined plan** may be regarded as an agreed plan against which variations will be measured. Any change to requirements could imply a potential change to the baselined plan. For projects, the baselines include the agreed scope and quality of work, schedule and costs.

As noted in Chapter 3, these will tend to be interdependent. For example, if the scope of the work is expanded, then the cost will be increased. Similarly, cost or scheduled duration could be reduced by decreasing the functionality delivered.

Variations on these baselines can be categorised as follows:

- **Changes of scope**: These generally originate outside the project, usually by the users, managers or project sponsors changing the requirements, or by the cost or time constraints of the project changing. In the case of the Water Holiday Company integration scenario, the users may, for example, add a requirement that when booking a boat online the client should also be able to purchase holiday insurance. On the other hand, a reduction in the project budget may mean that some planned system features are dropped.
- **Development changes**: These originate within the project and include changes which are routinely carried out as part of the normal process of developing and refining a product. Typically, this can be something as simple as an adjustment to a screen layout.
- **Faults**: This type of variation also originates within the project and is caused by the project team making errors. For example, misunderstood requirements in coding detected by system tests could lead to additional corrective work.

Some discretion will be exercised in accepting or rejecting scope and development changes but changes to correct design errors will normally be obligatory, since the system may not work satisfactorily unless they are corrected.

ACTIVITY 4.1

In order to encourage the UK boating holiday industry, the government decides that VAT on canal holiday bookings will be zero-rated with immediate effect. The management of the Water Holiday Company calculate that this could increase the demand for bookings by 30 per cent. At the same time, the government introduces a special tax on holiday insurance that has to be accounted for separately on invoices. When considering the implications of changes, the project team realise that although holiday insurance was included in the original requirement, it has been missed out from the system design. What effect might these changes have on the project?

4.3 CHANGE CONTROL ROLES AND RESPONSIBILITIES

It is important to clarify the various roles and responsibilities in change control. We are going to use a very specific set of terms here to identify and explain the various roles and responsibilities. In a real project environment, it is very unlikely that these precise terms will be used. However, there

should be people who carry out the following roles, whatever they may be called.

- The **project manager** oversees the process and ensures that all **requests for change** (RFCs) are handled appropriately. In most cases, the project manager also has the role of change manager.

  Project managers must ensure that user representatives approve any changes made to the project requirements. They also control the scope of the system to be developed: additional features will require more effort and increase costs. When the project sponsor agrees to additional features, adjustments may need to be made to the project's contractual price. The project manager also needs to be on the lookout for informal changes made outside the process. Informal changes are often discovered during project reviews and a retrospective RFC form may need to be completed so that records remain accurate.

- The **change requestor** recognises a need for a change to the project and formally communicates this requirement to the change manager by completing the RFC form.

- The **change manager** (who may be the project manager) is responsible for logging RFCs in the **change register**. They decide whether a feasibility study is required for the change. Where this is desirable, for example because the extent of the impact of the change is uncertain, one or more people will be assigned to carry out the study.

- The **change feasibility group** appointed above investigates the feasibility of a proposed change. It is responsible for researching how the change may be implemented, and assessing the costs, benefits and impact of each option. Its findings are documented in a feasibility study report.
- **The change control board** (CCB) decides whether to accept or reject the RFC forwarded by the change manager. The CCB is responsible for:
  - reviewing all RFCs forwarded by the change manager;
  - approving or rejecting each RFC based on its merits;
  - resolving change conflict, where several changes overlap;
  - resolving problems that may arise from any change;
  - approving the change implementation timetable.
- A **change implementation group** carries out the change. It is usually made up of staff from the project team.

4.4 THE CHANGE CONTROL PROCESS

Having identified the roles and responsibilities involved in change control, we look at the processes in more detail.

4.4.1 SUBMIT REQUEST FOR CHANGE

In the Water Holiday Company integration project, the project manager has been allocated the role of change manager. An

experienced office manager of one of the current booking call centres is selected to act as change requestor. Requests for change can come from anyone, but are all passed to the change requestor, who, in consultation with colleagues, decides whether the requested change is desirable from the users' point of view. If it is, the change requestor submits a RFC to the change manager. The RFC provides a summary of the change required, including:

- a description of the change needed;
- the reasons for change, including the business implications;
- the benefits of change;
- supporting documentation.

There will almost certainly be a standard form for the RFC.

4.4.2 REVIEW REQUEST FOR CHANGE

The change manager reviews the RFC and decides the nature and scope of the feasibility study/impact analysis required for the CCB to assess the full impact of the change. In the Water Holiday Company integration project, one request was to add the sale of holiday insurance to the online booking system. The change manager saw that it was difficult to assess the scope of the change as exact details of the requirement were unclear. The impact of the change on the existing design also needed to be carefully considered. The change manager opened an RFC entry in the change register and recorded that a feasibility study was required.

4.4.3 ASSESS FEASIBILITY OF CHANGE

The assessment of the feasibility of the RFC not only considers business and technical feasibility but also the impact upon the project in terms of time, cost and quality. For small changes, a team member might assess the change in a relatively informal manner. For major changes, several people could be involved for some time. Different change options will be investigated and reported on. The change feasibility study will culminate in definition of the:

- change requirements;
- change options;
- costs and benefits;
- risks and issues;
- change impact;
- recommendations and plan.

A quality review of the feasibility study is then performed in order to ensure that it has been conducted as requested and the final report is approved, ready for release to the CCB. All change documentation is then collated by the change manager and submitted to the CCB for final review.

The feasibility study itself carries a cost and sometimes the project manager records these costs in the change register. These costs may well affect the project's budget. For external

client projects, the feasibility study may be an additional service which has to be paid for by the customer.

In the Water Holiday Company integration project, two staff were assigned to look at the RFC for a facility to purchase holiday insurance online. One was a business analyst, who focused on the business implications, and the other a developer, who considered the technical impact of the change. After discussing the matter with the user representatives, they found that the change required two additional input fields on the booking screen, two additional data items on one of the tables in the database and a link to a separate insurance products application. They estimated that the changes required 10 additional staff days of effort.

4.4.4 APPROVE REQUEST FOR CHANGE

A project may have a range of levels at which changes can be reviewed and approved:

- Team leaders may be allowed to accept changes that will not require additional resources, and which do not affect other baselined products.
- The project manager may be allowed to decide upon changes that have a minor impact on project objectives within a tolerance level which has been agreed with the steering committee or project board.
- The CCB decides upon changes that will have a larger impact upon project objectives, but are constrained by any limitations on the budget available for changes.

- Some changes may be particularly large but have compelling reasons for their adoption. These changes may need resources not envisaged in the original plans. In these cases, an exception report will need to be produced for consideration by the steering committee/project board – see Section 3.6.

Whatever the level of review, the change needs to be recorded and reported.

The CCB will do one of the following:

- reject the change;
- request more information related to the change;
- approve the change as requested;
- approve the change subject to specified conditions;
- put the change on hold;
- refer the change to the project sponsors if the current budget or project duration would be affected by the change.

The CCB takes account of the overall profile of possible changes. A large number of minor changes could have an overall effect out of all proportion to their individual significance. Approved changes will necessitate revisions to the schedule and cost plan. The CCB should prioritise essential changes, while non-essential changes may be delayed so that they make less impact on work schedules.

In the Water Holiday Company integration project, the holiday insurance change was only one of several RFCs. The CCB had a budget of only 30 staff days left to allocate, but had requests that needed a total of 35 days. The CCB accepted changes requiring up to 25 days' effort, including the change to incorporate the holiday insurance requirements.

4.4.5 IMPLEMENT REQUEST FOR CHANGE

This involves the complete implementation of the change, including any additional testing that may be required. Where existing code is changed, **regression** and **integration testing** will be needed to ensure that existing functionalities will not be harmed (see Section 5.8). On completion, the change will be signed off in the change register. Where the additional work has been carried out by an external supplier, additional invoices for that work will be raised by the supplier. Additional payments would not be made, however, where the change was to correct an error made by the supplier.

4.5 CONFIGURATION MANAGEMENT

The change control system above ensures that the business case for the project is not undermined by an endless sequence of changes – for example, by increasing costs beyond the value of the benefits of the project or by extending development time and thus delaying the benefits. Once a change has been agreed, there are further challenges, such as ensuring that all documents and other products of the project are modified to reflect the change.

ACTIVITY 4.2

What deliverables of a project may be affected by the change to the Water Holiday Company specification to allow for holiday insurance to be recorded when a customer books a boating holiday online?

A project has many documents relating to each phase of the project, along with software objects such as database structures and code segments. A very basic need is therefore a central **project repository** or **library** where master copies of all documents and software objects are stored. There should be a system of **version numbers** for all products so that successive baselines can be identified. These requirements make it essential for one or more people on the project team to take up a role variously called **project** or **configuration librarian** or **configuration manager**. Part of that role is to make sure that all project products are controlled, so that, for example, all software developers working on components that exchange information work from the latest component specifications.

Configuration management has three major elements:

- configuration item identification;
- configuration status accounting;
- configuration control.

4.5.1 CONFIGURATION ITEM IDENTIFICATION

The items that will be subject to the **configuration management system** (CMS) need to be identified. Typically, these are baselined specifications, design documents, software components, operational and support documentation, and key planning documents such as schedules and budgets. Other items, such as IT equipment, may also be subject to configuration management. These items will be defined as **configuration items** (CIs) and their details will be recorded in a **configuration management database** (CMDB). Among the details recorded in the CMDB for a configuration item are:

- a CI reference number;
- its current status;
- its version number;
- any larger configurations of which it is a part;
- any components that it has;
- other products that it is derived from;
- other products that are derived from it.

Note that a configuration item could have components that are configuration items in their own right. A change to component CI is also a change to the larger entities to which it belongs. The larger entity may need to be re-assembled to allow the component change.

4.5.2 CONFIGURATION STATUS ACCOUNTING

After a change to a CI has been agreed, the project librarian sets the status of the CI accordingly. Configuration status accounting maintains a continuous record of the status of the individual items that make up the system. Key status settings might include '*under development*', '*released for test*' and '*operational*'.

4.5.3 CONFIGURATION CONTROL

This ensures that due account is taken of the status of each CI. For example, when recording the change to add holiday insurance to the boat booking transaction in our Water Holiday Company example, the configuration librarian may access the CMDB to see the current status of the software component. The librarian may find that the software component is already booked out to a software developer who is implementing a different approved change to the module. If the librarian were to release a copy of the baselined code to a second developer to add holiday insurance, that would create two different versions of the same software. The new change may be given to the developer already working on the component or there may need to be a delay while the first change is completed.

When a developer is happy that all the work associated with a change is complete, the new version of the software is passed to the librarian. The librarian then records the CI as having a new version number and as being ready for acceptance testing. Acceptance testing is usually carried out in a separate IT environment to operational processing. If the designated user representative approves the revised version of the system, a

request can be made to the librarian to make the revised application operational. This will create a new baseline for the component, which can only be changed via change and configuration management processes described above. The former version of the software is retained in case there is a need to 'fall back' to it if the new version turns out to be problematic.

Where changes are made to currently operational functionality, current users would need to be made aware of the changes to their systems. Database structures may need modification, which affects the interactions with other system components. Where organisations have extensive user-facing websites that are being continuously updated with new features, they may adopt DevOps technologies that automate many of the technical implementation tasks, such as integration testing. These technologies do not diminish the need for managerial attention to ensuring the basic business value of changes is maintained.

SAMPLE QUESTIONS

1. **The change control board should be made up of:**
    a. representatives of the key stakeholders in the project
    b. the project manager and team leader
    c. the project sponsors
    d. the project support office

2. **If there are doubts about the projected costs of a proposed change, it would be the responsibility of which of the following to investigate?**

   a. the change requestor

   b. the change manager

   c. the change feasibility group

   d. the change control board

3. **As a documented procedure, what is the purpose of configuration management?**

   a. to ensure the project remains within budget

   b. to identify and document functional characteristics of a system

   c. to record and report changes and their implementation status

   d. to verify conformance with requirements

POINTERS FOR ACTIVITIES

ACTIVITY 4.1

The changes may have the following effects on the project:

- The change to the rate of VAT should **not** involve changing the application, as there should already be a system function that allows VAT rates to be changed.

- The potential increase in bookings would not change the functional requirements, but would probably change the quality or ‘non-functional’ requirements; for example, the database would need to be able to hold more records. The equipment needed to run the application might need to be upgraded (this is a change to the scope of the project).
- The statutory change to accommodate holiday insurance tax could well mean changes to input screens and report layouts. This does not increase the scope of the specification which included this functionality, but will change the scope of the planned work. Not including functions to deal with the requirement for holiday insurance in the design is a straightforward fault and the project team should correct it.

ACTIVITY 4.2

Among the many deliverables that might be affected are:

- the interface design – screens and report layouts;
- software components;
- the database structure;
- test data and expected results;
- user manuals;
- training material.

5 QUALITY

LEARNING OUTCOMES

When you have completed this chapter you should be able to demonstrate an understanding of the following:

- definitions of ‘quality’;
- quality control and quality assurance;
- measurement of quality;
- detection of defects during the project life cycle;
- quality procedures: entry, process and exit requirements;
- defect removal processes, including testing and reviews;
- types of testing (including unit, integration, user acceptance and regression testing);
- the inspection process and peer reviews;
- the principles of ISO 9000 quality management systems;
- evaluation of suppliers.

5.1 INTRODUCTION

The key quality concern for IT projects is providing customers with the systems they need and which meet their requirements …

6 ESTIMATING

LEARNING OUTCOMES

When you have completed this chapter you should be able to demonstrate an understanding of the following:

- the effects of over- and under-estimating;
- effort versus duration;
- the relationship between effort and cost;
- estimates and targets;
- use of expert judgement, including its advantages and disadvantages;
- the Delphi approach;
- top-down estimating;
- bottom-up estimating;
- the use of analogy in estimating.

6.1 INTRODUCTION

In Chapter 2, we explained how to draw up a plan for a project. This involved allocating an estimated duration to each of the activities to be carried out. This allowed us to calculate the overall duration of the project and to identify when we would need the services of individuals to carry out their tasks. In this chapter, we will explore the ways in which these estimates can be produced.

6.2 WHAT WE ESTIMATE AND WHY IT IS IMPORTANT

The success criteria for almost all projects will contain a required date for finishing the project and some constraint on the amount of money that is available for the project. Estimates of the duration of the project and its costs are therefore crucial.

6.2.1 EFFORT VERSUS DURATION

As well as estimating the time from the start to the end of an activity, it is also necessary to assess the amount of effort needed. Duration should not be confused with **effort**. For example, if it takes one worker two hours to clear a car park of snow then, all other things being equal, it takes two workers only one hour. In both cases, the effort is two hours but the activity **duration** is two hours in one case and only one hour in the other.

There can be cases where the duration is longer: for example, where someone only works in the afternoons on a particular task. Often activities take longer than planned even though the effort has not increased. This may happen, for instance, when

you have to wait for approval from a higher level of management before a job is signed off. This distinction between effort and duration can be particularly important when assessing the probable cost of a project, as on some projects staff costs are governed by the hours actually worked (typically where staff complete timesheets), while on others the costs are governed by the time people are employed on the project (even if there is not always work for them to do).

6.2.2 THE EFFECTS OF OVER- AND UNDER-ESTIMATING

If effort and duration are under-estimated, the project can fail because it has exceeded its budget or has been delayed beyond its agreed completion date. This may be so even when staff have worked efficiently and conscientiously. Allocating less time and money than is really needed can also affect the quality of the final project deliverables: team members may work hard to meet deadlines but, as a consequence, produce sub-standard work.

On the other hand, estimates that are too generous can also be a problem. If the estimate is the basis for a bid to carry out some work for an external customer, then an excessively high estimate may lead to the work being lost to a competitor. Parkinson's Law ('work expands to fill the time available') means that an excessively generous estimate may lead to lower productivity. If a task is allocated four weeks when it really needs only three, there is a chance that, with the pressure removed, staff will take the planned time.

6.2.3 ESTIMATES AND TARGETS

Identifying an expected duration is very difficult. If the same task is repeated a number of times, each execution of the task is likely to have a slightly different duration. Take going to work by car. It is unlikely that on any two days this takes exactly the same time. The journey time varies because of factors such as weather conditions and the pressure of traffic. Thus, an estimate of effort or time is really a most likely effort/time with a range of possibility on each side. Within this range of times we can choose a **target**. An 'aggressive' target may get the job done quickly, but with a stronger possibility of failure. A more generous estimate is likely to expand the length of time needed, but have a safer chance of being met. The target, if at all reasonable, can become a **self-fulfilling prophecy**; with the commuting example, if you know that you are going to be late you may take steps to speed up, perhaps by taking an alternative route if the normal one is congested. Estimating can thus have a 'political' aspect. Some managers may reduce estimates, either to gain acceptance for a proposed project, or as a means of pressurising developers to work harder. There are clearly risks involved in such an approach, as well as ethical issues.

6.3 EXPERT JUDGEMENT

Effective estimating needs contributions from experts with experience and knowledge of creating the types of product that the project is to create and the techniques by which they are created.

6.3.1 USING EXPERT JUDGEMENT

Where do you start if you want to produce reasonable estimates? Although estimating is treated as a separate, isolated topic in project management and information systems development, it in fact depends on the completion of other tasks that provide information for estimates. For a start, you need to know:

- what activities are going to be carried out during the course of the project;
- how much work is going to have to be carried out by these activities.

For example, to work out how long it will take to install upgraded workstations in an organisation, we need to know approximately how long it takes to install a single workstation and how many workstations there are. We may also need to know how geographically dispersed the workstations are. The best person to tell us about these things would be an expert familiar with the tasks to be carried out and the environment in which they are to be done. As a consequence, most guides to estimating identify **expert opinion** or **expert judgement** as an estimating method.

'Phoning a friend' can be a sensible approach, but how do the experts themselves derive their estimates? There is a possibility that they have their own experts they can call, but at some point someone has to work out the estimate based on their own judgement – and the likelihood is that they end up comparing

current tasks with previously completed ones and using the actual durations from those old tasks as a basis for the new estimates.

The advantages of using expert judgement include the following:

- You involve people with the best experience of similar work in the past and the best knowledge of the work environment.
- If the people most likely to do the work participate in estimating it – they will be more motivated to meet the targets they themselves have set.

There are, however, some balancing risks:

- The task to be carried out may be a new one of which there is no prior experience.
- Experts can be prone to human error – they may, for example, under-estimate the time that they would need to carry out a task in case a larger figure suggests that they are less capable.
- It can be difficult for the project planner to evaluate the quality of an estimate that is essentially someone else's guess.
- Large, complex tasks may require the expertise of several different specialists.

6.3.2 THE DELPHI APPROACH

One method that attempts to improve the quality of expert judgement is the **Delphi technique**, which originated in the RAND corporation in the USA. There are different versions of this, one of which is '**planning poker**', but the general principle is that a group of experts are asked to produce, individually and without consulting others, an estimate supported by some kind of justification. These are all forwarded to a moderator who collates the replies and circulates them back to the group as a whole. Each member of the group can now read the anonymous estimates and supporting comments of the other group members. They may now submit a revised estimate with its justification. Hopefully, the opinions of the experts should converge on a consensus.

The justification for the technique is that it should lead to people's views being judged on their merits and reduce undue deference to more senior staff or the more dominant personalities.

6.4 BOTTOM-UP AND TOP-DOWN APPROACHES

Note that **bottom-up** and **top-down** approaches are not specific estimating methods, but two groups of estimating methods.

6.4.1 BOTTOM-UP

With bottom-up approaches, we break the task for which an estimate is to be produced into component sub-tasks and then break the component sub-tasks into sub-sub-tasks and so on,

until we get to elements that we think would not take one or two people more than a week to complete. The idea is that you can realistically imagine what can be accomplished in one or two weeks in a way that would not be possible for, say, one or two months. To get an overall estimate of the effort needed for the project, you simply add up all the effort for the component tasks.

This method is also sometimes called **analytical** or **activity-based estimating**. Some people (especially software developers) find the name 'bottom-up' confusing because the first part of the process is really top-down!

ACTIVITY 6.1

Which planning product identified in Chapter 2 could be the basis for an initial bottom-up estimate?

A bottom-up estimate is recommended where you have no accurate historical records of relevant past projects to guide you. A disadvantage of the method is that it is very time-consuming as, in effect, you have to draw up a detailed plan for the project first. Of course, you are going to have to do this anyway at some point. However, it may be a very tedious and speculative task if you have been asked for a rough estimate at the feasibility study stage of the project proposal.

ACTIVITY 6.2

You have been asked to organise the recruitment of staff for the new network support centre needed as a result of the Water Holiday Company integration project. Identify the component activities in this overall task, as you would for the first stage of the bottom-up approach to estimating effort.

6.4.2 TOP-DOWN

With the top-down approach, we look for some overall characteristics of the job to be done and, from these, produce a global effort estimate. This figure is nearly always based on our knowledge of past cases.

An example of top-down estimating is when house owners make decisions about the sum for which they should insure their house. The question here is the probable cost of rebuilding the house in the event of it being destroyed, for example by fire. Most insurance companies produce a handy set of tables where you can look up such variables as the number of storeys your house has, the number of bedrooms, the area of floor space, the material out of which it has been constructed and the region in which it is located. For each combination of these characteristics, a rebuilding cost will be suggested. The insurance company can produce such tables as it has records of the actual cost of rebuilding houses.

This is essentially a top-down approach because only one global figure is produced. In the unhappy case of a fire actually occurring, this figure would not help a builder to calculate how

much effort would be needed to dig the foundations, build the walls, put on the roof and all the other individual components of the building operation. However, a builder may be able to use past experience of the proportions of total costs usually consumed by foundation digging and other activities.

6.5 A PARAMETRIC APPROACH

The base estimate created when using a top-down approach can be derived in a number of ways. In the example of estimating the costs of rebuilding a house, a **parametric** method was used. This means that the estimate was based on certain variables or parameters (for example, the number of storeys in the house and the number of bedrooms). These parameters can be said to 'drive' the size of the house to be built: you would expect a house with three storeys and five bedrooms to be physically bigger than a bungalow with only two bedrooms. These parameters are therefore sometimes called **size drivers**. As values of the size drivers increase so would the amount of effort, so these can also be called **effort drivers**.

6.5.1 SIZE DRIVERS AND PRODUCTIVITY

Earlier we had an example where technicians were allocated the job of installing upgraded workstations in an organisation. Clearly, the more workstations there are, the bigger the job and the longer its duration. Hence the number of workstations is a size driver and an effort driver for this activity.

ACTIVITY 6.3

Identify the possible size and effort drivers in the Water Holiday Company integration for each of the following activities:

- creating training material for users;
- analysing business processes;
- carrying out acceptance tests;
- writing and testing software.

In order to produce an estimate of effort using this method, we also need a **productivity rate**. For example, in addition to the number of workstations we would need to know the average time needed to install the software on a single workstation. If the average was 12 minutes per workstation and there were 50 workstations, then we could guess the overall duration of the job would be around 50 × 12 minutes – that is, about 10 hours.

Ideally the productivity rate comes from records of past projects. Where these are not available, you can sometimes obtain 'industry' data that relate not to projects in a single organisation, but in a particular industrial sector. This kind of information can help managers to compare the productivity in their organisation with that of others – this is sometimes called **benchmarking**. If they find that they have much lower productivity, this may spur the search for more productive

ways of working. However, caution needs to be practised if the reason for using industry data is that local project data is missing; there can be large differences in productivity between organisations, because organisations and their businesses are so different.

ACTIVITY 6.4

In the earlier example about the time needed to drive to work, identify:

1. the size driver;
2. the productivity rate;
3. other factors that may cause a variation in the time it takes to get to work.

The additional factors are called **productivity drivers**. A key productivity driver when it comes to developing and implementing IT systems is experience. When putting a figure on how long a technical activity is going to take, such as developing software code, more experienced estimators will try to find out how experienced the people doing the work are.

Productivity drivers vary from activity to activity, but other drivers often include:

- the availability of tools to assist in the work;

- communication overheads, including the time it takes to get requirements clarified and approved;
- the stability of the environment – that is, the extent to which the work has to cope with changes to requirements or resources;
- the size of the project team: there is a tendency for larger jobs involving lots of people to be less efficient than smaller ones because more time has to be spent on management, planning and coordination at the expense of 'real work'.

The problems that can affect productivity are often considered at the same time as risks to the project in general (see Chapter 7).

6.5.2 FUNCTION SIZE MEASUREMENT

There was a time when almost all IT projects involved writing software of some description. This is now diminishing for many reasons, one of which is the tendency to use 'off-the-shelf' software applications. However, there are still many cases where software has to be written, and can cause particular challenges for effort estimation.

If we use a parametric approach, the first question is what to use as size drivers. If IT is old enough to have any real 'traditions', then one of the longest established of these would be to use **lines of code** as the size driver for software development. (When software is written, the programmer writes the instructions – as lines of code – in a form which is comprehensible to human experts. This 'code' is an electronic

document that can be changed, added to and printed. When the code is to be executed by the computer, the document is 'read' by a special piece of software which converts it into a format that the computer can interpret automatically.) From this very brief explanation it can be seen that:

- the code is a very technical product – it would need a software expert to estimate the number of lines of code;
- you will not know the exact number of lines of code until quite near the end of the project; most other size drivers are known at the beginning, or at least at an early stage, of the project.

Things are also complicated by there being many programming languages. Some are more 'powerful' than others – that is, they need fewer lines of code to define a particular procedure.

Rather than use this technical unit of size, which is invisible to everyone except the software developers, it is more convenient to use counts of externally apparent features of the software application as the size drivers. This would be rather like using the number of storeys, the floor space and so on to estimate the cost of a house, rather than the number of bricks. With software applications, this can be done by applying function points analysis (FPA).

COSMIC function size measurement

For the purposes of the Foundation Certificate, you do not need to know the details of the rules of function size measurement (FSM).

There are several different ways of carrying out FSM; the following is based on COSMIC function points and should be enough to give you a broad idea of the approach.

The general aim of the approach is to calculate an index number that correlates with the amount of functionality in a proposed system component and hence the likely effort needed to develop the software for the component.

Each component has a **boundary** with the outside world. Functions carried out by the component are always started by **events**, which are usually inputs from an external user of the component. As will be seen, this user is not necessarily a human.

The event is usually accompanied by data. In COSMIC, the inputs are called **entries** (E). For example, a potential customer could access the website for the Water Holiday Company and request information from the system about holidays. This event could be accompanied with data about the type of boat and the particular weeks the user is interested in. The system component can pass back messages to the user in response to the entries. These are known as **exits** (X). For example, exits could inform the user about the availability of the type of boat and weeks requested.

The amount of data that passes over the boundary is measured in terms of the number of logical entities to which it relates. For example, booking a boat would involve entities such as boat, customer and booking. Each of these will be dealt with by distinct sets of entries and exits.

As well as passing groups of data backward and forward across the boundary, the system component will also be accessing and updating internal data stores. In the booking example, the component would need to consult details of the types of boats and the weeks that are available in order to reply to the user's request for information. This movement from a data store is a **read** (R) in COSMIC terminology.

When a new booking is made, the persistent data store would need to be updated with details of the new customer and their booking. In COSMIC terms this would require at least two **writes** (W).

A count can be made of all the entries, exits, reads and writes contained in the component and this would be the **COSMIC function points** (CFPs) for the component.

Note that this measurement does not take account of internal processing such as calculations. The developers of the COSMIC approach found that the amount of this type of manipulation tended to be in step with the number of entries, exits, reads and writes. There could be some

complex scientific applications where calculations could be significant, and a separate assessment is needed for this.

The above examples from the Water Holiday Company assume that the 'system' is one single component, but in reality a system will usually have a number of layers. For example, the component could obtain data (entries) from an interface layer, rather than directly from the human user, and update data stores by calling a data management layer. In this case, the communications with the external layers can be analysed as entries and exits.

6.5.3 EXAMPLE OF COSMIC FP COUNTING

Within the Water Holiday Company booking system there is a transaction that records the final payment made by a customer who has already booked a holiday and paid a deposit. The details of the payment come from an external function that deals with payments by debit and credit cards. The key data items input are an account number, an amount and a date. The system checks that an account exists for the account number and, if it does not, issues an error message (see Table 6.1). Otherwise the new payment is recorded, and the amount paid on the account record is updated. A notification of the new amount outstanding on the account is issued.

Table 6.1 Example of COSMIC function point counting

Functional step	Data items/entity attributes	Entity	E/X/R/W	CFP
Input of payment	Account number	ACCOUNT	E	2
	Payment date	PAYMENT	E	
	Payment amount			
Check valid account	Account number	ACCOUNT	R	2
Error			X	
Update account details	Account number	ACCOUNT	W	2
	Total amount paid			
	Payment date	PAYMENT	W	
	Payment amount			
Notify new account balance	Account number	ACCOUNT	X	2
	Total amount paid			
	Value of sales			
	Balance			
	Payment date	PAYMENT	X	
	Payment amount			
Total CFPs				8

What does a CFP count really mean? It was suggested above that it is an index value that gives an idea of the amount of processing carried out by the transaction.

We can use a CFP count to find out the relative productivity of development projects that have already been completed. We may find that the average number of CFPs implemented per day is around five. This may seem a rather small number, but 'development effort' here includes the whole development cycle, from requirements gathering to testing. When a new project proposal comes along, a preliminary investigation may suggest that the delivered system would have a count of about 250 CFPs. The estimated effort is therefore in the region of 250/5 days – that is, 50 days.

6.6 ESTIMATING BY ANALOGY

The FSM approach (and, indeed, the more generic approach of using size drivers and productivity rates) is based on the assumption that we have the details of the size driver values and actual effort of past projects. Often, however, such records do not exist. For smaller organisations particularly, the IT projects that have been previously implemented may all seem to have their own peculiarities. For example, some may have involved the installation of vendor-supplied applications, some may have required specially written software, some a mixture of the two, and so on. This seems to suggest that previous experience is not a stable basis for estimating the effort for new

projects. However, in this kind of situation the **analogy**, or **comparative**, approach could be used.

The main steps with this method are as follows:

1. Identify the key characteristics of the new project.
2. Search for a previous project which has similar characteristics.
3. Use the actual effort recorded for the previous project as the base estimate for the new one.
4. Identify the key differences between the old and the new projects (it is unlikely that the old project is an exact match for the new one).
5. Adjust the base estimate to take account of the identified differences.

An analogy approach can be used to create a top-down estimate for a project. Where there is no past project that seems to be a useful analogy for the new project, an estimator can use analogy to select parts of old projects that seem similar to components of the current project (using analogy as part of a bottom-up approach).

As Table 6.2 shows, both analogy and the parametric approaches can be used either at the overall level of a project or for estimating the effort needed for components. The activity-based approach – breaking down the overall task into smaller components – seems almost by definition to be a bottom-up approach.

Table 6.2 Relationship between top-down/bottom-up and the three main estimating approaches

	activity-based/ analytical	parametric	analogy/ comparative
top-down		✓	✓
bottom-up	✓	✓	✓

6.7 CHECKLIST

As a project planner you may often need to use the effort estimates produced by experts from technical areas in which you are not knowledgeable. Are there any ways in which you can realistically review these estimates? It may be possible to assess the plausibility of the estimates by asking the estimator the questions below.

- What methods were used to produce the estimates?
- How is the relative size of the job measured (in other words, what are the size/effort drivers)?
- How much effort was assumed would be required for each unit of the size driver (in other words, what productivity rates are you assuming)?

- Can a past project of about the same size be identified which had about the same effort?
- If a job with a comparable size cannot be identified, can past jobs which had similar productivity rates be found?

SAMPLE QUESTIONS

XYZ ORGANISATION SCENARIO

Staff have managed to develop information systems at a rate of five function points per staff day. A new system has been assessed as requiring 120 CFPs to implement, but the staff available are relatively inexperienced and are only 80 per cent as productive as the staff usually used in such projects.

1. **An under-estimate of effort is MOST likely to lead to which of the following?**

   a. decreased productivity

   b. decreased quality

   c. a less competitive bid for a contract

   d. a longer project duration

2. **Which of the following estimating methods is MOST likely to be used bottom-up?**

   a. parametric

b. algorithmic

c. Delphi

d. activity-based

3. **In the XYZ scenario, which one of the following is 80 per cent?**

   a. a size driver

   b. an effort driver

   c. a productivity rate

   d. a productivity driver

4. **In the XYZ scenario, what would be the best estimate of effort for the project?**

   a. 30 days

   b. 25 days

   c. 24 days

   d. 20 days

POINTERS FOR ACTIVITIES

ACTIVITY 6.1

The work breakdown structure (or possibly the product breakdown structure).

ACTIVITY 6.2

Among the activities that may need to be carried out are:

- Create/agree job descriptions.
- Create job advertisements.
- Collect and assess applications and curricula vitae (CVs) from potential employees.
- Invite selected candidates for interview.
- Interview candidates.
- Notify successful and unsuccessful candidates.
- Request, await and check references.
- Confirm appointment.
- Arrange induction.
- Carry out induction processes.

This set of activities offers some good illustrations of the difference between elapsed time and effort. There will be some points – for example, when you are waiting for references – where little effort is expended but time will be passing.

ACTIVITY 6.3

The following are suitable answers:

- The number of functions that users need to be able to use.
- The number of different types of system user (as each will need to be interviewed for their requirements), and the number of different operations carried out in the system.

- The number of functions to be tested and the number of input and output data items to be tested.
- The number of different functions in the system, the number of inputs, outputs and tables accessed.

ACTIVITY 6.4

1. The size driver would be the distance driven to work.
2. The productivity rate would be the average speed of the car (for example, miles per hour).
3. We have already suggested that the weather and the amount of traffic congestion could have an effect on the travel time. In this case, the weather and traffic do not increase the size of the job to be done – the distance to work remains the same. These factors are best seen as influences on the productivity rate. In order to assess more accurately the time it takes me to go to work, I could take account of these intermittent constraints on my speed. I may be aware, for instance, that the rush-hour traffic in the morning tends to be significantly less heavy during school holidays. I could therefore perhaps allow myself to start off to work a few minutes later when it is half-term. On the other hand, I may start earlier if it is foggy, as I know that this can slow down the traffic.

7 RISK

LEARNING OUTCOMES

When you have completed this chapter you should be able to demonstrate an understanding of the following:

- identification and prioritisation of risks;
- assessment of the probability and impact of risks, that is risk exposure;
- risk reduction activities versus contingency actions;
- typical risks associated with information systems;
- assessment of the value of risk reduction activities;
- maintenance of risk registers.

7.1 INTRODUCTION

However carefully a project plan is assembled, it will be based on assumptions that could turn out to be wrong. There is always, to a greater or lesser extent, an element of risk, which has been defined in PRINCE2 as:

‘The chance of exposure to the adverse consequences of future events.’ …

8 PROJECT ORGANISATION

LEARNING OUTCOMES

When you have completed this chapter, you should be able to demonstrate an understanding of the following:

- relationship between programmes and projects;
- identification of stakeholders and their concerns;
- the project sponsor;
- establishment of the project authority (for example, project board, steering committee, etc.);
- membership of project board/steering committee;
- roles and responsibilities of the project manager, team managers and team leaders;
- desirable characteristics of a project manager;
- role of the project support office;
- the project team and matrix management;
- reporting structures and responsibilities;

- management styles and communication;
- team building and dynamics.

8.1 INTRODUCTION

The aim of this book is to describe ‘best practice’ in project management. Unfortunately, not all organisations follow these good practices, which is one of the reasons why there are project failures.

Some of these good practices relate directly to the personal skills and qualities of the project manager, but others relate to the organisational structures within which the project manager and the project team works. In this chapter, we describe the elements of the project management structure and roles that should exist in an organisation planning and executing a project. These roles may be known by different names, so wherever possible we will give commonly used alternative names of which we are aware. We will also touch upon the management approaches that project managers might adopt and the strategies for effective communication within the project.

8.2 PROGRAMMES AND PROJECTS

A **programme** is a collection or group of related projects. IT projects are often treated as individual and separate undertakings with their own distinct aims and objectives. However, sometimes grouping projects into programmes has advantages:

- **Reduction of duplication** – two projects may be developing a similar product.
- **Coordination of resources** – projects inevitably place demands on an organisation's resources, and it is far better to coordinate than to compete for them.
- **Management of interdependencies** – if projects are dependent on one another, they have to be properly coordinated. In such circumstances, a change in one project may introduce a delay or a consequential change in another.
- **Common objectives** – sometimes several different projects need to be completed for a particular business objective to be achieved.

Typically, each project is assigned a project manager, while a programme is led by a **programme director**, who should be an influential member of the organisation's senior management team acting as a champion for the programme. The senior status needed for this role means that it is usually only part-time. The role of **programme manager** is likely to be full-time and deals with the day-to-day coordination of the programme.

The integration of two businesses into one Water Holiday Company can illustrate the value of managing change as a programme of projects. The intention is to relocate the operations of the merged companies onto a 'green-field' site. In Chapter 7, we noted that the IT infrastructure installation

would need to wait for the construction work to be completed. The construction work would use different sets of skills and resources to other integration activities, so could be usefully treated as a distinct project within a broader integration project.

The implementation team for the installation of the IT infrastructure would have their own concerns about having new premises suitable for the IT infrastructure. Thus coordination, particularly during the later stages of fitting out, would be needed that could be ensured through effective programme management.

8.3 IDENTIFYING STAKEHOLDERS AND THEIR CONCERNS

A **stakeholder** is defined as anyone with a valid interest in an IT project or the products delivered by it. This group includes:

- all project personnel including managers, designers and developers;
- users, sponsors and other members of the organisation affected by the project;
- suppliers of software, hardware, consultancy and so on, to the project;
- contractors and subcontractors;
- members of the business community and, of course, financial backers.

Part of the project initiation process establishes who these people are and identifies their needs and concerns. Processes will have to be set up to ensure that their interests are represented and that they are consulted and kept informed. A simple table, identifying the major stakeholders, their interests, concerns and contributions, can be the basis of a **communication plan** – see Section 8.12 and Table 8.3.

ACTIVITY 8.1

List the main stakeholders in the Water Holiday Company integration project.

8.4 THE ORGANISATIONAL FRAMEWORK

A formal management structure is needed with defined project roles:

- Named personnel should be allocated to the roles, but several people may share a role and a single person may have more than one role.
- Roles must carry appropriate authority, and responsibility. **Accountability** is sometimes distinguished from **responsibility**. Broadly, it refers to having a duty to ensure something is done to meet a project requirement. Those responsible for carrying out the practical tasks needed to fulfil the requirement are monitored by the accountable person.

- Individuals must carry out their roles correctly and willingly and must clearly understand the objectives behind their work.
- Status within the organisation is not sufficient qualification for a particular role. Previous experience and/or training in the role is needed.

At the top of the project organisation is a **project sponsor**. This person would be a senior person within the organisation where the new system will be implemented. They champion the project at an appropriately high and influential level – normally board level. They represent the ultimate authority for the project. Crucially, the sponsor controls the funds to pay for the project. Below the project sponsor are the following roles:

- project board (or steering committee or project management board);
- project manager;
- project team leader/manager;
- team member.

In parallel to this structure there could be other functions:

- project assurance team;
- project support office or project management office;
- configuration management team.

The roles that they represent should be present in all projects to a greater or lesser degree, although they may have different names. Where a project is small, for example, the project manager and project team leader roles could be carried out by a single person.

8.4.1 PROJECT BOARD/STEERING COMMITTEE

We noted earlier the role of the project sponsor who is guardian of the business case for the project. They represent the customer who supplies finance for the project to be completed in order to achieve the benefits that fulfil their needs.

The **project board** provides a forum in which critical issues can be discussed and decisions taken that are outside the remit and competence of the project manager. The same function may be served by a **steering committee** or a **project management board**. This group includes the project sponsor or their representative and representatives of other major stakeholders. The PRINCE2 standard explicitly identifies the need for a senior supplier and user to be present. The group is not democratic as the project sponsor (referred to sometimes as the '**Executive**' or the '**Senior Responsible Owner**') chairs the meeting and has the final say.

Members of the board must be able to agree decisions relevant for their areas of responsibility. If they cannot, they will have to refer decisions back to their managers, and thus delay the project. However, if they are at too high a level, they may

attend meetings infrequently, leading to ineffective decision-making.

All projects should derive from the organisation's business strategy and meet specific business and corporate objectives. It is therefore the role of the project sponsor, apart from looking after the money, to ensure that the project:

- stays in line with the corporate objectives;
- meets the business requirements;
- retains its business case (that is, that the benefits continue to outweigh the costs of the project).

The project board/project steering committee must have one or more representatives of the users. These should ensure that the user requirements are captured, and deliverables are signed off as acceptable. They must check that changes to requirements are in line with the business needs and do not jeopardise the overall project objectives. As was seen in Chapter 4, a volume of perfectly valid changes can delay the project to such an extent that it will not deliver the expected benefits.

Suppliers of technical resources, including skilled staff and the software and equipment to be supplied, should be represented. They should also support the development team and, when necessary, represent their interests as difficulties arise, for example in the face of continual change requests. Sometimes the project manager could carry out this role.

If all or part of the project is contracted to an external supplier, that organisation should be represented. If there are a number of contractors, then more than one supplier representative may be required. However, this could detract from the main purpose of the project board/steering committee and therefore it may be more effective to set up a separate group to manage the external suppliers.

The **project assurance function** could also have a representative on the board or could report via another member of the board, ideally the project sponsor.

Large, geographically dispersed projects may need several user representatives, but care must be taken that the board is not too big and thus ineffective. Subgroup meetings can give a voice to those who should have one without detracting from the efficient working of the board; smaller groups are usually more effective at decision-making.

The infrastructure management of the business, responsible for the platforms on which the delivered IT will run, may be represented. Development staff are more likely to be project-orientated with a single project as their priority, while the work of infrastructure support staff may be structured by job queues dealing with requests from many different users with different needs and priorities. These two valid outlooks need to be carefully reconciled.

The project board is a decision-making body. It holds the purse-strings through the project sponsor and therefore has

ultimate control. The project manager will often be appointed by and report to the board. The board establishes the terms of reference and provides the management framework within which the project manager operates. It is the final arbiter on whether the project has met its objectives. In summary, the board initiates the project, controls its execution and eventually closes it down. Led by the project sponsor, it approves the following:

- project terms of reference (including the project initiation document);
- business case;
- budget;
- project plans;
- changes to project plans;
- quality plans, control and assurance processes;
- risk assessment and contingency plans;
- major changes to project requirements.

It receives feedback on the progress of the project from the project manager and also from the quality assurance function. The feedback enables them to sign off each stage of the project or other activity as defined in the plan, or require that it be reworked. It thus exercises overall control of the project.

In the following sections, references to the project sponsor usually mean the ‘project sponsor working through the project

board/steering committee'.

8.4.2 PROJECT MANAGER

The project manager is pivotal in the organisation of the project and has overall responsibility for the day-to-day management of the project. The role of the project manager is to ensure that the project is delivered on time, within budget and with the specified functions and quality.

The project manager produces, with the help of others, the various plans for the project (see Chapter 2). They monitor progress against the plan and make adjustments throughout the project. As milestones are reached, progress is reported to the board and team members (see Chapter 3). The project manager is set tolerances for activity completion and costs within which to work. Any deviation in the plans likely to take the project outside these tolerances should be reported to the board via an exception report with recommendations about the actions the project manager feels are necessary to correct the situation or to reduce its effect.

Inevitably, changes come along. The project manager assesses their impact on the project and reports to the board. It is the responsibility of the board to decide whether or not changes should be implemented, although, as seen in Chapter 4, this responsibility may be delegated, within constraints, to a change control board.

Should a risk materialise that is likely to affect agreed plans, the project manager will approach the board for permission to

put into effect any agreed contingency plan. The project manager is accountable for keeping up-to-date logs of both change requests and risks (see Chapters 4 and 7, respectively).

Because project managers have to lead their projects, they must be able to set clear goals so that those who report to them are aware of what is required. The simple production of plans and schedules does not by itself achieve this – see Section 8.5.

8.4.3 TEAM LEADER/TEAM MANAGER

The terms team leader and team manager are often used interchangeably. However, it is useful to distinguish between the two roles. Team leaders are each responsible for their own individual work team carrying out specific tasks needed to implement a work package. Team managers are in overall charge of a pool of specialist developers. They would have overall accountability for the execution of work packages authorised by the project manager.

Team leaders are close to the action. A team leader may be responsible for a specialist group of analysts, designers or programmers and hence work on a very specific part of the life cycle. Alternatively, on smaller projects, a team leader may lead a mixed team, in which case the responsibilities would encompass the whole of the life cycle. In the case of small projects, the project manager and team leader roles could be merged.

A team leader usually needs technical experience and knowledge. One specialised area is that of testing, where a team

becomes expert at the creation of test data based on the requirements specification, which is then run through the software to see whether or not they meet those requirements.

Team leaders have the task of allocating the tasks needed to implement an authorised work package to specific individuals and helping them to complete the activities within the scheduled time scales. They also act as mentors or advisers when necessary. They must be aware of how well their part of the project is going. They should be able to quickly notify the project manager (perhaps through a team manager) of delays so that action can be started to bring the project back on course before it goes seriously adrift.

8.4.4 TEAM MEMBER

However good the structure may be, without competent team members the project will not succeed. Ideally, team leaders should be able to select their teams. This is rarely possible, so the team leader has to be familiar with the qualities of the staff allocated to them and assign them to the tasks which most match their capabilities.

As we will see in Section 8.8 on matrix management, specialist staff often have only a short-term project relationship with the team leader, while their longer term development and deployment is the responsibility of a separate technical head (who could be called a team *manager*).

8.4.5 PROJECT ASSURANCE TEAM

This is usually a function rather than an actual team. The project sponsor and project board are **accountable** for project assurance; that is, they will be held to account for the good governance of the project. However, depending on the size and nature of the project, it is normal for responsibilities for project assurance to be delegated to one person or a small group. The tasks are the same; it is just the degree of activity that differs. Members of the project assurance function are appointed by and report directly to the project sponsor and the project board, not to the project manager. Their role is firstly project assurance (see Chapter 5), which focuses on ensuring that the project management procedures laid down in the project management plan are being followed effectively; for example, do the reports to the project board accurately reflect the true status of the project? Is risk management successfully controlling risks? Is the business case still relevant? They are responsible for checking that the quality control activities in the quality plan are carried out, standards are observed and procedures are followed. They will almost certainly **not** be carrying out quality control activities, such as testing, themselves. Being independent of the project manager, they can give an objective view on the quality of the products delivered and not just the timeliness of their arrival.

In the early stages of a project, the team would contribute to the setting of standards, the creation of procedures and the establishment of the quality review, quality assurance processes and the quality plan. This is particularly useful if the

project is venturing into areas of technology that are new to the organisation.

A large, complex IT project may need several people for this activity, each with their own specialism, such as security. Three specific roles sometimes identified are:

- **business assurance co-ordinator**, who, for example, would make sure that the IT application under development is compatible with existing business procedures;
- **technical assurance co-ordinator**, who would ensure that the operational environment is not compromised by the new system;
- **user assurance co-ordinator**, who ensures the application meets user needs.

ACTIVITY 8.2

Identify two managers that you have reported to. It could be one was more effective than the other. What were their good qualities and defects as managers?

8.5 DESIRABLE CHARACTERISTICS OF A PROJECT MANAGER

The role of the project manager is crucial. IT project managers traditionally progressed from technical areas such as software development through system design and team leadership to project management. However, this is changing as IT projects

are perceived more and more as business change programmes. (Put another way, nearly every business change programme will have IT elements.) It is also the case that the project management role is increasingly involved in managing the external suppliers of services.

Thus, while it is useful to have familiarity with IT issues and concerns, it is not always essential in order to manage what may really be a business change project. It is more important to have other skills and characteristics. A key quality is good communication skills at all levels. The project manager often needs to present a convincing case for a course of action to higher management. Stakeholders have to be kept informed and the project teams must be motivated. To gain the respect and confidence of those around them, project managers have to be effective communicators, good managers and effective organisers.

They need to have skills in:

- leadership (which includes the ability to provide direction and guidance when needed);
- motivation;
- planning;
- negotiation (being firm, flexible and able to compromise where appropriate);
- delegation.

They need to be:

- responsible;
- reliable;
- available (not just for this project, but contactable at all reasonable times);
- intelligent;
- sociable (able to mix well);
- approachable (they should be good listeners);
- knowledgeable in the business area for the particular project.

These characteristics do not just arrive with seniority. A potential project manager must possess some of these attributes and will need development in deficient areas before becoming fully effective.

Although this list may seem daunting, the ability of people who seem quite 'ordinary' to become competent project managers through application, self-discipline and appropriate training is remarkable.

8.6 PROJECT SUPPORT AND MANAGEMENT OFFICES

The qualities we have associated with project leadership seem heroic, but projects have many mundane clerical activities that have to be performed regularly, effectively and efficiently. The project manager is accountable for them, but, given the pressures on project managers, it is more effective to delegate

these tasks. This gives rise to what has become known as the **project support office** (PSO).

The organisational structure described in Section 8.4 is usually set up for a specific project. This means that the project board and project manager are appointed for a single project. The PSO, on the other hand, may be a longer living entity that supports several, often interrelated, projects. Where projects are interrelated, they may be coordinated as programmes. In these cases, it is convenient to combine project support with programme support – hence, we have a **programme and project support office** (PPSO).

The largely clerical nature of many of these tasks means that they are often amenable to automation, and this may be reducing the importance of the PSO. Instead, the concept of the variously named **project** or **programme management office** (PMO) – or even **programme and project management office** (PPMO) – has been developed, which is more focused on the strategic aspects of projects, such as training and the identification of good project management practice. This could include **portfolio management**, which is concerned, among other things, with the prioritisation of project proposals.

The precise tasks this office undertakes will vary, but some typical PSO services are described below.

8.6.1 TIME RECORDING

Time recording is essential for project control. In the past, clerical staff would collect and process the information required, but nowadays it is likely that project staff submit timesheet details online. Relieved of routine clerical work, project managers can focus on where effort expended varies from that expected and take necessary actions.

8.6.2 UPDATING PLANS

As details of progress are passed to the PSO, staff can use appropriate tools to update the plans and highlight problems to the team leader or project manager. In the event of changes to project team membership, the PSO can revise the plans so that the project manager is aware of the impact of such changes and can make any necessary modifications or alert the board to potential problems.

8.6.3 MAINTENANCE OF LOGS

The PSO can maintain project logs. The project manager can be warned of approaching risks via the risk register (see Chapter 7). Risks that have passed can also be noted. Requests for change (RFC) are recorded as they arrive and then passed on for action (see Chapter 4). Once again, there are tools that facilitate online input and the automated processing of data.

8.6.4 ARRANGING MEETINGS

A great deal of time, most of which involves finding dates and times at which all parties are available, can be spent on arranging meetings. Aids such as electronic diaries make this

easier, but it can still be time-consuming. Delegation to the PSO is natural. Having agreed dates and times, the PSO can:

- organise the necessary room and other requirements such as catering;
- issue the agenda – most meetings have a set agenda and it is simple to check with the chair of the meeting to see if any changes are required;
- circulate other documents needed – the PSO may already carry out the configuration control function, which controls the versions of documents;
- record and distribute the minutes of meetings – the key thing is to ensure that all actions are identified, along with who is responsible for carrying them out and in what time frame.

8.6.5 CONFIGURATION MANAGEMENT

The whole of Chapter 4 has been given over to change control and configuration management. The ability to keep track of documents and products to ensure that everybody is working with the latest version is very important to the success of the project. Like the PSO, within which it may function, configuration management has a life beyond the duration of the project, as the delivered IT system will continue to require amending and updating until it is finally replaced.

8.7 PROJECT TEAM

A project team is a small group of individuals with complementary capabilities working towards a common goal; they are responsible to and are guided and coordinated by a leader who is accountable to a project manager. We noted earlier that it was possible for the team leader and project manager roles to be merged.

A project team works in a different way from operational staff. It is brought together for the sole purpose of achieving the project objectives. On completion of the project, the project team can be disbanded. Team members are often drawn from specialist divisions within the organisation, such as the IT department, to work on the project and are assigned to other projects when it is over. Others may be selected from the user community as 'super-users' for their knowledge of the operational aspects of the application area and then return to their original jobs. The effect of being in a project team will be quite different for the two types of people, as will their expectations during and after the project.

Project team members are not always located together, although this is best; the team could be geographically dispersed. It may be possible to bring them together for short periods, but this would be expensive. Part of the solution to these problems could be matrix management.

8.8 MATRIX MANAGEMENT

So far it has been assumed that all staff report to the project manager, either directly or through team managers or leaders.

This is the best way of managing a project but is not always practical, particularly if team members are drawn from a variety of departments.

Senior staff on a project – for example, team leaders – normally report to the project manager. On large projects there could be a hierarchy where more junior team leaders report to the project manager through more senior, higher-level leaders. Management control may be weakened if there is not a clearly defined reporting structure, to the detriment of the project.

In matrix-managed teams, an individual reports to more than one person. One example of matrix management can be found on board a naval ship. The captain of the ship is responsible for everything that happens on the ship when it is at sea. Its complement of sailors will include navigators, engineers, electricians, cooks, medical staff and others. Each specialised trade will also have a shore-based manager to whom they will report and who is responsible for their performance on the ship and their training when ashore.

In an IT environment, many part-time team members are juggling their time between projects and have a line manager outside the project. The project manager would want commitments from line managers that required staff will be made available when needed for a project.

Organisations generally have a number of projects under way at the same time, each requiring a range of different skills. For example, business analysts may elicit requirements from the

users and ensure that they are clearly understood by the technical members of the project team. Analysts and designers would then need to convert the requirements into a design specification. They could come from a specialist department and normally report to a separate manager. Another department manages the IT infrastructure. They could second someone to the project to guide the designers so that the proposed solutions are compatible with the existing infrastructure.

Both software developers and testers could come from separate pools of specialists, or even come from a specialist group outside the organisation.

Putting all these together gives rise to a matrix structure as shown in Table 8.1.

Table 8.1 Example of a matrix organisation

Project	User department	Business analysis department	Infrastructure management	Software development department	Software assurance
A	X	X		X	X
B	X		X		
C			X	X	X

This sort of structure has the advantage that the project manager has a team of specialists and can concentrate on the project in hand and not be concerned with issues such as the long-term development of the staff involved. A risk is that a line manager could call their staff off the project for some higher priority task, such as emergency maintenance on an operational system. These issues should be discussed at the beginning of the project and a strategy agreed by the project board, to which all line managers sign up.

Even where all staff, including the project managers, work for the same line manager, they could be allocated to more than one project. Each person may have their own perceived

priorities and may put more effort into one project than another. This can lead to slippage on some projects, while others progress to schedule.

We noted in Section 8.2 that where many projects are carried out at the same time, particularly if they are related, there is often an umbrella organisation known as **programme management**. Where resources are limited, which is the normal situation, a **programme director** advised by a **programme board** would have the remit to make the final decisions about the allocation of resources between projects. While this may be irksome for the individual project manager, the organisation benefits as a whole. Obviously, plans have to be revised to take account of this situation and project managers cannot be held accountable for delays to their projects due to such changes.

With matrix management, the level of control that the project manager will have over the project team will vary. Where an individual is seconded to the project for its duration, then the project manager has greater control – as with the ship's crew. This is in many ways the ideal, but it does have its drawbacks. With every project, the level of resource required will vary over time. For example, in the early, analysis stages of a project, several analysts but only one part-time designer may be required. However, as the analysis phase is completed, more designers may be required, but fewer analysts. If these staff were permanently with the project they would be underutilised at times, adding unnecessary costs to the project.

8.9 TEAM BUILDING

It is possible that a group of developers who already work together are employed on a project. However, as projects are by definition temporary organisations created to carry out one-off undertakings, it is often the case that a project will bring together people who have not all worked together previously.

A team in an IT project is usually a collection of specialists with a requirement to work together towards a common goal. The composition of the team is likely to be based on compromises and not be the ideal combination of skills and personal characteristics. There are many common-sense aspects and a number of theoretical approaches to developing an effective team.

The Tuckman–Jensen model describes four basic phases through which a team goes before it becomes fully effective:

- **Forming:** The group members have just been brought together and are probably hesitant about their new environment, unsure of their new colleagues and possibly nervous about future developments. Members are polite to one another, tend to accept authority and tread carefully. Some initial contact with colleagues reveals common ground and possible allegiances.
- **Storming:** Individuals within the group have started to assert themselves and to form alliances. Conflict may arise as ‘pecking orders’ become established. Aims and objectives are becoming clearer, but there are different

views on how to proceed with the tasks ahead. Members now have a sense of belonging to a team, are gaining confidence and are likely to challenge the proposed methods of working.

- **Norming:** Internal conflicts are hopefully resolved, and the team members feel more comfortable and relaxed with their colleagues and their new surroundings. An acceptance of common values and behaviours develops, with open communication and constructive cooperation. The team starts to work as it should, with its overall capability being greater than the sum of its parts.
- **Performing:** The team is fully functional and has become a cohesive unit. Team morale is high, with good cooperation between members and a shared responsibility for the common goal. Team members are working hard and getting satisfaction as the team achieves its goals.

A fifth stage, **adjourning**, is sometimes added to describe when the team breaks up at the end of the project. The Tuckman–Jensen sequence of phases clearly represents the ideal team development. Good team leadership and people management are essential to allow the team to progress through the phases, and indeed the final stage may never be achieved if the personnel or the project circumstances are not right. It is quite possible to slip back a stage or two if unexpected developments are not well managed; for example:

- changes in team personnel – new arrivals can disrupt team morale and stability;
- a change in direction of the project, which means much team effort has been wasted;
- a change of leadership, which may mean the team needs to adapt to a new management style and could revert to the storming stage.

COMPLEMENTARY READING

The Thomas, Paul and Cadle (2012) book recommended as complementary reading at the end of this chapter gives more detail on the Tuckman and Jenson model with further references.

8.10 TEAM DYNAMICS

Building a team involves finding people with the appropriate skills who are available when you need them and are motivated to perform the tasks required of them. However, if the team is to work well together, a satisfactory mix of personality types and personal attributes is essential. A system for analysing and categorising people's personal characteristics was developed by Meredith Belbin, who defined nine **team roles**:

- **shaper** – an energetic team member with a strong need for achievement who drives the team along;
- **plant** – a creative and innovative team member (the term ‘plant’ is used because it was found that planting such a person in an uninspiring team was a good way to improve its performance);
- **resource investigator** – a team member who makes contacts outside the group to bring in ideas and information and to acquire materials/resources;
- **co-ordinator** – a chairperson who promotes decision-making and delegates well (not necessarily the team leader);
- **monitor evaluator** – a team member who is analytical and able to assess ideas and options, but is not creative;
- **team worker** – a team member who helps to maintain team spirit and cohesion;
- **completer finisher** – a conscientious and painstaking team member who is concerned with getting things finished (this is a very important team trait);
- **implementer** – a team member who attends to details, is hard-working and organises the practical side of the project;
- **technical specialist** – someone who can provide the team with technical expertise.

COMPLEMENTARY READING

For further details, see Meredith Belbin, R. (2013) *Management Teams: Why they succeed or fail*, 3rd edn. Abingdon: Routledge.

It is not suggested that a team has to have one person of each type or that each team member falls into only one role. An individual can have attributes from a number of different roles. The idea is that a team needs a satisfactory mix of roles played by its members to perform well. A Meredith Belbin analysis may help to define a team weakness, or indeed to clarify the reasons why a team is not performing well or why conflicts keep arising in an otherwise skilled, competent team.

8.11 MANAGEMENT STYLES

There are as many styles of management as there are managers. Much depends on the personality and capability of the manager (or leader) and the prevailing circumstances. The manager may have a natural preference for a certain style, but to be successful must vary their style to suit the circumstances.

For example, a **task-orientated** leadership style is sometimes distinguished from a **relationship-orientated** leadership style. Task orientation focuses on the technical aspects of the work, while relationship orientation emphasises such things as individual motivation and team morale. When a team is developing – the forming stage – it will need more direction

than when it is fully functioning – the performing phase. A task-orientated approach would be more effective than a relationship-orientated approach because the team members are not yet familiar enough with the tasks to be carried out and the environment in which they are to be carried out.

Leadership often has two phases. The first is where events demand a response and the leader needs to decide upon an effective course of action. This is followed by the implementation of the chosen course of action. The general approaches to this have been analysed on two axes: **autocratic versus democratic** and **directive versus permissive**:

- **directive autocrat**: the leader makes decisions alone and closely supervises their implementation;
- **permissive autocrat**: the leader makes decisions alone, but subordinates have latitude in their implementation;
- **directive democrat**: the leader makes decisions participatively, but implementation is closely supervised;
- **permissive democrat**: the leader makes decisions participatively, and subordinates have latitude in their implementation.

Individual team members will react differently to these approaches. Some prefer to have clear direction and to leave decision-making to others, whereas other team members like to play a part in establishing the direction of the project.

Autocratic/directive management provides clear direction and quick decisions. The leader is seen to be decisive, firm and effective. It may well be that the leader genuinely has technical knowledge and expertise that exceeds that of their subordinates. However, it can be synonymous with an uncaring, remote, unapproachable, controlling or bullying management style, which can demotivate a team.

Democratic/permissive management shares responsibility for decision-making and for the team's performance; thus, the team is likely to be more committed. However, this style can be perceived as weak and management can be seen as avoiding responsibility for difficult decisions. It may be difficult to enforce discipline if conflicts develop.

The ideal is a **situational management style**, which adapts to the demands of a situation. In some situations the task to be done is complex and needs to coordinate the efforts of different specialists. It could also be time-constrained – for example, when a critical operational system is being modified. In this case, the manager/leader would need to seek the advice of different specialists and consult on how they were going to work together. This would be 'democratic'. The actual implementation would need tight discipline and would be 'directive'.

In another example, there could be a central demand for compliance with a statutory requirement related, for example, to data security. The way the requirement is implemented might vary between the different systems in an organisation

because of variations in their interfaces. Here, the autocratic demand for compliance could be tempered by a permissive approach about how the requirements are met.

Whichever approach is used, the project manager must be an accomplished communicator and have the ability to use the most appropriate **methods of communication** in order to be fully effective.

8.12 COMMUNICATION METHODS

A project can be seen as a network of stakeholders or participants (including managers, users and developers) who each have particular needs to communicate with other stakeholders.

On the one hand, effective communication is needed so that, for example, co-workers are made aware of changes to project plans, requirements and designs that are the basis for their work. On the other hand, there is a risk of **information overload** when project participants are overwhelmed by information, much of which is not relevant.

Consideration must therefore be given to the information needs of project participants as well as the best methods to satisfy the needs. This leads to a **communication plan**.

ACTIVITY 8.3

It does not take long for the Water Holiday Company to realise that the integration of Canal Dreams and Minotours onto a single green-field site location and the transfer to a single booking system is going to take considerable time to implement. In the short term it is decided to keep the back-end booking functions of the two subsidiary companies separate and to retain the existing staff for at least two years. A single Water Holiday Company website is to be built with a common front end. The front end will conform visually to a style guide produced for the Water Holiday Company by a marketing consultancy. As far as booking is concerned, this should look as similar as possible for the two types of holiday, but may diverge to deal with differences with the products/services provided by Canal Dreams and Minotours. The new interface is to be designed and implemented by an external supplier, XYZ Systems. The work is to be treated as a project within the over-arching Water Holiday Company integration programme.

1. What information will XYZ need from the Water Holiday Company and its Canal Dreams and Minotours subsidiaries?
2. Where will the information come from and how will it be acquired?
3. What information would XYZ need to supply to the Water Holiday Company during the project?

Methods can be categorised as **active** or **passive** and as **formal** or **informal**. **Active** methods, such as a telephone conversation, require a response or reaction so that there is reinforcement or confirmation that the information or message has been received and understood. **Passive** methods such as a newsletter require no such confirmation. They leave to chance whether anyone reads or understands the material and should not be used for important messages.

Formal methods of communication have a set structure, such as a meeting of a project board, in contrast to **informal** methods such as a conversation, which carries no particular format and is not usually recorded.

It is also possible to categorise methods depending on whether the participants have to be available at the same time and/or in the same place for communication to take place (see Table 8.2).

By categorising methods in this way, it is possible to assess the suitability of possible methods of meeting a communication need. For example, you may decide that a short, weekly face-to-face meeting (an active, formal method that is same time/place) with the project sponsor would be better than an email (different time/place). This should be documented in the communication plan.

Table 8.2 Examples of same/different time/place communication

	Same place	Different place
Same time	Face-to-face meetings	Telephone Video conferences Chat
Different time	Bulletin boards Pigeon-holes	Documents Emails

By setting out a detailed communications plan with all stakeholders, rather than leaving things to chance, you stand a much better chance of getting it right. Table 8.3 is an example of an entry for one of the types of communication selected for a project.

Table 8.3 Example of entry in a communication plan

Name/description	Joint application development session
Target audience	User representatives and business analysts
Purpose	To elicit user requirements
Frequency/event	23rd April
Method of communication	Away day

Responsibility Ann Smith (Project Manager)

ACTIVITY 8.4

A critical success factor of the Water Holiday Company's integration is the customer experience at the boatyards and marinas where customers pick up and drop off hired boats. These establishments are also important to the booking system, as it needs accurate, up-to-date information on boat availability taking account of periods of non-availability for maintenance and other reasons. Historically, the boatyards in the UK have tended to be owned by Canal Dreams (the result of a policy of gradually extending its reach over the UK canal network through a programme of acquiring local boatyards); whereas the marinas in Greece, that service the needs of Minotours, tend to be locally owned.

Given the scenario described in Activity 8.3, outline the main communication events with boatyards that might be planned, giving the purpose and proposed methods of communication.

IT-based tools can support communication. Where work groups are small and collocated, simple, non-technological techniques such as moving around cards on a pin board can be an excellent way of showing the current state of the flow of work in the group. This approach can be digitised by using a

tool such as Trello – just one example among many – which can extend its reach to workers at distant locations. In Chapter 3 the use of Gantt charts to monitor projects was discussed; while these can provide an accurate overall picture of the project as a whole, they are unwieldy as a way of reflecting the immediate status of work for individual developers.

8.13 CONCLUSION

This chapter has only been able to touch briefly upon some aspects of organisational behaviour as it relates to projects. A good book on the important issues of organisational behaviour that goes beyond the immediate demands of the BCS Foundation Certificate in Project Management for Information Systems is recommended in the complementary reading.

COMPLEMENTARY READING

Thomas, P., Paul, D. and Cadle, J. (2012) *The Human Touch: Personal skills for professional success*. Swindon: BCS.

SAMPLE QUESTIONS

1. **Which of the following is NOT a name given to the group that has responsibility for committing resources**

to the project and approving variations to the project's objectives?

a. project board

b. project management board

c. project support office

d. steering committee

2. **Which of the following terms is used to describe an organisational structure where staff are responsible to a project manager for the duration of a project, but also have a manager who is responsible for their long-term staff development and work programme?**

a. the Tuckman–Jensen model

b. configuration management

c. matrix management

d. task orientation/relationship orientation

3. **At which stage of team formation does a team become fully functional as a cohesive group?**

a. performing

b. norming

c. storming

d. forming

4. **Which of the following is an example of different time/different place communication?**

   a. a project board meeting

   b. a progress report document

   c. a telephone conversation discussing a problem a user has with an IT system

   d. video conferencing with the supplier of a software application

POINTERS FOR ACTIVITIES

ACTIVITY 8.1

Stakeholders affected by the Water Holiday Company integration project include the two sets of booking staff at Canal Dreams and Minotours, customer-facing staff at boatyards and marinas, the owners of the new business entity, existing customers, previous and current management including the managing director, marketing manager, central finance, human resources department (who will have to carry out the delicate tasks relating to staff redundancy and relocation), premises management, building construction and fitting out specialists, and suppliers of IT and software products and services.

Without doubt many other stakeholders can be found!

ACTIVITY 8.2

Your answer will of course depend on your own experience. My own experience of the feedback from IT students returning from a year's IT work placement was that by far the most frequent management failing was lack of communication between managers and staff. Often this could be put down to the excessive workloads put on managers.

Communication tends to be important because IT projects require inputs from many different resources and this creates many opportunities for misunderstanding.

As far as good qualities were concerned, this was more nebulous, but often the rather vague quality of 'friendliness' was used.

This was very much a 'worm's eye' view of project management.

ACTIVITY 8.3

1. To answer this question properly you would need to identify the system life cycle that XYZ is going to use and then look at the external inputs for each phase. Let's assume a design/build/test cycle:
    - **Design:** The inputs to this will include system requirements, which include both business requirements and technical requirements about the existing back-end systems components that the front end will interface with. It will also need to know whether the system they have designed will be acceptable to the clients.

- **Build:** This will use the designs produced above. The designs may be elaborated as they are built, and so further reassurance about their acceptability is needed.
- **Test:** If good interface design practice has been followed, then some informal testing with prospective users will have been done at the Build stage. This will be a formal acceptance test using, as far as possible, real transactions.

2. Information sources and gathering methods:
    - Some requirements will have been defined in the supplier selection process as the basis for the contract for work.
    - The style guide exists as a document.
    - Representative users from Canal Dreams and Minotours could be interviewed to clarify lower-level requirements, particularly differences in the processes at the two subsidiaries.
    - Requirements at operational level ought to be signed off at managerial level to ensure they are compatible with longer term Water Holiday Company aspirations.
    - Details of the interface may exist in documentation, but there is a risk that this is not up-to-date. Access to actual code is desirable.
    - As transactions are built, demonstrations to booking specialists can obtain information about their acceptability. Note that there is the need to ensure

alignment between the requirements of Water Holiday Company operational staff and Water Holiday Company management.

- Integration testing will provide information about compatibility with existing back-ends.

3. Information needed by the Water Holiday Company:

   The information needed by the Water Holiday Company will mainly be concerned with the completion date. If the agreement with XYZ is fixed price, then costs will not be an issue. However, the basis for payment could be ‘time and materials’, which means that XYZ staff working for the Water Holiday Company keep a record of hours worked and XYZ then submit an invoice for payment for those hours. In this case, the Water Holiday Company will want details functionality completed and the type of work the developers have been doing.

XYZ will be anxious to record any changes to previously agreed requirements in order to ensure the additional costs are recouped. They would also want issues where Water Holiday Company staff have delayed progress to be recorded and resolved.

ACTIVITY 8.4

	Event	Purpose	Method
1.	Informal notification of merger plans to yards and marinas	To build up trust and confidence in local owners and employees	Personal visits
		To gather information about possible business risks	
2.	Official announcement of merger to general public	Start of process of ensuring existing customers are aware of continuity between old and new entities	Press release, which is sent to yards and marinas beforehand
			Provision of hot-line to deal with queries from staff and public
3.	User workshops for Canal Dreams and Minotours representative staff	To gather requirements of local yard/marina staff for common interface	One could have individual interviews of yards/ marina staff, but this could be time-consuming, and a joint workshop can help build a consensus
4.	Publication to staff of draft interface requirement with request for feedback	Quality check on completeness and appropriateness	Circulation of document with request for emails for feedback
5.	Launch of brand	To make public aware of new brand	Press release and media events
6.	Briefing on change over to new system	Make staff aware of main features of new interface	Briefings and training on regional basis
7.	Switch to new interface	Transition to new methods of interfacing, with bookings and so on, to be made with minimum technical/ clerical errors	Incremental switchover region by region with technical and business support on hand
8.	Request for feedback on operation	To identify need for fine-tuning changes	Reports through local managers/owners

ANSWERS TO SAMPLE QUESTIONS

CHAPTER 1

1. (a)
2. (c)
3. (a)
4. (a)

CHAPTER 2

1. (a)
2. (c)
3. (d)
4. (c)

CHAPTER 3

1. (b)
2. (c)

3. (d)
4. (a)

CHAPTER 4

1. (a)
2. (c)
3. (c)

CHAPTER 5

1. (a)
2. (c)
3. (b)
4. (c)

CHAPTER 6

1. (b)
2. (d)
3. (d)
4. (a)

CHAPTER 7

1. (a)
2. (c)

3. (d)
4. (a)

CHAPTER 8

1. (c)
2. (c)
3. (a)
4. (b)

PROJECT MANAGEMENT FOR IT-RELATED PROJECTS

Third edition

Edited by Bob Hughes

Ideal for IT practitioners with project management responsibilities, this book explains the principles of IT-related project management, including project planning, monitoring and control, change management, risk management, and communication between project stakeholders.

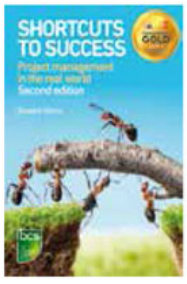
In line with the BCS Foundation Certificate in IS Project Management, each chapter includes an overview of learning objectives, detailed discussion of the syllabus content, activities, and multiple-choice questions for self-assessment. This new edition introduces the latest project management thinking, terminology and standards.

- **The only official textbook of the BCS Foundation Certificate in IS Project Management**
- **Learn how to plan, monitor, control and organise IT projects**
- **Explore change control and configuration management**
- **Understand the importance of building in product quality through the project management process**
- **Discover how project risks can be identified, approached and mitigated**

ABOUT THE AUTHORS

The five authors who developed this book are all either BCS examiners or course providers and include three Chief Moderators for the BCS Certificate in IS Project Management. They all have extensive experience of practitioner education and of practical project management.

You might also be interested in:



A comprehensive and modern guide to managing projects in an IT environment... Recommended.

Elizabeth Harrin FAPM, Director, Otobos Consultants Ltd and author

This is a project manager's 'must-have' book, and a great testimony to the author team's hard work in pulling together a wealth of practical advice for aspiring and current project managers. 10/10!

George Williams MBCS CITP, Management Consultant
Review of previous edition

The structured approach, with clearly laid out learning objectives, will appeal to trainers and academics as well as practitioners...A valuable update of a popular standard text.

Miles Shepherd, Vice President, Association for Project Management (APM)
Review of previous edition

Business, Information Technology

ISBN 978-1-78017-484-6

bcs The Chartered Institute for IT

Paperback available

BIBLIOGRAPHY

Belbin, R. M. (2013) *Management Teams: Why they succeed or fail*, 3rd edn. Abingdon: Routledge.

Blackstaff, M. (2012) *Finance for IT Decision Makers: A practical handbook*, 3rd edn. Swindon: BCS.

Holt, J. and Newton, J. (eds) (2020) *A Practical Guide to IT Law*, 3rd edn. Swindon: BCS.

Murray, A. (2016) *Information Technology Law: The law and society*, 3rd edn. Oxford: Oxford University Press.

Paul, D., Cadle, J., Eva, M., Rollason, C. and Hunsley, J. (2020) *Business Analysis*, 4th edn. Swindon: BCS.

Tate, M. (2015) *Off-The-Shelf IT Solutions: A practitioner's guide to selection and procurement*. Swindon: BCS.

Thomas, P., Paul, D. and Cadle, J. (2012) *The Human Touch: Personal skills for professional success*. Swindon: BCS.

INDEX