PROJECT-MANAGEMENT FOR IT-RELATED PROJECTS

Third edition

Bob Hughes (editor)

bcs
The Chartered Institute for IT

PROJECT-MANAGEMENT FOR IT-RELATED PROJECTS

Third edition

Bob Hughes (editor)

bcs
The Chartered Institute for IT

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 members and volunteer communities around the world.

Setting standards and frameworks

The Institute collaborates with government, industry and relevant bodies to establish good working practices, codes of conduct, skills frameworks and common standards. It also offers a range of consultancy services to employers to help them adopt best practice.

Become a member

Over 70,000 people including students, teachers, professionals and practitioners enjoy the benefits of BCS membership. These include access to an international community, invitations to roster of local and national events, career development tools and a quarterly

thought-leadership magazine. Visit www.bcs.org/membership to find out more.

Further Information

BCS, The Chartered Institute for IT,
First Floor, Block D,
North Star House, North Star Avenue,
Swindon, SN2 1FA, United Kingdom.
T +44 (0) 1793 417 424
F +44 (0) 1793 417 444
(Monday to Friday, 09:00 to 17:00 UK time)
www.bcs.org/contact
http://shop.bcs.org/

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

Published by BCS Learning and Development Ltd, a wholly owned subsidiary of BCS, The Chartered Institute for IT, First Floor, Block D, North Star House, North Star Avenue, Swindon, SN2 1FA, UK.
www.bcs.org

Paperback ISBN 9781780174846
PDF ISBN 9781780174853
ePUB ISBN 9781780174860
Kindle ISBN 9781780174877

British Cataloguing in Publication Data.
A CIP catalogue record for this book is available at the British Library.

information contained within it and neither the authors nor BCS Learning and Development Ltd shall be responsible or liable for any loss or damage whatsoever arising by virtue of such information or any instructions or advice contained within this publication or by any of the aforementioned.

Publisher's acknowledgements
Reviewers: David Reynolds, Brian Wernham
Publisher: Ian Borthwick
Commissioning editor: Rebecca Youé
Production manager: Florence Leroy
Project manager: Sunrise Setting Ltd
Copy-editor: Mary Hobbins
Proofreader: Sarah Cook
Indexer: John Silvester
Cover design: Alex Wright
Cover image: Baloncici
Typeset by Lapiz Digital Services, Chennai, India

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 in IS Project Management as a whole involved many BCS staff, including Malcolm Sillars, Rebecca Stoddart, Imelda Byrne, Steve Causer and Carol Lewis.

Jutta Mackwell was instrumental in initiating work on a second edition and Sharon Nickels managed the production of this version from the editor’s word-processed manuscript to the actual published text.

My thanks go to Becky Youé, who floated the idea of a third edition, and to Florence Leroy. Many improvements were due to suggestions by Noel Younger, Phil Baker and Mike Heselton, who used previous editions of the book in their work with the BCS Higher Education Qualification Diploma in IT Project Management. Helpful suggestions were also made by the two reviewers and by editor Mary Hobbins.

My exposure to the processes behind the latest update to the British Standard 6079 Guidance on Project Management, as the BCS representative on the BSI development group, have made me much more aware of the integration of IT project management with more generic project principles. I would therefore acknowledge my gratitude to the other group members for many new insights. The BCS Project Management Specialist Group, under the leadership of David Reynolds, with which I was for a time associated, has also been a source of new insights into the practice of project management. Charles Symons also provided timely information not only on functional software measurement but on issues relating to project success and failure in general.

This edition is dedicated, as were the previous ones, to the memory of Jimmy Robertson.

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 a new or modified system that creates benefits for an organisation, or, in the case of public sector projects, for society. Many IT projects implement new information technology (IT) applications within organisations. These are technical, but also involve changing the organisation in some way.

As well as business change projects, some IT-related projects generate new products, as in the case of computer games. These too would be expected to generate benefits for an organisation; in this case, new sales and increased income.

Difficult projects often involve innovation: a product different from anything before might be created; or new methods of delivery might be used. This differs from **business as usual** (**BAU**), where you are doing a routine job. A complication for IT specialists is that you often do a combination of BAU and project work.

A project comes about when one or more people have an idea about a desirable product or change. For this to become a project, a **business case** is needed showing that the value of the benefits of completing the project will be greater than the costs of implementing and operating the new (or revised) system that the project would create. This will not only need to consider business concerns, but also the technical difficulties of the project. This is underlined by the alternative name of **feasibility study** for the business case.

The project that emerges from a business case should have the following:

- A defined start point, which is when
  - the exploration of the idea moves into an organised implementation;
  - the idea obtains business backing and a project sponsor – an individual or group within an organisation who takes ownership of the project and ensures that it has the appropriate financial resources;
  - a commitment is made to provide the necessary resources;
  - responsibilities are defined;
  - initial plans are produced.
- A set of objectives, which
  - drive the actions of the organisation and project team towards a common goal;
  - should be stated and understood at the start of the project;
  - should be clear and unambiguous.
- A set of outputs or deliverables, which enable the objectives to be achieved.
- A date by which the objectives should be met and a budget setting the maximum allowable cost of the project.
- A unique purpose – routine activities are not projects.
- Benefits for the organisation which justify carrying out the project, and which are ideally measurable and greater than costs.

In some cases, the cost of the project is greater than the immediate benefits, but completion of the project enables other projects to be

implemented that will reap the benefits – this is often the case with IT infrastructure projects. In this case, the work may be organised as a **programme** comprising several projects which together achieve the programme's objectives.

1.2 SUCCESSFUL PROJECTS

To be successful, a project should:

- enable the stated objectives to be achieved;
- be delivered on time and within budget;
- deliver a system that performs to agreed specifications, including those relating to quality;
- satisfy the project sponsor and other interested parties. The term **stakeholder** refers to anyone who has an interest in the project; their role is discussed further in Section 8.3.

These are **project** objectives. The IT functions delivered, including both hardware and software, should enable the organisation to meet its **business** objectives. There will be other actions needed that do not involve IT development. For example, moving to web-based sales would require home delivery services to be negotiated. These business actions need to be part of the overall plan.

A new website supported by a new delivery service could enable an organisation to sell its products online to a wider market. However, while these project objectives might be achieved, the business objective of selling more products might be denied because of an external factor such as a general downturn in the market. The

products that the project delivers are distinguished from its **outcomes** – how it actually changes the outside world.

Objectives are often called **success criteria**. If they are satisfied, then the project can be deemed a success. There could be several ways by which the success criteria might be achieved.

Well-defined success criteria are SMART (**S**pecific, **M**easurable, **A**chievable, **R**esource-constrained and **T**ime-constrained). For example, '*increasing market share*' is too vague. '*Creating an online booking system that will be used by at least 30 per cent of customers in its first year*' is more specific and measurable. As well as being specific and measurable, success criteria must be clearly achievable. If it is clear that they are not, people are likely to ignore them. Finally, part of the statement of objectives for a project will always define a deadline and an overall target cost.

In summary, the project must have a **specified cost**, a **specified time/duration** and meet a **specified business requirement**. These three specifications are closely linked. Any change to one will affect the others. The project objectives that relate to cost, time and the degree to which functional and quality requirements are satisfied ('**scope**') are often called the '**project triangle**'.

The project sponsor and users typically want a system with a broad scope – capable of a multitude of functions – to be delivered immediately and at low cost. Generally, this is not possible and the agreed project objectives will be a compromise between cost, time and scope.

If the scope of requirements strays outside this area of compromise, it will increase cost and/or delivery time. The costs of the project

could therefore exceed the value of the benefits. Say the deadline for project completion has to be brought forward; to compensate, the scope could be reduced. This might well reduce the benefits of the project. Alternatively, more staff could be employed to work in parallel on the project, which would increase costs – and some project risks.

While this project book-keeping is important, the perceived value of the benefits of a project may in fact be quite subjective. The final judges of the success or failure of a project will be the project sponsor and the users of the products of the project. Sensitivity to their needs will be as important as the letter of any contract.

THE WATER HOLIDAY COMPANY INTEGRATION SCENARIO

The Water Holiday Company is a new business entity formed by the merger of two companies. One, Canal Dreams, had specialised in hiring out canal boats to holiday-makers in the UK. The other, Minotours, had chartered out yachts in the Aegean and Mediterranean seas as part of package holidays.

A marketing decision has been made to merge the websites for the two businesses, and the opportunity will be taken to extend access to online systems for making bookings and payments to smartphones. Whenever a sales transaction is made, there is a legal requirement to send the customer a follow-up email as confirmation. This email will be made into a vehicle for cross-selling, for example local car-hire.

A long-term goal is to centralise all Canal Dreams and Minotours operations (apart from boatyards and marinas) at a new, single green-field site. The objectives of these changes

are: (i) to reduce staff by removing duplicated business functions across the two formerly separated companies; and (ii) to exploit the opportunities for entering new boat-hire-related markets, such as river cruising in continental Europe.

In order to meet these business objectives, the proposed system will need certain functionalities. For instance, it should allow the potential customer to check the availability of a particular kind of a boat moored at a particular locality at a time convenient to the prospective hirer. These **functional requirements** specify what the system will do and include **quality requirements**. For example, if it takes too long for the system to respond to a query, your prospective hirer may swap to a competing website.

Requirements will include not just those of the organisation but also those imposed by external bodies, such as **legal requirements** relating to distance selling. This is one example of the growing importance of organisations ensuring **compliance** with the regulatory authorities.

COMPLEMENTARY READING

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

For a more detailed source on legal issues, see Murray, A. (2016) *Information Technology Law: The law and society*, 3rd

edn. Oxford: Oxford University Press.

Each functional requirement carries a cost. There is also an overall project **cost requirement**. The potential additional income through extra bookings and staff savings in the Water Holiday Company example on the previous page might not be enough to meet the cost of implementing the system.

Boating holidays are a seasonal business and so implementation of system integration will need to be at a quiet time of the year before the bookings for the next season start to come in. This implies a certain **deadline** for system implementation.

1.3 PROJECT MANAGEMENT

Having established and agreed objectives, we can now move to planning. Having produced good plans, monitoring and effective control of the project will then be needed to fulfil the plans and achieve the agreed objectives. The **project manager** is someone with experience of overseeing implementation work and takes responsibility for controlling the work in accordance with the plans. There should also be someone who is the **project sponsor** and understands the business environment in which the new system is to be installed. The project manager interacts with the project sponsor to ensure business needs are met.

A successful project cannot be guaranteed, but the following will help to enable success:

- **Clearly defined roles and responsibilities:** these must be clearly defined, documented and agreed.

- **Clear objectives and scope:** managers who embark on projects without clearly establishing the scope of the expected deliverables, together with cost, time and quality objectives, are creating problems for the future. At the start of the project, there should be **terms of reference**, a **project charter** or some other document that defines the project objectives. More detailed requirements would be developed within this scope.
- **Control:** it is important to establish at the outset how best to control the work and how to exercise that control. This will be specified in a **project management plan**.
- **Change procedures:** ideally, the project manager would work in a world where there is no change or uncertainty. Unfortunately, it has to be recognised that there is uncertainty and that change will happen. Appropriate change procedures are needed to deal with it. Chapter 4 describes these.
- **Reporting and communication:** clear reporting of project progress and any problems allows remedial action to be taken quickly. Effective communication with all stakeholders can help avoid conflicts.

A **project management method** is a set of processes used to run a project in a controlled and predictable fashion. The design and development procedures by which the objectives of the project are satisfied – for example, the use of object-orientated analysis – constitute the **development methods**.

There are various project management methods. A well-known one in the UK is PRINCE2. In general, project management methods are applicable to a broad range of project types, whereas development methods are specific to projects with particular types of deliverables.

This is because different deliverables require different development methods. Organising an office move, developing a software application and providing disaster recovery facilities are all projects needing their own methods of implementation, but all are controllable using the same project management processes. Different parts of a project often need different development methods, but the project management method will be common across it.

ACTIVITY 1.1

Assume that you are the manager of an office department that is going to be relocated to a building five miles away. Day-to-day management of the move will be delegated to one of your staff. What would be the main sequence of activities needed to plan and carry out the move? You want to leave as much of the detailed work as possible to your subordinates, but at which key points would you need to be involved to check progress?

Solution pointers for the activity can be found at the end of the chapter.

1.4 SYSTEM DEVELOPMENT LIFE CYCLE

Dividing a development method into a sequence of processes is a widely accepted practice. This allows systems to be designed and implemented using a methodical and logical approach. The number and names of these processes will vary from organisation to organisation. In some cases, stages will be combined or split.

With IT projects, there tends to be a divide between those that create new products and those that adapt existing applications and

components. Generally speaking, the processes shown in Table 1.1 belong to the **system development life cycles** (SDLC) that apply to both types of IT project.

This suggests a particular sequence of processes. However, an application under development could be delivered in incremental segments (see Section 1.7.3). Thus, one software component could be coded while another was still being specified. The key point is that all these technical processes have to be dealt with somewhere within the project.

Each process creates one or more tangible products or **deliverables**. Delivery of the products of each process can act as a **milestone** at which we can judge the progress and continuing viability of the project.

You could well work in an environment where the concern is with implementing business change. Here the focus tends to be on identifying and adapting existing applications provided by vendors. Because the components already exist, the design and construction processes are not carried out. Instead, a selection process is devised consisting of methods of evaluating the suitability of candidate products. These could be services that are delivered via the cloud. The products to be used are then selected. Some element of customisation is usually needed to modify the product for local use. An acceptance test will almost certainly be carried out.

Table 1.1 Activities in SDLCs for Build versus Adapt projects

Build	Adapt
Initiation	

Identification of the business case	
Project set-up	
Requirements elicitation and analysis	As for build, plus definition of selection criteria
Design	Identification of potential applications, services and suppliers
Construction	Evaluation and selection
	Customisation
Acceptance testing	
Transition to operation	
Review and maintenance	

It can be seen above that ‘build versus buy’ questions will have a major impact on the design of the system development life cycle for the project. This question is part of a broader set of questions that need to be answered about how exactly the desired outcomes of the project are to be met in practical terms. These practical activities and products can be referred to collectively as the ‘solution’. A major concern at this stage is establishing that the preferred solution will in practice lead to the business objectives being met.

Build versus buy

A basic software engineering principle is to avoid developing new functionality where existing tried and tested code already exists. The advantages of buying functionality are:

- It already exists, so it can be installed more quickly.
- It can be seen in action, so users can get a good idea of its quality.
- Existing users will have effectively tested it by reporting any defects, which will then have been removed by the supplier, so the functionality is likely to be more reliable.
- As there are many different organisations using the functionality, development costs will be shared and the cost of the application should be cheaper than if you had built it yourself.
- You do not have to employ software developers to build the new system, who may then become surplus to requirements.
- The vendor should supply updates to deal with statutory changes, so maintenance of the installed system will not be a responsibility of the host organisation.

However, there are disadvantages in buying existing applications, which may encourage the building of new software:

- Off-the-shelf software may not meet all the particular requirements of the host organisation. It could even have features you do not need, and this could confuse users.
- The organisation may have to change its business processes to fit in with the way in which the off-the-shelf application works.

- If you adopt an off-the-shelf application, you can be as good as your competitors who have the same system, but you cannot be better than them.
- Once you have adopted a particular off-the-shelf package, it may be difficult to change. Off-the-shelf software is often leased on an annual licence and if the vendor increases the licence fee, you may be trapped into having to pay it.
- If the vendor ceases to trade, this may put you in a difficult situation if you are dependent on vendor support for such things as training and updates to the system.

We will now look at the typical stages in an IT-related project life cycle.

1.4.1 Initiation

The first two processes – initiation and the identification of the business case – are, strictly speaking, not part of the project. Their purpose is to establish the justification for the project: an outcome could be a decision not to go ahead with the project.

The objective of project initiation is to decide the most appropriate way to respond to a request for some work to be done, while taking into account any business or technical strategies that the host organisation might have.

An organisation's managers see a need that can only be satisfied by some form of project. The need might be a problem to be solved, something to add to an existing system or some new way of delivering value to the organisation. The initiation process checks

that a problem or opportunity really exists and the change is desirable, and whether a project is the best way to implement the change. The result is a decision by the project sponsor on whether resources should be spent on further investigation of the feasibility of the proposal, including the business case for it. **Terms of reference** should be drawn up, outlining the scope of the proposal to be investigated and authorising staff to carry out the investigation. Staff carrying out the investigation need to have permission to gather information from those working in the areas affected, along with other stakeholders.

There are situations where an organisation's business objectives are best achieved through a **programme**; that is, a number of different projects, each dealing with a different element of the project. In this case most of the initiation tasks for a component project will have already been done when the programme was initiated.

1.4.2 Identification of the business case

The business case, or feasibility study, assesses whether the proposed development is viable in terms of the balance of costs and benefits, the technical requirements and the organisation's business objectives. The deliverable is a feasibility report which, among other things, presents the business with a range of options aimed at providing a solution.

Apart from the question of business viability – the benefits being greater than the costs – factors influencing the decision to adopt a particular option include:

- **Budget constraints:** the benefits may be greater than the estimated costs, but does the organisation have the resources to

pay for the investment? If it does have the resources, is this the best project to spend them on?

- **Technical constraints:** can the proposed project be completed with the technology currently available? This will require the setting out of a **technical strategy** or '**solution**' that provides practical steps for the achievement of the project's objectives.
- **Time constraints:** can the proposed project be completed in the available time?
- **Organisational constraints:** can the organisation cope with the changes that the new development will demand?

One or more of these constraints may prevent a project from being developed any further. The content of the business case is discussed in more detail in Section 1.9.

In some cases, the business objectives can only be met by carrying out more than one project. For example, a new data centre may require a construction project for a new building, which is treated as a separate project from the installation of IT equipment. As noted earlier, these projects may be grouped into a programme.

1.4.3 Project set-up

Based on the recommendation of the feasibility or **business case report**, the organisation will decide whether to go ahead with the full project. At this point a group variously called the **steering committee**, **project board** or **project management board** is set up to oversee the project in the organisation's interests. A project manager needs to be appointed and an initial project team set up to start work.

More detailed planning for the project takes place. Important decisions will be taken about how the declared project objectives are to be fulfilled. New **terms of reference** for the project to implement the new system, as opposed to simply investigating its feasibility, are drawn up.

1.4.4 Requirements elicitation and analysis

This phase defines the requirements of the new system in detail and identifies each business transaction. Some work on identifying requirements will already have been done when the business case was being identified. The elicitation or gathering of requirements could involve:

- interviewing users and their managers;
- examining documentation describing the current operations;
- analysing operational records created by the current system;
- observation of work practices;
- workshops – where groups of stakeholders and business analysts meet in intensive (often day-long) sessions, sometimes chaired by an independent **facilitator**, to identify and agree detailed requirements;
- questionnaire surveys.

In some cases, mock-ups or **prototypes** of parts of the new system could be used to help the users clarify their ideas about the requirements.

ACTIVITY 1.2

What kinds of people should the business analyst interview in the Water Holiday Company integration project in order to obtain the requirements for the new system?

Business analysis techniques, such as business process modelling or data analysis, will usually be applied to organise the raw data collected.

COMPLEMENTARY READING

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

At the end of this phase, a set of **requirements** is produced. This describes what the final system should be able to accomplish and lists all the major features of the end product. It forms the basis of the agreement between the customer for the new system and the developers.

We noted earlier in Section 1.2, that requirements include functional requirements (what the system is to do) and quality requirements (how well the system is to perform). Requirements can be prioritised, particularly where there is an overall cost requirement. There will also be trade-offs between quality requirements, where for example, security requirements are at the cost of ease of use.

At this point an outline of the test cases should be drafted, consisting of test transactions and the results expected from them. As will be

seen, these will be used to check that the delivered system conforms to the requirements statement when acceptance testing is done.

1.4.5 Design

If it is decided to build a new system, rather than adopting existing functionality, then a design phase is needed. This activity translates the requirements for the automated parts of the system into a design specification of the computer processes and data stores that will be needed. It will also be driven by – and might modify – the solution/ technical strategy produced as part of the feasibility study/business case.

The identification of the inputs, outputs, business rules and information that the system will process is known as **logical design**. The **physical design** is concerned with the actual appearance of the input and output screens and the printed reports produced by the implemented system. Different physical designs could satisfy the same underlying logical design. The term user experience design (UXD) is used as the physical design of the screens needs to take account of the whole user experience of using the system. For example, steps should be taken to avoid situations where users are required to input information that is not readily available to them. This forces them to stop completing a transaction while they search for details required and then resume input. Where inputting is done by employees, interface design becomes job design.

Physical design in this phase is essentially concerned with the system as it will appear to the outside world. Further **internal physical design** of the software and data structures will take place in the next phase.

Where an existing application provided by a supplier is to be used, its design is already established. In this case, the issue is finding the package whose features most closely match the business requirements. The process must be planned by which available packages are to be evaluated and those which represent good value selected. Evaluation may involve trying out demonstration versions of the software, site visits to existing users of the software, and careful study of the suppliers' documentation. In some cases, the existing software will need to be customised – that is, modified – to meet the organisation's particular needs.

ACTIVITY 1.3

List some of the screens and other possible inputs and outputs that would need to be designed in the new integrated Water Holiday Company online booking system.

1.4.6 Construction

This process has the objective of designing, coding and testing software, and ensuring effective integration between different software components. For example, the new Water Holiday Company integrated booking system would need the development of a common application to record holiday bookings made by customers online. In the case of the Water Holiday Company integration, it is probable that one of the online booking systems used by one of the two companies to be merged could be used as the basis for the new integrated system. It is therefore likely that as well as new code being developed, existing code will be amended to deal with the enhancement.

Procedure manuals will also be produced and new hardware may have to be acquired. In the case of the Water Holiday Company, additional servers will be needed to deal with the increase in internet transactions. During this phase, the requirements statement will be re-examined to ensure that it is being followed to the letter. Normally any deviations have to be approved through a formal change procedure (see Chapter 4).

1.4.7 Using an external application

Sections 1.4.5 and 1.4.6 assume that a new integrated system is to be developed that may use some existing functionality. As noted at the beginning of Section 1.4, an alternative would be to look at the market to see if an existing application is available. After all, there are many businesses in the travel and tourism sector that handle bookings on a daily basis. If this option is followed, then among the activities that need planning would be:

- identification of potential applications and their suppliers;
- planning of how the evaluation of the offerings is to be carried out;
- the selection of the most suitable application;
- customisation, that is the setting of system and other parameters in the application to make it compatible with your method of working.

1.4.8 Acceptance testing

In Section 1.4.4, we suggested that test cases should be outlined at the requirements analysis stage that can be used to ensure that the requirements have been met. These can now be used to check the

delivered system. This testing could be carried out by knowledgeable representatives of the users and IT support staff before its implementation as an operational system. It is inevitable during this stage that the user will uncover problems of which developers were unaware.

In the case of the Water Holiday Company integration, a problem with the introduction of smartphone access to bookings and payments is that users of the new facilities will be members of the public who the company does not currently know. Usability tests using external members of the public recruited for this purpose will be needed to test the customer-facing interfaces.

These acceptance-testing activities may overlap with the actual transition to the operational system. For example, some tests will need to be carried out on the system when it is actually installed on the equipment to be used in the final installation.

1.4.9 Implementation, installation and transition

Here the project reaches fruition. Hardware that has been purchased is delivered and installed. Software is installed, users trained and the initial content of databases set up. Apart from the technical aspects of the transition, such as the creation of a merged database, there is a general need to ready the organisation to take full advantage of the benefits of the new system, For example, in the case of the Water Holiday Company integration project, the general public need to be made aware of the new entity created by merging Canal Dreams and Minotours. There are various strategies for implementation; these are discussed later in Section 1.10.

1.4.10 Project closure

It is possible that a project could be abandoned during its development because its business case is no longer valid.

In the case of successful project completion, certain tasks will need to be done on closure, including:

- sign-off of acceptance documents by the project sponsor – this may be conditional on other key stakeholders giving approval first;
- handing over responsibility for maintenance and support to a permanent team;
- closing down accounts relating to the project;
- the project manager writing a **lessons learnt report**;
- releasing and re-allocating project resources, including the project team and the project manager;
- arranging publicity to tell the outside world about the project's success.

1.4.11 Review and maintenance

At an agreed interval after the system has been made operational, a **post-implementation review** is carried out by a business analyst not involved in the original project. The review checks that the operational system is delivering the benefits envisaged in the original feasibility report. Changes sometimes result if the system does not fulfil its original requirements, but users could also identify new requirements. There could also be alterations in government regulations or in the way the organisation does business. These might become maintenance work, or be projects in their own right.

In the section on benefits management (Section 1.9) we will see that although the role of the project manager stops with the successful creation of the new system, the project sponsor who represents the business interests of the organisation will continue to have responsibility for ensuring the new system delivers benefits.

1.5 PROJECT MANAGEMENT AND THE DEVELOPMENT LIFE CYCLE

There is a need for management reviews at which the progress and direction of the project can be formally assessed.

Most projects contain elements of uncertainty, which make it difficult to meet all planned targets. This uncertainty is greatest at the beginning of the project, when little may be known about detailed requirements or any novel technologies, but gradually decreases as the project progresses. This makes it difficult in the early stages to plan later phases in detail. For example, it would be difficult at the outset for the Water Holiday Company to plan in detail the software code to be developed without a deeper understanding of the requirements of the new common booking system.

For the purposes of control there is a need to break the project into manageable units of work. These are variously called **stages** or **phases**. This is similar to dividing development into processes, except that it focuses on how best to manage the project. No two projects will be broken down identically because no single project structure can provide the best management control for all projects. In some cases – perhaps in smaller projects – two or more activities in the system development life cycle, such as design and construction, might be treated as one stage for management purposes. A larger

project might split a single development activity into several management stages. For example, the construction phase might be broken into different management stages concerned with the construction of different parts of the system. At the start of a project, an initial overall plan of the work to be done will be presented with a detailed plan for the first stage. Detailed plans for each subsequent stage will be prepared as the start of it approaches.

The end of each stage is marked by a formal review, involving the project sponsor, which assesses the work completed and outstanding, and whether the business case is still valid. The review concludes with a formal sign-off of the current stage and the project sponsor’s approval of the plan for the next stage of the project.

1.6 ELEMENTS OF PROJECT MANAGEMENT

As well as consolidating or splitting up development processes in consultation with the project sponsor in order to facilitate their control, the project manager tailors project management procedures to improve control over the project. Although the overall project management process remains the same, the amount of effort needed for control varies. Small projects, for example, need less formal control.

The processes that need to be tailored include:

- planning and estimating;
- monitoring and control;
- issue management;
- change control;

- risk management;
- project assurance;
- project organisation;
- business change and benefits management.

All of these will be described in greater detail in later sections of this book, but a brief overview will be useful at this point.

1.6.1 Planning and estimating

Good planning increases confidence within the project team. The aim is to detail the activities, the sequence in which they are carried out and the resources they need. The plan shows when activities are likely to start and finish and the estimated staff effort. An **outline** plan for the whole project is made initially, then a **detailed** plan is made for each stage nearer the start of the stage. Chapter 2 looks at planning and Chapter 6 looks at estimating in more detail.

1.6.2 Monitoring and control

Tracking and control ensures that the project meets its commitments in terms of deliverables, quality, time and cost by tracking the current state of the project against the plan and identifying any need to re-plan. Chapter 3 examines project control in more detail.

1.6.3 Issue management

During the course of a project, obstacles will be identified which could affect the project's success. These problems may be outside the direct control of the project manager and need **escalation**, that is, reference to higher authorities. An example is where a promised

external resource is not delivered when promised. Issues could lead to authorised changes to the project requirements – these have their own special procedures (see Section 1.6.4). The project manager should ensure that a system is in place for recording these issues, monitoring their status and starting any actions needed.

1.6.4 Change control

Any project can be subject to change. A change is often the result of a modification to requirements. Any requests for change should be made through a formal change management process. Failure to incorporate necessary changes reduces the benefit obtained from the project. However, accepting changes in an uncontrolled manner can cause problems that affect the cost, time scales and overall business case.

Chapter 4 examines change control and configuration management.

1.6.5 Risk management

All projects are subject to risk. If risks are not managed, they can have a detrimental effect. A suitable risk management process assesses and manages project risks.

Risks are different from issues. A risk is an unplanned occurrence that could happen, but has not yet done so. An issue is an unplanned occurrence which has already happened and which requires the project manager to request or initiate action.

Risk management identifies and quantifies risks before they happen, and plans and implements actions to eliminate risks or reduce their probability or impact. Risk management ensures that projects are

only undertaken with a full understanding of the potential implications of the risks involved. Chapter 7 deals with risk in more detail.

1.6.6 Project assurance

When pressure mounts on a project to meet its deadline, it is tempting to ignore some of the checks and balances imposed by project standards and to focus exclusively on the work to be done. This lack of control can be dangerous and can lead to project failure. Project assurance is a set of procedures which ensures correct project control is maintained. This involves auditing by staff outside the project team.

1.6.7 Project organisation

A key factor in any project is effective project organisation, where the roles and responsibilities of all participants are clearly defined and understood. Chapter 8 explicitly addresses this topic.

1.6.8 Business change and stakeholder engagement

Changes to the IT used in an organisation often change the way the organisation works. IT can change people's jobs. It can even cause staff to lose their jobs, and in other cases create new roles which need to be filled. The driver for IT change is to enable an organisation to generate new value and benefits.

Winning stakeholders' support for a project is important for project success. Unless time has been spent communicating with stakeholders and making sure that users know exactly what to expect of the new system, the project could meet its formal requirements but still be seen as failing to provide benefits by its users.

The starting point for stakeholder engagement is the creation of a communication plan which identifies the key information stakeholders need about the project and when and how that information is to be conveyed. Chapter 8 on Project Organisation touches upon this topic.

1.7 DEVELOPMENT PROCESS MODELS

Section 1.4 identified the need for a well-defined, repeatable and predictable system development life cycle. A project's life cycle is shaped by the products it delivers. Whatever the life cycle, it will be a set of activities each of which follows a particular method. For example, design and construction could adopt an Agile approach. Every activity in the process will have defined inputs and outputs contributing directly or indirectly to the deliverables to the customer.

Effective development methods are made up of an overall set of techniques and activities from which team members working on a new project of a particular type can select the most appropriate subset. The method should never require a task that does not produce something useful to the project.

The conventional system development life cycle for IT projects was described in Section 1.4. This has certain general characteristics that could apply to a number of different life cycles. However, it assumes that **technical processes** will vary according to the type of project. For example, as we have already seen, there are different project types (with differences in the types of processes carried out) where software is to be developed and where an existing software application is to be obtained from a vendor. In Section 1.5, it was noted that to make projects more manageable, the processes could

be split up or sequenced in different ways. Here we look at some of the options for this.

1.7.1 Waterfall model

This **waterfall** model is anathema to many software developers. It is the basic phased model of a development cycle (see Figure 1.1). It is also known as the **one-shot** or **once-through** approach. The model takes its name from the way each phase cascades into the next. It is assumed that activities are normally done in a strict sequence, although there is some scope for reworking stages once they have been completed. Projects should produce a sequence of deliverables, such as the requirements statement, design documents and software structures, where the output from each phase is an input to the next.

This approach provides for feedback loops activated when there is a need to revisit an earlier stage to redesign, recode and so forth. It is possible to return to any previous phase, although this could well require extensive re-planning. The ideal is for quality control activities to be associated with each phase, so that once the deliverables of a phase have been signed off they should not need to be reworked.

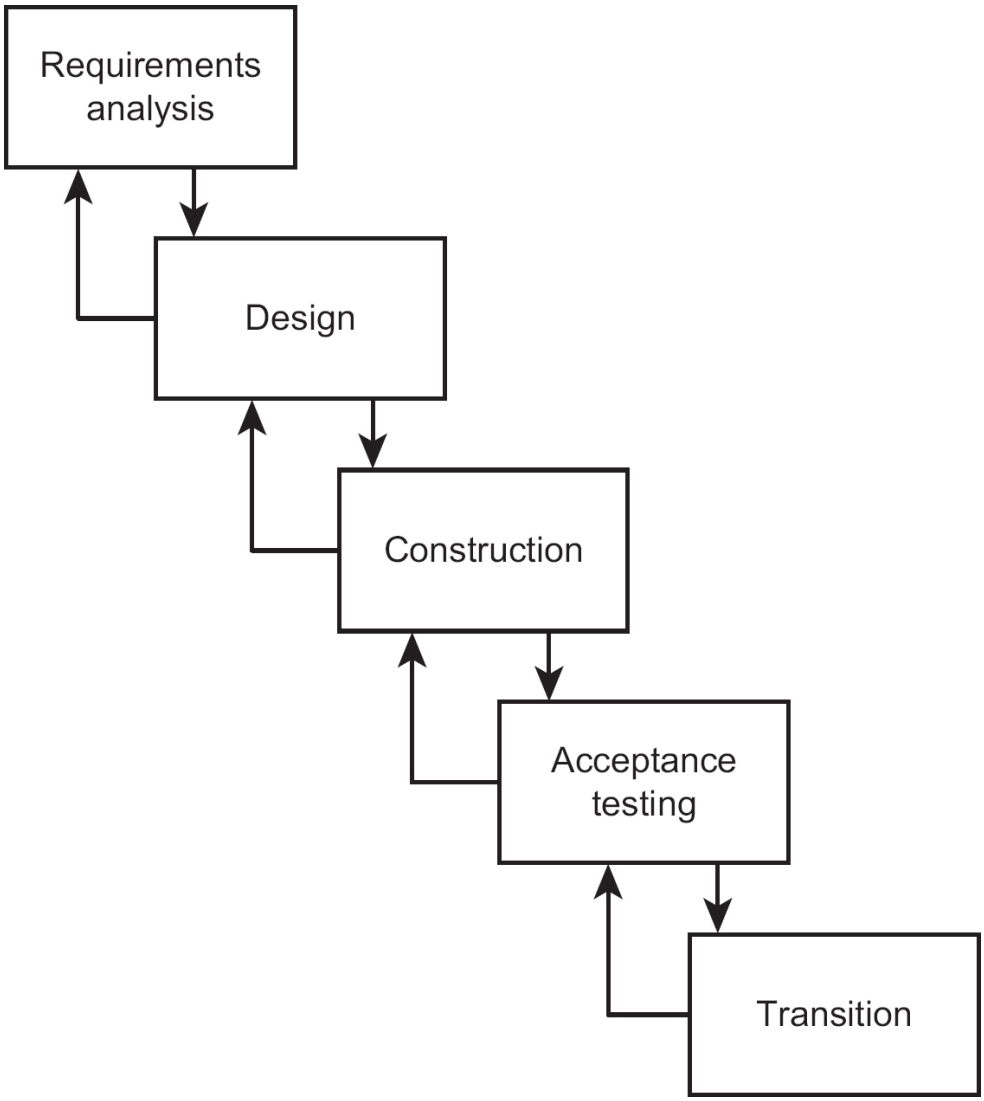
This model is probably best used on projects where requirements have been clearly defined and agreed. Critics of the approach argue that most projects do not have clear requirements at the beginning. As the model relies upon having each phase completed and signed off, it can become bureaucratic and time-consuming. For example, remedying a failing found during acceptance testing might require the participation of staff involved with the requirements, design and construction phases. It works best where there are few changes to

requirements during the development cycle, but whether changes will be needed will not be known before the project starts.

Another drawback is the amount of project documentation needed to pass information between stages and the effort needed to keep this up-to-date.

Despite these drawbacks, the approach provides convenient points when project sponsors can assess whether the project is still a valid investment.

Figure 1.1 The waterfall model



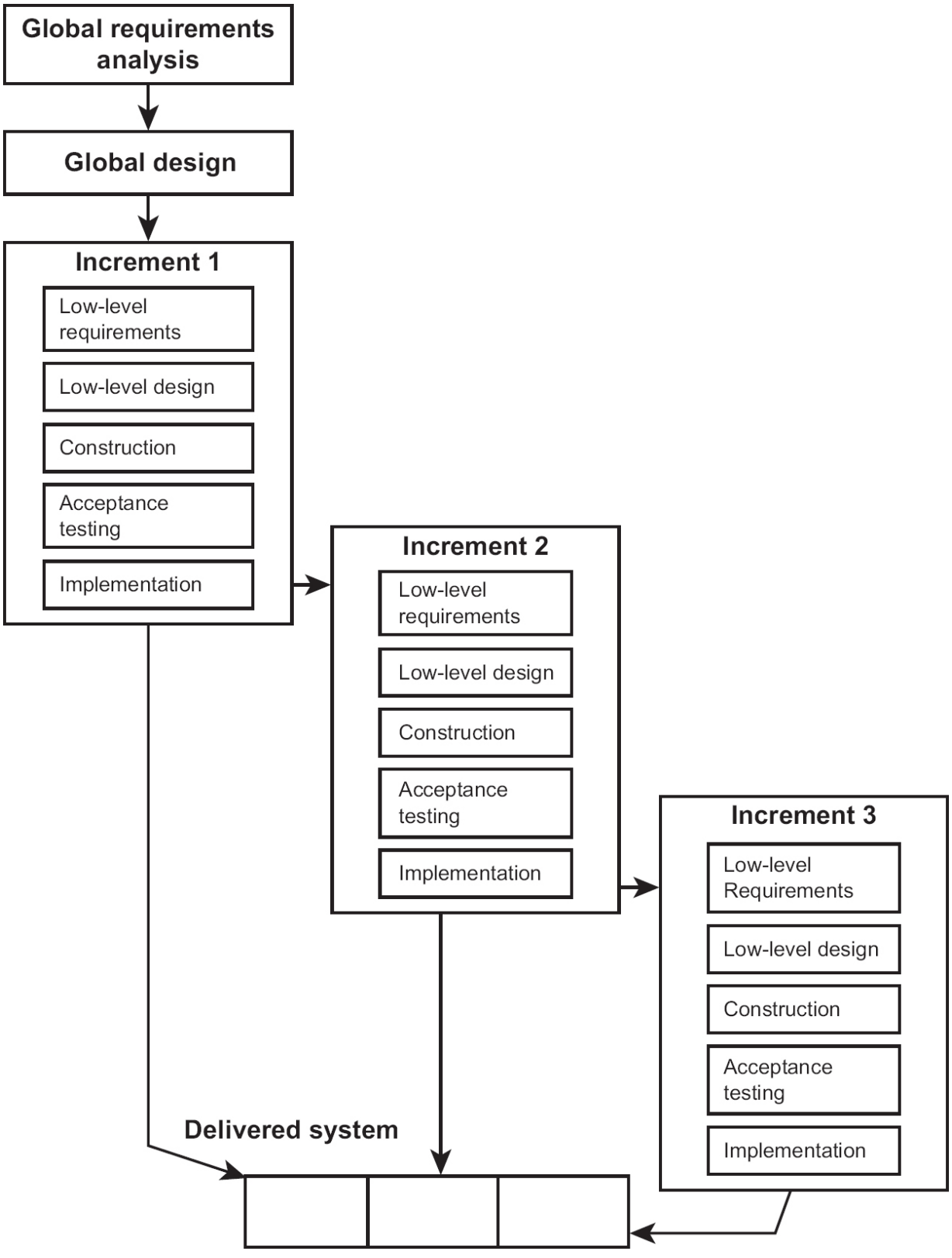
1.7.2 Incremental model

Although the incremental model (see Figure 1.2) is similar to the waterfall model, it involves the development and delivery of functionality in fragments or **increments**. Typically, global requirements are defined and an overall architecture designed. Then the product is developed in increments. After each increment is designed, developed and tested, it is system tested and then becomes operational, so that users get their new system in instalments. This approach works best when the requirements are

relatively well-known. It can work well with larger projects, as these are effectively broken into a series of mini-projects, each delivering an increment.

The incremental model is often used in conjunction with **timeboxes**. The deadline for completion of the increment is fixed and the features to be delivered by the increment are ranked according to importance. The least important features may be dropped to ensure that the deadline is met. The dropped features can be implemented in a subsequent increment if they are still required.

Figure 1.2 An incremental model

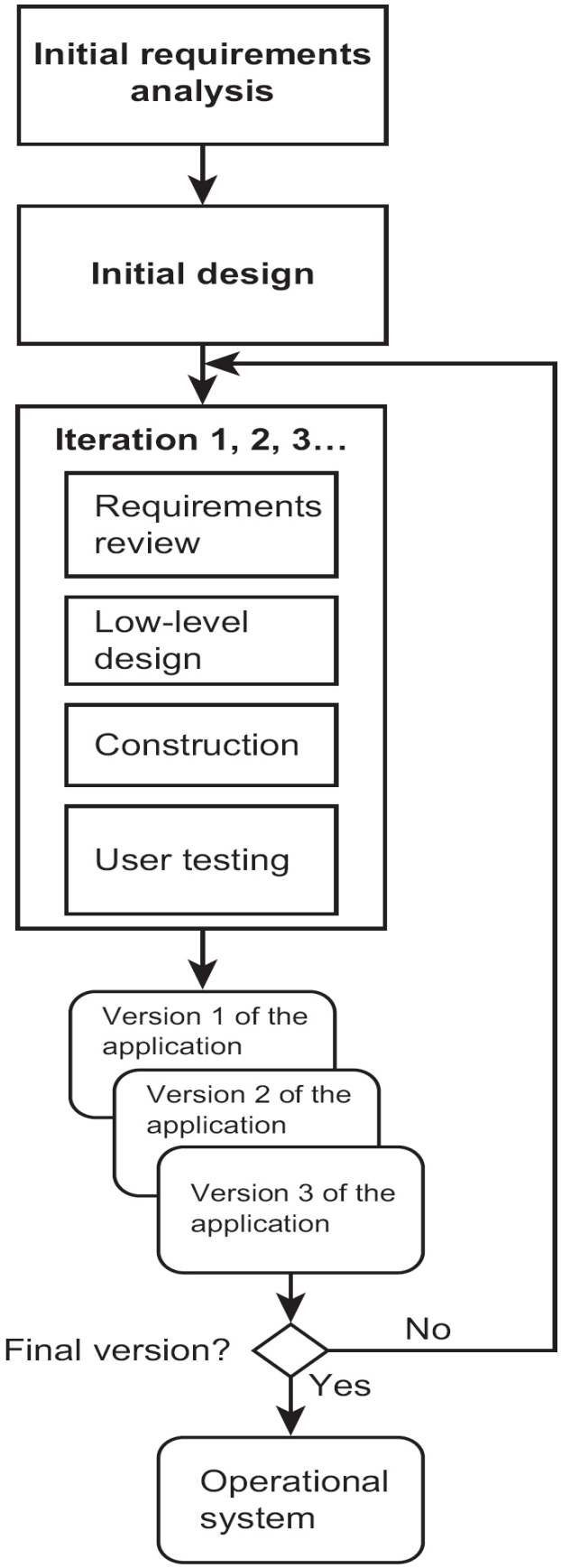


1.7.3 Iterative model

This model (see Figure 1.3) is suited to situations in which the requirements are not clearly understood and where there is a need to begin development quickly to create a version of the product which will demonstrate its look and feel. It is also referred to as an **evolutionary** approach.

Early versions, or **prototypes**, of the system are created to help the customer identify and refine requirements and design features. The customer can make suggestions for possible changes to be incorporated into a further version of the software, which is then built and evaluated.

Figure 1.3 An iterative model


Initial requirements analysis
Initial design
Iteration 1, 2, 3...
Requirements review
Low-level design
Construction
User testing
Version 1 of the application
Version 2 of the application
Version 3 of the application
Final version?
No
Yes
Operational system

A possible drawback associated with this model is not knowing when to stop iterating. The iterative approach is potentially difficult to monitor and control.

The incremental and iterative models work well together. An application can be broken down into a number of increments, each of which can be implemented through a series of iterations. The **Dynamic Systems Development Method** (**DSDM**) is an approach that formalises the combined incremental/iterative approach.

1.7.4 Agile project practices

Conventional phased projects are sometimes seen as having excessively time-consuming lines of communications. For example, users of a new system could be consulted at the requirements phase, but then have to wait a considerable time while the system is designed, built and tested. It is often only then that discrepancies in understanding of the requirements are detected.

A response to the perceived limitations of the waterfall approach has been demand for the adoption of what are called **Agile practices**. These tend to be practices that reduce bureaucratic obstacles by encouraging intense, informal communication between project participants. Some of these work at the level of individual work teams. **Scrum**, for example, breaks a project into a number of increments (see Section 1.7.2) of about two-week duration called sprints. Within sprints, individual activities are broken into a series of small steps that are listed in a **backlog**. Each day the project team members report on their progress in implementing the items in the backlog in short **‘stand-up’ meetings**. Other Agile approaches – such as **Extreme Programming** (**XP**) – focus on software

development practices. XP, for example, calls for software developers to work together in pairs, so the coding decisions of one developer are always checked by the other as the code is entered at a workstation.

These strategies are only applicable to some types of projects. They may be useful in software development, as code is relatively easy to change, but not in large infrastructure projects – although this may be changing with the development of computer-aided design.

Agile practices focus on creating real-time communication between project stakeholders. Thus, a knowledgeable product owner may be designated who acts as the sole authority on user requirements, thus avoiding lengthy requirements gathering. User representatives may work directly with developers creating ‘user stories’ precisely mapped to delivered units of functionality, and at the same time design the test cases and expected results that will be used to validate them. As noted above, developers may work in pairs so that there is instant checking on work as it is executed.

This sensitivity to the exact needs of users leads to greater user satisfaction, but at the cost of demands on the time of users. Local satisfaction is sometimes balanced by overall reduction in organisational efficiency with increased costs for the maintenance of a variety of non-standard products carrying out similar functions.

A challenge comes when projects have to integrate activities that relate to different professional disciplines. Agile practices are likely to be applicable to only a small subset of these activities. A common overall project management system is needed to coordinate the efforts of a mixture of activities, some of which are Agile and some of

which are conventionally organised. The Agile emphasis on fixed time and cost constraints helps this.

Regardless of the type of work, content managers of work packages need to be able to supply basic progress data to project management. Where Agile practices are involved, this may be easier to supply because work is more visible.

We will be returning to other relevant Agile practices in later chapters, particularly Chapter 5, which discusses quality issues.

1.8 PROJECT PLANS

The effective management of a project is based on a firm definition of **project governance**; that is, the principles, policies and frameworks by which a project is directed and managed. Among other things, this will set out who is accountable and responsible for each component of the project. **Accountability** refers to the management role which controls and reports on a particular activity within the project. The accountable person might well not be carrying out the work. Someone else who reports to the accountable person could have **responsibility** for the practical work. If there were external factors preventing work being completed as required, the responsible team member would need to consult the person accountable for the work. Thus, a manager might be made accountable for delivered software being correct. One way of ensuring correctness is by testing and the manager could allocate a system tester to have the responsibility of carrying out this task. If the tester finds that, for example, the testing is being delayed because of late changes to the functionality, they would need to refer to the manager for guidance. At the start of the project it might not be

possible for all the accountable and responsible people to be personally named. Instead **roles** would be defined and then be allocated to actual people later in the project.

During the project set-up phase (see Section 1.4.3), a project plan is produced that consists of several different types of document, including activity networks and Gantt charts (see Chapter 2). It is used as the basis for the final decision by an organisation's management to go ahead with the project, and as such is sometimes referred to as **project initiation documentation** (**PID**).

The PID is a set of documents that coordinate all project processes, bringing together all the planning documents used to manage and control the project. It is not cast in stone and will be amended as necessary during the lifetime of the project. The plan defines the project's scope, schedule and cost, as well as the supporting processes related to risk, procurement, human resources, communication and quality.

1.8.1 Project initiation documentation (PID)

Different project management methods give this document different names, but in essence it serves as an agreement between the sponsors and the developers of the project. It has the following key elements:

- An introduction, which describes:
  - the project background;
  - the document's purpose;
  - the business justification for the project, including a brief summary of costs and benefits (see Section 1.9).

- Project goals, objectives and deliverables, including:
  - project success and completion criteria;
  - an outline of the technical strategy by which the goals will be achieved.

The following sections constitute the **project management plan** and explain how the project will be managed:

- A project organisation chart, which includes:
  - the project sponsor;
  - the project manager;
  - major stakeholder representatives;
  - the lead supplier representative(s), if appropriate.

The organisation chart shows the reporting relationships of all those with managerial responsibility for aspects of the project. It also identifies the members of any steering committee (or project board or project management board) established for the project.

Chapter 8 discusses some of the issues involved in project organisation.

- A management control section, which covers:
  - the frequency, timing, recipients and format of progress reports;
  - how the plan will be produced and maintained;
  - what information will be monitored and recorded;
  - how the information will be recorded;

- how packages of work will be signed off and reviews conducted, and who is authorised to sign off packages of work;
- the people accountable/responsible for authorising the commitment of resources;
- the people accountable/responsible for recording and assessing the impact of any changes;
- the people accountable/responsible for authorising different levels of change to goals, objectives, deliverables, costs or completion date.

Chapter 3 is devoted to monitoring and control.

The following constitute the **initial project plan**, which describes the activities to be undertaken to complete the project:

- A **project structure** section that describes how the project will be broken down into manageable portions of work, which will be administered as stages/phases.
- A list of **project milestones**, that is significant events in the project for which dates need to be clearly specified. Milestones are used to measure the progress of a project and can be the start or completion of a major project phase. Milestones are events that consume no resources in themselves, but enjoy a great deal of attention from management at a senior level. Management resources representing the project sponsors will be needed to review the state of the project before or after the milestone has been reached.
- A **risks and assumptions** section identifying high-level risks to the project and specific actions to reduce or eliminate each risk. It

is useful to include in this section a list of assumptions made in producing the report.

See Chapter 7 for more on risk and its management.

- A **communication plan** that provides an overview of how the project will communicate with the wider business organisation, particularly with regard to changes needed in the business in order to make the implemented IT application effective.

See Chapter 8 for more on project communication,

- A **report sign-off** section.

The document should be signed off by staff who represent all areas and functions committing resources to the project and those who will be affected by the project. In so doing, they implicitly accept the assumptions listed. Final sign-off is given by the project sponsor. The project manager should not initiate any work that has not been explicitly or implicitly authorised in the project initiation document and signed off by the project sponsor.

1.8.2 Creating plans

The PID will contain an outline plan for the whole project and a more detailed one for the first stage. The detailed planning of later stages usually takes place closer to the beginning of those stages. This allows it to take account of the additional information gained during the progress of the project. Thus, plans are created at several levels:

- **Projects**, where the outline plan is approved by the project sponsor, supported by any steering group/project board.

- **Stages**, where the completion of any previous stage is signed off and the **stage plan** for the next stage is approved by the project sponsor.
- **Work packages**, which are the practical sets of activities that the project manager allocates to individuals or groups of developers under the guidance of a team leader or manager, who produces a detailed plan of how the work will be done.

The plan starts with a careful breakdown of the work to be done. The activities needed to carry out this work are identified plus the order in which they have to be carried out. An estimate of the duration of each activity is needed. Chapter 2 describes the use of Gantt charts for showing the schedule for a project.

The project plan then needs to be developed further to account for the various types of resources, including people, equipment and facilities needed for the activities identified. With internally resourced IT/business change projects, resources must be obtained from other parts of the company to work on a project. Managers of specialist departments will be key providers of resources such as office space, computer equipment and specialist expertise. Where goods or services from outside the organisation are needed, contract management will be important.

Resource planning identifies the skills needed for the project and when they are needed; for example, a testing expert may only be needed at certain times in the project. Availability of resources often affects the timing of activities.

A useful communication tool for the project manager is the **responsibility assignment matrix** (**RAM**), a simple matrix showing

individuals associated with the project on one axis and the activities for which they are responsible on the other. It provides a quick and easy view of who is responsible for each task.

Once the activities and resources have been identified, the costs are assessed. The business case for a project is based on it not costing more than a specific amount reckoned to be the value of project benefits. If the costs of the project exceed the value of its benefits, the project becomes uneconomic. It is also possible for an organisation to simply run out of money for a project.

Internal IT projects are usually paid for out of a user department's budget. Where a project is being carried out for an external customer, there may be a fixed-price contract. Whatever the case, it is necessary to estimate quantities and costs, set budgets and eventually control expenditure.

Sources of development staff

A major IT project often requires an abnormal amount of technical work that the business cannot provide by itself. Additional development skills might be acquired by:

- Recruiting individual specialists on temporary contracts, either directly or by using agencies. Pay for temporary staff is usually greater than that for permanent staff and there may be additional recruitment and training costs. Depending on the type of work, temporary staff could work from their own premises, but usually need workstations within the business.

- In the case above, the temporary staff are employed directly by the business. An alternative approach is to have a contract with an outside specialist company where the developers used are based and remain the employees of the specialist company. The specialist company is paid for the hours that their employees work.
- A more radical approach is to outsource one or more project activities, such as design and construction, to an outside specialist company. The outside company will have management responsibility for delivery of a subset of the project requirements, often for a fixed price. In this case, the supplier selection and the acceptance testing of deliverables would still consume some of the customer's resources.

1.8.3 Communication planning

Stakeholders, including the project team, have varying needs for communication about the project. These needs and the means by which they may be satisfied are recorded in a communication plan. The communication plan includes, among other things:

- the flows of communication during the project;
- how various communication tools will be used;
- what meetings will be held with what attendees, and at what times.

These issues are explored further in Chapter 8.

1.8.4 Quality planning

A carefully considered quality plan is needed in order to develop a system that meets all the functional and quality requirements documented during analysis.

Quality criteria can be applied to both project products and processes. It is possible to check that deliverable and intermediate products have specified qualities, and that the processes that created them were the correct ones.

Quality needs vary between projects. The deliverables of some projects could be safety critical and would require more stringent quality assessment. This would add to the costs of the project.

The customer must be the final judge of the product's quality, and must therefore be involved in quality evaluation. Chapter 5 further explores the role and content of the quality plan.

1.9 THE BUSINESS CASE AND BENEFITS MANAGEMENT

The business case is a key document for any project. As noted in Section 1.4.1, the proposal for a project may be triggered in a variety of ways. Once a proposal has been made, senior management will decide whether to go ahead with a proposal, using a combination of qualitative and quantitative criteria, as no single method gives enough information.

Qualitative criteria include **organisational fit**. Does the project fit with the organisation's strategic objectives? How does the proposed project contribute to the organisation's future capabilities and growth? The **business risks** associated with a particular project will be considered. These are external factors that could reduce the

value of a project. For example, the project deliverables might be produced as planned, but the demand for them might disappear because superior technologies have come onto the market.

A more persuasive reason for going ahead with a project, however, is often **financial justification**, where two of the most common quantitative financial criteria are **return on investment** and **payback period** – see Sections 1.9.1 and 1.9.2.

The contents of the business case document would include a description of the project, its objectives, benefits and scope, a cost–benefit analysis, a risk analysis, a conceptual solution, resource requirements and success criteria. Some organisations include alternative options in the business case in order to compare the recommended solution against other approaches. The business case needs to be carefully reviewed by project sponsors.

With IT-related projects, IT specialists should be involved in creating the business case because they contribute information about IT-related development and operational costs. However, the really important participants are the stakeholders who will use, manage and hopefully benefit from the newly developed systems. The project sponsor is accountable for the value of benefits of the proposed system and the non-IT costs of the project, such as those arising from organisational changes.

Both the **development** and the **operational** costs need to be considered. This means the project life cycle needs to be extended into a **system** (or **product**) life cycle that takes account of all the costs and income that can be expected over the whole life of the delivered system, up to and including any decommissioning costs.

This is similar to the concept of the **total cost of ownership** (**TCO**) of an asset.

A financially viable project is one where the value of benefits exceeds costs. Conveniently, some benefits can be measured in monetary terms. This would be the case where the project will enable staff costs to be cut. (Note that reducing cost is essentially the same as increasing income. If you save £100 on the fuel bill, you can spend that money somewhere else.) Other benefits are measurable, but putting a precise financial benefit on them is difficult, as in the case of software that diagnoses a particular medical complaint. Finally, some benefits are simply not quantifiable, as when the managing director decides to spend money on a fountain outside the new data centre to impress potential customers. In short, the client organisation will put their own value on many of the possible benefits.

Organisations often consider several projects at the same time. Where money is constrained, they will use business case reports to prioritise which project to start.

1.9.1 Return on investment (ROI)

One way of measuring the financial outcomes of different project proposals is through **Return on Investment** (**ROI**). The simplest form of ROI is calculated as:

$$\text{ROI} = \frac{(\text{annual average profit})}{(\text{total investment})} \times 100$$

Say a company uses an external accounting services business to administer their payroll at the cost of £7500 a year. As part of a review, the company examines the cost of bringing payroll in-house.

It finds it could run its own payroll at an initial cost of £10,000 to set up the new system and annual operating costs of £5000 (the sums are selected purely for ease of calculation). These cash flows are shown in Table 1.2 for a five-year period.

Table 1.2 Example cash flows

Year	Expenditure	Income	Cash flow	Accumulated
0	£10,000.00	£0.00	-£10,000.00	-£10,000.00
1	£5,000.00	£7,500.00	£2,500.00	-£7,500.00
2	£5,000.00	£7,500.00	£2,500.00	-£5,000.00
3	£5,000.00	£7,500.00	£2,500.00	-£2,500.00
4	£5,000.00	£7,500.00	£2,500.00	£0.00
5	£5,000.00	£7,500.00	£2,500.00	£2,500.00
total	£35,000.00	£37,500.00	£2,500.00	
			average profit	£500.00
			ROI	5.00%

Cash flows are calculated at the end of each year. Year 0 refers to expenditure that takes place before the new system is operational and the initial debt that exists from investment in the system. In this case it is £10,000.

Each yearly cash flow is calculated by subtracting the expenditure from income in each year. The ‘income’ here is actually the saving from not using the external service provider. We noted above that a saving is effectively the same as income. Annual average profit is the total of all the annual cash flows (including the initial investment) divided by the number of years of actual operation – that is £2500 divided by five years, which is £500. This is converted into ROI by

dividing it by the original investment of £10,000 and converting it to a percentage by multiplying by 100. This gives an ROI of 5 per cent.

1.9.2 Payback period calculation

Another way of measuring financial viability is by calculating the **payback period**. This is the point at which the delivered new system pays off the initial investment and starts to generate a surplus. If you look at the accumulated column in the table (which is a bit like the current bank balance of the project), it can be seen that this is zero in year 4. This means that at the end of year 4 the project investment has been paid off. The smaller the payback period, the better.

A more complicated assessment approach calculates the **net present value** of the project to be delivered. This calculates all the cash flows in terms of how much money you would need to invest now at a particular interest rate to get that amount of money in the future. This is called **a discounted cash flow** (or **DCF**).

COMPLEMENTARY READING

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

1.10 TRANSITION STRATEGIES

Transition is when the products of the project are deployed within the client organisation and are then used by operational staff to generate

the planned benefits of the project. The IT elements could well be just one part of a bigger system delivery that includes interfaces with existing systems and human interactions. When a new system is implemented it is often the case that data from predecessor systems will have to be transferred. With the Water Holiday Company integration project, data from two different organisations will need to be merged. The project team will recommend the most suitable changeover method for the project. The following options may be considered.

1.10.1 Direct changeover

In this case, the old system is discarded and immediately replaced by the new one. It can be considered a risky approach, but is relatively inexpensive if thorough testing has been done. DevOps technologies have been developed which automate many of the processes associated with updating web-based applications, but this is most likely to be successful where the IT system is changing but not the user operations.

1.10.2 Parallel running

This involves running the old and new systems together for a period of time using the same inputs and comparing the related outputs – so it serves as a continuation of the testing process. It is a safe, low-risk approach, but can be expensive, particularly in terms of duplicated labour costs. An important management decision is how long the two systems should be run in parallel. Parallel processing works best where there are batched outputs, for example with monthly payment runs in payroll.

1.10.3 Phased take-on

The phased approach breaks the system into components that will be introduced in sequence. It helps to minimise risks but can delay the implementation of the entire integrated system. However, it does present the opportunity to allow users to learn one system component at a time. It also fits neatly with incremental delivery.

1.10.4 Pilot changeover

Like the phased approach, this is a risk-reducing approach. With pilot changeover, the entire new system is introduced to just one business unit or location. It can only be used if the business unit or location can use the entire system independently. Problems can be addressed and fixed before the system is introduced company-wide, but company-wide deployment of the entire system is consequently delayed.

ACTIVITY 1.4

What would be the best implementation strategy for the Water Holiday Company integration project?

1.11 POST-IMPLEMENTATION REVIEW

The **post-implementation review** (**PIR**), or **project evaluation review**, is usually scheduled to take place some 6 to 12 months after the sign-off of the project. Its objective is to review the implemented system in terms of its contribution to business objectives, its usability, operating costs and reliability. It considers the following:

- whether the business and system requirements have been met;

- cost and benefit performance;
- operational performance;
- controls, auditability, security and contingency;
- ease of use.

The output from this process is a **post-implementation review report**. The review should be led by someone who is independent of the project, and should solicit feedback from users, operations and the support team. The review should address the operational system and not the development project. The additional effort required of them might lead to users not engaging with this review. If this is the case, it is essential to explain to the users the benefits of the process – for example, that changes which could improve the system may be identified as a result.

The PIR report needs to be distinguished from a **lessons learnt report**. When the project has been delivered by the project team, the project manager should write a report describing the major challenges experienced during the project and how they were managed. The purpose of this report is to identify lessons that would improve the execution of future projects.

SAMPLE QUESTIONS

Answers can be found on page 157

1. **Which of the following is NOT a characteristic of a project?**
    a. Ongoing nature
    b. Uniqueness

c. Clear objectives

d. Integration of interrelated tasks and resources

2. **Which of the following is NOT managed by the project manager?**

   a. Time, cost and scope

   b. The project team

   c. The project sponsor

   d. Engagement with the stakeholders

3. **Which of the following tools indicates who is responsible for what?**

   a. A responsibility assignment matrix

   b. A resource levelling chart

   c. An activity network chart

   d. A resource histogram

4. **Which one of the following does a payback calculation give to an organisation?**

   a. An assessment of how quickly an implemented system will produce a profit

   b. How much future income from an implemented project will be worth in present-day terms

   c. Annual average percentage profit of a project

   d. The overall value of the benefits that the project will deliver

POINTERS FOR ACTIVITIES

Activity 1.1

Among the activities that may be considered are the following:

i. Survey existing office requirements (for example, how many desks and chairs there are) and IT infrastructure, including servers and printers, used by the office.

ii. Survey new office space.

iii. Plan new office layout – who and what will go where?

iv. Schedule the sequence of moves – it might be that not all staff should be moved at once as this will allow for some continuity of service.

v. Select a removal company.

vi. Organise the setting up of the infrastructure.

vii. Pack up the old office.

viii. Transfer to the new office, including supervision of placement of furniture, and so on.

ix. Unpack in the new office.

x. Connect telephones, and so on.

As project manager you will probably want to be consulted when decisions have to be made, possibly at points (iii), (iv) and (v) as money is involved here. Useful checkpoints would be just before (vii) to check everything is in order to go ahead and after (x) to check out any outstanding problems.

Activity 1.2

There will be some interviews with fairly high-level managers, including the project sponsor (who may be the managing director who also owns the merged company), to clarify the business requirements of the proposed system. Given the nature of this project, marketing specialists will need to be consulted. It is essential that this consultation take place when the business case is being prepared.

Staff who currently carry out the clerical booking operations in both Canal Dreams and Minotours would need to be interviewed to see how their existing systems work, what data has to be held and what problems there are with the current systems. Operations staff at boatyards and marinas also have to be consulted as users of the new system who need information about the bookings each week, so that boats can be allocated and prepared and holiday-makers checked in. Staff in other departments need to be approached to document the interfaces between the booking operation and other parts of the overall Canal Dreams and Minotours businesses; for example, the central finance function and the maintenance staff who may need to schedule boat maintenance.

Note that users include IT operational staff, who would, for example, have requirements about system security.

We usually say that end-users should be consulted, but, of course, in internet transactions with the public it is not possible to identify who these will be beforehand. In this case, market research with recent customers could provide insights into the needs of online customers.

Activity 1.3

Where potential customers have to ‘volunteer’ to use the system, more careful design is needed than for those which employees are compelled to use. It is assumed that these external transactions will be the subject of design to ensure their ease of use, as part of broader user experience design. Transactions that are made via smartphones are ‘logically’ the same as ones made via a laptop, but need a different design to cater for the smaller amount of information that can be conveyed on a smaller display. The transactions within the overall user experience needing careful design include:

- browse locations and boats;
- check availability;
- record boat booking;
- cancel boat booking;
- change allowable details; for example, amend existing address and ‘crew list’.

Customer communications that could be made by sending email and/or text messages include: booking confirmation; final payment reminders; and proof of booking and other details needed when starting the cruise.

Activity 1.4

The holiday bookings tend to be seasonal, with very busy times of the year and the off-season when the business is dormant. This would seem to favour a direct changeover during a quiet period. This would allow a gradual build-up of traffic, making it easier to deal with any initial teething troubles. Note, however, that direct changeover is

often avoided because of the risk that if the new system is faulty there is no alternative system to fall back on.

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.

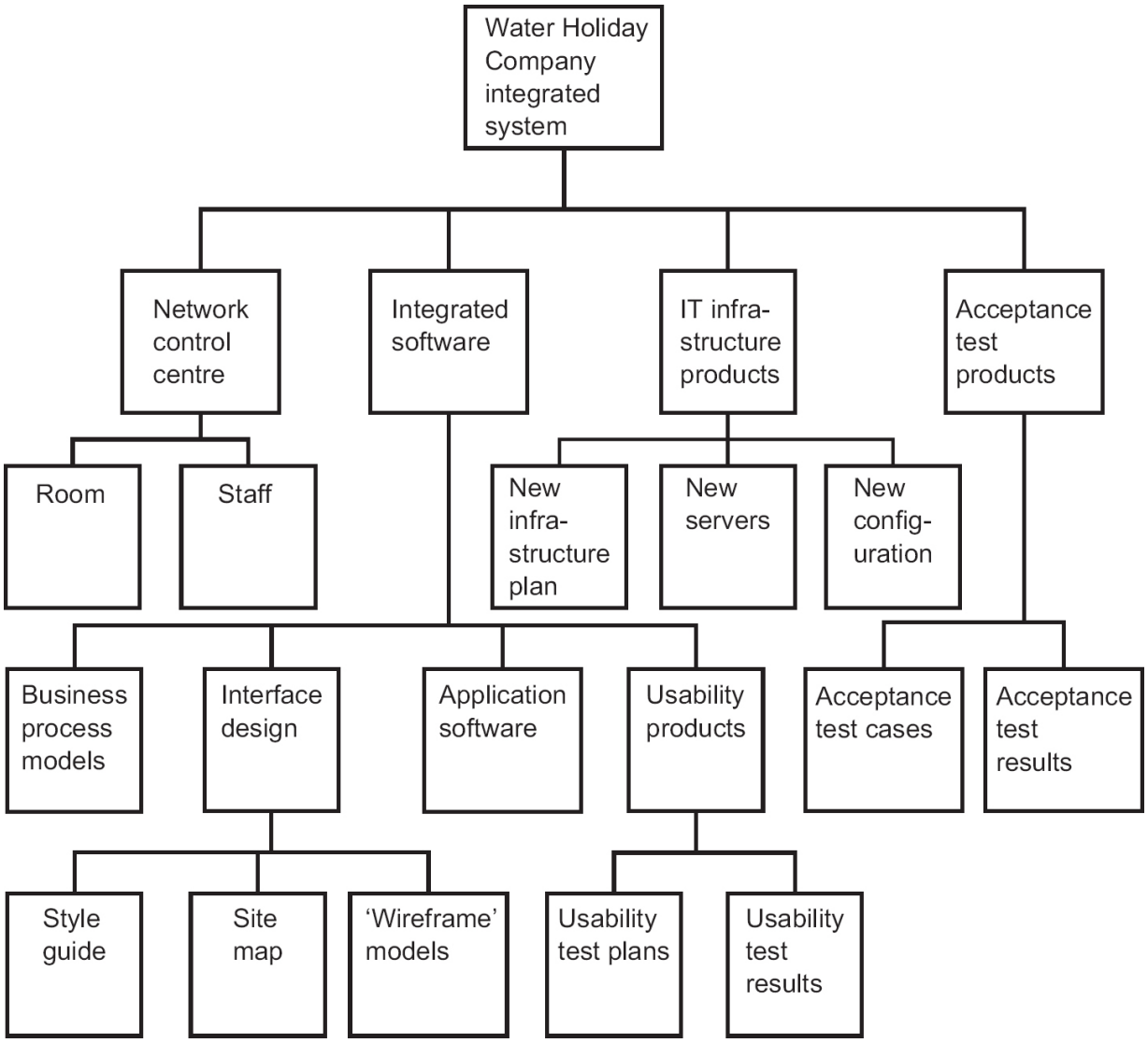
The deliverable and intermediate products can be written simply as a list of products, but sometimes they are shown in a **product breakdown structure** diagram (PBS); see Figure 2.1 for an example.

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



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

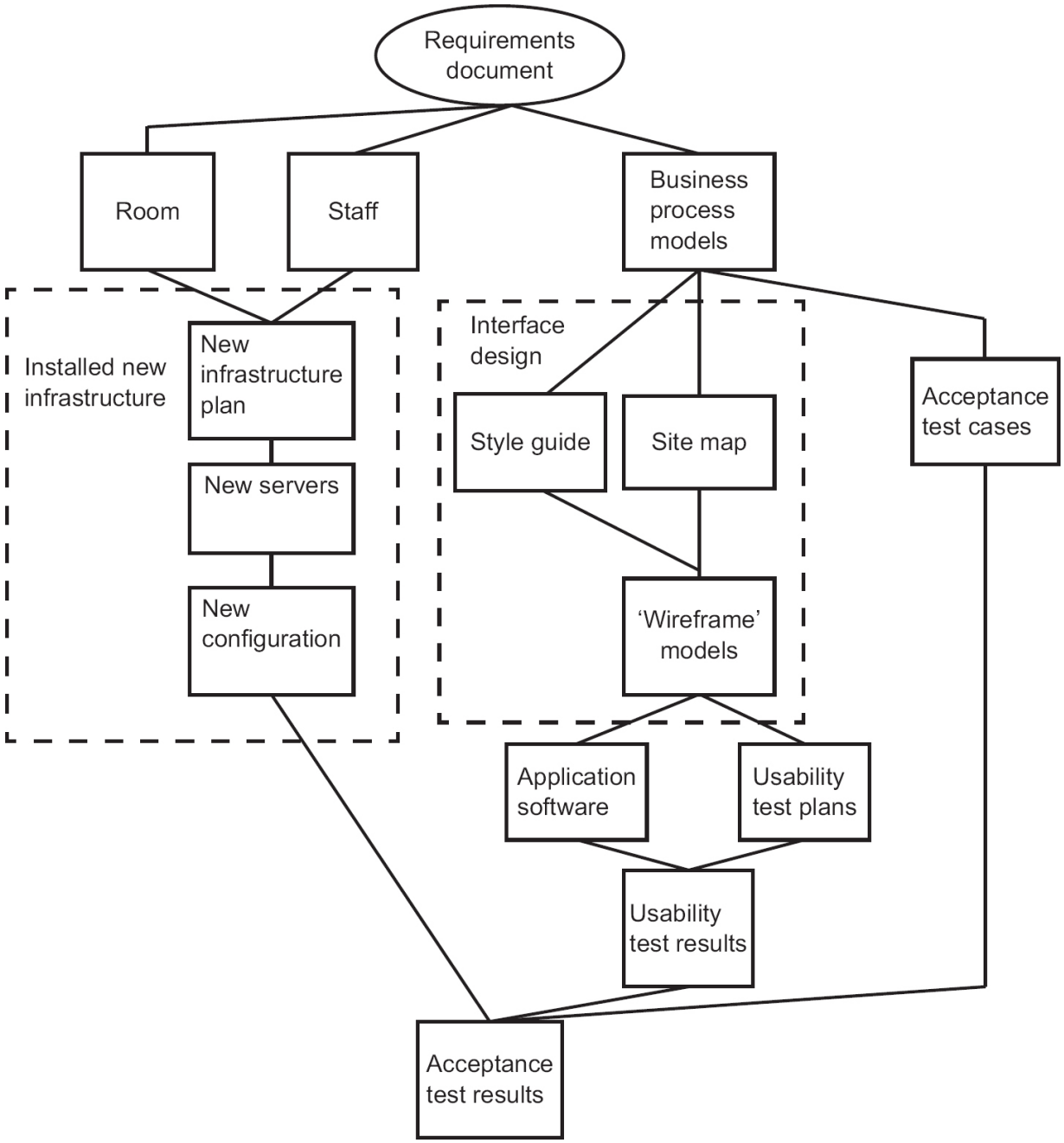
d. a project progress meeting

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



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).

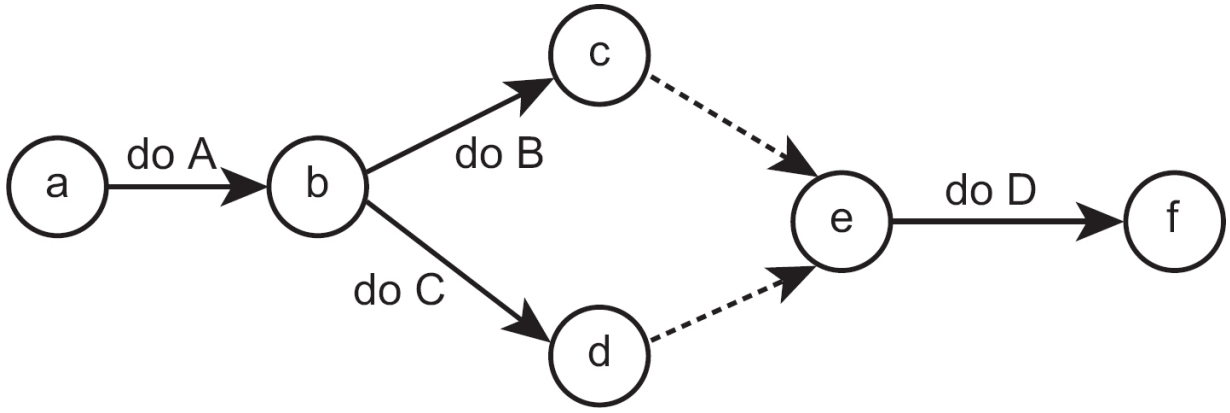
2.4 ACTIVITY PLANNING

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

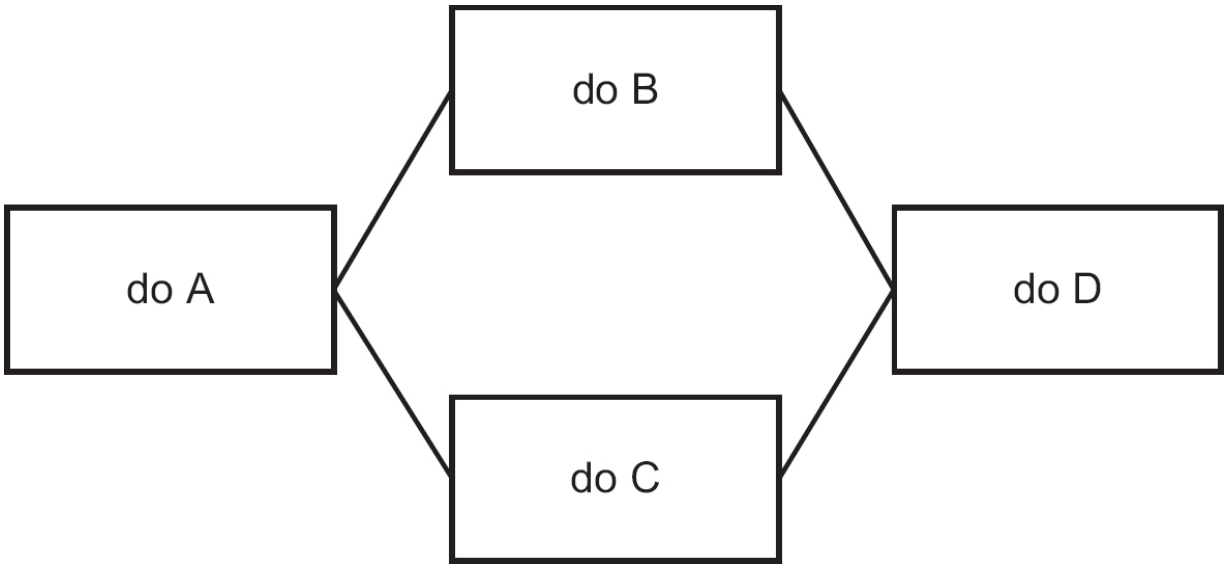
nodes) represent the ends of some activities and the starts of others. The arrows with broken lines indicate ‘dummy activities’, which simply show a dependency between two of the event nodes – for example c, the end of ‘do B’, and e, the start of ‘do D’.

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.

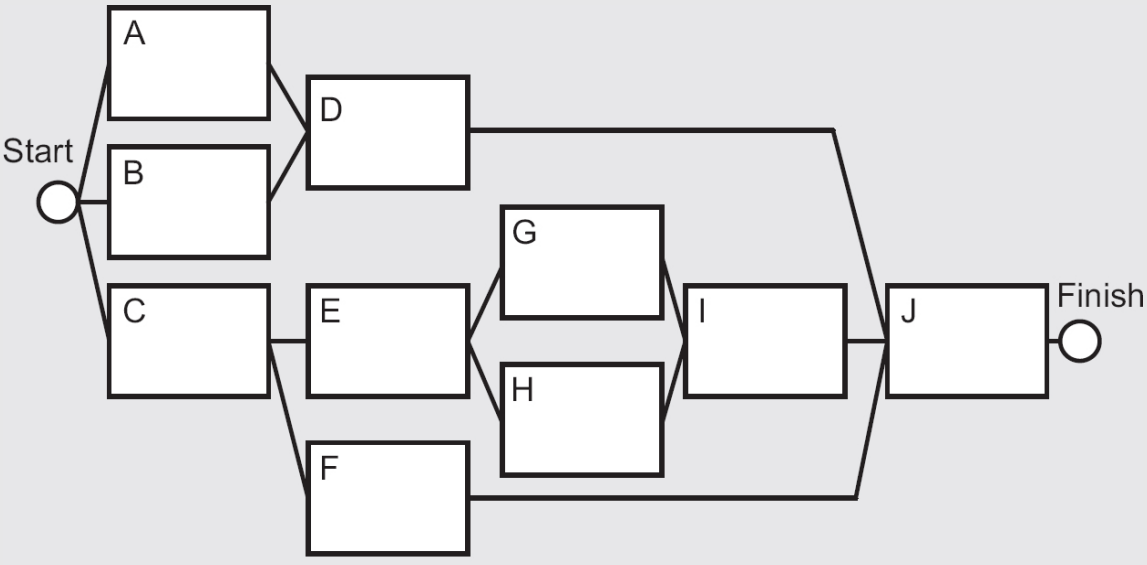
What are not taken into account at this point are **resource constraints** – for example, that a person will not be able to start on one task because he or she will not have finished another. These considerations are deferred because we do not know all the competing demands on a particular resource until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

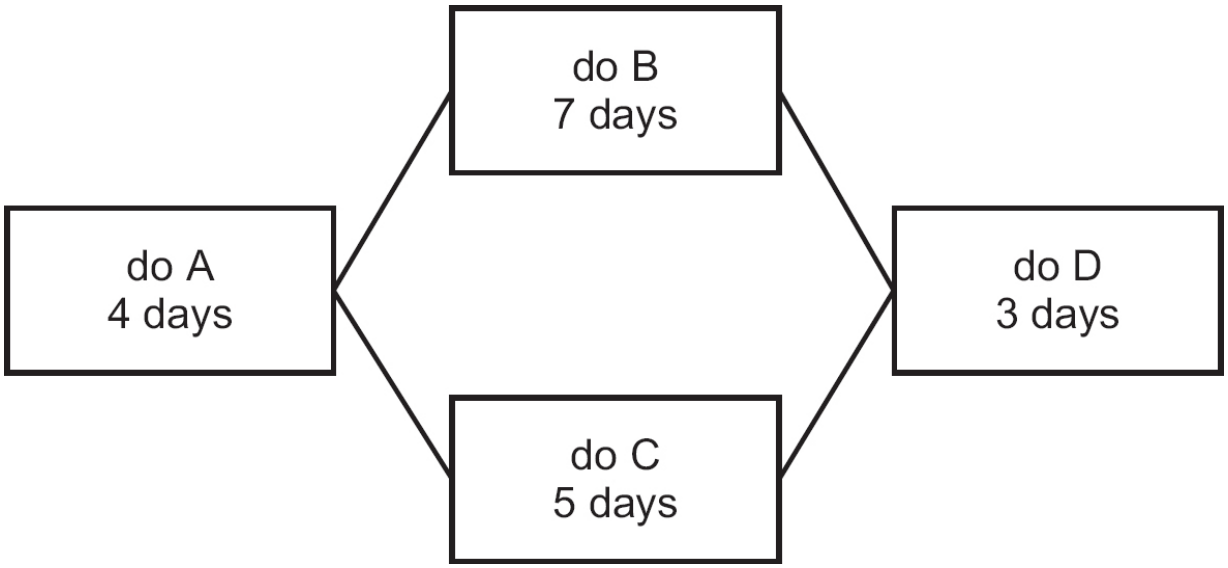
Let’s assume that we can allocate estimated durations to the activities in the activity network of Figure 2.4 (see Figure 2.5).

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.

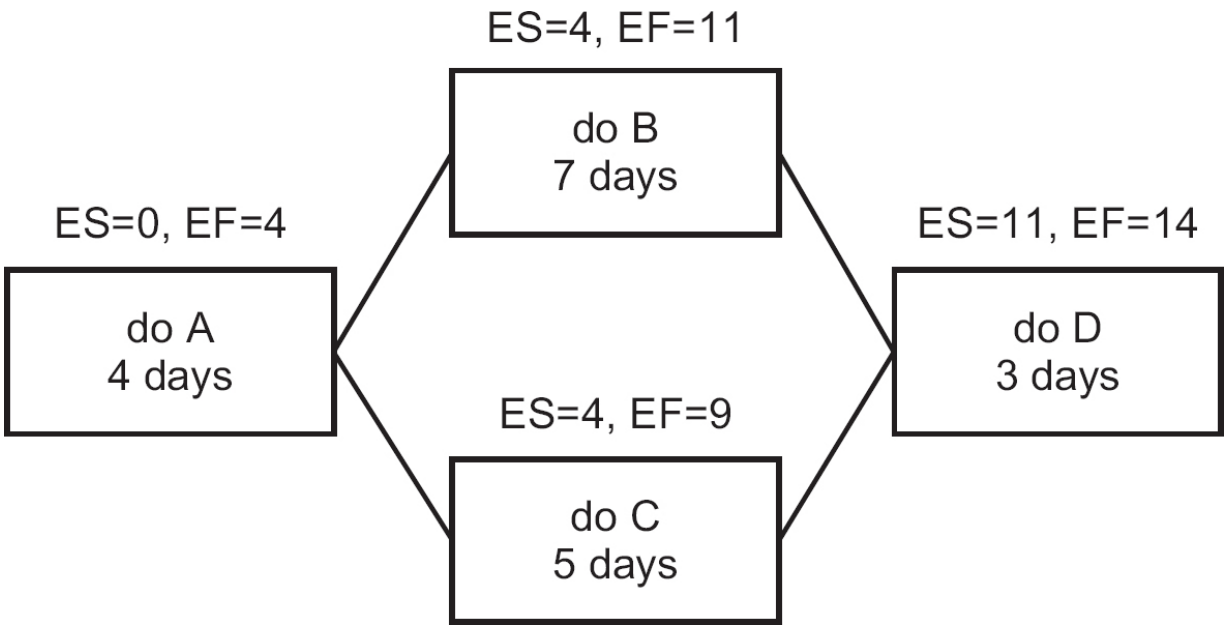
You may wonder why it is not the **following** day. Well, the convention is that when we say ‘do A’ finishes on day 4, we mean at the **end** of day 4. When we say ‘do B’ and ‘do C’ start on day 4 we really mean at the **end** of day 4, which of course really means the start of day 5. It is best just 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’.

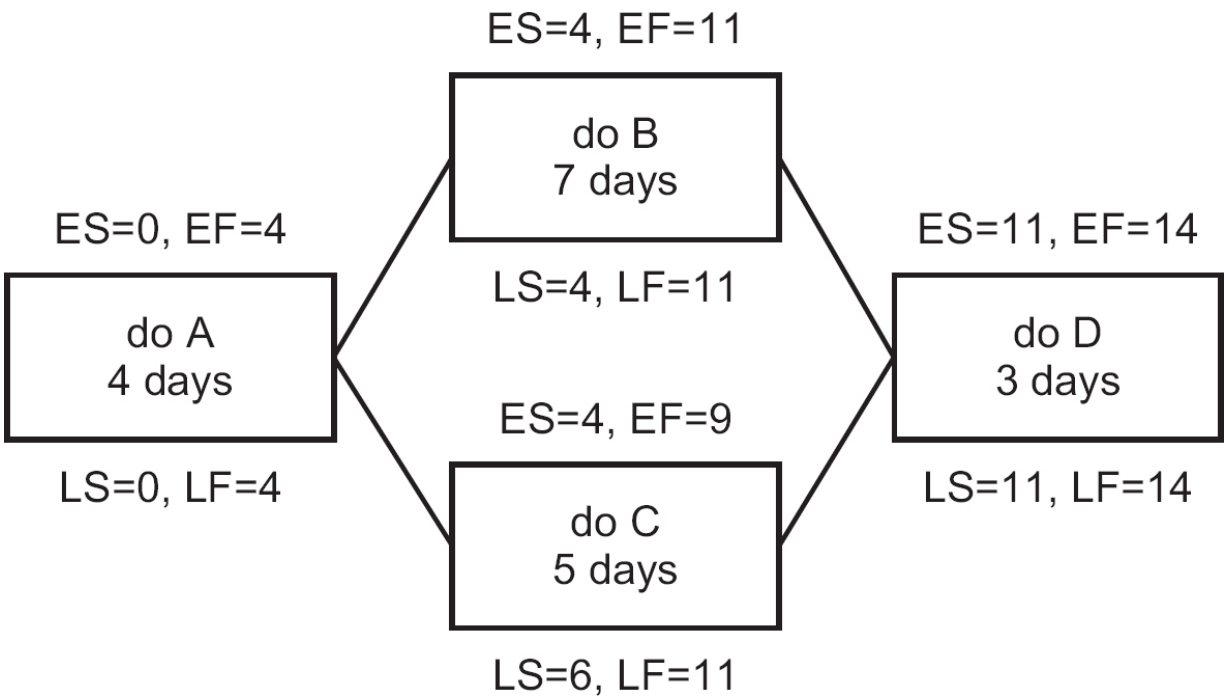
In the case of ‘do A’, we have to decide whether to base the latest finish on the latest start date of ‘do B’ or ‘do C’. The earlier of the two

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

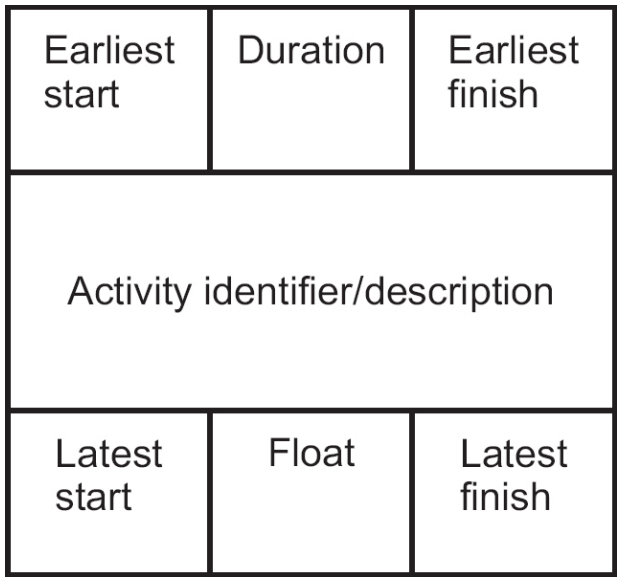


Figure 2.9 Activity box for ‘do C’

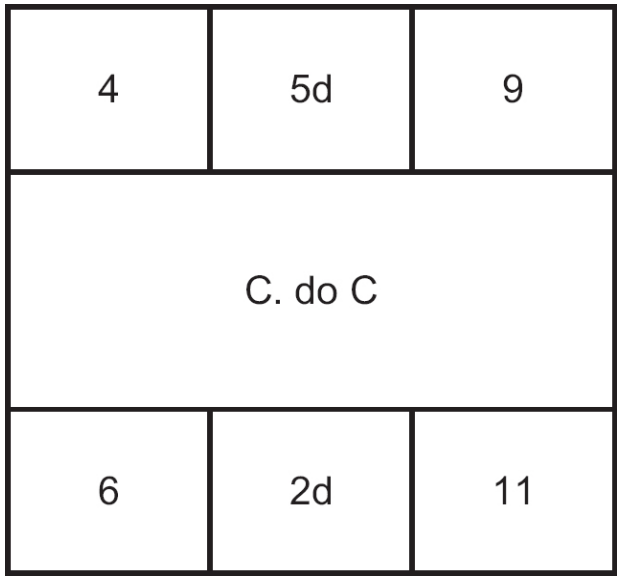
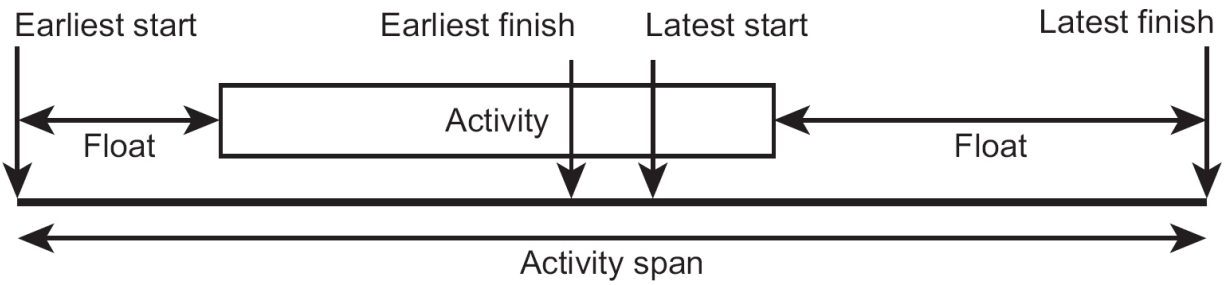


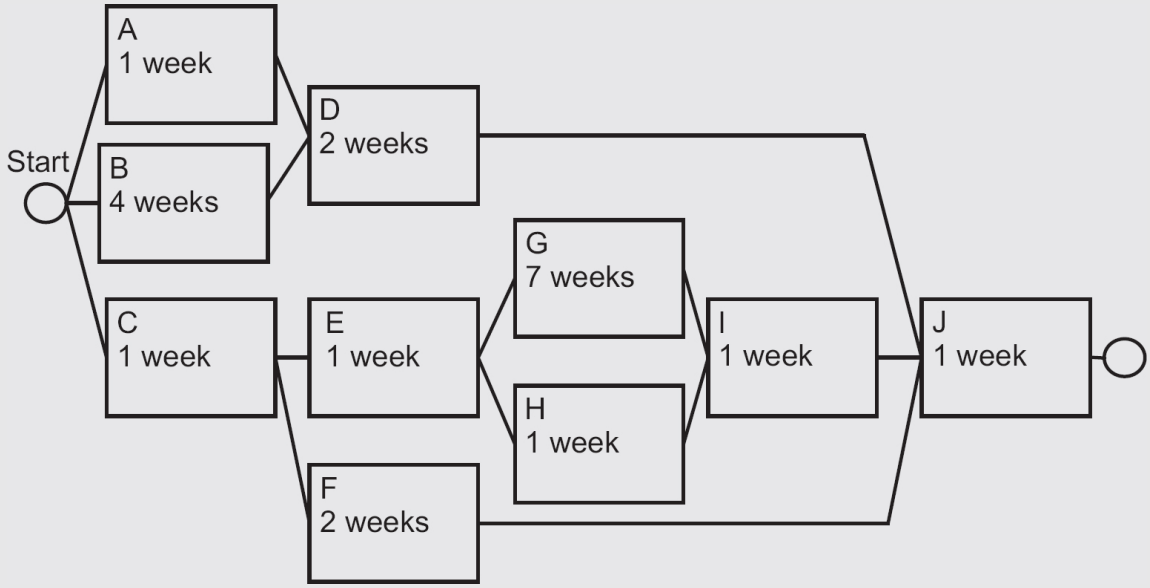
Figure 2.10 The activity span



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.



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

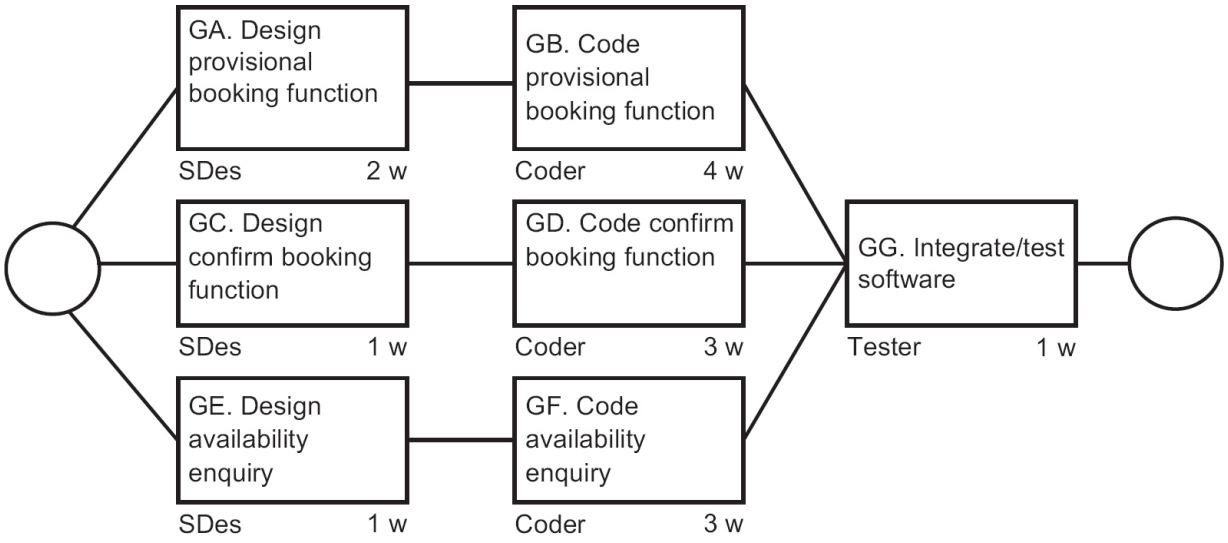
Match the following resource types and activities. More than one resource type might be needed for an activity.

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

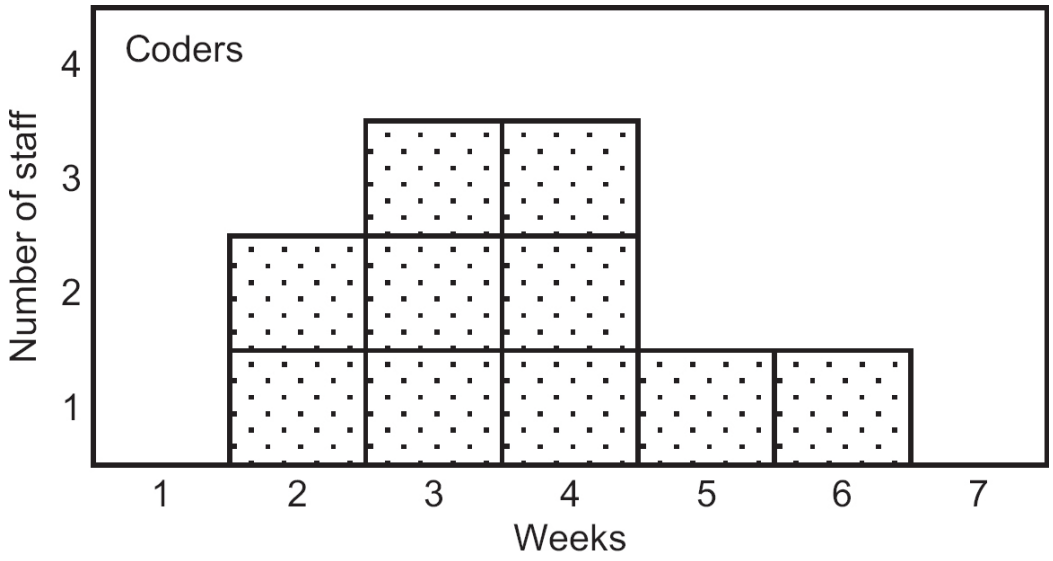


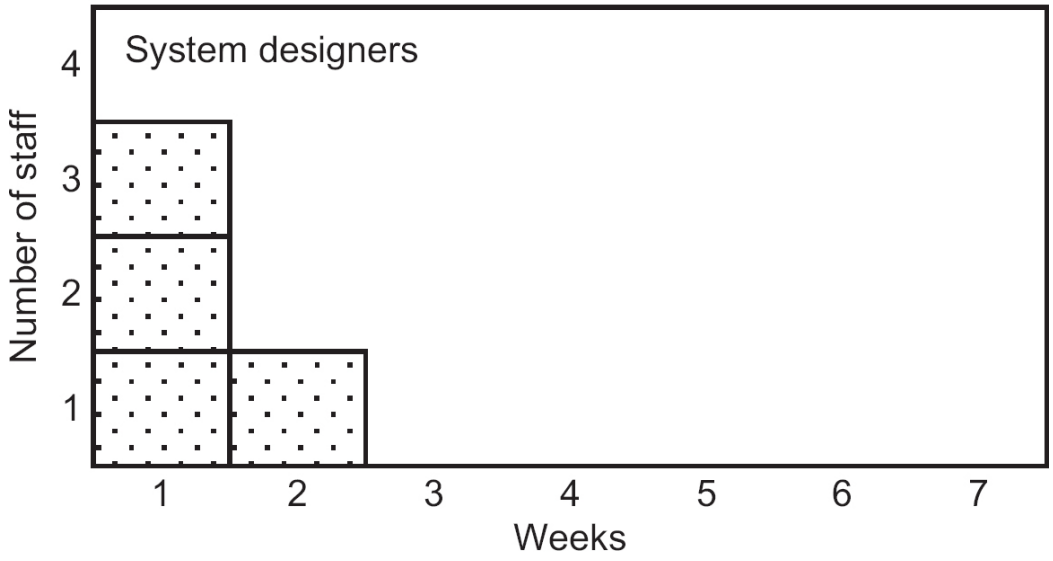
(SDes = System designer, w = week)

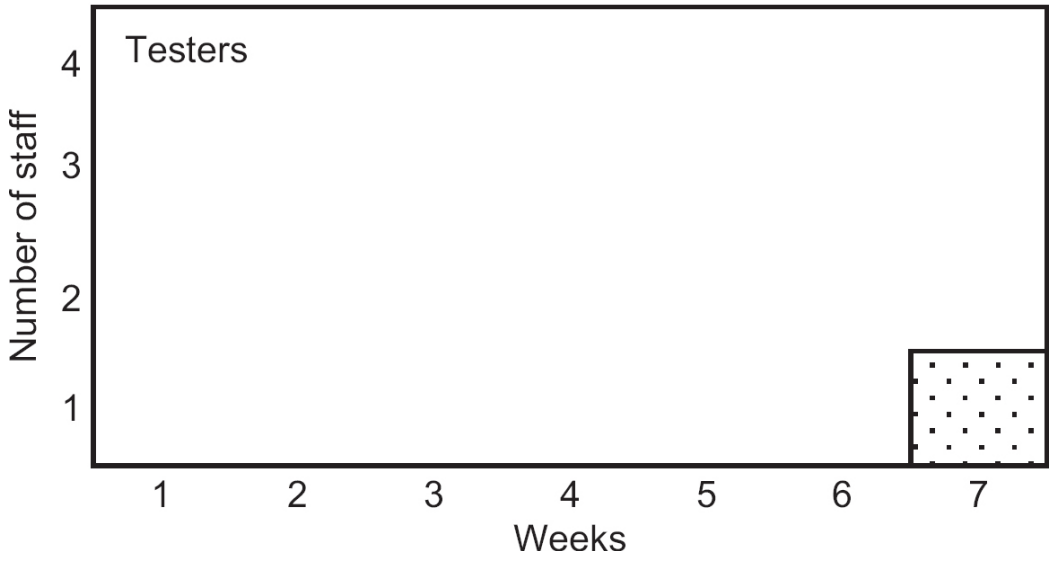
Table 2.1 Numbers of each resource type needed in each week

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.

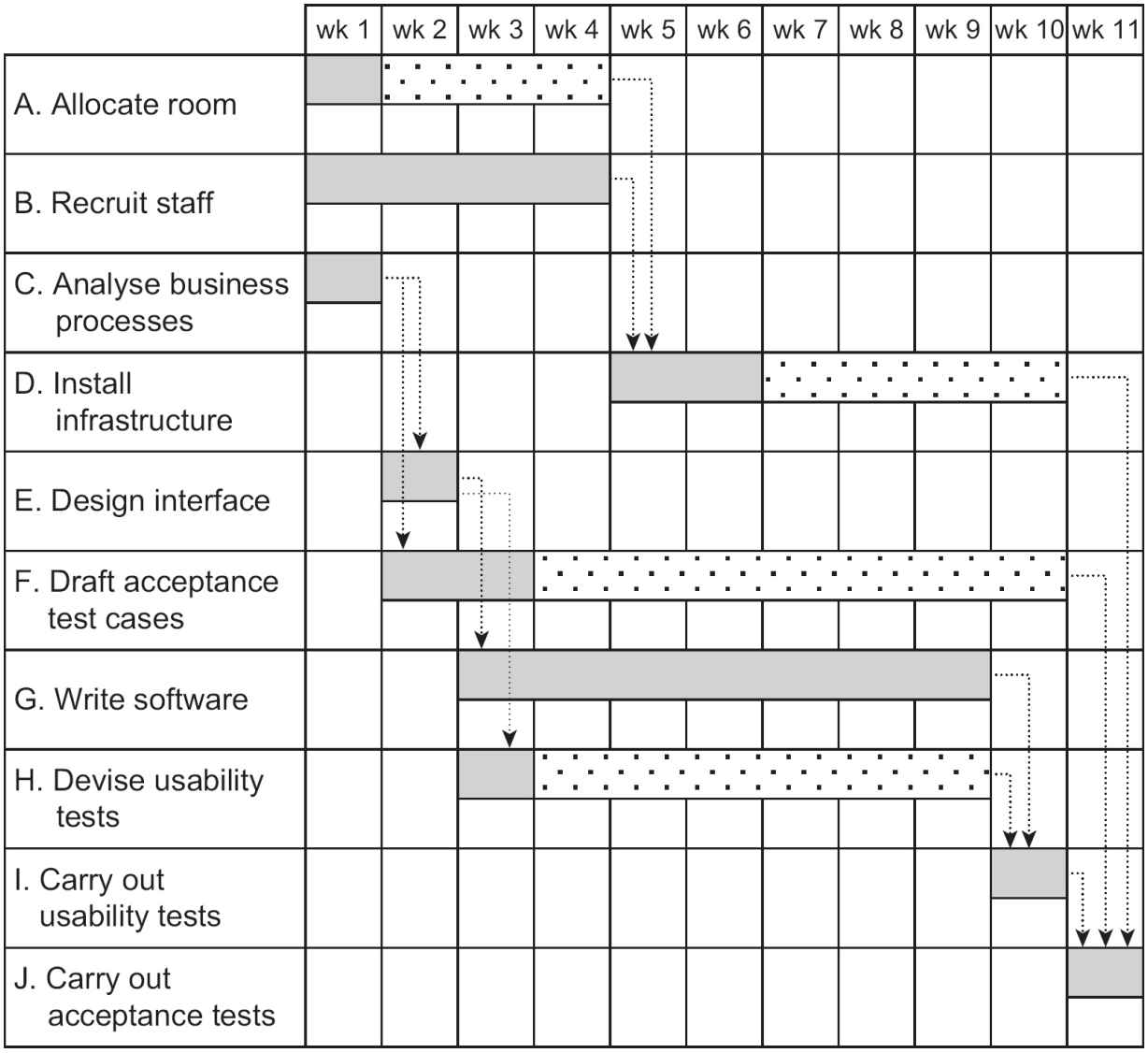
ACTIVITY 2.6

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



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

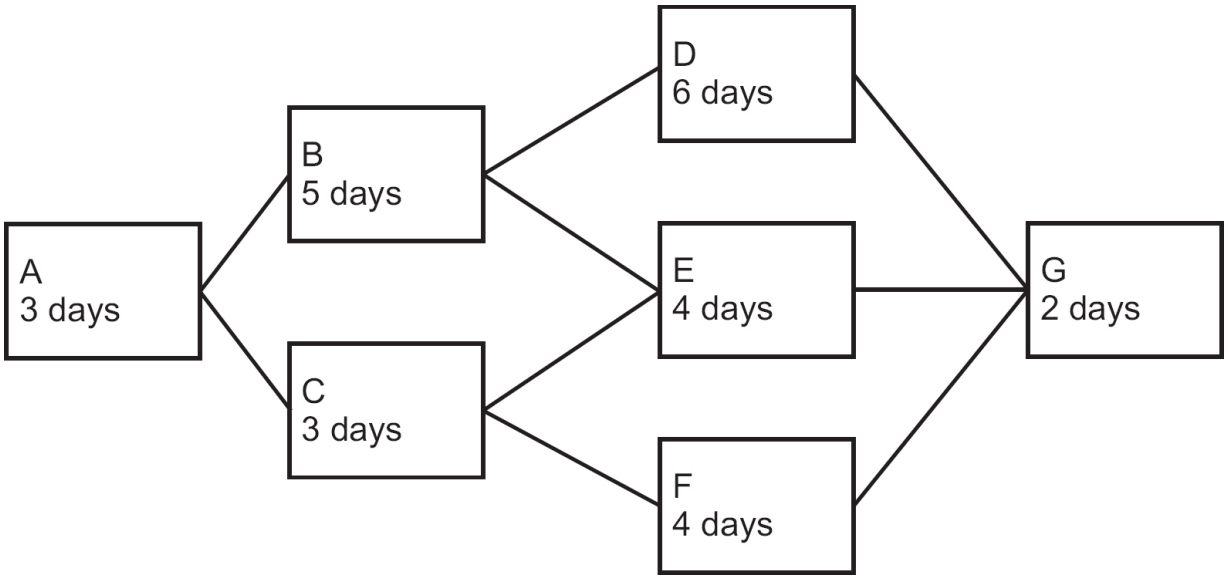
In this chapter it has been assumed that all the planning, and calculation of the consequences of particular planning decisions, will be done by hand with no computer assistance. It will be a relief for most people to know that there are software packages that will carry out most of the calculations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

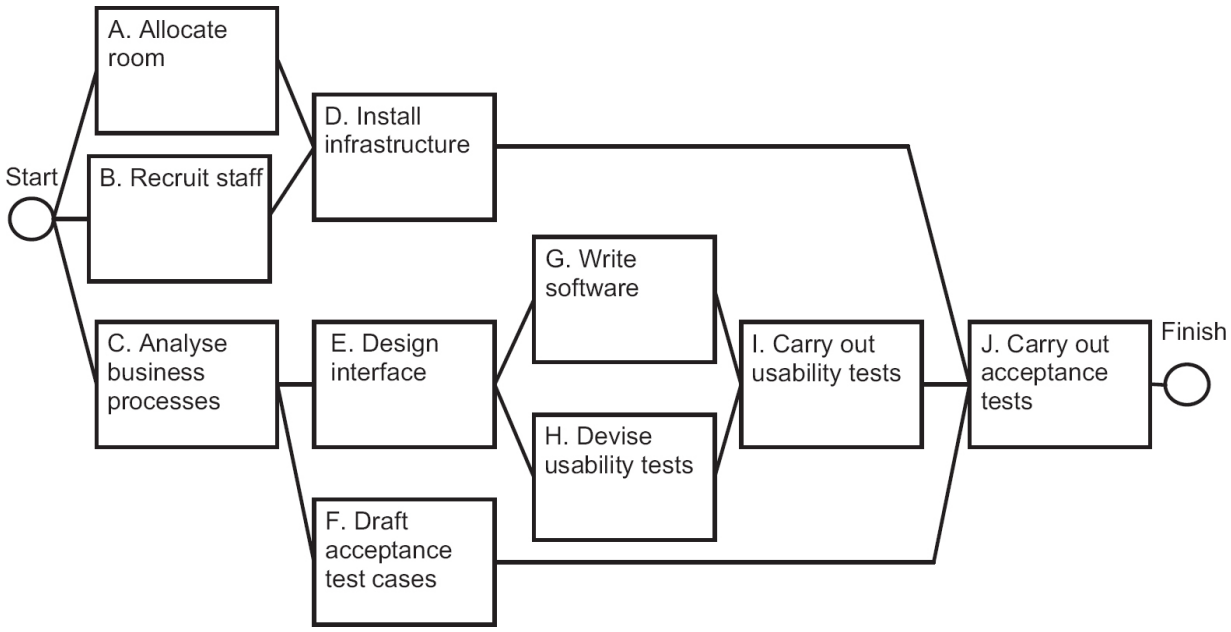
a. test results, error reports

b. trained users

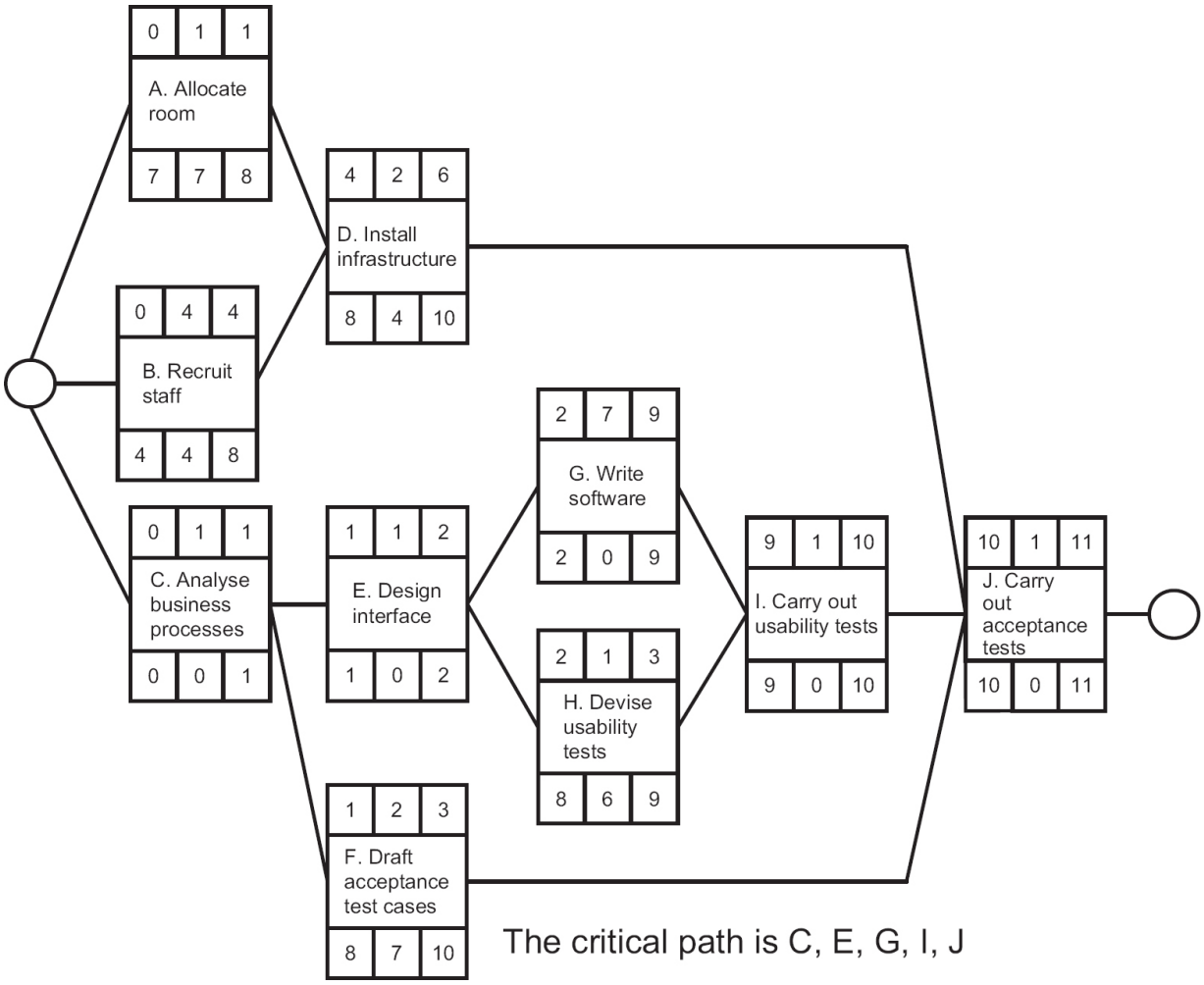
c. a new network

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

Activity 2.2


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

Activity 2.3



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.

This chapter explores the means by which a project is monitored and controlled so that it broadly fulfils its plan. The mechanism for this is the project control cycle.

3.2 THE PROJECT CONTROL CYCLE

The project control cycle involves the following sequence of steps:

a. producing a plan for the project to follow;
b. monitoring progress by collecting information about project performance;
c. comparing actual progress with the planned progress;
d. identifying variations from the plan;
e. applying corrective action if necessary.

Corrective action would usually involve changes to parts of the plan. These changes would have to be communicated to the project team and, where necessary, to stakeholders who might be affected by the changes.

Steps (b) to (e) are repeated to continue the **control cycle**, until the project is completed or abandoned.

Imagine a ship's voyage across the Channel from Dover to Calais as your project. The plan would involve following a certain route, aiming to arrive in Calais at a certain time. As the voyage progressed, the navigator would check the ship's progress against the planned course. If there was a difference, he or she could then decide that a change of speed or an alteration of course was necessary – this would be **corrective action**. The process would, of course, continue

until the ship arrived at its destination. Without this control cycle, the ship could continue on a fixed course and speed, and would be very unlikely to arrive at the planned destination or at the expected arrival time.

3.3 MONITORING PROGRESS

Monitoring progress is less straightforward in an IT project than in the ship example. The first question, which we tackle in the next section, is how to identify things that should be monitored. We usually know what the final objective of the project is, but how do we know how well we are progressing towards that objective?

3.3.1 What should we monitor?

The most obvious thing to monitor is the **progress** in creating **deliverables** and other, **intermediate**, project products, and in reaching **milestones** or **deadlines**. Difficulties arise when you want to monitor progress of things that are partially complete. The simple solution is to break the products and deliverables into smaller components that can be assessed as complete at shorter and more frequent intervals of time – for example, software can be broken down into smaller, relatively self-contained modules.

Where this is difficult, an alternative is to assess the percentage completion of an activity or deliverable. This can be problematic. If someone is building a wall, it is easy to see when it is half finished, but, particularly in the case of software, most project products are less obviously visible than with a wall.

The project manager often finds that an activity which appears to be completed has in fact delivered a defective product that requires the

activity to be reopened to carry out remedial work, which delays project progress. Hence, project control depends on effective **quality processes** that check the quality of the methods used to carry out each activity and the quality of the deliverables of each activity. This is covered in more detail in Chapter 5 on quality issues.

In Chapter 6, we describe **size** or **effort drivers**. These allow us to measure the size of the job to be done. In the case of building the wall, the number of bricks would be an obvious size driver: the bigger the wall, the more bricks it will need. The size/effort driver can be used to monitor progress. For example, if we know that the bricklayer will need to lay 200 bricks to build the wall but only 50 have been laid so far, then we can assume that the job is about 25 per cent complete.

In the end, the project's deliverables need to be useful to the people who will have to interact with them. The deliverables also need to enable the hoped-for **benefits** that motivated the project sponsors to invest in the project. The project will have been planned with this in mind. During the implementation of the project, changes may be made – such as reducing the functionality to be produced – and the impact of this on the benefits of the project must be assessed. This will need the approval of the sponsor.

The **use of resources** also needs to be monitored, which in IT projects are mostly 'human resources' or staff time. Also, **financial expenditure** should be carefully monitored. In the scenario in Activity 3.1 below, allowing the installer to stay in an hotel between installations in the same region may save on travelling time (and fuel costs) and speed up the installation rate, but it would need to be balanced against the additional cost of accommodation. Surprisingly,

however, financial expenditure on human resources is not always strictly monitored in IT projects if the project team are permanent employees in an IT department and therefore viewed as overheads.

ACTIVITY 3.1

There are 20 boatyards and marinas where Water Holiday Company customers collect and return their hired boats. As part of the new integrated booking system, online customers will be emailed an e-ticket, containing a barcode, which they will be expected to present at the relevant marina at the start of their holiday with evidence of their identity. The e-ticket requires new IT equipment to be installed at each boatyard/marina, and some additional training will be needed in other features of the new system – for example, recording the non-availability of boats for maintenance reasons.

It has been estimated that the installer will, on average, need a day to travel to a marina, install the new equipment and show local staff how it is used. Twenty days (or four working weeks) have been allocated for the installation of all the equipment.

However, at the end of the first week only three marinas have in fact been visited.

a. How long is it likely that the installation programme will now take?

b. What difference to the figure you have produced in (a) might be made by the following circumstances?

   - The installer started two days late because some items of equipment had not been delivered.

- The installer started with the marinas furthest afield, and needed extra time to travel to the area and back.

As well as **scope** – essentially the amount of functionality being produced – being reduced to meet a deadline, functionality to be delivered could increase because new requirements are discovered. If these additions to the work are not monitored and controlled, costs and delivery time will be affected.

Thus time, cost and the scope of deliverables need to be balanced. For example, it may be possible to accelerate the progress of a late project by employing more staff, but this would increase the project cost. On the other hand, it may be possible to meet the deadline within the budgeted cost by reducing features in the application to be delivered – see Section 1.7.2, where timeboxing was described.

The systems that bring these different types of project information together for consideration are often referred to as **dashboards**.

3.3.2 How should we monitor?

Monitoring involves collecting information about actual project progress. This enables the comparison of actual project performance with what was envisaged in a plan. **Formal monitoring methods** include the use of written reports, email and progress meetings. The frequency, format and content of these communications should be laid down at the start of a project in the **project management plan** (see Section 1.8.1).

Formal monitoring establishes routines so that people periodically focus on progress and commit themselves in writing. However,

preparing reports can be seen as an unproductive overhead. Staff need to be convinced of its value. Thus, **timesheets** can be effective in establishing the staff effort expended on distinct aspects of projects, but staff need to be persuaded to fill them in conscientiously.

Many phrases can describe **informal monitoring**: keeping one's ear to the ground; management by walking about; open door policy. All these make the manager aware of what team members are experiencing. Project managers need ways of maintaining good informal lines of communication with all project staff. This often allows problems to be resolved before they would otherwise appear in a progress report. However, a pitfall to avoid is the alienation of team members by over-supervision.

3.4 APPLYING CONTROL

There is no point in monitoring without **control**. This is done through the **reporting cycle**. Monitoring processes identify shortfalls in the progress of the project, that is **variances** between what has been planned and what has actually happened. To remedy variances, control needs to be applied to the project to bring it back on course. For example, staff might be transferred from non-critical work to critical activities that have fallen behind.

The reporting cycle defined in the project management plan identifies who should be producing progress reports, with what frequency and to whom they are sent. Remember that reporting is an overhead. Reports should therefore be concise, relevant and circulated only to those who need them. However, more concise reports require greater effort by the writer in order to save the time of

the readers. As someone once apologised: '*I am sorry this report is so long; I didn't have time to write a shorter one*.'

The **reporting structure** refers to the people involved in a project at various management levels. Generally, progress reporting starts at the level at which work is actually done and progresses up through a hierarchy.

In an IT context, team leaders gather progress information from their team and report up to their project manager. This may be done directly or through a project support office or other intermediary. The project manager then reports to the person or group that has been entrusted with overall responsibility for the project. This might be a project sponsor or executive supported by a project board, project management board or steering committee. This group could include representatives of the managers of the development team and the users as well as the project sponsor, who, as the provider of finance, inevitably has a decisive influence. They, not the project manager, have the authority to change the objectives of the project. They could allocate more resources to the project or reduce the scope of deliverables.

The report to the project board or steering committee is sometimes referred to as a **highlight report**. The intervals at which the reports are produced and the topics they report should meet the needs of the recipients and the importance of the information conveyed. It is important to obtain formal agreement with the reporting procedures laid down in the project management plan from all the major parties involved.

3.5 PURPOSE AND TYPES OF REPORTING

As far as the project manager is concerned, there are two aspects of reporting: the reports they receive from those who actually do the work, and the reports they write conveying progress information to higher management, including, most importantly, project sponsors. Reporting does not have to involve physical meetings, especially where projects are geographically dispersed.

3.5.1 Team meetings

The team leader might be able to obtain the progress information they need from team members without having a group meeting. However, meetings are not just about a team leader communicating with individual team members, but team members talking to one another. Where obstacles to progress have to be removed, a meeting can be a very efficient tool.

These are usually attended by team members, the team leader and possibly the project manager. On small projects with a single team, the project manager and team leader roles may be merged. A weekly frequency is usually appropriate (in some Agile projects there may even be daily ‘stand-up’ meetings). A report from the team leader to the project manager will be prepared. A typical agenda would include the following:

- each individual team member’s progress against their plans;
- reasons for variances;
- expected progress – which looks forward to what each team member is going to do;
- current problems or **issues**;

- possible future problems – which may involve reviewing the risks recorded in the project risk register (see Chapter 7) that could affect this part of the project.

It is important that all those attending have a reason for attending and a contribution to make. These meetings are often referred to as **checkpoint** meetings and the progress report produced in this case is a **checkpoint report**. (In an Agile project, a **backlog** list identifying tasks completed and those still to be done would be updated.) As issues are identified, they may be recorded in an **issues log**, which will be updated as they are resolved.

The project manager receives the reports from team leaders and produces a summary report (sometimes called a **highlight** report) for the sponsors of the project. Where there are a number of teams, each with its own leader working in parallel on a project, the project manager may well hold project coordination meetings with the team leaders to review and remove obstacles to project progress.

The summary report typically includes the following information:

- details of the progress of the project against the plan;
- current milestones achieved;
- deliverables completed;
- resource usage;
- reasons for any deviation from the plan;
- new issues and unresolved issues;
- changes to risk assessments;
- plans for the next period and products to be delivered;

- graphical representations of progress information.

3.5.2 Project sponsor meetings

You will recall from Chapter 1 that the project sponsor is the individual responsible for safeguarding the interests of the client. The project sponsor might work through meetings of a steering committee or project management board. These meetings will be attended by designated representatives of users, suppliers and other stakeholders, with the project manager in attendance and with secretarial support perhaps provided by a project support office or project management office. The structure and responsibilities of the various roles in this structure are covered in Chapter 8.

The frequency of meetings will have been agreed and recorded in the project management plan. The exact timing depends on the project size: larger projects may have fewer and less frequent top-level meetings, but more meetings of managers at intermediate levels. Meetings can be timed to coincide with significant project events such as the completion of a particular project phase or stage (that is, a milestone) or other, external, triggers such as requirements for financial approvals.

Items for the agenda are similar to those for team meetings. The summary report described above, from the project manager to the board, is circulated prior to the meeting. The board is authorised to decide upon any necessary corrective action arising from progress information. This is fed back down the reporting chain and thus completes the reporting cycle.

3.5.3 Programme board/steering committee meetings

Organisations sometimes group projects into **programmes**, where a number of projects all contribute to a set of over-arching objectives (see Chapter 8). In these cases a programme board may be set up, to which individual project boards would report. These boards would have less frequent, less detailed meetings related to programme management, but with essentially a similar agenda to those of the project boards. These would have more of a business focus than a project focus.

3.6 TAKING CORRECTIVE ACTION

Here we will examine how corrective action can be applied in a controlled way. The project manager's role is to manage on a day-to-day basis, applying minor corrections as required. However, corrections of a more major nature will need to be referred to higher authority.

This is the reason for allocating a **tolerance** within which the project manager has authority to make changes or apply corrective action. For example, you, as a project manager, may be allocated 10 per cent tolerance on time and cost on a project worth £100,000 and lasting 25 weeks. This means you are authorised to agree to changes incurring additional costs up to £10,000 or an overrun of two and a half weeks (see Chapter 4 on change control). If a situation occurred in which the project was expected to overrun by more than two and a half weeks, this would be an **exception**. In this case, you would need to prepare an **exception report** for submission to a special meeting of the project board, or equivalent. This exception report would describe a number of **options** designed to correct the overrun and the board would decide how to proceed.

Tolerance and **contingency** pools are sometimes distinguished. Tolerances can be assigned to individual activities within the project. The contingency pool is a set of resources that is controlled by the project manager and can be allocated at the discretion of the project manager wherever additional resources are needed. However, if the contingency is used to buy resources in one place, less will be available for other emergencies. These resources may be added to where activities are completed early and developer time is freed up. Where activities can be completed early, the opportunity should be seized as this will release staff to deal with unanticipated delays elsewhere.

An exception report typically includes the following information:

- background;
- reasons why the exception arose;
- options;
- risks;
- **exception plans** showing how the project plans need to be amended in order to implement the suggested options;
- amended business case;
- recommendations.

When members of the project board (or equivalent) scrutinise the exception report, they will be particularly concerned to ensure that the business case for the project will be preserved: that is, that the costs of the project will not exceed the value of its eventual benefits. If the board are satisfied with the exception report, the project manager is given authority to proceed using the chosen option and

its associated exception plan. Chapter 4, on change, describes an alternative approach where changes need to be made to systems that are already operational.

Where monitoring reveals a shortfall in the expected progress, control is applied to bring the project back on track. There are a number of standard control strategies, which may or may not require an exception report. These are considered below.

3.6.1 Work harder, longer or faster

This may work to solve a short-term problem or to meet a critical deadline. It will fail if over-used. Staff will become tired, stressed and then demotivated. If overtime is paid, then project costs increase, but not by as much as with the next option.

3.6.2 Increase resources

Resources in this context mean people. Adding more staff can increase the amount of work done, but usually decreases productivity. The introduction of more staff involves a period of induction while they familiarise themselves with the work. The current staff may be involved in this process and the overall effect could be to delay progress. An exception report and plan are needed if the additional costs go beyond the tolerances agreed for the project.

3.6.3 Re-plan

Although some project activities have consumed more staff effort or taken longer than planned, others may have taken less effort or time. Internal movement of staff from tasks completed early can make up for delays elsewhere without extra cost. Some may unkindly attribute

this to poor planning, but the truth is that there will always be uncertainty about exactly how long tasks will take.

3.6.4 Extend the time scale

Changes to deadlines need negotiation, usually through the exception reporting process described above. Extending the deadline is often seen as weak management or as allowing the project to get out of control, but can be the most sensible option. It increases costs, as staff are allocated to the project for longer; however, sometimes the reason a project is late is that it has not been possible to assign all the staff originally planned, and so budget may not be a problem. The negative impact of delay can be reduced by delivering more valuable functionality on time (or even earlier) while postponing less valuable functions.

3.6.5 Reduce the project scope

This is an attractive option which also needs negotiation with management and drafting of an exception plan. Deliverables may be removed from the plan or delayed until after the planned project end date. This does not affect the originally planned cost or duration of the project, but the value of benefits to the user may be reduced. As noted in Chapter 1, this is the preferred solution of some Agile project management approaches.

3.6.6 Terminate the project

If no acceptable alternative can be found, this may be the only remaining sensible action. Terminating the project would be justified if it is clear that the remaining costs of the project will exceed the

projected value of its benefits when delivered. Despite this, terminating the project may be politically unacceptable.

ACTIVITY 3.2

In the Water Holiday Company integration project, activity G, 'Write software' (see Figure 2.11), has been outsourced to an external software development company, XYZ. XYZ find that the task 'Code provisional booking function' is going to take five weeks rather than four. Their contract with the Water Holiday Company states that they will have to pay a penalty of £500 for each week of delay in delivering the software.

The options considered by XYZ are:

a. Be a week late and pay the penalty.

b. Split the 'Code provisional booking function' into two subcomponents requiring three weeks of work each and bring in an extra developer to work in parallel with the one currently assigned to this function. Software developers cost £400 a week.

c. Get the Water Holiday Company to accept a delivered system on time but with some functionality missing. The supplier will agree to provide updates to implement the missing functionality at no extra charge, although this will require an additional week of coding work at a later date.

Which would be the most cost-effective option for the software supplier?

3.7 GRAPHICAL REPRESENTATION OF PROGRESS INFORMATION

In Chapter 2 we produced a Gantt chart showing the activities needed in a project and the relationships between those activities. It also showed the resources – mainly developers – allocated to each activity. In some cases an activity could be delayed as a developer was not available to do the work. When the project is put into action, activities may in fact be longer or shorter than planned, and the staff available may change. The question tackled here is how project managers can get a picture of the current state of this complex ever-changing situation so that they can intervene to keep it on target.

3.7.1 Gantt chart

Progress information can be shown on a Gantt chart by putting a bar through an activity box showing the estimated percentage completion. Figure 3.1 shows the Gantt chart that was produced in Chapter 2 (Figure 2.13), updated to show the actual situation at the end of week 3 of the project. At this point the following information could be reported:

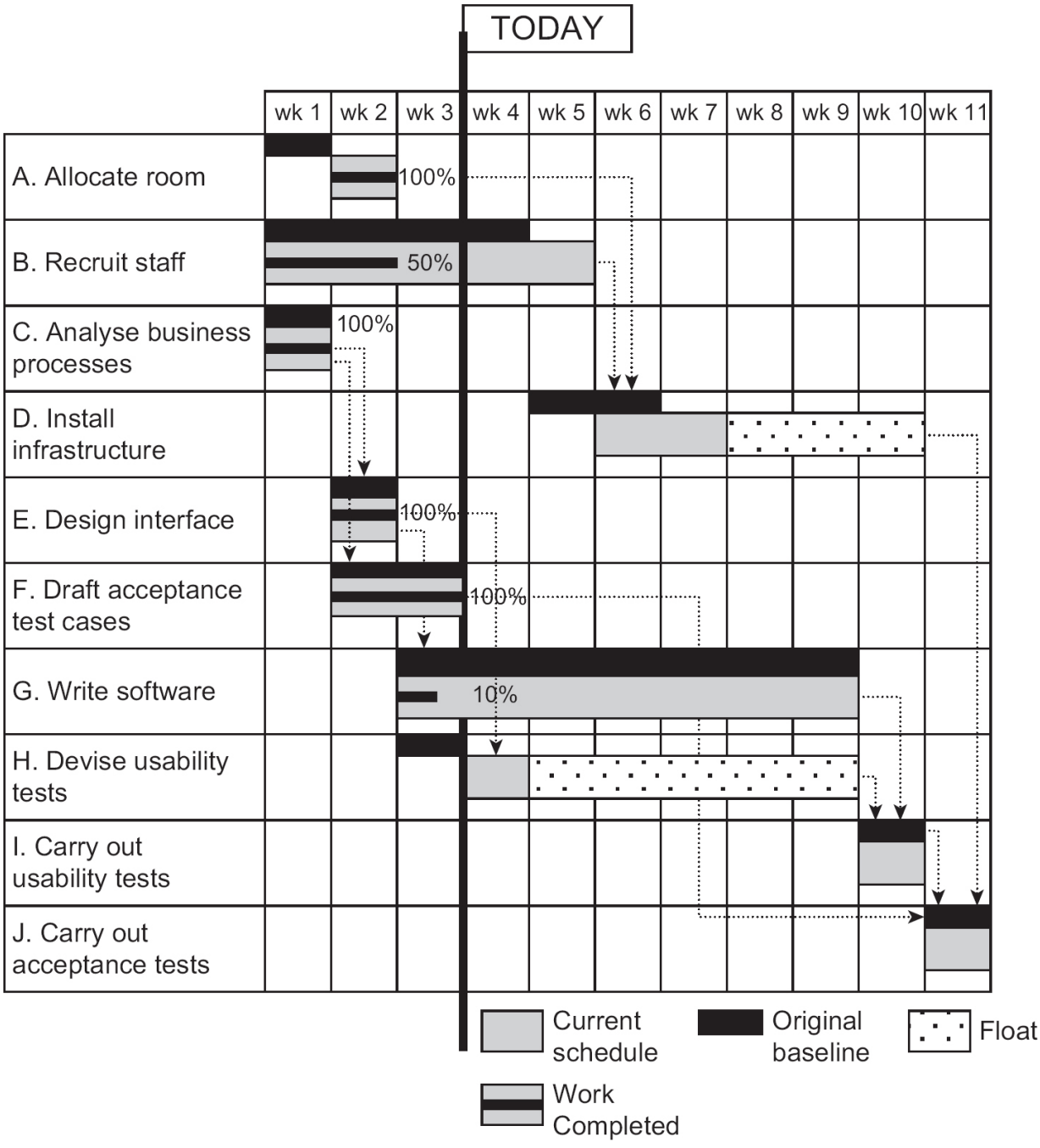
Activity reference	Name	Progress
A	Allocate room	Started a week late, completed a week later
B	Recruit staff	Reported as 50% complete, but should be more like 75% at this point. An extra day is added to planned duration

C	Analyse business processes	Completed on time
D	Install infrastructure	This started a week late because of the late finish of activity A
E	Design interface	Completed on time
F	Draft acceptance test cases	Completed on time
G	Write software	10% completed
H	Devise usability tests	Start delayed by one week

We need to compare the current situation with the original plan. We said above that the project is always in a state of flux. To cope with this, we take a snapshot of the details on the Gantt chart. We call this a **baseline**. There can be several of these, but an important baseline will be the agreed schedule at the start of the project, before any work. This is shown in Figure 3.1 by the black boxes. The copy of the baseline is then marked up to show actual progress; for instance, to show that task A should have started in week 1, but actually started in week 2. We now have a baseline plus the changes to the baseline.

To reduce the amount of detail, at an agreed point, we can create a new baseline of the Gantt chart where all the activities are shown as starting and finishing on the new times, and the fact that these are changes is removed. This is effectively a new plan that can be used to record progress for the next period of the project.

Figure 3.1 A Gantt chart that has been updated with actual progress



ACTIVITY 3.3

In week 4, the following actions take place:

a. Activity B. Recruit staff. There have been difficulties with finding qualified staff and effectively no progress has been made. It is reported that two further weeks, in addition to those already scheduled, will be needed.

b. Activity G. Write software. There is a discrepancy in the requirements which means that progress has been held up for the week. Currently, it is hoped that the developers will be able to catch up over the next few weeks.

c. Activity H. Devise usability tests. This has now been completed.

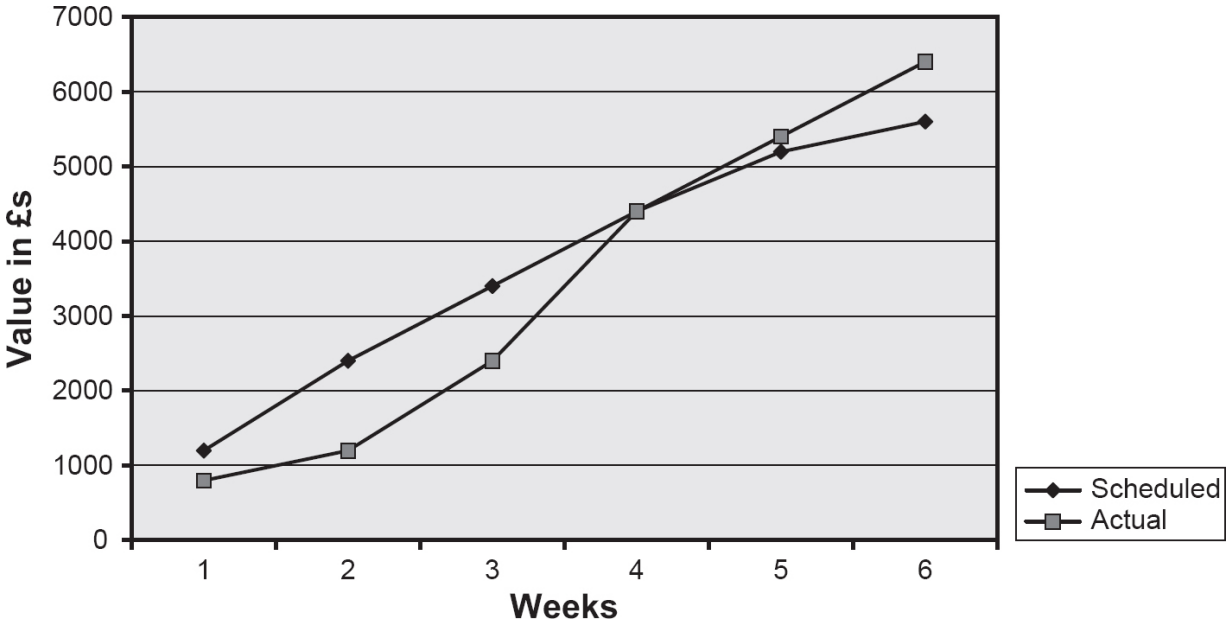
Update the Gantt chart in Figure 3.1 to take account of these changes.

3.7.2 Cumulative resource chart

A cumulative resource chart can be used to present resource usage details (see Figure 3.2). This is sometimes called an **S-curve chart**, because the pattern of activity is that a project starts with a relatively low number of staff at the planning and requirements gathering stage. As the project progresses, more and more staff are needed as the development and implementation activities multiply. The demand for staff then decreases as the implementation proceeds – for example, software developers are not needed full-time once they have constructed their software components. This pattern of activity – a low demand for staff time which gradually builds up and then declines – leads to a cumulative resource chart where the line seems to have an approximate S-shape.

These charts normally have two sets of data points: one showing the expenditure that was planned and the other the actual expenditure for comparison.

Figure 3.2 A cumulative resource chart



This is a convenient visual representation of project progress suitable to show to management.

Figure 3.2 shows that for most of the project we were under-spending, but there has recently been a surge in expenditure, so that currently we have spent more than planned. However, we do not know whether this is due to poor productivity or whether we have actually produced more than was scheduled – work may have been completed early, leading to some expenditure also being incurred earlier.

The traditional S-curve chart does not show any of the following:

- whether the project is ahead or behind schedule;
- whether the project is getting value for money;
- whether problems are over or just beginning.

3.7.3 Earned value analysis

If we plot a third line, the earned value, then we can see if we are ahead or behind time, and above or below budget. **Earned value analysis** (**EVA**) shows the budget that was originally allocated to items of work that have been completed. When the work is finished, we can say that this value has been 'earned'. If an external supplier is involved and had a contract with fixed prices to be paid on delivery of various products, they would see these payments as earned value.

As an example, activity G, '*Write software*', is made up of a number of tasks that can be seen in Figure 2.11. To complete the overall activity within seven weeks, the supplier needs to have completed all the design work by the end of the second week of activity and all coding by the end of the sixth week. If the designers are priced at £400 a week, then at the end of the second week, four staff weeks of design work should have been completed. This means there is a **planned value** (PV) of £400 × 4 (that is, £1,600) at the end of the second week. Now if, for example, the task GA '*Design provisional booking function*' is **not** completed, because that was originally planned as two weeks' work, the **earned value** (EV) at this point in time would be only £400 × 2 for the work completed in tasks GC and GE (that is, £800).

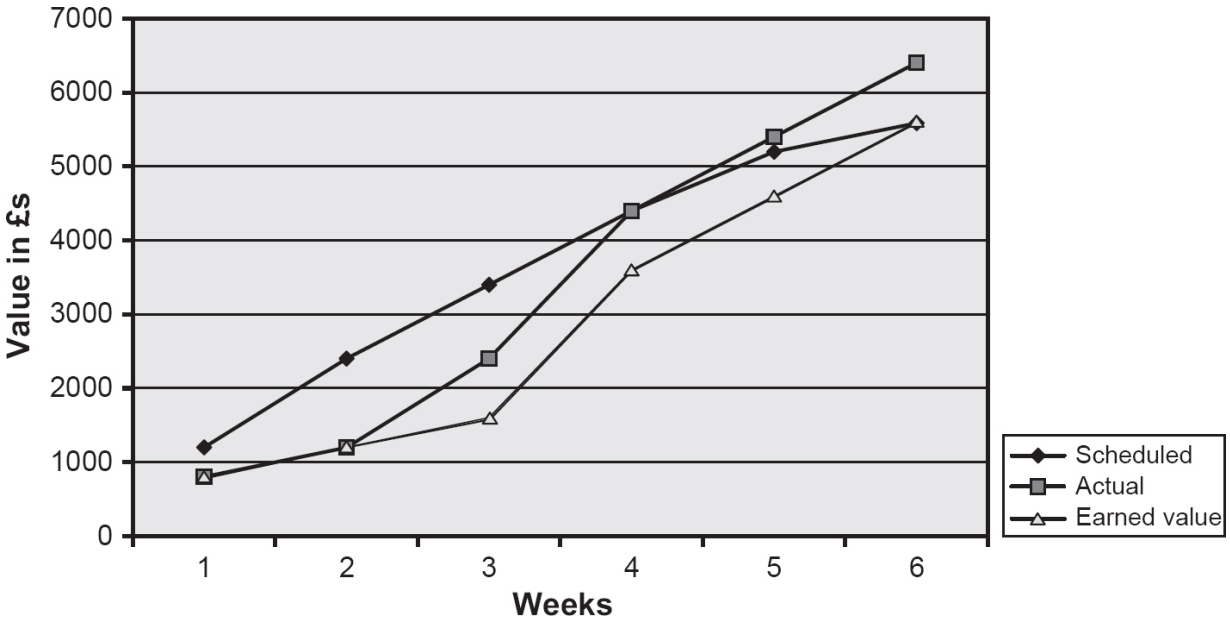
Two ratios provide guidance on the progress of the project. The **schedule performance indicator** (SPI) is calculated by dividing the

current EV by the corresponding PV. In the example above, this is £800/£1600, that is, 0.5. A ratio value below 1.00 means the project is behind schedule. **The cost performance indicator** (CPI) is the EV divided by the actual cost (AC) of the work completed. The £800 value earned by the completion of tasks GC and GE costs £800, so EV/AC is 1.00. This means that the costs are exactly those expected for the work done. An index value greater than 1.00 shows work costing less than expected, and a value less than 1.00 that there is a cost overrun.

In Figure 3.3, a line showing earned value has been added to the cumulative resource graph. This shows, for example, that at the end of week 6, the project is on schedule and has completed the work that was planned. However, it can be seen that expenditure is greater than planned. This may be a case where the project manager got the project back on schedule by buying in overtime.

Please note that candidates for the BCS Foundation Certificate in IS Project Management are not expected to know the details of how earned value is calculated.

Figure 3.3 Earned value graph



SAMPLE QUESTIONS

1. **Which of the following would you most expect to see in a routine report from a project manager to a project board?**

   a. Costs and benefits

   b. Progress against plan

   c. Configuration status information

   d. Project staff appraisals

2. **Which of the following is NOT involved in collecting progress information?**

   a. Team progress meetings

   b. Timesheets

   c. Comparing planned and actual costs

   d. Informal monitoring

3. **Which of the following would be most likely to give rise to an exception report?**

   a. A new issue being raised

   b. A proposal to make a change to a deliverable

   c. The unexpected loss of a key team member

   d. Project tolerance being exceeded

4. **What is the purpose of earned value analysis?**

   a. Assessing progress

   b. Collecting progress information

   c. Estimating the required effort

   d. Calculating benefits

POINTERS FOR ACTIVITIES

Activity 3.1

a. If we keep to the original planned installation rate of one marina/boatyard a day, 17 days (that is, about three to four weeks) are needed to complete installation, as there are 17 boatyards and marinas left. However, if we decide that the experience of the first week shows that the original installation rate was unrealistic, then we may project that three boatyards can be visited in a week. This would lead to between five and six weeks being needed to complete installation.

b. In the first scenario, the reason for only three boatyards being visited was simply a late start rather than a low installation

rate. The remaining estimate of 17 days would seem to be reasonable.

In the second scenario, the reason for lateness was a lower installation rate. However, as the installation programme proceeds, the installation rate should improve as the nearer marinas and boatyards are dealt with, and journey times get shorter. Revising the planned time would be premature at this point.

This activity should illustrate how informal information gathering can help interpret more formal reports and the risk of extra learning time being needed if a new developer is added to the team who is unfamiliar with the application.

Activity 3.2

a. Option: be a week late and accept the penalty. The penalty would be £500, but there would also be the cost of an extra week's work (£400). This would cost £400 + £500 in all – that is, £900.

b. Option: split the functionality into two components that can be developed in parallel. Currently £1,600 (4 × £400) has been allocated to the task. The new plan would increase this to £2,400 (6 × £400) – that is, an overall increase of £800.

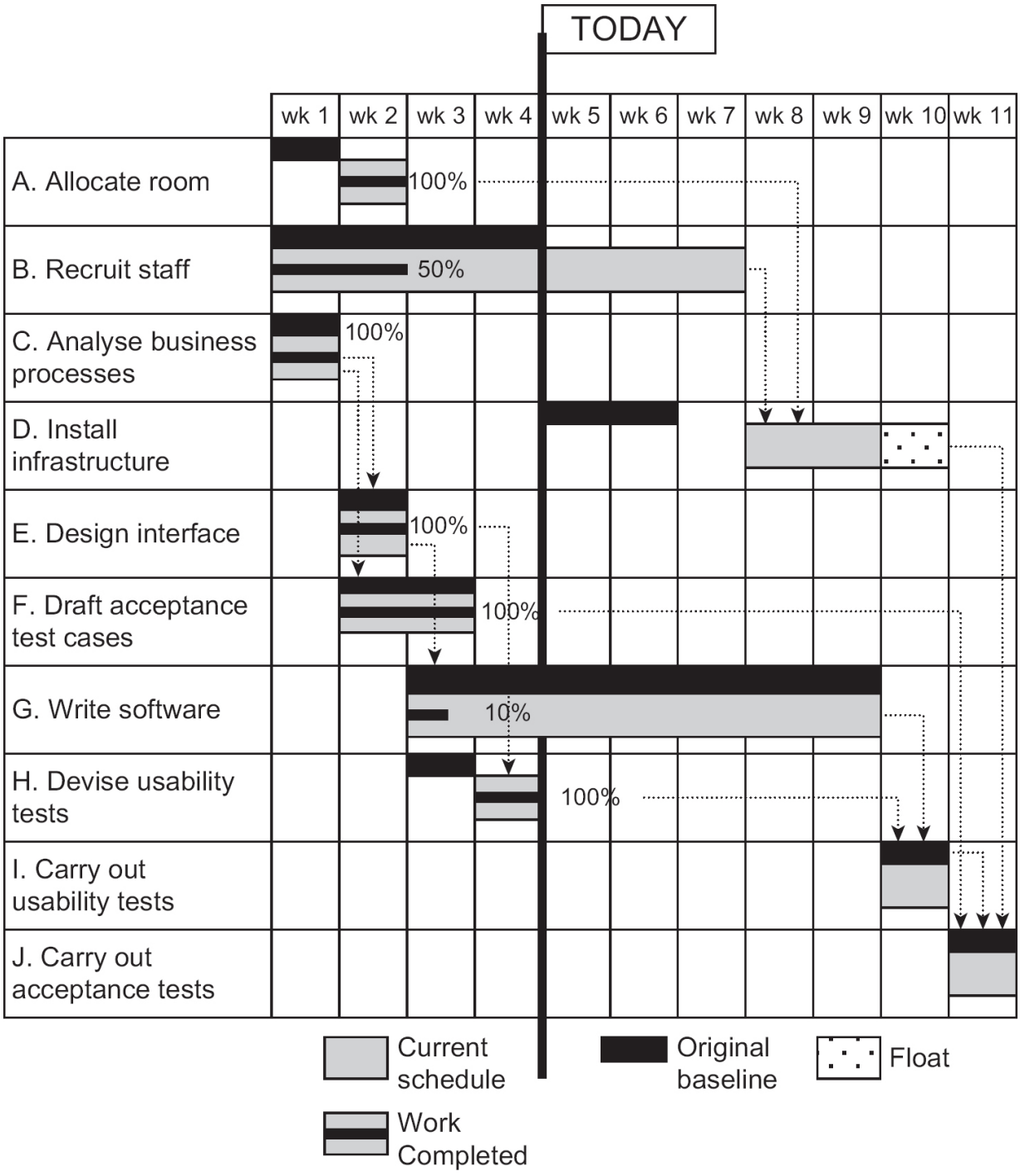
c. Option: staggering delivery. There would be an extra week's work for the delayed enhancement. This would cost £400.

Thus option (c) would appear to be the best. Note that this is a very simplified scenario, and does not take into account many issues such as the risk of possible loss of reputation with some of the options.

Activity 3.3

See Figure 3.4.

Figure 3.4 A Gantt chart that has been updated with actual progress up to week 4


TODAY
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
100%
B. Recruit staff
50%
C. Analyse business processes
100%
D. Install infrastructure
E. Design interface
100%
F. Draft acceptance test cases
100%
G. Write software
10%
H. Devise usability tests
100%
I. Carry out usability tests
J. Carry out acceptance tests
Current schedule
Original baseline
Float
Work Completed

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 at a price they can afford. In order to achieve this, quality needs to be built into a product. It cannot be added to a product after it has been created – except with great difficulty and cost. It is like trying to alter the foundations of a building once it is complete. To build in quality requires a commitment from all parties to the project, from the project sponsor through all the levels of management to the technical staff, customers, users and the clerical support staff. Philip Crosby's seminal book on quality opened:

> Quality is free. It's not a gift, but it is free. What costs money are the unquality things – all the actions that involve not doing the job right in the first place.
>
> (Philip B. Crosby, *Quality is Free*, New York: Mentor, 1980)

Crosby was not arguing that increasing the quality of a product did not cost money; rather, that the costs of remedying lapses in quality would be even greater. There is a relationship between quality, cost and time. A higher level of quality can increase the duration of a project and/or its cost. This is called the **cost of conformance**. But the more effort goes into quality, the less expenditure is needed on correcting and compensating for faults at the end of the project – the **cost of non-conformance**. The project manager must balance the two types of cost.

It is important to distinguish between **quality control** and **quality assurance**. Quality control is concerned with the practical activities that check the quality of a deliverable or intermediate product, for example that manufactured light bulbs actually work. Assurance activities do not check product quality directly but instead check that

the required steps that establish product quality are in place and have been carried out. This might check, for example, that **process standards** belonging to the activity have been followed, such as a checking process for the purity of raw materials used in a process. This could involve examining evidence such as testing plans and sign-off documents. Section 5.4 explores this in more detail.

5.2 DEFINITIONS OF QUALITY

A definition of quality is *a degree or level of excellence*, as in the phrase ‘high-quality goods’. This definition is subjective: for example, when comparing cars, people argue about the quality of different makes.

Another definition is *conformance to standard*. Within a project process there will be certain standards to which developers are expected to conform. These standards help reassure the project’s clients that they will get value for money for the project’s products. However, the generally accepted definition which should apply to all projects is that the deliverables should be ‘*fit for purpose*’. Again referring to cars: if you need to get to work on time, a Rolls-Royce may not be the best vehicle to navigate through heavy traffic – a scooter might be more effective. The original international standard on quality, ISO 8402:1994, formally defined quality as ‘*the totality of features and characteristics of a product or service which bear on its ability to satisfy stated or implied needs*’.

One aspect of this ‘fit for purpose’ definition, reliability, shows how the concept applies. If the Water Holiday Company’s booking system is out of action for a time, it would be annoying but not life threatening. However, if the control systems in an aircraft in flight fail,

that would be disastrous. Hence, the effort devoted to making sure that the aircraft system does not fail would be greater than that required for the Water Holiday Company integration project. The required quality varies depending on the type of system under development and the money the customer is prepared to pay.

Those responsible technically for a project must advise the customer on the benefits of a well-engineered system, but the person paying usually calls the tune. However, suppliers also have a professional and legal commitment to the general public to ensure that the systems they produce are safe. The possibility of an aircraft crashing in an urban area makes us all stakeholders in aircraft safety.

As IT systems become more complex, it is impossible to ensure that they will never fail. Thus, it is important to examine the ways in which the systems under development will behave in the event of various types of failure. This will directly influence the development process and how quality is measured. For example, a control system for a nuclear power plant will be designed to ‘fail safe’ – that is, in the event of a systems failure it will revert to a safe state, for instance by closing down. Obviously, such a requirement must be specified in the quality criteria for the system. Where something is inherently dangerous, as in this example of the nuclear power plant, the developers would have a duty of care, not just to the project sponsor, but to the world at large. It is also likely that governments, either individually or collectively, will enact legal requirements in such circumstances with which there must be **compliance**.

The testing of deliverables for required qualities needs the deliverable to actually exist. This is rather late in the development life

cycle. It is therefore useful to identify **process standards** governing how products are created that can be applied at an earlier stage.

5.3 QUALITY CHARACTERISTICS

The definition of '*fitness for purpose*' needs elaborating so that it can be applied practically. Another international standard, SQuaRE (standing for Software Quality Requirements and Evaluation), is documented in the ISO 25000 series of standards and defines a set of standard characteristics by which software quality can be measured. (This has replaced a previous version, ISO 9126.) It specifies the following high-level quality characteristics:

- **Functional suitability:** Does the system as delivered meet the functional requirements of the user? Meeting user expectations is more than just meeting a specification.
- **Performance efficiency:** For example, how fast does the system execute functions? What hardware resources does it need? How many users can access the system at one time?
- **Compatibility:** Can the new system operate with existing systems? In particular, can it share information with other applications?
- **Usability:** Is the system straightforward to use? How much training is required for someone to use it?
- **Reliability:** How often does the system break down? How long does it take to put right? In general, software that has been in use by a large number of people over a long period of time is likely to be reliable. This is because over time most of the faults

will have been detected and corrected. Hence the use of the term '**software maturity**'.

- **Maintainability:** Can this software application be modified easily and without introducing unexpected errors?
- **Portability:** how easy is it to move the software from the particular platform on which it was developed to another environment?

The standard itself goes into much more detail for each heading. The top-level qualities are broken into sub-qualities that contribute to the higher-level quality.

Not all qualities are relevant for all projects and project managers need to select those useful for their projects. For example, in the Water Holiday Company booking system, response time in answering queries about the availability of boats is an important part of usability. For the application, engineering measurements have to be mapped to values on a scale reflecting user satisfaction – for example, a response time under five seconds might correspond to 'acceptable'.

5.4 QUALITY CRITERIA

The level of product quality required has to be specified at the beginning, before the product is developed. In Chapter 2, the concept of a **product definition** or **description** was introduced. Each product definition included a section headed **quality criteria**. These criteria enable checks that a product is fit for its purpose in the final deliverable, or in the creation of other products, by seeing whether it meets the quality criteria specified. The product definition

refers to types of product, not specific instances. For example, for a software module, processing test data correctly could be a quality criterion, but the actual test data will be based on the particular module's specification. Product definitions are produced at the start of projects and specifications will elaborate and extend the initial criteria.

Effective quality criteria must be **specific**, **measurable** and **achievable**.

As an example, the aircraft control software could have a specific requirement that the product must never fail. It is measurable by running the product continuously until it fails. Unfortunately, it can only be demonstrated that the requirement has not been met. We can never prove that it will not fail at some point in the future. Hence the 'never' requirement is not achievable. We can, however, provide an estimate of the probability of failure which will get smaller as testing intensifies. As noted with the nuclear power station, we can also plan for the possibility of failure, for example, by allowing a reversion to manual control.

ACTIVITY 5.1

Refer back to the discussion about product definitions (Section 2.2.1). Draw up quality criteria that can be used to assess a product definition (**not** the product it defines). Add any other headings that you consider should be standard for all product definitions/descriptions. Specify how the criteria can be measured.

5.5 QUALITY CONTROL VERSUS QUALITY ASSURANCE

We now discuss how the quality criteria of a product created by a project are checked. Ideally, the project takes place in an organisation committed to quality with standards already in place for certain activities. If this is not so, part of project set-up will be the creation of the framework for managing quality.

ACTIVITY 5.2

The following are examples of good and bad quality criteria:

- All screen layouts should have similar layouts and use the same terminology.
- Screens should be user friendly.
- The system should be able to handle 50 transactions.
- The system should allow for 20 users at any one time without degradation.
- The response time should not be longer than three seconds.

Comment on the effectiveness of each of these quality criteria.

ACTIVITY 5.3

Maintainability is defined as a quality characteristic. How can it be measured?

ACTIVITY 5.4

How can the reliability of a system be measured?

An organisation that carries out many similar projects can usefully develop organisational standards for product definitions and quality practices applicable, with appropriate adjustments, to all its project work. This quality framework is called the **quality management system** (**QMS**). It may be based on the ISO 9000 series of standards (see Section 5.10). Within the QMS, there will be a quality strategy and quality assurance processes. They are reviewed and, if necessary, modified to meet the specific requirements of each project.

A **quality strategy** defines the QMS and includes:

- procedures and standards for creating a **project quality plan**;
- definitions of **quality criteria**;
- quality **control** procedures;
- quality **assurance** procedures;
- a statement of compliance with or allowed deviation from **industry standards**;
- acceptance criteria;
- an allocation of **responsibilities** for defining or undertaking quality-related activities.

The methods for exercising quality control are discussed later, but generally these are either practical tests or a review. As seen in Section 5.1, quality assurance stands alongside quality control but is external to it. Quality assurance is an audit to confirm that proper

procedures are in place and applied correctly. The correctness of a software component can be verified by running tests and checking the results. That is control. Assurance would involve checking that the testing has actually been done by looking at evidence such as copies of the expected and actual results of testing.

Quality control would normally be an integral part of the project team's work. The quality assurance activities, on the other hand, are usually carried out by people outside the project team who report directly to the steering committee/project board. This separation of responsibilities helps to ensure that the process is transparent and reduces possible conflicts of interest.

Quality criteria are specified for each component. As each component is completed, a control process is undertaken to ensure that the criteria have been met. This is followed, at an appropriate time, by an assurance process which confirms that the agreed procedures have been followed and that all products have undergone the necessary checks.

You will recall from Chapter 1 that the systems development life cycle has a number of stages – for example, requirements analysis is followed by systems design, followed by construction and testing. Each stage contains quality control processes, usually reviews, and a quality assurance process takes place at the end of each stage. If proper quality control is found to be lacking, corrective action may be mandated. The quality control and, if necessary, the quality assurance processes are repeated until the quality criteria are met.

5.6 QUALITY PLANNING

The creation of a **quality plan** is vital to the success of a project. It specifies the particular standards that apply to the project. Ideally, they should be taken from existing organisational standards. These in turn may derive from industry standards. However, modifications to organisational standards may be needed because of the special characteristics of a project.

The plan also specifies how, when and by whom the quality control activities should be undertaken, the quality assurance processes to be followed and who will carry them out. It may also include configuration management and change control procedures (see Chapter 4).

The quality plan itself is subject to quality control and quality assurance processes.

It is common for the quality plan to be integrated with the **project management plan** described in Section 1.8.1.

5.7 DETECTING DEFECTS

Quality control of software products tends to use testing to establish the quality of deliverables to the client, and reviews for the intermediate products such as requirements, specifications and design documents. Software can be subject to both testing and review.

5.7.1 The V model

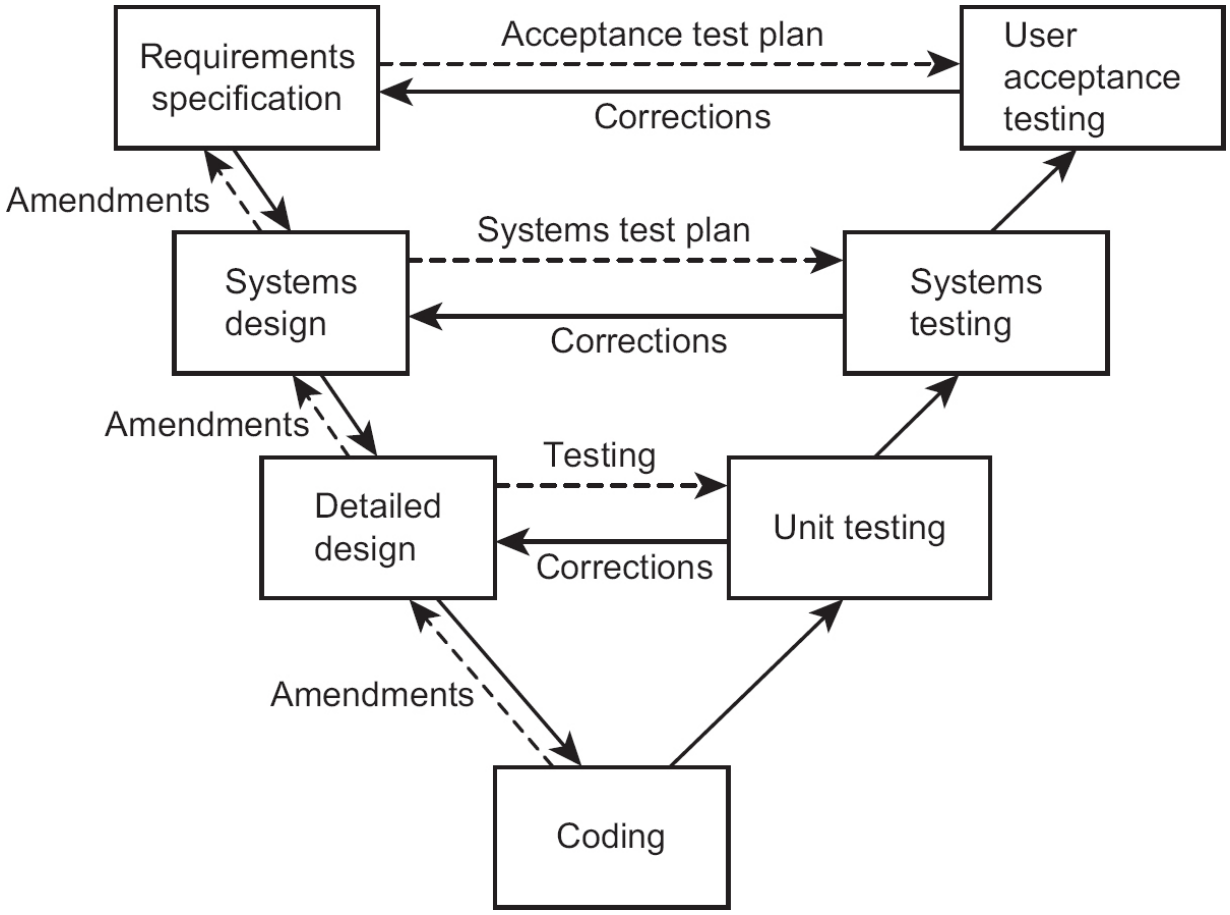
The V model is a useful model of the systems development process (see Figure 5.1), in which the solid lines represent the forward

progress of the project and the dashed lines represent the way in which quality control is exercised.

There are two quality control processes at work: one between stages and the other across the V. For example, the requirements specification describes the functions and quality attributes required in the system. This should include an acceptance test plan showing how the requirements are going to be assessed in the final system.

Using the acceptance test plan, testing can be undertaken to demonstrate that the final system to be delivered meets the requirements. This link between the requirements specification and user acceptance testing is shown by the dashed arrow between the two, identified as the ‘acceptance test plan’.

Figure 5.1 A simplified V model



Systems design follows requirements specification as shown by an arrow. Systems design has two dashed arrows: one across to systems testing and the other back to requirements specification. The horizontal arrow shows the systems testing that needs to be carried out to validate the design after it has been implemented. The arrow to the requirements specification indicates that the design process may discover errors in the systems requirements. For example, gaps may be found in the definition of the requirements, or two requirements may be found to be inconsistent. The same kind of links to the previous source document and to the appropriate type of testing are applied to each stage of development.

5.7.2 Process requirements

The importance of **product quality criteria** has been stressed. However, the qualities required in a product are captured by following the correct processes that create the product. This makes it necessary to specify the **process requirements** for each stage and activity.

Entry requirements state what must exist before the stage or activity can begin. For example, before design can begin, the requirements documentation must be agreed and signed off and any design techniques to be used must be specified. If design begins before requirements are agreed, this will lead to problems if the requirements are changed during design. The design may then have to be reworked at additional cost; worse still, the need to amend the design in line with changes in the requirements may be forgotten.

Implementation requirements define how the process should be done. For example, in design, implementation requirements may specify the use of techniques such as entity modelling or logical process modelling. Implementation requirements also specify the software tools that are to be used.

Exit requirements indicate what should be in place for a successful sign-off of the activity. The exit requirements for design documentation are that it:

- is complete;
- has been reviewed;
- meets the standards agreed;
- covers all the requirements for this component;
- leaves no outstanding issues.

5.7.3 Defect removal process

Before a defect can be removed, it must be identified. This is relatively straightforward – though more costly – in the later stages of development, when test cases can be run though the system to see if they give the expected results (dynamic testing). In the earlier stages of a project, quality checking could involve the following:

- desk checking;
- document review;
- walkthrough;
- peer review;
- inspection;
- pair programming;
- static testing.

When specifying quality criteria that apply to, say, a software specification, the technique for checking if those criteria are met should be specified, along with the type of staff and tools needed.

Desk checking

Desk checking is the most basic of the review activities. When you produce something, you yourself should be the first person to check it. After all, you are the person who knows most about what you were trying to do. This may mean reading it through and checking for mistakes and ambiguities in a document, or working through the logic of software to identify slips with code. It will, hopefully, remove many trivial errors before the product is subjected to more vigorous scrutiny.

ACTIVITY 5.5

Read through the following table and identify any errors:

Data	Item range	Cross-checks
Time: hours	0−23	
Time: minutes	0−60	
Time: seconds	0−60	
Day	1−31	Cannot be greater than 30 if month = 2, 4, 6, 7, 9 or 11
Month	1−12	
Year	>2010	

Document review

This too is desk checking, but by people other than the author to ensure that the document meets specified quality criteria, such as appropriateness and clarity. The readers may have particular concerns: a user could be concerned that a requirement satisfies the daily needs of their job; a coder might be concerned that it gives a clear indication of what the software must do. Products created in a project are not self-contained, and must be compatible with other products.

Concerns might include: is it complete, unambiguous, self-consistent and consistent with related documents? Is it clear? Are all technical terms properly used and understood? Does it conform to the agreed layout?

Walkthrough

A **walkthrough** is a particular technique where a scenario is created of a real-world situation (called ‘user stories’ in Agile). How the new system will deal with the situation is worked through and discussed, step by step. The participants could be drawn from both technical and user groups, each of which may have a different view on the proposed transaction. This is particularly useful in assessing early proposals for interface designs.

Peer review

A **peer review** is a type of document review that is often carried out on a design document or actual code. The author’s co-workers (or **peers**) examine copies of the document and make comments about it. The issues considered include:

- Is the proposed design technically feasible?
- Is there an easier or better way to achieve the same objectives?
- Does the design conform to company standards for such processes?
- Can the design communicate with other parts of the system?
- Does the design cover all the requirements that should be included?
- Are there any ambiguities?

The peer reviewers of software often carry out a version of a walkthrough where they **dry-run** the code – that is, take some test input data and manipulate it on paper according to the instructions in the code.

Peer reviews can be done relatively informally within the project team. However, the time needed by the reviewers needs to be officially scheduled – after all, they have their own software on which to work as well.

Inspection

An **inspection** is a formal review of a product. Its purpose is to review the product in order to identify defects in a planned, independent, controlled and documented manner. It is a process with the following structure:

- Preparation:
  - setting up the review meeting – including the time, place and who is to attend;
  - distributing documentation – for example, the product and its description;
  - annotation of the product by reviewers, if it is a document, and recording defects.
- Meeting:
  - discussion of potential defects identified by reviewers, which should confirm whether they are defects or not, but not seek to produce a solution;
  - agreement of follow-up appropriate to each defect.
- Recording:
  - follow-up actions and responsibilities;
  - agreement of outcome and sign-off if appropriate.

- Follow-up:
  - advising the project manager of the outcome;
  - planning remedial work;
- signing off when complete.

There are four roles within this process:

1. The **moderator** sets the agenda and controls the review process, ensures actions and required results are agreed and, once the process is complete, signs off the review.
2. The **author** provides the reviewers with relevant product documentation, answers questions about the product during the review and agrees actions to resolve defects.
3. **Reviewers** undertake the review, assessing the product against the quality criteria, identifying potential defects and ensuring that the nature of any defect is clearly understood by the author.
4. The **scribe** takes notes of the agreed actions, who is to carry them out and who is to check corrections. He or she confirms these details at the end of the meeting.

With peer reviews, inspections and walkthroughs, no attempt should be made during the meeting to solve the problems identified. Problems should be recorded. The author will then go away and seek to come up with solutions. The review can then be repeated if it is felt that the changes required were of sufficient significance. An alternative is for one person to be given responsibility for ensuring that the necessary alterations are made by the author.

Pair programming

The review techniques described above depend on copies of documents, including code, being printed and additional reviewing activities being scheduled. To avoid this, in Agile development environments, code developers sometimes work in pairs. The pair take turns to type in code at the workstation while the other advises and checks on what is being entered. This is rather like a real-time version of a peer review.

Static testing

Some software tools carry out **static testing** by analysing the structure of the code. Among other analyses, such tools look at the branches and loops in a program and calculate a measure of complexity. The more complex a software component is, the more difficult it will be to maintain.

All the above processes take place during the activities on the left-hand side of the V model. The quality control processes on the right-hand side are dominated by dynamic testing.

5.8 DYNAMIC TESTING

Dynamic testing is divided into various levels:

- unit;
- integration;
- systems;
- user acceptance;
- regression.

For each type of testing, sets of test data and expected results must be produced. Referring back to the V model, a test plan should have been produced at the appropriate stage in the development process on the left-hand side of the V. The plan should contain the necessary guidance for producing the test data, if not the test data itself.

5.8.1 Unit testing

Unit tests are the first tests carried out on a software component. It is often done by the coder. In the V model this checks the coder has faithfully implemented the detailed design. The detailed test plan for the component should cover the range of inputs expected and how they are handled by each function in the component. The test should test various input combinations and sequences. Testing is then extended to cover, for example, numbers just inside and just outside any specified limits. All tests are designed to ensure that this particular unit will not fail because of bad data or unusual combinations but will handle them in a predefined way.

A good practice is for test-first data and expected results to be drafted before the coding is commenced. Thinking about the detail of how inputs will be converted into outputs can clarify requirements early on.

ACTIVITY 5.6

Assume that the table of time/date checks drawn up (and corrected) in Activity 5.5 has now been implemented in an input screen in an IT system. Draw up a set of test data and expected results that could be used to test that the checks on data are being carried out correctly.

Test results should be carefully checked against the expected results. There are automated tools that can save human effort here after initial set-up. Discrepancies are recorded so that the software can be amended and retested. Sometimes, however, it is the expected results that are wrong! Resulting amendments are also recorded. If problems arise later, there is then a trail that can be followed to establish how they were introduced.

There are tools – commonly called **test harnesses** – which can simulate the software or hardware that supply data to or use data from the module under test. The test harness can record data input and output and sometimes the routes through the program exercised. Automated tools can also simulate keyboard input and capture and compare screen output with expected results. In addition, dynamic analysis can identify code not executed by the test data used. This is useful as it may show up a shortcoming in the test data or unneeded code in the software.

Once unit testing has been successfully completed, the unit can be signed off and registered as a configuration item (see Chapter 4).

5.8.2 Integration testing

Integration testing links a number of system components and runs them as a whole. This checks that the units communicate properly with each other. Discrepancies include data items being treated as different formats in different units, and conditions being set in one component that another cannot cope with. Some of these can be avoided by the use of shared databases and database management systems (DBMS) and by careful modular design that keeps interactions between components to a minimum.

As errors are found, they will be recorded and corrected. The integration test will then need to be repeated. The repetition of testing makes this a fruitful area for automation.

Agile approaches stress the need for early and frequent integration testing. In some environments this can even be done while software components are still under development. The idea is to catch discrepancies between components as early as possible.

5.8.3 Systems testing

Systems testing is the final stage in testing by the development professionals. It involves running the whole system on the infrastructure that will be used when the system is operational. It may sound just like a step up from integration testing, but there are many additional issues which must be addressed; for example:

- Does the system run correctly on the infrastructure to be used for the final system?
- Are the response times within the tolerances set in the requirements specification?
- Can the system cope with the planned workloads?
- What is the effect of high loading on the system?

Again, there are tools to help. They can be used to simulate large numbers of users and high volumes of data.

5.8.4 User acceptance testing

User acceptance testing is the crucial test. Can the users operate the system? Does it meet their expectations, not just their

requirements? Users should be involved in the development process from the beginning and have had sight of how the system works before this point so that there are few surprises. While it is helpful to involve users in earlier testing activities, seeing the faults that arise during testing may make them anxious.

Acceptance testing underlines the importance of having clear requirements from the start. Users should have a well-defined acceptance test plan to guide them as they test whether the system meets their expressed needs. Where discrepancies are found, the issue must be recorded, the source of the problem established and, if need be, the system must be modified. Sometimes, the system modification may have to be to the users' way of working.

5.8.5 Regression testing

Regression testing is quite different from the earlier testing processes. Regression testing needs to take place at each stage of testing. Whenever a fault is found and the offending piece of software identified, it has to be corrected. Unfortunately, the correction itself could introduce further errors or uncover ones that were masked by the first error. Regression testing involves running an agreed set of test data through the system again to confirm that not only has the original error been corrected but no further errors have been introduced or uncovered. Regression testing can largely be automated by using a standard set of test data and expected results.

5.8.6 Management of testing

The project manager has certain key concerns in relation to testing:

- The way the system has been designed has an impact on how easy it is to test it. The development of components that are as self-contained as possible and where there are external outputs, that indicate the internal state of the unit, make testing easier. In some cases, special platforms have to be set up to support testing. Thus, the way the testing is to be done needs to be taken into account at an early stage of the project planning.
- The number of defects is hidden in the software and thus unknown, so the time needed to complete testing can be uncertain. Hence the need for quality checking at the earlier stages of development, which can reduce defects before testing is started.
- If testing is finished too early, the reliability of the resulting operational system may be jeopardised. In most systems there are critical functions that are heavily used. For example, in the Water Holiday Company scenario, the function that checks the availability of vessels will, if successful, be the precursor to an income-generating booking and should therefore be subjected to very careful testing.

5.9 EVALUATING SUPPLIERS

With the increase in specialisation, it is quite usual for a new system to be developed by an external supplier rather than in-house. Where the development responsibilities devolved are significant, a rigorous selection process should take place with **invitations to tender** being sent to potential suppliers. The organisation would then have to evaluate proposals from suppliers based on the scope of what they can undertake and their price. There are many articles and books on

contract negotiation and management; the focus here is on the quality aspects.

COMPLEMENTARY READING

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

Where new functionality is being created by the supplier rather than an existing application being installed, the project manager will be particularly concerned with the implications for the quality of deliverables. Care about specifying the quality of deliverables will be important, but the final quality of these products will only be known when they are delivered towards the end of the project. This is rather late, so the alternative is to focus on **process quality** and how well the supplier carries out their activities.

The types of question that could be asked of the supplier include:

- How are their projects organised? (For example, do they follow a framework like PRINCE2, the standard for project management procedures sponsored by the UK government?)
- Do they use a defined development methodology, such as the Dynamic Systems Development Method (DSDM)?
- How is quality control exercised?
- At what points are quality reviews held?
- Is there a quality assurance process?

- Do practitioners have appropriate professional certification?
- Is there a configuration management system in place?
- How are change requests handled?

This list is by no means exhaustive. The response to each question should be supported by evidence. For example, if it is claimed that software quality is assessed by reviews, then examples of the outputs from the reviews can be examined. Observing some of their reviews can increase confidence in their quality processes.

As well as the quality of a potential supplier's proposal, the standing of the supplier itself needs to be assessed. There might be an important post-project relationship with the supplier for ongoing maintenance, and their continuing financial stability would need consideration. These wider procurement issues are beyond this book's scope – but see the complementary reading box above.

Such detailed inquisition would be time-consuming for the organisation, particularly if there is a large number of potential suppliers. It may be easier to assess suppliers on the basis of their accreditation to professional and standards organisations. Most of the concerns raised in the questions above are covered by ISO 9001 quality management system (QMS) accreditation, which is discussed next.

5.10 ISO 9001:2015

ISO 9001:2015 is the international standard for **quality management systems** (which we introduced in Section 5.5) and is aimed at producers and suppliers of any products and services, not necessarily software. However, a subsequent document, ISO 90003,

describes the way ISO 9001 can be applied to software development projects.

Organisations can be inspected and awarded ISO 9001 certification by accredited auditors. This means that, as a potential client, you can assume a particular standard of quality management without having to carry out detailed checks yourself.

However, while ISO 9001 states that a quality level should be specified, it does not say what that level should be. For example, for the Water Holiday Company project, it could be stated that a performance quality requirement is for an average response time of 3 seconds for a boat availability query, but equally it could state 6 seconds. Thus, having a set of ISO 9001 procedures only guarantees that a level of quality has been specified, not that this level is accepted universally as appropriate. This allows the client of an ISO 9001:2015 supplier to negotiate the quality criteria that they personally need.

ISO 9001:2015 is based on the following principles:

- **Customer focus:** The supplier focuses on what the customer really needs.
- **Leadership:** Top supplier management must have a genuine concern for quality, which is communicated through word and deed across all levels of the organisation.
- **Engagement of people:** For quality to be delivered, there must be '*competent, empowered and engaged*' staff. This is more likely where there is an ethos of teamwork, open discussion and knowledge-sharing.

- **Process approach:** The processes used to produce quality products and services must be defined, understood and managed.
- **Improvement:** Effective management of processes is a foundation for finding ways of improving them. Improvements will be supported by evaluation of current practices, education and innovation.
- **Evidence-based decision-making:** Decision-making is based on the analysis of data and information. This enhances the understanding of cause and effect when delivering products and services that meet customer needs.
- **Relationship management:** Most supplier organisations themselves depend on other suppliers of products and services in their supply chains. This will have a significant influence on the quality of their outputs. Thus, effective relationships with other members of their supply chains are essential.

It was stressed above that ISO 9001:2015 is relevant to a broad range of organisation types. There is an expanding number of supplementary standards written to show how the generic ISO 9001 principles can be interpreted and implemented in different business contexts. An updated version of ISO 90003 has been published that applies ISO 9001 to software development: ISO 90003:2018.

The UK-based TickIT*plus* scheme is designed to enable organisations involved with software development, and also with the delivery of other IT-related products and services, to be accredited as compliant with ISO 9001:2015. As it relates to IT, the scheme can also take account of other IT standards, such as ISO 15504:2102, which enables the assessment of IT process quality.

TickIT*plus* also goes beyond a simple yes/no model of compliance. It can allocate an organisation to a particular capability level, that is: foundation, bronze, silver, gold and platinum. The concept of capability level is discussed further below.

Capability maturity models

In the discussion of ISO 9001:2015, we assumed that the motivation was to allow customers to assess potential suppliers. Sometimes, however, the managers of a supplier organisation want to assess their own quality processes to find ways of improving them. They want to gauge their current level of effectiveness, but also identify what needs to be done to get it to a higher level. This brings in the concept of **maturity models**, where the organisation is assessed as being at a particular level of **process maturity**.

One example of this is the capability maturity model (CMM) originally developed by the Software Engineering Institute at Carnegie Mellon University for the US Department of Defense. This comprises five levels:

1. **Initial:** Any organisation would be at this level by default. Good quality work may be done, but customers cannot be sure that this is always the case.
2. **Managed:** Some basic project management and other systems are in place.

3. **Defined:** The way each task in software development is done is defined to enable consistent good practice.
4. **Quantifiably managed:** Processes and their products are measured and controlled – through, for example, the number of errors created in each process.
5. **Optimising:** The measurement data collected is analysed to find ways of improving processes.

The assessment of an organisation's maturity level can be done internally, or external auditors could be employed so that the maturity level can be published externally, as in the case of ISO 9001.

SAMPLE QUESTIONS

1. **'Fitness for purpose' defines which of the following?**
   a. The quality of the project deliverables
   b. The usability of the delivered IT application
   c. The capability of the staff who will implement the IT application
   d. The capability of the staff who will use the system when it is operational
2. **User acceptance testing is an example of which of the following?**
   a. Project control
   b. Project monitoring
   c. Quality control

d. Quality assurance

3. **Which of the following is NOT a defined role in inspections?**
   a. The moderator
   b. The project manager
   c. The scribe
   d. The author

4. **ISO 9001:2015 is a standard that defines which of the following?**
   a. Project management standards
   b. IT project deliverables
   c. Quality management systems
   d. Software testing procedures

POINTERS FOR ACTIVITIES

Activity 5.1

Quality criteria could include:

1. A product definition or description must contain the following headings or sections (from Chapter 2):
   - identity;
   - description;
   - derived from;
   - components;

- format;
- quality criteria.

Additional headings could include the following:

- author;
- owner;
- date first compiled;
- date of last amendment;
- version number.

The ‘derived from’ section must refer to valid product types.

2. The ‘format’ section must provide enough information to allow someone to create an instance of the product in the correct form.

   One can easily identify other criteria.

   These criteria meet the requirement of being specific (for example, a list of obligatory section headings), measurable (the sections are either there or not there) and achievable (it is possible, without difficulty, to ensure that each heading is present).

   The quality of a product description will most likely be measured through a review process, such as inspection by a fellow developer. What this process does not do is to ensure fully that the correct information is entered for each heading. This may require a further set of quality criteria, which again should be subject to review.

Activity 5.2

Criterion	Assessment
All screen layouts should have similar layout and use the same terminology	This is relatively clear, and measurements could be devised, but more detailed checklists of things to look for (as might be found in a style guide) would be helpful.
Screens should be user friendly	This is subjective and therefore not measurable.
The system should be able to handle 50 transactions	This is not measurable as there is no indication as to the period of time within which the transactions have to be handled.
The system should allow for 20 users at any one time without degradation	This is better than the previous criterion, but clarification of what each user might be doing could be requested.
The response time should not be longer than three seconds	This is a mixture. It is clear that a response time of three seconds or less is required, but it does not specify under what conditions.

Ideally the last two criteria should be combined so that a baseline of three seconds is given with a loading of 20 simultaneous users. This can still be improved upon. For example, it could be stated that the response time should be less than four seconds for at least 95 per cent of the time with a loading of 20 simultaneous users and should never exceed 10 seconds. Such a statement does allow for the odd

occasion when the three seconds might be exceeded. These examples show how difficult, yet important, it is to get the quality criteria correctly specified for each product in the development process.

Activity 5.3

As can seen in the main text of this chapter, there is a difference between measuring the quality of an existing software component – where actual performance can be measured – and assessing the likely quality of an application as it is being built. In an existing software component, we could collect statistics about the amount of effort that has been needed to implement actual changes.

If the software is being created, we could examine the code to see if it has characteristics that are likely to lead to maintainability. A system would be maintainable if it satisfied the following criteria (among others):

- The structure of the software is clear.
- The names used for items of data and procedures are indicative of the nature and purpose of each of these items.
- The purpose and method of manipulations of data in code is clear and unambiguous.
- Documentation is present to support code.

There are other possible software engineering criteria that could be discussed, such as loose coupling of components (minimal cross-references) and cohesion (all code for a function being together).

The measurement for these criteria would probably be a peer review process.

Activity 5.4

There are two ways in which the reliability of a system has been traditionally measured:

1. **Meantime between failure** (MTBF) specifies how long the system runs without failing. These days, this would be specified in weeks or even months. Something which fails every day would not be very popular with those operating the system.
2. **Meantime to repair** (MTTR) specifies how long it takes to repair the system when it fails. It is no good if a system cannot be restored to a working situation in a reasonable length of time. That length of time needs to be specified as part of the acceptance criteria. Note that this measure is also related to the attribute of maintainability.

Other valid measurements could be considered, such as the percentage availability of the system.

Activity 5.5

The errors include:

- Time: minutes should be in the range 0−59.
- Time: seconds should be in the range 0−59.
- Day: July (month 7) has 31 days, February never has more than 29 days and there is no leap year check. The cross-check should be:

- ‘If month = 2 and year is leap, up to 29;
- if month = 2 and year is not leap, up to 28;
- if month = 4, 6, 9 or 11, up to 30.’

Activity 5.6

You may find some ‘holes’ in the above test data. It should illustrate that although devising test data is not the most glamorous job, creating effective test cases does require the kind of attention to detail that we normally expect of software developers. Devising test data will also trigger questions about the precise nature of the requirements – for example, is there really no upper limit on year?

Data item	Test description	Input	Expected result
Day	Not numeric	XX	Error message
	Outside lower range	0	Error message
	Inside lower range	1	Accept
	Outside upper range	32	Error message
Month	Not numeric	July	Error message
	Outside lower range	0	Error message
	Inside lower range	1	Accept
	Outside upper range	13	Error message
	Inside upper range	12	Accept
	Cross-check with day	day = 31 month = 1, 3, 5, 7, 8, 10, 12	Accept
	Cross-check with day	day = 31 month = 2, 4, 6, 9, 11	Error message
	Cross-check with day	day = 30 month = 4, 6, 9, 11	Accept
	Cross-check with day	day = 30 month = 2	Error message
Year	Not numeric	xx	Error message
	Outside lower range	2009	Error message
	Inside lower range	2011	Accept
	Cross-check with day	day = 29 month = 2 year divisible by 4	Accept
	Cross-check with day	day = 29 month = 2 year not divisible by 4	Error message

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.’

Risks are always made up of cause and effect. An unwelcome outcome, such as the late delivery of the product of a project, could have a number of causes, such as an estimate being wrong, staff not being available or a requirement being changed.

In Chapter 1 we distinguished between **project objectives** and **business objectives**. Similarly, some risks are **project-related** – that is, they threaten the successful achievement of the project’s objectives – and others are **business risks** and can be problematic even if the project objectives have been fully met. In this context ‘business’ describes the general field of activity of organisations regardless of whether they are in the public or private sector. One can, for example, talk of schools being in the business of teaching.

Project risks can relate to **development**, such as delayed delivery, which can strike during the execution of a project. **Operational** project risks emerge when the delivered system is put into action, that is made operational – an example of this is vulnerability to cyber-attacks – and are usually caused by incomplete requirement gathering or poor quality deliverables. There is an overlap between quality and risk management as it is likely that the product qualities described in Section 5.3 could be redefined as risks, that is the possibility of operational systems going wrong.

Business risks occur when the planned benefits of a correctly delivered project are not achieved because of external business factors. In the Water Holiday Company scenario, a slump in the demand for boating holidays could mean that the investment in the new integrated booking system does not pay for itself. On the other

hand, risks involve opportunities as well as threats. Unexpected events might be to our advantage if we can modify our plans quickly to take account of them.

When a risk actually occurs, it becomes a **project issue** – that is, a problem that needs to be resolved. Risk management actions can reduce the probability of the project issue emerging or define actions to reduce the damage it causes.

The objectives of risk management are to identify, address and minimise risks before they become threats to the successful completion of a project. However, we need to be aware of the 'law of diminishing returns', which suggests that the initial effort and expenditure provide the best return and that the benefits from further spending to solve a problem gradually become smaller. Buying one smoke detector for your house when you have none could make a big difference to your safety, but buying another when you already have five will probably make little difference. The cost when a risk actually occurs needs to be balanced against the costs of actions to reduce the likelihood of the risk occurring.

7.2 RISK MANAGEMENT

The management of risks is similar to the management of any activity. There is a control cycle for tracking risks – see Figure 7.1 – similar to the project control cycle described in Section 3.1. This risk management cycle is repeated throughout a project.

Figure 7.1 shows how major risks are identified and then plans made to deal with them. These plans include activities that enable other activities to be undertaken in the event of a risk turning into a real problem. For example, taking back-ups of important data allows

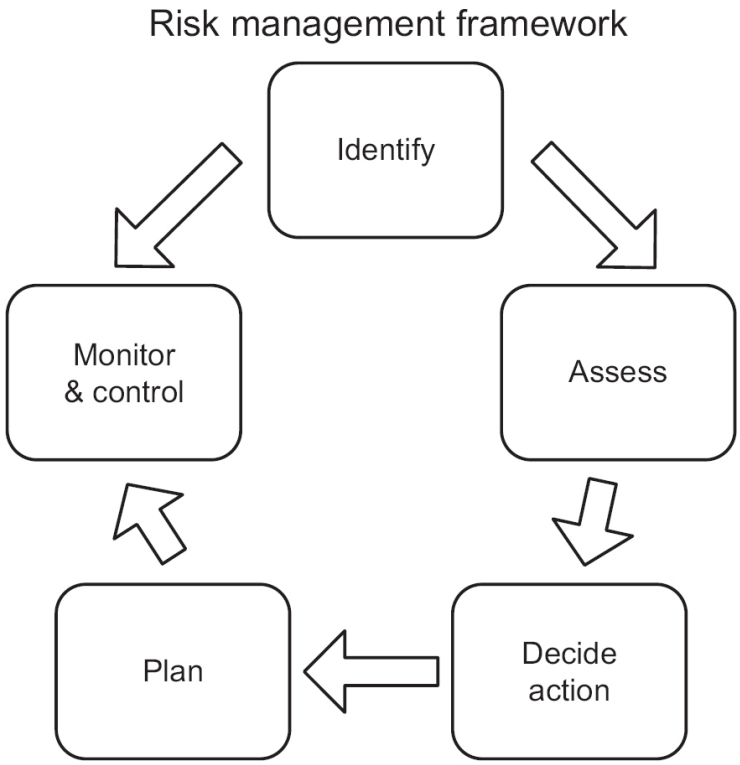
them to be restored if the originals are damaged. The project is then executed and is monitored and controlled to see where risks have materialised and where appropriate actions need to be initiated.

Although risk is being treated as a separate topic in this chapter, risk planning is embedded as part of general project planning. As planning takes place, assumptions will have to be made; for example, that a particular resource will be available when needed. An assumption that could turn out to be incorrect becomes a risk. Risks can be eliminated at the planning stage by changing the way a project is implemented. For example, risks associated with developing software can be removed by buying in existing software functionality.

During the implementation of the plan, the monitoring of risks would be integrated in generic project control – see Chapter 3.

Generic risks that affect nearly all projects are probably best dealt with by adoption of relevant, recognised, good preventative practices.

Figure 7.1 Risk management cycle



7.3 IDENTIFYING RISKS

The risk identification process begins with the gathering of potential problems or issues that are already known. These include particular views, trends or constraints within the project development environment. These recognised problems often indicate the causes of the risks that will eventually need to be managed.

ACTIVITY 7.1

Reread the Water Holiday Company integration project scenario in Chapter 1 and identify features of the project or its environment which you think could lead to difficulties.

There are a number of aids to building the initial list of risks. Specialist risk publications, application development guidelines and company standards provide **prompts** or **checklists** containing a number of **generic** business and project risks that originate from outside and inside the project. These can be used to determine which risks in the list apply to the project. Some of these checklists relate to projects in a particular industrial sector, like the well-known Barry Boehm ‘Top Ten Software Project Risks’:

1. **Personnel shortfalls** – for example, developers not being familiar with the technologies needed for the project.
2. **Unrealistic schedules and budgets** – in some cases this could be because not all essential requirements have been identified.
3. **Developing the wrong functions and properties.**
4. **Developing the wrong user interface.**
5. **Gold-plating** – this is development of software functionality that is not really needed and ends up not being used.
6. **A continuous stream of requirements changes.**
7. **Shortfalls in externally furnished components.**
8. **Shortfalls in externally performed tasks.**
9. **Real-time performance shortfalls.**
10. **Straining computer-science capabilities** – current technologies and expertise are not sufficiently well developed to satisfy the requirements and the project becomes effectively a research rather than a development project.

Not all risks in a generic list will be relevant to a specific project. Some can be discounted at once. For example, shortfalls in externally performed tasks would not be relevant to a project which is completely in-house. Others may be dropped after the more detailed assessment described in Section 7.4.

As well as **generic** risks applying to almost any project, there are **specific risks** peculiar to the circumstances of the particular project – the areas of potential risks to the Water Holiday Company integration project identified in Activity 7.1 were specific to that project. Methods of gathering information about specific risks include:

- interviewing experts or stakeholders within the project;
- brainstorming workshops with stakeholders to identify risks – the discussion of a risk by one member may lead to others recognising further risks;
- searching past project documentation, particularly any 'lessons learnt' reports.

It is helpful, in identifying a risk, to recognise the true focus of the problem. As noted already, delays in activities can be caused by a variety of factors, so you should not say:

> 'Development of the application software may overrun.'

Instead, you should say:

> 'There is a risk that the application programmers are too inexperienced in the chosen language and therefore the application software may be completed late.'

Note that this risk statement is only appropriate when it is set in the context of the given project. In this example, it is a fact that the programmers are inexperienced in the chosen language. This fact, or issue, is not a risk but a cause of a potential risk. The risk still has an element of uncertainty about it. The fact that there is limited experience of the chosen programming language only becomes a risk in our project if the difficulty of the design and coding of the application software makes demands that are too great for the inexperienced programmers.

ACTIVITY 7.2

Given the list of possible issues generated in Activity 7.1, identify five possible risks for the Water Holiday Company integration project.

7.4 ASSESSING THE RISK

We have to recognise that we may not be able to take action for all possible risks identified. Therefore, we need to prioritise the risks, but before this can happen we need to evaluate the risk itself.

7.4.1 Risk evaluation criteria

The evaluation of the seriousness of a risk is based on two key criteria:

- the **probability** that the risk will occur;
- the **impact** that the risk will have, should it occur.

Together, risk and impact give an idea of the magnitude of the risk – the **risk exposure**. A third factor is the **proximity** of the risk, which takes account of the fact that the magnitude of the risk may vary throughout the project – for example, once coding has been successfully completed, some risks relating to coding disappear.

7.4.2 Risk exposure

We noted above that the **impact**, or severity of a risk, is the adverse effect that the risk will have on the project should it materialise. This impact might be a longer development **time**, a reduction in the **scope** of the deliverable, a reduction in the **performance of the deliverable**, or an increase in the **resources needed**. The scope and the performance are often combined as a reduction in **quality**. The increased resources, both of materials and labour, are usually referred to as increased **costs**. The turning of a risk into a real issue needing action could affect the business case because of increased costs or reduced benefits.

A risk can be viewed as an **opportunity**. Let's examine the example on the previous page, of developing application software in a language (say Java) with which developers are not familiar. The plan for the software development could increase the expected duration of the coding tasks to take account of the developers' lack of experience. If, however, the developers were able to pick up Java very quickly, their tasks could be completed earlier than scheduled. In this case the project manager ought to exploit this opportunity and start the next tasks as soon as possible. The time gained here will be a useful buffer if other problems occur later in the project.

Impact is not the only issue that affects the seriousness of a risk (or risk exposure). A risk could cause immense damage if it occurs – as

in the example of an aircraft crashing into a workplace – but in practice be dismissed because of the minute probability of it occurring.

7.4.3 Risk proximity

The **proximity** relates to the period in the project when the risk could occur. A given risk is more likely to occur during one or more particular activities. After a certain project milestone it might no longer be applicable, or at least have a reduced impact. The risk of inexperienced Java programmers delaying completion of work will affect the software development stage. Once the software coding is over, this will no longer be a risk. In Chapter 1, it was noted that uncertainty about a project was greatest at the beginning because of all the unknowns associated with a new project. As knowledge is gained about the application and technical domains during the project, much of this uncertainty is reduced.

7.5 QUANTITATIVE APPROACHES TO RISK

Risk assessment can be **quantitative** – based on seemingly precise mathematical values – or **qualitative** – based on broader management intuition.

Quantitative risk assessment

When a quantitative approach is used, probability is represented as either a percentage between 0 per cent and 100 per cent or a value in the range of 0.00 to 1.00. 0 per cent or 0.00 means

there is absolutely no chance of something happening, while 100 per cent or 1.00 means it is absolutely certain that it will happen. A probability of 0.40 means there is a 4 out of 10 chance of something happening.

Impact is most conveniently measured as a monetary value reflecting the financial loss of the risk should it actually occur, but is sometimes measured in time (that is, the amount of delay caused).

The values for probability and potential impact of a particular risk can be used to calculate **risk exposure**.

Risk exposure = impact × probability

For example, if there were a 0.10 probability of IT equipment worth £20,000 being stolen, the risk exposure would be £20,000 × 0.10, that is, £2,000. (Note that all the numbers here are picked for ease of the arithmetic, not because they are realistic.) Crudely, this risk exposure value can be compared to the amount that might be paid as an insurance premium. If there were 100 organisations with IT equipment of the same value and the same chance of theft and they all contributed £2,000 to a pool, the pool would be big enough for 10 per cent of them to withdraw £20,000 if they were robbed. (This is a simplified model: in real life the 10 per cent would have to be based on an average over several years. It is unlikely that it would be exactly 10 per cent in any one year.)

An advantage of the quantitative model is that it is easy to assess the effectiveness of a risk reduction action. Say that in the above example an organisation decided to buy a burglar

alarm for £1,500 (once again, this figure has been picked simply to make the calculation easy) and it is estimated that it would reduce the probability of a successful theft to 1 per cent (or 0.01). A risk reduction leverage can be calculated as follows:

Risk reduction leverage (RRL) = (RE_{before} − RE_{after}) / cost of risk reduction

RE_{before} is the risk exposure before the risk reduction action is taken – that is, £2,000.

RE_{after} is the risk exposure after the action (the installation of the burglar alarm) – that is, £20,000 × 0.01, or £200.

The calculation of RRL is therefore (£2,000 − £200) / £1,500) = 1.2.

Because the RRL is greater than 1.0, it means that the reduction action is worthwhile. (This could be compared to the cost of the burglar alarm being offset by a reduction in insurance premiums.)

There are a few problems with the practical application of quantitative risk assessment:

- Unless you have a very large set of data about past occurrences of the particular risk, identifying the probability of a risk may end up as guesswork.
- In our simplistic example, the cost of the theft was exactly £20,000. In practice the amount of damage can vary, and so this value could be guesswork. Where there is a large amount of data about past occurrences of the risk, it may be possible to produce a table showing the probability of

different ranges of cost – but this kind of information is unlikely to be available to a project planner.

- Quantitative risk exposure values are based on the principle that when risks actually occur, the situation can be remedied by using resources put aside to meet possible losses. However, this assumption does not hold where the loss caused by a particularly large risk occurring is simply too large and would exhaust the fund. The bankruptcy of the client organisation might be an example of these **show-stoppers**.

7.6 THE QUALITATIVE APPROACH TO PROJECT RISK ASSESSMENT

Because of these problems, modern practice in project risk management tends to adopt a more qualitative approach, and it is on this that we will focus in the remainder of this chapter.

7.6.1 Risk probability

While quantitative risk assessment requires access to data about past projects, other approaches to obtaining qualitative assessments were noted in Section 7.3, including interviewing experts or stakeholders and brainstorming in a workshop. Various qualitative descriptions of probability can be used to describe the probability of a risk occurring, such as ‘extremely likely’, ‘very high’, ‘high’, ‘medium’, ‘low’, ‘very low’ or ‘improbable’. Similar descriptions are used in the qualitative assessment of the impact the risk will have on cost, quality or time.

Quantitative values can still be assessed, but these will be expressed as being within a range – for example, a 20–50 per cent probability of occurrence – and then be mapped to one of the categories of the probability and the impact (see Tables 7.1 to 7.4).

The Delphi method (see Section 6.3.2) is a more formal version of the expert approach to risk assessment. Risk assessment is very closely associated with effort estimation and in some cases can be carried out at the same time.

Table 7.1 Mapping qualitative and quantitative assessments of risk probability

Index	Impact level	
4	High	Greater than 50% chance that the risk will occur
3	Significant	30–50% chance that the risk will occur
2	Moderate	10–29% chance that the risk will occur
1	Low	Less than 10% chance that the risk will occur

Table 7.2 Mapping qualitative and quantitative assessments of cost impact

Index	Impact level	
4	High	Greater than 20% above project cost tolerance

3	Significant	Up to 20% above the project cost tolerance
2	Moderate	Greater than 50% of the project cost tolerance but still within it
1	Low	Within 50% of cost tolerance

Table 7.3 Mapping qualitative and quantitative assessments of scope impact

Index	Impact level	
4	High	Inability to meet mandatory project functionality
3	Significant	Shortfalls in key functionality
2	Moderate	Shortfalls in secondary functionality
1	Low	Some minor functions missing

Table 7.4 Mapping qualitative and quantitative assessments of time impact

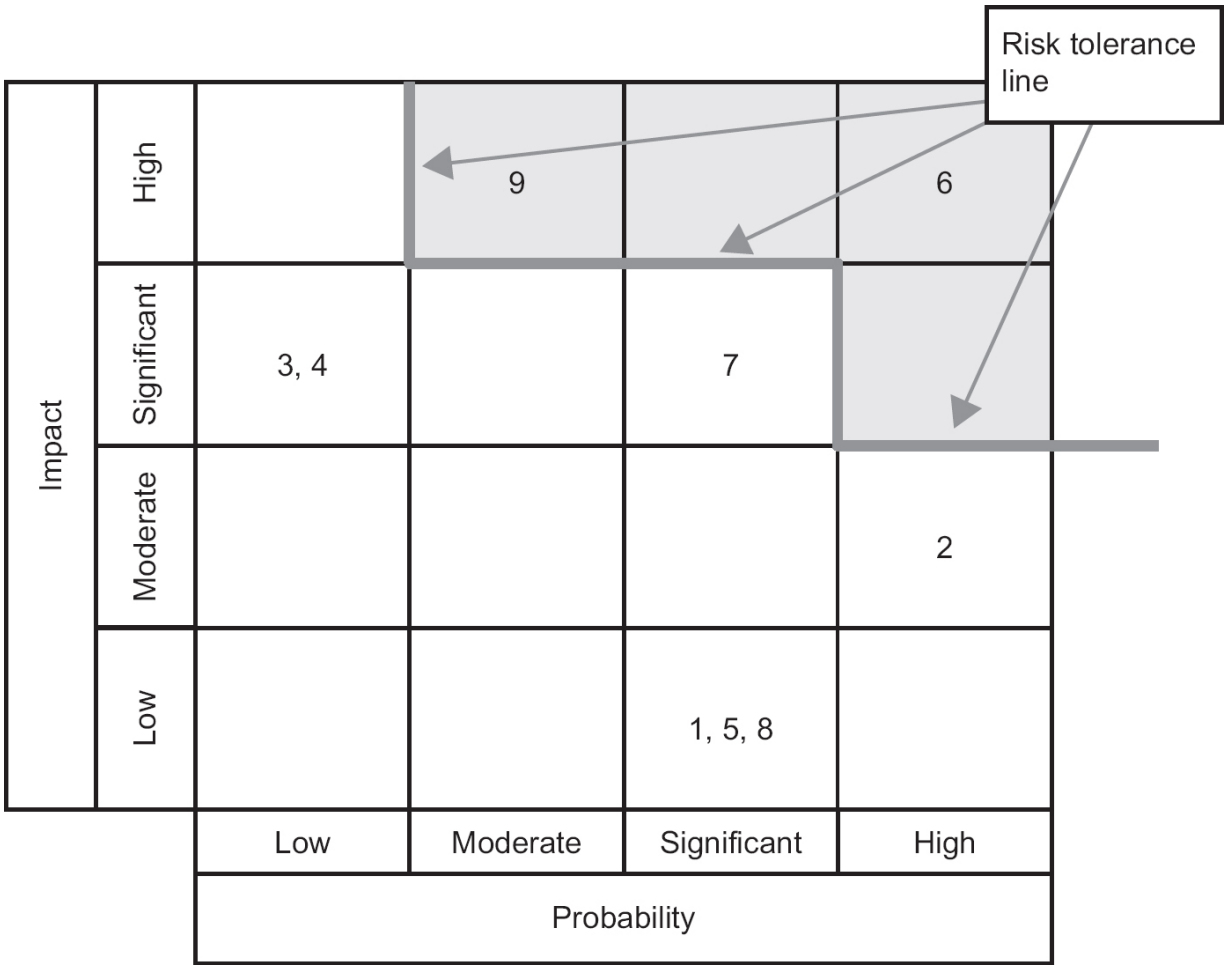
Index	Impact level	
4	High	Greater than 20% above project time tolerance
3	Significant	Up to 20% above the project time tolerance
2	Moderate	Greater than 50% of the project time tolerance but still within it

1	Low	Within 50% of project time tolerance

7.6.2 Prioritising risks

Once the evaluations of risk probability and impact have taken place, the risks can be prioritised to ensure that the risk management effort is placed where it is needed most. We have already seen that where a quantitative approach has been taken, a risk exposure value can be calculated. The bigger this value is, the more serious it is. However, this calculation cannot be done if we have made qualitative assessments. To aid decision-making for qualitative assessment, a **probability impact grid**, sometimes known as a **summary risk profile**, can be used (see Figure 7.2). In the grid, the numbers uniquely identify each risk. Some organisations will have three probability impact grids to cover the impacts on time, cost and quality, respectively; other organisations combine them. Organisations refine their understanding of the risk profiles over time and become able to judge more accurately the threat of a risk and the action to be taken to deal with it.

Figure 7.2 Probability impact grid



7.7 DECIDING THE APPROPRIATE ACTIONS

Uncertainty about a project due to the risks identified can be reduced by taking **risk management actions**. These actions aim to reduce, transfer or eliminate a risk. They are additional tasks that could affect the schedule, costs, functional scope and quality of the deliverables. An alternative is to take no action to reduce the possible occurrence of the risk; instead a contingency plan could be put in place consisting of actions to reduce the damage caused once the risk had actually materialised.

7.7.1 Accepting the risk

The organisation could accept the risk and take no further action other than to monitor it. It could be argued that taking no action is not a risk management action, but it is mentioned here as a possible option. This option could be chosen if the risk probability is low, the impact is acceptable, or it is thought that none of the other actions listed below would be appropriate. The actions might be rejected because the cost of action outweighs the impact cost – the risk reduction leverage in Section 7.5 measures this – or because it is not practical in the particular context. In the case of the programmers' Java inexperience, the cost of action was judged to exceed the cost of the impact of the risk.

7.7.2 Avoiding the risk

This is sometimes called 'risk prevention'. The organisation could prevent the risk from occurring by removing or replacing the activity in which the risk is embedded. In the example where the programmers' inexperience of the chosen language is a risk, the risk might be prevented by deciding to use an existing vendor-supplied application.

7.7.3 Reducing the risk

The organisation could still take the planned action, but reduce the probability of the risk occurring. Again, any reduction action will take place before the expected risk occurs. In the software development example above, steps could be taken to try to ensure that the development was not late despite the inexperience of the programmers. For example, one or more specialists experienced in the chosen language could be recruited to join the development team to act as advisers to reduce technical misunderstandings of the features of the 'new' programming language.

7.7.4 Transferring the risk

The organisation may take action to transfer the risk to another party. For example, an organisation with inexperienced programmers could contract the work out to a software house. If the software house was working to a fixed-price contract, then the problem of possible cost overruns would be transferred to them.

7.7.5 Contingency

The organisation may decide not to take any action before the risk occurs, but instead to plan an action to be taken once the risk occurs, or if it becomes more certain that the risk will occur. For example, it is very difficult to think of many practical ways of eliminating the possibility of key staff becoming ill at a critical point in the project. Contingency planning might identify staff who could cover the role of a staff member made unavailable in such circumstances. This action differs from other actions as generally it only incurs costs if the risk materialises. As with all actions, there will be costs associated with managing the risk (see Section 7.5). There may also be costs associated with creating the conditions that would allow the contingency action to take place, as is the case when back-ups of files have to be taken to allow the contingency action of restoring files if they are corrupted.

ACTIVITY 7.3

Explain the risk management actions that could be taken to reduce the risks to the Water Holiday Company integration project scenario that were noted as a result of Activity 7.2.

When selecting the risk management actions to be taken, it may well be that some actions are appropriate not just for one particular risk, but for several. For example, there is a risk that a delivered software function might not perform at the outset to the satisfaction of users, so that time is wasted in re-work. Two risk management actions are identified: to have a walkthrough of the interface design with users to ensure their requirements are met; and to have a peer review of the code to reduce faults in it. It happens that the peer review of code also identifies changes that reduce the risks of the code being difficult to maintain in the future.

The action chosen will have an impact on plans. In the above example, two new review activities need to be inserted into the project schedule. It is important to ensure that the benefits of these actions outweigh the benefits of inaction. Two extra reviews might take up two days, say, so the reduction in testing and correction would need to be more than this for it to be worthwhile.

Key decisions in risk management are how many risk management actions to approve and in relation to which risks. Initial focus is likely to be on the **show-stoppers** – risks that would prevent completion of the project.

Where a quantitative approach has been adopted, the risk exposure figures for individual risks could be summed to get an overall project risk exposure. If necessary, a number of actions could be planned to reduce the **risk exposure** of the project to an acceptable level. As we saw in Section 7.5 with the discussion of risk reduction leverage, these actions would incur costs.

With the qualitative approach, a **risk tolerance line** has been shown on the probability impact grid in Figure 7.2. The organisation will not

approve a project with risks that occur above this line; therefore, action would be taken to ensure a new position on the grid for these risks by reducing their probability or impact, or both.

7.8 PLANNING, MONITORING AND CONTROL

The risk identification, risk assessment and the selection of risk management actions occur primarily during project planning, but the processes described above will also continue throughout the life of the project as new risks are identified. There may also be secondary risks that result from actions to reduce initial risks. For example, outsourcing software development to a software house because of the inexperience of in-house developers will itself generate new risks relating to the reliability of the external suppliers.

The monitoring of risks should be part of the project control cycle (see Chapter 3). The monitoring process is usually a mixture of periodic reviews – say once a month – and reviews after milestones, such as the end of a project stage.

It is important to have a structured project risk plan to document the planning and to facilitate the monitoring and control process. This will consist of a **risk register**, also known as a **risk log**, which lists all the risks identified with the project (see Figure 7.3), and the individual risk records (see Figure 7.4). Typical headings are shown in the figures, but others could be added – for example, a risk category or the earliest date that the risk could occur. Initially, only the risk description and owner need be recorded. The post-risk management values and plan number are entered after the appropriate actions have been approved and the risk plans formulated.

Figure 7.3 Risk register

Risk register							
Risk ID	Risk title	Risk owner	Post -risk management				Plan no.
			Prob.	Cost	Qual	Time	
1							
2							
3							
and so on.							

For each risk in the risk register, an individual **risk record** will be created (see Figure 7.4). Note that Figure 7.4 shows the probability and impacts both before and after any agreed risk management action is taken – see the lines for 'pre-risk management' and 'post-risk management'. In addition to the risk register and risk records, plans of the chosen actions need to be documented. As noted earlier, there is not necessarily a one-to-one relationship between a risk and a risk plan. An individual risk may necessitate a number of plans before the risk exposure is reduced to an acceptable level, or there may be one plan that addresses a number of identified risks.

Figure 7.4 Risk record

Risk record				
Risk ID	Risk title		Risk category	Plan no.
Risk owner	Initiation date	Earliest occurrence	Review date	Deletion date
Risk description				
Impact description				

Probability/impact values	Probability	Impact on		
		Cost	Quality	Time
Pre-risk management				
Post-risk management				

History			
Version	Date	Action	Comments

Figures 7.3 and 7.4 refer to a **risk owner**. The risk owner is responsible for ensuring adequate management of the risk, including how and at what intervals a risk will be monitored. If the nature of a risk changes during this process, it may be necessary to revise earlier risk management action plans.

7.9 SUMMARY

Managing risk is a continuous process. It involves identifying the risks and analysing them to establish their probability, impact, proximity, exposure and priority. Remedial actions need to be determined and plans produced to implement these actions, followed by scheduled monitoring and appropriate control. The whole risk management process needs to be made visible by adopting sound communication mechanisms.

Most of the risks that you will experience will already have happened on a previous project. A key aspect of risk management is learning from past projects. This is the main reason for the writing of a **lessons learnt report** at the end of a project, as mentioned back in Section 1.4.10.

SAMPLE QUESTIONS

1. **Which of the following best defines a contingency action?**
    a. An activity that is planned to take place if a risk materialises
    b. An action taken at the start of a project that reduces the potential damage if a certain risk does materialise
    c. An agreement that the users accept a particular risk
    d. An activity that prevents a risk from materialising
2. **What is maintenance of a risk log designed to do?**
    a. eliminate risk
    b. save money
    c. control risk

d. prevent system development failure

3. **Which of the lists below best identifies what is examined when a risk is assessed?**

   a. Probability, proximity, owner

   b. Impact, probability, team experience

   c. Cost, benefit, business case

   d. Probability, proximity, impact

4. **Which of the following is NOT an action that would reduce a risk?**

   a. accept it

   b. prevent it

   c. reduce it

   d. transfer it

POINTERS FOR ACTIVITIES

Activity 7.1

Among the facts that could lead to difficulties in the Water Holiday Company integration project are:

- Two sets of employees, some of which are at a high managerial level, are to be merged. There could be conflicts between the two.
- Two different holiday booking systems are to be merged. There will almost certainly be different detailed requirements relating to

the different types of boating holidays.

- More specifically, there could be incompatibilities between database structures of the systems to be merged.
- Some staff will need to be made redundant in order to gain the cost benefits of the merger. Some staff may leave of their own accord if they do not approve of changes. They may take their skills and expertise to competitors. In any case, specialist skills and knowledge could be lost.
- IT aspects of integration depend on physical prerequisites such as the completion of the new merged service centre.
- Technical issues when the integrated system is launched may lead to the system going off-line or having reduced availability. This could lose potential business.
- There is a danger of the previous customers of Canal Dreams and Minotours not being aware of the new Water Holiday Company identity.
- Possible changes in international trading relationships could affect cross-border businesses such as Minotours.

Note that these are issues of concern at the point where the first plans are being made. They might feature as ‘risks’ in the business case document. Some of the concerns will be eliminated at the planning stage through the choice of project activities that address them.

Activity 7.2

The following risks might be identified. Other risks can certainly be added.

Issue	Possible risk
There could be conflicts between the two sets of staff from Canal Dreams and Minotours	1. Differences in views on new system functionality lead to delays in reaching agreement
Two different holiday booking systems are to be merged	2. Failure to identify requirements of both sets of system users leads to an incomplete set of functions
There could also be incompatibilities between database structures of the systems to be merged	3. Failure to take account of data needs of both sets of system users leads to shortcomings in merged organisational database
Specialist skills and knowledge could be lost	4. Lack of access to knowledgeable system users leads to incorrect/incomplete assumptions about requirements
	5. Knowledgeable system users not available for acceptance testing, and other tasks associated with transfer to operational use
IT aspects of integration depend on physical prerequisites such as the completion of the new merged service centre	6. Delayed access to new premises means installation of new IT infrastructure delayed

Technical issues when the integrated system is initiated may lead to systems going off-line or having reduced availability	7. Technical issues when the integrated system is initiated 8. Lack of availability of new system impacts bookings
Previous customers of Canal Dreams and Minotours not being aware of new Water Holiday Company identity	9. Lack of links from the old businesses' websites to the new website leads to lost customers
Possible changes in international trading relationships	10. In most cases above, there are things we don't know, but we can find out. In this case, we can speculate but not find out for sure. Risk avoidance is probably the right approach (see 7.7.2)

Activity 7.3

Note that the actions below are just suggestions. You may be able to find other, perhaps more effective, risk management actions.

Risk	Possible risk management actions
1. Differences in views on new system functionality lead to delays in reaching agreement	Escalation process established that allows relatively impartial jurisdiction on disputes
2. Failure to identify requirements of both sets of	Separate business analysis of both businesses

system users leads to incomplete set of functions	Production/comparison of cross-referenced functional requirements for both organisations
3. Failure to take account of data needs of both sets of system users leads to shortcomings in merged organisational database	Separate data analysis of both businesses Production/comparison of cross-referenced data models for both organisations
4. Lack of access to knowledgeable system users lead to incorrect/incomplete assumptions about requirements	Incentives to key subject experts in both organisations with reassurances about continued employment
5. Knowledgeable system users not available for acceptance testing, and other tasks associated with transfer to operational use	
6. Delayed access to new premises means installation of new IT infrastructure delayed	Existing premises are acquired and re-purposed
7. Technical issues when the integrated system is initiated	Exhaustive system and acceptance testing
8. Lack of availability of new system affects bookings	‘Soft’ start-up of new system in the off-season for bookings Automated parallel-running of old and new systems with

		possible fall back to old systems Phased transfer to new system
9.	Lack of links from websites for old businesses to new leads to lost customers	Retain old websites with links to new one Mailing campaigns to previous customers

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)

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

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

Cover photo: iStock © Baloncici

ISBN 978-1-78017-484-6
9 781780 174846

bcs The Chartered Institute for IT

Paperback available