MODULE 4

INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT

OBJECTIVES

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

- define the scope of 'software project management';
- understand some problems and concerns of software project managers;
- define the usual stages of a software project;
- explain the main elements of the role of management;
- appreciate the need for careful planning, monitoring and control;
- identify the stakeholders of a project and their objectives;
- define the success criteria for a project.

1.1 Introduction

This textbook is about '*software* project management'. The first question is whether the management of *software* projects is really that different from that of other projects. To answer this, we need to look at some key ideas about the planning, monitoring and control of software projects. We will see that all projects are about meeting objectives. Like any other project, a software project must satisfy real needs. To do this we must identify the project's stakeholders and their objectives. Ensuring that their objectives are met is the aim of project management. However, we cannot know that a project will meet its objectives in the future unless we know the present state of the project.

1.2 Why is Software Project Management Important?

This book is for students of software engineering and computer science and also those studying business information systems. More technically oriented students can be impatient at having to study something which keeps them away from their code. So why is it important to become familiar with project management?

The information in this paragraph comes from a National Audit Office report, Improving IT Procurement, November 2004.

First, there is the question of money. A lot of money is at stake with ICT projects. In the United Kingdom during the financial year 2002–2003, the central government spent more on contracts for ICT projects than on contracts related to roads (about £2.3 billion as opposed to £1.4 billion). The biggest departmental spender was the Department for Work and Pensions, who spent over £800 million on ICT. Mismanagement of ICT projects means that there is less to spend on good things such as hospitals.

There has been some debate about the precise validity of the Standish findings but the key point about the prevalence of IT project failings remains clear.

Unfortunately, projects are not always successful. In a report published in 2003, the Standish Group in the United States analysed 13,522 projects and concluded that only a third of projects were successful; 82% of projects were late and 43% exceeded their budget.

The reason for these project shortcomings is often the management of projects. The National Audit Office in the UK, for example, among other factors causing project failure identified '*lack of skills and proven approach to project management and risk management*'.

1.3 What is a Project?

Dictionary definitions of 'project' include: 'A specific plan or design' 'A planned undertaking' 'A large undertaking: e.g. a public works scheme', *Longman Concise English Dictionary*, 1982.

The dictionary definitions put a clear emphasis on the project being a *planned* activity.

The emphasis on being planned assumes we can determine how to carry out a task before we start. Yet with exploratory projects this might be difficult. Planning is in essence thinking carefully about something before you do it – even with uncertain projects this is worth doing as long as the resulting plans are seen as provisional. Other activities, such as routine maintenance, will have been performed so many times that everyone knows exactly what to do. In these cases, planning hardly seems necessary, although procedures might be documented to ensure consistency and to help newcomers.

Programme management is often used to coordinate activities on concurrent jobs.

The activities that benefit most from conventional project management are likely to lie between these two extremes – see Figure 1.1.

There is a hazy boundary between the non-routine project and the routine job. The first time you do a routine task it will be like a project. On the other hand, a project to develop a system similar to previous ones that you have developed will have a large element of the routine.

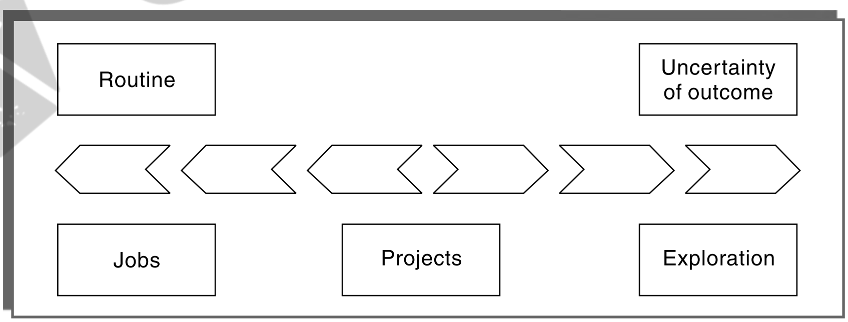

FIGURE 1.1 Activities most likely to benefit from project management

The following characteristics distinguish projects:

- non-routine tasks are involved;
- planning is required;
- specific objectives are to be met or a specified product is to be created;
- the project has a predetermined time span;
- work is carried out for someone other than yourself;
- work involves several specialisms;
- people are formed into a temporary work group to carry out the task;
- work is carried out in several phases;
- the resources that are available for use on the project are constrained;
- the project is large or complex.

The more any of these factors apply to a task, the more difficult that task will be. Project size is particularly important. The project that employs 20 developers is likely to be disproportionately more difficult than one with only 10 staff because of the need for additional coordination. The examples and exercises used in this book usually relate to smaller projects in order to make the techniques easier to grasp. However, the techniques and issues discussed are of equal relevance to larger projects.

EXERCISE 1.1

Consider the following:

- producing an edition of a newspaper;
- putting a robot vehicle on Mars to search for signs of life;
- getting married;
- amending a financial computer system to deal with a common European currency;
- a research project into what makes a good human–computer interface;
- an investigation into the reason why a user has a problem with a computer system;
- a second-year programming assignment for a computing student;
- writing an operating system for a new computer;
- installing a new version of a word processing package in an organization.

Some seem more like real projects than others. Put them into an order most closely matching your ideas of what constitutes a project. For each entry in the ordered list, describe the difference between it and the one above which makes it less worthy of the term 'project'.

There is no one correct answer to this exercise, but a possible solution to this and the other exercises you will come across may be found at the end of the book.

Some argue that projects are especially problematic as they are temporary sub-organizations. A group of people is brought together to carry out a task. The existence of this sub-organization cuts across the authority of the existing units within the organization. This has the advantage that a group containing various specialists is focused on a single important task. However, the project is likely to be seen as disruptive to

For example, see Rolf A. Lundin and Andres Söderholm (1995) 'A theory of the temporary organization' *Scandinavian Journal of Management* 11(4) 437–55.

others. Also, expertise built up during the project may be lost when the team is eventually dispersed at the end of the project.

1.4 Software Projects versus Other Types of Project

F. P. Brooks (1987). 'No silver bullet: essence and accidents of software engineering'. This essay has been included in *The Mythical Man-Month*, Anniversary Edition, Addison Wesley, 1995.

Many techniques in general project management also apply to software project management, but Fred Brooks identified some characteristics of software projects which make them particularly difficult:

Invisibility When a physical artefact such as a bridge is constructed the progress can actually be seen. With software, progress is not immediately visible. Software project management can be seen as the process of making the invisible visible.

Complexity Per dollar, pound or euro spent, software products contain more complexity than other engineered artefacts.

Conformity The 'traditional' engineer usually works with physical systems and materials like cement and steel. These physical systems have complexity, but are governed by consistent physical laws. Software developers have to conform to the requirements of human clients. It is not just that individuals can be inconsistent. Organizations, because of lapses in collective memory, in internal communication or in effective decision making, can exhibit remarkable 'organizational stupidity'.

Flexibility That software is easy to change is seen as a strength. However, where the software system interfaces with a physical or organizational system, it is expected that the software will change to accommodate the other components rather than vice versa. Thus software systems are particularly subject to change.

1.5 Contract Management and Technical Project Management

In-house projects are where the users and the developers of new software work for the same organization. However, increasingly organizations contract out ICT development to outside developers. Here, the client organization will often appoint a 'project manager' to supervise the contract who will delegate many technically oriented decisions to the contractors. Thus, the project manager will not worry about estimating the effort needed to write individual software components as long as the overall project is within budget and on time. On the supplier side, there will need to be project managers who deal with the more technical issues. This book leans towards the concerns of these 'technical' project managers.

1.6 Activities Covered by Software Project Management

Chapter 4 on project analysis and technical planning looks at some alternative life cycles.

A software project is not only concerned with the actual writing of software. In fact, where a software application is bought 'off the shelf', there may be no software writing as such, but this is still fundamentally a software project because so many of the other activities associated with software will still be present.

Usually there are three successive processes that bring a new system into being – see Figure 1.2.

1. **The feasibility study** assesses whether a project is worth starting – that it has a valid *business case*. Information is gathered about the requirements of the proposed application. Requirements elicitation can, at least initially, be complex and difficult. The stakeholders may know the aims they wish to

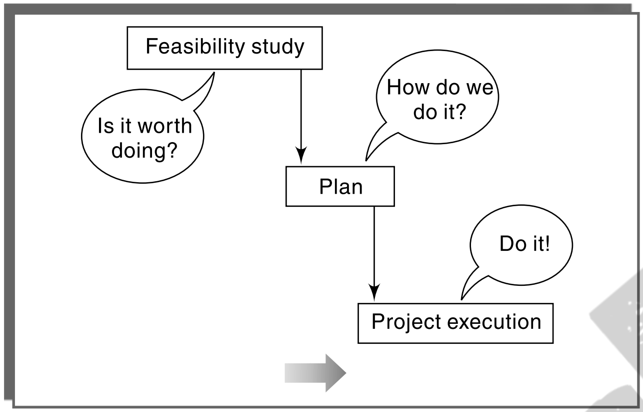

FIGURE 1.2 The feasibility study/plan/execution cycle

pursue, but not be sure about the means of achievement. The developmental and operational costs, and the value of the benefits of the new system, will also have to be estimated. With a large system, the feasibility study could be a project in its own right with its own plan. The study could be part of a strategic planning exercise examining a range of potential software developments. Sometimes an organization assesses a *programme* of development made up of a number of projects.

> Chapter 2 explores some further aspects of programme management.

2. **Planning** If the feasibility study indicates that the prospective project appears viable, then project planning can start. For larger projects, we would not do all our detailed planning at the beginning. We create an outline plan for the whole project and a detailed one for the first stage. Because we will have more detailed and accurate project information after the earlier stages of the project have been completed, planning of the later stages is left to nearer their start.

> The PRINCE2 method, which is described in Appendix A, takes this iterative approach to planning. Annex 1 to this chapter has an outline of the content of a plan.

3. **Project execution** The project can now be executed. The execution of a project often contains *design* and *implementation* sub-phases. Students new to project planning often find that the boundary between design and planning can be hazy. Design is making decisions about the form of the *products* to be created. This could relate to the external appearance of the software, that is, the user interface, or the internal architecture. The plan details the *activities* to be carried out to create these products. Planning and design can be confused because at the most detailed level, planning decisions are influenced by design decisions. Thus a software product with five major components is likely to require five sets of activities to create them.

Figure 1.3 shows the typical sequence of software development activities recommended in the international standard ISO 12207. Some activities are concerned with the *system* while others relate to *software*. The development of software will be only one part of a project. Software could be developed, for example, for a project which also requires the installation of an ICT infrastructure, the design of user jobs and user training.

> Figure 1.3 suggests that these stages must be done strictly in sequence – we will see in Chapter 4 that other, iterative approaches can be adopted. However, the actual activities listed here would still be done.

- *Requirements analysis* starts with *requirements elicitation* or requirements gathering which establishes what the potential users and their managers require of the new system. It could relate to a *function* – that the system should do something. It could be a quality requirement – how well the functions must work. An example of this is

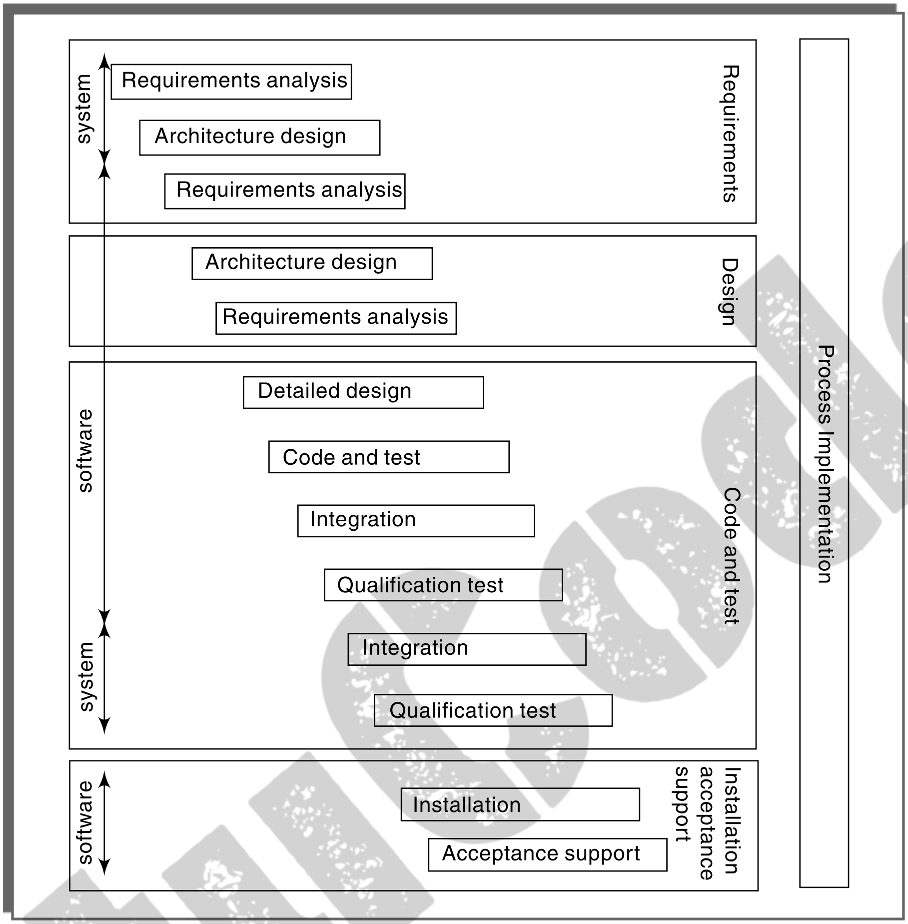

FIGURE 1.3 The ISO 12207 software development life cycle

dispatching an ambulance in response to an emergency telephone call. In this case transaction time would be affected by hardware and software performance as well as the speed of human operation. Training to ensure that operators use the computer system efficiently is an example of a *system requirement* for the project, as opposed to a specifically *software requirement*. There would also be *resource requirements* that relate to application development costs.

- *Architecture design* The components of the new system that fulfil each requirement have to be identified. Existing components may be able to satisfy some requirements. In other cases, a new component will have to be made. These components are not only software: they could be new hardware or work processes. Although software developers are primarily concerned with software components, it is very rare that these can be developed in isolation. They will, for example, have to take account of existing legacy systems with which they will interoperate. The design of the *system architecture* is thus an input to the *software requirements*. A second architecture design process then takes place that maps the software requirements to *software components*.
- *Detailed design* Each software component is made up of a number of software units that can be separately coded and tested. The detailed design of these units is carried out separately.

- *Code and test* refers to writing code for each software unit. Initial testing to debug individual software units would be carried out at this stage.
- *Integration* The components are tested together to see if they meet the overall requirements. Integration could involve combining different software components, or combining and testing the software element of the system in conjunction with the hardware platforms and user interactions.
- *Qualification testing* The system, including the software components, has to be tested carefully to ensure that all the requirements have been fulfilled.
- *Installation* This is the process of making the new system operational. It would include activities such as setting up standing data (for example, the details for employees in a payroll system), setting system parameters, installing the software onto the hardware platforms and user training.
- *Acceptance support* This is the resolving of problems with the newly installed system, including the correction of any errors, and implementing agreed extensions and improvements. Software maintenance can be seen as a series of minor software projects. In many environments, most software development is in fact maintenance.

EXERCISE 1.2

Brightmouth College is a higher education institution which used to be managed by a local government authority but has now become autonomous. Its payroll is still administered by the local authority and pay slips and other output are produced in the local authority's computer centre. The authority now charges the college for this service. The college management are of the opinion that it would be cheaper to obtain an 'off-the shelf' payroll package and do the payroll processing themselves.

What would be the main stages of the project to convert to independent payroll processing by the college? Bearing in mind that an off-the-shelf package is to be used, how would this project differ from one where the software was to be written from scratch?

EXERCISE 1.3

Assume that a software organization development has been asked to carry out a feasibility study to develop the payroll package for Brightmouth College. The development organization plans to develop the software by customizing one of its existing products. What are the main steps through which the project manager of the organization would carry out the feasibility study?

1.7 Plans, Methods and Methodologies

A plan for an activity must be based on some idea of a *method* of work. For example, if you were asked to test some software, you may know nothing about the software to be tested, but you could assume that you would need to:

- analyse the requirements for the software;
- devise and write test cases that will check that each requirement has been satisfied;

- create test scripts and expected results for each test case;
- compare the actual results and the expected results and identify discrepancies.

While a *method* relates to a type of activity in general, a *plan* takes that method (and perhaps others) and converts it to real activities, identifying for each activity:

- its start and end dates;
- who will carry it out;
- what tools and materials – including information – will be needed.

The output from one method might be the input to another. Groups of methods or techniques are often grouped into *methodologies* such as object-oriented design.

EXERCISE 1.4

This should ideally be done in groups of about four, but you can think about how you would go about this exercise on your own if needs be. You are probably in a building that has more than one storey. From the point of view of this exercise, the bigger the building the better.

In a group of four, work out how you would obtain an accurate estimate of the height of the building. (If you happen to be in a single-storey building, you can estimate the floor area instead!) Plan how you would carry out any actions needed to obtain your estimate. Spend 20 minutes on this – you must remain in the same room for this planning phase. Once planning is complete, implement your plan, timing how long it takes to produce your final figure.

If there is more than one group carrying out this exercise, after completion of the task you can compare answers and also the approach you used when coming up with your answer.

1.8 Some Ways of Categorizing Software Projects

Projects may differ because of the different technical products to be created. Thus we need to identify the characteristics of a project which could affect the way in which it should be planned and managed. Other factors are discussed below.

Compulsory versus voluntary users

In workplaces there are systems that staff have to use if they want to do something, such as recording a sale. However, use of a system is increasingly voluntary, as in the case of computer games. Here it is difficult to elicit precise requirements from potential users as we could with a business system. What the game will do will thus depend much on the informed ingenuity of the developers, along with techniques such as market surveys, focus groups and prototype evaluation.

Information systems versus embedded systems

Embedded systems are also called real-time or industrial systems.

A traditional distinction has been between *information systems* which enable staff to carry out office processes and *embedded systems* which control machines. A stock control system would be an information system. An embedded, or process control, system might control the air conditioning equipment in a building. Some systems may have elements of both where, for example, the stock control system also controls an automated warehouse.

EXERCISE 1.5

Would an operating system on a computer be an information system or an embedded system?

Outsourced projects

While developing a large project, sometimes, it makes good commercial sense for a company to outsource some parts of its work to other companies. There can be several reasons behind such a decision. For example, a company may consider outsourcing as a good option, if it feels that it does not have sufficient expertise to develop some specific parts of the product or if it determines that some parts can be developed cost-effectively by another company. Since an outsourced project is a small part of some project, it is usually small in size and needs to be completed within a few months. Considering these differences between an outsourced project and a conventional project, managing an outsourced project entails special challenges.

Indian software companies excel in executing outsourced software projects and have earned a fine reputation in this field all over the world. Of late, the Indian companies have slowly begun to focus on product development as well.

The type of development work being handled by a company can have an impact on its profitability. For example, a company that has developed a generic software product usually gets an uninterrupted stream of revenue over several years. However, outsourced projects fetch only one time revenue to any company.

Objective-driven development

Service level agreements are becoming increasingly important as organizations contract out functions to external service suppliers.

Projects may be distinguished by whether their aim is to produce a *product* or to meet certain *objectives.*

A project might be to create a product, the details of which have been specified by the client. The client has the responsibility for justifying the product.

On the other hand, the project requirement might be to meet certain objectives which could be met in a number of ways. An organization might have a problem and ask a specialist to recommend a solution.

Many software projects have two stages. First is an objective-driven project resulting in recommendations. This might identify the need for a new software system. The next stage is a project actually to create the software product.

This is useful where the technical work is being done by an external group and the user needs are unclear at the outset. The external group can produce a preliminary design at a fixed fee. If the design is acceptable the developers can then quote a price for the second, implementation, stage based on an agreed requirement.

EXERCISE 1.6

Would the project, to implement an independent payroll system at the Brightmouth College described in Exercise 1.2, above, be an objective-driven project or a product-driven project?

1.9 Stakeholders

These are people who have a stake or interest in the project. Their early identification is important as you need to set up adequate communication channels with them. Stakeholders can be categorized as:

- *Internal to the project team* This means that they will be under the direct managerial control of the project leader.
- *External to the project team but within the same organization* For example, the project leader might need the assistance of the users to carry out systems testing. Here the commitment of the people involved has to be negotiated.
- *External to both the project team and the organization* External stakeholders may be customers (or users) who will benefit from the system that the project implements. They may be contractors who will carry out work for the project. The relationship here is usually based on a contract.

B.W. Boehm and R. Ross, 'Theory W software project management: principles and examples', in B. W. Boehm (ed.) (1989) Software Risk Management, IEEE Computer Society Press.

Different types of stakeholder may have different objectives and one of the jobs of the project leader is to recognize these different interests and to be able to reconcile them. For example, end-users may be concerned with the ease of use of the new application, while their managers may be more focused on staff savings. The project leader therefore needs to be a good communicator and negotiator. Boehm and Ross proposed a 'Theory W' of software project management where the manager concentrates on creating situations where all parties benefit from a project and therefore have an interest in its success. (The 'W' stands for 'win–win'.)

The role and format of communication plans will be explained in greater detail in Chapter 11 on managing people in software environments.

Project managers can sometimes miss an important stakeholder group, especially in unfamiliar business contexts. These could be departments supplying important services that are taken for granted.

Given the importance of coordinating the efforts of stakeholders, the recommended practice is for a *communication plan* to be created at the start of a project.

EXERCISE 1.7

Identify the stakeholders in the Brightmouth College payroll project.

1.10 Setting Objectives

Among all these stakeholders are those who actually own the project. They control the financing of the project. They also set the objectives of the project. The objectives should define what the project team must achieve for project success. Although different stakeholders have different motivations, the project objectives identify the shared intentions for the project.

Objectives focus on the desired outcomes of the project rather than the tasks within it – they are the 'post-conditions' of the project. Informally the objectives could be written as a set of statements following the opening words '*the project will be a success if. . . .*' Thus one statement in a set of objectives might be '*customers can order our products online*' rather than '*to build an e-commerce website*'. There is often more than one way to meet an objective and the more possible routes to success the better.

There may be several stakeholders, including users in different business areas, who might have some claim to project ownership. In such a case, a *project authority* needs to be explicitly identified with overall authority over the project.

This authority is often a *project steering committee* (or *project board* or *project management board*) with overall responsibility for setting, monitoring and modifying objectives. The project manager runs the project on a day-to-day basis, but regularly reports to the steering committee.

> This committee is likely to contain user, development and management representatives.

Sub-objectives and goals

An effective objective for an individual must be something that is within the control of that individual. An objective might be that the software application produced must pay for itself by reducing staff costs. As an overall business objective this might be reasonable. For software developers it would be unreasonable as any reduction in operational staff costs depends not just on them but on the operational management of the delivered system. A more appropriate *goal* or sub-objective for the software developers would be to keep development costs within a certain budget.

> Defining sub-objectives requires assumptions about how the main objective is to be achieved.

We can say that in order to achieve the objective we must achieve certain goals or sub-objectives first. These are steps on the way to achieving an objective, just as goals scored in a football match are steps towards the objective of winning the match. Informally this can be expressed as a set of statements following the words '*To reach objective. . ., the following must be in place. . .*'.

The mnemonic SMART is sometimes used to describe well-defined objectives:

- *Specific* Effective objectives are concrete and well defined. Vague aspirations such as '*to improve customer relations*' are unsatisfactory. Objectives should be defined so that it is obvious to all whether the project has been successful.
- *Measurable* Ideally there should be *measures of effectiveness* which tell us how successful the project has been. For example, '*to reduce customer complaints*' would be more satisfactory as an objective than '*to improve customer relations*'. The measure can, in some cases, be an answer to simple yes/no question, e.g. '*Did we install the new software by 1 June?*'
- *Achievable* It must be within the power of the individual or group to achieve the objective.
- *Relevant* The objective must be relevant to the true purpose of the project.
- *Time constrained* There should be a defined point in time by which the objective should have been achieved.

> This still leaves a problem about the level at which the target should be set, e.g. why, say, a 50% reduction in complaints and not 40% or 60%?

EXERCISE 1.8

Bearing in mind the above discussion of objectives, comment on the appropriateness of the wording of each of the following 'objectives' for software developers:

(i) to implement the new application on time and within budget;

(ii) to implement the new software application with the fewest possible software errors that might lead to operational failures;

(iii) to design a system that is user-friendly;
(iv) to produce full documentation for the new system.

Measures of effectiveness

These concepts are explained more fully in Chapter 13 on software quality.

Measures of effectiveness provide practical methods of checking that an objective has been met. 'Mean time between failures' (mtbf) might, for example, be used to measure reliability. This is a *performance* measurement and, as such, can only be taken once the system is operational. Project managers want to get some idea of the performance of the completed system as it is being constructed. They will therefore seek *predictive* measures. For example, a large number of errors found during code inspections might indicate potential problems with reliability later.

EXERCISE 1.9

Identify the objectives and sub-objectives of the Brightmouth College payroll project. What measures of effectiveness could be used to check the success in achieving the objectives of the project?

1.11 The Business Case

The business case should be established at the time of the project's feasibility study. Chapter 2 explains the idea of a business case in more detail.

Most projects need to have a justification or business case: the effort and expense of pushing the project through must be seen to be worthwhile in terms of the benefits that will eventually be felt. A cost–benefit analysis will often be part of the project's feasibility study. This will itemize and quantify the project's costs and benefits. The benefits will be affected by the completion date: the sooner the project is completed, the sooner the benefits can be experienced. The quantification of benefits will often require the formulation of a *business model* which explains how the new application can generate the claimed benefits.

A simple example of a business model is that a new web-based application might allow customers from all over the world to order a firm's products via the internet, increasing sales and thus increasing revenue and profits.

Any project plan must ensure that the business case is kept intact. For example:

- that development costs are not allowed to rise to a level which threatens to exceed the value of benefits;
- that the features of the system are not reduced to a level where the expected benefits cannot be realized;
- that the delivery date is not delayed so that there is an unacceptable loss of benefits.

1.12 Project Success and Failure

The project plan should be designed to ensure project success by preserving the business case for the project. However, every non-trivial project will have problems, and at what stage do we say that a project is actually a failure? Because different stakeholders have different interests, some stakeholders in a project might see it as a success while others do not.

Broadly speaking, we can distinguish between *project objectives* and *business objectives*. The project objectives are the targets that the project team is expected to achieve. In the case of software projects, they can usually be summarized as delivering:

- the agreed functionality
- to the required level of quality
- on time
- within budget.

A good introduction to the issues discussed here can be found in A. J. Shenhar and O. Levy (1997) 'Mapping the dimensions of project success' *Project Management Journal* 28(2) 9–12.

A project could meet these targets but the application, once delivered could fail to meet the business case. A computer game could be delivered on time and within budget, but might then not sell. A commercial website used for online sales could be created successfully, but customers might not use it to buy products, because they could buy the goods more cheaply elsewhere.

We have seen that in business terms it can generally be said that a project is a success if the value of benefits exceeds the costs. We have also seen that while project managers have considerable control over development costs, the value of the benefits of the project deliverables is dependent on external factors such as the number of customers. Project objectives still have some bearing on eventual business success. As we will see in Chapter 2, increasing development costs reduce the chances of the delivered product being profitable. A delay in completion reduces the amount of time during which benefits can be generated and diminishes the value of the project.

The assessment of the value of project benefits is explored in greater depth in Chapter 2.

A project can be a success on delivery but then be a business failure, On the other hand, a project could be late and over budget, but its deliverables could still, over time, generate benefits that outweigh the initial expenditure.

Some argue that the possible gap between project and business concerns can be reduced by having a broader view of projects that includes business issues. For example, the project management of an e-commerce website implementation could plan activities such as market surveys, competitor analysis, focus groups, prototyping, and evaluation by typical potential users – all designed to reduce business risks.

Because the focus of project management is, not unnaturally, on the immediate project, it may not be seen that the project is actually one of a sequence. Later projects benefit from the technical skills learnt on earlier projects. Technical learning will increase costs on the earlier projects, but later projects benefit as the learnt technologies can be deployed more quickly, cheaply and accurately. This expertise is often accompanied by additional software assets, for example reusable code. Where software development is outsourced, there may be immediate savings, but these longer-term benefits of increased expertise will be lost. Astute managers may assess which areas of technical expertise it would be beneficial to develop.

For a wider discussion of the relationships between successive projects, see M. Engwall (2003) 'No project is an island: linking projects to history and context' Research Policy 32 789–808.

Customer relationships can also be built up over a number of projects. If a client has trust in a supplier who has done satisfactory work in the past, they are more likely to use that company again, particularly if the new requirement builds on functionality already delivered. It is much more expensive to acquire new clients than it is to retain existing ones.

1.13 What is Management?

We have explored some of the special characteristics of software. We now look at the 'management' aspect of software project management. It has been suggested that management involves the following activities:

- planning – deciding what is to be done;
- organizing – making arrangements;
- staffing – selecting the right people for the job etc.;
- directing – giving instructions;
- monitoring – checking on progress;
- controlling – taking action to remedy hold-ups;
- innovating – coming up with new solutions;
- representing – liaising with clients, users, developer, suppliers and other stakeholders.

EXERCISE 1.10

Paul Duggan is the manager of a software development section. On Tuesday at 10.00 a.m. he and his fellow section heads have a meeting with their group manager about the staffing requirements for the coming year. Paul has already drafted a document 'bidding' for staff. This is based on the work planned for his section for the next year. The document is discussed at the meeting. At 2.00 p.m. Paul has a meeting with his senior staff about an important project his section is undertaking. One of the programming staff has just had a road accident and will be in hospital for some time. It is decided that the project can be kept on schedule by transferring another team member from less urgent work to this project. A temporary replacement is to be brought in to do the less urgent work but this may take a week or so to arrange. Paul has to phone both the human resources manager about getting a replacement and the user for whom the less urgent work is being done, explaining why it is likely to be delayed.

Identify which of the eight management responsibilities listed above Paul was responding to at different points during his day.

Much of the project manager's time is spent on only three of the eight identified activities, viz., project planning, monitoring, and control. The time period during which these activities are carried out is indicated in Fig. 1.4. It shows that project management is carried out over three well-defined stages or processes, irrespective of the methodology used. In the project initiation stage, an initial plan is made. As the project starts, the project is monitored and controlled to proceed as planned. However, the initial plan is revised periodically to accommodate additional details and constraints about the project as they become available. Finally, the project is closed. In the project closing stage, all activities are logically completed and all contracts are formally closed.

Initial project planning is undertaken immediately after the feasibility study phase and before starting the requirements analysis and specification process. Figure 1.4 shows this project initiation period. Initial project planning involves estimating several characteristics of a project. Based on these estimates, all subsequent project activities are planned. The initial project plans are revised periodically as the project progresses and more project data becomes available. Once the project execution starts, monitoring and control activities are taken up to ensure that the project execution proceeds as planned. The monitoring activity involves monitoring the progress of the project. Control activities are initiated to minimize any significant variation in the plan.

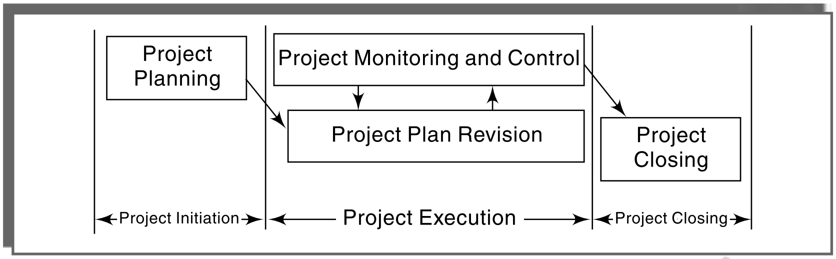

FIGURE 1.4 Principal project management processes

Project planning is an important responsibility of the project manager. During project planning, the project manager needs to perform a few well-defined activities that have been outlined below. Note that we have given a very brief description of these activities in this chapter. We will discuss these activities in more detail in subsequent chapters. Several best practices have been proposed for software project planning activities. In Chapter 3 we will discuss Step Wise, which is based on the popular PRINCE2 (PRojects IN Controlled Environments) method. While PRINCE2 is used extensively in the UK and Europe, similar software project management best practices have been put forward in the USA by the Project Management Institute's 'PMBOK' which refers to their publication 'A Guide to the Project Management Body of Knowledge.'

- *Estimation* The following project attributes are estimated.
- *Cost* How much is it going to cost to complete the project?
- *Duration* How long is it going to take to complete the project?
- *Effort* How much effort would be necessary for completing the project?

The effectiveness of all activities such as scheduling and staffing, which are planned at a later stage, depends on the accuracy with which the above three project parameters have been estimated.

- *Scheduling* Based on estimations of effort and duration, the schedules for manpower and other resources are developed.
- *Staffing* Staff organization and staffing plans are made.
- *Risk Management* This activity includes risk identification, analysis, and abatement planning.
- *Miscellaneous Plans* This includes making several other plans such as quality assurance plan, configuration management plan, etc.

Project monitoring and control activities are undertaken after the initiation of development activities. The aim of project monitoring and control activities is to ensure that the software development proceeds as planned. While carrying out project monitoring and control activities, a project manager may sometimes find it necessary to change the plan to cope with specific situations and make the plan more accurate as more project data becomes available.

At the start of a project, the project manager does not have complete knowledge about the details of the project. As the project progresses through different development phases, the manager's information base gradually improves. The complexities of different project activities become clear, some of the anticipated risks get resolved, and new risks appear. The project parameters are re-estimated periodically incorporating new understanding and change in project parameters. By taking these developments into account, the project manager can plan subsequent activities more accurately with increasing levels of confidence. Figure 1.4 shows this aspect as iterations between monitoring and control, and the plan revision activities.

1.14 Management Control

Management, in general, involves setting objectives for a system and then monitoring the performance of the system. In Figure 1.5 the 'real world' is shown as being rather formless. Especially in the case of large undertakings, there will be a lot going on about which management should be aware.

EXERCISE 1.11

An ICT project is to replace locally held paper-based records with a centrally organized database. Staff in a large number of offices that are geographically dispersed need training and will then have to use the new ICT system to set up the backlog of manual records on the new database. The system cannot be properly operational until the last record has been transferred. The new system will only be successful if new transactions can be processed within certain time cycles.

Identify the data that you would collect to ensure that during execution of the project things were going to plan.

This will involve the local managers in *data collection*. Bare details, such as 'location X has processed 2000 documents', will not be very useful to higher management: *data processing* will be needed to transform this raw *data* into useful *information*. This might be in such forms as 'percentage of records processed', 'average documents processed per day per person' and 'estimated completion date'.

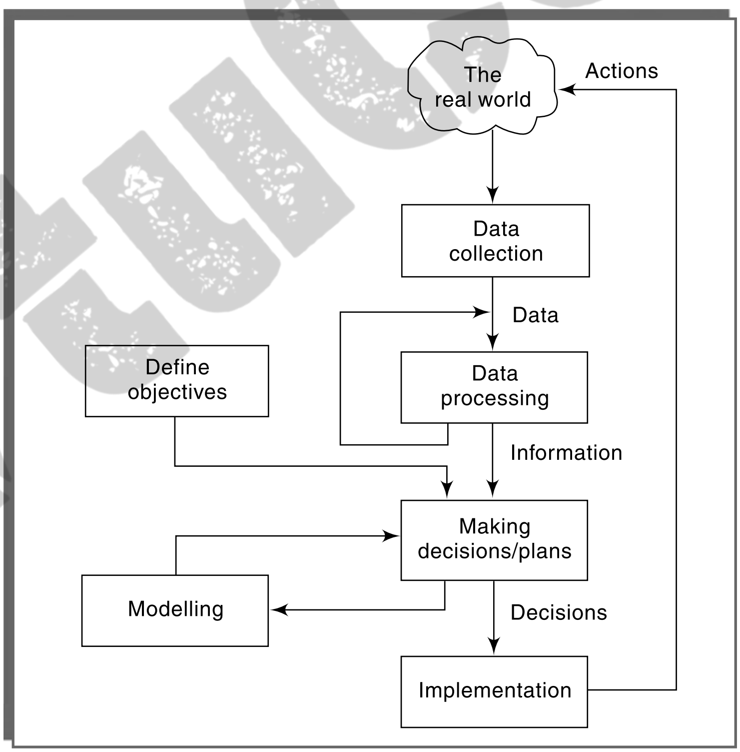

FIGURE 1.5 The project control cycle

In our example, the project management might examine the 'estimated completion date' for completing data transfer for each branch. These can be checked against the overall target date for completion of this phase of the project. In effect they are comparing actual performance with one aspect of the overall project objectives. They might find that one or two branches will fail to complete the transfer of details in time. They would then need to consider what to do (this is represented in Figure 1.5 by the box *Making decisions/ plans*). One possibility would be to move staff temporarily from one branch to another. If this is done, there is always the danger that while the completion date for the one branch is pulled back to before the overall target date, the date for the branch from which staff are being moved is pushed forward beyond that date. The project manager would need to calculate carefully what the impact would be in moving staff from particular branches. This is *modelling* the consequences of a potential solution. Several different proposals could be modelled in this way before one was chosen for *implementation.*

Having implemented the decision, the situation needs to be kept under review by collecting and processing further progress details. For instance, the next time that progress is reported, a branch to which staff have been transferred could still be behind in transferring details. This might be because the reason why the branch has got behind in transferring details is because the manual records are incomplete and another department, for whom the project has a low priority, has to be involved in providing the missing information. In this case, transferring extra staff to do data inputting will not have accelerated data transfer.

It can be seen that a project plan is dynamic and will need constant adjustment during the execution of the project. Courses and books on project management (such as this one) often focus considerable attention on project planning. While this is to be expected, with nearly all projects much more time is spent actually doing the project rather than planning it. A good plan provides a foundation for a good project, but is nothing without intelligent execution. The original plan will not be set in stone but will be modified to take account of changing circumstances.

1.15 Traditional versus Modern Project Management Practices

Over the last two decades, the basic approach taken by the software industry to develop software has undergone a radical change. Hardly any software is being developed from scratch any more. Software development projects are increasingly being based on either tailoring some existing product or reusing certain pre-built libraries. In either case, two important goals of recent life cycle models are maximization of code reuse and compression of project durations. Other goals include facilitating and accommodating client feedbacks and customer participation in project development work, and incremental delivery of the product with evolving functionalities. Change requests from customers are encouraged, rather than circumvented. Clients on the other hand, are demanding further reductions in product delivery times and costs. These recent developments have changed project management practices in many significant ways. In the following section, we will discuss some important differences between modern project management practices and traditional practices.

- *Planning Incremental Delivery* Few decades ago, projects were much simpler and therefore more predictable than the present day projects. In those days, projects were planned with sufficient detail, much before the actual project execution started. After the project initiation, monitoring and control activities were carried out to ensure that the project execution proceeded as per plan. Now, projects are required to be completed over a much shorter duration, and rapid application development and deployment are considered key strategies. The traditional long-term planning has given way to adaptive short-term planning. Instead of making a long-term project completion plan, the project manager now plans all incremental deliveries with evolving functionalities. This type of project management is often called extreme project management. Extreme project management is a highly flexible approach to project management that concentrates on the human side of project management (e.g., managing project stakeholders), rather than formal and complex planning and monitoring techniques.

- *Quality Management* Of late, customer awareness about product quality has increased significantly. Tasks associated with quality management have become an important responsibility of the project manager. The key responsibilities of a project manager now include assessment of project progress and tracking the quality of all intermediate artifacts. We will discuss quality management issues in Chapter 13.
- *Change Management* Earlier, when the requirements were signed off by the customer, any changes to the requirements were rarely entertained. Customer suggestions are now actively being solicited and incorporated throughout the development process. To facilitate customer feedback, incremental delivery models are popularly being used. Product development is being carried out through a series of product versions implementing increasingly greater functionalities. Also customer feedback is solicited on each version for incorporation. This has made it necessary for an organization to keep track of the various versions and revisions through which the product develops. Another reason for the increased importance of keeping track of the versions and revisions is the following. Application development through customization has become a popular business model. Therefore, existence of a large number of versions of a product and the need to support these by a development organization has become common. In this context, the project manager plays a key role in product base lining and version control. This has made change management a crucial responsibility of the project manager. Change management is also known as configuration management. We will discuss change management in Chapter 9.

EXERCISE 1.12

Assume that the development of the pay roll package of Brightmouth College has been entrusted to an organization who would develop it by customizing one of its products. Discuss the main stages through which the organization could carry out project development?

CONCLUSION

This chapter has laid a foundation for the remainder of the book by defining what is meant by various terms such as 'software project' and 'management'. Among some of the more important points that have been made are the following:

- Projects are by definition non-routine and therefore more uncertain than normal undertakings.
- Software projects are similar to other projects but have some attributes that present particular difficulties, e.g. the relative invisibility of many of their products.
- A key factor in project success is having clear objectives. Different stakeholders in a project, however, are likely to have different objectives. This points to the need for a recognized overall project authority.
- For objectives to be effective there must be practical ways of testing that the objectives have been met.
- Where projects involve many different people, effective channels of information have to be established. Having objective measures of success helps unambiguous communication between the various parties to a project.

2.4 Evaluation of Individual Projects

We will now look more closely at how the feasibility of an individual project can be evaluated.

Technical assessment

Technical assessment of a proposed system consists of evaluating whether the required functionality can be achieved with current affordable technologies. Organizational policy, aimed at providing a consistent hardware/software infrastructure, is likely to limit the technical solutions considered. The costs of the technology adopted must be taken into account in the cost–benefit analysis.

Cost–benefit analysis

Any project aiming at a return on investment must, as a minimum, provide a greater benefit than putting that investment in, say, a bank.

Even where the estimated benefits will exceed the estimated costs, it is often necessary to decide if the proposed project is the best of several options. Not all projects can be undertaken at any one time and, in any case, the most valuable projects should get most resources.

Cost–benefit analysis comprises two steps:

- *Identifying all of the costs and benefits of carrying out the project and operating the delivered application* These include the development costs, the operating costs, and the benefits expected from the new system. Where the proposed system is a replacement, these estimates should reflect the change in costs and benefits due to the new system. A new sales order processing system, for example, could only claim to benefit an organization by the increase in sales due to the use of the new system.
- *Expressing these costs and benefits in common units* We must express each cost and benefit – and the *net benefit* which is the difference between the two – in money.

Most direct costs are easy to quantify in monetary terms and can be categorized as:

The different types of benefits will be discussed in greater detail in the context of benefits to management later in this chapter.

- *Development costs*, including development staff costs.
- *Setup costs*, consisting of the costs of putting the system into place, mainly of any new hardware but also including the costs of file conversion, recruitment and staff training.
- *Operational costs* relating to operating the system after installation.

Exercise 2.1

Brightmouth College is considering the replacement of the existing payroll service, operated by a third party, with a tailored, off-the-shelf computer-based system. List some of the costs it might consider under the headings of:

- Development costs
- Setup costs
- Operational costs

List some of the benefits under the headings:

- Quantified and valued benefits
- Quantified but not valued
- Identified but not easily valued

For each cost or benefit, explain how, in principle, it might be measured in monetary terms.

Cash flow forecasting

Typically products generate a negative cash flow during their development followed by a positive cash flow over their operating life. There might be decommissioning costs at the end of a product's life.

As important as estimating the overall costs and benefits of a project is producing a cash flow forecast which indicates when expenditure and income will take place (Figure 2.1).

We need to spend money, such as staff wages, during a project's development. Such expenditure cannot wait until income is received (either from using software developed in-house use or from selling it). We need to know that we can fund this development expenditure either from the company's own resources or by borrowing. A forecast is needed of when expenditure, such as the payment of salaries, and any income are to be expected.

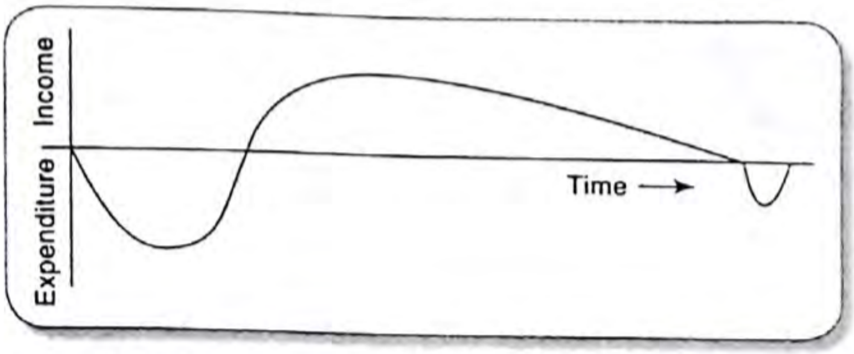

FIGURE 2.1 Typical product life cycle cash flow

Accurate cash flow forecasting is difficult, as it is done early in the project's life cycle (at least before any significant expenditure is committed) and many items to be estimated (particularly the benefits of using software) might be some years in the future.

When estimating future cash flows, it is usual to ignore the effects of inflation. Forecasts of inflation rates tend to be uncertain. Moreover, if expenditure is increased due to inflation it is likely that income will increase proportionately.

The difficulty and importance of cash flow forecasting is evidenced by the number of companies that suffer bankruptcy because, although they are developing profitable products or services, they cannot sustain an unplanned negative cash flow.

2.5 Cost–benefit Evaluation Techniques

We now take a look at some methods for comparing projects on the basis of their cash flow forecasts.

Table 2.1 illustrates cash flow forecasts for four projects. In each case, it is assumed that the cash flows take place at the end of each year. For short-term projects or where there are significant seasonal cash flow patterns, quarterly, or even monthly, cash flow forecasts could be appropriate.

TABLE 2.1 Four project cash flow projections – figures are end of year totals (£)

Year	Project 1	Project 2	Project 3	Project 4
0	–100,000	–1,000,000	–100,000	–120,000
1	10,000	200,000	30,000	30,000
2	10,000	200,000	30,000	30,000
3	10,000	200,000	30,000	30,000
4	20,000	200,000	30,000	30,000
5	100,000	300,000	30,000	75,000
Net profit	50,000	100,000	50,000	75,000

Exercise 2.2

Consider the project cash flow estimates for four projects at JOE shown in Table 2.1. Negative values represent expenditure and positive values income.

Rank the four projects in order of financial desirability and make a note of your reasons for ranking them in that way before reading further.

Net profit → 2

The net profit of a project is the difference between the total costs and the total income over the life of the project. Project 2 in Table 2.1 shows the greatest net profit but this is at the expense of a large investment. Indeed, if we had £1 m to invest, we might undertake all of the other three projects and obtain an even greater net profit. Note also that all projects contain an element of risk and we might not be prepared to risk £1 m. We shall look at the effects of risk and investment later in this chapter.

Cash flows take place at the end of each year. The year 0 represents the initial investment made at the start of the project.

Moreover, the simple net profit takes no account of the timing of the cash flows. Projects 1 and 3 each have a net profit of £50,000 and therefore, according to this selection criterion, would be equally preferable. The bulk of the income occurs late in the life of project 1, whereas project 3 returns a steady income throughout its life. Having to wait for a return has the disadvantage that the investment must be funded for longer. Add to that the fact that, other things being equal, estimates in the more distant future are less reliable than short-term estimates and we can see that the two projects are not equally preferable.

Payback period → 3*, 4

The *payback period* is the time taken to break even or pay back the initial investment. Normally, the project with the shortest payback period will be chosen on the basis that an organization will wish to minimize the time that a project is 'in debt'.

Exercise 2.3

Consider the four project cash flows given in Table 2.1 and calculate the payback period for each of them.

The advantage of the payback period is that it is simple to calculate and is not particularly sensitive to small forecasting errors. Its disadvantage as a selection technique is that it ignores the overall profitability of the project – in fact, it totally ignores any income (or expenditure) once the project has broken even. Thus the fact that projects 2 and 4 are, overall, more profitable than project 3 is ignored.

Return on investment → 4

The *return on investment* (ROI), also known as the *accounting rate of return* (ARR), provides a way of comparing the net profitability to the investment required. There are some variations on the formula used to calculate the return on investment but a straightforward common version is:

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

Exercise 2.4

Calculating the ROI for project 1, the net profit is £50,000 and the total investment is £100,000. The return on investment is therefore calculated as

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

$$= \frac{50{,}000/5}{100{,}000} \times 100 = 10\%$$

1. 10%
2. 2%
3. 10%
4. 12.5%

Calculate the ROI for each of the other projects shown in Table 2.1 and decide which, on the basis of this criterion, is the most worthwhile.

The return on investment provides a simple, easy-to-calculate measure of return on capital. Unfortunately, it suffers from two severe disadvantages. Like the net profitability, it takes no account of the timing of the cash flows. More importantly, this rate of return bears no relationship to the interest rates offered or charged by banks (or any other normal interest rate) since it takes no account of the timing of the cash flows or of the compounding of interest. It is therefore, potentially, very misleading.

Net present value

The calculation of *net present value* is a project evaluation technique that takes into account the profitability of a project and the timing of the cash flows that are produced. This is based on the view that receiving £100 today is better than having to wait until next year to receive it. We could, for example, invest the £100 in a bank today and have £100 plus the interest in a year's time. If we say that the *present value* of £100 in a year's time is £91, we mean that £100 in a year's time is the equivalent of £91 now.

> Net present value (NPV) and internal rate of return (IRR) are collectively known as discounted cash flow (DCF) techniques.

The equivalence of £91 now and £100 in a year's time means we are discounting the future income by approximately 10%. If we received £91 now and invested it for a year at an annual interest rate of 10%, it would be worth £100 in a year's time. The annual rate by which we discount future earnings is known as the *discount rate* – 10% in the above example.

> Note that this example uses approximate figures.

Similarly, £100 received in two years' time would have a present value of approximately £83 – in other words, £83 invested at an interest rate of 10% would yield approximately £100 in two years' time.

> A rate of 10% may be unrealistic but is used here for ease of calculation.

The present value of any future cash flow may be obtained by applying the following formula

$$Present\ value = \frac{value\ in\ year\ t}{(1+r)^t}$$

where r is the discount rate, expressed as a decimal value, and t is the number of years into the future that the cash flow occurs.

1yr = $\frac{1}{(1+0.10)} = 0.9091$ 2yr = $\frac{1}{(1+0.10)^2} = 0.826$

More extensive or detailed tables may be constructed using the formula *discount factor* $\frac{1}{(1+r)^t}$ for various values of *r* (the discount rate) and *t* (the number of years from now).

Alternatively, and rather more easily, the present value of a cash flow may be calculated by multiplying the cash flow by the appropriate discount factor. A small table of discount factors is given in Table 2.2.

The NPV for a project is obtained by discounting each cash flow (both negative and positive) and summing the discounted values. It is normally assumed that any initial investment takes place immediately (indicated as year 0) and is not discounted. Later cash flows are normally assumed to take place at the end of each year and are discounted by the appropriate amount.

TABLE 2.2 NPV discount factors

Year	Discount rate (%)					
	5	6	8	10	12	15
1	0.9524	0.9434	0.9259	0.9091	0.8929	0.8696
2	0.9070	0.8900	0.8573	0.8264	0.7972	0.7561
3	0.8638	0.8396	0.7938	0.7513	0.7118	0.6575
4	0.8227	0.7921	0.7350	0.6830	0.6355	0.5718
5	0.7835	0.7473	0.6806	0.6209	0.5674	0.4972
6	0.7462	0.7050	0.6302	0.5645	0.5066	0.4323
7	0.7107	0.6651	0.5835	0.5132	0.4523	0.3759
8	0.6768	0.6274	0.5403	0.4665	0.4039	0.3269
9	0.6446	0.5919	0.5002	0.4241	0.3606	0.2843
10	0.6139	0.5584	0.4632	0.3855	0.3220	0.2472
15	0.4810	0.4173	0.3152	0.2394	0.1827	0.1229
20	0.3769	0.3118	0.2145	0.1486	0.1037	0.0611
25	0.2953	0.2330	0.1460	0.0923	0.0588	0.0304

Exercise 2.5

Assuming a 10% discount rate, the NPV for project 1 (Table 2.1) would be calculated as in Table 2.3. The net present value for project 1, using a 10% discount rate, is therefore £618. Using a 10% discount rate, calculate the net present values for projects 2, 3 and 4 and decide which, on the basis of this, is the most beneficial to pursue.

Table 2.3 Applying the discount factors to project 1

Year	Project 1 cash flow (£)		Discount factor @ 10%	Discounted cash flow (£)	
0	–100,000	- 100,000	1.0000	–100,000	-100000
1	10,000	30 000	0.9091	9,091	27273
2	10,000	30000	0.8264	8,264	24792
3	10,000	30000	0.7513	7,513	22539
4	20,000	30000	0.6830	13,660	20490
5	100,000	30000	0.6209	62,090	18627
Net Profit	£50,000		NPV: £618		13721

It is interesting to note that the net present values for projects 1 and 3 are significantly different – even though they both yield the same net profit and both have the same return on investment. The difference in NPV reflects the fact that, with project 1, we must wait longer for the bulk of the income.

The main difficulty with NPV for deciding between projects is selecting an appropriate discount rate. Some organizations have a standard rate but, where this is not the case, then the discount rate should be chosen to reflect available interest rates (borrowing costs where the project must be funded from loans) plus some premium to reflect the fact that software projects are normally more risky than lending money to a bank. The exact discount rate is normally less important than ensuring that the same discount rate is used for all projects being compared. However, it is important to check that the ranking of projects is not sensitive to small changes in the discount rate – have a look at the following exercise.

Exercise 2.6

Calculate the net present value for each of the projects A, B and C shown in Table 2.4 using each of the discount rates 8%, 10% and 12%.

For each of the discount rates, decide which is the best project. What can you conclude from these results?

Table 2.4 Three estimated project cash flows

Year	Project A (£)	Project B (£)	Project C (£)
0	–8,000	–8,000	–10,000
1	4,000	1,000	2,000
2	4,000	2,000	2,000

(Contd)

3	2,000	4,000	6,000
4	1,000	3,000	2,000
5	500	9,000	2,000
6	500	−6,000	2,000
Net Profit	4,000	5,000	6,000

Alternatively, the discount rate can be thought of as a target rate of return. If, for example, we set a target rate of return of 15% we would reject any project that did not display a positive net present value using a 15% discount rate. Any project that displayed a positive NPV would be considered for selection – perhaps by using an additional set of criteria where candidate projects were competing for resources.

Internal rate of return

One disadvantage of NPV as a measure of profitability is that, although it may be used to compare projects, it might not be directly comparable with earnings from other investments or the costs of borrowing capital. Such costs are usually quoted as a percentage interest rate. The internal rate of return (IRR) attempts to provide a profitability measure as a percentage return that is directly comparable with interest rates. Thus, a project that showed an estimated IRR of 10% would be worthwhile if the capital could be borrowed for less than 10% or if the capital could not be invested elsewhere for a return greater than 10%.

The IRR is calculated as that percentage discount rate that would produce an NPV of zero. It is most easily calculated using a spreadsheet or other computer program that provides functions for calculating the IRR. Microsoft Excel, for example, provides IRR functions which, provided with an initial guess or seed value (which may be zero), will search for and return an IRR.

One deficiency of the IRR is that it does not indicate the absolute size of the return. A project with an NPV of £100,000 and an IRR of 15% can be more attractive than one with an NPV of £10,000 and an IRR of 18% – the return on capital is lower but the net benefits greater.

Another objection to the internal rate of return is that, under certain conditions, it is possible to find more than one rate that will produce a zero NPV. However, if there are multiple solutions, it is always appropriate to take the lowest value and ignore the others.

NPV and IRR are not, however, a complete answer to economic project evaluation.

- A total evaluation must also take into account the problems of funding the cash flows – will we, for example, be able to repay the interest on any borrowed money at the appropriate time?
- While a project's IRR might indicate a profitable project, future earnings from a relatively risky project might be far less reliable than earnings from, say, investing with a bank. We might undertake a more detailed risk analysis as described below.
- We must also consider any one project within the financial and economic framework of the organization as a whole – if we fund this one, will we also be able to fund other worthy projects?

1 → 10.17%
2 → 3.08%
3 → 15.24%

Exercise 2.7

Check if the projects *A*, *B*, and *C* shown in Table 2.4 are worth taking up when the rate of interest on borrowed capital is 15%.

2.6 Risk Evaluation

Every project involves risk. We have already noted that *project* risks, which prevent the project from being completed successfully, are different from the *business* risk that the delivered products are not profitable. Project risks will be discussed in Chapter 7. Here we focus on business risk.

Risk identification and ranking

In any project evaluation we should identify the risks and quantify their effects. One approach is to construct a project risk matrix utilizing a checklist of possible risks and classifying risks according to their relative importance and likelihood. Importance and likelihood need to be separately assessed – we might be less concerned with something that, although serious, is very unlikely to occur than with something less serious that is almost certain. Table 2.5 illustrates a basic project risk matrix listing some of the business risks for a project, with their importance and likelihood classified as high (H), medium (M), low (L) or exceedingly unlikely (—). So that projects may be compared, the list of risks must be the same for each project assessed. It is likely, in reality, that it would be longer than shown and more precise.

The project risk matrix may be used as a way of evaluating projects (those with high risks being less favoured) or as a means of identifying and ranking the risks for a specific project.

Table 2.5 A fragment of a basic project/business risk matrix for an e-commerce application

Risk	Importance	Likelihood
Client rejects proposed look and feel of site	H	—
Competitors undercut prices	H	M
Warehouse unable to deal with increased demand	M	L
Online payment has security problems	M	M
Maintenance costs higher than estimated	L	L
Response times deter purchasers	M	M

Risk and net present value

Where a project is relatively risky, it is a common practice to use a higher discount rate to calculate net present value. This risk premium might, for example, be an additional 2% for a reasonably safe project or 5% for a fairly risky one. Projects may be categorized as high, medium, or low risk using a scoring method and risk premiums designated for each category. The premiums, even if arbitrary, provide a consistent method of taking risk into account.

Cost–benefit analysis

A rather more sophisticated approach to the evaluation of risk is to consider each possible outcome and estimate the probability of its occurring and the corresponding value of the outcome. Rather than a single cash flow forecast for a project, we will then have a set of cash flow forecasts, each with an associated probability of occurring. The value of the project is then obtained by summing the cost or benefit for each possible outcome weighted by its corresponding probability. Exercise 2.8 illustrates how this may be done.

Exercise 2.8

BuyRight, a software house, is considering developing a payroll application for use in academic institutions and is currently engaged in a cost–benefit analysis. Study of the market has shown that, if BuyRight can target it efficiently and no competing products become available, it will obtain a high level of sales generating an annual income of £800,000. It estimates that there is a 1 in 10 chances of this happening. However, a competitor might launch a competing application before its own launch date and then sales might generate only £100,000 per year. It estimates that there is a 30% chance of this happening. The most likely outcome, it believes, is somewhere in between these two extremes – it will gain a market lead by launching before any competing product becomes available and achieve an annual income of £650,000. BuyRight has therefore calculated its expected sales income as in Table 2.6.

Table 2.6 BuyRight's income forecasts

Sales	Annual sales income (£)	Probability	Expected value (£)
	i	p	$i \times p$
High	800,000	0.1	80,000
Medium	650,000	0.6	390,000
Low	100,000	0.3	30,000
Expected Income			500,000

Development costs are estimated at £750,000. Sales levels are expected to be constant for at least four years. Annual costs of marketing and product maintenance are estimated at £200,000, irrespective of the market share. Would you advise going ahead with the project?

This approach is frequently used to evaluate large projects such as the building of motorways, where variables such as traffic volumes, and hence the total benefit of shorter journey times, are uncertain. The technique, of course, relies on being able to assign probabilities of occurrence to each scenario, which requires extensive research.

When used to evaluate a single major project, the cost–benefit approach, by 'averaging out' the negative and positive outcomes of the different scenarios, does not take full account of 'worst-case scenarios'. Because of this, it is more appropriate for the evaluation of a portfolio of projects where overall profitability is the primary concern, more successful projects can offset the impact of less successful ones.

Risk profile analysis

An approach which attempts to overcome some of the objections to cost–benefit averaging is the construction of risk profiles using sensitivity analysis.

This involves varying each of the parameters that affect the project's cost or benefits to ascertain how sensitive the project's profitability is to each factor. We might, for example, vary one of our original estimates by plus or minus 5% and recalculate the expected costs and benefits for the project. By repeating this exercise for each of our estimates in turn we can evaluate the sensitivity of the project to each factor.

By studying the results of a sensitivity analysis we can identify those factors that are most important to the success of the project. We then need to decide whether we can exercise greater control over them or otherwise mitigate their effects. If neither is the case, then we must live with the risk or abandon the project.

Using decision trees

The approaches to risk analysis discussed previously rather assume that we are passive bystanders allowing nature to take its own course – the best we can do is to reject over-risky projects or choose those with the best risk profile. There are many situations, however, where we can evaluate whether a risk is important and, if it is, decide a suitable course of action.

Such decisions will limit or affect future options and, at any point, it is important to be able to assess how a decision will affect the future profitability of the project.

As an example, say a successful company is considering when to replace its sales order processing system. The decision largely rests upon the rate at which its business expands – if its market share significantly increases (which it believes will happen if rumours of a competitor's imminent bankruptcy are fulfilled) the existing system might need to be replaced within two years. Not replacing the system in time could be an expensive option as it could lead to lost revenue if it cannot cope with increased sales. Replacing the system immediately will, however, be expensive as it will mean deferring other projects already scheduled.

It is calculated that extending the existing system will have an NPV of £75,000, although if the market expands significantly, this will be turned into a loss with an NPV of –£1 00,000 due to lost revenue. If the market does expand, replacing the system now has an NPV of £250,000 due to the benefits of being able to handle increased sales and other benefits such as improved management information. If sales do not increase, however, the benefits will be severely reduced and the project will suffer a loss with an NPV of –£50,000.

The company estimate the likelihood of the market increasing significantly at 20% – and, hence, the probability that it will not increase at 80%. This scenario can be represented as a tree structure as shown in Figure 2.2.

The analysis of a decision tree consists of evaluating the expected benefit of taking each path from a decision point (denoted by D in Figure 2.2). The expected value of each path is the sum of the value of each possible outcome multiplied by its probability of occurrence. The expected value of extending the system is therefore £40,000 ($75{,}000 \times 0.8 - 100{,}000 \times 0.2$) and the expected value of replacing the system £10,000 ($250{,}000 \times 0.2 - 50{,}000 \times 0.8$). The company should therefore choose the option of extending the existing system.

− 20000
60000
40000

50000
− 40000
10000

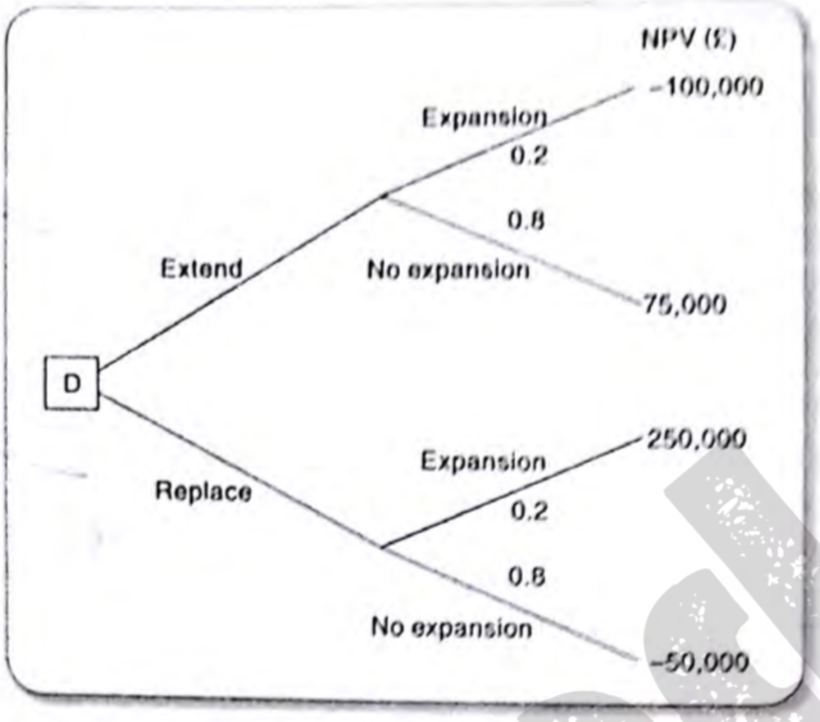

FIGURE 2.2 A decision tree

2.7 Programme Management

Ferns' paper appeared in the *International Journal of Project Management* August 1991.

It should now have been made clear that there will be an element of risk with any single project. Even where projects produce real financial benefits, the precise size of those benefits will often be uncertain at the start of the project. This makes it important for organizations to take a broad view of all its projects to ensure that while some projects may disappoint, organizational developments overall will generate substantial benefits.

We introduced project portfolios in Section 2.3. We will now examine how careful management of programmes of projects can provide benefits. D. C. Ferns defined a programme as '*a group of projects that are managed in a coordinated way to gain benefits that would not be possible were the projects to be managed independently*'.

Programmes can exist in different forms, as can be seen below.

Business cycle programmes

The collection of projects that an organization undertakes within a particular planning cycle has already been discussed under the topic of project portfolios. We have seen that many organizations have a fixed budget for ICT development. Decisions have to be made about which projects to implement within that budget within the accounting period, which often coincides with the financial year.

Strategic programmes

Several projects together can implement a single strategy. For example, the merging of two organizations' computer systems could require several projects each dealing with a particular application area. Each activity could be treated as a distinct project, but would be coordinated as a programme.