CHAPTER
3

Choosing a project and writing a proposal

Aims:

To introduce techniques for choosing an appropriate project, and to discuss the skills needed to write a satisfactory project proposal.

Learning objectives:

When you have completed this chapter, you should be able to:

- Choose an appropriate project.
- Write a project proposal.
- Make effective decisions when choosing your project supervisor.

- This chapter is relevant for both undergraduate and postgraduate projects.

3.1 Introduction

Because the field of computing is extremely diverse, covering a vast range of topic areas from sociological and management issues to highly technical hardware and software developments, it is not always easy to decide on a suitable project for your degree course. The types of projects accepted in different university departments also vary.

Some academic departments may permit students to pursue a highly technical programming project (provided it includes a satisfactory requirements capture, design and implementation), while others require more academic content which emerges from critical evaluation, analyses, and literature surveys. Chapter 1 introduced the general categories for most computing projects: *research-based, development, evaluation, industry-based* and *problem-solving*. This chapter introduces the skills you will need and some tips for choosing an appropriate computing project for your course. It will then discuss ways to present an acceptable proposal for your project and offers advice on choosing a supervisor, if this is possible within your institution.

Note

As you read this chapter, you may wish to read Chapter 5 alongside it. Chapter 5 covers literature searching and literature reviews – activities that can help you understand and contextualise your chosen subject area. Without at least starting a literature search, it may be difficult to define your project or what you hope to achieve.

3.2 Choosing a project

Identifying your project can be the most difficult stage of all. Just as an artist ponders over a blank canvas, you must decide on the type of project you would like to pursue over the following six months, twelve months, three years or more. When choosing your project, keep these important principles in mind.

- You must be capable of doing the proposed project in the time available. You must ensure that your project is not overly ambitious and that you have all the relevant skills needed (for example, don't choose a project in an area where you have failed subjects before). However, as part of your project you may want to broaden your knowledge by developing new skills or enhancing existing skills. If so, be sure to allow sufficient time for these tasks.
- Choose a project that interests you. Remember, you will be working on your project for probably six months or more. It is important that you enjoy your work and do not become bored or lose motivation during this time.
- Consider your personal development and choose a project that will assist you in your goals. In recent years, higher education institutions have promoted the concept of personal development planning (PDP) for students. Personal development planning is defined by The Higher Education Academy (2008) as 'a structured and supported process undertaken by an individual to reflect upon their own learning, performance and/or achievement and to plan for their personal, educational and career development'. Through personal development planning, individuals can identify their aspirations regarding personal development and decide (plan) how best to achieve these aims. Your personal development plan may help you identify a project or choose from a number of ideas you might be considering. Projects that support you in

reaching your personal goals would take priority over those that do not. When choosing a project, consider your future career aspirations. This may influence your project choice.

- Your project should have a serious purpose and a clear outcome that will benefit someone. If you undertake a software development project, for example, you should ideally have a **real** client for whom you are developing the system. Many students undertake software development projects for imaginary clients (for example, developing a web site for a make-believe taxi firm, boutique or sports club), but these projects tend to be below average. Armed with advice from their supervisor, students implement what they feel is appropriate (which may be very different from what an actual client would need). Motivation tends to slip as students realise their software will never be used by a real firm but will only sit on a shelf and gather dust after all their hard work.

  By securing a real client for your project you will get a much clearer, more relevant set of project objectives. You will also be more motivated, knowing you are working for someone other than yourself (you won't want to let them down). Having a real client will also help with your project's management as you will be expected to produce deliverables to the client by specific dates. Having such short-term goals keeps you motivated and the project on-track.

  If you need help obtaining a good client for your project, your supervisor should help you. He or she may already have a client contact you can use. Your supervisor can also provide some official support, for example, writing a letter of introduction for you.
- Your project has a clear outcome (in terms of deliverables) that focuses your work and direction. Without a clear target, you may lose your focus and motivation as your project progresses.
- Your project links suitably with your degree course. For example, you would not pursue a highly technical hardware-based development on an information systems course or perform a detailed systems analysis project on an artificial intelligence course.
- Your project is of sufficient scope and quality to fit the requirements of your course.
- Your project idea is something that interests you, but not a personal issue about which you may have a subjective view that could cloud your perspective and influence your results.
- The resources you require for your project are available or can be obtained; for example, software, hardware, a particular client, user or organisation.

3.2.1 Techniques and Information Sources

As you choose a suitable project, take advantage of the various techniques and information sources at your disposal.

Lecturers'/departmental lists

Sometimes this is the only source of acceptable project ideas. These projects may have been proposed by academic staff in your department or in other departments, or they may be small projects requested by local industry.

Industrial projects

If you have spent a year in industry as part of your course (a sandwich placement) or are working part-time, your past or present company may have a project you could undertake. Alternatively, your supervisor, friends or family might have industry contacts that could be a prospective client. However, industrial-based projects could have some inherent problems. Firstly, the company's objectives could differ from yours in terms of the project's deliverables. What might be acceptable to the company may not have sufficient academic rigour for your course requirements. If your supervisor is advising you to take your project in one direction (academically challenging) and the company wants you to take the project in a different direction, it can be difficult to reconcile the differences. Secondly, industry contacts are not always stable or reliable. Sometimes companies go bust, are taken over, departments close down, staff are moved on, people resign or retire. Can you guarantee that your company contact (and, for that matter, the company) will be there throughout your project? If not, will someone else be able to help? Thirdly, access to companies can be difficult. If you are doing a project based at your university this is not a problem, but if you have to travel long distances to meet with the client, will this be feasible? Fourthly, as an outsider working for the company, will you be granted access to all you need for the project – data, files, people? Some companies may be reluctant to allow outsiders (people who are not directly employed by them) to have access to certain information. You might also find it difficult to talk to certain people in the company if you don't work there. You may be asked to sign confidentiality agreements or clauses before the company will provide access to certain people or information. Finally, will you have access to the hardware and software you need to undertake your project? If the company uses particular applications or systems, will you need to access these remotely or can your institution provide appropriate support?

Despite these potential pitfalls, there are a number of advantages to working with an outside company for your project. Your project will have a clear goal and likely a clear set of requirements specified by the client. Working for an external body will provide additional motivation as you will be working for someone other than yourself and have targets to achieve. Finally, you may well have additional support for your project that may not be available otherwise – for example, the company may offer training, you may have access to sophisticated hardware and software and you may benefit from expert industrial knowledge.

Past projects

Usually your department or university library will hold copies of previous projects. These can provide you with working ideas (for example, on how you could develop the work further) and some sense of the scope and amount of work expected. Alternatively, you can use the Internet to search for past projects at other institutions. For example, PhDs are sometimes made available on-line. Past projects may also provide a suitable basis from which to start your own project – picking up from where a previous student left off. If you undertake a development project, you may be developing an existing piece of software further, introducing additional functionality or creating new features to an older system. You need to be aware of the amount of effort required to upgrade existing systems. Quite often existing systems are poorly structured, badly commented and lacking documentation. It can be harder to enhance a piece of software that already exists than to develop a system from scratch. In addition, previous programmers may have used unusual code to create the system and it may take you a long time to deconstruct their software.

Talking with colleagues

Your peers can often provide a different perspective on ideas you might have. They can be helpful in highlighting potential shortcomings of your intended project and may suggest alternatives.

Reading around subject areas

If you read books, journals and articles on a topic that interests you, you can often discover areas that authors have identified as requiring further research and development. As you improve your understanding of the topic area, you may identify a gap that you wish to investigate further. Whatever happens, reading around your intended subject area does no harm and helps you gain a solid understanding of the subject on which you will build your project. Chapter 5 covers literature searching and it may be worth reading that chapter before completing your proposal.

Clustering

You might wish to pursue a project in a particular field but are unsure exactly which aspects of the topic to focus upon. Clustering can help you identify aspects within a topic area that link together and are worthy of further investigation. Clustering is performed in two stages. First, you should list keywords related to your topic area. Second, once you have exhausted all the words and phrases you can think of, you cluster them into related groups and patterns. Doing this can help you identify specific topics that interest you and form the basis of your intended project.

Clustering can be used to develop *Research Territory Maps* (RTMs), *Relevance Trees* and *Spider Diagrams*. An RTM, sometimes called an *affinity diagram*, shows how topics relate to one another within your chosen field or fields of study. RTMs provide you with your own conceptual model of your research area. They represent *your* interpretation of the field – one that you are comfortable with, can clarify and arrange the literature into and can modify as your knowledge of the field grows. These maps can be enhanced with thicker and thinner connecting lines to emphasise the strength of relationships between subjects. Figure 3.1 provides an example of an RTM – in this case, a high-level conceptual map of the field of *software engineering* (remember, this is an interpretation and you may or may not agree with its structure). RTMs will identify specific topics you might wish to focus on within larger subject areas or, for broader studies, inter-related subjects that are dependent and require investigation.

Relevance trees, discussed in more detail by Sharp *et al.* (2002: 36) and Saunders *et al.* (2007: 74–75), are similar to RTMs in that they try to model your field of study. Relevance trees differ from RTMs in their hierarchical structure. While RTMs identify related topics and the links between them, relevance trees break down a particular subject or research question into lower and lower levels of detail, identifying how a subject

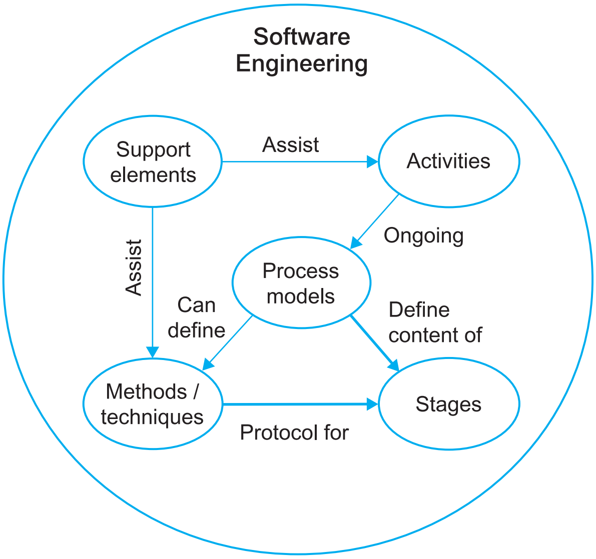

Figure 3.1 A high-level RTM for *software engineering*

is composed or identifying the factors affecting a research question posed. RTMs provide a holistic interpretation of the field of study while relevance trees provide a hierarchy of topics that constitute that field of study. An example of a relevance tree for *artificial intelligence* is shown in Figure 3.2.

Another way of structuring your thoughts and identifying the composition of a subject is through the use of *spider diagrams*. These diagrams are similar to RTMs in that they show how topics within a subject area relate. They are also similar to relevance trees in that they show how topics break down from a central idea, subject or research question. In spider diagrams a central node represents the topic of interest and lines emanating from this node identify how the topic can be organised into its constituent parts. Colours are often used to group ideas and topics. Figure 3.3 provides a spider diagram interpretation of the field of software engineering. This diagram is adapted from the RTM in Figure 3.1.

Remember that relevance trees, RTMs and spider diagrams are structured by **you** to represent your *own* interpretation of your chosen subject area. Other authors may decompose the subject area into an alternative structure or use different terminology for the same things. You must be aware of these differences so that you aren't confused by what appears to be contradictory information, which you gather from your literature search. For example, in Figure 3.2, some authors may subsume *Knowledge representation* within *AI techniques* or might disregard *Philosophical issues* entirely, while others may include other topics not identified here.

Brainstorming

If you are really struggling for a project idea, brainstorming can provide the answer. Brainstorming involves listing all your project ideas on a piece of paper, in any order and as quickly as possible. Write anything down, even if it sounds completely irrational, as

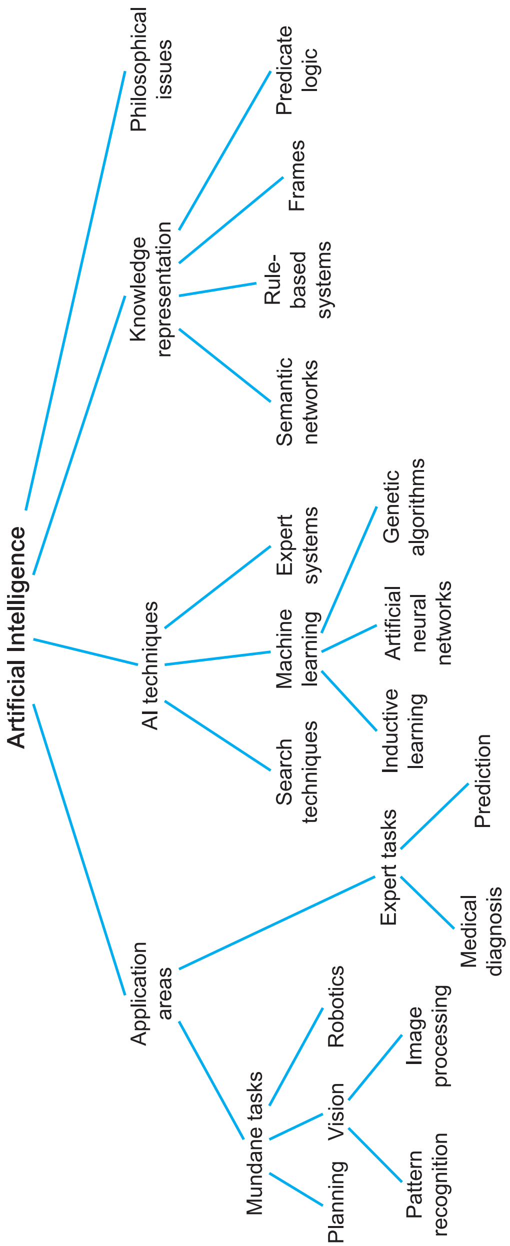

Figure 3.2 Example of a relevance tree for *artificial intelligence*

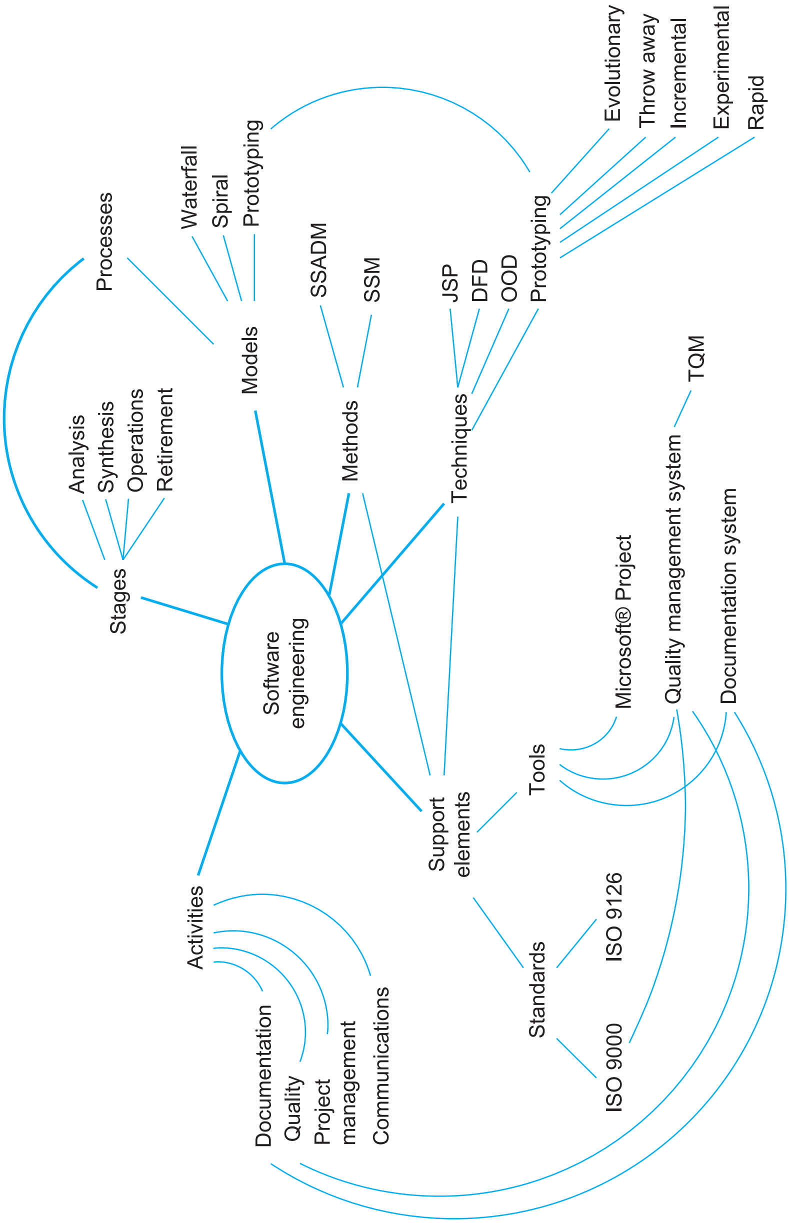

Figure 3.3 Example of a spider diagram for *software engineering*

the process of brainstorming should not be stifled. When you have finally exhausted all your ideas, look at each one in turn and evaluate and assess it in more depth. What may have sounded ridiculous at first may actually lead to a good project idea; perhaps when viewed from a different angle. You might also like to group your thoughts (using clustering) as this may help to clarify in your own mind where your real interests lie. One way to choose between topics is to toss a coin – not to see which way the coin lands but to see how you feel you want the coin to land while it is spinning in the air.

Chapter breakdown

Once you have an idea for your project, it is a good idea to identify how your project will break down into a number of chapters for the final report. If you have difficulty identifying a number of specific chapters for your final report, it may mean you are unclear about the project's detail and don't really understand what it is you hope to achieve. Breaking down your project into chapters will also give you an indication of its scope. If you can identify only two or three chapters, maybe your project is not sufficiently broad. Conversely, if you can identify ten or more chapters you may be trying to do too much.

3.2.2 Additional considerations

After identifying your project, think about these additional considerations:

The 'so what?' test

To ensure that you do not pursue a project that has little value, take the 'so what?' test (Herbert, 1990: 7). Ask yourself, Is the topic meaningful? If you complete the project successfully, will it be of value to anybody? What contribution will it make? Pursuing a meaningless project can lead to poor motivation as your project progresses and you begin to question the point of your work.

Justification

Can you explain your project and justify it (that is, pass the 'so what?' test) in simple terms to the woman or man in the street? If so, you have a good understanding of the subject area and the topic you want to pursue. While your explanation may still be too technical or deep for the average person to understand, feeling that you can explain it in simple terms indicates that the topic is clear to you.

Numerating your understanding

Can you put a number on how much you know about your chosen subject; for example, 80%? If you are able to numerate your understanding about a topic it means that you have, at least, a concept of that field of study and an awareness of its magnitude. If you have no idea what your understanding is, you have no idea of your subject area's depth or breadth and to undertake a project in this area would be very risky.

This principle was initially presented by Lord Kelvin, who stated:

> *When you can measure what you are speaking about and express it in numbers, you know something about it: when you cannot measure it, cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind. It may be the beginning of knowledge, but you have scarcely in your thought advanced to the stage of science.*

This idea, of being aware of the sum total of your understanding about a specific topic, is sometimes referred to as *metaknowledge*. In some ways, the wiser people become, the more they realise just how little they really know. This is especially true when people place their understanding within the broader context of world knowledge even though their own expertise in a particular subject may be very deep. Figure 2.2 identified this transition of self-awareness in the research as a matrix with four quadrants, from *blissful ignorance*, to *demoralised*, to *confident*, to *complacent*.

Contacts

When you identify the contacts you require for your project, are they available, accessible and willing to help? For example, do you have contacts within a local company who have volunteered to help you with a case study? If not, your project will face problems that need to be dealt with sooner or later.

What do you already know?

Orna and Stevens (1995: 29) suggest that, as you identify your research area, you also think about what relevant information you already know or have access to. This will help you clarify your strengths and, perhaps, form a foundation for your RTM, relevance tree or spider diagram. You might also identify what you want to **learn** by conducting your project. What are your educational objectives? Are there any skills you would like to develop, new programming languages to learn or new techniques to master? If so, include a need for these skills within your project to 'force' yourself to learn them.

Ethical issues

Most institutions have procedures in place to assess, evaluate, monitor and control ethical issues when approving student projects. While these concerns were originally aimed at projects involving medical or sociological studies (for example, drug testing, vulnerable groups and hazardous substances), the codes of practice developed have spilled over into all disciplines. Most projects in computer science and information systems generally do not raise ethical issues. However, if you plan to involve others in your project in any way – conducting a survey, developing a system for a vulnerable group (requiring their interaction and feedback), undertaking action research (working with people), developing a database to store personal data, or working on medical systems – you may well need to obtain ethical clearance for the project from your department, university or some other external agency. Most institutions with have an Ethical Advisory Committee that manages these issues. The areas of concern that usually need addressing when considering ethical issues include:

Data protection

If your project will handle any personal information relating to living people, you need to be aware of the Data Protection Act (DPA) and its restrictions on your project. The DPA, introduced in 1998 in the UK, governs the protection of personal data. These data can include names, addresses, exam grades, medical data, telephone numbers, anniversaries and more. The main principles of the Act that you should be aware of include:

- Data should only be used for the specific purpose for which it was gathered in the first place.
- Individuals have the right to access data held about them.
- Data may not be disclosed to third parties without permission of the individual.
- If personal data are kept, these data must be appropriately protected.
- Personal data should be kept for no longer than necessary.

Recruiting participants

Ethical issues will make you think a little bit more about how to recruit participants for your project (for example, a cross-section of users to give feedback on your system). You will strengthen your study by considering these areas:

- Selection criteria – will this be unbiased and lead to a reasonable cross-section of participants?
- What consent will be required from participants (or their parents/carers/guardians) to take part?
- Will there be any financial incentive or otherwise to take part (which may affect the objectivity of the results)?
- Will participants be able to withdraw at any stage (this should be allowed)? Will any deception be involved? For example, intentionally to evaluate a reaction or unintentionally which might affect the results?

Vulnerable groups

These can include children, pregnant women, people with a mental illness, prisoners, people over 65, people with learning disabilities, etc. Quite obviously these people need some kind of protection and, although you may feel your work will not affect them (for example, just asking their opinions about some software you are developing to meet their needs), ethical clearance will probably still be needed to ensure they are not being exploited in any way. You should also consider whether or not any feedback you receive from such groups will be entirely objective – it could be skewed because of their relationship to you and you should acknowledge and discuss this issue in your final report.

Training

Will you require any training in order to undertake the project (for example, from hardware or software applications to dealing with vulnerable groups)? Will participants require any training? What form will this take and will it raise any issues?

Using these ideas, sources and approaches will assist you in deciding on your project. However, although you may now have an idea for a project that you feel is of suitable quality and scope for your course, you must now 'sell' it to others with a *project proposal.*

3.3 Preparing a project proposal

3.3.1 Introduction

It is normal, in most institutions, for you to prepare a proposal for your project so it can be assessed for acceptability. Unless you can present an acceptable proposal, your project will not even start. It can serve as a contract between you, your department and project supervisor – but don't expect it to be used against you if you achieve more than you actually intended to do! In many cases, projects can and do change direction as they proceed; as you become more aware of the topic area and the problem which you are investigating. This is acceptable provided the scope and quality of your project do not become 'watered down' and you are not heading so far away from your initial intentions that the project becomes unrecognisable. If this were the case, you would need to obtain permission for significant changes and possibly have to submit a new proposal.

When preparing your proposal, follow these two golden rules:

1. Follow any guidelines precisely. Most institutions require specific information; for example, project title, project objectives, resource requirements and so forth. Failure to complete these sections may mean your proposal is rejected without even being read, for example, because you failed to get an academic signature or did not complete an essential section properly.
2. Proofread thoroughly and get someone else to check it. Any errors or omissions will appear sloppy and put your commitment and proposed project in a bad light.

There are no universal standards for project proposals, although all proposals should include certain pieces of information. This content emerges from your proposal's *implicit content* and *explicit sections*, which are discussed below.

3.3.2 Implicit content

In general, your proposal should address five principal areas. These may not be identified explicitly in the structure of your proposal, but they should be addressed implicitly within the proposal's content. They are:

1. **Introduction to the subject area.** This will provide the reader with an understanding of the field in which your project lies and an idea of where and how your project fits into this field. This aspect will set your project into an overall *context*

and will show that it is bound within a recognised field – not an idea that you've had that makes no sense and has no recognisable foundation.

2. **Current research in the field.** This will emphasise that your project is not based in a field that is out-of-date and that you are aware of current issues within that field of study. It will also imply that you have done some preliminary research into the topic area and are not approaching your project with little background or motivation.
3. **Identify a gap.** You should be able to identify some aspect of the field that requires further investigation or study. There is no point in repeating the work of others (unless you are evaluating their approaches) and this component emphasises that the field is not exhausted and is worthy of further investigation.
4. **Identify how your work fills the gap.** Having identified a gap in the field, your proposal should show how your project intends to fill this gap, or at least go some way to investigating it further. This will emphasise the *contribution* your project will make.
5. **Identify risks and solutions.** It is also useful in a project proposal to highlight any risks your project might face, and ways you envisage dealing with them. If you do not identify potential risks to your project, your proposal's assessor will not know whether you have considered the risks. If they feel you have not accounted for potential risks to your project, your proposal may not be accepted, as they may not appreciate that you have potential contingency plans in place.

3.3.3 Explicit sections

Detailed below are the most common sections that project proposals should include. If you receive no guidance as to the content of your project proposal, include, at the very least, the first three of the following sections in your document.

Title

This should be clear and concise. Try to avoid using acronyms if possible. Examples of clear and concise titles include:

- 'Evaluation of soft systems methods as analysis tools in small software houses';
- 'Artificial neural networks for software development cost estimation';
- 'Development of process models for building graphical software tools'.

Aims and objectives

Aims identify at the highest level what it is you hope to achieve with your project – what you intend to achieve overall. An aim is a broad statement of intent that identifies your project's purpose. Objectives, on the other hand, identify specific, measurable achievements that build towards the ultimate aim of your project. They are more precise than aims as they are 'quantitative and qualitative measures by which completion of the project will be judged' (Turner, 1993: 108). They represent major components of your project that direct your work activity (Weiss and Wysocki, 1992: 13).

Identifying aims and objectives clarifies, both for you and the reader, what you specifically hope to achieve with your project. You will use your aims and objectives to assess your project at the end. For example, did you really achieve all that you set out to do? Because of this, aims and objectives should be clear and unambiguous. Chapter 4 discusses aims and objectives further.

Examples of aims and objectives are:

Aim:

- Evaluate artificial intelligence techniques for modelling weather patterns.

Objectives:

- Identify and evaluate existing weather pattern modelling techniques.
- Identify artificial intelligence approaches suitable for modelling weather patterns.
- Design and develop at least three artificial intelligent systems for modelling weather patterns.
- Compare and contrast the developed systems with one another and existing approaches to modelling weather patterns.

Expected outcomes/deliverables

This section of your proposal will identify precisely what you intend to submit at the end of the project. It may well identify a written report that covers particular points and makes certain recommendations. A chapter breakdown may be included where appropriate. It can describe programs and user documentation and it might include models and algorithms that will be developed to address specific problems. You might also be delivering a functional specification for a piece of software, a prototype, or a test plan.

These three headings represent the minimum set of sections your project proposal should include. In addition, consider including the following:

Keywords

Keywords are used to identify the topic areas your project draws on. People use keywords to see at a glance what subjects your project relates to which might not be clear from your project's title alone. Libraries and databases use keywords to help classify material. You might be limited on the number of keywords you can use; for example, four or five. Remember, keywords are not necessarily single words but can be simple phrases as well; for example, artificial intelligence or software engineering.

Introduction/background/overview

This section provides an overview of your project and introduces the background work to it. In this section you might wish to include reasons why you feel you are a suitable candidate for performing the project (why you feel you can do it, what skills are required and how you fulfil these requirements), why the topic interests you specifically, and why you chose the project in the first place. This section might also include an introduction to the industry or organisation being investigated or evaluated. Overall, this section will set the scene for the project.

Related research

This section identifies other work, publications and research related to your topic. It will demonstrate that your project does not exist in an academic vacuum but relates to other research topics and fields of current interest. Related research can also help demonstrate your understanding of your topic area, showing the reader that you are aware of what is currently happening in the field and are conversant with other topics that impinge upon it.

Type of project

You might wish to identify the type of project you are undertaking, for example, *research-based*, *development*, *evaluation*, etc. However, make sure these terms are recognised and provide more detail if appropriate.

Research questions and hypotheses

Your project proposal may also include the research question you intend to investigate and, hopefully, answer to some extent within your project. Computing projects do not necessarily set out to answer particular questions, but for some projects (particularly research degree projects) a statement of your research question is essential. Examples of research questions are:

- Does the size of an organisation affect its commitment to software quality standards?
- What is the relationship, if any, between software maintainability and coding structure standards?
- Is there an optimum solution to the prediction of software development costs?
- How do large organisations maintain quality standards in the development of internal software?

While research questions on their own are 'open-ended opportunities to satisfy one's curiosity' (Rudestam and Newton, 2005: 74), they are often linked closely with one or more hypotheses. A hypothesis is 'a tentative proposition which is subject to verification through subsequent investigation' (Verma and Beard, 1981: 184 cited by Bell, 2007: 32). Although you do not have to define hypotheses alongside a research question, they do present potential 'answers' to the question(s) you have posed and provide definitive statements that will focus your research. For example, suppose your project intended to answer the fourth research question posed above. One of the following hypotheses might be investigated based on that research question:

Hypothesis #1: Large organisations invariably employ recognised standards to maintain internal software quality.

Hypothesis #2: Large organisations generally have quality departments which oversee the implementation of procedures that ensure the quality of internal software.

It is also worth mentioning the importance of maintaining research *symmetry* with respect to research questions and hypotheses. Research symmetry implies that your 'results will be of similar value whatever you find out' (Gill and Johnson, 2002 cited by Saunders et al, 2007: 20). With this in mind, it is important to realise the implications of the hypotheses you have stated. If they are true you must ask yourself 'so what – was that really worth proving?' Thus, each hypothesis you state should have a similar value if proved.

Methods

This section describes the research and project methods you will use in performing your project. This section should not identify methods that you might be investigating as part of your project, but those methods you are actually using. It might include development methods that you are using as part of a systems development (for example, SSADM); survey methods for a case study evaluation and evaluation methods for comparing two or more systems. Research methods would include those introduced in Chapter 2, such as *action research*, *case study*, *survey* and *experiment*.

Resource requirements

You might need to identify any resource requirements for your project, such as hardware, software and access to particular computers. If you have access to particular resources, this fact should be pointed out in this section. Quite clearly, if the resources for your project are not available in your department, or are too expensive to obtain, your project will be unacceptable. However, if you know you need a particular piece of software or hardware, you must find out its cost and include this information within this section. A proposal that omits this information may be rejected if the assessor does not know how inexpensive or available the item is and might assume it is beyond your project's budget.

Note, if you are relying on an external source (a company, for example) for computer access, hardware, software, case studies and the like, it will be your responsibility to ensure these are available and to bear any risk if the resources fail. While your department will accept responsibility if something goes wrong with your project because their own software or hardware fails, they will not be responsible for external sources of support that you have arranged.

Within this section or under a separate heading, you might include a list of the literature you will need to perform your project – for example, specific journals, company reports, books, etc. If these materials are unavailable, realistically speaking, your project may be impracticable and you may need to change its focus. Access to particular companies for performing case studies could also be identified here. Without this access your project might flounder, so it is important to show you have contacts that can be utilised.

Project plan

It is very useful to present a project plan as part of your proposal. This emphasises that the project is 'doable' in the time allowed; it shows that you have some idea of the work involved and you have a clear pathway to follow in order to complete that work. The best way to present a project plan is by using a visual representation such as a *Gantt chart*. These figures are described in Chapter 4. While the presentation of a Gantt chart is important, for the purposes of your project proposal, limit your chart to a single page. A multi-page project plan is difficult to read and, for a proposal, only a general overview is required.

3.3.4 Reviewing your proposal

The second golden rule for preparing a project proposal states that you should proof-read your complete proposal thoroughly.

Check your proposal for spelling mistakes, omissions and grammatical errors. Have you included all the sections you were supposed to and have you completed them in sufficient depth? Is the proposal well presented (typed rather than hand-written, for example)? Do the sections flow logically?

The following are two examples of final-year project proposals for a student on a taught bachelor's degree. Both proposals represent the same project and have been kept short for clarity.

Example

Title:
Software migration.

Project type:

Aims and objectives:
- Migrate a series of software applications from a mainframe to a client/server systam within a local company.

Outcomes and deliverables:
- Connectivity to the mainframe for approx 1000 PCs;
- Full integration into a client server environment;
- Education of users;
- Coding and testing completed.

Research methodology:
PRINCE II.

Hardware and software requirements:
All available at local comnpany.

This proposal is quite poor. Its *Title* is rather vague and only represents the type of project that is being proposed. The section identifying *Project type* has been left blank and the *Aims and objectives* represent a basic, technical, industry-type project with no academic content or justification. Expected *Outcomes and deliverables* emphasise this point and merely identify the technical outcomes of the project. The *Research methodology* section identifies the method that will be evaluated, rather than the research methods that will be employed (PRINCE II is a project management method that is used to manage large projects). The proposal also includes a number of spelling mistakes and abbreviations. Overall, this project lacks any academic quality or rigour and is poorly presented.

Let's look at this project proposal from a new angle:

Example

Title:
Project management issues of software migration.

Project type:
Evaluation project, industry based.

Aims and objectives:
Aim: To evaluate the use of the PRINCE II method as a means of managing the migration of software from a mainframe to a client server system.
Objectives: An evaluation of tools and methods to assist the technical aspects of the migration and organisational management aspects.
Evaluation of similar companies performing migration for comparative purposes.
The migration of a series of applications at a local company (to which access has been obtained) will be used as a vehicle for critically evaluating the PRINCE II method in particular.

Outcomes and deliverables:
A report detailing the following:
– an explanation of the perceived benefits of such a migration;
– an analysis of the difficulties experienced;
– a critical evaluation of the PRINCE II methodology and its application;
– an outline methodology for future migration projects;
– a discussion and evaluation of alternative tools and methods for software migration.

Research methodology:
Case study, action research.

Hardware and software requirements:
All available at a local company.

This proposal is a far better representation of an academic project than the preceding one. Although the project is based on the same software migration, it identifies, far more clearly, the academic side of the project and the critical evaluation required by such projects. All sections are now completed correctly; for example, *Research methodology* identifies those methods actually employed and *Project type* has now been identified. The proposal reads well and has been checked for errors and omissions.

3.4 Choosing your supervisor

Academic departments have different ways of assigning project supervisors to students. There are only a finite number of projects a supervisor can effectively supervise and you may find you are allocated someone who knows little about your field (although this **must not** happen at research degree level). If you are lucky enough to be able to choose your own project supervisor, there are a number of considerations you should contemplate when making your choice. Sharp *et al.* (2002: 31) identify five questions that students should ask of potential supervisors:

1. 'What are their records in terms of student completions?'
2. 'What are their views on the management of student research – and, in particular, the supervisor's role in it?'
3. 'How eminent are they in their specialisms?'
4. 'In addition to being knowledgeable about their subjects, have they high competence in research methodology?'
5. 'How accessible are they likely to be?'

The fifth point noted here can relate not only to a supervisor's general availability but to their approachability as well. It is all very well being able to see your supervisor regularly, but if you do not trust or get along with your supervisor, this time is wasted.

While research degree students will require a supervisor to be an expert in their subject area, this is less important for taught degree students. At the taught degree level a supervisor's role may be more managerial and pastoral than technical – for example, helping you with project plans, checking you are achieving your milestones and assisting you with any project-related problems. At the taught degree level, your supervisor, although not an expert in your chosen field, can still be a good supervisor and may well have sufficient technical understanding to assist you when necessary. You should also be able to approach other members of academic staff in your department (or other departments) for technical assistance. However, if there is no one in your department who is knowledgeable in your chosen field, your project is probably not appropriate anyway. Chapter 7 looks in more detail at the student/supervisor relationship and discusses how to effectively manage the time you spend with your supervisor.

3.5 Summary

- Choosing the right project is probably the most important stage of any project.
- A number of techniques have been presented that you can use to assist you with choosing a suitable project.
- When preparing a proposal there are two golden rules; follow any guidelines precisely and proofread it thoroughly.
- A project proposal should include, at least implicitly: *background*, *related research*, *identification of a gap*, *how your project fills that gap* and *risks and contingency plans*.

- Project proposals should include, at the very least, the sections *project title, aims and objectives* and *expected outcomes/deliverables.*
- Questions have been presented that you should ask yourself before you choose your project supervisor (if this is possible within your own institution).

3.6 Exercise

1. Can you think of any ways to improve the 'corrected' version of the proposal in Section 3.3.4?

3.7 Action points

- Try to build an RTM, relevance tree and/or spider diagram for your own computing project.
- Put together a proposal for your own project using the ideas and skills you have learnt in this chapter.

Solutions to selected exercises

1. The following are areas in which this proposal is lacking and could be improved still further (did you spot these shortfalls and did you identify any others?).

    Objective 1 – other than Prince II, no other methods or tools are explicitly identified in the proposal although they are alluded to.

    Objective 2 – there is no reference to the companies needed for this contrasting evaluation or any indication of how the data might be obtained for this part of the study.

    Outcome 3 – the report is going to comment on the effectiveness of Prince II in migration projects only. This is not clear from Outcome 3.

    There is no project plan so it is difficult to see how long the migration will take and how long the research component of the project will take.

    No risks or contingency plans are identified – for example, what if something goes wrong with the migration and the project is significantly delayed?

CHAPTER

4

Project planning and risk management

Aims:

To introduce techniques and approaches to project planning and risk management.

Learning objectives:

When you have completed this chapter, you should be able to:

- Understand the five elements of projects that need to be managed.
- Describe the typical stages of an academic project from a project management perspective.
- Define a project in terms of aims and objectives.
- Discuss the activities performed during the initial planning stage of a project.
- Understand the use of project management techniques for project planning.
- Manage risk in your project effectively using a defined risk management process.

- This chapter is relevant for both undergraduate and postgraduate projects.

4.1 Introduction

4.1.1 Overview

Before tackling the actual work you need to complete your project, it is important to have some idea of how you are going to undertake that work. Without an appropriate project plan in place you will have little direction, you will lose sight of where your project is going and you will not complete your project on time. In this chapter we will look at some techniques you can use to plan how you will tackle your project. With a reasonable project plan in place it will enable you to manage your time more effectively, decide on priorities for your project at different stages and give you clear direction and motivation. We will also look at risk management, which will enable you to identify, manage and control any potential risks to your project. We will begin by discussing a generic interpretation of the project process before using this framework to introduce a number of project management techniques.

4.1.2 The project process

Figure 4.1 provides a generic view of the project process and introduces the five elements inherent in all projects – *time, resources, cost, scope* and *quality*. This view can be applied to any project to a greater or lesser extent – from large industrial projects right down to small, one-person projects like your own. The diagram shows that all projects consume *time, resources* and *money* (a budget or cost) in order to produce a particular product which has its own *scope* and *quality*. This product can be something tangible like a bridge, a report or a software system; or it could be something intangible like a change in working practices, restructuring a department, improving company profits and so on (although these projects would undoubtedly have some tangible documentation associated with them).

In the case of your own project, Figure 4.1 shows that this will take a certain amount of time to complete and it will require the use of certain resources – primarily **you,** but probably other resources too – for example, your supervisor, a client/user, computer hardware, books and software. You should be aware that you will need 'access' to these resources as your project progresses and be confident that they will be available as and when you need them.

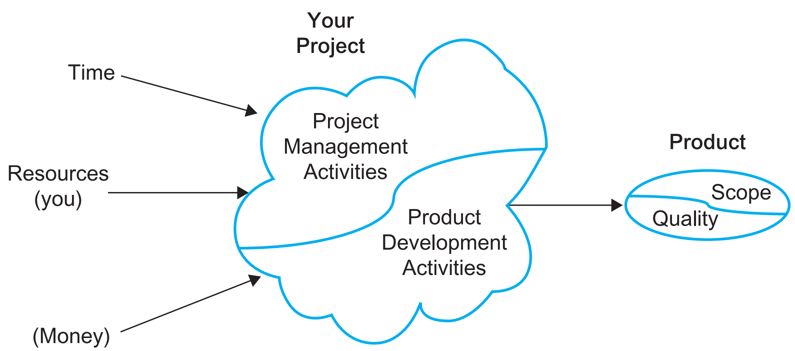

Figure 4.1 A generic overview of the project process

Although commercial projects have budgets associated with them, it is unlikely that you will have any real budget to manage. Some institutions do provide small budgets for student projects (for example, to purchase hardware, software, books or perhaps travel to visit clients) but normally it will be your supervisor or client who will be responsible for this issue.

Figure 4.1 goes on to show how these 'inputs' are used to produce the project's product. The project itself consists of two main activities – *project management activities* (the focus of this chapter) and *product development activities*. Project management activities are concerned with planning how you will undertake your project, controlling your project as it is progressing, checking your progress, meeting milestones, monitoring deliverables and managing risk. Product development activities are involved with the actual project work itself; for example, developing a program, writing reports, literature searching, meeting clients, quantitative research, qualitative research and seeing your supervisor – in other words, everything you need to do to complete your particular project successfully. Generally speaking, project management activities should take no more than around **10%** of your overall effort on the project. Many students do not stick to this principle and spend many hours planning and re-planning the minutiae of their project, rather than getting on with the real task in hand. Their projects might be well planned and controlled but they leave themselves insufficient time to pursue a worthy project.

You should be aware that the 10% effort you put into your project's management is not distributed evenly throughout the life span of the project. You will spend a lot of project management effort towards the start of your project planning how you will perform it, and less effort, as the project continues, actually controlling it.

The final stage of the project process in Figure 4.1 shows the outcome from the project – the project's product – i.e., the artefact that is finally assessed. This will probably be a report of some kind, a thesis or dissertation, a presentation, perhaps a fully documented computer program and an associated user guide and demonstration, a new model or algorithm, a literature survey, a case study, etc. There are two aspects to this 'product' – it will have a certain *scope* (what it covers, what it does – i.e., what it achieves) and a certain level of *quality* (how well it does it). These are two aspects of your project over which you have control. You can, for example, reduce the scope of your product in order to improve its quality. Alternatively, you may want to cover more aspects in your report, or include more functionality in your program at the possible expense of quality. This is a difficult trade-off to make and is discussed in more detail in Chapter 7.

4.1.3 The project's stages

From a project management perspective, all projects progress through **five** main stages during their lifetime; from the time the project is established as an initial idea to the time the project is finally completed. These stages apply to all kinds of projects; from your own academic computing project to large industrial projects spanning several years. At this level of detail specific activities that might be unique within academic computing projects are not of interest. We are interested in the broader stages in which project activities are performed. Each of these stages requires managing in one way or another and there are different considerations you will have to make as your project progresses through them. The five main stages are:

1. *Definition*
2. *Planning*
3. *Initiation*
4. *Control*
5. *Closure*.

Project definition and *project planning* collectively relate to your project's ***inauguration***. Project inauguration refers to the activities you perform before you actually start work on the main body of the project itself (the product development activities). Project definition is the preliminary stage of this process and includes the activities presented in Chapter 3; deciding on your project and getting it approved by submitting an acceptable proposal. In addition, this stage also includes establishing a more detailed project definition in order to prepare the ground for project planning. Project planning is the stage in which you decide how you will fulfil your aims by deciding how to approach the work you need to perform. Project definition and project planning are the focus of the project management activities within this chapter. Note that you will also use these preliminary stages to plan how you will manage the risks associated with your project. Section 4.4 introduces the risk management process that runs in parallel and complements the project management stages.

The following three stages of the process represent the project management activities you perform as you undertake the product development activities of your project – i.e., actually 'doing' your project and working on developing the project's product. *Initiation* represents the activities that you perform to start the main content of your project. It involves arranging yourself into some kind of routine and can include the initial work you perform on your literature survey. If you are working on a group project you will have to assign tasks to, and organise, other members of your project team. You will arrange to meet with your project supervisor and lay down some ground rules and routines for the work ahead. This is discussed in more detail in Section 7.1.2.

Once you have organised yourself and your project is underway you will need to *control* it as it is progressing. Project control is covered in detail in Chapter 7. The last stage of any project is *closure*. In your computing project this will represent the completion of your project, writing up your report, perhaps preparing for a final presentation or viva voce examination, completing any programs and associated documentation and finally handing everything in. How you complete your project by performing these activities is the subject of Chapters 8 and 9.

4.2 Project definition

4.2.1 Overview

The purpose of project definition is to clearly specify what it is you hope to achieve with your project. As mentioned above, this stage initially includes deciding on your project and putting together a proposal (covered in Chapter 3). In many ways your project definition and your initial project proposal are closely linked. Both should be written at the start of your project and, while your proposal aims to get your project accepted, your definition will help to clarify what it is you are really setting out to achieve.

Your project definition must identify the *aims* and *objectives* of your intended work. Chapter 3 briefly introduced the difference between aims and objectives for the purposes of producing a project proposal. In this section these ideas are extended so that your project can be defined clearly in these terms. Defining your project in this way is important for a number of reasons;

- If you have difficulty defining your project in terms of aims and objectives then you will have difficulty deciding on the work you ought to be doing and what your focus will be. It might also mean that your understanding of the subject area is lacking and you need to do some additional preliminary research in the topic area or, more drastically, choose an alternative project.
- It gives you a clear target for which to aim. This provides a continual reference point against which you can assess your progress.
- It provides you with a means of evaluating your success at the end. For example, did you achieve all that you intended to do or more?

4.2.2 Defining your aims

Your project should be defined at two levels. At the top level you define your project's aim or *goal*. All projects have one major aim that they hope to achieve and your computing project is no exception. If you are ever in any doubt over what work you ought to be doing or which direction you ought to be taking, you can refer to your project's aim to guide you. Examples of typical aims for computing projects are:

- to evaluate the effectiveness of requirements capture techniques in small software development companies in India;
- to develop and evaluate a user interface for statistical software packages;
- to design a methodology for GUI development of technical courseware material;
- to investigate and produce an evaluation of fourth generation languages for database development.

Each of these aims provides you with an understanding of that project's main purpose. They identify the area of investigation and the focus of the intended work. In order to achieve these aims each project will have a set of *objectives;* smaller sub-goals that are significant steps towards achieving the project's aim.

4.2.3 Setting objectives

As Chapter 3 specified, objectives identify significant measurable achievements you hope to make that build towards the ultimate aim of your project. Having identified and defined your project's aim you should continue to define your project in terms of its objectives. For a project expected to last approximately one year, you wouldn't expect to identify more than twelve objectives for your project. If your project has more objectives than this it may be that you are attempting to do too much or that you are breaking your project down into too much detail.

Take, as an example, a computing project that is going to evaluate artificial neural networks for predicting stock market indices (not an easy task!). You might identify the following aims and objectives for this project.

Project's Aim:

- Develop and evaluate an Artificial Neural Network to predict stock market indices.

Project's Objectives:

1. Complete a literature search and literature review of existing stock market prediction techniques.
2. Develop a suitable artificial neural network model.
3. Identify and collect suitable data for analyses and evaluation.
4. Evaluate the model using appropriate statistical techniques.
5. Complete final report.

Note how the objectives build towards the ultimate aim of the project. They also appear in approximately chronological order – in other words, they identify the order in which you would expect to tackle the work. Notice, also, how you could further break down these objectives. For example, objective 2 would need you to investigate, evaluate and identify a suitable tool and topology before you could develop a suitable neural network. Objective 4 may require you to investigate and learn how to use some suitable statistical techniques or statistical software packages. However, breaking objectives down into progressively lower and lower levels of detail serves little purpose other than to cloud your vision of your ultimate goal. This will become clear in the following sections, which discuss how to break down the actual work you will need to do to complete your project using *work breakdown structures*.

4.2.4 SMART objectives

The objectives identified above still require further refinement. This is achieved through the application of the SMART technique. There are a number of definitions of the SMART acronym, the most applicable to student projects being:

- **S**pecific
- **M**easurable
- **A**ppropriate
- **R**ealistic
- **T**ime-Related.

For each of the objectives you identify in your project you should apply these criteria to ensure that those objectives are clearly defined. For example, take Objective 1 from our example project:

1. Complete a literature search and literature review of existing stock market prediction techniques.

Applying the SMART criteria:

- Is this *specific*? Does this provide us with enough idea of what we should be doing? (Yes)
- Is it *measurable*? How can we measure progress on this objective? How will we know when the objective is completed? How much literature must we access in order to conclude that the literature search is complete? (Probably not)
- Is it *appropriate*? Is it an appropriate objective to have, bearing in mind the long term goal of our project? (Yes)
- Is it *realistic*? In the time we have available can we realistically expect to complete this task? (Probably – although until we start this task we won't know how much work we will need to do on it)
- Is it *time-related*? Have we identified how long the task will take and when we expect to complete it by? (We don't know yet but we will attempt to pin this down during project planning)

According to SMART, the problems we have with this objective at the moment are that it is not clearly measurable and we don't know how long it will take. Because of the nature of this objective we may well have to accept this as it stands. It is difficult to know how long a literature search will take as the search usually 'snowballs' as it continues. Our best approach will be to allocate a specific amount of time to this objective and draw a line underneath it at that point – on the understanding that we have done the best we can in the time available. However, until we complete our project planning, discussed in the following section, we don't know how long to allocate for this objective so we shall leave it open-ended for now.

4.3 Project planning

Although you are now clear about what you intend to achieve with your project, what you must now do is identify the work you need to do in order to fulfil these aims. Project planning assists you by identifying the work you need to perform, clarifying the order in which you should tackle the work, and revealing how long you need to do it. It is at this point that you may realise that your project is either overly complex or of insufficient depth for the requirements of your course. You may then decide to redefine your project (expanding or reducing its scope) before re-planning your work once more.

Project planning is performed through a series of six steps that utilise a number of project management techniques:

1. *Work breakdown*
2. *Time estimates*

3. *Milestone identification*
4. *Activity sequencing*
5. *Scheduling*
6. *Re-planning.*

Three techniques that are suitable for these stages are *Work Breakdown Structures*, *Activity Networks* and *Gantt charts*. Each of these techniques will be looked at in turn as the six steps of project planning are discussed.

4.3.1 Step 1 – Work Breakdown

Work breakdown structures (WBSs) are used to break your project down into lower and lower levels of detail to reveal exactly what work you will need to do to complete your project. You should begin a WBS by breaking your project down into its main objectives that you identified during your project's definition. You might only be able to break your project down into two or three main areas of work or you might be able to identify several broad areas of activity.

Figure 4.2 provides an example of a WBS for the Artificial Neural Network (ANN) stock market project introduced earlier. Five main objectives have been identified that need to be performed to complete this project. Notice how these tasks represent the five objectives identified earlier.

You should continue to develop your WBS by breaking your objectives down into lower and lower levels of detail. You may well find that some activities can be broken down further than others. For example, in Figure 4.2 the WBS has identified that the *Literature survey* will actually require the completion of a *Literature search* and a *Literature review* (although in Chapter 5 you will see that the literature survey process is much more complicated than this). To develop the ANN it will first involve

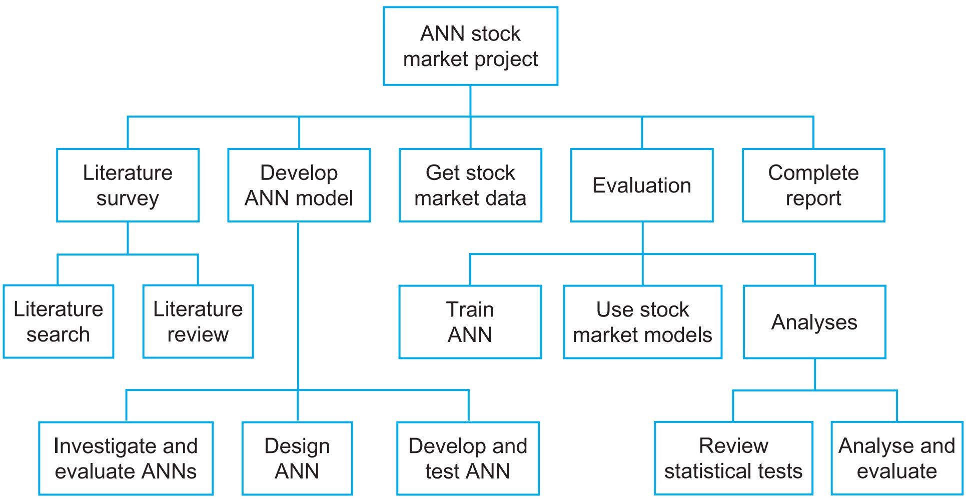

Figure 4.2 An example of a work breakdown structure

investigating and evaluating ANN topologies and tools (*Investigate and evaluate ANNs*) before designing the ANN (*Design ANN*) and then developing and testing it (*Develop and test ANN*).

Evaluation will involve three activities; training the ANN developed (*Train ANN*), using the market models evaluated from the literature review (*Use stock market models*) and performing an *Analyses* of the two approaches. Notice how*Analyses* has been broken down into another level of detail – showing that it requires a review of appropriate statistical tests and tools (*Review statistical tests*) before analysing and evaluating the results (*Analyse and evaluate*).

As you break down your project in this way you should ensure that tasks at all levels are separate from one another and an activity in one part of the structure is not repeated or revealed within another area of work. If this happens you may be duplicating effort on your project unnecessarily or your WBS may be incorrect.

You can continue to break these activities down further but you must stop somewhere; otherwise you could be identifying work which might take five minutes to complete! A general rule of thumb is that you should continue to break your project down into activities that take no less than around 5% of your project's total effort. For example, there is little point in identifying activities that will take you less than a week to complete in a six month project. If you do this you may spend more time adjusting and controlling your plans as your project progresses than actually doing any work (remember, only 10% of your time should be spent on project management activities). There are always unforeseen events in projects and activities will invariably take longer than you expect. Planning at too fine a detail is unwise as things will certainly happen to affect minutely planned activities before your project has finished.

4.3.2 Step 2 – Time estimates

When a project's aims and objectives are identified it provides little indication of exactly how long the project will take to complete. You would hope that your project is of a suitable scope to keep you busy during the allotted time and is of sufficient depth for you to obtain a good grade. However, it is not until you break the project down using a WBS that you really begin to see just how much work is involved.

Now that the project is broken down into a number of tasks it is much easier to estimate how long the project will take. It is far harder, for example, to predict how long it will take to complete the project's *Evaluation* than it is to predict the effort needed for individual tasks that make up that activity; *Train a neural network, Use stock market models* and perform the *Analyses*. You may, however, feel that these lower level tasks are still not explicit enough and there is nothing to stop you breaking them down further within reason. This is just what was done with the *Analyses* activity in the example.

Focussing now on the lowest level of the WBS it is possible to make reasonably accurate predictions of the effort needed to perform these activities and consequently the project as a whole. For example, using the WBS in Figure 4.2, the following time estimates in Table 4.1 can be made for the **lowest** level tasks:

By working backwards from our overall project duration we have also been able to allocate an appropriate amount of effort to our *literature search* and *literature review*. This goes some way to satisfying the SMART criteria applied to our preliminary project objective in the previous section.

Activity	Estimated Duration
Literature search	8 weeks
Literature review	4 weeks
Investigate and evaluate ANNs	4 weeks
Design ANN	4 weeks
Develop and test ANN	2 weeks
Get stock market data	1 week
Train ANN	1 week
Use stock market models	2 weeks
Review statistical tests	2 weeks
Analyse and evaluate	4 weeks
Complete report	8 weeks
Total Effort	40 weeks (approximately 10 months)

Table 4.1 Time estimates for example stock market ANN project

You should be reasonably happy with this estimate of the total project effort as it is much more accurate than you could have achieved working from the project's title alone. You might now realise that, perhaps, you have aimed to do too much in the time available (you might have only six months to complete the project) and need to reduce what you intend to achieve. Alternatively you might decide to allocate yourself less time to complete particular activities if you feel your estimates for these tasks were conservative.

4.3.3 Step 3 – Identify milestones

Milestones are significant steps towards the completion of a project. They help you to appreciate your progress by providing you with intermediate reference points. This enables you to assess, at the earliest opportunity, how you are progressing against your planned schedule. Because you know these milestones are leading you towards the ultimate goal of your project you can use these as intermediate goals at which to aim. Figure 4.3 provides a simple illustration of this point. In this figure the milestones are providing mini targets that you can use to focus your work in the short term.

To identify milestones you should focus on your project's work breakdown structure and identify any key stages that appear to be significant breakthroughs in your project's

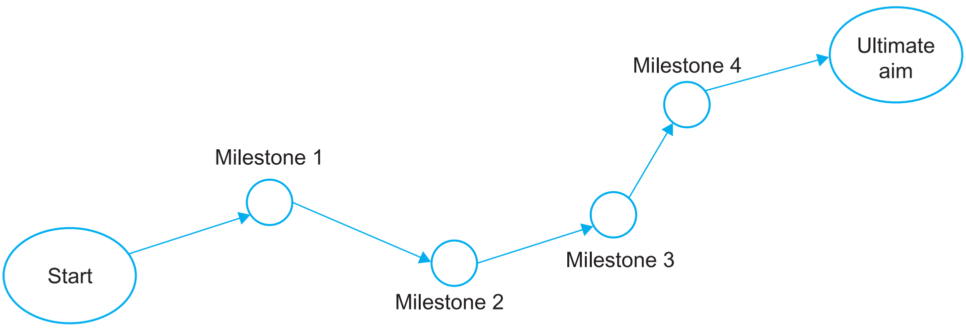

Figure 4.3 Milestones leading to the project's ultimate aim

progress. It is best to do this at the top level of the WBS and use some (if not all) of your project's objectives as milestones. These milestones then identify areas of work that, when completed, indicate you have achieved a significant step along the way. The number of milestones that you will identify for your project will vary depending on the project's size. For a year-long project, six or seven milestones should be more than adequate as these would represent, on average, the completion of approximately two months' work. Milestones can also be associated with the production of various reports, documents or sub-systems – for example an interim deliverable or a project proposal, etc. One milestone you must always identify is the project's completion.

For simplicity only two milestones will be identified in the example project we are looking at; the completion of the literature survey (milestone 1; M1) and the completion of the project as a whole (milestone 2; M2). M1 shows that a significant step has been made in establishing the project's foundation. You would expect to achieve this milestone within the first 12 weeks. M2 is the end of the project and clearly represents a significant event in the work! How these milestones are symbolised in the project plan is discussed in the following sections.

4.3.4 Step 4 – Activity sequencing

You now have an understanding of the work you need to perform in the project and the effort required to complete the individual tasks involved. An *activity network* can now be used to identify the order in which you should perform that work. Activity networks were first developed towards the end of the 1950s and are sometimes referred to as PERT networks, CPM or network diagrams. Two forms of activity network were developed at that time – Activity-on-the-arrow networks and activity-on-the-node networks. We will look at the simplest form of these diagrams in which activities are represented by rectangles or *nodes* – the activity-on-the-node network.

Activity-on-the-node diagrams represent the tasks you are performing in your project as nodes connected by arrows. The arrows show the order in which activities must be performed. For example, in Figure 4.4, Task A can start at any time as it does not rely on any other task completing. Task A would therefore start at the beginning of the project. Task B cannot start until Task A has finished and Task D can only start after **both** tasks B and C have completed successfully. Task C is similar to Task B in that it cannot start until Task A has ended.

If this representation is applied to the example stock market project introduced earlier it results in the activity-on-the-node representation shown in Figure 4.5. In this example the completion of the project's report has been identified as an activity that is

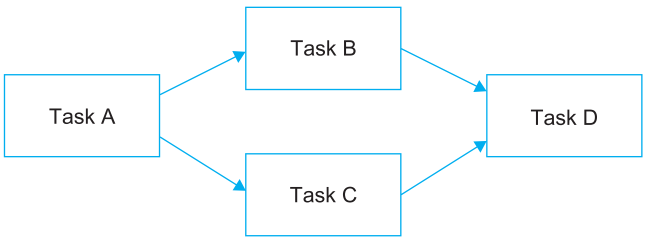

Figure 4.4 An example of a simple activity-on-the-node diagram

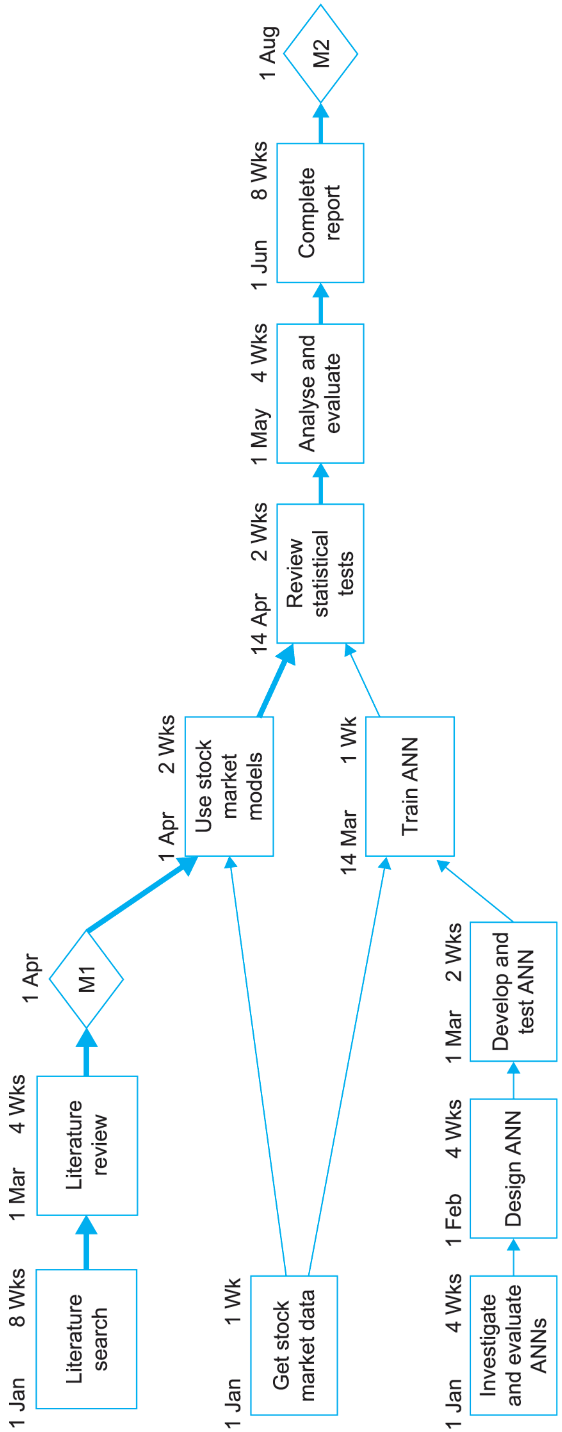

Figure 4.5 An example of an activity network

performed during the last eight weeks of the project. In reality, however, you would probably be working on your project's report throughout the lifetime of your project and the activity identified here really represents the final drawing together of the report; checking and completing your references, writing your abstract and contents listing, proofreading and spell-checking.

There are three additional markings to Figure 4.5 that are not shown in the example of Figure 4.4, which have yet to be explained. The first point to note is that the milestones identified earlier have been included as diamonds called M1 and M2; M1 being the completed literature survey and M2 representing the completed project. Notice how these have been placed in the relevant positions on the diagram and represent the completion of the significant step they are identifying.

The second point to note is that dates and durations have been added to each task node. Each activity now has two figures – the start date of the activity, shown at the top left of each node, and the duration of the activity (in weeks), shown on the top right. These durations are taken from the time estimates made earlier in Table 4.1. It is up to you what time 'granularity' you use for your project (hours, days, weeks, months or even years) but, in a student project of this size, weeks or months are suitable.

For simplicity it will be assumed that a month consists of exactly four weeks and there are no breaks in the project for holidays! However, in reality holidays, sickness, revision, field trips and the like can often impede your progress and these events should be considered when forming project plans.

The date, which is noted at the top left-hand corner of each activity, represents the time at which that activity can start. For activities that can start straight away (i.e., they do not need any other activities to have completed beforehand) this is simply the start date of the project. In the example, three activities can start straight away – *Literature search, Get stock market data* and *Investigate and evaluate ANNs*. All these activities have the same start time – 1 January – which represents the start date of the project overall (this start date has been chosen for simplicity).

To calculate the start times of subsequent activities it is necessary to look at the tasks leading into them. For example, in this simple case, *Literature review* can begin as soon as the *Literature search* has completed. As the *Literature search* takes eight weeks to complete (approximately two months) the *Literature review* can begin from 1 March onwards. The first milestone (M1 – complete literature survey) occurs when this review is completed and consequently, as the review takes four weeks to complete (approximately one month), M1 will (should) occur on 1 April. Notice how M1 has no duration associated with it because it does not represent any work but simply an event in time.

Calculating the start time for activities with more than one task leading into them is not quite so straightforward. When two or more tasks lead into another, that task can only start when **all** its preceding tasks have completed. For example, in Figure 4.5 notice that *Train ANN* starts on 14 March – this is when you would expect *Develop and test ANN* to complete, **not** when *Get stock market data* has completed (which is due to finish on 7 January). Remember that subsequent activities can only begin when **all** preceding activities leading into them have completed.

Continuing with the calculation of start times for each activity in the project, the final milestone, M2 (completed project), is reached. Thus, it is possible to conclude that the project should be completed by 1 August. However, this may be optimistic, as it does not account for any delays or problems that might occur.

The final additional marking to this network diagram is the *critical path* which still requires explanation. This path is the longest route through the project network and is represented in Figure 4.5 by the bolder arrowed lines. It identifies the activities in the project that must not be delayed, as to do so will delay the project overall. For example, if *Complete report* were to take twelve weeks instead of eight, the project would finish on 1 September – four weeks later than before. This is because *Complete report* lies on the critical path and any delay to this activity will therefore affect the project overall.

To identify the critical path you work backwards through the network diagram from the project's final milestone. Begin, therefore, at M2 and look to see which task(s) leading to this milestone is causing it to occur on 1 August. There is only one activity in this case leading into M2, *Complete report,* so this task is on the critical path. Looking next at *Complete report* we see, once again, that only one task leads into it – *Analyse and evaluate.* Consequently, *Analyse and evaluate* must also be on the critical path.

You continue to work your way backwards through the network, along the critical path, until you either reach the project's start or an activity that has two or more activities leading into it. In the latter case, *Review statistical tests* is the first activity in this situation. However, just as before, you look back through the network to see which activity(ies) is (are) forcing *Review statistical tests* to start on 14 April and see that, in this case, it is *Use stock market models* **not** *Train ANN. Use stock market models* is also, therefore, on the critical path. Continue in this way, working backwards through the network, until you reach the start of the project – in this case ending up at the *Literature search.* The critical path is thus identified by the bolder arrowed lines linking each of these critical activities together.

There is no reason why you cannot have more than one critical path in your project network. In some cases, two or more activities may force a subsequent task to start on a particular date. In these cases, proceed as before, following all critical paths back to the start of the project or to a point where they rejoin. The activity network is now complete.

This representation has made several assumptions. The first is that you can perform several tasks simultaneously. This often happens in computing projects where you might, for example, be performing aspects of your literature search and literature review alongside an initial systems analysis or program design. This also allows you to avoid becoming bored with one activity or another because you can switch between them as your project progresses. However, although identifying several simultaneous tasks may be satisfactory for group projects, where several members of the project team can work on tasks separately, for individual projects this can cause a problem. To identify instances when you are expecting to work on too many activities simultaneously and to see how you can deal with this problem, you must use a *Gantt chart,* which is introduced in the following section.

The second assumption made is that once activities are completed, they are finished with and your project moves on. In reality, however, many activities are ongoing throughout the lifetime of your project, for example, the literature survey and report writing. However, emphasis on these ongoing activities changes as the project progresses. There are also situations where activities are repeated and you find yourself performing a loop – for example, the literature search and literature review, which are part of the repetitive literature survey process discussed in Chapter 5. An example of loops occurring within software development projects is when an evolutionary prototyping approach is used. This approach is discussed in detail in Chapter 6.

These situations cannot be planned explicitly using ordinary activity network diagrams and, although there are networking techniques that can be used to identify repetition and loops, they are not widely available. Consequently, project planning tends to identify distinct activities that occur either in parallel or in sequence and limits activity network plans to these representations.

4.3.5 Step 5 – Scheduling

Gantt charts are similar to activity networks in that they attempt to represent a project in diagrammatical form. However, unlike activity networks, they do not show the relationships between tasks, but they do explicitly show the durations of activities and identify instances when tasks are performed simultaneously.

Just like activity-on-the-node networks, Gantt charts represent a project's activities as rectangles or nodes, and milestones by diamonds. In this case, however, the size (length) of an activity's node represents the duration of that activity. For example, in Figure 4.6 a Gantt chart has been put together for the example stock market ANN project. The scale running along the bottom of this chart represents the dates during which the project is performed. Notice how each activity in this chart is represented by a rectangle, which is as long as the activity's estimated duration. For example, the *Literature search* lasts for two months starting at the beginning of the project. It is therefore drawn up until 1 March. The *Literature review* follows on from the *Literature search* and lasts for one month – again shown by the length of the task box – finishing by 1 April. Milestone 1 (M1) is shown at the appropriate point at the end of this task.

It is important to keep an eye on the activity network when drawing a Gantt chart to ensure that activities are performed in the correct sequence and activities do not start in the Gantt chart before all their preceding tasks have completed. Some Gantt charts allow you to include the arrowed connections between activities like those in the activity network. However, trying to include all this information on one diagram does mean it is very messy and difficult to follow. It is therefore best to use both these techniques simultaneously – each one complementing the other.

In Figure 4.6 notice how activities that do not fall on the critical path of the activity network have shaded extensions to them. These shaded areas represent an activity's *slack* or *float* time. Remember that activities on the critical path cannot be delayed without delaying the project overall. This implies that activities which don't lie on the critical path can be delayed to some extent without affecting the project. The extent to which an activity can be delayed without affecting the project is called its *slack* or *float*.

To identify slack time in your project, you need to focus on activities that do not lie on the critical path. Work your way backwards through your project (starting at the project's completion milestone as before) until you meet one of these activities. In the example, looking at the activity network in Figure 4.5, the first activity encountered like this, working back through the network, is *Train ANN*. *Train ANN* leads into *Review statistical tests* (which *is* on the critical path). As long as *Train ANN* is not delayed for so long that it starts to delay *Review statistical tests*, the project will not be affected. Thus, *Train ANN* could be delayed so that it finishes no later than the start of *Review statistical tests* (i.e., 14 April). This delay represents the slack time of *Train ANN* and is shown as the shaded area in Figure 4.6. In this case the delay can be up to three weeks before the impact is felt on the project overall.

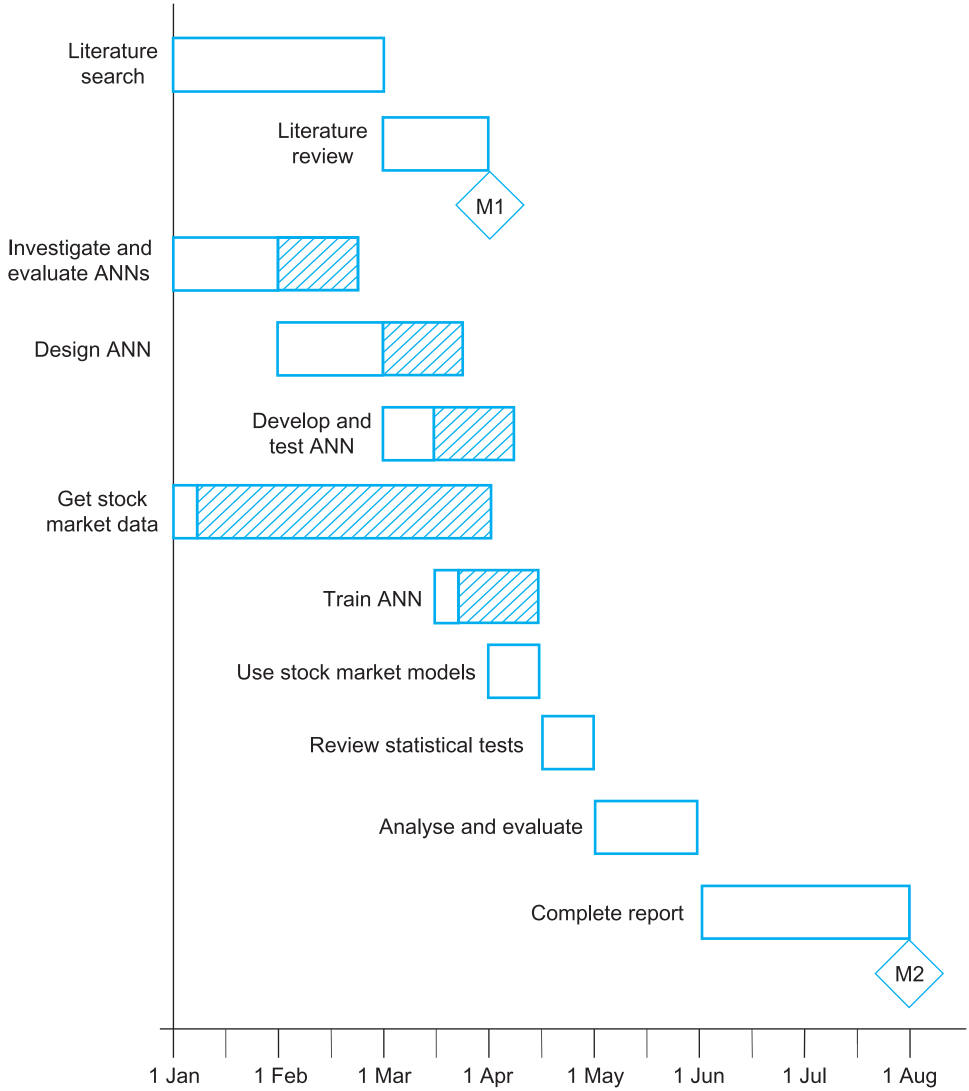

Figure 4.6 An example Gantt chart

Working backwards from *Train ANN* we will now look at *Develop and test ANN*. Because this task leads directly into *Train ANN*, this activity may also be delayed by three weeks without affecting the project overall. This is because we know that delaying *Develop and test ANN* by three weeks will only delay *Train ANN* by three weeks, which is fine as *Train ANN* has three weeks of slack. The same argument is applied to *Design ANN* and *Investigate and evaluate ANNs*. Consequently, these activities have the same float applied to them as *Train ANN* (i.e., three weeks).

Get stock market data is the only other activity not lying on the critical path that still needs considering. This activity must complete before *Use stock market models* and *Train ANN* can begin. As *Use stock market models* is on the critical path, we must ensure that

Get stock market data finishes no later than the critical start time of *Use stock market models*. Thus, *Get stock market data* can be delayed so that it finishes no later than 1 April (the start of *Use stock market models*). Notice, if it was, that it would also delay *Train ANN* by two weeks. This is acceptable because *Train ANN* is not on the critical path (it has three weeks of slack) and so delaying this task by two weeks will not impact on the project. Be careful in these situations however, to ensure that if a task is allocated slack that this will not impinge on other non-critical tasks to such an extent that they become critical and delay the project overall.

Notice that some of the slack we have identified on tasks relies on subsequent activities with slack being delayed (for example, *Get stock market data* running into *Train ANN*), while other activities with slack do not rely on subsequent non-critical tasks moving (for example, *Train ANN*). This implies that there are different kinds of slack that can be identified in Gantt charts. In fact, there are four kinds of slack one could identify in a project plan – *free* slack, *independent* slack, *interfering* slack and *total* slack. However, for the purposes of a student project, you do not need to go into this level of detail on the Gantt chart. What we have identified in this case is simply *total* slack, which is sufficient for projects of this nature. The total slack for a task represents the amount of time that task can be delayed by, assuming all other tasks with slack are able to be delayed to accommodate it where necessary. The Gantt chart is now complete.

What this chart now highlights is that there are times when you need to perform more than one activity at a time. For example, looking at the first week of the project in January you will see how you should be working on the *Literature search, Investigate and evaluate ANNs* and *Get stock market data* all at the same time. For group projects this is not a problem as these tasks can be assigned among team members. However, for individual projects, this might well be unacceptable and something needs to be done about it.

One solution might be to use the float time on various activities. For example, *Get stock market data* could be delayed for a few weeks without affecting the project overall and it would reduce the number of activities that needed to be performed during the first week of the project. However, this is only putting off the inevitable. At some stage in the project, *Get stock market data* will have to be tackled and it will inevitably clash with some other work at that time. The problem lies in the fact that ten months' worth of work (40 weeks) is being attempted within seven months (1 January to 1 August) with only one person available. This is impossible unless you are able to do more than one activity at a time. If you cannot, you must accept that your project will take ten months to complete and you should adjust your Gantt chart accordingly.

Project management software packages are well suited to these kinds of problems – known as *scheduling*. They attempt to schedule out people's (resources') time on projects in order to achieve a balanced allocation of work over a project's life span.

In this case, a popular project management package called Microsoft Project has been used. Figure 4.7 shows a print-out of a Gantt chart from this package for the example ANN stock market project. Notice how similar this is to the representation shown in Figure 4.6. Microsoft Project was then used to re-plan the project on the understanding that only one person was available to do the work. Microsoft Project rescheduled the plan to that shown in Figure 4.8. Notice how the project is now scheduled to last for ten months, finishing at the end of October, and only one activity is being performed at any one time. However, this is not necessarily an ideal solution as, for example, there now

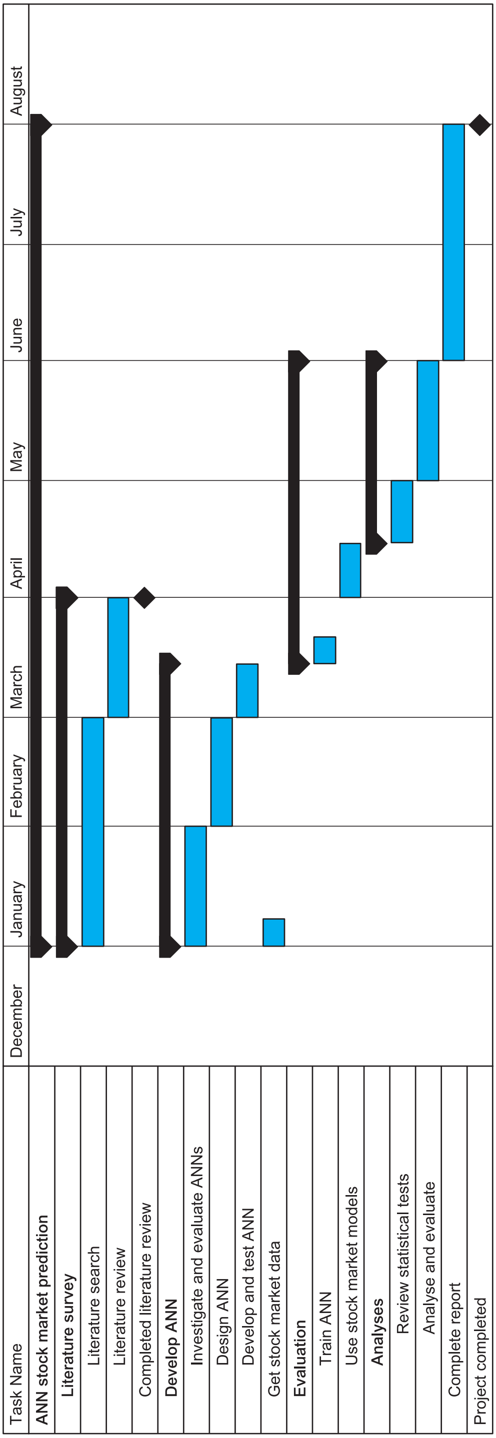

Figure 4.7 Gantt chart using Microsoft Project

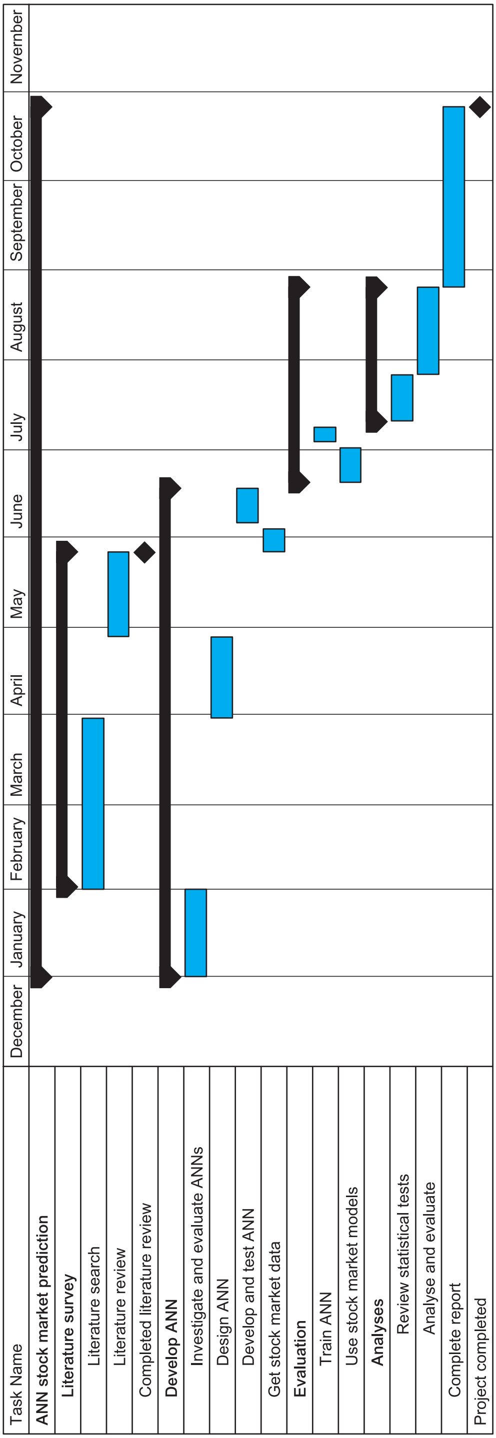

Figure 4.8 Scheduled Gantt chart of example project

appears to be quite a delay between performing the *Literature search* and writing the *Literature review* – two activities that, in reality, are closely intertwined. With this in mind, you should always pay close attention to scheduling adjustments that are made by project management software tools.

4.3.6 Step 6 – Re-planning

Now that you have completed all your plans, you may realise you are trying to do too much in the time available. Re-planning simply means that you go back through your plans, adjusting and rescheduling them accordingly. Project management software tools are particularly useful for making these changes and assessing the impact of your adjustments. However, try not to spend too long on this stage by getting drawn into the usability of these tools and end up using them for their own sake. You may find yourself re-planning and rescheduling at minute levels of detail rather than getting on with the 'real' work of your project (remember, the 'real' work should take around 90% of your effort).

Note, also, that plans you have produced should not be cast in stone. For instance, in the example project, you may find that after completing your investigation of ANNs you decide that it might be more appropriate to use an off-the-shelf package rather than develop your own ANN model. This will clearly lead to some reworking of the plan and may release some time later in the project for you to concentrate on other activities.

4.3.7 Rolling wave planning

A technique that can help you when your project is not all that clear is *rolling wave planning*. Rolling wave planning means that you do not construct a detailed plan at the project's inception but a *skeleton plan,* which only identifies the key stages of your project. Your project planning is thus performed 'on the fly' as your project progresses. You make decisions as to where you are actually heading and what work you will have to perform in the subsequent stage of your project, as you complete the previous stages. Thus, your planning detail ebbs and flows (like a rolling wave) as your project progresses and you make decisions on where to go and what to do next.

As a skeleton plan is relatively broad it can be suitable for many projects. Although it is of little use if you don't have *any* idea of what you want to do, it can help you to identify universal milestones that you must adhere to – for example, complete a literature survey, hand in your final report, etc. – whatever these turn out to be. Figure 4.9 provides an example of a typical rolling wave skeleton plan – in this case a software development-type project that lasts for about six months. Although this plan does not provide explicit detail about what this project is really about, it does identify the significant tasks that need to be completed and by when.

4.3.8 Project Initiation Document

The project initiation document (PID) is a document that draws together many of the sections discussed above in one place, representing a definitive overview of the project – its purpose, objectives, outline, plan, risks, etc. It can form a contract in terms of defining what the project will achieve. PIDs come in various shapes and sizes with different content

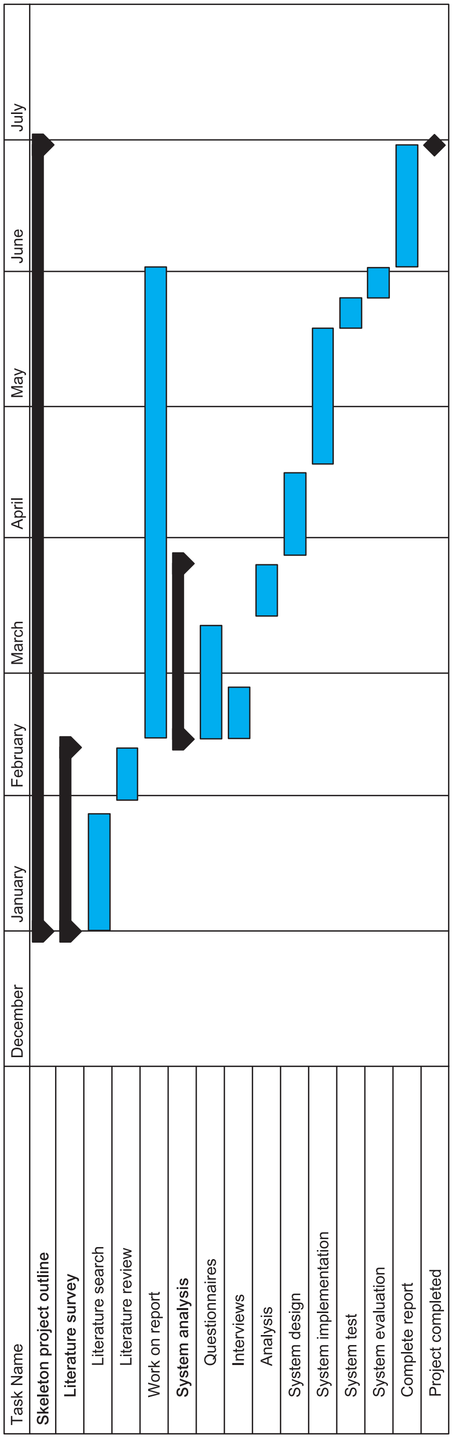

Figure 4.9 Example rolling wave skeleton plan for a software development project

requirements. Many companies have their own definitions of what a PID should contain. It is always a good idea to put together a PID at the start of your project. The components used in the project proposal (see Section 3.3) can form the basis of your PID, namely:

- Title
- Aim and objectives
- Expected outcomes/deliverables
- Introduction/background/overview
- Project type
- Related research
- Research question/hypothesis
- Methods
- Resource requirements
- Project plan.

These may need some updating once your project is accepted as you may have made some progress and made some changes since the proposal was put together. In addition it would be worthwhile including the following three sections in the PID:

1. **Risks.** Include a list of critical risk factors and means of dealing with these risks should they occur. How might you reduce their impact or limit their chances of occurring in the first place? Risk management is covered in the following section and this will give you some ideas of what to include in the PID.
2. **Organisation.** If you are undertaking a group project it would be worthwhile outlining how your team will be organised; who is the team leader, secretary, spokesperson; how often the team will be meeting; communication arrangements (email addresses, web site, etc.); configuration management, etc.
3. **Milestones.** Now that your project plan is completed you can identify the major milestones within the PID. These milestones should include both project specific ones (i.e., any deliverables you are producing) and generic, course related ones (for example, deadlines for interim reports, final reports, presentations, etc.).

You might to keep you PID in a project folder so that you can refer to it easily as your project progresses. Sections such as the project plan and risk plan may need to be readily available and accessible. The idea of establishing a project folder is covered in Section 7.1.2 when project initiation is discussed.

4.4 Risk management

4.4.1 Introduction

Risk management is a process that is inexorably linked to project management; it runs in parallel with project management and follows a very similar process. Just as project management involves the development of a project plan and control of the project using that plan as the project progresses, risk management involves the identification of risks at the project's outset and the control of those risks as the project unfolds. As you undertake the stages of project management you will also be incorporating the activities of the risk management process. In this section we introduce a risk management process that you can use to manage and control risks within your own project. The four main stages of this risk management process are:

1. *Identify risks*
2. *Assess impact of risks*
3. *Alleviate critical risks*
4. *Control risks.*

4.4.2 Identify risks

As you are putting together your project plan, you should also be identifying any *sources* of risk to your project. These risks can be individual events (*event-driven risks – acute*) that might have an impact on your project (for example, your supervisor leaving, your hard disk crashing, etc.) or they may be longer term risks that evolve over time (*evolving risks – chronic*) before eventually coming to a head (for example, underestimating the time it will take you to develop part of your system, deteriorating relationship with your client, etc.).

Whether the risks to your project are *event-driven* or *evolving*, they can be further classified as either *technical* or *non-technical* risks. Technical risks refer to any risks that are associated with the hardware or software you might be using. For example, will there be problems interfacing the components of the software and hardware system you are developing? How well do you know the programming language and platform you are going to work with? Is your project dependent on the development of an algorithm that may be difficult, if not impossible, to develop? Is the specification for the system clear? Are the requirements likely to change? Is the project beyond your technical ability? What are the chances of your hard disk crashing and you losing all your data?

Non-technical risks are all other risks associated with your project. These can include such things as losing your client, your user or your supervisor; illness; over-running your time estimates; discovering work during your literature search that already covers (in depth) what you intended to do; etc.

As well as identifying potential risks to your project it is also useful to identify risk *triggers* (sometimes called risk *symptoms*) at this stage too. Risk triggers are events or things that happen during the course of your project (not necessarily 'bad' things) that might give an indication that something is wrong or that one of the risks you have identified is increasingly likely to occur. They are useful in that they give you warning, ahead of time, that something will happen and you can be better prepared to deal with it when it does. For example, missing preliminary milestones in your project is a good indication that your project is going to over-run; struggling with a straight-forward implementation of a component in a new programming language will probably mean you will encounter severe implementation difficulties later on; having difficulty in arranging a meeting with your client early on may be a good indicator that s/he may be difficult to contact and meet later when you are desperate for feedback towards the end of your project.

All projects encounter problems to some extent, so don't be disheartened if some of the risks you identify during this stage of the risk management process occur. The point of risk management is to ensure you are in a position to deal with these risks if they do occur and you are not facing them ill-prepared.

4.4.3 Assess impact of risks

Having identified the risks associated with your project in the first stage of the risk management process, you should then calculate their *impact*. The impact of any risk on your project is given by the following equation:

$$\text{Risk impact} = \text{likelihood} \times \text{consequence} \tag{4.1}$$

Thus, although a risk may be highly likely to occur, if its consequences are low its impact will also be low. Similarly, if a risk has severe consequences for your project but its chances of happening are very low, its impact is also calculated as low. The risks you need to worry about are those that are highly likely to occur and have significant consequences to your project if they do.

Example

Turner (1993: 242) provides an interesting illustration of how the risk impact equation works in practice. If we consider the likelihood of a severe earthquake (say, greater than Force 7 on the Richter scale) occurring in the British Isles, we would probably conclude that the chances of this occurring are small. We then consider the consequences of this risk on two different kinds of building – a car park and a nuclear power station. In the case of a car park the consequences of a severe earthquake are quite small – a few cars may get damaged and (unfortunately) a small number of people may be injured or killed if it collapsed. As the consequences of the risk are relatively low (unless it is your car that gets crushed!) and the likelihood is low, the overall impact of this risk is also deemed to be low. Thus, in the British Isles, car parks are not constructed to be 'earthquake-proof'.

In the case of the nuclear power station however, the consequences of a major earthquake could be catastrophic – hundreds of thousands of people could be killed. In this case, while the likelihood of the earthquake is the same as for the car park, the consequences are much higher – leading to an overall risk impact that is deemed very high. Thus, nuclear power stations within the British Isles are constructed to withstand such events.

Turner (1993: 256) goes on to provide a quantitative measure for assessing the risks to your project. A risk's likelihood is classed according to a three-point scale – Low/Medium/High. Similarly, a risk's consequence is measured on a five-point scale – Very Low/Low/Medium/High/Very High. Turner assigns numbers to these measures as shown in Tables 4.2 and 4.3.

By assessing each of the risks to your project according to these scales, you can determine a risk's impact as a value between 1 and 15 based on Equation 4.1 (1×1 being the

Risk Likelihood	Score
Low	1
Medium	2
High	3

Table 4.2 Risk likelihood scores

Risk Consequence	Score
Very Low	1
Low	2
Medium	3
High	4
Very High	5

Table 4.3 Risk consequence scores

lowest score and 3×5 being the highest possible score). For example, suppose we feel that there is a small chance that we may lose our client during the course of our project (s/he may have told us in a preliminary meeting that their department may be restructured during the next six months – risk likelihood is *medium;* 2), yet the consequences of this are quite severe (how would we assess our final system? We might feel the consequences are therefore *high;* 4). The overall risk impact of this risk is $2 \times 4 = 8$.

Although this (dimensionless) number doesn't really mean anything as it stands, it does provide us with a *relative* measure that we can use to compare all the risks we identify in our project. We can then rank the risks to our project according to this measure and begin to focus on the 'critical' ones.

Depending on how many risks you identify you can choose to categorise *critical risks* in one of two ways; either the 80/20 rule or by impact factor with *RAG grading*. The 80/20 rule works on the theory that approximately 20% of your risks will cause approximately 80% of your problems. You should therefore focus on these critical risks and be confident of addressing 80% of the problems your project is likely to face.

In reality, your risk list may not partition easily into a 80/20 split, or there may be another natural break point where three or four risks appear significantly 'riskier' than others. You may therefore decide to focus on critical risks by impact factor alone – for example, those that have an impact factor greater than nine. One approach you can use here is *RAG grading*. RAG stands for Red, Amber, and Green and is used to classify risks according to their impact factor. Those with an impact factor of 1 to 5 are classified as green risks; those with an impact factor of 6 to 10 are amber risks; those with an impact factor greater than 10 are red risks. It is the red risks (critical risks) that you need to focus your attention on. Although not critical, you should keep a wary eye on amber risks, and green risks can generally be ignored.

Whichever approach you use, you will end up with a short list of critical risks that you are going to do something about. How you deal with these critical risks is discussed in the following section.

4.4.4 Alleviate critical risks

There are three ways that you can deal with the critical risks you have identified in your project: *avoidance*, *deflection* (sometimes called *transfer*) and *contingency*. Avoidance means reducing the chances that the risk will occur at all. Deflection means passing the risk onto someone or something else. Contingency means accepting that the risk is going to occur and putting something in place to deal with it when it does.

Example

Pym and Wideman (1987) provide a neat analogy that contrasts these approaches. Take the situation in which someone is about to be shot at. They can avoid this risk by moving quickly to somewhere safe (avoidance). They can deflect this risk by putting something (a shield) or someone between themselves and the assassin (deflection). Or, they can assume they are going to be hit by some bullets and ring the ambulance service in advance (contingency). You may like to think which of these approaches to dealing with this risk *you* prefer!

The nature of the risks within your own project will influence which of these approaches are suitable. Although your supervisor should be able to give you advice in this area, the following examples may provide you with some ideas you could adopt in your own project.

- **Contingency.** If you feel there is a chance some tasks may over-run in your project it might be an idea to build some contingency into your project plan at the start. You could, for example, aim to complete your project four weeks ahead of its submission date – giving you four weeks of flexible time (float) that can be used if unexpected delays occur. Alternatively, you might like to add 10% to each of the time estimates you have made to cover any possible delays.
- **Deflection.** You might be able to get someone else to do part of the work for you in an area where you are weak (although you would have to acknowledge this clearly in your report). For example, you might be undertaking a research-based project focussing on the impact of a particular software technology in an organisation. The project may require the development of a software system (and your programming skills may be weak) but this is not the main focus of your work. It might, therefore, be possible, to get someone else to develop that software system for you while you remain focussed on the important evaluation component of the project. This is certainly something that would need approval from your institution or supervisor to ensure that the integrity of your own work was not compromised.
- **Avoidance.** If there is a risk that you will not be able to develop a program because you are considering using a new programming language, perhaps you could resort to using a language with which you are familiar.
- **Avoidance.** If your software system requires the development of an algorithm to solve a problem and you are unsure whether you can do this or not, you might be able to use an existing algorithm instead (but make sure this does not reduce the potential marks you can achieve for your work).
- **Contingency** and **Avoidance.** Is there a possibility that your computer might crash and you might lose all your work? Contingency planning would involve arranging an alternative machine you could use if such a problem occurred. Avoidance would involve making sufficient back-ups of your work to ensure that your work would not be lost if this happened.

4.4.5 Controlling risks

You have now identified all the risks in your project, focussed on the critical ones and decided what you will do about them. This is not the end of the process; you cannot just sit back and relax. While risk management can make you aware of the risks involved and put things in place to deal with those risks, it does not take away the risks to your project entirely. The last stage of the risk management process is concerned with planning your risk strategy approach, monitoring risks as your project progresses and dealing with those risks if they occur. In Chapter 7, when we look at controlling your project as it is progressing, we will examine some specific problems your project might face in more detail (and ways to deal with those risks). This section is concerned with the process of controlling the risks rather than the risks themselves.

The first task you must undertake as part of this stage is to decide how you will manage the risks you have identified. This is not in terms of how you will *deal* with those critical risks, as this was covered in the previous section. This task involves deciding how you will go about controlling risks in your project and making sure you have the right resources available to deal with the risks – in other words, what is your *strategy* going to be towards risk management in your project?

One strategy you can adopt is to identify specific *checkpoints* in your project's progress, when you will re-visit your critical risk list and adjust it according to your latest understanding. However, be aware that triggers or risk events can still occur outside these checkpoints and should be dealt with accordingly. You may decide to hold checkpoints at the end of every week, fortnight or month. Alternatively, you may decide to hold check points at the end of particular stages of your project – for example, after the literature search is completed, after the design is completed and so on. Another alternative would be to arrange checkpoints during meetings with your supervisor – s/he may wish to know how your project risks are changing.

Part of the process involves checking for those risk *triggers* you identified during risk identification. Another part involves invoking your contingency plans when risks occur, while another part requires you to constantly monitor and update your critical risk list. Risks are not stationary; they will evolve over time. Some risks will become more 'risky' (their impact will increase), while others will become less so. As your project progresses some risks may be 'promoted' to your critical list, whilst others may be 'demoted' from it. Risks can therefore move from green to amber to red risks and vice versa in the RAG grading scheme.

4.5 Summary

- Project planning consists of two stages; defining what it is you want to achieve and planning how you will achieve this. Project definition involves identifying your project's aims and objectives.
- Planning itself consists of six steps; identifying the tasks involved using *Work Breakdown Structures;* estimating the duration of these tasks; identifying critical stages in your project called *milestones;* identifying the order in which activities should be performed using *activity networks;* scheduling your time so that you

are not trying to do more than you can physically achieve using *Gantt charts;* and re-planning your project to fit the time available.

- Project management software packages, such as Microsoft Project, can be used to assist you with planning and managing your project. While you can put together your own Gantt charts and activity networks by hand, such as those shown in Figures 4.5 and 4.6, project management software tools can automate this process for you. However, these packages do take time to learn and you can often find yourself spending more time planning and 'tweaking' your project with these packages than actually doing any real work. Remember – you should only spend around 10% of your time at most on project management.
- Risk management is performed in parallel with project management and involves the following four stages: *risk identification, risk quantification, risk alleviation* and *risk control.*

4.6 Further reading

Barker, S. and Cole, R. (2007) *Brilliant project management: What the best project managers know, say, do*, Prentice Hall, Englewood Cliffs, USA.

Levine, H.A. (2002) *Practical project management: tips, tactics and tools,* John Wiley, New York.

Lock, D. (2007) *Project management* (9th Edition), Gower, Aldershot, UK.

4.7 Exercises

1. Try to identify objectives for the example projects listed in Section 4.2.2.
2. Apply the SMART criteria to the remaining objectives in the example in Section 4.2.3.
3. Try to state whether the technical and non-technical risks identified in Section 4.4.2 are *event-driven* or *evolving* risks.
4. What are the risks associated with the ANN stock market project? Can you identify any triggers in this project?

4.8 Action points

- Identify aims and objectives for your own computing project.
- Apply the SMART criteria to your own project's objectives.
- Follow the six steps of planning to complete your project's plan.
- Apply the four stages of the risk management process to your own project. What are the three main risks to your project and what do you intend to do about them?