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

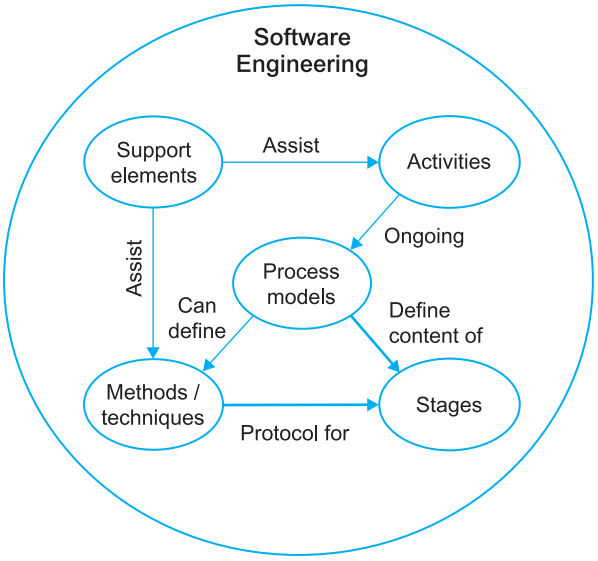

Figure 3.1 A high-level RTM for *software engineering*

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

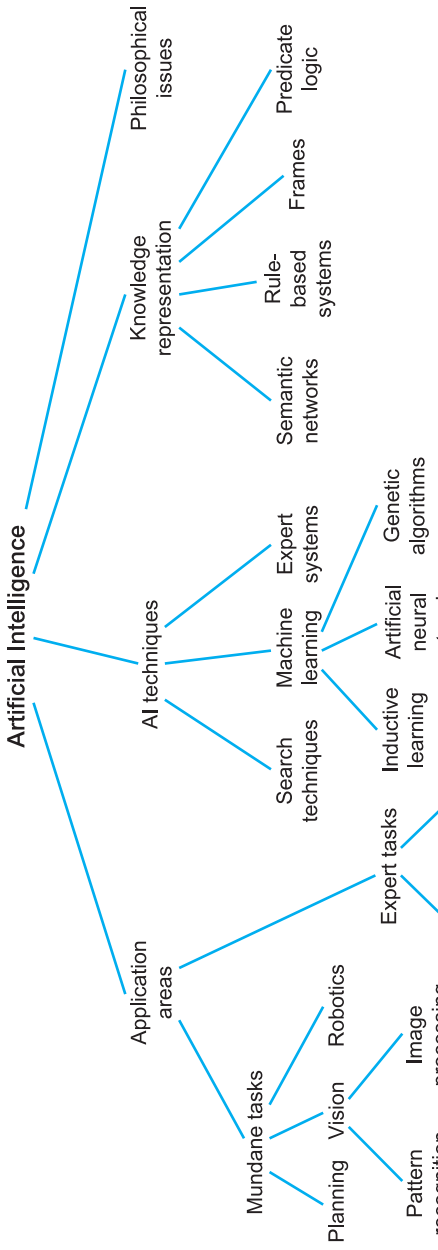

Figure 3.2 Example of a relevance tree for *artificial intelligence*

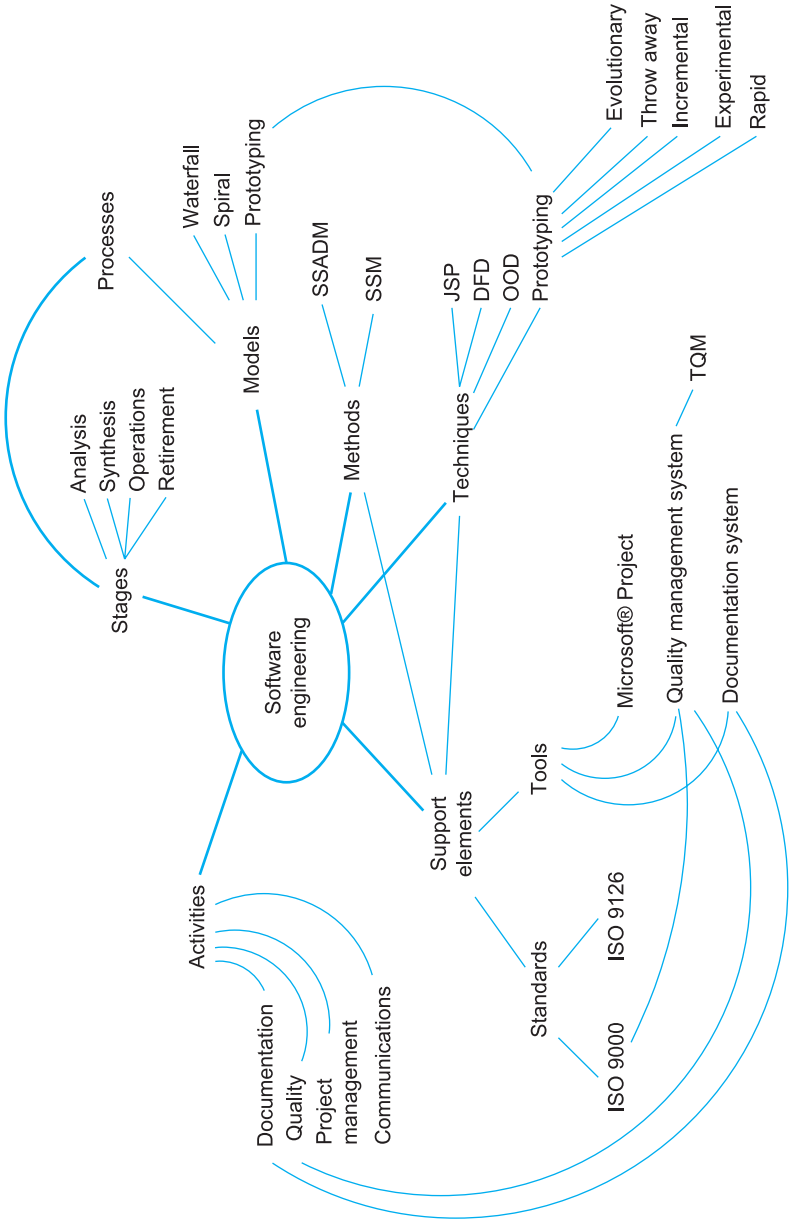

Figure 3.3 Example of a spider diagram for *software engineering*

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?