CHAPTER 3

Project planning

Aims

To introduce the basics of project planning.

Learning objectives

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

- describe the typical stages of an academic computing project;
- 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.

3.1 Introduction

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. At this level interest lies 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 these stages. 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. Project definition is the preliminary stage of this process and includes the activities presented in Chapter 2: 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 identifying and deciding on how to approach the work you need to perform. Project definition and project planning are the focus of this chapter.

The following three stages of the process represent the main 'bulk' of your project work – that is, actually 'doing' your project and working on developing the project's product. The project's product in this context represents a product in the widest sense of the term. Your project's product would be your written report, a fully documented piece of software, a new model or algorithm, a literature survey, a case study and so on. The product represents your project's expected outcome and deliverable.

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 definitive 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.

Once you have organised yourself, and your project is under way, you will need to *control* it as it is progressing. Project control is covered in detail in Chapter 5. The last stage of any project is *closure*. In your computing project this will represent the final 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 6 and 7.

3.2 Project definition

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 2). In many ways your project definition and your initial project proposal are closely linked. They should both be written at the start of your project. 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 2 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 at 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?

3.2.1 **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 the UK.
- To develop and evaluate a user interface for statistical software packages.
- To design a methodology for GUI development of technical courseware material.
- To 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.

3.2.2 **Setting objectives**

As Chapter 2 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:

- To develop and evaluate an artificial neural network to predict stock market indices.

Project's objectives:

1. Complete 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 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 section, which discusses how to break down the actual work you will need to do to complete your project using *work breakdown structures*.

3.3 **Project planning**

Although you are now clear about what you intend to achieve with your project, what you now need to identify is 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 replanning your work once more.

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

- Work breakdown
- Time estimates
- Milestone identification
- Activity sequencing
- Scheduling
- Replanning

Three techniques that are suitable 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.

3.3.1 **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 3.1 provides an example of a WBS for the artificial neural network

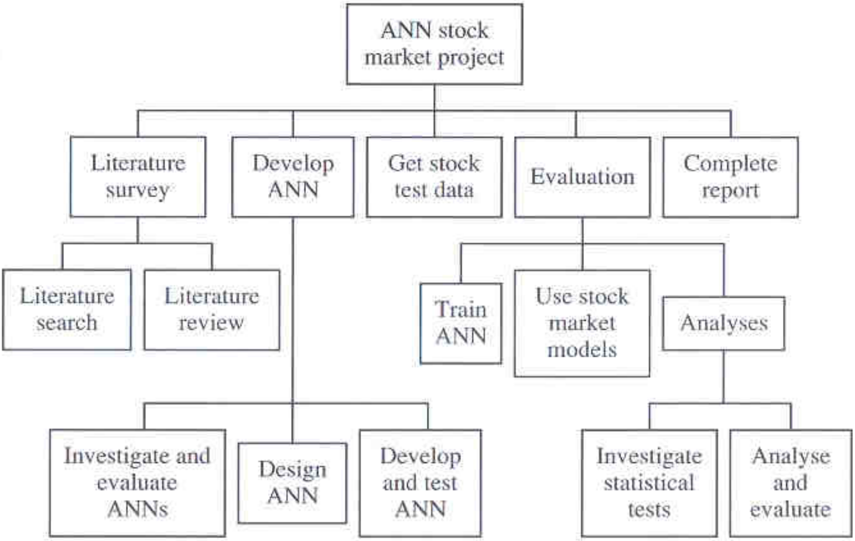

Figure 3.1 *An example of a work breakdown structure*

(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 3.1, it has been identified in the WBS that the *Literature survey* will actually require the completion of a *Literature search* and a *Literature review* (although in Chapter 4 you will see that the literature survey process is much more complicated than this). Developing the ANN will first involve 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 (*Train ANN*), using the market models evaluated from the literature review (*Use stock market models*) and performing *Analyses* of the two approaches. Notice how *Analyses* has been broken down into another level of detail, showing that it requires an investigation and application of appropriate statistical tests and tools (*Investigate 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 that 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 duration. 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. 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.

3.3.2 Step 2: Time estimates

Identifying a project's aims and objectives 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 can 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 project.

Focusing 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 3.1, the time estimates in Table 3.1 can be made for the *lowest* level tasks.

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 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.

3.3.3 Step 3: Milestone identification

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

Table 3.1 *Time estimates for example project*

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 test data	1 week
Train ANN	1 week
Use stock market models	2 weeks
Investigate statistical tests	2 weeks
Analyse and evaluate	4 weeks
Complete report	8 weeks
Total effort	40 weeks (approximately 10 months)

are leading you towards the ultimate goal of your project you can use them as intermediate goals at which to aim. Figure 3.2 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 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. One milestone you will always identify is the project's completion.

For simplicity only two milestones will be identified in the example project, 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 completing 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.

3.3.4 **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. We will look at the simplest form of these diagrams, in which activities are represented by rectangles or *nodes* (known as activity-on-the-node).

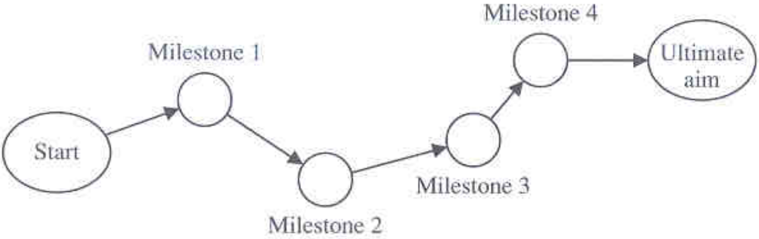

Figure 3.2 *Milestones leading to the project's ultimate aim*

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 3.3, 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 3.4. In this example the completion of the project's report has been identified as an activity that is 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 additions to Figure 3.4 that are not shown in Figure 3.3 and which have yet to be explained. The first point to note is that the milestones identified earlier have been included as ovals 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 figures 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 3.1. It is up to you what time 'granularity' you use for your project (hours, days, weeks, months or even years) but in a 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 so on can often impede your progress, and these events should be considered when forming project plans.

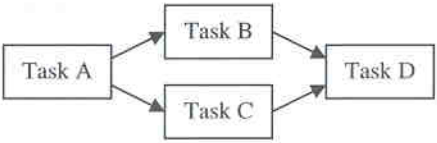

Figure 3.3 *An example of a simple activity-on-the-node diagram*

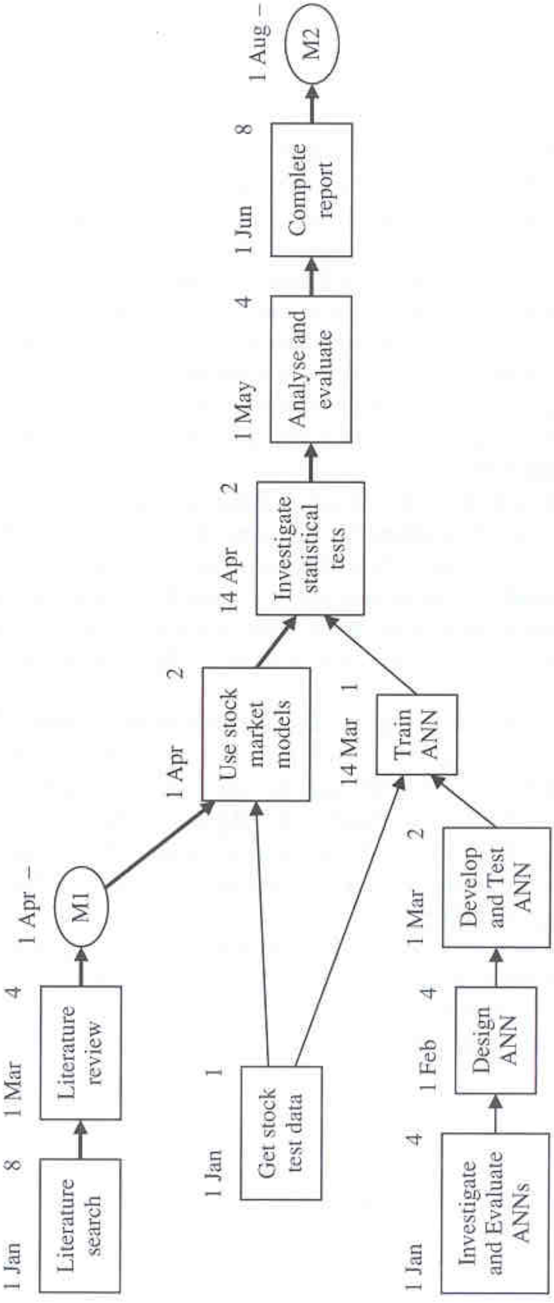

Figure 3.4 An example of an activity network

The date, which is noted at the top left-hand corner of each activity, represents the time at which that activity will 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 test 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.

To calculate the start times of subsequent activities it is necessary to look at the tasks leading to them. For example, in this simple case, the *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 occur on 1 April. Notice also how M1 has no duration assigned to it because it does not represent any work as such but 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* preceding tasks have completed. For example, in Figure 3.4 notice that *Train ANN* starts on 14 March – this is when you would expect *Develop and test ANN* to complete, *not* when *Get stock test data* has completed. Remember that subsequent activities can only begin when *all* preceding activities leading to 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 addition 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 3.4 by the bolder arrowed lines. It shows 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 now 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 affect the project overall.

To identify the critical path you work backwards through the network diagram. 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* 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 until you either reach the project's start or an activity that has two or more activities leading into it. In the latter case *Investigate statistical tests* is the first activity in this situation. However, just as before, you look back through the network to see which activity is forcing *Investigate 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 following 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 two 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 survey 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 yourself 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, activities exist that are ongoing throughout the lifetime of your project; for example, the literature survey and report writing activities. However, emphasis on these 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 4. A particular example of loops occurring within software development projects is when an evolutionary delivery approach is being used.

These situations cannot be planned explicitly using ordinary activity network diagrams, and although there are some 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.

3.3.5 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, which show the relationships between tasks, Gantt charts show explicitly the durations of activities and identify instances when tasks are performed simultaneously.

Just like activity networks, Gantt charts represent a project's activities as nodes. In this case, however, the length of each node is used to signify the duration of each activity. For example, in Figure 3.5 a Gantt chart is presented for the example 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 each 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.

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 that

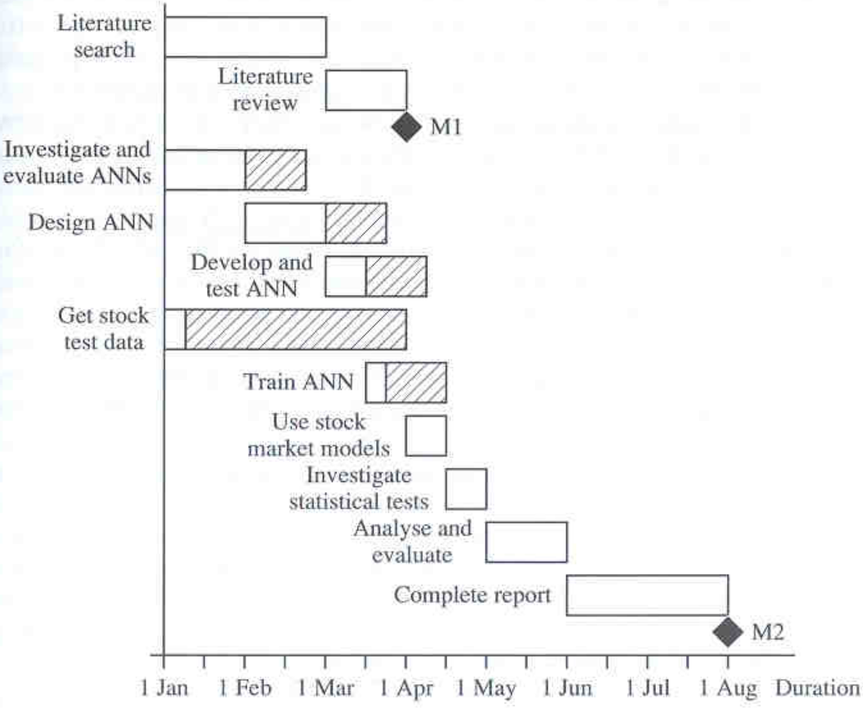

Figure 3.5 *An example of a Gantt chart*

activities do not start in the Gantt chart before all their preceding tasks have completed. Some Gantt charts allow you to include the arrowed lines linking activities like those in the activity network. However, trying to include all this information on one diagram does lead to the diagram becoming very messy and difficult to follow.

The Gantt chart also differs from the activity network in that milestones are now represented by diamonds. In addition, 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 means that activities that do not 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 time.

To identify slack time you need to focus on activities that do not lie on the critical path. You then work your way backwards through your project until you meet one of these activities. In the example, looking at the activity network in Figure 3.4, the first activity encountered, working back through the network, that does not lie on the critical path is *Train ANN*. *Train ANN* leads into *Investigate statistical tests*, which is on the critical path. As long as *Train ANN* is not delayed for so long that it starts to delay *Investigate statistical tests* the project will not be affected. Thus, *Train ANN* could be delayed so that it finishes no later than the start of *Investigate statistical tests*. This delay represents the slack time of *Train ANN* and is shown as a shaded area in Figure 3.5.

Because *Investigate and evaluate ANNs*, *Design ANNs* and *Develop and test ANN* all lead into *Train ANN*, these activities may also be delayed by the same duration as *Train ANN* without affecting the project. Consequently, these activities have the same float applied to them as *Train ANN* (i.e. three weeks). *Get stock test 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* can begin. Thus, *Get stock test data* can be delayed up until 1 April but notice that it would delay *Train ANN* by two weeks if it were. This is acceptable because *Train ANN* is not on the critical path and thus delaying it by two weeks will not impact on the project. 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 ANNs* and *Get stock test data* all at the same time. For group projects this is not a problem as these tasks can be assigned between 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 test 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 test data* will have to be done and it will inevitably clash with some other work then. The problem lies in the fact that ten months' worth of work is being attempted within seven months with only one person available. This is impossible, unless you are able to do more than one activity at a time. If you cannot, then 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 balance out people's time on projects in order to achieve a satisfactory allocation of work over a project's life span.

In this case, a popular project management package called Microsoft Project has been used. Figure 3.6 shows a print-out of a Gantt chart from this package for the example project. Notice how similar this is to the representation shown in Figure 3.5. Microsoft Project was then used to replan 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 3.7. 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 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.

3.3.6 Step 6: Replanning

Now that you have completed all your plans you may realise that you are trying to do too much in the time available. Replanning 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 replanning and rescheduling at minute levels of detail rather than getting on with the real work of your project.

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 to concentrate on other activities.

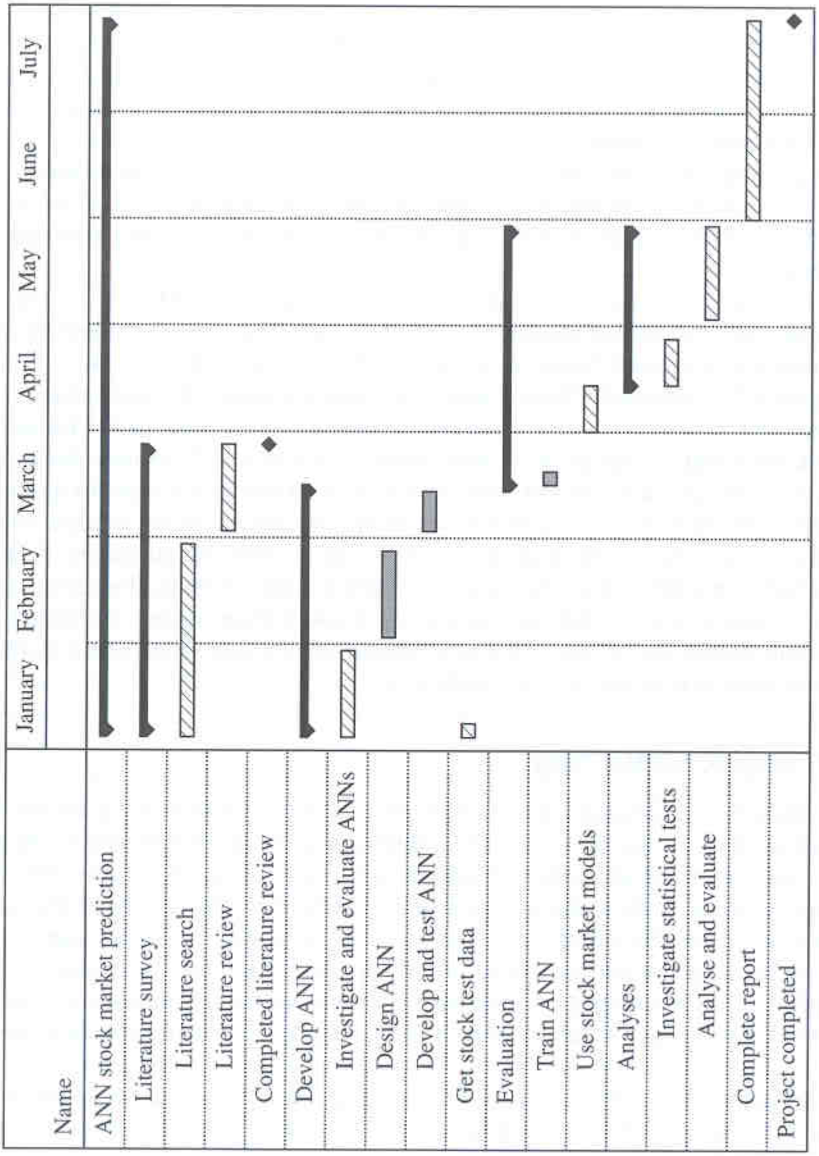

Figure 3.6 *Microsoft Project's Gantt chart of example project*

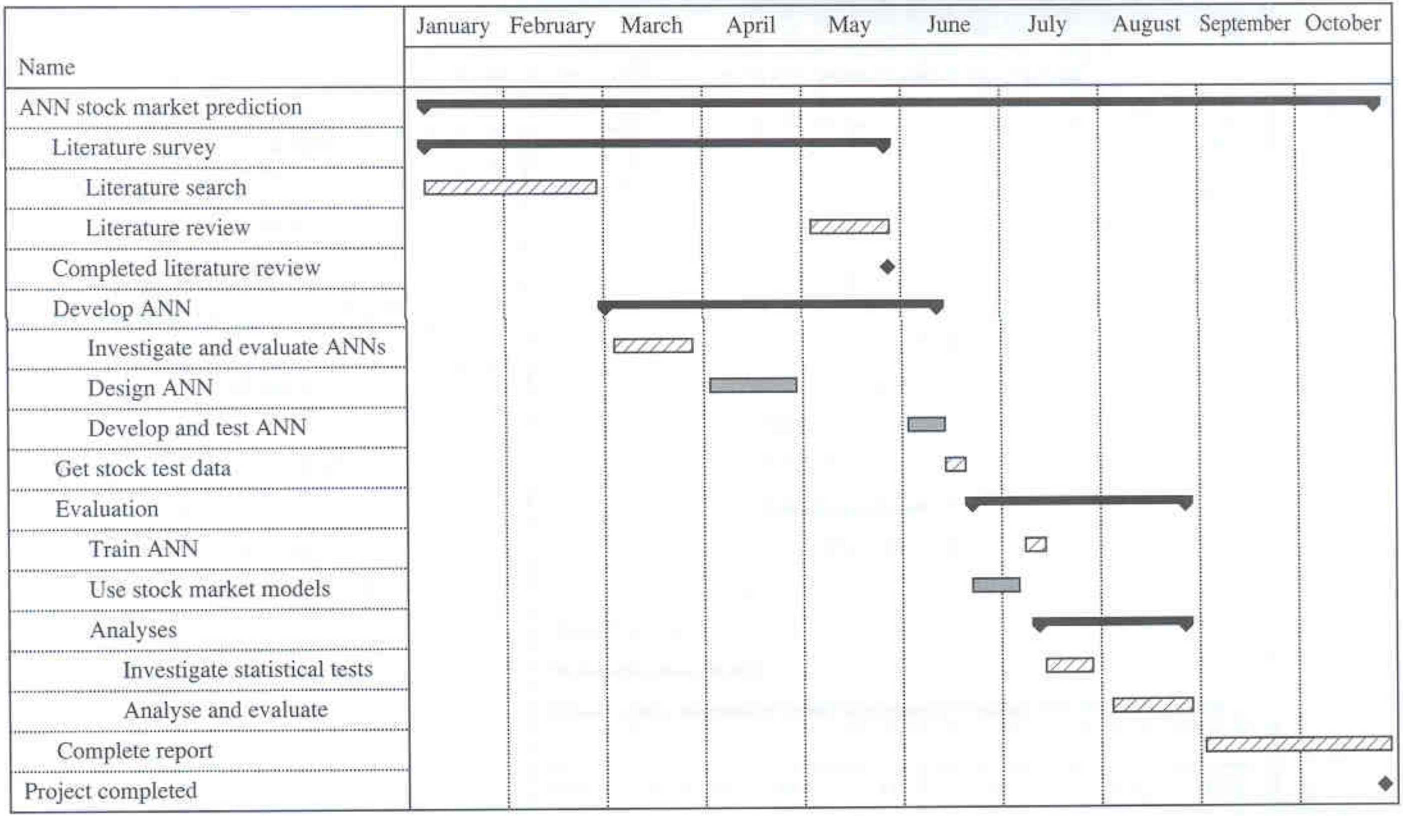

Figure 3.7 *Scheduled Gantt chart of example project*

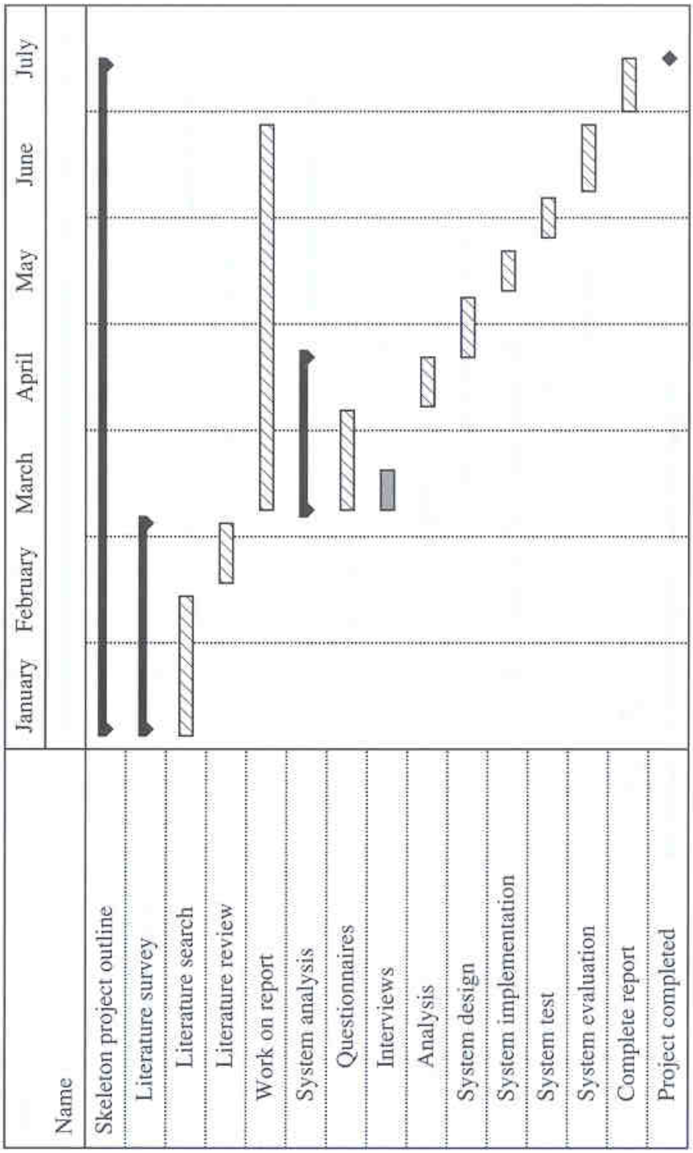

Figure 3.8 *Example rolling wave skeleton plan for a system development project*

3.3.7 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 project plan at the project's inception but a 'skeleton' plan which 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 following stage of your project as you complete 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 and so on – whatever these turn out to be. Figure 3.8 provides an example of a typical rolling wave, skeleton plan – in this case a system 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.

3.4 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 using *Gantt charts* so that you are not trying to do more than you can physically achieve, and replanning 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 3.4 and 3.5, 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.

3.5 **Further reading**

Burton, C. and Michael, N. (1992) *A Practical Guide to Project Management*, Kogan Page, London.

Weiss, J.W. and Wysocki, R.K. (1992) *5-Phase Project Management, A Practical Planning and Implementation Guide*, Addison-Wesley, Reading, Massachusetts.

3.6 **Exercises**

1. Try to identify objectives for the example projects mentioned in Section 3.2.1.

2. Identify aims and objectives for your own computing project.

3. Follow the six steps of planning to complete your project's plan.