12 SOFTWARE CONTRACTS AND LICENCES

After studying this chapter, you should:

- *understand the purpose of contracts in the computer industry and the different types of contractual arrangement that are commonly used;*
- *be familiar with the main issues that such contracts address;*
- *understand the different types of liability for defective software that can arise and the factors that affect these.*

12.1 WHAT IS A CONTRACT?

A contract is simply an agreement between two or more persons (the parties to the contract) that can be enforced in a court of law. The parties involved may be legal persons or natural persons. There is no specific form for a contract; in particular, in England and Wales a contract need not be written down. The following are essential:

- All the parties must intend to make a contract.
- All the parties must be competent to make a contract, that is, they must be old enough and of sufficiently sound mind to understand what they are doing.
- There must be a 'consideration', that is, each party must be receiving something and providing something.

Contract law is largely based on common law. It has a long history and is well adapted to handling the disputes that arise in fulfilling commercial agreements.

The existing contract law showed itself perfectly adequate to handle contracts for the supply of computers, software and associated services. However, the coming of the internet and ecommerce has created a need for new provisions to deal with such matters as electronic signatures and which country's laws should govern transactions made over the internet when the parties to the transaction are in different countries.

In this chapter we shall not be very much concerned with the law relating to contracts. We shall be much more concerned with what should go into contracts for providing software and services in different circumstances and with issues of liability when software fails.

12.2 LICENCE AGREEMENTS

It is very unusual for customer to 'buy' software in the strict sense of the word 'buy'. As was said in the last chapter, what is usually bought is a copy of the software together with a licence to use it in certain ways. There are two main types of licence, although there are innumerable variants of each:

- **Retail** software is software aimed at a mass market and selling for a one-off fee of a few tens or hundreds of pounds per copy. Large organisations can usually negotiate bulk licences for multiple copies at a considerable discount and in some cases site licences may be offered covering all computers on a given site. The licence is usually granted *in perpetuity* (that is, for ever).The licence offers no provision for updates or corrections to be supplied nor for any support to the user, except possibly a help line. If a new version is required, in order to interface with new versions of other software perhaps or to run on an upgraded platform, the user must buy it from scratch.
- **Corporate** software is software aimed at large organisations. It is often quite specialised and may be aimed at a market with fewer than 100 customers, although some packages, such as those for human resources management and customer relationship management, may have many thousands of customers. There is typically a substantial up-front licence fee of several tens or hundreds of thousands of pounds together with an annual charge of some 20 per cent of the initial fee, which may be described as a maintenance charge. The fees may well depend on usage: the annual fee for a payroll package, for example, will usually depend on the number of employees while that for a computer-aided design package will depend on the maximum number of simultaneous users. In return for these not inconsiderable charges, the customer gets help with the initial installation and configuration, regular upgrades (including those necessary to comply with legislative changes) and an efficient and responsive help line. There may well be a user group. Training and consultancy are available but the fees are high; suppliers of this sort of software rely on such incidental income to maintain their profitability.

Software licences are usually long and complicated. Many of their provisions are intended to protect the supplier in situations that are fairly unlikely to occur. Some of them may be unenforceable in the UK because of provisions in the Unfair Contract Terms Act 1977, which is discussed in the section below on contracts for bespoke software.

Open-source licences for software allow anyone to use the source code and other design material, either in its original form or after modification, subject to certain conditions. Software covered by an open-source licence is usually, but not invariably, available free of charge. Typical conditions to be found in open-source licences include requirements to keep the names of the authors and a statement of copyright in the code and only to re-distribute the software under identical licence conditions; the latter requirement prevents code that was acquired free from being redistributed for profit. The Open Source Initiative, which was launched in 1998, exists to promote software licensed in this way. Closely related to open-source software is 'free software', a concept promoted by the Free Software Foundation, established by Richard Stallman in 1985 to support the GNU project. The OSI and the Free Software Foundation have been successful in getting free or open-source software accepted and widely used. The underlying ideological

principles that motivate, and separate, the two movements have not, however, achieved the same degree of acceptance.

A completely different type of licence is a marketing licence. Companies that produce software packages frequently do not have the resources or expertise to market the software outside their own countries and they therefore license agents to sell it for them in other countries. Computer games are usually produced by fairly small companies that have no capacity to market their games. Marketing licences are normally granted for a fixed period; they may be restricted to a specified geographical area or to specific types of platform. Thus a UK company that produces an HR (human resources) package may grant a licence to an Australian company to market the package in Australia and New Zealand for a period of two years. The marketing company agrees to pay the supplier a fixed percentage of the income it receives from each sale of the package. At the end of the two-year period, the supplier may choose to renew the licence, if sales have been satisfactory, or it may look for a new agent in the hope of increasing sales. Marketing licences may be exclusive, that is, they stipulate that no other company will be allowed to market the software in the prescribed geographical area. Exclusive licences generally encourage the marketing company to market the product more energetically than they would do if other companies were competing to sell the same product in the same area.

12.3 OUTSOURCING

IT outsourcing contracts are inherently complex and depend very much on individual circumstances. It is not appropriate to go into detail here about such contracts but the following is a list of just some of the points that need to be addressed:

- How is performance to be monitored and managed?
- What happens if performance is unsatisfactory:
- Which assets are being transferred? It is often the case that the customer outsourcing its IT operations already has equipment and software and that it wishes the supplier to take this over. It is important to agree on precisely what is to be transferred.
- Very often IT staff will be transferred to the supplier. How this is to be done is covered by the TUPE regulations, discussed in Chapter 9 in the section on takeovers and outsourcing.
- Audit rights. The customer needs to ensure that both its external auditors and its internal auditors have adequate access rights to be able to fulfil their duties.
- Contingency planning and disaster recovery.
- Ownership of the intellectual property rights in software developed during the contract.
- Duration of the agreement and termination provisions.

The first two items above are key element in IT outsourcing and are often treated as a separate agreement known as a **service level agreement**.

Related to outsourcing is the provision of application services over the internet using a client-server system with a remote server. This is an instance of what is commonly known as 'cloud computing'. It is not a new idea. Payroll services have been provided in this way since the late 1960s. Because of the provisions of the Data Protection Act 1998, it is important to ensure that all the servers potentially involved meet the EU data protection requirements described in Chapter 13, in particular the eighth data protection principle.

12.4 CONTRACTS FOR BESPOKE SOFTWARE

Although private individuals normally buy software by purchasing a licence for a standard package and then, possibly, configuring it to meet their needs, commercial organisations quite commonly want to purchase a system configured specifically to meet its needs. Such systems are known as **tailor-made** or **bespoke** systems. A bespoke system may consist of a single PC equipped with a word processor, a spreadsheet, and a set of macros adapted to the customer's needs or it may consist of several thousand PCs spread across 50 offices in different parts of the world, connected by a wide-area network, with large database servers and a million lines of specially written software.

Contracts for bespoke software are often long and complicated. We shall describe here only the main points of such a contract, as they affect the information systems staff likely to be involved, whether on the client side or the supplier side.

12.4.1 What is to be produced

It is clearly necessary that the contract states what is to be produced. Information systems engineers will be familiar with the problems of producing requirements specifications. A specification sets out the detailed requirements of the client. Ideally, the specification should be complete, consistent and accurate and set out all that the client wants to be done in the performance of the contract. Unfortunately, we know that it is very difficult to achieve this ideal standard and, even if we succeed, the requirements of the client may evolve as the contract proceeds, and sometimes the changes may be substantial. How are these changes to be accommodated by a contract which, in a sense, freezes the requirements of the parties to those at one particular time by incorporating the original specification into the contract? The answer is that the contract should provide a procedure for making variations to the specification or job description, then follow this through by providing a method of calculating payment for work done to facilitate the changes, and also perhaps provide for a variation of the level of anticipated performance, and maybe also vary the method of acceptance testing. In other words, the contract should anticipate events and provide an agreed formula for modification.

Producing software for a client is not, usually, a matter of simply handing over the text of a program which does what is required. It is important, therefore, that the contract states what precisely is to be provided. The following is a non-exhaustive list of possibilities:

- source code;
- command files for building the executable code from the source and for installing it;

- documentation of the design and of the code;
- reference manuals, training manuals and operations manuals;
- software tools to help maintain the code;
- user training;
- training for the client's maintenance staff;
- test data and test results.

Alongside the issues of what is to be produced, there is the question of the standards to be used. The supplier is likely to have company standards, methods of working, quality assurance procedures, etc. and will normally prefer to use these. More sophisticated clients will have their own procedures and may require that these be adhered to. In some cases, the supplier may be required to allow the client to apply quality control procedures to the project. The contract must specify which is to apply.

12.4.2 Intellectual property rights

When an organisation commissions bespoke (that is, tailor-made) software from another company, it is important that both sides think carefully about the licensing and ownership of the copyright in the software produced. There was a time when the customer would expect to take ownership of the copyright in all the software supplied. This is no longer realistic. Parts of the software supplied are likely to be proprietary products that the supplier has developed to enable him to construct such systems quickly and efficiently; they are part of his fixed assets and it would be absurd to transfer ownership of them to a customer. Other parts of the software may be products acquired from other sources (possibly open source software) and the supplier will not have the right to assign copyright in them. The customer's needs can almost certainly be satisfied by a suitable licence. However, the customer must take into account the long-term need for maintenance and ensure that he has the right, for example, to give another contractor access to the source code and documentation for maintenance purposes.

It is important that the contract should also state just what legal rights are being passed by the software house to the client under the contract. Ownership in physical items such as books, documents or discs will usually pass from the software house to the client, but intellectual property rights, discussed in the previous chapter, present more problems. It is important for the contract to state precisely who is to own these rights.

A related topic is confidentiality. It is almost inevitable that, when a major bespoke software system is being developed, the two parties will acquire confidential information about each other. The commissioning client may well have to pass confidential information about its business operations to the software house. On the other side of the coin, the software house may not want the client to divulge to others details of the program content or other information gleaned about its operations by the client. It is usual in these circumstances for each party to promise to maintain the confidentiality of the other's secrets, and for express terms to that effect to be included in the contract.

12.4.3 Management of the contract

It is important that each party should understand the framework for making decisions relating to the contract and the mechanism by which these decisions will be agreed and recorded. Failure in this respect is ultimately responsible for most of the contractual wrangles that affect software contracts. Each party needs to know who, of the other party's staff, has day to day responsibility for the work and what the limits of that person's authority are. Each party should therefore be required party to nominate, in writing, a Project Manager. The Project Managers must have at least the authority necessary to fulfil the obligations which the contract places on them. It is particularly important that the limits of their financial authority are explicitly stated, i.e. the extent to which they can authorise changes to the cost of the contract.

Regular progress meetings are essential to the successful completion of a fixed price contract and the contract should require that they are held. The minutes of progress meetings, duly approved and signed, should have contractual significance in that they constitute evidence that milestones have been reached (so that stage payments become due) and that delay payments have been agreed.

In almost all cases where work is being carried out for a specific client, the client will have to fulfil certain obligations if the contract is to be completed successfully. The following is a (non-exhaustive) list of possibilities:

- provide documentation on aspects of the client's activities or the environment in which the system will run;
- provide access to appropriate members of staff;
- provide machine facilities for development and testing;
- provide accommodation, telephone and secretarial facilities for the company's staff when working on the client's premises;
- provide data communications facilities to the site.

A particular difficulty may arise if the client does not provide critical information on time, since this can lead to delays that cost the contractor money. The contract should provide a clear mechanism for calculating payments due for such delays. A similar mechanism will be required for calculating extra payments due when the client requests changes at a late stage – a very common occurrence.

Delay payments and payments for variations to the original requirements are, perhaps, the commonest cause of contractual disputes, not only in software engineering but in most other contracting industries - the construction industry is a notorious example. One reason for this is that competitive bidding for fixed price contracts often means that the profit margin on the original contract is very low so that companies seek to make their profit on these additional payments.

12.4.4 Acceptance procedure

Acceptance procedures are a critical part of any fixed price contract for they provide the criteria by which successful completion of the contract is judged. The essence of the

acceptance procedure is that the client should provide a fixed set of acceptance tests and expected results and that successful performance of these tests shall constitute acceptance of the system. The tests must be provided at or before the start of the acceptance procedure; within reason, there may be as many tests as the client wishes but extra tests cannot be added once the test set has been handed over. The purpose of this restriction is to ensure that the acceptance procedure can be completed in reasonable time.

Other points to be addressed under this heading include who shall be present when the tests are carried out and what happens if the tests are not completed successfully.

12.4.5 Termination of the contract

There are many reasons why it may become necessary to terminate a contract before it has been completed. It is not uncommon, for example, for the client to be taken over by another company which already has a system of the type being developed, or for a change in policy on the part of the client to mean that the system is no longer relevant to its needs. It is essential, therefore, that the contract make provision for terminating the work in an amicable manner. This usually means that the supplier is to be paid for all the work carried out up to the point where the contract is terminated, together with some compensation for the time needed to redeploy staff on other revenue earning work. The question of ownership of the work so far carried out must also be addressed.

12.5 CONTRACTS FOR CONSULTANCY AND CONTRACT HIRE

Contract hire is an arrangement in which the supplier agrees to supply the customer with the services of a certain number of staff at agreed daily or hourly charge rates. The customer takes responsibility for managing the staff concerned. Either party can terminate the arrangement at fairly short notice, typically one week, either in respect of a particular person or as a whole. The supplier's responsibility is limited to providing suitably competent people and replacing them if they become unavailable or are adjudged unsuitable by the client.

Because the supplier's involvement and responsibility are so much less than in a fixed price contract, the contract for such an arrangement is usually fairly simple. Payment is on the basis of a fixed rate for each man day worked; the rate depends on the experience and qualifications of the staff. Issues such as delay payments, acceptance tests and many others simply do not arise; however, as mentioned earlier, ownership of intellectual property rights generated in the course of the work must be addressed.

Contract hire is sometimes referred to disparagingly as 'body shopping'. Closely related are the freelance agreements under which individuals sell their own services to clients on a basis similar to contract hire.

Consultancy is essentially an up-market version of contract hire. Consultants are experts who are called in by an organisation to assess some aspect of its operations or its strategy and to make proposals for improvements. This means that the end product of a consultancy project is usually a report or other document.

Consultancy projects are usually undertaken for a fixed price but the form of contract is very much simpler than the fixed price contracts so far described. There are two reasons for this. First, the sums of money are comparatively small and neither side stands to lose a great deal. Second, although it is possible to demonstrate beyond doubt that a piece of software does not work correctly and thus that the supplier has failed to fulfil the contract, it is not usually possible to demonstrate unequivocally that a report fails to fulfil a contract. The client has to rely on the desire of the supplier to maintain a professional reputation and in practice this usually proves sufficient to ensure that the work is of an acceptable standard.

There are four important aspects of a consultancy contract:

- Confidentiality. Consultants are often in a position to learn a lot about the companies for which they carry out assignments and may well be in a position to misuse this information for their own profit.
- Terms of reference. It is important that the contract refers explicitly to the terms of reference of the consultancy team and, in practice, these are perhaps the most common source of disagreements in consultancy projects. As a result of their initial investigations, the consultants may discover that they need to consider matters which were outside their original terms of reference but the client may be unwilling to let this happen, for any one of a number of possible reasons.
- Liability. Most consultants will wish to limit their liability for any loss that the customer suffers as a result of following their advice. Customers may not be happy to accept this and, in some cases, may insist on verifying that the consultant has adequate professional liability insurance, that is, insurance to cover damages that might be imposed by a court for any professional failure.
- Who has control over the final version of the report. It is common practice for the contract to require that a draft version of the final report be presented to the client. The client is given a fixed period to review the report and ask for changes. The revised version that is then submitted by the consultant should be the final version.

A time and materials contract (often referred to as a 'cost plus' contract) is somewhere between a contract hire agreement and a fixed price contract. The supplier agrees to undertake the development of the software in much the same way as in a fixed price contract but payment is made on the basis of the costs incurred, with labour charged in the same way as for contract hire. The supplier is not committed to completing the work for a fixed price, although a maximum payment may be fixed beyond which the project may be reviewed. Many of the complications of fixed price contracts still occur with time and materials contracts – ownership of rights, facilities to be provided by the client, progress monitoring arrangements, for instance – but others, such as delay payments and acceptance testing do not; this is not to say that no acceptance testing is done, only that it has no contractual significance since nothing contractual depends on its outcome.

It may be wondered why any client should prefer a time and materials contract to a fixed price contract – surely it is better to have a contract which guarantees performance for a fixed price rather than one in which the price is indeterminate and there is no guarantee of completion? In the first place, it often happens that the work to be carried out is not sufficiently well specified for any supplier to be prepared to offer a fixed price; part of the supplier's task will be to discover what is required and to specify it in detail. Secondly, a supplier always loads a fixed price contract with a contingency allowance, to allow for the risk that unexpected factors will cause the project to require more resources than originally estimated. If all goes well, the supplier makes an extra profit; this is the reward for risk taken. By accepting a time and materials contract, this risk and the possibility of extra profit (in the form of a lower cost) are effectively transferred to the client, who also avoids the dangers of having to pay excessive sums to have minor changes incorporated into the specification. All this having been said, it remains the case that there has been a strong movement away from time and materials towards fixed price, noticeably in the defence field.

12.6 LIABILITY FOR DEFECTIVE SOFTWARE

Suppliers of software and hardware are very reluctant to give a contractual commitment that it is fit for any purpose whatsoever. Standard terms and conditions will invariably contain a clause that tries to limit the supplier's liability if it turns out that the software or hardware is defective. The law, however, limits how far such clauses can be effective.

Most contracts will limit the extent of any liability either to the purchase price of the product or to some fixed maximum figure. This means that, if the product completely fails to work, the supplier agrees to refund the purchase price or possibly a bit more if some other maximum is specified.

The Unfair Contract Terms Act 1977 restricts the extent to which clauses in standard terms and conditions limiting liability can be effective. In particular, it is not possible to limit the damages payable if a defect in the product causes death or personal injury. This applies as much to software as it does to motor vehicles, say. Thus if a company supplies software to control a light railway link and a defect in the software leads to an accident in which people are killed or injured, then any clause in the contract for the supply of that software that claims to restrict liability will not be enforceable in respect of the claims for damages for the deaths and injuries.

This restriction is an important one for companies that produce safety critical software and it also led at least one hardware supplier to refuse to sell its products to the nuclear industry. However, although this is very relevant when buying, say, a motor car, it is not very relevant to most individuals or companies when dealing with software, because the software that they use or develop is very unlikely to cause death or personal industry. They are much more likely to be concerned about software that doesn't do what it was claimed to do or that has too many bugs to be usable.

At this point we need to distinguish between consumer sales and non-consumer sales. For a sale to be treated as a consumer sale, the buyer must be a private person; the

buyer must buy from a seller who is acting in the course of a business; and the goods must be of a type ordinarily supplied for private use or consumption. In a consumer sale, the requirements of the Sale of Goods Act 1979 and the Supply of Goods and Services Act 1982 cannot be excluded. The most important requirement of the Sale of Goods Act in the context of software is that goods sold must be fit for the purpose for which such goods are commonly supplied. This means, for example, that if you buy a printer for use on your computer at home, it must be capable of printing reliably and clearly, at a usable speed.

There is, unfortunately, a problem about software. Because software is intangible, it has never been satisfactorily decided whether or not it comes under the definition of 'goods' and hence it is not clear whether the Sale of Goods Act 1979 applies to the sale of software. It was generally thought that it applied to the sale of retail software or software sold under 'shrink-wrapped' licences, that is, licences that are deemed to come into operation when the package containing the disk on which the software is supplied is opened. However, with the coming of the internet, software is typically downloaded and users are asked to tick a box saying that they accept the licence conditions. It is not at all clear that the Sale of Goods Act applies in such cases. Instead, they would come under the Supply of Goods and Services Act 1982. This only requires that 'reasonable skill and care' has been used. This is a very difficult test to apply and, in practice, provides little protection.

The Unfair Contract Terms Act 1977 again comes to the rescue. Whether or not the sale is a consumer sale, it allows liability to be limited or excluded only to the extent that it is reasonable to do so.

A COURT CASE

The view that a court will take depends very much on the circumstances of a particular case but a particularly illuminating example is the 1996 case of *St Albans City and District Council v International Computers Ltd (ICL)*. The facts were that the council had ordered a computer system from ICL to enable it to compute the Community Charge (a system of local taxation that is no longer in use) for the forthcoming year. ICL insisted on using its standard terms and conditions which stated that its liability 'will not exceed the price or charge payable for the item of Equipment, Program or Service in respect of which liability arises or £100,000 (whichever is the lesser). . .'. Errors in the software and incorrect advice from ICL's project manager meant that the population of the area was overestimated, so the residents were undercharged and the council lost £1.3 million.

The judge at the initial hearing found that the software was not fit for the purpose for which it was provided and that ICL's project manager had been negligent. ICL was therefore in breach of contract. The clause limiting liability had to be measured against the requirement of reasonableness. The judge noted that ICL was a substantial organisation with world-wide product liability insurance of £50 million; that all potential suppliers of the system dealt on similar standard terms; that the council was under pressure to install the system before the Community Charge

was introduced; that although the council was a business and not a consumer, that it did not usually operate in the same commercial field as a normal business and that it would be impractical for it to insure against commercial risks. On balance, the judge found that the clause limiting liability to £100,000 was not reasonable and was therefore ineffective. ICL appealed, but the Court of Appeal confirmed the judgement. However, the value of this case as a precedent will be limited, for each case turns on its facts.

12.7 HEALTH AND SAFETY

The Health and Safety at Work Act (1974) completely changed the approach to safety in Britain. European thinking was also affected by the Act and there has been fruitful exchange of ideas.

It is not appropriate to go into detail about the provisions of the Act here but it is important to understand the implications that the Act has for information systems engineers. The Act states 'It shall be the duty of every employer to ensure, so far as is reasonably practicable, the health, safety and welfare at work of all his employees'. It breaks this duty up into a number of more specific duties; the ones that are of particular concern to software engineers are as follows:

- provision and maintenance of safe plant;
- provision and maintenance of safe systems of work;
- provision of such information, instruction, training, and supervision as necessary;
- ensuring the workplace is maintained in a safe condition; and
- provision and maintenance of a safe working environment and adequate welfare arrangements.

The Act also requires employers to ensure that their activities do not expose the general public to risks to their health and safety. Manufacturers of equipment to be used at work also have a responsibility to ensure that it is safe.

Failure to comply with the Health and Safety at Work Act is a criminal offence and, in serious cases, can lead to criminal proceedings being taken against individuals.

Trains, ships and aeroplanes are all places where people work and where the general public are present. The safety obligations listed above therefore apply to them. Furthermore, they all involve equipment that is controlled by software and accidents may occur as a result of defects in that software.

Modern manufacturing plant is usually software-controlled and can be dangerous; robots in particular can be dangerous for people working with them. Modern chemical plants, oil refineries and power stations, especially nuclear ones, are all software-controlled and software failures can result in accidents that affect not only the workforce but also the general public.

This book is not the appropriate place to discuss how software is or should be written in order to meet the high levels of reliability required for safety-related applications. You should understand clearly, however, that there is an extensive literature on the subject and that there are national and international standards relating to it. An organisation that undertakes to develop safety-related software without employing staff who are familiar with the appropriate development techniques is likely to be in breach of the Health and Safety at Work Act. The following clauses in the BCS Code of Conduct (and similar clauses in other codes) are of particular relevance in this area:

You shall:

1a have due regard for public health, privacy, security and wellbeing of others and the environment.

2a only undertake to do work or provide a service that is within your professional competence.

2b NOT claim any level of competence that you do not possess.

2c develop your professional knowledge, skills and competence on a continuing basis, maintaining awareness of technological developments, procedures, and standards that are relevant to your field.

2d ensure that you have the knowledge and understanding of Legislation and that you comply with such Legislation, in carrying out your professional responsibilities.

FURTHER READING

The Open Source Initiative:
http://opensource.org

The Free Software Foundation:
www.fsf.org

The Gnu project:
www.gnu.org/

The following book, from the BCS/Springer Practitioner series, covers IT outsourcing in depth and is written specifically for information systems professionals. It contains a chapter dedicated to contractual matters:

Sparrow, E. (2003) *Successful IT outsourcing*. Springer Verlag, London.

Fuller coverage of the material in this chapter, including health and safety issues, can be found in Chapters 5, 9 and 10 of the following book:

Bott, M.F., Coleman, J.A., Eaton, J., and Rowland, D. (2000) *Professional issues in software engineering,* 3rd ed. Taylor and Francis, London.