CCS2710 Professional Issues in Information Technology

Contract Law

Iraklis Paraskakis

UNIVERSITY of York
Europe Campus
CITY College

1. Introduction to Software Contracts

- *A software contract* is a contract for the supply of software (either bespoke or off-the-shelf).

- Many such contracts are not sale contracts, but are *licence agreements* that allow a customer to use software in return for a *licence f*ee.

- Many contractual problems occur because of the unique nature of computer software.

- **Q. What is unique about computer software?**

  Software is traded as a commodity but it is intangible and it is difficult to measure its ‘quality’.

1. Introduction to Software Contracts

- When a contract is entered into for the licensing or custom development of software:
  - Both parties should know precisely what is expected in terms of performance and the standards required.
  - The contract should provide a fair means of identifying responsibilities and resolving disputes.
  - A comprehensive specification must be drawn up.
  - Liability for any injury to persons or property as a result of errors in the software should be considered.

1.1. Example of contractual difficulties

- A company wishes to computerise to increase efficiency. The manager suggests that some ready-made software packages are evaluated, since these will be less expensive than writing custom software from scratch.
  - *Who will evaluate the software*?
  - *Will the priorities of these people will be the same*?
  - *What is the result*?

Who will evaluate the software?

- Computer professionals,
  - who understand software but do not have a deep knowledge of the particular application of the proposed software.
- Managers and directors,
  - who are unfamiliar with software and have some knowledge of the application.
- Potential users of the system.

Q. Will the priorities of these people will be the same?

- Professionals
  - want a system they can integrate easily.
- Managers
  - want something cheap.
- Users
  - want ease of use, flexibility.

Q. What is the result?

- A lack of communication between these people,
  - legal advisors
  - and the suppliers of the software.
- A product may be delivered that is
  - cumbersome,
  - not what the users wanted,
  - expensive
  - and runs too slowly to be of any practical use.

- The software company claims that it was not given sufficient guidance, and suggests that the problems will be overcome if the client buys faster hardware.

- The software contract is examined and the client realises that it does not provide any mechanism for resolving the dispute. The client refuses to pay and there is a legal struggle, both sides claiming that the other has failed to honour the terms of the contract.

1.2. Fundamentals of software contracts

- Software is usually acquired by a *licen*ce, which is granted by the software publisher to the person acquiring the software.

- The licence may be for a fixed period of time.

- **Q. Why do you need a licence to use software? What does the licence do?**

  – Allows a user to run the software, otherwise breaking the CDPA - effectively granting exemption from copyright law. Allows the seller to sell to more than one customer. Ownership of copyright may be transferred in some cases - assignment of copyright.

1.2. Fundamentals of software contracts (cont'd)

- The nature of software and the fact that it is normally acquired via a licence has two legal implications:
  - The *Sale and Supply of Goods Act 1994* (which amended the earlier *Sale of Goods Act 1979*) does not apply to computer software.
  - Computer software is exempted from much of the *Supply of Goods and Services Act 1982*.
- These acts 'imply' terms into a contract for sale or supply of goods or services. The implied terms, which cannot be modified or excluded, give rights to the consumer.

1.2.1. Sale and Supply of Goods Act (1994) and software

- The SSGA implies terms into contracts of sale such as
  - The goods must match their description in the contract.
  - The goods must be of 'satisfactory' quality.
  - The goods must be fit for their intended purpose.
  - The seller has the right to sell the goods. But 'goods' are defined by the earlier Sale of Goods Act (1979) as *'all personal chattels other than things in action and mone*y'.
  - Copyright is a 'thing in action' (other examples are shares or money orders). Software is therefore excluded from the definition of 'goods' under the SSGA.
  - However, computer hardware or computer media (magnetic/optical disks) are treated as goods.
  - **Q. In fact, the SSGA does not apply to most software contracts since a licence is not a contract for sale of goods. Why not?**
    - No "property in possession" changes hands.
- So, the terms contained in the SSGA which are implied into a contract for the sale of goods will not apply to a software contract.

1.2.1. The Supply of Goods and Services Act 1982 and computer software

- The SGSA implies terms into contracts in which the ownership of goods changes hands, and in contracts for the hire of goods and contracts for services.
- For reasons given (see previous slides), software is excluded from the 'supply of goods' part of the SGSA.
- The SGSA is relevant if a programmer is contracted to write a computer program, since this is a 'service'.
- The supply of an expert system could also be deemed to be providing a service.
- The terms implied by the SGSA are as follows:
  - The supplier, if acting in the course of business, must carry out the service with reasonable care and skill.
  - In the absence of an agreed time for performance, the supplier will carry out the service in a reasonable time.
  - Unless the contract fixes the payment, the supplier will be paid a reasonable amount.

Q. What constitutes ‘reasonable care and skill’ in software development?

- The short answer is – Don’t know!
- Almost impossible to measure consistently what is “reasonable care and skill”
- There are few standard practices in software development against which a developer’s level of care can be gauged.

1.2.3. The Supply of Goods and Services Act 1982 and expert systems

- The SGSA may cover the supply of expert systems, since the giving of advice by the system could be interpreted as the supply of a service.

- If an expert system is supplied by a dealer, who is the supplier? The dealer or the software company that made the expert system?
  - **Q. Why is this question important?**
    - It is vital to know who the parties are in the contract of sale (in case the system is defective). Only the parties to a contract can sue on it. Contractual liability for defects in the expert system will rest with the dealer or developer.

1.2.3. The Supply of Goods and Services Act 1982 and expert systems

- In the case of an off-the-shelf expert system that has been supplied by a dealer, the customer relies on the dealer to provide a suitable and effective system.

- If the customer specifies the system he wants, the other party in the contract will be the software company. The dealer will be an *agent* of the software company.

1.3. Breach of contract

- If a party to a contract breaches its terms, the remedy depends on the type of terms that have been broken.
- There are two types of terms in contracts; *warranties* and *conditions*.
  - A breach of *condition* gives the aggrieved party the right to **cancel the contract** and **claim damages**.
  - A breach of *warranty* allows the aggrieved party to **claim damages only** —the contract remains in force and must be completed by both parties.
-

1.3.1. *Example of breach of condition*

- Failure to deliver a product by an agreed date is a breach of condition; the buyer can cancel the contract.
- The buyer may also claim damages which would be equivalent to the difference in cost of buying a similar computer elsewhere.
- He can also claim other expenses and losses incurred as a direct result of the breach (e.g. loss of business as a result of not having the computer).

1.3.2. Example of breach of warranty

- A supplier agrees to deliver a computer system with terminals that are a particular colour, but delivers terminals with different colour.
- This is a breach of warranty unless there is a special reason why a particular colour was specified.
- The buyer will be entitled to damages only, and will still have to pay the agreed price of the computer.

1.4. Exemption clauses

- An exemption clause excludes or restricts the liability of a party who is in breach of contract.
- There are two types:
  - *Exclusion clauses* — give total exemption, e.g. a supplier may exclude his liability for late delivery if this is caused by circumstances beyond his control.
  - *Limitation clauses* - limits liability to a specified amount, e.g. software supplier may limit his liability for faulty software to the licence fee he has been paid.

1.4.1. Limitation of liability

- A person drafting a contract is always keen to limit or exclude their liability while ensuring that the other party is bound to perform his part of the contract.
- However, a one-sided state of affairs is rare since exemption clauses are controlled by statute - the *Unfair Contract Terms Act 1977*.
- This Act limits the extent to which liability can be excluded or limited for breach of contract or negligence.
- For example, business liability for death and personal injury caused by negligence cannot be excluded or limited in any way.

2. Contracts For Bespoke Software

- Appropriate software may not be available ‘off-the-shelf’ for certain specific tasks.
- Software can be written or adapted by a specialist software firm — a ‘software house’.
- We now consider common terms found in a contract for writing software.

- ***2.1. Definitions***
  - The first clause describes the parties to the contract, together with definitions relating to the software and hardware on which it will be installed.
  - The client’s full business name will be abbreviated to ‘client’ or ‘customer’. This saves space.
  - The terms ‘software’ and ‘hardware’ will usually be defined, which will assist with the readability, interpretation and construction of the contract.

2.2. Licence agreement

- Does the buyer actually *own* the software?
  - Usually, the buyer will licence the software rather than have ownership. A contract for writing software is therefore — in most cases — a licence agreement.
- If it is especially important for the buyer that the software it requires is not sold elsewhere, it should insist on an exclusive licence (an exclusive right to use the software) or an assignment of copyright (a transfer of copyright ownership).
- There are three other important points concerning the licence agreement;
  - *duration*,
  - *transferral*
  - and *scope*.

2.2. Licence agreement

- *2.2.1. Duration*
  - A licence must be for a fixed period of time. If a duration is not stated, it is likely that the licence will endure for as long as copyright subsists in the software.

- *2.2.2. Transferral*
  - The contract should state whether licence can be transferred to a third party. By default, licence agreements are usually assumed to be transferable.

- *2.2.3. Scope*
  - Is it permissible to run the software on more than one machine? If the buyer is a member of a group of companies, can the other members of the group also use the software? Can the software be transferred from one member of the group to another?

- All these questions should be anticipated when drawing up the contract, and should be discussed with the software house.

2.3. Contract price

- A contract is usually in the form of a licence agreement, so payment is termed **a *licence fee***. However, the fee may be called the *price* since it may also include training and documentation.
- The price should be stated precisely, and the contract should provide a mechanism for calculating the cost of additional work outside of the terms of the contract.
- The easiest way to do this is to state an hourly rate for programmers, analysts etc.
- If a lump sum is to be paid, does it include
  - maintenance and training?
  - documentation?
  - the cost of media such as magnetic disks and tapes?
- If the client is late paying,
  - does the contract include provision for charging him interest?
- What if the client refuses to pay?
- The contract should ensure that there is no ambiguity regarding the time for payment.

2.4. Specification

- The specification is the main provision in the contract that indicates what the software is going to do, and how it is going to do it. It is crucially important.
- Often the client isn't sure what he wants, and changes the specification during the development process. The original contract must allow for these changes.
- If the changes to the specification are considerable, it may be best to terminate the existing contract and negotiate a new one. This is called ***novation.***
  - A software house should never allow itself to be in a position where it has to renegotiate a contract!
- If the client is very vague about the specification, it would be better to build a prototype system and then use this to derive a much more explicit specification for the final software.
  - **Q: What is the problem with prototypes?**

2.4.1. Technical issues in the specification

The specification should indicate the following:

- A detailed description of the tasks the software will perform.
- The equipment on which the software will run.
- How quickly the software will carry out the operations required, bearing in mind any requirements for networking and concurrent use.

- *2.4.2. Micron Computer Systems v Wang (1990)*
  - Micron complained that the system they purchased from Wang did not provide 'transaction logging'.
  - The Judge made this observation:

> '..The acknowledged absence of a transaction logging facility is not in reality a fault in the system which was sold to Micron can only complain about its absence if Micron can establish a contractual term, express or implied, of an actionable representation, to the effect that the system included such a facility. In order to make good its case on transaction logging, Micron must therefore establish that they made known to Wang that they required such a facility..'

In other words, you can't sue for inadequacies in the product that are due to inadequacies in the specification.

Recent cases have questioned this principle.

2.5. Time for completion

- The usual method of dealing with late completion is to include a term in the contract which gives the client a right to ***liquidated damages***. These may be quantified
  - as a certain sum of money for every week completion is late (e.g. 500.00 € per week).
- A sum for liquidated damages must be a ***genuine*** estimate of financial losses that the client will suffer as a result of late delivery, not a penalty.
- When does the client know that completion has taken place?
  - The software may be installed but only partially working, or the documentation may not be complete.
- A possibility here is the concept of '**substantial completion**', where a large percentage of the agreed price is paid on the completion of a substantial part of the system, and the rest is retained until the remaining work has been completed.
- Of course, the meaning of '**substantial**' should be defined in the contract.

2.6. Maintenance and enhancements

- Software is like lettuce —testing reveals the presence of bugs, not their absence. Many bugs appear only after a long period of time.
- If a bug appears, this be a breach of warranty and the client can ask the software house to correct the error.
- The software house will wish to limit its responsibility for correcting such errors to a specified period of time.
- A software house will usually offer an ancillary contract for maintenance. Such an agreement will also provide for enhancements and updates of the software.
- If the client considers wants to be able to modify the software himself, a contract term should state this (recall the rules of error correction in the CCPR 1992).
- *2.6.1. Andersen Consulting v. CHP Consulting*
  - Ex-employees of Andersen set up their own business providing maintenance of Andersen's computer programs. Andersen argued that a term in their licence agreement prevented maintenance by third parties. However, the contract was American, in which a few terms had been changed for use in England. The judge refused to grant an injunction, and criticised the contract. Illustrates the need for a clear maintenance clause and contracts correctly phrased for English Law.

2.7. Escrow

- What happens if a software house goes out of business?
- Will its clients be able to maintain and modify their software or find another company to do this for them?
- Many contracts include an ***escrow clause*** to cover such situations. This is a form of insurance or guarantee should something happen to the software house.
- Escrow describes the situation where a software house deposits (with an independent person) the source code of the program together with copies of all the design documents (e.g. flowcharts, DFDs etc.).
- One organisation that provides this service is the National Computing Centre.
- Should the software house go out of business then the source code and documents will be released to the client, who will then have everything needed to arrange for the software to be supported.

2.8. Intellectual property rights

- The contract may impose duties on both parties associated with intellectual property rights.

- **Q. What will the client want?**
  - Use the software without any restriction. For example, what if the software infringes the copyright, patent or trade mark of a third party?
  - Protect any details of his business the software house has obtained.
  - During development, the staff of the software house will probably have access to confidential information such as client accounts, debtors and creditors.

- **Q. What will the software house want?**
  - Protect the copyright subsisting in the software, e.g. to prevent employees of the client from copying the software for their own use.
  - Protect the confidentiality of any special techniques for writing and testing software that the client's staff may have seen during the development and installation of the software.

- As well as contract terms, copyright law and the law of confidence will give protection to both parties.

- The client should insist on an *indemnity* term in the contract which prevents legal action being taken against him if the software infringes the intellectual property rights of a third party.

2.9. Other terms and standard contracts

- A contract for writing software may also cover training of the client's staff, conditions for termination of the licence and so on.

- In many cases, it will be possible to use or adapt a standard form contract, such as those published by the Institute of Purchasing and Supply.

2.10. Independent professional supervision

- It may be advisable to have large contracts overseen by a member of the BCS. He would be responsible for:
  - ensuring that the specification is met
  - general supervision
  - ensuring that payments are made and the completion date is met
  - fixing rates for delays or extra work
  - authorising time extensions for unavoidable delays
  - acting as an arbitrator
- The supervisor will probably be paid by the client.

3. Shrink-Wrap Licensing

- With microcomputer software, there is no opportunity for a signed licence agreement.

- The response of the software industry to this problem is the ‘shrink-wrap licence’. The contract is displayed on the packaging, visible through a clear plastic film. The terms are deemed accepted if the package is opened.

- Shrink-wrap licence purports to be a direct contract between the software producer and the consumer, quite separate from the contract of sale between the dealer and customer.

- Can such a contract actually exist in law?

3. Shrink-Wrap Licensing (cont'd)

- A contract requires three elements;
  - *offer*,
  - *consideration*
  - and *acceptance*.

- The display of the licence on the packaging constitutes an offer. Consideration relates to the fact that the licensee is paying a fee to use the software. Acceptance may be indicated by breaking open the package.

- Shrink-wrap licences are problematic because the opportunity to read the terms often comes after the contract is made, i.e. after the customer has paid and the software is physically handed to the customer.

- An important principle in contract law is that it is not possible to unilaterally introduce new terms into a contract after it has been made.

- So, shrink-wrap licences appear to be a unique form of contract, not two separate agreements.

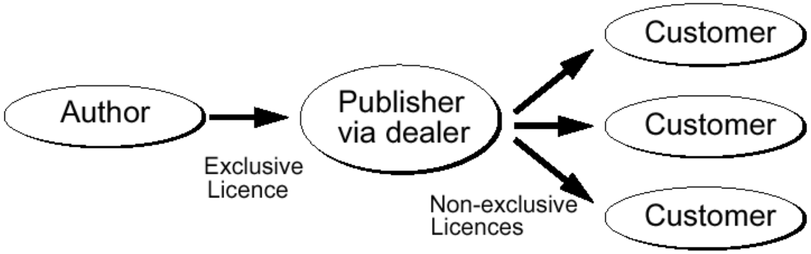
- Recent case law *(Beta Computers Ltd vs. Adobe Systems Ltd 1996* - see Bainbridge p.239) supports the notion that the customer may reject the software at any time before acceptance.

4. Contracts Between Author and Publisher

- Many companies publish software that has been developed by self-employed freelance programmers.
- **Q. Why do freelance authors publish their software through other companies?**

  Reputation and distribution network.

- The freelance owns the copyright in the software, so he will grant a licence to the publisher permitting it to market the product on the basis of a royalty payment.



- Normally, the licence will be exclusive, giving a single publisher the sole right to deal with the software. The publisher then has the same rights under copyright law as if it owned the copyright itself.

- **Q. Is this state of affairs advantageous to the author?**
  - Yes, since the author may not have sufficient funds to mount an infringement action himself.

4.1. Payment

- The software author will be paid a royalty by the publisher, which may be a fixed sum for every unit sold or a percentage of the price charged for the software.

- Alternatively, the author may be paid a lump sum for an exclusive licence.

- Some care should be taken when quantifying the amount of royalty payments in the contract:
  – If royalties are based on the price of the software, is this the retail price or the payment the publisher receives from a dealer?
  – Do dealers receive discounts for bulk purchases that could affect the amount of royalty payment?
  – What happens if the publisher does not try hard to market the software?
  – Can the author check the publisher's accounts at regular intervals?
  – How frequently will royalty payments be made?
  – Is the author able to terminate the contract if the publisher ceases to market the software?

5. Summary

- Software is unique in the way it is traded, and this has many implications for software contracts.
- Software is not ‘goods’; as a result, much legislation that protects the interests of customers does not apply.
- Contracts for bespoke software are complex and must be drafted with care - standard contracts can help.
- Shrink wrap licences purport to be a separate agreement between the software company and the purchaser; however, legally they appear to be inextricably linked with the contract of sale.
- Software is sometimes written independently and marketed via a publishing house; the publisher usually pays a royalty fee to the author.