CE29x Team-Project Challenge

Contracts

Professor Anthony Vickers
Room: 1NW.3.17 e-mail: vicka

with acknowledgements to Keith Primrose and Michael Fairbank

Announcements

- Git/Jira good practice tips:
  - https://blog.intracto.com/how-to-save-a-puppy-by-creating-a-clean-git-repo
- SAMT Survey
- Individual Challenge week (Monday 18th Jan – Friday 22nd Jan)
  - There will be an explanatory presentation by myself on the Monday morning.
  - The Challenges will require approximately 4 days work to complete and make a good pass.
  - Presentations and marking take place on the Friday – keep that day fully available.

Contracts - Context

- We have considered the legal framework and how it applies to our disciplines
- We have looked at laws protecting copyright, designs and patents
- We know that software and hardware sometimes go wrong - but what recourse do people have
- This lecture is about contracts and liability

Considerations

What can a company or an individual do when the software they have purchased does not perform as expected?

- What sort of agreements are there?
- What does the law have to say?
- What happens when one party fails to meet obligations?
- How can far can producers go in limiting liability?

Lecture outline

1. The sale of goods Act 1979
2. Contract Law
3. Contracts for Outsourcing
4. Contracts for Bespoke Software
5. Contracts for Consultancy and Contract Hire
6. The structure of a Contract
7. Ownership Issues
   - Using software licences

1. Sale of Goods Act 1979

Sale of Goods Act 1979

- Quality of supplied goods defined as:
  - Fitness for all the purposes for which goods of the kind in question are commonly supplied.
  - Appearance and finish.
  - Freedom from minor defects.
  - Safety
  - Durability.
- Can software quality be measured, and are minor defects OK?

St Albans City Council v ICL

- ICL is a former British computer company – now taken over by Fujitsu
- Council required a new system to administer the newly created Community Charge (Poll Tax, approx 1990).
- System failed in many aspects
  - Performance
  - Reliability
  - Database reported wrong figures
- Council under-recovered £1.3M revenue in first year.
- Finally awarded full amount after appeal.
- Actual cited acts were: Supply of Goods and Services Act 1982 and the Unfair Contract Terms Act 1977.

St Albans City Council v ICL (2)

- Important definition was made by the Judge:
  - Software was Goods (as defined by the Sale of Goods Act)
  - Without Software, Hardware could not function.
  - Software was not just information but actively defined the way the total system worked.
  - Also, an implication was made, if Software was not goods then what else could it be?
- However, current best practice is to define requirements, both functional & non-functional, within Contract of Supply rather than attempt to invoke the Sale of Goods Act after delivery.

2. Contract Law

Contract Law

- A contract is an agreement between two parties that is enforceable in law.
  - It sets out the aims of both parties, their obligations and responsibilities.
  - It defines at what point the contract is complete
  - It can also set out what should happen if problems arise

Contract Law

- Not every agreement is a contract. There must be:
  - An offer and an acceptance
  - ***valuable consideration*** ***i.e.***
    - Each party must be receiving something and providing something
  - An intention by all parties to make a contract
  - All parties must be competent (e.g. old enough and of sound mind)
- A contract between two parties can be made orally or in writing.
  - i.e. you can have a verbal contract

Validity

- Not every contract is valid (can be upheld in law) e.g.
  - A contract to commit a crime
  - A contract where one or other party is disqualified
- Even legal contracts can have restrictions e.g.
  - Sale of Goods Act (1979) & Sale and Supply of Goods Act (1994)
  - Supply of Goods and Services Act (1982)
  - Laws covering employment contracts
  - Etc.

Liability for Defective Software

- Contracts sometimes try to exempt the developers from liability for defective software
  - However the Unfair Contracts Act 1977 restricts the extent to which liability can be limited
  - Not possible to limit liability at all if the defect causes death
  - Also have the Sale of Goods act 1979

Specifics

- A contract should be as precise as possible re obligations and undertakings of those involved.
- A contract e.g. will specify
  - The system to be developed
  - The deliverables
  - The timescale
  - The payments
  - Etc

The Unforeseen

- The unforeseen happens
- When it does the courts rely on:
  - The contract itself
    - The conditions
    - The warranties (beware it has a very specific meaning)
  - Inducements – what the parties said when they were negotiating the contract
    - If someone made false statements then the Misrepresentation Act (1967) might apply
  - Implied terms – if the contract says nothing what should it have said?
    - Accepted practice
    - Precedents

3. Contracts for Outsourcing

Contracts for Outsourcing

- Large organisations regularly sub-contract design and implementation /manufacture to other parties
  - Called outsourcing
- It could be for Hardware and/or Software Development
- The work will be covered by a contract

Contracts for Outsourcing

- Outsourcing contracts are normally long and complex
- They try to make things as explicit as possible
    - How is performance to be monitored + managed?
    - What if performance is unsatisfactory?
    - Which assets are being transferred from the client to the outsourcer?
        - Existing IT equipment?
        - Existing staff?
    - Duration and termination conditions of contract?
    - Ownership of the IP generated?
    - Audit rights
        - The customer's existing external and internal auditors may still need access to the newly outsourced assets for inspection
    - Contingency planning + disaster recovery

4. Contracts for Bespoke software

Two domains for Software

- In regard to software we can consider two radically different domains:
  - Shrink-wrap software
    - An identical piece of software is sold to any customer who wants to buy it
  - Bespoke software
    - Custom software is written specifically for a large client

Contracts for Bespoke software

- These should specify what is to be produced,
  - E.g. based on a SRS document
  - But a good contract should specify how the requirements can be varied by the customer after the SRS is finalised,
    - including payment and time consequences
- Should specify the acceptance testing procedures
- What QA standards will be used
- What methodologies used during development

Contracts for Bespoke software

- The contract should also specify
  - Should the client get the source code?
  - And the build files? Build tools?
  - Unit tests used? What test results to expect?
  - Documentation
  - User manuals?
  - User training?
  - Training for client's maintenance staff

Contracts for Bespoke software

- It should be made clear who will own the IP rights
- Points to make clear:
  - What if we build their product faster using our in-house class libraries which we developed over many years?
  - Do we want to allow the client to use those afterwards?
  - Could the client sue us afterwards for continuing to use those class libraries in other projects?
  - Usually developer will try to keep the copyright themselves

Contracts for Bespoke software

- Any confidentiality clauses – to protect both parties
  - Protect our in-house software tools and methods
  - Protect their in-house business methodologies, customer contacts and trade secrets

Contracts for Bespoke software

Acceptance criteria should be specified in the contract

- Ideally, the client will produce a fixed set of acceptance criteria, which when passed, will result in acceptance of the system
  - Ideally this list is fixed, at the start of acceptance testing
  - It cannot grow
- Should specify who will be present during acceptance testing

Contracts for Bespoke software

Termination criteria should be specified,

- E.g. what if the client's business is taken over half-way through the contract?
- How to fix this amicably?
- Usually contract would pay the developer proportional to the work already done.

- What financial penalties are imposed on either party for walking away mid-way through
- Or for the developers failing to deliver on-time/at all.

5. Contracts for Consultancy and Contract Hire

Contracts for Consultancy and Contract Hire

- Big consultancies often just supply a number of people to the client
  - E.g. Cap Gemini leases a team of 1 DBA + 3 Application senior-programmers plus one business analyst to a client
- Or freelance computer contractors might get a placement with client

Contracts for Consultancy and Contract Hire

- Payment arrangements are usually fairly simple
  - People are just paid by the hour / day, to do whatever the client wants them to do
- Contract hire of persons is straightforward
  - E.g. pay a programmer to sit there and program
- Consultancy is more glorified form of this
  - Consultant might be an expert asked to assess some aspect of the client's operations, and try to improve it
  - Final deliverable might be a document

Contracts for Consultancy and Contract Hire

- It is important that consultancy contracts specify:
  - Confidentiality rules
  - Terms of reference
    - - what the contractor is expected to do when there
  - Liability
  - Who has control of the final deliverable

5. The structure of a Contract

The structure of a Contract

- Three main parts:
  1. Introduction
     - Names of the parties
     - Glossary of terms
  2. Standard Terms and Conditions
  3. Annexes or Appendices - define the specific project in detail
     - Requirements
     - Deliverables
     - Acceptance criteria
     - Price and Payment schedule etc etc

Standard Ts & Cs

- Scope of the Work
- Clients responsibilities
- Project Control
- Warranty and Maintenance
- Change Procedures
- Ownership of copyright
- Confidentiality
- Indemnity
- Termination criteria
- Arbitration procedures
- Relevant jurisdiction
- etc. etc.

Standard Ts & Cs

- See Sample Terms and Conditions for a Fixed-Price Contract
    - Referred to in this lecture's quiz

6. Ownership issues

Ownership issues

- If you design a system or write a bit of software who 'owns' it?
- Recall from the CE201 lecture on IP:
  - If you work for a company as designer or programmer (or even in a related role) then the company probably owns the copyright
  - If that company is a sub-contractor then it possible that the organisation that commissioned the design owns it
  - If you are a self-employed contractor …
    - It could be you, or the company that contracted you, or the company that contracted them
- You should make it so your contract specifies these details clearly

Exploiting Ownership

- How does the owner of a design or of copyright exploit it.
  - As with any other property right, it may be
    - sold,
    - licensed,
    - mortgaged,
    - assigned or transferred,
    - given away,
    - or simply abandoned.

Exploiting Ownership

Owners have three main options

1. Keep it "in house"
   - This is what many manufacturers do (although they may also license in another country).
2. Assign it to another party for a fixed sum or for royalties.
   - This is what authors of books typically do they assign copyright to the publisher for a fee.
   - The new owner is then free to exploit the copyright
3. Grant one or more licences to copy the work
   - With a software licence the user is granted the right to use the software and “copy it into the memory of their machine”, or similar legal wordings

Licence Agreements

- Usually software is not sold to the client outright.
- Usually the client is given a licence that allows certain usage actions
- May limit the client to usage on a limited number of machines
    - Or a max number of login accounts
    - Or usage for a given time period
    - May include upgrades and patches for the given time period

Licence Agreements

- Open-source licences allow anyone to use the source code and other design material, subject to certain conditions
  - E.g. GNU public licence

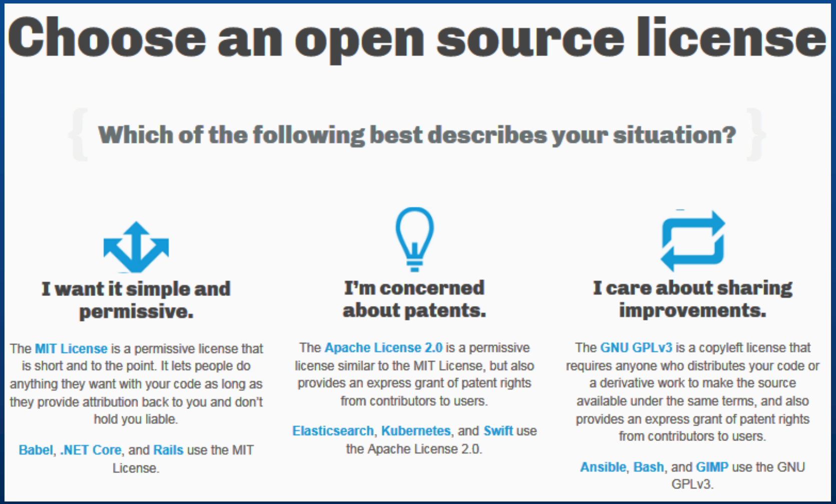
- Note that just finding code that is uploaded to github does not make it “free and open source”; the software will still be subject to copyright.
  - You need to check its licence to get permission to use it
  - Also note that “No licence ≠ FOSS”
    - “No licence = lawsuit bait”

Licence Agreements

- Think carefully what licence you put on code that you upload to the internet
  - And to the code you give to a client

Links:

- Comparison of free and open-source licenses
- http://choosealicense.com/



Choose an open source license

Which of the following best describes your situation?

I want it simple and permissive.

The MIT License is a permissive license that is short and to the point. It lets people do anything they want with your code as long as they provide attribution back to you and don't hold you liable.

Babel, .NET Core, and Rails use the MIT License.

I'm concerned about patents.

The Apache License 2.0 is a permissive license similar to the MIT License, but also provides an express grant of patent rights from contributors to users.

Elasticsearch, Kubernetes, and Swift use the Apache License 2.0.

I care about sharing improvements.

The GNU GPLv3 is a copyleft license that requires anyone who distributes your code or a derivative work to make the source available under the same terms, and also provides an express grant of patent rights from contributors to users.

Ansible, Bash, and GIMP use the GNU GPLv3.

Further reading

- “Software Contracts and Licences”, (Chapter 12 of course textbook, F. Bott, *Professional Issues in Information Technology,* 2nd Edition).
- Tip: Think about a licence for your CE29x team product, and licences of any tools you have used to make it

- Quizzes:
  1. Quiz on Contracts on Moodle

Bibliography

Ayers, R. 1999. "*The Essence of Professional Issues In Computing*", Harlow: Prentice Hall

Chapter 12, Bott, F. 2014. "*Professional Issues in Information Technology*", Second edition