

MOLINARI LEGAL CONSULTANCY

UNITED ARAB EMIRATES



LIABILITY IN THE INFORMATION SOCIETY

Software, information and the legal environment

by Ottavia Molinari

LIABILITY IN THE INFORMATION SOCIETY

Software, information and the legal environment

The modern organization and businesses of the new millennium are using extensively the information technology to such an extent that we can say that all aspects and functions in our lives are totally dependent on it ¹. The social vulnerability ² and great concern in the matter were clearly highlighted by the Year 2000 Millennium bug ³ when the worldwide estimated costs for checking and repairing the information systems amounted to the figure of £ 400 billions ⁴.

A computerized information system consists of a number of interrelated components that work with the aim to convert inputs into data and its most relevant components are the hardware and the software ⁵. While the term "hardware" ⁶ describes the physical components (or its invariable part) of computers, telecommunications, and other information technology devices ⁷, the term "software" ⁸ describes the variable part of computers such as the various kinds of programs, which are used to operate these and their related devices. It seems generally recognized that while the "hardware" is legally identified as a tangible product and therefore classified as a "good", the legal status of software still remains open to various interpretations which vary from the theory of an intangible product as a "service" to the opposite conception of a tangible product and therefore as a "good". A response to the question, still not answered definitely in European law, on whether the software is a good or a service has extreme relevance for a determination of satisfactory forms of protection. We can easily imagine how many could be the practical examples of defective software applications ¹¹, which may cause physical injury,

-
1. http://elj.warwick.ac.uk/jilt/cases/97_2fuji/ Lloyd I, "Software Patents After Fujitsu" . New Directions or (another) Missed Opportunity?, Case Note 1997 (2) *The Journal of Information, Law and Technology (JILT)*. Case Note published on 30 June 1997. "Software design has developed from a restricted craft activity to a major global industry whereby the turnover and profits of software companies such as Microsoft may easily dwarf those of giant industrial enterprises".
 2. Peter de Jager and Richard Bergeon's "Managing 00: Surviving the Year 2000 Computer Crisis" John Wiley and Sons, 1997 222 pp.
 3. http://elj.warwick.ac.uk/jilt/bookrev/98_1y2k/ John S. Craparo, " The Year 2000 Problem, a Review of the Literature", Book Review, 1998 (1) *The Journal of Information, Law and Technology (JILT)*.
 4. <http://www.richmond.edu/jolt/v6i2/note4.html> Comment, Philip j. Landau, *Products Liability in the New Millennium: Product Liability and the Y2K Crisis*, 6 RICH. J.L. & TECH.12 (Fall 1999).
 5. Paul Bocij, Dave Chaffey, Andrew Greasley, Simon Hickie, *Business Information Systems*, Technology Development and Management; pag. 697.5
 6. <http://www.pcmec.com> A clear description of hardware components with diagrams and pictures.
 7. Hussain , K. and Hussain, D. (1995) *Information System for Business*, Prentice Hall, Hemel Hempstead. Chapter 4 and 5.
 8. <http://whatis.techtarget.com/> This site is a glossary giving succinct definitions of terms relating to software.
 9. <http://berkowitzfirm.com/expertsystemsliability.htm> Potential theories of Legal Liability for Defective Expert System software. By Rober D. Sprague and Leslie G. Berkowitz9
 10. <http://itlaw.law.strath.ac.uk/ENLIST/subjects/liability/commentary/eu.html> *Liability for defective software in the European Union*. Software Goods or Service- pag ¼.
 11. <http://www.badsoftware.com/support1.htm> Liability for Bad Software and Support . A paper presented to the Software Support Professionals Association Executive Briefing, San Diego, October 3, 1996 and to the Software Support Conference East, March 12, 1997 by Cem Kaner.

loss of records, business, productivity and damages to all computer systems etc. For example, in the medical field it has been suggested that anesthesia software provided to patients undergoing operations may result less expensive and more effective in monitoring all conditions of the patients than the traditional one. However in the event of failure, it could even cause the death of the patient and in this case it is still not clear who should be considered liable: whether the anesthesiologist, the hospital or the software developers?¹².

Software can be classified in three main categories: A) The "STANDARD SOFTWARE" also called " OFF-THE-SHELL" package (which represent the majority of software purchased by private consumers) where fixed intangible information and adoption of programs are recorded into physical disks called CD-ROM and made immediately available to customers; and this type of software is identified as a good ¹³. However, we have to recall that the same version of software programs can be also downloaded through Internet without the physical support of a floppy disk and in the same way it seems it would be equally identified as a good. B) The software "MADE TO MEASURE" is a program, ideated and designed for specific clients in order to meet their detailed necessities; this type of software is identified as a *service* in consideration that the contract previews the work in the area of design. C) Finally, the "CUSTOMIZED SOFTWARE represents a category in between the other two above mentioned since its programs are characterized by a composition of standard applications directly selected by the customer and additional programs specially ideated by programmers in order to meet their necessities. This type of software is considered as a *service* in consideration of the great amount of work provided by the creation of new special programs. The classification conducted above is of relevance in case of software failure. The engineering design ¹⁴ and manufacturing company can be found negligent if they have failed to use reasonable care either in manufacture or design of a software product ¹⁵ placed into the market corrupted, especially if the company may lack in a proper quality system ¹⁶. In the normal course of applications customers purchasing the faulty software would realize such failure only when trying to install the program in the computer. The main argument in case of dispute shall be based on applied and available general software ordinary ¹⁷ standard and work-processing program manufactured by other producers ¹⁸.

Software's quality is also measured according to its suitability for the works or the problems it is intended to facilitate or solve: whether it can meet the customers' business requirements and if it can perform with certain reliability. The quality of software ¹⁹ is dependent basically on two

12.<http://www.salon.com/tech/feature/1999/08/05/anesthesia/index.html> by Andrew Leonard, "A proposal for open-source anesthesia software heightens the drama of the question; Who's at fault when software fails?"

13. Stuurman C. "Product Liability for Software in Europe. A Discussion of the EC Directive of 25 July 1985" in Vandenberghe GPV (ed.) [1989] Advanced Topics of Law and Information Technology, Kluwer, Deventer, pp127-147.

14.<http://www.computer.org/tab/seprof/code.htm> Software Engineering Code Of Ethics And Professional Practice (1999). In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to Eight Principles herein described.

15.<http://www.richmond.edu/joilt/v6i2/note4.html> The Richmond Journal of Law and Technology, vol.VI, Issue 2, fall 1999. Philip J. Landau.

16.<http://www.kaner.com/theories.htm> Software Liability (1997). See also Legal Issues and Software Quality. Kaner, Jeynote address to annual meeting of the Software Division of the American Society for Quality, October 1997).

17. For example, a computer programmer who represented himself as possessing skill and qualification necessary to design and develop a computer program for an oil company implied possession of the skill and diligence ordinarily possessed by well-informed members of the computer programming trade and breached an implied promise of having reasonable skill and ability when he failed to design the computer program needed. *Data Processing Services Inc. v. LH. Smith Oil Corp.*, 492 N.E.2d 314, 320 (Inc.Ct. App. 1986) USA.

18. How software may fail? Note Theme 1. Software, Information and Legal Environmental.

19. Comments On Software Quality by Watts S. Humphrey, Fellow, Software Engineering Institute Carnegie Mellon University, PA. 2Bguide-NCCUSL meeting.

factors: the number of errors (or bugs ²⁰) and the suitability for its intended use. While the number of errors is quite easy to be verified, the suitability purpose is much more difficult to quantify since it is dependant on a number of elements. However, we can say that the basic technical criteria for such analysis include correct functions, speed and easiness of use.

The software environment, as for example the co-existence with other programs and sharing of sources should be also considered an additional element of quality assessment.

A claim for damages incurred for defective software is not specifically dealt by software laws and therefore, it would be essential to clarify whether the software is a good or a service keeping in mind that the *product liability law* is applied only to goods often found under *consumer protection measures* ²¹. However, in consideration that defective software may be used in a single computer as well as in a net of integrated computers, its program failure could cause damages ranging from a small amount to millions of dollars and these damages would be difficult to quantify and insure prior to the purchase and utilization of the software. As a result of this, the manufacturers and developers now tend to minimize their legal responsibility applying limited liability clauses ²² in their licensing contracts with the end users or to exclude it entirely ²³ as for instance in the contract of "shrink-wrap licence" ²⁴.

If in possess of a regular contract, the consumer who has bought a defective software from the retailer may, on presentation of a valid receipt, usually within a week time ²⁵, ask for the refund of the money paid ²⁶. However, if the damages occurred are higher than the price of the software he may file a claim directly against the producer. When a contract does not exist, the final user must seek protection under the *general tortious liability law* (which is more time spending and expensive) under the principles of negligence ²⁷, breach of warranty ²⁸ and fault of the manufacturer or in case of negligence certification ²⁹ of independent software testing companies when inspecting, testing and approving defective software products.

By Ottavia Molinari

Data 4/5/02

20. Software bugs are defects in a program, which are caused by human error during programming or earlier in the lifecycle. See also Jones, C. (1996) *Software Quality; Analysis and Guidelines for Success*.

21. Uniform Commercial Code (U.C.C.) See also comment of Adv. Bruno Secchi regarding the protection²

22. http://elj.warwick.ac.uk/jilt/cases/97_3stal/stalban.htm Caveat Vendor? A review of the Court of Appeal Decision in St. Albans City and District Council v International Computer Limited. 1997(3)

23. <http://www.cix.co.uk/~brethertons/corpfile/articles> A case in the Scottish Court of Session has determined that shrink-wrap licences can be enforceable.

24 http://www.batnet.com/oikoumene/softwr_licnsetrnds98.html#distribute Trends In Software Licensing And Legal Protection For Software 1998 by Fred M. Greguras.

25. In Italy as per Dr. Leg. 22 May 1999 n.185, in reception of Directive 97/7/CE. In case of long distance contract for goods and services to protect customers when buying and opening defective software sealed products the prescription time for refund would be 10 days while in absence of warranty 3 months from the date of finalizing the contract. [Http://www.tariffe.it/club/normative.htm](http://www.tariffe.it/club/normative.htm)

26. Cem Kaner & David Pels. *Bad Software: What To Do When Software Fails*. Chapter 1. (1997).

27. <http://www.berkowitzfirm.com/expertsystemsliability.htm> *Potential Theories of Legal Liability for Defective Expert System Software*. Dr. Robert D. Sprague and Leslie G. Berkowiz.

28. Alan L. Dorris & Jerry Purswell, *Warnings and Human Behavior : Implications for the Design of Product Warnings*, 1J.PROD. LIAB.225, 225 (1977).

29. <http://www.kaner.com/coverage.htm> Software Negligence and Testing Coverage. By Cem Kaner.