UIC John Marshall Journal of Information Technology & Privacy Law

Volume 8
Issue 2 *Computer/Law Journal - Spring 1988*

Article 2

Spring 1988

Strict Products Liability and Computer Software, 8 Computer L.J. 135 (1988)

L. Nancy Birnbaum

STRICT PRODUCTS LIABILITY AND COMPUTER SOFTWARE

by L. NANCY BIRNBAUM

I. INTRODUCTION

Software manufacturers must become aware of the legal implications which products liability doctrine can have upon their software production. In *American Standard Handbook of Software Law*, John C. Lautsch states that "if software is to become as important to the American economy as many predict, products liability law may one day be the most practical area of business knowledge for a Software Writer."[1] The purpose of this paper is to give software manufacturers a much-needed overview of the evolution of strict products liability doctrine, its effect on software manufacturers, and the ways to avoid lawsuits and win those which may arise. According to Lautsch, it is very important for software manufacturers to understand this field of law:

> There are no reported products liability cases turning on defectively designed or manufactured code. However, the rate at which programmed devices are spreading throughout society, such cases seem inevitable. The prudent Software Producer is familiar with the business implications of products liability law and has taken management steps to reduce the risk that either he or his company will produce and market code that causes injury, death, or property damage.[2]

II. A HISTORY OF STRICT PRODUCTS LIABILITY

Products liability is a field of law which began to develop during the mid-1800s but has expanded in recent years. Strict products liability is the liability of a non-negligent seller to a third person with whom the seller has no privity of contract.[3] This means that the seller is liable for a person's injuries even when there is no contract between the seller and the injured person. The evolution of strict products liability doctrine began with the case of *Winterbottom v. Wright*.[4] In that case, the

1. LAUTSCH, AMERICAN STANDARD HANDBOOK OF SOFTWARE LAW § 10.1 (1985).
2. *Id.* § 10.11.
3. W. PROSSER, LAW OF TORTS 641 (4th ed. 1971).
4. 10 M. & W. 109, 152 Eng. Rep. 402 (1842).

plaintiff's employer contracted with the defendant to keep mail coaches in good repair. The plaintiff, a mail coach driver, was seriously injured as a result of inadequate repairs by the defendant. The mail coach driver sued the repair company but, without the existence of strict products liability doctrine, the court found that, since there was no privity of contract, the mail coach driver had no claim against the repair company. The policy applied in this case was *caveat emptor*—let the buyer beware. This suit was one of the last important cases where such reasoning was applied.

The first precedent for strict products liability was *Thomas v. Winchester*.[5] In *Thomas*, the defendant was a drug manufacturer who mislabelled a poison as a harmless medicine. The defendant sold the medicine to a dealer who sold it to another dealer who sold it to Mr. Thomas. Mr. Thomas then gave it to his sick wife who became critically ill as a result of ingesting the poison. The court ruled that, since "the defendant's negligence put human life in imminent danger," this case was an exception to the *Winterbottom* ruling.[6] Therefore, no privity of contract was needed because the product was inherently dangerous. That is, since the defendant knew that the drug was not to be used by the buyer—it was intended for use by a medically ignorant consumer—and that it would be dangerous to mislabel the drug, the defendant was held liable. This decision did not overrule the precedent set in *Winterbottom*; it simply instituted an exception to that rule. Privity of contract was still required unless the product was inherently dangerous.

In 1916, the decision in *MacPherson v. Buick Motor Co.*,[7] extended the inherently dangerous standard initiated in *Thomas v. Winchester* to products which are dangerous if incorrectly manufactured. The *MacPherson* case involved a car manufacturer who failed to inspect a defective wooden wheel which Buick had purchased from a wheel manufacturer. The court decided that the car was foreseeably dangerous if a component of it was defective and, therefore, the manufacturer should have subjected the component parts to tests before placing the finished product on the market. In other words, the manufacturer made a representation of safety by offering the goods for sale. He assumed a responsibility to the consumer which was not based on a contract but rather on a relationship due to the consumer's purchase. As a result of the court's ruling in *MacPherson*, the *Thomas* exception swallowed the *Winterbottom* rule. The emerging rule was that a seller is liable for negligence in the manufacture or sale of a product which may

5. 6 N.Y. 397 (1852).
6. *Id.* at 437.
7. 217 N.Y. 382, 111 N.E. 1050 (1916).

reasonably be expected to cause harm if it is defective.[8] That is, the rule applies to all products which, if negligently manufactured, could cause physical harm or property damage.

After *MacPherson*, the courts were flooded with products liability cases. In the leading case, *Henningsen v. Bloomfield Motors*,[9] a car manufacturer and a car dealer were held liable to the wife of the purchaser of a defective automobile. In order to recover, the plaintiff was not required to prove that the defendants were negligent or knew of the defect. Using an implied warranty of safety theory and the doctrine of strict products liability, the court decided that privity of contract between the parties was unnecessary and that proof of negligence was irrelevant to liability.

As strict products liability developed rapidly in the courts, there was a move toward standardization in the *Restatement (Second) of Torts*. In states:

> 402A. Special Liability of Seller of Product for Physical Harm to User or Consumer.
>
> (1) One who sells any product in a defective condition unreasonably dangerous to the user or consumer or to his property is subject to liability for physical harm thereby caused to the ultimate user or consumer or to his property, if
>
> (a) the seller is engaged in the business of selling such a product, and
>
> (b) it is expected to and does reach the user or consumer without substantial change in the condition in which it is sold.
>
> (2) The rule stated in subsection (1) applies although
>
> (a) the seller has exercised all possible care in the preparation and sale of his product, and
>
> (b) the user or consumer has not bought the product from or entered into any contractual relation with the seller.[10]

Greenman v. Yuba Power Products,[11] was the first case in which the Restatement rule was applied. In this case, the court held a manufacturer strictly liable for the injuries sustained by the plaintiff while operating a defective wood lathe. Since *Greenman*, the Restatement rule has been adopted in one form or another by most jurisdictions in the United States. As a result, a new doctrine has arisen in the law, that is, *caveat vender*—let the vendor beware.

A. ANALYSIS OF SECTION 402A

In order to understand section 402A, the wording of the rule and its

8. *Id.* at 643.
9. 32 N.J. 358, 161 A.2d 69 (1960).
10. RESTATEMENT (SECOND) OF TORTS § 402A (1965).
11. 59 Cal. 2d 57, 27 Cal. Rptr. 607, 377 P.2d 897 (1963).

ambiguities require clarification. A brief analysis follows. Since Section 402A applies only to the purchase of a product and not a service, software must be considered a product for section 402A to apply. Strict liability doctrine evolved from cases involving the sale of goods[12] so products liability applies to transactions involving a sale of goods. Although the distinction between products and services is usually clear, it is not in the case of computer software. Mass distribution, mass production, and mass marketing of an item definitively result in a product rather than a service.[13] In *Triangle Underwriters v. Honeywell*,[14] the court stated that a transaction is for a "'service' rather than a 'sale' when 'service predominates' and the sale of items is 'incidental'." *Triangle Underwriters*.[15] For example, if a hospital gives a blood transfusion, the transaction is for the health care service rather than for the sale of blood.[16] On the other hand, if a furniture store sells a bedroom set and agrees to deliver it, the transaction is for the sale of the bedroom set rather than for the delivery service. Therefore, a transaction is for a service when the service is the essence of the transaction and for a product when the product is the essence of the transaction.

If the transaction is for a product, there are two types of defects which can result in a strict products liability suit—design defects and manufacturing defects. A design defect results when the product is manufactured in accordance with the manufacturer's plans but the plans for the item were unreasonably dangerous.[17] That is, the product does not perform as safely as a reasonable customer expects it to perform, or the risk inherent in the design of the product outweighs the benefits of using such a design.[18] When a case arises involving a design defect, the court assesses the possibility and value of using alternative product designs.[19] By using models of the same product for comparison, the plaintiff attempts to show that a nondefective design was reasonably available.[20] If it is determined that the manufacturer should have known of the potential for injury (regardless of the manufacturer's ac-

12. Nycum & Lowell, *Common Law and Statutory Liability for Inaccurate Computer-Based Data*, 30 EMORY L.J. 445, 461 (1981).

13. Note, *Strict Products Liability and Computer Software: Caveat Vendor*, 4 COMPUTER/L.J. 373, 391 (1983).

14. 604 F.2d 737 (2d Cir. 1979).

15. *Id.* at 742.

16. *See* Perlmutter v. Beth David Hosp., 308 N.Y. 100, 123 N.E.2d 792 (1954). The court found that the furnishing of blood was secondary to the provision of hospital services.

17. M. GEMIGNANI, COMPUTER LAW 425 (1985).

18. M. SCOTT, COMPUTER LAW § 7.12 (1985).

19. Brannigan & Dayhoff, *Liability for Personal Injuries Caused by Defective Medical Computer Programs*, 7 AM. J.L. & MED. 123, 124 (1981).

20. *Id.* at 135.

tual knowledge), the court then decides whether the manufacturer should have altered the design before placing the product on the market.[21]

On the other hand, a manufacturing defect results from the incorrect implementation of a safe design.[22] When the product deviates from the manufacturer's intended design,[23] (that is, the product fails to satisfy the designer's specifications) a manufacturing defect exists. An example of a manufacturing defect is a programmer's mistake in carrying out the system designer's instructions.[24]

Section 402A is ambiguous in defining "unreasonably dangerous." According to the *Restatement (Second) of Torts*, "The article sold must be dangerous to an extent beyond that which would be contemplated by the ordinary consumer who purchases it, with the ordinary knowledge common to the community as to its characteristics."[25] A product may be unreasonably dangerous not only due to an error in manufacturing or due to an unreasonably dangerous design, but also if adequate instructions or warnings are not provided.[26]

One misunderstanding which manufacturers have about section 402A is the assumption that, by disclaiming liability in a contract, they can avoid strict products liability suits. Strict liability cannot be disclaimed, however, and limitations in contract do not apply.[27] Warranty disclaimers are ineffective against strict liability, as the courts have held that a consumer cannot and should not be bound by a disclaimer that he has never seen.[28] For this reason, warning labels, may be ineffective as protection against strict liability, if it is likely that the average consumer will not read them.[29] For example, it is commonly known that many software users do not read users' manuals. If a manufacturer's only notice to a consumer is in the users' manual, it is unlikely that consumers will be aware of the warning. A court may hold the manufacturer liable despite the warning unless it is one which the consumer cannot avoid reading.

B. Limitations of Strict Products Liability

Despite the fact that strict products liability may seem all-encompassing, there are limits to its applicability. Strict liability extends only

21. *Id.* at 137.
22. Gemignani, *supra* note 17, at 425.
23. Scott, *supra* note 18.
24. Brannigan & Dayhoff, *supra* note 19, at 138.
25. Restatement (Second) of Torts, *supra* note 10.
26. Prosser, *supra* note 3, at 659.
27. Gemignani, *supra* note 17, at 413.
28. *See* Prosser, *supra* note 3, at 656.
29. Lautsch, *supra* note 1, at § 10.11.

to those third parties whom a manufacturer can expect to be endangered by use of the product.[30] If the retailer is expected to alter or inspect the product, and injury results because the retailer failed to do so, the manufacturer is not liable.[31] For example, if the manufacturer of a product advises the retailer that he expects the retailer to assemble it before selling it to a consumer, the manufacturer is not liable for an injury resulting from the incorrect assembly of the product by the retailer. If a product is unavoidably dangerous or the danger of its use is generally known (i.e. drugs with known side effects, cigarettes), manufacturers generally escape liability.[32] Unavoidably dangerous products are not covered by strict products liability because the product is no more dangerous than the ordinary consumer expects and the purpose of strict products liability is not to discourage the manufacture of unavoidably dangerous products—such as drugs with known side effects—if there are no alternative products available.

Another limitation of liability is that section 402A applies only when personal injury or property damage results from the defect.[33] Purely economic injuries, such as customer loss or poor business decisions resulting from defective software, are not covered by section 402A.

Although the aforementioned limitations of liability exist, strict products liability is still onerous. "Strict liability in tort exists if there is a defect in the product, even though there is no negligence on the part of the supplier."[34] Liability is absolute; it does not depend on a lack of due care by the seller or the manufacturer.[35] Nor is the manufacturer freed from liability if no amount of care could have prevented the injury.[36] Also, the manufacturer cannot claim that he has not made a contract with the injured party because lack of privity is not a defense.

C. Proof Required for Liability

In order to establish a products liability case, a plaintiff must prove three things. First, the product was defective when it was sold or leased.[37] If the product was mishandled by a retailer and the defect occurred while the retailer possessed it, the manufacturer is not liable. Second, the product was used in an intended or reasonably foreseeably

30. PROSSER, *supra* note 3, at 662.
31. Prosser, *supra* note 3, at 660.
32. Prosser, *supra* note 3, at 660.
33. SCOTT, *supra* note 18, § 7.12.
34. I. BROWN, THE LAW OF COMPUTERS 54 (1971).
35. M. GEMIGNANI, LAW AND THE COMPUTER 57 (1981).
36. Brannigan & Dayhoff, *supra* note 19, at 129.
37. SCOTT, *supra* note 18, at § 7.12.

manner.[38] If the consumer uses a lawn mower to trim hedges, the manufacturer will not be held liable because the product was not intended to be used for trimming hedges and the use was not reasonably foreseeable. On the other hand, if someone sits on a table and it collapses, the manufacturer may not defend himself by arguing that the table was not intended to be sat upon because it is reasonably foreseeable that people will sit on tables. Third, the defect was a proximate cause of the injury.[39] If the tires of a car are defective and the owner hits a pedestrian as a direct result of driving while intoxicated, the manufacturer cannot be held liable.

D. Policy Reasons for Strict Products Liability

As a result of the broad language of section 402A, "any person or company which, as part of its regular business, had anything to do with fabricating or distributing a defective product that reached the stream of commerce and hurt people, may be sued for products liability."[40]

This ominous description of strict products liability may prompt some manufacturers to wonder why the rule is so stringent—maybe the law is "out to get" manufacturers. There are many policy reasons for the evolution of strict products liability. "Society has decided that it is not too much to ask that an item not injure people."[41] One goal of products liability is referred to as loss spreading. Rather than placing the burden of hospital costs and property costs on the unlucky consumer who buys a "lemon," the law attempts to spread the burden more evenly across a wide section of society in order to reduce the burden on any one individual.[42] Since the product is useful despite its ability to do harm if defective (i.e. automobiles), the manufacturer is encouraged to buy insurance and pass the cost of the insurance on to the users of the product. Thus, those who share the benefit of the product also share the burden and those who are injured by the product are adequately compensated for their injuries.

Another reason for products liability is the representation of safety that the manufacturer makes when placing goods on the market.[43] An ordinary consumer assumes that, when he buys a new car, the wheel will not fall off after driving five miles. A consumer is able to make this assumption because he knows that the manufacturer may be held liable for products that are defective. The manufacturer's duty of a care

38. Scott, *supra* note 18, at 97.12.
39. Scott, *supra* note 18, at 97.12.
40. Lautsch, *supra* note 1, at § 10.4.
41. Note, *supra* note 13, at 390.
42. Gemignani, *supra* note 17, at 418.
43. Prosser, *supra* note 3, at 651.

arises because of its affirmative conduct in placing the product on the market.[44] Relating to the manufacturer's duty of care owed to the consumer is the public's demand for maximum legal protection from product defects against which it is helpless to protect itself.[45] In short, consumers rely on the reputation of the manufacturer.

Prior to strict products liability, an injured consumer had to rely on privity of contract in order to receive compensation for his injury.[46] First, the consumer would sue the retailer by relying on their contract or a warranty. Then, the retailer would sue the person who sold the product to him or her through their contract or warranty. That seller could sue the person who sold the product to him and the suing would continue until the manufacturer was sued by the person who originally bought the item from him or her. Obviously, strict products liability substantial decreases court costs and lessens the burden on the court system by allowing the injured party to sue the manufacturer directly.

The policy of strict liability evolved because it is often not feasible for a consumer to prove negligence. If the product is very complex, it may be impossible or extremely difficult for a consumer who knows little about the workings of the product to identify the source of the negligence which was responsible for the defect.[47] Under strict liability principles, the consumer must only prove that the product was defective and, as a result, unreasonably dangerous. In applying strict liability, the court will then hold the manufacturer liable even though the plaintiff has not proven that the manufacturer was negligent. For example, a consumer buys a soda bottle and, in the process of unloading his groceries, he picks up the bottle which then explodes in his hand. Glass flies into his eyes resulting in a loss of sight in one eye. Consequently, he sues the manufacturer of the bottle. He proves that the bottle was defective—not a difficult task since non-defective bottles do not explode under ordinary circumstances. He also proves that the bottle was unreasonably dangerous—another easy task since the explosion of a glass bottle is not expected by the ordinary consumer and can result in physical injuries. Should he have to determine how the bottle was defective in order to prove that the manufacturer's negligence resulted in the defect? Determining why the bottle was defective is nearly impossible after it has exploded. But, without knowing why the bottle was defective, the plaintiff cannot show that the manufacturer was negligent. Therefore, if the courts applied a negligence theory to this type of case, it would be almost impossible for the injured consumer to recover dam-

44. R. NIMMER, THE LAW OF COMPUTER TECHNOLOGY § 7.06 (1985).

45. PROSSER, *supra* note 3, at 651.

46. PROSSER, *supra* note 3, at 651.

47. Lanoue, *Computer Software and Strict Products Liability*, 20 SAN DIEGO L. REV. 439, 448 (1983).

ages despite the fact that the manufacturer must have been negligent. This sticky situation resulted in the evolution of strict products liability doctrine which settles cases that negligence theory is inadequate to handle.

Still another reason for applying strict liability to products is that the manufacturer will be more careful about the safety of the products which he puts on the market.[48] Since the manufacturer is in the best position to discover and prevent defects, the manufacturer should be encouraged to do so. If the manufacturer puts a potentially dangerous product on the market without warning the consumer of the risk, he should be held liable for any resulting damage;[49] the party in the best position to detect and correct defects should be responsible for damages by defective products.[50] The goal of products liability is "to compel designers and manufacturers of commercial goods to design their products with the safety of the user in mind."[51]

Finally, application of strict liability to products insures that injured parties receive adequate compensation. Since manufacturers usually hold more assets and are more likely to have insurance than retailers, an injured consumer may not receive enough compensation for his injuries if he may only sue the retailer.[52]

Strict products liability is usually *not* applied when the contracting parties are on equal footing, i.e. two large companies. In deciding whether to apply strict liability, there must be a consumer (usually domestic with little bargaining power) as one party to the transaction. The court also reviews the primary purpose of the transaction, the nature of the business, and the manufacturer's relationship to the customer.[53]

The American Standard Handbook of Software Law states that "the government's correct role is to influence the design and manufacturing of products themselves so that they operate safely regardless of the user's behavior."[54]

III. PRODUCTS LIABILITY EXAMPLES INVOLVING SOFTWARE

The rest of this paper will focus on how strict products liability affects software manufacturers. A software manufacturer could be held

48. GEMIGNANI, *supra* note 17, at 420.
49. GEMIGNANI, supra note 17, at 421.
50. Note, *supra* note 13, at 373.
51. LAUTSCH, *supra* note 1, at § 10.2.
52. GEMIGNANI, *supra* note 35, at 62.
53. Note, *supra* note 13, at 390.
54. LAUTSCH, *supra* note 1, at § 10.2.

liable under strict products liability in many potential scenarios. The liability costs could be astronomical. Suppose a design developed by a defective architectural program resulted in the Hyatt skywalk collapse or the Hartford arena roof collapse. What if a defective air traffic control program caused the Korean airline disaster?[55] Computer problems have already been blamed for a near collision of two jet liners and the closing of a nuclear plant[56]—what if these computer problems had not been discovered until it was too late? The manufacturer's liability would be enormous.

Here are just a few scenarios. A computer-controlled, hospital life-monitoring system crashes and the patient dies.[57] A program used for air traffic control fails to monitor one aircraft and a crash results.[58] A family's home computer is used to detect burglary and fires, the program proves defective, a fire occurs, and the entire family perishes.[59] A program controlling the tracks for subway cars goes haywire and directs two cars to the same track; a crash results.[60] A program is used to run a chemical plant; the program malfunctions resulting in the release of a toxic chemical into the atmosphere.[61] A computerized device used to monitor the administration of anesthesia causes a patient's death.[62] A bridge designed by a defective computer program collapses.[63]

According to the Wall Street Journal, several injuries have already resulted from computer malfunctions.[64] The chance of a products liability lawsuit being filed against a software manufacturer increases daily, given the recent development of strict products liability and increasing reliance on computers. Defective computer software has the capability of producing catastrophes of astronomical proportions and it seems that it is only a matter of time before strict liability is imposed for defective computer software. Computers are being used more and more to save labor and speed processes. Uses are becoming more and more sophisti-

55. *A.L.I.-A.B.A. Course of Study Materials: Computer Law* 245 (1984).

56. Gemignani, *Products Liability and Software*, 8 RUTGERS COMPUTERS & TECH. L.J. 173 (1981).

57. LAUTSCH, *supra* note 1, at § 10.1.

58. Gemignani, *supra* note 56, at 197.

59. Note, *supra* note 13, at 374.

60. Nycum, *Liability for Malfunction of a Computer Program*, RUTGERS COMPUTER & TECH. L.J. 1 (1979).

61. R. FREED, COMPUTERS AND LAW: A REFERENCE WORK 38 (1976).

62. *Id.*

63. *Id.*

64. Wall St. J., Jan. 28, 1987, at 1, col. 6. A man's computerized radiation-therapy machine malfunctioned killing him with excess radiation. California stopped payment on checks which would have paid bondholders $4 million in excess interest because of a software bug. Computerized medical products such as programmable pacemakers have been subject to computer errors. A computerized air-defense system's frequency problem resulted in the deaths of twenty sailors when a missile's signal was not received.

cated. It is a "greater challenge for people to anticipate and provide for, in advance, the almost infinite number of circumstances that might be encountered when" a method is computerized.[65] Because of the public's greater reliance on computers the legal system is adapting to new problems raised by this reliance. For example, programmers are expected to use the highest degree of care and to be responsible for the consequences of their mistakes.[66]

IV. REASONS TO APPLY STRICT PRODUCTS LIABILITY DOCTRINE TO SOFTWARE

Some manufacturers might argue that, despite the possibility of the above-mentioned scenarios and lawsuits occurring, manufacturers could not be held liable since products liability does not apply to computers. Although there have not been any cases in this area, the policy reasons behind strict products liability doctrine seem to indicate that it pertains to computer software. Like the products to which strict liability pertains, computers and their software benefit society. Software defects are difficult to predict and can be unreasonably dangerous.[67] As with the doctrine of strict liability, the software manufacturer is better able to assess risks than the technically illiterate user and he can either prevent them from becoming realities, or warn customers of their existence.[68] By putting their programs on the market, software manufacturers invite the public to use them, implying that the product is safe.[69]

The above factors add to the probability that strict products liability will be applied to computer software. Assume, however, that the courts decide that strict liability does not apply. This could result in a random application of the law. There is no good reason to treat computer software differently from other products and a random application of the law is not warranted. For example, a person would then be able to win a suit against an automobile manufacturer for a defective steering mechanism but he would be denied recovery if a defective computer program in the car had caused the same injuries.[70]

Some people may grant that computer software is subject to products liability, but that it should be subject to a negligence standard rather than strict liability standard. Under a negligence standard, the injured party would have the burden of showing that the programmer

65. FREED, *supra* note 61, at 39.
66. *A.L.I.-A.B.A. Course of Study Materials: Computer Law*, *supra* note 55, at 244.
67. GEMIGNANI, *supra* note 17, at 420.
68. GEMIGNANI, *supra* note 17, at 420.
69. S. MANDELL, COMPUTERS, DATA PROCESSING, AND THE LAW 122 (1984).
70. Lanoue, *supra* note 47, at 447.

failed to use due care in creating the program. This burden would include finding the mistake in thousands of electronic bits and proving that the injury was reasonably foreseeable. This burden would be imposed even though the injured person may know nothing about the operation of a computer. Under a strict liability standard, the plaintiff must still show that the product was defective and unreasonably dangerous but does not have the heavy burden of proving that the programmer or the manufacturer was negligent.[71]

The best argument against applying strict liability to computer software is that software is not a product but rather it is the result of a service. "Thus far efforts to find strict liability as to the services themselves have entirely failed."[72] This does not mean that one cannot be held liable when providing services. The supplier of a service can be held liable for negligence.[73] A service is defined as something which is rarely duplicated allowing little chance for quality control or defect testing. For example, doctors perform a service because every case and every person is different.[74] In *Barbee v. Rogers*,[75] an optometrist failed to fit a pair of lenses correctly. The defendant was not held strictly liable because lenses are "not a finished product offered to the general public in regular channels of trade."[76] Also, the defect was in the service performed not in the product and the policy reasons for applying strict products liability were not present. Services are ordinarily intangible.[77] For example, a doctor giving a general examination is providing a service. Another definition of a service is the expertise provided by people such as architects, engineers, and attorneys.[78]

In contrast to a service, a product is a manufactured item such as a car or a soda bottle[79] which is in the stream of commerce; that is, available to the general public.[80] Products are "items suitable for prepackaged, off-the-shelf purchases."[81] A product is normally considered a tangible object.[82]

There are conflicting opinions as to whether software is a product or a service. For the most part, courts have held that programs are

71. Lanoue, *supra* note 47, at 447.
72. PROSSER, *supra* note 3, at 679.
73. Brannigan & Dayhoff, *supra* note 19, at 124.
74. Lanoue, *supra* note 47, at 450.
75. 425 S.W.2d 342 (Tex. 1968).
76. Barbee v. Rogers, 425 S.W.2d 342, 346 (Tex. 1968).
77. Brannigan & Dayhoff, *supra* note 19, at 130.
78. *See* Lanoue, *supra* note 47, at 444.
79. Lanoue, *supra* note 47, at 444.
80. MANDELL, *supra* note 69, at 122.
81. Note, *supra* note 13, at 383-4.
82. Brannigan & Dayhoff, *supra* note 19, at 130.

products.[83] Because of the problem of program "piracy," where a thief "steals" a program, copies it and sells it under a different or even the same name, courts are easing up on the barrier of intangibility for program patents. Therefore, they may relax the doctrine of intangibility for computer software as a product as well.[84] Another factor which may influence the decision whether to treat computer software as a product is that courts have refused to impose professional liability on computer specialists.[85] One reason that courts impose professional liability on doctors is that they are not normally in a better condition to discover a defect than their patients are.[86] For example, doctors are not in a better position than patients to discover a defect in a stethoscope. Software is what some courts call a "hybrid" case in which the courts must determine whether the transaction was primarily for a product or a service.[87] In *Triangle Underwriters*,[88] the court held that a transaction for software is a saie not a service.

Although an argument can be made that a software transaction involves a service rather than a sale of goods, courts are moving away from artificial distinctions between sale and service transactions.[89] The courts' policy reasons for limiting the scope of strict liability,[90] however, actually support the hypothesis that strict liability should be imposed on software manufacturers. First, software manufacturers are in a better position to prevent the risk of harm from defective programs than the

83. *Cf.* RRX Industries v. Lab-Con, 772 F.2d 543 (9th Cir. 1985) (the court, applying California law, held that, in a transaction for employee training, repair services, system upgrading, and a software package, the sales aspect predominates); *cf.* Chatlos Systems, Inc. v. National Cash Register Corp., 479 F. Supp. 738 (D.N.J. 1979), *aff'd*, 635 F.2d 1081 (3d Cir. 1980) (the court, applying New Jersey law, held that a leasing arrangement involving computer hardware and software was a transaction for a sale of goods); *but cf.* Data Processing Services v. L.H. Smith Oil Corp., 492 N.E.2d 314 (Ind. Ct. App. 1986), *aff'd*, 493 N.E.2d 1272 (1986) (since DPS was retained to design a system for Smith's specific needs, the court held a services transaction was involved).

84. Lanoue, *supra* note 47, at 446.

85. SPECIAL COMM. ON COMPUTERS & LAW ASSOC. OF THE BAR OF N.Y., *Committee Reports Tort Theories in Computer Litigation*, 38 REC. A.B. CITY N.Y. 426, 430 (1983).

86. Note, *supra* note 13, at 380.

87. Note, *supra* note 13, at 380.

88. 604 F.2d 737, 742 (2d Cir. 1979).

89. *See* Johnson v. Sears, Roebuck & Co., 355 F. Supp. 1065 (E.D. Wis. 1973) (the court rejected a sales/service analysis of a transaction for hospital treatment and found that imposition of strict liability should occur on an ad hoc basis); *see* Newmark v. Gimbel's Inc., 54 N.J. 585, 258 A.2d 697 (1969) (beauty parlor operator held strictly liable for injuries resulting from a permanent wave application).

90. *See* La Rossa v. Scientific Design Co., 402 F.2d 937 (3d Cir. 1968) (the court refused to hold a manufacturer strictly liable for a defectively designed chemical plant because the plant was specially designed and had no impact on the public); *see* Immergluck v. Ridgeview House, Inc., 53 Ill. App. 3d 472, 368 N.E.2d 803 (1977) (court refused to apply strict liability to a nursing home operator because policy reasons were lacking).

technologically ignorant user.[91] Second, in many cases the programmer is the only person who can prevent errors or warn the user of possible problems.[92] Indeed, the injured party may not even know that a computer is being used[93] (such as when a computer is used for air traffic control). Third, testing a software product is simpler than testing some other products such as structures like homes or office buildings because the programmer has ample opportunity to test the program through simulation.[94] Holding software manufacturers strictly liable will provide an incentive to avoid accidents and to insure against unavoidable risks.[95]

In addition to the similarities between the policy reasons for strict liability in software cases and strict liability in other cases, other factors the argument that software should be considered a product. In general, public opinion is that software is a commodity to be bought and sold.[96] One might argue that a defective design is unreasonably dangerous but software is intangible and, therefore, software manufacturers cannot be held liable for defectively designing software. But, the design of an automobile is also intangible. The difference is that the car's design becomes a concrete object and most computer program designs do not.[97] There are, however, numerous exceptions to this general statement where a software produces tangible output, such as software which assists architects in drafting the blueprints to a building. Software with tangible output should not be treated differently than software that does not produce a tangible output. Therefore, if software with tangible output is ruled by strict liability standards, software without tangible output should also be ruled by strict liability principles.

Another problem with considering software sales as service transactions involves plug-in modules. These modules, which are inserted into the computer to become a part of the hardware, seem like a product. But, the law should not base liability on the medium of a particular program. This type of rule would simply provide the manufacturer with an incentive to avoid certain types of media and sell the same software product through a different media.[98]

An argument can be made that mass-produced and mass-marketed software is a product but custom-made software is a service. The problem with this type of legal development is that a "custom-made" pro-

91. *Cf.* Note, *supra* note 13, at 389.
92. Nycum, *supra* note 60, at 17.
93. Nycum, *supra* note 60, at 17.
94. Lanoue, *supra* note 47, at 450.
95. Nycum, *supra* note 60, at 17.
96. Lanoue, *supra* note 47, at 453.
97. Nycum, *supra* note 60, at 17-18.
98. GEMIGNANI, *supra* note 17, at 422.

gram can be the result of a slight modification of another program; the result is a blurred distinction between mass-produced software and custom-made software.[99] "Where the computer was an integral part of the operation upon which a mass of consumers depend, a court would have little trouble finding that the travelers were the ultimate consumers as they were the party for whose benefit the items were purchased."[100]

While a program can be manufactured on a large scale or on an individual scale so too, can products such as automobiles.[101] Therefore, scale of production is a poor benchmark for determining whether software is a product of a service.

Various cases also seem to dictate that the courts will consider software a product. In *Halstead v. United States*,[102] a defect in a navigational chart resulted in a plane crash. The defendant claimed that the essence of the transaction was the conveyance of information and, therefore, he had supplied a service rather than a product. Applying Colorado law, the court ruled against the defendant stating, "If suitable for mass marketing, the information is in some sense a fungible good for which the manufacturer placing it on the market must assume responsibility."[103] Since a program is a tangible expression of an idea that can be put to use, it is similar to the idea of a navigational chart.[104] Therefore, a mass marketed program may be viewed by the courts as a product thereby allowing the application of strict products liability to software.

Another argument is that software is like electricity, because electricity is a form of energy similar to computer program impulses. At least one court has held that electricity is a product, therefore programs should be considered products.[105] In *Ransome v. Wisconsin Electric Power Co.*,[106] electricity provided and distributed through transmission lines, might well be a service, but the electricity itself was considered a product. The court decided that electricity is a product because it is distributed in the stream of commerce and consumer self-protection is not feasible. As with electricity, most software is in the stream of commerce and consumer self-protection is not feasible.

Michael C. Gemignani believes courts will eventually apply strict liability to software manufacturers. "If imposition of strict liability in tort would make the manufacturers of company hardware and software

99. Note, *supra* note 13, at 398.
100. Note, *supra* note 13, at 398.
101. Lanoue, *supra* note 47, at 444.
102. 535 F. Supp. 782 (Conn. 1982).
103. *Id.* at 791.
104. Brannigan & Dayhoff, *supra* note 19, at 130.
105. Lanoue, *supra* note 47, at 447.
106. 87 Wis. 2d 605, 275 N.W.2d 641 (1979).

more careful and thoughtful in their race to develop an ultimate product, that alone would justify its application."[107]

V. PREVENTION OF STRICT PRODUCTS LIABILITY LAWSUITS

As a result of the seemingly all-encompassing, no-fault definition of strict products liability, a manufacturer may think that there is no way in which it can prevent a lawsuit. The attitude of waiting, and dealing with a problem after a suit is initiated is called crisis management.[108] Crisis management can be economically expensive as well as reputation damaging. Prevention of products liability lawsuits through quality and safety control—reducing product hazards—is known as safety management.[109]

How can one prevent products liability suits through safety management? Giving users proper warnings and directions is one way to avoid suits.[110] Embedding warnings in a program is usually necessary since it is commonly known that users do not read instruction manuals accompanying software.[111] An effective warning should be placed on socially valuable products which cannot be made safer with current technology.[112] In supplying warnings, a manufacturer should advise the user of the possibility and impact of errors and of the limits of the program. Warnings must be clear and not just state that errors are possible.[113] Clear warnings include the proper ways to use a program to avoid dangerous results and the external factors which may affect the program's use or results.[114]

Instructions are closely related to warnings but one cannot be substituted for the other. The instructions must be sufficient to make the product safe.[115] Specifying the performance expected of the system as clearly as possible is an important facet of instructions.[116] A lack of information about the product can be considered a product defect.[117] In *Midgley v. S.S. Kresge Co.*,[118] [119] a retailer sold a refracting telescope to a thirteen-year-old boy. A warning on the product indicated that per-

107. Gemignani, *supra* note 56, at 204.
108. LAUTSCH, *supra* note 1, at § 10.7.
109. *Id.* LAUTSCH, *supra* note 1, at § 10.7.
110. Prosser, *supra* note 3, at 661.
111. *Lautsch, supra* note 1, at § 10.6.
112. *Id.* LAUTSCH, *supra* note 1, at § 10.6.
113. Nycum, *supra* note 60, at 19.
114. BENDER, 2 COMPUTER LAW § 11.03[2].
115. LAUTSCH, *supra* note 1, at § 10.6.
116. FREED, *supra* note 61, at 30-1.
117. GEMIGNANI, *supra* note 35, at 65.
118. 65 Cal. App. 3d 67, 127 Cal. Rptr. 217 (1976).
119. 168 Wash. 456, 12 P.2d 409 (1932).

manent damage to the eyesight might result if the telescope were used to look at the sun. A sun filter which was provided with the product was to be used to view the sun. Instructions on how to attach the sun filter were included with the telescope but no diagram was provided. As a result of the absence of a diagram, the child attached the sun filter incorrectly and his sight was damaged. The court held the defendant liable because the defendant knew that the product would be used by technically unsophisticated persons and that the instructions provided would be the user's only guide. Therefore, the instructions and warning were insufficient to render the product safe. This case has bearing on the case of software in that the purchasers of software are often technologically illiterate and the instructions provided by the software manufacturer are normally the user's only guide. Thus, a manufacturer must beware of insufficient instructions or warnings which may render its product the object of a lawsuit.

Besides using effective warnings and clear instructions, a manufacturer should ensure that the claims it makes for a product's performance are true. In *Baxter v. Ford Motor Co.*, the manufacturer was held liable when a pebble shattered a car's windshield which had been advertised as shatterproof. Also, puffed-up advertising and sales pitches may result in the dishonoring of a disclaimer by a court. (Disclaimers may be honored, however, if they are present in fairly negotiated contracts.)[120]

There are several other ways to avoid lawsuits. The obvious one is to take all steps to detect and correct any program malfunctions.[121] Using condition statement and error checking techniques to avoid input errors can reduce the risks of harm to users.[122] The potential user interface errors are: (1) no response; (2) inappropriate response; (3) incorrect recognition; and (4) inappropriate timing. A user might not respond if the user does not understand that the program requires a response. An inappropriate response occurs when the user hits the wrong key. If the user does not notice a hazard in the program, incorrect recognition results. Inappropriate timing may occur when a user responds too quickly or too slowly—the program should have adequate timing margins.[123] Software programmers should take all steps possible to avoid these user interface errors. For example, the program could check critical responses by requiring the user to retype the request.

There are many steps which management may take to avoid product liability suits. They may involve some budget modifications, but it

120. Gemignani, *supra* note 56, at 177-8.

121. FREED, *supra* note 61, at 29.

122. *Id.* at 33.

123. LAUTSCH, *supra* note 1, at § 10.5.

must be remembered that it is much less expensive to avoid law suits than to be involved in one. The following is a list of options a manager might employ:

1) Specify a management group which will be concerned with product safety.[124]
2) Employ a top management officer to head the safety effort, who is capable of causing a redesign.[125]
3) Include a corporate attorney in system planning, who is acquainted with the technology.[126]
4) Use third party documentation writers to provide clear users' manuals.[127]
5) Initiate a program to teach employees product safety and to remind them of the product safety goal.
6) Reward employees for reducing hazards in a product's code.[128]
7) Make proper time and budget allocations for product safety testing.
8) Require safety even at the sacrifice of efficiency, (i.e., if the input is critical, check the answer with more than simply a Y or N response).[129]
9) Make certain that the advertising and sales personnel describe the design and specifications of the program correctly.
10) Organize a program which effectively receives and acts upon consumer complaints about product safety.[130]
11) Voluntarily correct any errors which are discovered after the product is marketed.[131]
12) Ensure that managers know the trend of consumer comments and the software problems which occur.[132]
13) Ensure that usage reports and complaints are reviewed by designers and implementers to analyze possible corrections.[133]
14) Keep an organized record of changes made to a product during the development of the design.
15) Record product inspections and tests results as they occur.
16) Develop a program to assure that product safety risks are not produced by subcontractors or vendors.
17) Implement an organized program for investigating problems and recommending design changes.[134]

124. LAUTSCH, *supra* note 1, at § 10.7.
125. LAUTSCH, *supra* note 1, at § 10.10.
126. FREED, *supra* note 61, at 30.
127. LAUTSCH, *supra* note 1, at § 10.6.
128. LAUTSCH, *supra* note 1, at § 10.10.
129. LAUTSCH, *supra* note 1, at § 10.7.
130. LAUTSCH, *supra* note 1, at § 10.10.
131. *A.L.I.-A.B.A. Course of Study Materials: Computer Law*, *supra* note 55, at 245.
132. LAUTSCH, *supra* note 1, at § 10.10.
133. LAUTSCH, *supra* note 1, at § 10.7.
134. LAUTSCH, *supra* note 1, at § 10.10.

18) Have a reaction plan for timely steps in the event of product safety problems after the software has been marketed.[135]
19) Develop a systematic way to keep management informed of current legislation, agency regulations, and court rulings in the products liability field.
20) Conduct periodic checks to see if the product safety process is adequate.
21) Use an organized plan to correct product hazards.[136]
22) Review new products with risk management in mind.[137]

There are several steps which programmers should be encouraged to take in order to avoid products liability lawsuits. These include:

1) Develop structured programs.
2) Employ extensive testing.
3) Use comprehendable commands.
4) Employ stress testing (i.e., can the user answer correctly while under stress?)
5) Ensure that critical commands reduce the possibility of human error.
6) Use safe screen design to lessen the likelihood of error.
7) Design easy to use and easy to understand documentation.
8) Permit user override in the event of a hazardous situation.
9) Ensure that the program works with minor input errors (recognition of a command allows that the space bar be hit before the command).
10) State what the software will and will not do in the documentation.[138]

Two important approaches to avoiding products liability lawsuits bear mentioning. First, if no amount of caution can remove the serious risk of harm from use of the product, the product should not be developed.[139] Second, "Programmers should . . . make every feasible effort to make users (and other foreseeably affected persons) actually aware of the potential dangers of personal injury or property damage.[140]

A. How to Win a Strict Products Liability Lawsuit

In the event that, after taking all possible precautionary measures, a software manufacturer is the defendant in a products liability suit, it may win by following these guidelines. First, during voir dire, the attorney should pick people who recognize the social utility of computers

135. Lautsch, *supra* note 1, at § 10.7.
136. Lautsch, *supra* note 1, at § 10.10.
137. *A.L.I.-A.B.A. Course of Study Materials: Computer Law*, *supra* note 55, at 246.
138. Lautsch, *supra* note 1, at § 10.9.
139. Freed, *supra* note 61, at 34.
140. Mandell, *supra* note 69, at 123.

and not people who view computers as a threat to their employment.[141]

In addition to the way the attorney chooses a jury, there are many ways to win a products liability suit through the use of various defenses. Although most courts use very strict standards before they will consider plaintiff's conduct a defense to strict products liability, assumption of risk is an affirmative defense.[142] For example, if a user knows of the danger, the danger is obvious to the user, or the danger is commonly known and it is reasonable to assume that the user is familiar with it, then an assumption of risk defense might be invoked.[143] The defense of contributory negligence, however, will not be accepted simply because the plaintiff does not discover the defect or fails to guard against a possible defect.[144]

A second possible defense is called the "state of the art defense." That is, the defect resulted from a customary practice in the industry; the product available was the ultimate in existing technology, or the product resulted from the best technology reasonably available;[145] and the danger was inevitable given the current state of the art.[146] In order to invoke the state of the art defense, the product must be as safe as currently technology and economics permit and the buyer must have been alerted to the risks involved.[147]

If the product was grossly misused by the consumer in a manner not reasonably foreseeable by the manufacturer, the manufacturer cannot be held responsible. For example, the software may have been run on the wrong computer, the user may have subjected the software to a prolonged period of use without adequately checking it for flaws,[148] or the product may have been used in some other unforeseeable way.[149]

Two other possible defenses exist. In one case, the manufacturer must prove that the flaw could not be discovered prior to the occurrence of the harm.[150] This is a very heavy burden of proof because more testing and simulation usually will expose virtually every defect. In the second case, the manufacturer may prove that the retailer sold the product to the consumer when the retailer had knowledge of the defect and failed to correct it or warn the purchaser.[151] Therefore, by

141. FREED, *supra* note 61, at 30.
142. MANDELL, *supra* note 69, at 123.
143. PROSSER, *supra* note 3, at 649.
144. GEMIGNANI, *supra* note 17, at 444.
145. GEMIGNANI, *supra* note 56, at 200.
146. GEMIGNANI, *supra* note 35, at 59-60.
147. GEMIGNANI, *supra* note 17, at 448.
148. Gemignani, *supra* note 56, at 200.
149. R. BIGELOW & S. NYCUM, YOUR COMPUTER AND THE LAW 132 (1975).
150. Gemignani, *supra* note 56, at 200.
151. BIGELOW & NYCUM, *supra* note 149, at 132.

notifying retailers of all software defects, a manufacturer might avoid being held liable for a defective product.

B. Protecting Against Economic Losses—Insurance

Suppose, after doing everything possible to prevent a products liability lawsuit, a software manufacturer is sued and it loses the case. The best way to protect the software company against excessive and catastrophic financial losses is through insurance. First, itemize the risks: determine which ones are already covered by the existing insurance policy and analyze the rest. Are they normal business risks and, therefore, uninsurable because of high premiums? Insure the rest and continually reanalyze the uninsurable risks since the insurance premiums may decrease.[152]

When buying insurance, shop around. There are several kinds of insurance which cover products liability. Comprehensive (general liability) insurance covers damage to third parties by the insured party but certain intangible computer processes may not be covered—check with the insurer.[153] Errors and omissions insurance is also available but it is expensive and it usually contains high deductibles. Also, it often does not cover programs—only the output of the programs.[154] Some insurance companies now offer insurance for software manufacturers which is comparable to malpractice insurance.[155]

Also, check what is covered by an insurance policy. For comprehensive general liability insurance, amounts which a manufacturer legally must pay the injured party and legal costs are covered. For comprehensive, and errors and omissions insurance, punitive or exemplary damages, fines, penalties, and the cost to correct the problem are *not* covered. Also, comprehensive insurance does not cover liability assumed under contract, pollution liability, damage to the product arising from the program, or the loss of tangible property use resulting from performance delays or the failure of the product to perform in the way intended. Errors and omissions insurance does not cover bodily injury and property damage resulting from a professional service, liability under contract, disputed fees, dishonesty, fraud, or mechanical and electrical failure.[156]

In addition to the above items to be aware of, insurance purchasers should take other steps to maximize their investment. Check for an ad-

152. FREED, *supra* note 61, at 31.

153. J. SOMA, COMPUTER TECHNOLOGY AND THE LAW § 3.35 (1983).

154. FREED, *supra* note 61, at 43.

155. Henkel, *User Suits Give Rise to 'Malpractice' Insurance*, COMPUTERWORLD 24, Sept. 26, 1983.

156. *A.L.I.-A.B.A. Course of Study Materials: Computer Law*, *supra* note 55, at 248.

equate discovery period. That is, if the error occurs during the period of the insurance policy but is not discovered until after the policy expires, will the insurance policy cover it?[157] Use a broker who is an expert in computer-related insurance.[158] Chances are that he will be able to understand the company's unique problems and, therefore, provide more effective help than a technically illiterate broker. Give simple information about what needs to be covered.[159] Most underwriters do not understand technical jargon and they will be more willing and able to help if they know exactly what kind of insurance is required.

VI. CONCLUSION

By increasing awareness of potential legal problems, software manufacturers and programmers can do a great service to their companies. Through recognition of the existence of strict products liability law and its bearing on the computer software industry, a software manufacturer can take steps to avoid becoming a defendant in a products liability lawsuit. If unable to avoid a lawsuit, it can now choose the appropriate defense. And, in the event that the manufacturer loses the suit, the insurance which it has obtained as a protective measure might result in the avoidance of bankruptcy or, at the very least, a decrease in the company's financial loss.

157. MANDELL, *supra* note 69, at 123.
158. *A.L.I.-A.B.A. Course of Study Materials: Computer Law*, *supra* note 55, at 249.
159. *A.L.I.-A.B.A. Course of Study Materials: Computer Law*, *supra* note 55, at 249.