Bad Bytes: The Application of Strict Products Liability to Computer Software

Lori A. Weber
BAD BYTES: THE APPLICATION OF STRICT PRODUCTS LIABILITY TO COMPUTER SOFTWARE

INTRODUCTION

Computerization is rapidly becoming an essential component of our functioning society. This growth in automation has increased the risk that manufacturers and suppliers of software will be exposed to liability when their products fail to perform as expected. Although to date there have been few products liability cases involving computer software, the frequency of litigation involving other intangibles suggests that, as computer use expands,

1 See, e.g., William M. Bulkeley, Abstracts, Wall St. J., Jan. 23, 1981, at 31 (consultant predicts “all executives will eventually be computer-literate”); Sally Reed, Decade’s Top Jobs, N.Y. Times, Oct. 13, 1985, § 12, at 17 (estimating computer use to multiply ten-fold over next decade); Robert J. Samuelson, Our Computerized Society, Newsweek, Sept. 9, 1985, at 73 (as many as 10 million home computers in use by 1990).

2 See generally R. Lee Hagelshaw, The Computer User’s Legal Guide 86-88 (1985) (defining “software”). Although various definitions exist, in general, “software” is the intangible portion of a computer system which issues a set of instructions to the computer itself. See id. For purposes of this Note, the terms “computer software” and “computer programs” are used interchangeably.

3 See John C. Lautsch, American Standard Handbook of Software Law 263 (1985) (computers “will inevitably become a focus of liability”); L. Nancy Birnbaum, Strict Products Liability and Computer Software, 8 Computer/L.J. 135, 144 (1988) (strict liability inevitably will be imposed because defective software is capable of producing catastrophes); David A. Hall, Note, Strict Products Liability and Computer Software: Caveat Vendor, 4 Computer/L.J. 373, 374-75 (1983) (courts have not addressed application of products liability to computer programs, but increased computer use makes issue ripe for adjudication).

4 See Susan Lanoue, Comment, Computer Software and Strict Liability, 20 San Diego L. Rev. 439, 440 (1983). Defects in computer programs may occur because an error in program instructions took place during copying or because the original program contained some flaw. Id. “A program malfunctions basically in one of two ways: either it will not run at all, or it will not run according to its program.” Michael C. Gemignani, Product Liability and Software, 8 Rutgers Computer & Tech. L.J. 173, 176 (1981).


6 See, e.g., Sears, Roebuck & Co. v. Employers Ins. of Wausau, 585 F. Supp. 739, 744
products liability actions against software manufacturers will become increasingly common. 7

Recently, in Winter v. G.P. Putnam's Sons, the Ninth Circuit stated in dictum that computer software may be sufficiently tangible to be considered a "product" for purposes of products liability law. This Note will consider whether the Ninth Circuit was correct in its characterization of computer software. Part One will discuss the classification of software as a "good" under the Uniform Commercial Code and examine the approach courts have taken in determining the applicability of the Code to cases involving

(N.D. Ill. 1983) (liability may exist for incorrect information since insurance policy made no distinction between physical instruction manual and its intellectual content). Charts containing instrument approach information for aircraft or topographical information have been held to be products subject to products liability. See infra note 71 and accompanying text; see also Raymond T. Nimmo, The Law of Computer Technology § 7.06[2][b], at 7-26 (1986) (analogizing software to aeronautical charts). See generally Andrew T. Bayman, Note, Strict Liability for Defective Ideas in Publications, 42 Vand. L. Rev. 557 passim (1989) (discussing policies underlying application of strict liability to publications).

9 Cf. Joseph P. Zammit & Maria A. Savio, Tort Liability for High Risk Computer Software 1 (1987). "We are faced with the increasing likelihood that cases involving 'malfunctioning' computer software causing bodily injuries and property damage will be before our courts." Id.

9 938 F.2d 1033 (9th Cir. 1991).

10 Id. at 1035-36. The plaintiffs in Winter became severely ill after consuming poisonous mushrooms described in The Encyclopedia of Mushrooms as edible. Id. at 1034. Because the expressions and ideas contained in the book were deemed intangible and "[p]roducts liability law is geared to the tangible world," id., the Ninth Circuit held that the publisher could not be held liable under a products liability theory for information contained in the book. Id. at 1036. However, the court suggested that "[c]omputer software that fails to yield the result for which it was designed may . . . [constitute a product]." Id. In support of its decision, the court cited the Restatement (Second) of Torts, which provides the following:

The rule stated in this Section is not limited to the sale of food for human consumption, or other products for intimate bodily use, although it will obviously include them. It extends to any product sold in the condition, or substantially the same condition, in which it is expected to reach the ultimate user or consumer. Thus the rule stated applies to an automobile, a tire, an airplane, a grinding wheel, a water heater, a gas stove, a power tool, a riveting machine, a chair, and an insecticide. It applies also to products which, if they are defective, may be expected to and do cause only "physical harm" in the form of damage to the user's land or chattels, as in the case of animal food or a herbicide.

RESTATEMENT (SECOND) OF TORTS § 402A cmt. d (1965) [hereinafter RESTATEMENT].

Regarding the court's remarks on computer software, software law expert Michael A. Epstein stated, "'It's not the holding; it's dicta. But, in any event, for the court to come down and signal its acquiescence for the development of the law in this direction is substantial.'" Victoria Slind-Flor, Supplier Pulls Software—Ruling's Dicta Causes Uproar, Nat'l L.J., July 29, 1991, at 3. The significance of the court's statement was demonstrated when a computer software supplier temporarily withdrew its products from the market in response to this dictum. Id.
/software. Part Two will focus on products liability law and the question of whether computer software constitutes a "product." Part Three will consider the potential impact of strict products liability on the computer industry and the First Amendment issues raised by the imposition of this type of liability. Finally, Part Four will suggest a fact-specific, policy-oriented approach for determining whether strict products liability theory should be applied to computer software.

I. THE CHARACTERIZATION OF SOFTWARE UNDER THE UNIFORM COMMERCIAL CODE

As computer use has become widespread, the public increasingly has encountered software malfunctions, or "bugs,"\(^1\) which can cause property damage or even bodily injury.\(^2\) When a malfunction occurs, the characterization of software as either a good or a service is critical to determining what, if any, contractual liability may be incurred by the manufacturers and suppliers of software.\(^3\)

---

\(^1\) See JAMES V. VERGARI & VIRGINIA V. SHUE, FUNDAMENTALS OF COMPUTER—HIGH TECHNOLOGY LAW 601 (1991). A "bug" is "[a]n error or malfunction in a computer program. Such errors can range from barely detectible quirks to fatal flaws." Id.

Debugging software is an arduous task that continues even after the program has been introduced into the marketplace. See Hall, supra note 3, at 396. Because the creation of software programs is a complicated process, "it is just about impossible ever to debug a program completely." See Roy N. Freed, Products Liability in the Computer Age, 17 JURIMETRICS J. 270, 275 (1976); see also Pezzillo v. General Tel. & Elec. Info. Sys., 414 F. Supp. 1257, 1265 (M.D. Tenn. 1976) (roughly 80% of programmer's time devoted to debugging), aff'd, 572 F.2d 1189 (6th Cir. 1978); Gemignani, supra note 4, at 185 (no level of testing can ensure that program is completely debugged). Thus, even the most rigorous software examinations can result in a program that malfunctions. Id.


If a court determines that a defendant software manufacturer was providing a service, the plaintiff will be forced to rely on negligence principles, requiring a demonstration that (1) the defendant had a duty, (2) the defendant breached the duty owed to the plaintiff, (3)
This is because contracts involving "transactions in goods"13 are subject to the provisions of article 2 of the Uniform Commercial Code, while those involving services are not.14 If a court concludes that a software transaction is subject to the Code, the manufacturer and supplier may be held liable for breach of the implied warranties of merchantability and fitness for purpose.15 However, because computer software contains both tangible and intangible properties, its classification as either a good or a service is problematic.16

Several courts considering the characterization of software have held that computer programs are goods for purposes of the

---

13 U.C.C. § 2-102.
14 See id. ("unless the context otherwise requires, this Article applies to transactions in goods"). "Goods" are defined under the Code as "all things . . . which are movable at the time of identification to the contract other than money . . . , investment securities and things in action." Id. § 2-105(1). See generally Andrew Rodau, Computer Software: Does Article 2 of the Uniform Commercial Code Apply?, 35 Emory L.J. 853 (1986) (discussing application of U.C.C. to software).
15 See U.C.C. §§ 2-314 to -315. Section 2-314 states in relevant part: "Unless excluded or modified, a warranty that the goods shall be merchantable is implied in a contract for their sale if the seller is a merchant with respect to goods of that kind." Id. § 2-314. Merchantability with respect to software means, inter alia, that the program is fit for the ordinary purposes for which it is to be used. Id. § 2-314(2)(c). Section 2-315 provides:

Where the seller at the time of contracting has reason to know any particular purpose for which the goods are required and that the buyer is relying on the seller's skill or judgement to select or furnish suitable goods, there is unless excluded or modified . . . an implied warranty that the goods shall be fit for such purpose.

Id. § 2-315.

However, it should be noted that § 2-316(2) allows the Code's implied warranties to be modified or excluded from the transaction altogether, thus potentially limiting or eliminating liability for breach of such warranties. See id. § 2-316(2).
16 See Kerry M.L. Smith, Comment, Suing the Provider of Computer Software: How Courts Are Applying U.C.C. Article Two, Strict Tort Liability, and Professional Malpractice, 24 Willamette L. Rev. 743, 747-51 (1988). Although computer software contains both tangible and intangible components, the "intangible program" generally is considered the primary function of all software because the tangible object embodied by the program is useless without the intangible commands that make the software work. See Horovitz, supra note 12, at 132-33 ("The medium should not be confused with the message."). Contra Ad- vent Sys. v. Unisys Corp., 925 F.2d 670, 675 (3d Cir. 1991) (focusing on "tangibility" and "moveability" of programs contained in physical mediums). The Third Circuit noted that the computer program, like music or literature, is an intellectual product, and like these other literary works, can become "goods" once embodied in a physical medium. Id. However, the professional expertise employed in software creation arguably warrants the classification of software as services. See Walker, supra note 5, at 12-13.
Uniform Commercial Code. As early as 1985, the Ninth Circuit acknowledged that the sale of software packages could constitute a sale of goods. However, the court also recognized that the nature of software requires a fact-specific analysis of the essence of the transaction in order to determine whether, in any given case, software is primarily a good or a service. Courts faced with this issue have concentrated on "whether the goods aspect or the service aspect predominated," rather than simply looking to the physical attributes of the software itself. In making this determination, certain variables have consistently been recognized by courts. If, for example, software is provided along with a corresponding hardware package, it is more likely to be considered a sale of goods, but similar transactions that include custom programming have been characterized as services.

---

\[^{17}\text{See, e.g., Triangle Underwriters, Inc. v. Honeywell, Inc., 604 F.2d 737, 742-43 (2d Cir. 1979) (hardware included with computer system rendered "essence of contract sale of goods"); Chatlos Sys. v. National Cash Register Corp., 479 F. Supp. 738, 742 (D.N.J. 1979) (lease agreement for hardware/software was "sale of goods" subject to U.C.C.), aff'd, 635 F.2d 1081 (3d Cir. 1980); Analysts Int'l Corp. v. Recycled Paper Prods., Inc., No. 85 C 8637, 1987 WL 12917, at *3 (N.D. Ill. 1987) (custom software qualified as goods since any other interpretation rendered contract useless). But see Data Processing Servs., Inc. v. L.H. Smith Oil Corp., 492 N.E.2d 314, 318 (Ind. Ct. App. 1986) (holding that contract for sale of software created for specific customer, unaccompanied by sale of hardware, was contract for services).}\]

\[^{18}\text{ RRX Indus. v. Lab-Con, Inc., 772 F.2d 543, 546 (9th Cir. 1985).}\]

\[^{19}\text{ Id. The court, noting that the nature of "software . . . var[i]es depending on the needs of the individual consumer," found the underlying transaction to be a sale of goods because the services provided with the software were only "incidental" to the sale of the program. Id.}\]

\[^{20}\text{Anthony Pools v. Sheehan, 455 A.2d 434, 438 (Md. 1983) (quoting GEORGE I. WALLACH, THE LAW OF SALES UNDER THE UNIFORM COMMERCIAL CODE § 11.05[3], at 11-28 (1981)) (describing predominant purpose test applied to mixed or hybrid transactions).}\]

\[^{21}\text{John M. Conley, Non-Contract Theories of Recovery Against Vendors of Defective Software 3 (PLI Patents, Copyrights, Trademarks, and Literary Property Course Handbook Series No. 230, 1986) ("courts have focused . . . on whether the transfer of goods or the provision of a service is at the heart of the contract"); see also supra note 17 (cases applying predominant purpose test to software).}\]

\[^{22}\text{See supra note 17.}\]

\[^{23}\text{See, e.g., Data Processing Servs., Inc. v. L.H. Smith Oil Corp., 492 N.E.2d 314, 318-19 (Ind. Ct. App. 1986) (custom software represented programmer's skill and was thus part of contract for services); Micro-Managers, Inc. v. Gregory, 434 N.W.2d 97, 100 (Wis. Ct. App. 1988) (custom programming rendered contract one for services). In Data Processing Servs., the court emphasized that "[t]he sale of computer hardware or generally available standardized software was not . . . involved." Data Processing Servs., 492 N.E.2d at 319.}\]
II. PRODUCTS LIABILITY LAW AND SOFTWARE

A. Is Software a “Product” for Purposes of Strict Liability?

Just as article 2 applies only to “transactions in goods,”24 strict products liability under section 402A of the Restatement (Second) of Torts (“Restatement”) applies only to “products.”25 Although courts have repeatedly held software to be a “good” within the purview of the Uniform Commercial Code, they have not yet directly addressed the issue of whether software is a “product” under the Restatement.26 Some commentators have simply taken for granted that if software is a “good,” it must necessarily be a “product.”27 However, because these words are terms of art used by the Code and the Restatement, this assumption may not be valid.28 Instead, it is helpful to compare the analyses used by courts in interpreting both the Code and the Restatement for clas-

24 U.C.C. § 2-102; see also supra notes 17-23 and accompanying text (discussing applicability of Uniform Commercial Code to “transactions in goods”).
25 Restatement, supra note 9, § 402A(1); see Hall, supra note 3, at 375; Lanoue, supra note 4, at 447 n.37. Traditionally, strict products liability has been applied only in situations involving defective products. See La Rossa v. Scientific Design Co., 402 F.2d 937, 942 (3d Cir. 1968). “Professional services do not ordinarily lend themselves to [strict products liability] because they lack the elements which gave rise to the doctrine.” Id.; see also W. PAGE KEETON ET. AL., PROSSER AND KEETON ON THE LAW OF TORTS § 104, at 720 (5th ed. 1984) (product use overlap with service provided, primary problem applying strict liability to services).
26 See supra note 5.
27 See, e.g., CONLEY, supra note 21, at 20 (concern that software may not be a product for purposes of strict products liability “seems misplaced, given the tendency to treat software as goods for Uniform Commercial Code purposes”); Walker, supra note 5, at 9 (“software may be considered a product, since it . . . may be classified as a good”).
28 Because the words “good” and “product” are often used interchangeably in common speech, it is arguable that the distinction is purely semantical. The dictionary defines a product as “a thing produced by labor” and “the totality of goods ma[de] available.” THE RANDOM HOUSE DICTIONARY OF THE ENGLISH LANGUAGE 1544 (Stuart B. Flexner et al. eds., 2d ed. 1987); see also Restatement, supra note 9, § 402A cmt. c (1965) (noting interchangeable use of “goods” and “products”). The Restatement, discussing the justifications for the imposition of strict liability, states that

the seller, by marketing his product for use and consumption, has undertaken and assumed a special responsibility toward any member of the consuming public who may be injured by it; that the public has the right to and does expect, in the case of products which it needs and for which it is forced to rely upon the seller, that reputable sellers will stand by their goods.

Id. (emphasis added).
29 See ZAMMIT & SAVIO, supra note 7, at 2-3. “[I]t does not necessarily follow that because software is viewed as goods for commercial law purposes, it should be viewed as a product for personal injury or property damages purposes.” Id.
sifying items which have tangible and intangible qualities.29

As previously discussed, courts considering whether software should be classified as a good for purposes of the Uniform Commercial Code look to the essence of the transaction or its predominant purpose.29 Under this so-called “dominant thrust” analysis,31 a major consideration is whether the contract was primarily for the programmer's knowledge and skill or for the tangible medium that stores the information.32 Under this approach, a program that is made-to-order is more likely to be considered a service than one that is mass-produced.33 A similar approach has been taken by courts under strict products liability theory in cases involving mixed or hybrid transactions.34 Courts look to the facts to determine whether the bargain was entered into primarily for the skills of the programmer or for the software itself.35 In applying this test to computer software, the determinant factor would probably be

29 See id. at 3 (analogizing cases involving characterization under the Restatement and characterization under the Code for hybrid transactions).
30 See supra notes 17-23 and accompanying text.
32 See id. at 319 (concluding custom-made software is a service because “[w]hile a tangible end product . . . may be involved incidentally . . . , it is the skill and knowledge of the programmer which is being purchased in the main”).
33 See, e.g., L.H. Smith Oil Corp., 492 N.E.2d at 318-19 (custom program held to be contract for services not subject to Code); Micro-Managers, Inc. v. Gregory, 434 N.W.2d 97, 100 (Wis. Ct. App. 1988) (custom computer programming contract considered contract for services under Code).
34 See infra notes 35-37 and accompanying text.
35 See John J. Fossett, The Development of Negligence in Computer Law, 14 N. Ky. L. Rev. 289, 307 (1987). Several variables are considered in determining whether or not software is a “product” against which strict products liability may be applied. Id. “Of central importance to this determination is whether the seller or manufacturer of the product has employed mass-marketing techniques (e.g., mass production, large-scale distribution of finished products, and mass market advertising) . . . .” Id. However, the courts’ reluctance to impose products liability against mass-produced publications demonstrates that mass-marketing is not necessarily determinative. See Bayman, supra note 6, at 564-69. This is true even though the publications probably reach far more purchasers. Id.

For early computers to function properly, custom programs developed by experts were necessary; their use was thus an expensive undertaking. See Ronald N. Weikers, Comment, “Computer Malpractice” and Other Legal Problems Posed by Computer “Vaporware”, 33 Vill. L. Rev. 835, 871-73 (1988). However, technological advances revolutionized software, transforming it from a system-dependant component part to a distinct product capable of functioning in conjunction with many different computer systems. Id. This opened the door to the “mass-market[ing]” of “canned software” which changed the characterization of computer programs from a mere component part of a specific system to “a product that is distinct from the computer hardware in which it is used.” Id. at 873-74.
“the form of the software or the manner in which it is marketed.”

Thus, software that is produced in large quantities is more likely to be considered a product than that which is custom-made. Given the similarities between these two analyses and the common factors that have been emphasized by the courts, it is suggested that software which passes as a “good” under the Code would also be considered a “product” under the Restatement.

Additionally, it is interesting to note that the damages already available to an aggrieved buyer under the Uniform Commercial Code are comparable to those available under strict products liability theory. Section 2-715 of the Code specifies that a buyer should be allowed to recover for “injury to person or property proximately resulting from any breach of warranty.”

Two comments to the Code indicate that what must be proved in an action based on breach of warranty is comparable to that which must be proved under the Restatement.

B. Application of Strict Products Liability

Assuming that software is classified as a product, strict products liability for software sold in a defective or unreasonably dangerous condition may be imposed upon a seller or manufacturer, but only if the software reached the consumer without substantial change, the consumer used the software in a reasonable fashion,

---

36 Zammit & Savio, supra note 7, at 2-3.
37 See id. at 3.
38 See Conley, supra note 21, at 4 (damages available under the Code comparable to those of other tort theories).
39 See U.C.C. § 2-715(2)(b) (emphasis added).
40 Compare Restatement, supra note 9, § 402(A) (1965) with U.C.C. § 2-314 cmt. 13 and id. § 2-715 cmt. 5.
41 Restatement, supra note 9, § 402A. The Restatement provides the following:
(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.

Id. Strict products liability in tort had its genesis in Greenman v. Yuba Power Prods., Inc., 377 P.2d 897, 900 (Cal. 1962), which announced that strict liability would exist when a man-
and personal injury or property damage was actually caused by the software.\textsuperscript{43} Defective computer software may result from the manufacturing process, the failure to warn of the dangers involved in its use, or the flawed design of the software.\textsuperscript{44}

The first of these defects, the manufacturing defect, is the most easily detected because a program that does not conform to the manufacturer's specifications can be readily discovered through careful monitoring of the manufacturing process.\textsuperscript{46} The second type of defect, the failure to warn of the dangers involved in the use of software, seems to be within the complete control of the manufacturer;\textsuperscript{45} however, ensuring that warnings are conspicuous...

---

\textsuperscript{43} See Dix W. Noel, Defective Products: Abnormal Use, Contributory Negligence, and Assumption of Risk, 25 VAND. L. REV. 93, 95-105 (1972) (manufacturer or supplier not liable when injury resulted from abnormal use of product); see also W. Page Keeton, Manufacturer's Liability: The Meaning of "Defect" in the Manufacture and Design of Products, 20 SYRACUSE L. REV. 559, 563 (1969) (plaintiff must demonstrate injury not caused by plaintiff's own "misuse").

\textsuperscript{44} See Restatement, supra note 9, § 402A (1); see also Richard Raysman & Peter Brown, Strict Product Liability for Software and Data, N.Y.L.J., Sept. 15, 1988, at 3-4. Generally, recovery in strict products liability can be had only for bodily injury or property damage, not for pure economic loss. Id.; Birnbaum, supra note 3, at 140.

In Champeau v. Fruehauf Corp., 814 F.2d 1271, 1277-78 (8th Cir. 1987), the court held that the plaintiff failed to establish causation and therefore declined to impose liability on the manufacturer of a computer-assisted antilock braking system. Id.; see also Murray, supra note 41, at 653 (product defect, which at very least is contributory cause of harm, is threshold requirement for cause of action in strict liability). See generally Michael D. Scott, Computer Law § 7.12 (1985) (discussing strict liability).

\textsuperscript{45} See KEETON ET AL., supra note 25, § 99, at 694-98; see also Fluor Corp. v. Jeppeson & Co., 216 Cal. Rptr. 68, 71 (Cal. Ct. App. 1985) ("defect" encompasses several "injury-producing deficiencies").


\textsuperscript{47} See Restatement, supra note 9, § 402A cmt. j. The Restatement provides that "[where warning is given, the seller may reasonably assume that it will be read and heeded; and a product bearing such a warning, which is safe for use if it is followed, is not in defec-
may be a more difficult task than at first appears. Computer software is integrated into many products and often is not apparent to the user; thus warnings accompanying the software may easily be overlooked. The third type of defect is that which occurs in the design of the software itself. A design may be defective if it is dangerous beyond the contemplation of an ordinary consumer possessing ordinary knowledge or if its danger outweighs its utility.

A number of arguments may be made in favor of applying strict products liability to computer software. If it is held inapplicable, plaintiffs would be forced to rely on negligence principles. This would give software manufacturers added protection from liability by dramatically increasing the plaintiff's burden of proof. In fact, case law reveals that plaintiffs have rarely been successful in negligence actions against computer manufacturers.

Because strict liability is often imposed upon manufacturers of the mechanical parts of malfunctioning products, it seems inconsis-

tive condition, nor is it unreasonably dangerous.” Id. Thus, there are situations where an adequate warning removes a product from strict liability since the requisite defect or danger is lacking. See Murray, supra note 41, at 657-58.

See Fossett, supra note 35, at 289 (noting many uses of computer programs); Lanoue, supra note 4, at 448. It may be particularly difficult to perceive potential dangers associated with “an industrial process . . . not understood.” Id.; see also Hall, supra note 3, at 374 (computers even permeate games and “special effects”).

See Birnbaum, supra note 3, at 139, 150. Simply adding warning labels may not be sufficient if consumers would not read them. Id. at 139. A warning in a users' manual may also be inadequate because “it is commonly known that many software users do not read users' manuals.” Id.

See Stephen R. Brenneman, Computer Malfunctions—What Damages May Be Recovered in a Tort Product Liability Action, 2 SANTA CLARA COMPUTER & HIGH TECH. L.J. 271, 279 (1986). When software functions improperly, approximately half of the mistakes “are the result of defects in design.” Id. Imposing strict liability for lingering design defects in a line of software could cripple software manufacturers because these defects endure beyond a program's introduction into the marketplace. See Lawrence, supra note 45, at 15 (strict liability may drive medical software manufacturers out of business).

See KEETON ET AL, supra note 25, § 99, at 698-700. Much of the difficulty in products liability centers around the meaning of the term “design defect.” Id. at 698.

See infra notes 52-56 and accompanying text.

See Birnbaum, supra note 3, at 145-46 (discussing negligence standard applied to computer software).

Id. “This burden would include finding the mistake in thousands of electronic bits . . . .” Id. at 146; see also Lanoue, supra note 4, at 448. Even a knowledgeable user could find it nearly impossible “to isolate the specific act of negligence” that caused the harm. Id.

See Lawrence B. Levy & Suzanne Y. Bell, Software Product Liability: Understanding and Minimizing the Risks, 5 HIGH TECH. L.J. 1, 9 (1990); see also Conley, supra note 21, at 10 (“negligence theories are of no use in computer cases”).
tent and fundamentally unfair to allow manufacturers of the same product's defective software components to be held only to a negligence standard. Moreover, some commentators have asserted that imposing strict liability on manufacturers profiting from software sales would help provide incentive for ensuring the safety of computer programs.

III. Ramifications of Strict Products Liability

A. Detrimental Effects of Imposing Strict Liability on the Computer Industry

Initially, public policy appears to warrant the imposition of strict liability to computer software. However, applying strict products liability against software manufacturers may have "far reaching implications" for our computerized society. Because the reasonableness of the manufacturer's actions is irrelevant in strict liability, software manufacturers would in effect be compelled to guarantee the safety of their programs. This increased exposure to liability would cause insurance companies to raise their premiums, resulting in costlier computer programs. Increased insurance rates could inhibit the development of innovative software because, with insurance costs on established products currently escalating to rates as high as 1500% per year, it seems unlikely that

55 See Lanoue, supra note 4, at 449. For example, "[i]t would be undesirable to permit an injured consumer to collect under strict products liability for a defective steering mechanism, but not for a defective computer program in the car which may have caused the same injuries." Id.

56 See Smith, supra note 16, at 759; see also LAUTSCH, supra note 3, at 264 (imposition of strict liability on software will "compel designers and manufacturers of commercial goods to design their products with the safety of the user in mind"); Birnbaum, supra note 3, at 143 (strict liability promotes product safety).

57 See Lanoue, supra note 4, at 447-49 (applying strict liability policies to software). See generally Birnbaum, supra note 3, at 145-50 (justifications underlying imposition of strict liability to software).

58 See Walker, supra note 5, at 14.

59 See RESTATEMENT, supra note 9, § 402A (2)(a); see also Birnbaum, supra note 3, at 140 ("liability is absolute").

60 See Walker, supra note 5, at 1 (higher insurance costs result in increased prices and manufacturing stoppage); Frank Lowenstein, Software Liability; Should Software Publishers Be Liable for Bugs?, TECH. REV., Jan. 1987, at 9-10 (cost of liability "not built into software pricing"); Jill Andresky et al., A World Without Insurance?, FORBES, July 15, 1985, at 40 (tremendous premium increases for greatly reduced coverage). The increase in products liability actions has caused an insurance crisis that may lead to widespread unavailability of insurance, a "chilling prospect." Id. at 43.
manufacturers of high-risk or new products would be able to obtain or afford the necessary insurance coverage.\textsuperscript{61}

The application of strict liability in this area may force producers of beneficial technology out of business entirely because of the fear of potential liability\textsuperscript{62} and the burdens of defending claims.\textsuperscript{63} In fact, one software supplier has already withdrawn some of its products from the market for fear of potential litigation.\textsuperscript{64} The effect of compelling software manufacturers to follow the lead of manufacturers of medical equipment, vaccines, and drugs in ceasing production of beneficial products to avoid potential liability would be unfortunate.\textsuperscript{65} If the makers of products considered safe are forced to retreat from the marketplace, it appears that new or experimental products have little hope of survival.\textsuperscript{66} Thus, although the merits of applying strict liability are in many cases unquestionable, it is submitted that the costs of imposing strict products liability against software manufacturers may at times far outweigh the benefits.\textsuperscript{67}

\textsuperscript{62} See id. at 22. In the medical technology industry, for example, some manufacturers of medical equipment, vaccines, and drugs, fearing liability, have already ceased production of these beneficial products. Of all the manufacturers that were producing the vaccines required for school children, only one company continues to manufacture the combination measles, mumps, and rubella vaccine. \textit{Id.}
\textsuperscript{63} Id. at 20. Even manufacturers who have consistently prevailed against strict liability claims have found the legal fees associated with defending the claims too much to bear. For example, G.D. Searle & Co. ceased selling its “Copper 7” and “Tatum T” intrauterine devices in the United States because, although these products were considered “the safest . . . on the market,” insurance coverage became unavailable. \textit{Id.} Moreover, despite Searle’s victories in eight lawsuits, the legal costs for four of the suits accounted for nearly 14% of the company’s sales for 1985. \textit{Id.}

The “most dramatic” results of products liability claims are the bankruptcies of A.H. Robins, manufacturer of the poorly designed “Dalkon Shield,” and Manville Corporation, which was buried by an avalanche of asbestos claims. \textit{Id.}
\textsuperscript{64} See \textit{Slind-Flor, supra} note 9, at 3.
\textsuperscript{65} See Walker, \textit{supra} note 5, at 15 (application of products liability law to emerging products “may be too . . . [much] for the commercial system to bear”); see also \textit{supra} note 62 (strict products liability may have adverse effect on medical technology industry).
\textsuperscript{66} See \textit{supra} note 63 and accompanying text.
\textsuperscript{67} See \textit{supra} notes 60-63 and accompanying text. “The prospect of paying out huge judgments is all companies need to tip the balance against staying in the business. If those that remain were to raise prices enough to cover potential liabilities, the [products] . . . they make could become prohibitively expensive.” Brody, \textit{supra} note 61, at 22.

C. Defective Information: First Amendment Concerns

Over the past two decades, courts began to address the issue of the applicability of strict products liability when injury is caused by defective information.68 Although manufacturers have attempted to avoid this type of liability by claiming they merely provided a service,69 courts have at times allowed such recovery by focusing on the nature of the defective information, its intended use, and the resultant injury in order to determine whether or not the information is a product.70

In fact, courts considering analogous cases have held that intangible information embodied in a physical medium is a product.
subject to strict products liability. This line of cases is particularly relevant to the analysis of strict liability and software because computer programs generally "either contain[] information or assist[] in the generation or manipulation of information."

However, because computer software is often a vehicle for disseminating information, the application of strict liability in this context may violate the First Amendment. For this reason, most jurisdictions have declined to hold distributors of other forms of information strictly liable. In fact, no court has extended strict liability to the distribution of ideas or information physically embodied in books or other publications. Reasoning that the prospect of strict liability would "chill expression" and be "inconsistent with free speech principles," courts generally separate the physical properties of the medium from the information contained

71 See, e.g., Brocklesby v. United States, 767 F.2d 1288, 1295 (9th Cir.) (publisher of instrument approach procedures strictly liable for injuries caused by defective chart), cert. denied, 474 U.S. 1101 (1985); Salomey v. Jeppeson & Co., 707 F.2d 671, 676-77 (2d Cir. 1983) (mass-produced navigational charts are products); Aetna Casualty & Sur. Co. v. Jeppeson & Co., 642 F.2d 339, 341 (9th Cir. 1981) (products liability applies to publisher of defective instrument approach charts); Fluor, 216 Cal. Rptr. at 71 (aeronautical charts are products).

The basis of liability in these cases was an analogy of the charts to other tools of navigation, which, if defective, created a substantial risk of harm. See Jones v. J.B. Lippincott Co., 694 F. Supp. 1216, 1217 (D. Md. 1988); see also Winter, 938 F.2d at 1036 (analogizing aeronautical charts to compass and computer software, both geared to yield specific results). Computer programs may similarly be viewed as highly technical instruments, capable of causing great injury, if defective. See Smith v. Linn, 563 A.2d 123, 127 (Pa. Super. Ct. 1989) (complex materials adopt qualities of "product[s]"), aff'd, 587 A.2d 309 (Pa. 1991).

72 Levy & Bell, supra note 54, at 6; see also Nimmer, supra note 6, ¶ 7.06[2][b], at 7-26 (software may be characterized as information).

73 See Bayman, supra note 6, at 576. "imposing strict liability on writers and publishers under products liability law effectively would nullify . . . constitutional constraints . . . " Id.


75 See Lippincott, 694 F. Supp. at 1217; Linn, 563 A.2d at 126-27.

therein.\textsuperscript{77} In \textit{Winter},\textsuperscript{78} for instance, the Ninth Circuit rejected a products liability claim against the publisher of a reference guide to mushrooms that misidentified poisonous varieties of the fungus as edible.\textsuperscript{79} Granting summary judgment to the defendant, the court stated that the “book [contained] pure thought and expression” and was not subject to products liability.\textsuperscript{80} Thus, although allocating the costs of injuries resulting from erroneous information is an appealing concept, courts recognizing “the high priority [placed] on the unfettered exchange of ideas” are reluctant to apply strict products liability in this context.\textsuperscript{81} Because of this reluctance, courts may determine that applying strict products liability to information supplied through computer software creates too great a burden on First Amendment protections.\textsuperscript{82}

\textbf{IV. Suggested Approach}

In deciding whether strict products liability should be applied in any given case involving software, courts should be mindful of the far reaching effects of their decisions.\textsuperscript{83} The widespread application of strict products liability to cases involving defective software could have a crippling effect on technology and on the computer industry as a whole.\textsuperscript{84} Furthermore, because computer programs are media used for the dissemination of ideas and information, the imposition of strict liability could conflict with established constitutional protections of speech under the First Amend-

\textsuperscript{77} \textit{Lippincott}, 694 F. Supp. at 1217; see also \textit{Winter}, 938 F.2d at 1034 (distinguishing “thoughts” from “substance”); \textit{Cardozo}, 342 So. 2d at 1056 (differentiating between physical elements of books and ideas within).

\textsuperscript{78} 938 F.2d 1033 (9th Cir. 1991).

\textsuperscript{79} Id. at 1033-34.

\textsuperscript{80} Id. at 1036. The court further held that because a publisher is under no duty to verify intellectual content, the plaintiffs’ negligence action also failed. Id. at 1036-37.

\textsuperscript{81} Id. at 1035. The court acknowledged that strict liability’s effect of inhibiting the development of new products is more acceptable than the potential loss of “the latest ideas and theories.” Id.

The concept of applying First Amendment protection to computer programs is not outlandish; computer output has already been analogized to a person’s “written expression of ideas.” Bayman, \textit{supra} note 6, at 576. Further, software that is able to imitate human thinking processes already exists. See Steven J. Frank, \textit{Tort Adjudication and the Emergence of Artificial Intelligence Software}, 21 SUFFOLK U. L. REV. 623, 624, 640-41 nn.71-76 (1987); see also Freed, \textit{supra} note 10, at 280 (computer output compared to “human information processing”).

\textsuperscript{82} See \textit{supra} notes 68-81 and accompanying text.

\textsuperscript{83} See Walker, \textit{supra} note 5, at 14.

\textsuperscript{84} See \textit{supra} notes 57-67 and accompanying text.
In light of this potential conflict, it is suggested that courts employ a three-step analysis when determining whether strict products liability is appropriate for injuries arising from defective software.

The initial step in this analysis is to determine whether the computer program should be characterized as a "product" within the meaning of the *Restatement*. If the software is found to be a service, rather than a product, the strict liability claim should be dismissed. Alternatively, if the software is found to be a product, the second step in the analysis would be to determine whether the expression of ideas or information contained in the program is entitled to First Amendment protections. If the information is not constitutionally protected, the final inquiry would be an examination of the policies underlying strict products liability law coupled with a balancing of the risks of the software's use against its beneficial purposes to determine whether the application of this type of liability is appropriate in the particular case. These policies include (1) allocating risk to those most able to protect themselves, (2) compensating for injury, (3) alleviating the evidentiary burden of proving negligence, (4) compensating for a purchaser's inability to inspect and his or her consequent reliance on the manufacturer's expertise, and (5) discouraging the marketing of

---

85 See *supra* notes 72-81 and accompanying text.
86 See *supra* notes 24-37 and accompanying text.
88 See *infra* notes 89-93 and accompanying text.
89 See Escola v. Coca Cola Bottling Co., 150 P.2d 436, 441 (Cal. 1944) (Traynor, J., concurring). The cost of injury may be overwhelming to an injured person, and this allocation of risk places the financial burden on the manufacturer, who is better equipped to afford it. Id. The basis for "cost-spreading" is that the manufacturer has the ability to insure the product against injury and to pass the expenses along to "the public as a cost of doing business." Id.; see also John. W. Wade, *On the Nature of Strict Tort Liability for Products*, 44 Miss. L.J. 825, 826 (1973) (manufacturers distribute costs to consumers through pricing).
90 See Birnbaum, *supra* note 3, at 143; Keeton et al., *supra* note 44, § 1, at 5.
92 See Lawrence, *supra* note 45, at 11 ("superior knowledge" enables manufacturer to detect and remove defects); see also Fossett, *supra* note 35, at 306 (strict liability is imposed on party best able discern and remove defects).
defective products. Under this flexible, fact-specific analysis, if a court determines that the policies weigh in favor of strict liability, relief should be granted to the injured party.

CONCLUSION

With the expanding presence of computers in our society, injuries to persons and property due to software malfunction are certain to occur with greater frequency. As a result, courts inevitably must consider whether strict products liability should be imposed upon the manufacturers and suppliers of defective computer programs. Because computer technology has not yet evolved to the point where software can be made completely safe, utilization of the suggested three-step analysis would provide predictability and help ensure even application of the law. Until a greater level of certainty in computer design can be achieved, the use of this balancing test will encourage manufacturers to develop safer products while limiting the potentially adverse impact on emerging beneficial technology.

Lori A. Weber

---


94 Bayman, supra note 6, at 562-63. Flexible application of strict liability is important because in many situations, the court's determination enormously affects innovation, and may in fact "determine whether a given product or industry survives." Id.